FRAMEWORK CONCEPTUAL DE CIBERSEGURIDAD PARA APLICACIONES DE INTERNET DE LAS COSAS

JOHAN SMITH RUEDA RUEDA

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB
FACULTAD DE INGENIERÍA
MAESTRÍA EN TELEMÁTICA – MODALIDAD INVESTIGACIÓN
GRUPOS TECNOLOGÍAS DE INFORMACIÓN Y PENSAMIENTO SISTÉMICO
BUCARAMANGA, COLOMBIA
MARZO DE 2018

FRAMEWORK CONCEPTUAL DE CIBERSEGURIDAD PARA APLICACIONES DE INTERNET DE LAS COSAS

JOHAN SMITH RUEDA RUEDA

Trabajo de grado para optar al título de Magíster en Telemática, modalidad Investigación

Director
Jesús Martin Talavera Portocarrero, Ph.D.
Ingeniero de desarrollo
Nuance Communications, Brasil

Co-director
Ing. José Daniel Cabrera Cruz
Universidad Autónoma de Bucaramanga, Colombia

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA - UNAB
FACULTAD DE INGENIERÍA
MAESTRÍA EN TELEMÁTICA – MODALIDAD INVESTIGACIÓN
GRUPOS TECNOLOGÍAS DE INFORMACIÓN Y PENSAMIENTO SISTÉMICO
BUCARAMANGA, COLOMBIA
MARZO DE 2018

AGRADECIMIENTOS

Quiero agradecer de manera muy especial a las personas e instituciones que contribuyeron de alguna u otra forma en la realización de este proyecto.

A Jesús Martin Talavera Portocarrero, tutor y director de este trabajo de investigación, por su orientación en esta investigación, sus continuas enseñanzas, y sus contribuciones en mi proceso de formación como investigador.

Al profesor José Daniel Cabrera Cruz, co-director de este trabajo, por sus orientaciones metodológicas y recomendaciones para la elaboración de este documento.

Al Centro de Excelencia y Apropiación en Internet de las Cosas – CEA-IoT, proyecto soportado por las instituciones que apoyaron este trabajo: el Ministerio de Tecnología de la Información y Comunicaciones – MinTIC de Colombia y al Departamento Administrativo de Ciencia, Tecnología e Innovación – Colciencias, a través del Fondo Nacional de Financiamiento para la Ciencia, Tecnología y la Innovación Francisco José de Caldas (ID Proyecto: FP44842-502-2015).

A la fundación OWASP, por sus aportes de libre acceso en temas de ciberseguridad con el propósito que las organizaciones puedan concebir, desarrollar, adquirir, operar y mantener aplicaciones que se puedan confiar, y que aboga por acercarse a la seguridad de las aplicaciones de forma integral.

Johan Smith Rueda Rueda

DEDICATORIA

Quiero dedicar este trabajo a mi madre, que con amor y paciencia me enseñó a escribir mis primeras palabras. A mi padre, por su apoyo incondicional y por darme un gran consejo que hasta hoy he seguido: “Nunca dejes de aprender”. A mis hermanos, por su colaboración en algunas de mis responsabilidades mientras dedicaba tiempo a mi formación y al desarrollo de esta investigación.

Johan Smith Rueda Rueda

«*La verdadera ignorancia no es la ausencia de conocimientos, sino el hecho de rehusarse a adquirirlos*».

–Karl Popper

LISTA DE ACRÓNIMOS

AR	Arquitectura de referencia
AS	Arquitectura de software
ATAM	*Architecture Tradeoff Analysis Method*
IoT	*Internet of Things*
IoT-CyDM	*IoT Cybersecurity Domain Model*
SMITH	*Security Management in Internet of THings*

Framework conceptual de ciberseguridad para aplicaciones de internet de las cosas

Johan Smith Rueda Rueda[1]
Jesús M. Talavera Portocarrero[2], José Daniel Cabrera Cruz[2]
[1]Estudiante de maestría, [2] Tutores de la investigación
Universidad Autónoma de Bucaramanga - UNAB
Marzo de 2018

RESUMEN

El internet de las cosas – IoT, es uno paradigmas tecnológicos con rápido crecimiento en los últimos años, en el que objetos inteligentes o cosas, interactúan entre sí y con recursos físicos y/o virtuales a través de Internet. Junto con este crecimiento hace resonancia uno de los retos que presenta este paradigma, la seguridad de aplicaciones IoT.

Este trabajo de investigación parte del problema que existen aplicaciones IoT inseguras por la falta de guías que orienten a los desarrolladores en la implementación del dominio de la ciberseguridad en la fase de diseño y la evaluación de estas. La hipótesis planeada es que, mediante un *framework*, compuesto por diferentes tipos de modelos, se puede orientar al equipo de desarrollo sobre cómo considerar ciberseguridad en las aplicaciones IoT.

Desde este punto de partida, en este trabajo se propone un *framework* conceptual de ciberseguridad para aplicaciones IoT, llamado *SMITH Framework*. Este *framework* está compuesto por dos modelos: el primero, un modelo de gestión de la ciberseguridad cuyo propósito es orientar a los desarrolladores de aplicaciones IoT las consideraciones de ciberseguridad que deben tenerse en cuenta desde la fase de diseño de una solución IoT; el segundo, un modelo conceptual del dominio de la ciberseguridad en el que se presenten seis componentes de seguridad y su relación con el dominio de IoT.

Para verificar la hipótesis planteada, se hizo una validación del *SMITH Framework* basada en el método ATAM, en el que se diseñó una aplicación IoT orientada por elementos del framework propuesto. Los resultados arrojados permitieron conocer que sí es posible orientar al equipo de desarrollo en la implementación de la ciberseguridad en la fase de diseño de una aplicación IoT, confirmando la hipótesis planteada.

Palabras claves: aplicaciones IoT, ciberseguridad, framework conceptual, Internet de las cosas, modelo conceptual, modelo de ciberseguridad.

Cybersecurity Conceptual Framework for the Internet of Things Applications

Johan Smith Rueda Rueda[1]
Jesús M. Talavera Portocarrero[2], José Daniel Cabrera Cruz[2]
[1]Estudiante de maestría, [2] Tutores de la investigación
Universidad Autónoma de Bucaramanga - UNAB
Marzo de 2018

ABSTRACT

The Internet of things - IoT, is a technological paradigm with rapid growth in recent years, in which smart objects or things interact with each other and with physical and/or virtual resources through the Internet. To all this boom resonates one of the challenges presented by this paradigm, the security of IoT applications.

This research work based on the problem that there are insecure IoT applications due to the lack of guides that guide developers in the implementation of the cybersecurity domain in the design phase and the evaluation of these. The planned hypothesis is that, through a framework, composed of different types of models, it can guide the development team on how to consider cybersecurity in IoT applications.

From this starting point, this work proposes a conceptual cybersecurity framework for IoT applications, called SMITH Framework. This framework is composed of two models: SMITH Model and IoT-CyDM. The first is a cybersecurity management model whose purpose is to guide the IoT application developers the cybersecurity considerations that they must consider in the design phase of an IoT solution. The second is a model of the domain of cybersecurity, which presents six security components and their relationship with the domain of IoT.

To verify the hypothesis, we made validation of the SMITH Framework based on the ATAM method, in which we design an IoT application. Elements of the proposed framework guided the design of this IoT application. The results showed that it is possible to guide the development team in the implementation of cybersecurity in the design phase of an IoT application, confirming the proposed hypothesis.

Keywords**:** Conceptual framework, conceptual model, cybersecurity, Internet of things, IoT applications, security model.

CONTENIDO

Pág.

ÍNDICE DE FIGURAS

Pág.

ÍNDICE DE TABLAS

Pág.

INTRODUCCIÓN

Uno de los retos que enfrenta el internet de las cosas ó IoT, por sus siglas en inglés, son las medidas de seguridad implementadas en las aplicaciones IoT que salvaguarden la información que estas generan, trasmiten y almacenan. Este trabajo de investigación buscar contribuir en la solución de este problema con la propuesta de un *framework* conceptual, llamado *SMITH Framework*, el cual se compone de dos modelos: el *SMITH Model* y el modelo IoT-CyDM. El propósito del *SMITH Framework* es orientar a los equipos de desarrollo de aplicaciones IoT en la implementación de componentes de ciberseguridad desde la fase de diseño, y la evaluación de aplicaciones ya construidas. Este documento se divide en siete capítulos:

En el capítulo 1, se presenta la descripción general del proyecto, donde se presenta el problema de investigación, se introduce el contexto de este y explica dos razones que lo fundamentan. También se presenta la motivación de este trabajo, en el que se presentan y argumentan las dos formas en que se abordó el problema dando origen a los resultados de este trabajo de investigación.

En el capítulo 2, se presenta el marco referencial que contiene la fundamentación teórica de esta investigación. En la sección 2.1, llamado marco conceptual, se enumeran los conceptos relevantes tratados en esta investigación. La sección 2.2 corresponde al marco teórico, en el que se presenta las teorías que fundamentan este trabajo organizadas en cuatro áreas del conocimiento: ingeniería del software, ciberseguridad, internet de las cosas y computación distribuida. En la sección 2.3 se expone el estado del arte acerca de dos temáticas: la primera, los *frameworks* de seguridad propuestos para aplicaciones IoT; la segundo, sobre el estado actual de la ciberseguridad en IoT. Seguidamente, en la sección 2.5, se presenta el marco normativo y estándares considerados en este trabajo. En la sección 2.6, el marco contextual y antecedentes y, finalmente, en la sección 2.7 se presentan las consideraciones finales del capítulo.

En el capítulo 3, se detallan los aspectos metodológicos que orientaron esta investigación. En las secciones 3.1 y 3.2, se describe el tipo y enfoque de investigación utilizados y el universo objeto de estudio, respectivamente. En la sección 3.3 se enumeran las técnicas e instrumentos utilizadas para la recolección de información. En sección 3.4 se muestra la metodología usada para el desarrollo de esta investigación, se enumeran y explican las actividades

realizadas para alcanzar los objetivos propuesto, las cuales se organizan en fases y subfases.

A partir de esta parte del documento se presentan los resultados obtenidos en el desarrollo de los objetivos. Para una mejor estructuración del documento y facilidad de lectura, los resultados de este trabajo se presentan en capítulos individuales, uno por cada fase de la metodología propuesta, en la que cada fase que está directamente relacionadas con los objetivos específicos planteados.

En el capítulo 4, se propone un modelo para la gestión de la ciberseguridad en aplicaciones IoT, llamado *SMITH Model*. A lo largo de capítulo se describe el proceso realizado para proponer este modelo. El resultado de este proceso es el *SMITH Model*, un modelo de gestión de ciberseguridad para aplicaciones IoT. Este modelo está compuesto por tres elementos: (i) el modelo de ciberseguridad y la descripción de este; (ii) una guía de buenas prácticas ciberseguridad para el aseguramiento de aplicaciones IoT y (ii) un instrumento de evaluación para validar aplicaciones IoT ya construidas, identificando los componentes de ciberseguridad que no han sido considerados.

En el capítulo 5, se propone el modelo IoT-CyDM, que puede definirse como una representación del dominio de la ciberseguridad a través del lenguaje de modelamiento unificado – UML, un lenguaje semiformal de modelamiento de software y uno de los más conocidos dentro de la ingeniera del software. El modelo IoT-CyDM está compuesto por seis componentes de seguridad: autenticación, autorización, gestión de claves criptográficas, gestión de identidad, disponibilidad y no repudio.

En el capítulo 6, se valida el *framework* propuesto a través de un caso de estudio en el que se diseña una aplicación IoT usando los modelos propuestos. La validación se fundamentó en el método ATAM, un método para la evaluación temprana de arquitecturas de software basada en escenarios (Kazman et al., 1998). El propósito de ATAM es evaluar las consecuencias de las decisiones arquitectónicas en relación con determinados atributos de calidad, permitiendo establecer si una arquitectura particular satisface los atributos de calidad.

En el capítulo 7, se presentan las conclusiones obtenidas y el trabajo futuro de esta investigación. Dentro de las conclusiones descritas en este capítulo se confirma la hipótesis planteada en esta investigación y se presentan las

principales contribuciones realizadas y se listan otras contribuciones resultados de este trabajo de investigación.

Finalmente, se presentan las referencias que sustentaron esta investigación y los anexos.

1. DESCRIPCIÓN GENERAL DEL PROYECTO

En este capítulo se expone la información general del proyecto de investigación, se contextualiza al lector sobre el origen y el enfoque que se dio del problema a abordado en esta investigación. Asimismo, se plantean la forma en que se resolvió el problema y se explica la selección de dichas elecciones, y se presentan los temas centrales de esta investigación.

El resto del capítulo está dividido en seis secciones. En la sección 1.1, se presenta el problema de investigación que se aborda en este trabajo. Primeramente, se contextualiza el problema que se aborda en este trabajo; luego se presentan y explican dos razones que se consideraron para sustentar el problema de investigación. Finalmente se justifica la importancia de realizar esta investigación.

En la sección 1.2, se manifiesta la motivación para realizar este trabajo y el porqué de la forma es que se abordará el problema de investigación. Son dos las motivaciones manifestadas para abordar el problema identificado: la primera, el modelamiento del dominio de la ciberseguridad para IoT, y segundo, el uso de buenas prácticas de la ingeniería del software en proyectos telemáticos.

En la sección 1.3, se formula la pregunta de investigación que guio este trabajo. La hipótesis planteada con el objetivo de responder la pregunto de investigación se presenta en la sección 1.4. En la sección 1.5, describen los objetivos de proyecto de investigación, general y específicos, con los cuales se busca verificar si la hipótesis planteada se cumple o no. Finalmente, en la sección 1.6, se enumeran las contribuciones realizadas en este trabajo de investigación.

1.1. PROBLEMA DE INVESTIGACIÓN

En esta sección, se presenta el problema de investigación que se pretende abordar en este trabajo. Esta sección está organizada de la siguiente manera: en la Subsección 1.1.1 se introduce el contexto del problema, que se sitúa en el proceso de desarrollo de aplicaciones IoT y la seguridad como requisito de calidad de estos; En la subsección 1.1.2, se describe el problema que se abordó en el presente trabajo.

1.1.1 Contexto. Internet de las cosas, *Internet of Things* – IoT, es un paradigma que se basa en la interacción de objetos inteligentes, las cosas, entre sí y con recursos físicos y/o virtuales a través de Internet (Cavalcante et al., 2016). La tendencia de IoT ha ido en aumento en los últimos años y, hoy en día, varias de

las tecnologías emergentes están relacionadas con IoT, como espacios de trabajo inteligentes, hogares conectados, por nombrar algunos (Gartner Inc, 2016a).

Gartner Inc. (2015) previó que para el año 2016 estarían en uso 6.400 millones de cosas conectadas, y que para el 2017, 8.400 millones de cosas conectadas estarán en uso, con un aumento de 31 % desde respecto al año anterior. Otras datos más recientes estima que en el año 2017 hubieron 2196.4 millones de dispositivos IoT conectados, y estima que para el año 2018 habrán 27767.7 millones de dispositivos conectados[1] (Statista, 2018). Asimismo, instituciones como Cisco (2016b) e Intel (2016) han estimado que en 2020 estarán en uso unos 50.000 millones y 200.000 millones de dispositivos conectados, respectivamente.

Internet de las Cosas está transformando el estilo de vida de las personas, nuestra ciudadanía y nuestras ciudades. Además, IoT está transformando empresas y ya está siendo estratégico para ellas (Cisco, 2016a; Essery, Michael, 2016; Gartner Inc, 2016d). Otra estimación de Gartner dice que para el año 2020, más de la mitad de los nuevos procesos y sistemas empresariales incorporarán algún elemento de Internet de las Cosas (Gartner Inc, 2016b). Forrester Research en su informe *Predictions 2018: IoT Moves From Experimentation To Business Scale* afirma que el IoT se convertirá en la columna vertebral del futuro valor del cliente (Forrester, 2017).

Aunque el IoT es un nuevo paradigma, no es una nueva tecnología en sí misma. IoT es un ecosistema integrado por varias tecnologías, algunas de ellas con varias décadas de existencia, otras más recientes más recientes y quizá otras que aparecerán en el futuro. Autores como I. Lee & Lee (2015) afirman que el IoT cuenta con cinco tecnologías esenciales, a saber: (i) *Radio Frequency Identification* – RFID, (ii) *Wireless Sensor Network* – WSN, (iii) sistemas de middleware, iv) computación en nube, y iv) aplicaciones IoT de usuario final. Pero el IoT no se limita a estas cinco tecnologías, el ecosistema IoT incluyen diferentes tecnologías.

Las aplicaciones IoT se componen de procesos de ingeniería del software que permite a las personas interactuar con el resto de la aplicación IoT y sus servicios. La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software (IEEE, 1990). Desde la ingeniería de software se han propuesto varios modelos de desarrollo para la creación aplicaciones de usuario final (Ruparelia, 2010). Uno de ellos es el ciclo de vida de desarrollo de sistemas, *Systems*

[1] Estos datos se encuentran discriminados en el número de dispositivos IoT por *Connected & smart home, Smart cities, Personal IoT, Industrial IoT, Medical IoT, Connected car.*

Development Life Cycle – SDLC, que integra seis fases: planificar, analizar, diseñar, desarrollar, implementar y mantener.

En la segunda fase del SDLC, se entiende el problema que se desea o necesita resolver a través del software, se establece metas y se identifican los requisitos del software o sistema. La ingeniería de software clasifica los requisitos de un sistema en dos: los funcionales y lo no funcionales o requisitos de calidad. Los requisitos funcionales describen lo que debe hacer el sistema, mientras que los requisitos de calidad son características que el sistema debe cumplir para considerarse un producto de calidad, por ejemplo, usabilidad, fiabilidad, rendimiento y seguridad, por nombrar algunos.

En la ingeniería de software, la seguridad es un requisito de calidad. Según la norma ISO/IEC 25010: 2011, la seguridad es el «grado en que un producto o sistema protege la información y los datos para que las personas u otros productos o sistemas tengan el grado de acceso a los datos adecuado a sus tipos y niveles de autorización» (NIST, 2011). Según esta norma, el requisito de seguridad se compone de las siguientes características: Confidencialidad, integridad, no repudio, responsabilidad y autenticidad.

La seguridad como todo requisito de calidad debe considerarse en la fase de diseño, pero debido a las condiciones actuales, donde el mercado es abierto y competitivo, haciendo que el retorno de la inversión sea crítico y facilitar la vida de los clientes tienden a tener prioridad, el desarrollo de aplicaciones IoT se centra en atender requisitos funcionales y dejar de lado el requisito de seguridad (Emm, Unuchek, & Kruglov, 2016).

Esta mala práctica de ciberseguridad, sumándose a otras malas prácticas como el uso de una configuración predeterminada y que la conectividad de estas aplicaciones IoT a menudo se añade a una red de comunicación pre-existente que no se haya creado con la seguridad desde el diseño (Kaspersky Lab, 2016), ha colocado al IoT en el foco principal de la inseguridad en la actualidad.

Según el *Kaspersky Security Bulletin* (2016), dos de las causas del gran número de problemas de seguridad que presenta IoT son las malas prácticas de seguridad que tienen usuarios y las organizaciones y, que no hay actualizaciones de firmware para muchos dispositivos convirtiéndolos en un objetivo atractivo para los ciberdelincuentes porque a menudo tienen conectividad 24/7.

Entre los problemas de seguridad y privacidad de IoT están las familias de malware como troyanos, gusanos y ransomware. Los troyanos explotan vulnerabilidades en los dispositivos IoT, cuyo objetivo más común es la creación

de botnets[2], para llevar a cabo ataques de denegación de servicio distribuidos, DDoS – *Distributed Denial-of-Service*, contra importantes servicios de Internet. Hasta la fecha, dos de las botnets más potentes en la historia fueron Mirai y Leet.

La seguridad de las aplicaciones IoT es uno de los retos que enfrenta este paradigma en la actualidad. Gestionar la seguridad, reduciendo el riesgo al mínimo aceptable, es una de las claves para para incentivar la aceptación del IoT en las ciudades, las organizaciones y los hogares. Para *Microsoft Corporation*, la seguridad es el primer obstáculo para la adopción corporativa de IoT durante 2017 (Microsoft Colombia, 2016). En este mismo sentido, para Cisco, el IoT puede convertirse en la amenaza de ciberseguridad más desafiante del mundo, porque generará nuevos objetivos de ataque como vehículos, edificios y plantas de fabricación (Cisco, 2015). Forrester Research en su informe *Predictions 2018: IoT Moves From Experimentation To Business Scale* afirma que la seguridad en IoT seguirá siendo una preocupación clave (Forrester, 2017).

1.1.2 Problema. El ecosistema IoT se compone de muchas tecnologías, incluyendo hardware, plataformas, comunicaciones, protocolos, redes y aplicaciones que interactúan entre sí en una solución que facilita la vida de los ciudadanos y las empresas (Tuck, 2016). Estas tecnologías tienen un grado diferente de madurez, algunas existen desde varias décadas, mientras que otras han surgido recientemente. Este es el caso de las redes de sensores inalámbricas, cuyo origen data de la década de 1980, y ahora hace parte del ecosistema de IoT, junto a otras tecnologías más recientes (Manrique, Rueda-Rueda, & Portocarrero, 2016; Rueda R. & TalaveraP., 2017).

Dos de las razones de la dificultad en la gestión de la seguridad de la información son: (i) la falta de un modelo que conduzca a los desarrolladores de aplicaciones IoT en el desarrollo seguro de aplicaciones, orientando sobre qué recursos garantizar y cómo debe hacerse a través de todo el ecosistema IoT[3], y (ii) la falta de talento de ciberseguridad en las empresas (Chant, 2017; Cobb, 2016a; Intel Security & CSIS, 2016; ISACA, 2016a).

La primera razón, IoT ha estado evolucionando desde las últimas décadas, pero su grado de madurez no es alto en algunos aspectos como la seguridad. Existen guías y directrices de seguridad sobre tecnologías específicas de ecosistemas de IoT, como la computación en nube (ISO/IEC, 2015).

[2] Una botnet es una red de computadoras comprometidas, que es controlada por un tercero y usada para transmitir malware o para lanzar ataques.
[3] Esta afirmación está fundamentada en las conclusiones a las que se llegó en el estado del arte descrito en este documento ver sección 2.3.

En cuanto a la segunda razón, actualmente hay una escasez de profesionales preparados para afrontar los retos en materia de ciberseguridad en las organizaciones (Deloitte, 2018; Intel Security & CSIS, 2016; Oltski, 2017). Según Cisco, el IoT también afectaran a los desafíos ya existentes en la contratación de ciberseguridad, ya que representa una oportunidad sin precedentes para conectar personas, procesos, datos y cosas (Cisco, 2015).

De acuerdo al informe de Capgemini (2018), el 55% de las empresas que participaron en el estudio, afirman que la brecha de talento digital se está ampliando y las habilidades de ciberseguridad ocupan el primer lugar tanto en la demanda como en la brecha de talento, es decir, la diferencia entre demanda y oferta. Se estima que para el año 2021 habrán 3.5 millones de puestos de trabajo de ciberseguridad (Herjavec Group, 2017).

Esto da como resultado la implementación parcial o nula de los controles necesarios para asegurar los activos de información, que estos controles no sean efectivos, o que las aplicaciones de IoT desarrolladas en una organización, bien para implementarla en sí misma o para venderla a un tercero, no consideren la seguridad desde la fase de diseño.

Patel y Cassou (2015) afirman que en el desarrollo de aplicaciones IoT intervienen varios roles como son el experto del dominio, el diseñador de software, el desarrollador de aplicaciones, el desarrollador de dispositivos y el administrador de redes. Todos los roles están enfocados a una parte en el diseño de una aplicación IoT, y no se visualiza uno que sea experto en temas de la seguridad de la información. La seguridad es un tema transversal a la arquitectura de una aplicación IoT, y de cualquier sistema, y si esta no se considera desde un punto de vista holístico puede crear brechas de seguridad en la aplicación IoT. Por esta razón, es importante que a los roles definidos por Patel y Cassou (2015) se adicione un rol que gestione la implementación de elementos de ciberseguridad durante todo el proceso de desarrollo, comenzando por los requisitos de calidad, pasando por el diseño y las demás etapas de desarrollo.

De acuerdo con las responsabilidades de los roles en el desarrollo de aplicaciones IoT descritas por Patel & Cassou (2015), el diseñador de software «define la estructura de una aplicación IoT especificando los componentes de software y sus relaciones de generación, consumo y comando». De acuerdo con lo anterior, los diseñadores de software son los responsables de especificar las arquitecturas de aplicación que describen todo el dominio de la solución IoT, con sus requisitos funcionales representados en los componentes de sistema y la relación entre ellos, describiendo así el funcionamiento de la solución IoT.

Del mismo modo, los requisitos de calidad deben ser considerados desde la fase de diseño, pero ciertos requisitos de calidad, como la seguridad, son más complejos para representar en componentes de software debido a los conocimientos específicos requeridos, y a que se debe considerar en los diferentes niveles de la arquitectura de las aplicaciones IoT.

Estos sistemas y software se desarrollan para generar, transmitir y almacenar datos. Estos sistemas deben cumplir con características de la ciberseguridad tales como confidencialidad, integridad, no repudio, responsabilidad y autenticidad. Para gestionar la ciberseguridad se requieren habilidades y conocimientos técnicos, comprender el dominio de seguridad y la gestión de los riesgos asociados con la TI.

En resumen, el problema que se aborda en este trabajo se muestra a continuación:

> Existen aplicaciones IoT inseguras por la falta de guías que orienten a los desarrolladores en la implementación del dominio de la ciberseguridad en la fase de diseño y la evaluación de estas.

Cualquier organización es un potencial blanco de un ataque cibernético, sin importar si es una pequeña empresa (York Risk Services Group, 2015). No gestionar la seguridad trae consigo varias implicaciones, tales como (i) problemas de seguridad y privacidad para las organizaciones y usuarios; (ii) facilitar que la propia infraestructura de TI forme parte de un ataque cibernético, como es el caso de las Botnets utilizadas para el ataque DDoS y (iii) aumentar el riesgo de ser blanco de un ataque cibernético que provoque fugas y robo de Información y pérdidas económicas. En consecuencia, estas implicaciones generan un retraso en la implementación de soluciones IoT en ciudades y organizaciones.

Cuando una empresa ha sido víctima de un ataque cibernético, existe una amplia gama de costos directos e intangibles que contribuyen al impacto general del incidente cibernético, algunos de ellos son: las notificaciones del ciberataque a los clientes y su protección después del incidente, el cumplimiento normativo de las investigaciones técnicas, los aumentos de las primas de seguros, la interrupción o destrucción de los procesos organizacionales, la pérdida de valor de las relaciones con los clientes, las pérdidas de ingresos por concepto de contratos, la devaluación del nombre comercial y la pérdida de propiedad intelectual (Mossburg, Gelinne, & Calzada, 2016).

Según una encuesta realizada por *Kaspersky Labs* (2015a) a más de 5500 empresas en 26 países de todo el mundo en 2015, el 90 % de las empresas admitieron un incidente de seguridad y el 46 % de las empresas perdió datos

sensibles debido a una amenaza de seguridad interna o externa. Este informe también afirma que en promedio las empresas pagan US$551.000 para recuperarse de una violación de seguridad y las pequeñas y medianas empresas (PYMES) gastan cerca de US$38.000; siendo este es el gasto directo necesario para recuperarse de un ataque. Asimismo, los costos indirectos para las empresas son de US$69.000 y US$8.000 para las PYMES. Además, afirma que las tres consecuencias principales de un incidente fueron la pérdida de acceso a la información crítica de la empresa, el daño a la reputación de la empresa y la pérdida temporal de la capacidad para comerciar. Otro informe de *Kaspersky Labs* (2015b) afirma que el malware, los ataques de *phishing* y las fugas de datos accidentales por parte del personal fueron tres amenazas de seguridad que condujeron a la pérdida de datos.

Cada año las empresas invierten cada vez más recursos en mejorar la seguridad de sus activos. Según Gartner Inc. (2016c), el gasto mundial en seguridad de IoT en 2014 fue de US $ 231.86 millones, en 2015 US$281.54 millones, en 2016 de US$348.32, y predice que en 2017 se podrá llegar a US$433.95 y US$547.20 para 2018.

De acuerdo a un reporte de Lloyd's of London (2017), un ciberataque global podría desempeñar pérdidas económicas estimadas es $53.000 millones de dólares, siendo un costo mayor que el generado por el huracán Sandy que azotó a Estados Unidos en 2012.

El *Cisco 2017 Annual Cybersecurity Report* (2017) afirma que el 50 % de las organizaciones luego de un incidente de seguridad se enfrentaron al escrutinio público y que el 22 % de las organizaciones que sufrieron un ataque informático perdieron clientes, de las cuales el 40 % perdió más del 20 por ciento de sus clientes. Asimismo, el 29% de las organizaciones atacadas perdió ingresos, y el 23 % perdió oportunidades de negocio.

La gestión de la seguridad en las aplicaciones de IoT es uno de los puntos más importantes y críticos durante el desarrollo o la implementación de una solución. Para la gestión de la seguridad, podemos considerar dos momentos: (i) del equipo de desarrollo que diseña y construye la solución IoT; y (ii) del equipo de TI de la organización que implementa una solución ya creada.

Facilitar la implementación del dominio de la ciberseguridad en la etapa de diseño de aplicaciones de IoT es relevante ya que es el primer momento para gestionar la seguridad, lo que significa que todo lo que se construye tendrá la seguridad como un requisito por cumplir y no como una característica opcional.

1.2. MOTIVACIÓN

Para la realización de este trabajo de investigación se tuvo en cuenta dos motivaciones: El primero, el modelamiento del dominio de la ciberseguridad, y la segunda, la implementación de las buenas prácticas de la ingeniería de software en proyectos telemáticos.

1.2.1 Modelamiento del dominio de la ciberseguridad. El dominio de la ciberseguridad es un campo muy amplio y especializado, para el cual hay una escasez de profesionales en el área (ISACA, 2018). Sumado a esto, el paradigma del Internet de las cosas está en su etapa de formación, todavía hace falta normalización en muchos aspectos, incluso, en la ciberseguridad; donde no hay claridad del todo sobre qué recursos proteger y cómo hacerlo.

Por esta razón, en este trabajo se pretende concebir un *framework* de ciberseguridad para aplicaciones IoT, con el objetivo de apoyar la implementación de la ciberseguridad en el diseño y construcción de aplicaciones IoT.

Un *framework* se puede entender como un diseño reutilizable, ya sean modelos y/o código, que puede ser especializado y ampliado para proporcionar una parte de la funcionalidad general de muchas aplicaciones (ISO/IEC/IEEE, 2010). Para esta investigación se define *framework* como un diseño reutilizable compuesto de modelos. Un modelo es una representación de un proceso, dispositivo o concepto del mundo real (ISO/IEC/IEEE, 2010). El *framework* propuesto está compuesto por dos modelos, que de forma diferentes pretende representar el dominio de la ciberseguridad para proteger los activos de la información en aplicaciones IoT.

El primer modelo, es un modelo para la gestión de la ciberseguridad en aplicaciones IoT. Este modelo está compuesto por una representación del dominio de la ciberseguridad en IoT, la descripción de dicha representación, guía de uso, buenas prácticas y herramientas, como, por ejemplo, instrumento para evaluar los componentes de ciberseguridad en una aplicación IoT ya construida.

El segundo modelo es de tipo conceptual, y representa el dominio de la ciberseguridad. Un modelo conceptual, es un modelo de los conceptos relevantes para algún esfuerzo (ISO/IEC/IEEE, 2010). En este modelo se tuvieron en cuenta los conceptos relevantes de la ciberseguridad para guiar la implementación de este dominio en la construcción de aplicaciones IoT. Este modelo está basado en el modelo anterior y pretende identificar los elementos que lo componen y sus relaciones, permitiendo a los diseñadores de aplicaciones

IoT hacer uso de este modelo para proveer, desde un punto de vista de diseño, todos los criterios de seguridad necesarios en la aplicación IoT a construir.

1.2.2 Buenas prácticas de la ingeniería del software en proyectos telemáticos. De acuerdo al concepto de ingeniería de software dado por la IEEE (1990), esta área del conocimiento tiene dos formas de abordase: el primero de ellos es la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software; el segundo, es el estudio de dicho enfoque.

Este proyecto de investigación pretendió abordar la ingeniería del software desde el estudio del enfoque de esta ingeniería, pero no se limita al proceso de desarrollo de software, sino que se integró con otras áreas del saber cómo la telemática, la ciberseguridad y el modelamiento para realizar un aporte al dominio del internet de las cosas, siendo más preciso, al diseño, construcción y evaluación de aplicaciones IoT.

Para el desarrollo de una aplicación IoT se puede seguir una de las metodologías de desarrollo de sistemas o software. Una de estas metodologías es el ciclo de vida de desarrollo de sistemas, *Systems Development Life Cycle* – SDLC, el cual considera seis fases: planificar, analizar, diseñar, desarrollar, implementar y mantener.

En la primera fase, la de planificar, se consideran asuntos relacionados con el desarrollo del proyecto, como es el personal, el esquema y actividades de trabajo. En la segunda fase, la de analizar, se entiende el problema, se establecen los objetivos y se establecen los requerimientos de la aplicación IoT. En el análisis de requisitos se identifican los requisitos funcionales y de calidad de la aplicación IoT. En este trabajo se abordará uno de los requisitos de calidad del software, la seguridad.

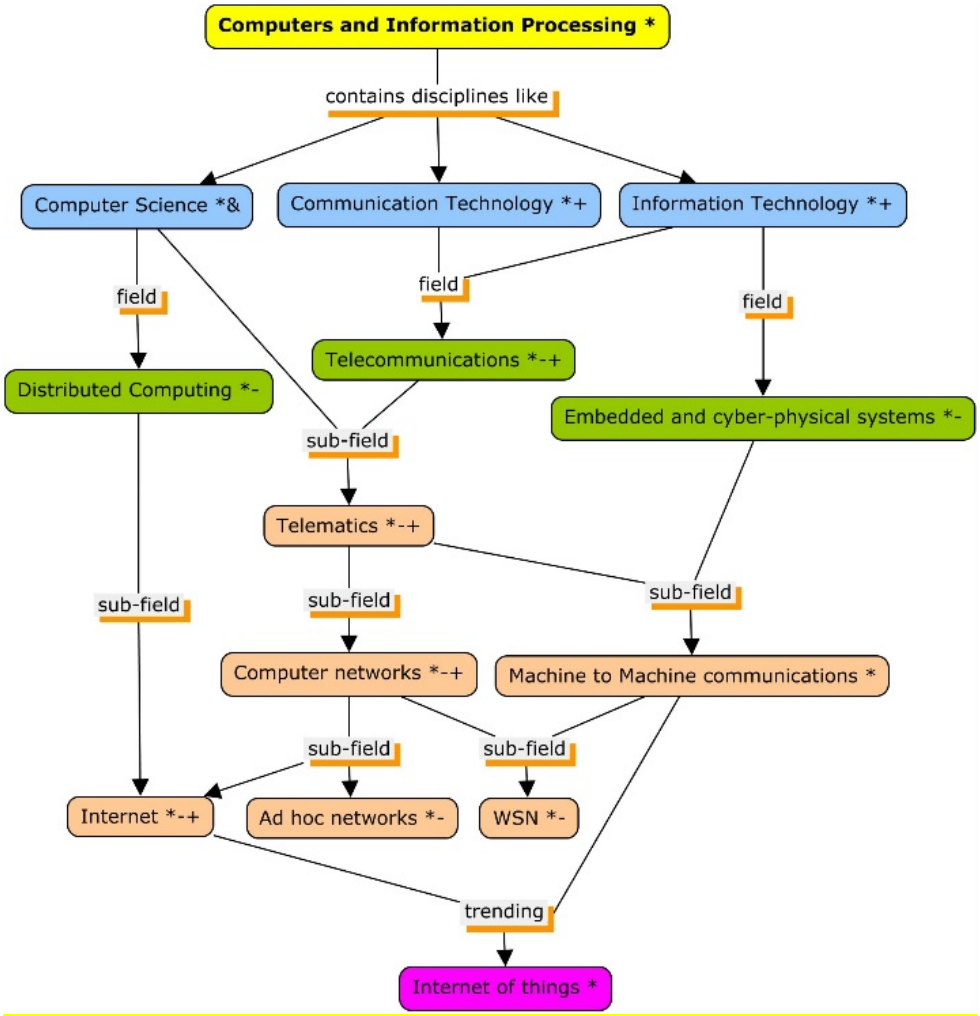
Para entender la importancia de los requisitos dentro de una aplicación IoT se debe entender cuál es el proceso de desarrollo de software. El proceso de desarrollo de software es el proceso por el cual las necesidades del usuario se traducen en un producto de software (IEEE, 1990), en este caso, en una aplicación IoT. Este proceso implica que las necesidades del cliente se traduzcan en requisitos, transformando estos requisitos en diseño, para luego implementar el diseño en componentes hardware y software integrados en una arquitectura IoT, el cual será probado y revisado hasta que se obtenga una aplicación IoT para su uso operativo.

Para Pietro (2010), en las redes de sensores inalámbricas – WSN, se deben usar las técnicas y la experiencia madurada por la ingeniería de software, tomando

las metodologías, técnicas y abstracciones que mejoren el proceso de desarrollo de aplicaciones WSN, fomentando la confianza en el diseño y que la efectividad se demuestre en el mundo real. Las WSN tiene algunas similitudes y diferencias con el IoT, pero también es necesario abordar su desarrollo usando ya la madurez desarrollada por la ingeniería del software, tras muchos años de experiencias, acumulado de pruebas y errores, llevándola al área robusta que se conoce hoy en día.

Pero la telemática y la ingeniería del software no son campos de conocimiento completamente aislados. En un trabajo previo se analizó conceptualmente el origen del Internet de las cosas y su relación con el campo de la telemática, como se muestra en la Figura 1. De acuerdo con un análisis conceptual, la telemática tiene su origen en las ciencias computacionales y las telecomunicaciones (Manrique et al., 2016); y dentro de las ciencias computacionales se encuentra la ingeniería del software. En este orden de ideas, la telemática hereda las buenas prácticas de la ingeniería del software para el desarrollo de sistemas.

Figura 1. Relación conceptual entre internet de las cosas y la telemática



Fuente: Manrique, Rueda-Rueda, & Portocarrero (2016).

1. 3 PREGUNTA DE INVESTIGACIÓN

Con el objetivo de abordar el problema suscitado en la subsección 1.1.2, se define la siguiente pregunta de investigación:

> ¿De qué manera se puede facilitar la incorporación y evaluación de la ciberseguridad en aplicaciones IoT?

1.4 HIPÓTESIS

Para responder a la pregunta de investigación identificada en la sección anterior se define la siguiente hipótesis:

> Mediante un *framework*, compuesto por diferentes tipos de modelos, se puede orientar al equipo de desarrollo sobre cómo considerar ciberseguridad en las aplicaciones IoT.

1.5 OBJETIVOS

Para verificar la hipótesis, se formuló el siguiente objetivo general de esta investigación:

> Concebir un *framework* de ciberseguridad para aplicaciones IoT, partiendo de un modelo conceptual, representado mediante un lenguaje de modelamiento y finalizando con un proceso de validación.

Este objetivo general se alcanzará a través de la satisfacción de los siguientes objetivos específicos – OE:

- OE1: Formular un modelo conceptual de la gestión de la seguridad de la información para el diseño y evaluación de aplicaciones IoT.
- OE2: Representar, mediante un lenguaje de modelamiento, el dominio de la seguridad de la información para aplicaciones IoT.
- OE3: Validar, mediante una técnica diseñada para tal fin, el *framework* formulado.

1.6 CONTRIBUCIONES.

Las principales contribuciones de este trabajo de investigación son:

- Un *framework* conceptual para la gestión de la seguridad para aplicaciones IoT. A su vez, este *framework* está compuesto por dos modelos: *SMITH Model* y IoT-CyDM.
- Un modelo para la gestión de la ciberseguridad en aplicaciones IoT, llamado *SMITH Model* que contiene una guía de buenas prácticas para el aseguramiento de aplicaciones IoT e instrumentos de evaluación de elementos de ciberseguridad en aplicaciones IoT ya construidas, ver capítulo 4.
- Un modelo del dominio de ciberseguridad para aplicaciones IoT, llamado IoT-CyDM, ver capítulo 5.
- Un caso de estudio que describe cómo usar los elementos del *SMITH Framework* para el diseño de una aplicación IoT que considere elementos de ciberseguridad, ver capítulo 6.

Además de las contribuciones principales realizadas, en entre trabajo de investigación surgieron otras contribuciones, las cuales se mencionan a continuación:

- Una contribución a la literatura en el tema de *frameworks* de seguridad para aplicaciones IoT, en el cual se listan, clasifican y analizan dichas propuestas de seguridad y se identifica una brecha de investigación en esta área, ver sección 2.3.1.
- Una metodología que puede ser usada para la creación de modelos basado en requisitos de calidad. En este trabajo se modelo el requisito de seguridad, ver sección 4.1.
- Una revisión sistemática de la literatura sobre arquitecturas de referencia para aplicaciones IoT, y una evaluación para determinar el nivel de completitud de estas, ver subsección 4.2.1.
- Una arquitectura IoT genérica que considera las funcionalidades comunes que debe tener una aplicación IoT, ver subsección 4.3.4.
- Una representación del dominio IoT, que puede ser usado en otros contextos investigativos, ver subsección 5.1.2.
- Un artículo de conferencia titulado "*Contrasting Internet of Things and Wireless Sensor Network from a Conceptual Overview*" que puede encontrarse en la IEEE Xplore.
- Un artículo titulado "Similitudes y diferencias entre Redes de Sensores Inalámbricas e Internet de las Cosas: Hacia una postura clarificadora" próximo a publicar en la Revista Colombiana de Computación, volumen 18, número 2 (en proceso de publicación).

2. MARCO REFERENCIAL

En este capítulo, se presenta la fundamentación teórica de esta investigación, en el que se presentan sus bases teóricas y conceptuales, el estado actual de la investigación, el contexto en el que se desarrolla y los estándares que se tuvieron en cuenta para su desarrollo. Este capítulo se organiza en seis secciones:

En la sección 2.1, se presenta el marco referencial. En este marco se presentan los conceptos relevantes para esta investigación, los cuales se organizaron en cuatro áreas: ingeniería del software, ciberseguridad, telemática y modelamiento.

En la sección 2.2, se muestra el marco teórico en el que se describen las teorías que fundamentan esta investigación. Estas teorías se organizaron en las tres áreas del conocimiento, ingeniería del software, ciberseguridad y el Internet de las Cosas, que al interactuar dan base a los resultados presentados en esta investigación.

En la sección 2.3, se presenta el estado del arte orientado a dos temáticas: la primera de ellas, sobre *frameworks* y modelos de seguridad para aplicaciones IoT. En esta sección está organizada así: el protocolo del estado del arte; el estado del arte con los trabajos agrupados por modelos y *frameworks*; un análisis realizado a dichos trabajos, donde se identifica las tendencias en la construcción de dichas propuestas, los recursos IoT y las propiedades de la seguridad de la información que protegen; por último, las conclusiones a las que se llegaron y la brecha de investigación identificada. La segunda temática sobre el estado actual de la ciberseguridad en IoT en el cual, no solo se consultaron artículos científicos, sino también informes y estudios realizados por organizaciones reconocidas en el área de la ciberseguridad. Este estado actual se organizó en dos subsecciones: lo relacionado con el *malware* IoT y lo relacionado con los dispositivos IoT. Finalmente se presentan las conclusiones.

En la sección 2.4, se presenta el marco normativo y estándares, que indica las normas, políticas o estándares aceptadas internacionalmente que influenciaron este trabajo de investigación. En la sección 2.5 se presenta el marco contextual y los antecedentes de esta investigación. Finalmente, en la sección 2.6 se presenta las consideraciones finales de este capítulo.

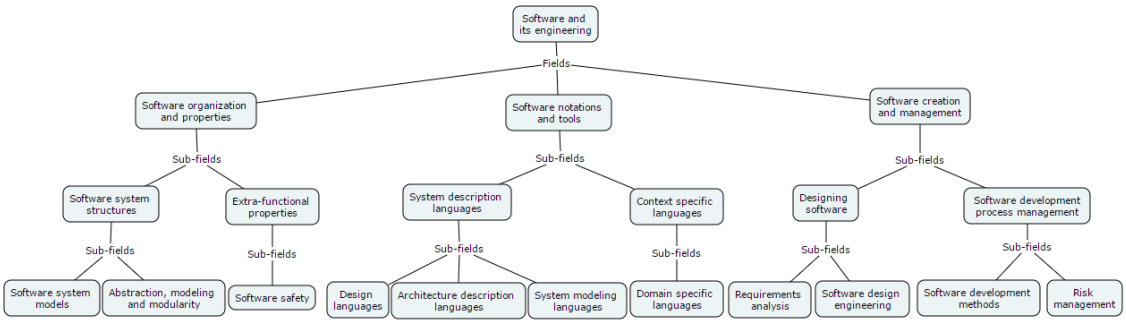
2.1 MARCO CONCEPTUAL

Es esta sección se presentan los conceptos relevantes para esta investigación, relacionados con la ingeniería del software, ciberseguridad, internet de las cosas

y modelamiento, que por su significado particular necesitan precisarse en su definición (Morán Delgado & Alvarado Cervantes, 2010). Los conceptos descritos se organizaron en cuatro áreas: ingeniería del software, ciberseguridad, telemática y modelamiento.

2.1.1 Ingeniería del software. Como punto de partida, se hizo una revisión conceptual en el tesauro de ACM[4] para conocer los campos y sub-campos de la ingeniería del software, como se muestra en la Figura 2.

Figura 2. Relación conceptual de la ingeniería del software.



Fuente: Autor.

Y los términos relacionados son:

2.1.1.1 Arquitectura de referencia. Una arquitectura de referencia es un modelo mapeado en elementos de software que implementan la funcionalidad definida en el modelo de referencia (Software Engineering Institute, 2016).

2.1.1.2 Arquitectura de software. De acuerdo a la ISO/IEC/IEEE 4201:2011 (2011), una arquitectura de software son «conceptos fundamentales o propiedades de un sistema en su entorno incorporado en sus elementos, relaciones y en los principios de su diseño y evolución».

2.1.1.3 *Framework*. Un *framework* es un diseño reutilizable, ya sean modelos y/o código, que puede ser especializado y ampliado para proporcionar una parte de la funcionalidad general de muchas aplicaciones (ISO/IEC/IEEE, 2010).

2.1.1.4 Framework conceptual. En un trabajo de Maxwell (2005), mencionan que «Miles y Huberman (1994) definieron un marco conceptual como un

[4] Se puede acceder al Thesarus de ACM ingresando al siguiente enlace http://taxonomies.labs.crossref.org/?p=109

producto visual o escrito, que "explica, ya sea gráficamente o en forma narrativa, las principales cosas a estudiar –los factores clave, conceptos o variables– y las supuestas relaciones entre ellos" (p. 18)».

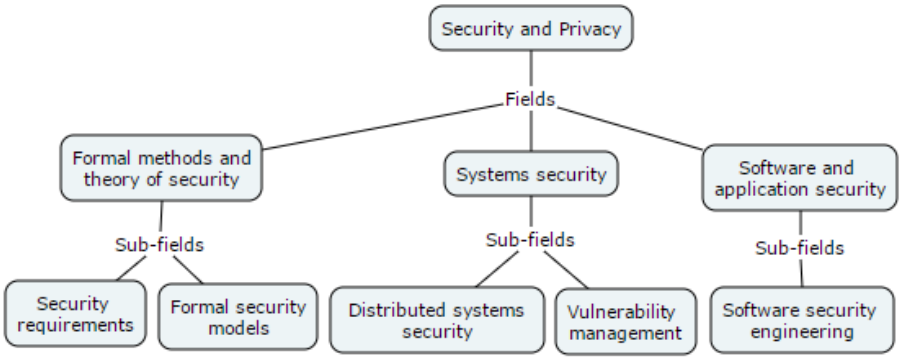
2.1.1.4 Modelo de referencia. Un modelo de referencia es una división de funcionalidad en elementos junto con el flujo de datos entre esos elementos (Software Engineering Institute, 2016).

2.1.1.5 Requisito de calidad. Para los autores de este proyecto, se usa este término para referirse a aquellos requisitos que describen cómo el software hará sus funcionalidades, basado en atributos de calidad del software (Adams, 2015). En la literatura, estos requisitos también se conocen como requisito no funcional (Roman, 1985), atributo de calidad (B. W. Boehm, Brown, & Kaspar, 1978), restricciones (Roman, 1985), metas o *goals* (Mostow, 1985), requisitos extra funcionales (Shaw, 1989) o requisitos no conductuales (Davis, 1993).

2.1.2 Ciberseguridad. La ciberseguridad es «la protección de los activos de información mediante el tratamiento de las amenazas a la información procesada, almacenada y transportada por sistemas de información interconectados» (ISACA, 2016b, p. 9, mi traducción).

Para conocer los campos y sub-campos de la seguridad, como dominio general, se hizo una revisión conceptual en el tesauro de ACM, como se muestra en la Figura 3.

Figura 3. Relación conceptual de la seguridad.



Fuente: Autor.

Este término puede ser confundido con otros términos como es la seguridad informática y la seguridad de la información:

- ***Seguridad informática***. La seguridad informática - *infosec*) son medidas y controles que garantizan la confidencialidad, integridad y disponibilidad de los activos del sistema de información, incluyendo hardware, software, firmware e información procesada, almacenada y comunicada (Committee on National Security Systems, 2010, mi traducción).

- ***Seguridad de información***. La seguridad de la información es la protección de la información y los sistemas de información contra el acceso, uso, divulgación, interrupción, modificación o destrucción no autorizados a fin de proporcionar confidencialidad, integridad y disponibilidad (NIST, 2013, mi traducción).

La seguridad de la información es el término amplio que abarcar a la ciberseguridad y a la seguridad informática. Otros términos relacionados con la ciberseguridad son:

2.1.2.1 Ciberespacio. El ciberespacio se defino como «un dominio global dentro del entorno de información que consiste en la red interdependiente de infraestructuras de sistemas de información incluyendo Internet, redes de telecomunicaciones, sistemas informáticos y procesadores y controladores integrados» (Committee on National Security Systems, 2010, p. 22, mi traducción).

2.1.2.2 Ciberincidente. Un ciberincidente se define como «las medidas adoptadas mediante el uso de redes informáticas que tengan como consecuencia un efecto real o potencialmente adverso sobre un sistema de información y/o la información que en él se encuentre» (Committee on National Security Systems, 2010, p. 22, mi traducción).

2.1.2.3 Incidente de seguridad. Un incidente de seguridad es «una situación evaluada que pone en peligro la confidencialidad, integridad o disponibilidad de un sistema de información o la información que este sistema procesa, almacena o transmite; o que constituya una violación o amenaza inminente de violación de políticas de seguridad, procedimientos de seguridad o políticas de uso aceptable» (Committee on National Security Systems, 2010, p. 35, mi traducción).

2.1.2.4 Ingeniería de seguridad. La ingeniería de seguridad es un enfoque interdisciplinario y medios para posibilitar la realización de sistemas seguros. Se centra en definir las necesidades de los clientes, los requisitos de protección de seguridad y la funcionalidad requerida en el inicio del ciclo de vida del desarrollo de sistemas, documentar los requisitos y continuar con el diseño, la síntesis y la

validación del sistema mientras se considera el problema completo (CNSS, 2010).

2.1.3 Telemática. La telemática es un concepto de tecnología de la información sobre la transmisión de datos a larga distancia. En los vehículos en movimiento, la telemática se refiere al uso integrado de telecomunicaciones e informática, como las pantallas del salpicadero que muestran la posición actual del vehículo en un mapa o en aplicaciones de seguimiento centralizado (Internet of Things Guide, 2016).

2.1.4 Internet de las cosas. El Internet de las cosas – IoT es un paradigma que se basa en la interacción de objetos inteligentes, las cosas, entre sí y con recursos físicos y/o virtuales a través de Internet (Cavalcante et al., 2016).

2.1.5 Modelamiento. Los términos relacionados con el modelamiento son:

2.1.5.1 Dominio. Según el estándar ISO/IEC/IEEE 24765 (2010), un dominio es «un ámbito distinto, dentro del cual se exhiben características comunes, se observan reglas comunes y sobre las cuales se mantiene una transparencia de distribución».

2.1.5.2 Lenguajes de modelamiento. Un lenguaje de modelamiento es «cualquier lenguaje de computadora gráfico o textual que aprovisione el diseño y la construcción de estructuras y modelos siguiendo un conjunto sistemático de reglas y *frameworks*» (Techopedia, 2017).

De acuerdo a Talavera Portocarrero (2016), existen dos enfoques para el modelado de software: (i) los lenguajes de descripción de arquitecturas – ADL, por sus siglas en inglés, *Architecture Description Languages*; y (ii) y los lenguajes de modelado estándar – OMG, por sus siglas en inglés, *Object Management Group*.

El lenguaje de modelamiento unificado – UML y el lenguaje de modelado de sistemas – SysML son ejemplos de OML; en cuanto a las ADL's, existen ADL formales y ADL semiformales. Talavera Portocarrero (Talavera Portocarrero, 2016) afirma que en la práctica, UML 2 es una ADL semiformal.

2.1.5.3 Modelo. Un modelo es una representación de un proceso, dispositivo o concepto del mundo real; también puede ser definido como una abstracción semánticamente cerrada de un sistema o una descripción completa de un sistema desde una perspectiva particular (ISO/IEC/IEEE, 2010).

2.2 MARCO TEÓRICO

En esta sección se presenta la fundamentación teórica de esta investigación, la cual está basada en tres áreas: la ingeniería del software, la ciberseguridad y el internet de las cosas. En cuanto a la ingeniería del software, se hace énfasis en la importancia de los requisitos en el desarrollo de un software o sistema de calidad; en lo relacionado con la ciberseguridad, se presenta lo que abarca esta disciplina y la gestión del riesgo para proteger los activos de información. Finalmente se habla del Internet de las cosas, su origen, tecnologías fundamentales, los dominios de aplicación, el ecosistema IoT y la construcción de aplicaciones IoT: los enfoques existentes y los roles que intervienen.

2.2.1 Ingeniería del software. La ingeniería del software, como término y como disciplina, se considera que nace en la *NATO Software Engineering Conference*, un evento organizado por la OTAN en 1968 (Naur & Randell, 1969). Un tema tratado en esta conferencia fue lo que los asistentes llamaron 'La crisis del software' o 'la brecha del software', para referirse a las dificultades para construir software de calidad.

Según la definición de la IEEE (1990), la ingeniería del software es un término muy amplio que abarca el estudio y la aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, operación y mantenimiento de software. De esta forma, la ingeniería del software no solo incluye la producción de software sino también la investigación de los enfoques descritos en la definición anterior.

2.2.1.1 Proceso de desarrollo de software. El proceso de desarrollo de software es el proceso por el cual las necesidades del usuario se traducen en un producto de software. Este proceso implica que estas necesidades se traduzcan en requisitos de software, transformando estos requisitos en diseño, para luego implementar el diseño en código fuente, el cual será probado y revisado hasta que se obtenga un software listo para su uso operativo (IEEE, 1990).

Para Sánchez *et al*. (2012, p. 34), el proceso de desarrollo de software es mucho más amplio, y lo define como un «conjunto coherente de políticas, estructuras organizativas, tecnologías, procedimientos y artefactos que se necesitan para concebir, desarrollar, implantar y mantener un producto software».

El proceso de producción de software se considera una ciencia disciplinada, una actividad profesional sometida al estudio científico y objetivada en técnicas y métodos mayoritariamente aceptados por la comunidad profesional en virtud de la experiencia acumulada (S. Sanchez et al., 2012).

El ciclo de vida de desarrollo de software es un conjunto coherente de actividades para la producción de software (Pressman, 2010; Sommerville, 2011). Existen varios modelos que describen el proceso de desarrollo de software, como son el modelo en cascada, el modelo b, el modelo incremental, el modelo V, el modelo espiral, el modelo de rueda y radio, el modelo de proceso unificado (Ruparelia, 2010). Eso por nombrar los modelos de desarrollo tradicionales. También se han propuesto otra clase de modelos o métodos llamados ágiles, entre los que se encuentran el *Extreme Programming – XP*, Scrum, la familia de metodologías Crystal entre otros (Abrahamsson, Salo, Ronkainen, & Warsta, 2017).

2.2.1.2 Ingeniería de requisitos. Según el *Software Engineering Body of Knowledge* – SWEBOK (2014), el área de conocimiento de requisitos de software, hace referencia al levantamiento, análisis, especificación y validación de los requerimientos del software; así como la gestión de los requisitos durante todo el ciclo de vida del producto de software. Dentro de la ingeniería del software existe un término ampliamente usado para denotar el manejo sistemático de los requisitos, denominado como ingeniería de requerimientos (Kotonya & Sommerville, 1998; Pohl, 2010).

La ingeniería de requisitos es el proceso de descubrir, documentar y administrador los requisitos para un sistema o software y cuyo objetivo es producir un conjunto de requisitos del sistema que, en la medida de lo posible, sea completo, coherente, relevante y refleje lo que el cliente realmente desea (Sommerville & Sawyer, 1997). Loucopoulus y Karakostas (1995) consideran la ingeniería de requisitos como una combinación de tres procesos concurrentes e interactuantes: (i) obtener conocimiento relacionado con un dominio de problema, (iii) garantizar la validez de dicho conocimiento y (iii) especificar el problema de una manera formal. Esto no resulta del todo sencillo.
Un requisito es una propiedad que debe ser exhibida por algo, ya sea un software o sistema, con el fin de resolver un problema en el mundo real; y es una combinación compleja de varias personas que de una manera u otra están implicadas o conectadas con esta característica en el entorno en el que el sistema o software operará (IEEE Computer Society, 2014).

Existen dos tipos de requisitos del software: los requisitos funcionales y los requisitos de calidad o no funcionales. Los requisitos funcionales son aquellos servicios o funcionalidades que el sistema debe proveer, cómo debería reaccionar a entradas particulares y cómo debería comportarse en situaciones específicas (Sommerville, 2011). Los requisitos de calidad describen cómo el software hará sus funcionalidades, basado en atributos de calidad del software (Adams, 2015).

Una de las principales fuentes de problemas de la ingeniería de requisitos es el conflicto entre la complejidad de los sistemas y las limitaciones humanas al manejar simultáneamente grandes cantidades de información compleja (Yamamoto, Morris, Hartsough, & Callender, 1982).

Brooks (1987, p. 18) hablando sobre los requisitos afirma que «la parte más difícil de construir un sistema software es decidir lo que se va a construir. […] Ninguna otra parte del trabajo perjudica el sistema resultante si se hace mal. Ninguna otra parte es más difícil de rectificar más tarde».

2.2.1.3 Importancia de los requisitos en el desarrollo de software. Un buen levantamiento, análisis, especificación y validación de los requerimientos del software o sistema puede mejorar significativamente la calidad del producto final. Para Moore (1994), el principal beneficio de una buena gestión de los requisitos es que se hace una mejor comprensión de las necesidades reales y subyacentes de los usuarios finales; lo que se traduce en que el producto cumplirá con criterios más cualitativos y subjetivos de los usuarios, completándose de una manera más eficiente y oportuna.

La utilidad de un sistema de software está determinada tanto por su funcionalidad como por sus características de calidad (Chung & do Prado Leite, 2009). Los requisitos de calidad juegan un rol critico durante el desarrollo del sistema (Chung, Nixon, Yu, & Mylopoulos, 2012), ya que expresan las necesidades y limitaciones de un producto de software que contribuyen a la solución de algún problema del mundo real (IEEE Computer Society, 2014).

Los requisitos de calidad se aplican al sistema, más que a características o servicios individuales del mismo (Sommerville, 2011). Por esta razón es importante gestionar adecuadamente los requisitos de calidad, ya que son los más costosos y difíciles de corregir una vez que se ha implementado un sistema de software (Brooks, 1987; Davis, 1993).

Es necesario un análisis a fondo de este tipo de requisitos, para proporcionar a los usuarios un software de alta calidad, con atributos de fiabilidad y seguridad como resultado de un claro análisis de requisitos (Hwang & Park, 2017; Mead & Stehney, 2005). Los requisitos de calidad son difíciles de probar, por lo que generalmente se evalúan subjetivamente (Adams, 2015).

La gestión de los requisitos de calidad también es relevante en el desarrollo de sistemas IoT, ya que, cualquier cambio en las etapas posteriores de desarrollo del sistema supone un costo enorme, aumentando el tiempo de diseño y de

implementación (Mahalank, Malagund, & Banakar, 2016). Una buena gestión de los requisitos se traduce en un sistema IoT de calidad, y por lo tanto seguro.

2.2.1.4 Evaluación de arquitecturas. Una arquitectura de software «es la estructura o estructuras del sistema, que comprenden elementos de software, las propiedades visibles externamente de esos elementos y las relaciones entre ellos» (Rozanski & Woods, 2005), sirviendo como modelo para el sistema y el equipo de proyecto que lo desarrolla, definiendo las asignaciones de trabajo que deben llevar a cabo los equipos de diseño e implementación (Owens, 2005). La documentación de una arquitectura de manera efectiva es tan importante como elaborarla, ya que si esta no se entiende o es malinterpretada no pude cumplir sus objetivos como la visión unificadora para el desarrollo del sistema y el software (Clements, Garlan, Little, Nord, & Stafford, 2003). La arquitectura de software es un diseño destinado a garantizar un determinado conjunto de atributos de calidad. De acuerdo a Caracciolo, Lungu y Nierstrasz (2014) los requisitos de calidad se suelen especificar en la práctica, pero raramente se validan con técnicas automatizadas.

La importancia de evaluar una arquitectura de software es que permite encontrar los problemas y corregirlos a tiempo. El costo de corregir un error durante las fases de requisitos o diseño tienen una menor magnitud que corregir un error durante la fase de pruebas. La arquitectura determina la estructura de un proyecto de software, las bibliotecas de control de configuración, los programas y presupuestos, los objetivos de rendimiento, la estructura del equipo, la organización de la documentación y las actividades de prueba y mantenimiento están todos organizados en torno a la arquitectura (B. Boehm, n.d.).

Los métodos para la evaluación de arquitecturas de software se pueden clasificar en dos grupos: (i) los métodos de evaluación temprana y (ii) los métodos de evaluación tardía (Roy & Graham, 2008). En la Tabla 1 se muestra la taxonomía de evaluación arquitectónica de software.

El primer grupo de métodos se aplican en una arquitectura de software antes de su implementación. Estos métodos evalúan que la arquitectura en la fase de diseño, y validan que cumplan con los requisitos de calidad. La ventaja de evaluar una arquitectura en la fase de diseño es que resulta fácil cambiar una decisión arquitectónica inapropiada, ya que, si estos cambios se aplican en fases posteriores del desarrollo pueden resultar costosos.

En cuanto al segundo grupo de métodos, Roy & Graham (2008) afirman que hay poca evidencia sobre la fortalezas y debilidades en la aplicación de estilos arquitectónicos y patrones de diseño en las aplicaciones de software; también afirman que libros y artículos de investigación abordan la utilidad su utilidad, pero

no proporcionaron ningún dato empíricamente validado para mostrar su efectividad.

Tabla 1. Taxonomía de la evaluación arquitectónica de software

Evaluación	Aplicado a la arquitectura de software			Fuente
	Categoría	Nombre		
Evaluación temprana	Basado en escenarios	SAAM	Scenario-based Software Architecture Analysis Method	Kazman, Asundi y Klein (1994)
		ATAM	Architecture-based Tradeoff Analysis Method	Kazman et al. (1998)
		ALPSM	Architecture Level Prediction of Software Maintenance	Bengtsson y Bosch (1999)
		ALMA	Architecture-Level Modifiability Analysis	Bengtsson et al (2004)
		SBAR	Scenario Based Architecture Reengineering	Bengtsson y Bosch (1998)
		SALUTA	Scenario-based Architecture Level UsabiliTy Analysis	Folmer, Gurp y Bosch (2004)
		SAAMCS	SAAM for Complex Scenarios	Lassing, Rijsenbrij y van Vliet (1999)
		ESAAMI	Extending SAAM by Integration in the Domain	Molter (1999)
		ASAAM	Aspectual Software Architecture Analysis Method	Tekinerdogan (2004)
		SACAM	Software Architecture Comparison Analysis Method	Bergner, Rausch, Sihling y Ternité (2005)
		DoSAM	Domain-Specific Software Archi- tecture Comparison Model	Stoermer, Bachmann y Verhoef (2003)
	Matemático basado en modelos: fiabilidad	Basado en ruta	Shooman	Shooman (1976)
			Krishnamurty and Mathur	Krishnamurty y Mathur (1997)
			Yacoub et al.	Yacoub et al. (1999)
		Basado en estado	Cheung	Cheung (1980)
			Kubat	Kubat (1989)
			Laprie	Laprie (1984)
			Gokhale y Trivedi	Gokhale y Trivedi (2002)
	Matemático basado en modelos: rendimiento	SPE	Software Performance Analysis	Smith (1990)
		WS	Williams and Smith	Williams y Smith (1998)
		PASA	Performance Assessment of Software Architecture	Williams y Smith (2002)
		CM	Cortellessa and Mirandola	Cortellessa y Mirandola (2000)
		BIM	Balsamo et al.	Balsamo et al. (1998)
		ABI	Aquilani et al.	Aquilani et al. (2001)
		AABI	Andolfi et al.	Andolfi et al. (2000)
Evaluación tardía	Basado en métricas	TLC	Tvedt et al.	Tvedt, Lindvall, y Costa (2002)
		LTC	Lindvall et al.	Lindvall, Tvedt y Costa (2003)
	Basado en herramientas	FA	Fiutem and Antoniol	Fiutem y Antoniol (1998)
		MNS	Murphy et al.	Murphy, Notkin y Sullivan (1995)
		SSC	Sefika et al.	Sefika, Sane y Campbell (1996)

Fuente: Roy y Graham (2008)

Babar y Gorton (2004) compararon cuatro métodos de evaluación de arquitecturas de software, tres de los cuales son métodos de evaluación

temprana: SAAM, ALMA y ATAM. Los autores resaltan que solo el método ATAM brinda soporte integral de procesos.

Según Ionita, Hammer y Obbink (2002) los métodos basados en escenarios son fáciles de comprender y aplicar, y el esfuerzo de aplicación es relativamente bajo. Los autores también mencionan que otros resultados de estos métodos son que hay mejor comunicación entre los *stakeholders* y descubrimientos arquitectónicos significativos y mejoras si la evaluación se realiza al principio de la fase de desarrollo.

2.2.2 Ciberseguridad. La ciberseguridad es una sub-área de la seguridad de la información (ISACA, 2013). La ciberseguridad es la protección, usando tecnologías, procesos y prácticas, de los activos de información, a través del tratamiento de amenazas que ponen en riesgo la información que es recolectada, procesada, almacenada y transportada por los sistemas de información que se encuentras interconectados (Bayuk et al., 2012; ISACA, 2016b; Touhill & Touhill, 2014).

Josiah Dykstra (2015) define la ciberseguridad como una ciencia, la cual es amplia, y abarca tecnologías y las prácticas utilizadas para proteger las redes informáticas, las computadoras y los datos contra los daños. También afirma que, como disciplina, la ciberseguridad requiere conocimiento auténtico para explorar y razonar sobre el "cómo y por qué" se construyen o implementan controles de seguridad. La ciberseguridad protege la disponibilidad, confidencialidad e integridad de los activos digitales (Fowler, 2016).

La gestión del riesgo es el proceso de identificar y evaluar los riesgos, tomando medidas para reducir el riesgo a un nivel aceptable (Hopkin, 2017). Los principales elementos de este proceso son: el establecimiento del contexto para la gestión de riesgos, la evaluación de riesgos, el tratamiento de riesgos, el monitoreo de riesgos y la revisión de riesgos (Green, 2016). El primer paso en la gestión del riesgo es la evaluación del riesgo, donde se determina el alcance de la amenaza potencial, las vulnerabilidades y el riesgo asociado con un sistema de tecnología de la información; el segundo paso es la mitigación del riesgo, que consiste en priorizar, evaluar e implementar los controles de reducción del riesgo recomendados del proceso de evaluación del riesgo (Stoneburner, Goguen, & Feringa, 2002). Dentro de la gestión del riesgo está la gestión de los riesgos cibernéticos, donde se tratan los riesgos asociados a los sistemas cibernético y sus amenazas (Refsdal, Solhaug, & Stølen, 2015).

Entre los nuevos escenarios y desafíos que enfrenta en la actualidad la ciberseguridad se encuentra el *Bring Your Own* Device – BYOD, el *cloud computing* y Big Data, las aplicaciones móviles y el Internet de las Cosas

(Fundación Telefónica, 2016). Para la ciberseguridad, el internet de las cosas supone un gran desafío, porque la información generada, procesada, transportada y almacenada es mayor, y seguirá creciendo exponencialmente a medida que IoT se va desplegando en las ciudades y organizaciones. A parte del volumen de información, la seguridad y privacidad de los datos generados es otro desafío, ya que, con el IoT, en internet hay información que antes no solía compartirse.

La ciberseguridad protege la información y sus propiedades. En seguridad de la información, como área más amplia, y dentro de la ciberseguridad, la información tiene unas propiedades como la confidencialidad, la integridad, la disponibilidad y el no repudio.

2.2.3 Internet de las cosas. Internet de las cosas o el término de origen *Internet of Things* – IoT es considerado como un paradigma (Atzori, Iera, & Morabito, 2010; Borgia, 2014; I. Lee & Lee, 2015), y en la actualidad no hay un término estandarizado que lo defina. Para Atzori, Iera y Morabino (2017) lo que IoT realmente representa no está completamente claro, generando duda de si es más un nombre de moda especulativamente montado, para aumentar la atención en estudios y productos en torno a tecnologías maduras, más que un elemento real de discontinuidad tecnológica.

Internet de las cosas es el término más conocido para referirse al paradigma en el que objetos cotidianos pueden equiparse con capacidades de identificación, detección, interconexión y procesamiento que les permitan comunicarse entre sí y con otros dispositivos y servicios a través de Internet para lograr algún objetivo. Existen otros términos, menos populares, para referirse a este paradigma, algunos de ellos son el Internet del todo (*Internet of Everything* – IoE) o Internet industrial (I. Lee & Lee, 2015).

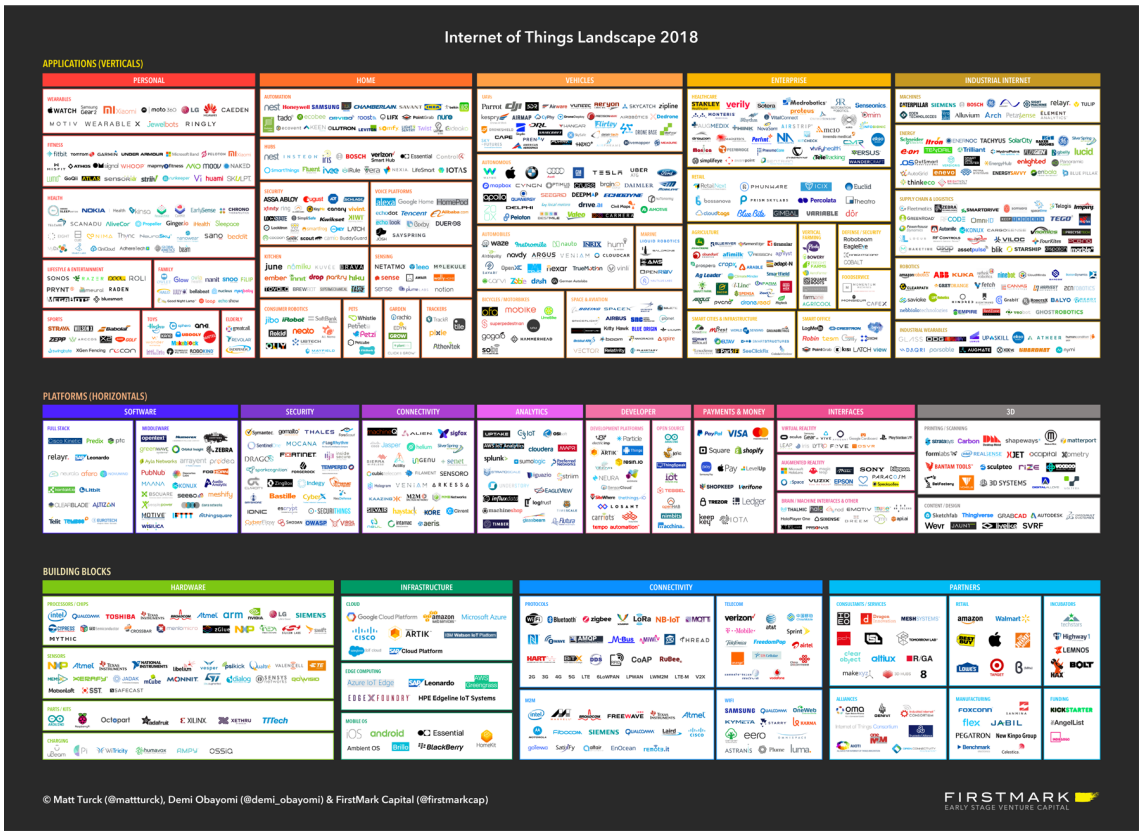
El termino *Internet of Things* fue acuñado por Kevin Ashton, uno de los fundadores del *Auto-ID Center* del *Massachusetts Institute of Technology* – MIT, en el año 1999.

En 2009, Kevin Ashton afirmaba que los computadores de esa época y por lo tanto internet, eran prácticamente dependientes de los seres humanos para recabar información y que los datos que se disponían en internet fueron inicialmente creados por humanos, a base de teclear, presionar un botón, tomar una imagen digital o escanear un código de barras (Ashton, 2009). Además, Ashton afirmó que las cosas son mucho más importantes que la información y las ideas; que la economía, la sociedad y la supervivencia se basan en las cosas y no en las ideas o información. Asimismo, plasmó la visión de lo que hoy se conoce como Internet de las cosas:

> Si tuviéramos computadoras que supieran todo lo que había que saber sobre las cosas –utilizando datos que recopilaron sin ayuda de nosotros– podríamos rastrear y contar todo, y reducir en gran medida la pérdida y el costo. Sabríamos cuándo necesitábamos reemplazar, reparar o recordar, y si estaban frescos o pasados lo mejor posible (2009, p. 1, mi traducción).

El ecosistema IoT está integrado por muchas tecnologías, hardware y software, desarrolladas por diferentes fabricantes. Matt Turck (2018) consolido gran parte de las tecnologías que componen este ecosistema en una gráfica, como se muestra en la Figura 4.

Figura 4. Ecosistema IoT en 2018.



Fuente: Tuck (2018)[5].

2.2.3.1 Dominios de aplicación. No existe una categorización estandarizada para clasificar los dominios de aplicación del IoT, dependiendo así de la forma en que los investigadores o instituciones propongan. Borgia (2014) clasifica los dominios de aplicación del IoT en tres grandes grupos: (i) dominio industrial, (ii) dominio de ciudades inteligentes y (iii) dominio de la salud y bienestar. En cada uno de estos dominios Borgia establece unos subdominios y dentro de estos, ejemplos de aplicaciones del IoT.

[5] Para ver esta imagen en tamaño completo ingrese al siguiente link: http://mattturck.com/wp-content/uploads/2018/01/Internet-of-Things-2018-1.png

Atzori *et al.* (2010) clasifica los dominios de aplicación de IoT en: Transporte y logística, saludo, ambientes inteligentes, personal y social y el futurista.

2.2.3.2 Construcción de aplicaciones IoT. Borgia (2014) describe dos enfoques para la construcción de aplicaciones IoT: el enfoque vertical y el enfoque horizontal. En el enfoque vertical, cada aplicación tiene su propia infraestructura TIC y dispositivos dedicados y las aplicaciones similares no comparten ninguna característica para la gestión de servicios y la red, resultando una redundancia innecesaria y el aumento de los costos. Por otro lado, el enfoque horizontal es más flexible que la vertical, en el cual, las aplicaciones comparten infraestructura, el ambiente y los elementos de la red y una plataforma operativa común gestiona la red y los servicios, abstrayendo una amplia gama de fuentes de datos para permitir que las aplicaciones funcionar correctamente.

2.2.3.3 Roles en el desarrollo de aplicaciones IoT. En la construcción de una aplicación IoT intervienen una serie de roles o *stakeholders*. Patel y Cassou (2015) definen cinco roles que intervienen en el desarrollo de aplicaciones IoT, como se muestra en la Tabla 2.

Tabla 2. Roles en el desarrollo de aplicaciones IoT.

Rol	Habilidades	Responsabilidades
Experto del dominio	Comprende los conceptos de dominio, incluyendo los tipos de datos producidos por los sensores y que serán consumidos por los actuadores, las interacciones del usuario y cómo el sistema se divide en regiones.	Especificar el vocabulario de un dominio de aplicación que deben utilizar las aplicaciones en el dominio.
Diseñador de software	Conceptos de arquitectura de software, incluyendo el uso adecuado de modos de interacción como publicar-suscribirse, comando y solicitud-respuesta para su uso en la aplicación.	Definir la estructura de una aplicación IoT especificando los componentes de software y sus relaciones de generación, consumo y comando.
Desarrollador de aplicaciones	Experiencia en diseño de algoritmos y uso de lenguajes de programación.	Desarrollar la lógica de aplicación de los servicios computacionales en la aplicación.
Desarrollador de dispositivos	Comprensión profunda de las entradas/salidas, y protocolos de los dispositivos individuales.	Escribir controladores para los sensores, actuadores, almacenes de datos y aplicaciones de usuario final utilizados en el dominio.
Gerente de redes	Comprensión profunda del área específica donde se va a desplegar la aplicación.	Instalar la aplicación y la configuración de middleware.

Fuente: Patel & Cassou (2015, mi traducción).

2.2.4 Computación distribuida. En esta subsección se explica los conceptos de *cloud computing*, *fow computing* y *dew computing*, sus características y su alcance dentro del dominio de IoT. Estos tres conceptos se consideran como paradigmas de computación distribuida escalable vinculados en una división vertical, complementaria y jerárquica, del cual en la parte superior se encuentra

el *cloud computing*, en un nivel más abajo el *fow computing* y finalmente el *dew computing*, que se posiciona como un nivel básico para las dos anteriores (Skala, Davidovic, Afgan, Sovic, & Sojat, 2015).

2.2.4.1 *Cloud computing*. El paradigma de *cloud computing* es un modelo que permite el acceso a la red omnipresente a un conjunto de recursos computacionales configurables, que se pueden aprovisionar y liberar rápidamente con un mínimo esfuerzo de gestión o proveedor de servicios (Mell & Grance, 2011). Estos recursos computacionales pueden ser redes, servidores, almacenamiento, aplicaciones y servicios.

Lo que hace que el *cloud computing* sea tan popular son sus características, entre los que se encuentran el autoservicio bajo demanda o el concepto de 'pagar por usar', amplio acceso a la red, la puesta en común de los recursos, elasticidad rápida, servicio medido, facilidad de uso, la virtualización, que es centrado en internet, la variedad de los recursos que ofrece, la adaptabilidad automática, escalabilidad y la optimización de recursos (Mell & Grance, 2011; Vaquero, Rodero-Merino, Caceres, & Lindner, 2008).

La arquitectura del *cloud computing* está compuesto por cuatro capas: capa de hardware, capa de infraestructura, capa de plataforma y capa de aplicación (Q. Zhang, Cheng, & Boutaba, 2010). La capa de hardware es responsable de administrar los recursos físicos de la nube, incluyendo servidores físicos, enrutadores, conmutadores, sistemas de energía y refrigeración; la capa de infraestructura crea un conjunto de recursos de almacenamiento y computación dividiendo los recursos físicos utilizando tecnologías de virtualización; la capa de plataforma consta de sistemas operativos y marcos de aplicaciones; y la capa de aplicación consta de las aplicaciones reales de la nube.

2.2.4.2 *Fog computing*. El *fog computing* es un paradigma de recursos horizontales, físicos o virtuales, que reside entre dispositivos finales inteligentes y nubes o *Data Centers* tradicionales, y proporciona computación distribuida, almacenamiento y conectividad de red (Iorga et al., 2017). El *fog computing* puede resolver algunos problemas del *cloud computing* como la latencia no fiable, la falta de apoyo a la movilidad y la conciencia de la ubicación proporcionando recursos y servicios elásticos a los usuarios finales en el borde de la red, mientras que la computación en nube es más acerca de proporcionar recursos distribuidos en la red principal (Yi, Li, & Li, 2015).

El *fog computing* se caracteriza por la conciencia contextual de localización y la baja latencia, la distribución geográfica, las redes de sensores a gran escala y tener un gran número de nodos, apoyar la movilidad, las interacciones en tiempo real, predominio de acceso inalámbrico, heterogeneidad, interoperabilidad y

federación y apoyo al análisis en tiempo real y la interacción con la nube (Iorga et al., 2017).

Iogar *et al*. (2017) hacen una claridad entre el término *fog computing* y *edge computing*. *Edge* es la capa de red que abarca los dispositivos finales inteligentes y sus usuarios, y que también es conocida como red IoT; y aunque *fog computing* a menudo es llamado erróneamente *edge computing* hay unas diferencias claves:

- *Fog* funciona con la nube, mientras que *edge* se define por la exclusión de nubes y *fog*.
- *Fog* es jerárquica y *edge* tiende a limitarse a un pequeño número de capas periféricas.
- Además de la computación, *fog computing* también aborda la creación de redes, el almacenamiento, el control y la aceleración del procesamiento de datos.

2.2.4.3 *Dew computing*. El término más básico para definir el *dew computing* es como un *Cloud computing* a pequeña escala (Dobre, Mavromoustakis, Garcia, Ivanova Goleva, & Mastorakis, 2017). El *dew computing* es un paradigma que se basa en el concepto de micro-servicios, los cuales son eficientes e independiente a los servicios del *fog* y *cloud computing*, aunque puedan colaborar con ellos (Jóźwiak, 2017a; Skala et al., 2015).

La idea básica de este paradigma es explotar de una mejor forma las capacidades existentes, y las que puedan surgir, de los nodos de borde o *Edge nodes* de IoT para impulsar más la inteligencia, el poder de procesamiento y capacidades de comunicación hacia los nodos de borde donde se originan los datos y se usa la información, es decir, a los sensores, controladores y actuadores (Jóźwiak, 2017b).

2.3 ESTADO DEL ARTE

En esta sección se presenta dos estados del arte realizados en esta investigación. El primero trata de los modelos y *frameworks* de seguridad propuestos para aplicaciones IoT; el segundo, aborda el estado actual de la ciberseguridad en IoT.

2.3.1 *Frameworks* de seguridad para aplicaciones IoT. Este estado del arte tiene como objetivo realizar un diagnóstico y pronóstico sobre *frameworks* de seguridad en aplicaciones IoT. Para su planeación se definió los siguientes elementos:

- **Pregunta guía**. Este estado del arte busca dar respuesta a la siguiente pregunta: ¿Existe un modelo o *framework* para la gestión de la seguridad en la fase de diseño en toda la aplicación IoT?
- **Objetivos**. Los objetivos de este estado del arte son: (i) Conocer si existe un *framework* de seguridad que oriente la implementación de elementos de ciberseguridad desde la fase de diseño de una aplicación IoT, y (ii) conocer qué recursos del dominio de IoT y propiedades de la información son protegidos.
- **Justificación**. La importancia de realizar este estado del arte es conocer si existe un *framework* o modelo de seguridad, que oriente al equipo de desarrollo de aplicaciones IoT en la implementación de la ciberseguridad en todo el dominio IoT en el que podemos fundamentar esta investigación. De lo contrario, identificar la brecha de investigación en la que se aporte desde esta investigación. Otra razón subyacente es que no existen un estado del arte previo sobre esta temática.
- **Categorías de búsqueda**. Para las categorías de búsqueda, se eligió las siguientes palabras claves: *security model*, *security framework* e *Internet of Things*.
- **Fuentes de información**. La fuente de información que se consultó es la IEEE Xplore[6] y Scopus[7].

Seguidamente, se realizó la búsqueda en bases de datos especializadas descritas en 'Fuentes de información' usando la siguiente cadena de búsqueda:

'((*security model* OR *security framework*) AND (*Internet of things* OR IoT))'

Las categorías de análisis establecidas son las siguientes: (i) visión general de los *frameworks* y modelos, si son conceptuales o componentes de software, sus definiciones, propósito, y el dominio de la aplicación; (ii) las propiedades de la seguridad de la información (confidencialidad, integridad, disponibilidad, autenticación y confianza) que salvaguarden; y (iii) los recursos que ellos protegen.

Se analizaron 22 artículos, tres de los cuales son modelos y 19 son *frameworks*, como se muestra en la Tabla 3.

[6] Para más información sobre IEEE Xplore visite http://ieeexplore.ieee.org/Xplore/home.jsp
[7] Para más información sobre Scopus visite www.scopus.com

Tabla 3. *Frameworks* y modelos de seguridad para IoT.

ID	Propuesta	Framework
P1	Atamli & Martin (2014)	Modelo
P2	Bohli *et al*. (2015)	Framework
P3	Condry y Nelson (2016)	Modelo
P4	Chen *et al*.(2014)	Framework
P5	Ge y Kim (2015)	Framework
P6	Hellaoui *et al*. (2016)	Framework
P7	Hernandez-Ramos *et al*. (2015)	Framework
P8	Huang *et al*. (2015)	Framework
P9	Liu *et al*. (2014)	Framework
P10	Lize *et al*. (2014)	Framework
P11	Mozzaquatro *et al*. (2016)	Framework
P12	Namal *et al*. (2015)	Framework
P13	Neisse *et al*. (2014)	Framework
P14	Pacheco *et al*. (2016)	Framework
P15	Pacheco y Hariri (2016)	Framework
P16	Radomirovic (2010)	Modelo
P17	Serna *et al*. (2014)	Framework
P18	Singh y Bhandari (2016)	Framework
P19	Tahir *et al*. (2016)	Framework
P20	Yang y Fang (2011)	Framework
P21	Zegzhda y Stepanova (2015)	Modelo

Fuente: Autor.

2.3.1.1 Modelos de seguridad para IoT. En la literatura se han propuesto los siguientes modelos de seguridad para IoT:

Radomirovic (2010) propone un modelo llamado *Dense Internet of Things Model* el cual consiste en una red de comunicación asíncrona y un adversario de Dolev-Yao con habilidades de impresión de huellas digitales. Este modelo está enfocado a la capa de red de una aplicación IoT.

Atamli & Martin (2014) proponen un modelo conceptual para el modelado de la seguridad IoT basada en la teoría de los grafos. No se pudo identificar qué recursos protege este modelo ni hay claridad del problema que se quiso abordar.

Condry & Nelson (2016) proponen un modelo conceptual, que combina las capacidades de los dispositivos IoT inteligentes con las pasarelas del sistema de control mediante la respuesta en tiempo real para operaciones de control seguras. La solución propuesta utiliza dispositivos de extremo y *gateway*, emplea una combinación de procesamiento, criptografía, procesamiento de señal/imagen y capacidades de comunicación para funciones de autenticación y autorización. Según los autores, este modelo es más seguro, escalable y resistente con el rendimiento en tiempo real en comparación con los enfoques tradicionales. Este es un modelo que aborda la autenticación y la autorización,

pretende proteger las comunicaciones entre los clientes IoT y el dispositivo *gateway*.

2.3.1.2 *Frameworks* de seguridad para IoT. En la literatura se han propuesto los siguientes *frameworks* de seguridad para IoT:

Yang & Fang (2011) propusieron un *framework* conceptual que se basa en tres elementos vinculados entre sí: la comunicación, el control y la computación. Los autores presentan a IoT como el vínculo entre computación y control, siendo este vínculo el nuevo reto de seguridad. El *framework* propuesto es un "control de seguridad" entre la computación y el control y está compuesto por siguientes aspectos: seguridad de arquitectura, seguridad de terminales, seguridad de transporte, seguridad de control, protección de privacidad, gestión de seguridad, método y mecanismos de evaluación. Este *framework* aborda la seguridad y la privacidad de las aplicaciones IoT, pero no clarifica qué capas de IoT está dirigido ni que recursos protege.

Chen, Abdelwahed & Erradi (2014) llamado *ACSM Framework,* es un enfoque para la gestión de la seguridad cibernética basada en un modelo autónomo para ecosistemas de IoT. El *framework* ACSM está destinado a proteger los sistemas de computación en red de ciber adversarios. Este enfoque tiene como objetivo la realización de un sistema de autoprotección, que tiene la capacidad de estimar, detectar y reaccionar de forma autónoma a los ataques cibernéticos en una etapa temprana. Además, el enfoque integra diversas técnicas basadas en modelos que incluyen: (i) estimación en tiempo real y controles de seguridad basales para predecir y eliminar posibles ataques cibernéticos; (ii) análisis de datos para identificar y clasificar los ataques y (iii) un método de optimización multicriterio para seleccionar la respuesta activa óptima para desplegar contramedidas mientras se mantienen las funciones del sistema. Los recursos que protege este *framework* son los sistemas de red general. Los resultados experimentales muestran que ACSM realiza de manera eficiente un sistema de autoprotección sin requerir intervención humana y que sólo en casos excepcionales se requiere de intervenciones manuales. Como trabajo futuro lo autores planean extender el *framework* con respuestas de intrusión dinámicas reconfiguradas y análisis forense digital y, además, reducir el tiempo de recuperación de ataques desconocidos mejorando la precisión de detección de ataques y los procedimientos de aprendizaje para mejorar aún más las tasas de falsas alarmas.

Liu, Yin, Guo & Fang (2014) proponen un *framework* conceptual llamado AEC, el cual es un *framework* de autenticación mediante CSP (*Communicating Sequential Processes*) para IoT, para verificar la autenticación de protocolos. El problema que abordaron los autores es que la autenticación es una propiedad

de seguridad importante y sus definiciones formales propuestas no están ampliamente acordadas. Además, estas definiciones no pueden expresar fielmente los requisitos de seguridad y privacidad diversa en IoT. Para resolver estos problemas, los autores propusieron un AEC, que incluye tres formas de autenticación: autenticación de entidades, autenticación de acciones y autenticación de reclamaciones, y formalizó cada definición mediante CSP. La propiedad de la seguridad de la información que aborda este *framework* es la autenticación; no se pudo identificar a qué capa de aplicación está enfocada ni que recursos protege.

Lize, Jingpei & Bin (2014) proponen un *framework* general para el desarrollo de modelos de confianza en IoT. Este *framework* es resultado de que los autores establecieran un mecanismo formal de control de la gestión de confianza basado en el modelado de arquitectura de IoT. Para ello, los autores descompusieron IoT en tres capas: capa del sensor, capa del núcleo y capa de la aplicación. Cada capa está controlada por la gestión de confianza para fines especiales: auto-organización, enrutamiento afectivo y multi-servicio, respectivamente. En este mecanismo, la toma de decisiones final es realizada por el solicitante de servicios de acuerdo con la información de confianza recopilada, así como la política del solicitante. Una vez hecho esto y utilizando la teoría de conjuntos formal basada en semántica y difusa, realizando todo el mecanismo anterior cuyo resultado es el *framework* propuesto el cual protege todos los recursos de las capas del sensor, del núcleo y de aplicación.

Neisse *et al*. (2014) proponen un kit de herramientas de seguridad basado en modelos llamdo SecKit, el cual apoya la especificación y la aplicación de políticas de seguridad integradas en todas las capas del *framework* iCore: Service, CVO y VO. SecKit es una colección de metamodelos que proporciona la base para herramientas de ingeniería de seguridad, complementos, componentes de tiempo de ejecución y extensiones para abordar los requisitos de seguridad, protección de datos y privacidad. SecKit está orientado a las ciudades inteligentes y busca proteger los datos de los usuarios.

Serna, Morales, Medina y Luna (2014) proponen un *framework*, el cual es un artefacto de software, que aborda los problemas de seguridad y privacidad en VANETs (*Vehicular Ad-Hoc Network*). El marco propuesto consiste en tres elementos: (i) un sistema de autenticación entre dominios capaz de proporcionar un servicio de estado de certificados en tiempo casi real; (ii) un mecanismo para evaluar cuantitativamente el nivel de confianza de una entidad emisora y establecer una relación de interoperabilidad al vuelo y (iii) un modelo de mejora de la privacidad que aborda la privacidad en términos de enlazamiento. Este framework se ubica en la capa de red de una aplicación IoT de tipo VANET y protegiendo sus comunicaciones.

Bohli, Skarmeta, Moreno, García y Langendörfer (2015) proponen un *framework* conceptual llamado SMARTIE (*Secure and sMARter ciTIEs data management*), el cual los autores definen como una plataforma segura para proteger sensores y dispositivos, permitir el control de acceso a los recursos y proporcionar capacidades seguras de almacenamiento y procesamiento de datos. Este *framework* está enfocado al desarrollo de aplicaciones en el dominio de las ciudades inteligentes.

Ge y Kim (2015) proponen un *framework* de software cuyo objetivos principales es describir todos los posibles caminos de ataque en IoT, evaluar el nivel de seguridad de IoT a través de métricas de seguridad y evaluar la efectividad de las estrategias de defensa. En este *framework* hay cinco pasos: (i) preprocesamiento, (ii) generación de modelos de seguridad, (iii) visualización y almacenamiento, (iv) análisis de seguridad, y (v) cambios y actualizaciones. Este *framework* está enfocado a la fase de diseño de la capa de red de una aplicación IoT genérica. Como trabajo futuro, los autores consideran introducir múltiples objetivos, considerar diferentes estrategias de defensa, abordar la heterogeneidad y la movilidad.

Namal, Gamaarachchi, MyoungLee y Um (2015) proponen un *framework* de software, el cual es un *framework* autónomo de gestión de confianza para aplicaciones y servicios de IoT basados en la nube y altamente dinámicos. Este *framework* se basa en el modelo MAPE-K, compuesto por las fases de analizar, planificar, ejecutar y conocimiento, y usando el lazo de retroalimentación de dicho modelo evalúa el nivel de confianza en un ecosistema de la nube de IoT. Los autores utilizaron los bucles de control de retroalimentación del modelo MAPE-K para mejorar la consistencia del sistema, la robustez y la escalabilidad con la introducción de conceptos de nube.

Zegzhda y Stepanova (2015) proponen un modelo de software de estado finito del comportamiento del agente de IoT que tiene como objetivo mantener la sostenibilidad topológica adaptativa en un entorno hostil a través del uso de la arquitectura híbrida con centros de control reasignables y el equilibrio de número de enlaces entre entidades IoT. Es un modelo para el desarrollo, que se ubica en la capa de red de una aplicación IoT.

Huang *et al*. (2015) proponen SecIoT, un *framework* de seguridad para IoT, que incluye mecanismos de autenticación seguros, un sistema flexible de acceso basado en roles y una interfaz de indicadores de riesgo de seguridad para ayudar a los usuarios a entender y controlar los riesgos de seguridad del sistema. SecIoT proporciona autenticación esencial, asegura las comunicaciones, soporta la autorización del usuario y ofrece notificación de riesgos. Este framework está orientado a la protección de los dispositivos, utilizando la autenticación y la

autorización. Como trabajo futuro los autores consideraron la mejora de la disponibilidad y la gestión de la confianza.

Hellaoui, Bouabdallah y Koudil (2016) proponen un framework de software llamado TAS-IoT, es un framework de seguridad adaptativo basado en la confianza. Conta de dos algoritmos para autenticar el mensaje recibido o no, en función del nivel de confianza que asocia al remitente del mensaje. Esta solución de seguridad adaptativa tiene como objetivo reducir la sobrecarga de autenticación mediante la autenticación de paquetes sólo cuando es requerida. Es un modelo orientado a la fase de desarrollo de aplicaciones IoT y a la capa de dispositivos, protegiendo a los nodos. Como trabajo futuro los autores creen interesante considerar recomendaciones poco fiables y realizar evaluaciones más avanzadas, como, por ejemplo, redes con pérdidas y otras topologías.

El trabajo de Hernandez-Ramos *et al*. (2015) es un framework conceptual que propone un conjunto de mecanismos de autenticación y autorización ligeros para soportar objetos inteligentes durante su ciclo de vida, los cuales son compatibles con el modelo de referencia arquitectónico de la IoT-A. La arquitectura resultante tiene por objeto proporcionar un enfoque de seguridad integral que se aproveche en el diseño de nuevos y ligeros protocolos de seguridad para entornos restringidos para IoT. Este *framework* se centra en la fase de diseño de aplicaciones IoT y el recurso que protege son los dispositivos. Los autores afirman que su trabajo futuro se centra en la integración de la solución propuesta en el IETF ACE WG, así como en la definición de escenarios alternativos, con el fin de evaluar y comparar la idoneidad de dicho escenario.

Mozzaquatro *et al*. (2016) propone un *framework* de software el cual es un mecanismo formal de control de la gestión confianza basado en el modelado de arquitectura de IoT. Los autores dividieron IoT en tres capas: capa de sensor, capa de núcleo y capa de aplicación. Cada capa está controlada por la gestión de confianza para fines especiales: auto-organizado, enrutamiento afectivo y multi-servicio, respectivamente. Y la toma de decisiones final es realizada por el solicitante de servicios de acuerdo con la información de confianza recopilada, así como la política del solicitante. Este *framework* aborda la fase de desarrollo de aplicaciones IoT y está orientado a salvaguardar la privacidad. No se identificó a qué capa está dirigido este framework, ni que recursos protege.

Pacheco y Hariri (2016) proponen un *framework* conceptual llamado *IoT Security Framework*, el cual es una arquitectura que se puede utilizar para guiar el desarrollo de la seguridad de las infraestructuras inteligentes IoT. Este *framework* puede utilizarse para identificar las vulnerabilidades potenciales y el mecanismo de mitigación apropiado. Este framework está orientado al dominio de las infraestructuras ciber inteligentes como son los hogares y edificios

inteligentes, está orientado a todas las capas de IoT y considera la protección de todos los recursos de una aplicación de infraestructuras ciber inteligentes.

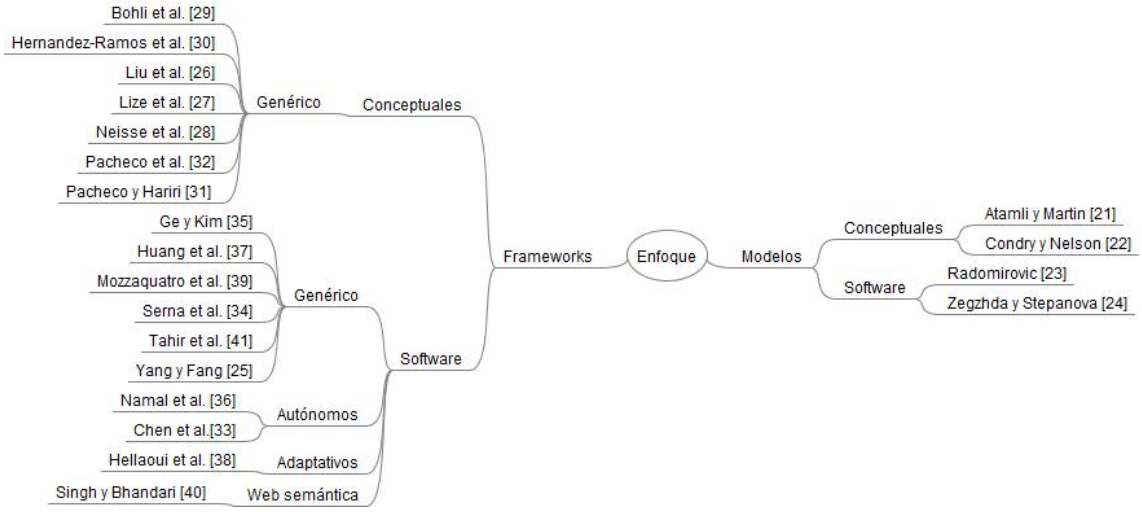
Pacheco *et al*. (2016) proponen un *framework* conceptual llamado *IoT Security Development Framework* – ISDF, cuyo principal objetivo, es proporcionar el apoyo arquitectónico para desarrollar servicios de IoT altamente seguros y confiables, que puedan detectar y tolerar de forma proactiva los daños maliciosos que pueden deberse a ataques; fallos, ya sean maliciosos o naturales, o a los accidentes. ISDF está orientado al diseño de aplicaciones IoT para vehículos inteligentes. Igualmente, considera aspectos de seguridad y privacidad, autenticación y autorización, integridad y no repudio; y busca proteger: dispositivos, como sensores, actuadores y los dispositivos de entretenimiento del vehículo; las comunicaciones, los servicios y las aplicaciones.

Singh y Bhandari (2016) proponen un *framework* basado en web semántica, llamado NSSA, para la seguridad de la red consciente a la situación. Los autores quisieron abordar los problemas que tenían los enfoques tradicionales de seguridad de redes, los cuales requerían un modelo formal para representar entidades de una red y debería tener la capacidad de acomodar nuevas entidades, representar las relaciones entre las entidades y también adaptarse a los cambios de configuración en la red. Como trabajo futuro, los autores consideran la implementación detallada de las ontologías en el editor de ontologías y la definición de políticas de seguridad usando el lenguaje de las reglas web semánticas.

Tahir *et al*. (2016) proponen un framework de software, llamado ICMKeyStream, el cual protege contra amenazas a nivel de dispositivos y red. Este framework está basado en la tecnología ICMetric, junto con *Secure Remote Rabbit Protocol*, para asegurar las entidades y sus intercomunicaciones para proporcionar seguridad para el IoT, proporcionando así, autenticación, confidencialidad y no repudio para flujos de datos continuos.

2.3.1.3 Tendencias de construcción. Dentro de cada propuesta de seguridad se identificó los tipos: los enfoques conceptuales y los de software; y dentro de estos, se identificaron algunas tendencias en la forma en que son construidos, como se muestra en la Figura 5.

Figura 5. Tendencias en la construcción de *frameworks* y modelos de seguridad para IoT.



Fuente: Autor.

2.3.1.4 Recursos IoT que protegen. También se analizó qué tipo de recurso IoT protegen cada propuesta de seguridad analizadas en este estado del arte. Para clasificar los recursos protegidos de definieron cuatro capas, de acuerdo a la arquitectura de referencia de la ITU-T (2012): capa de dispositivo, capa de red, capa de servicio y capa de aplicación; también se definió un campo extra para clasificar a aquellas propuestas que no describieron con claridad hacia qué capa estaba enfocada dicha solución, como se muestra en la Tabla 4.

Tabla 4. Recursos IoT protegidos por *frameworks* y modelos de seguridad.

Propuesta	Capa de dispositivo	Capa de red	Capa de servicio	Capa de aplicación	No se especifica
P1					x
P2	x				
P3		x			
P4		x			
P5		x			
P6	x				
P7	x				
P8	x				
P9					x
P10	x	x		x	
P11					x
P12			x		
P13	x		x		
P14	x	x		x	
P15	x	x		x	
P16		x			
P17		x			
P18		x			

Propuesta	Capa de dispositivo	Capa de red	Capa de servicio	Capa de aplicación	No se especifica
P19	x	x			
P20					x
P21		x			

Fuente: Autor.

Como se aprecia en la tabla anterior, el mayor esfuerzo para proteger recursos IoT se ubican en la capa de dispositivos y en la capa de red. Existen propuestas más completas, en las cuales se abordan tres de las cuatro capas definidas, como son las propuestas de Lize *et al*. (2014), Pacheco *et al*. (2016) y Pacheco y Hariri (2016).

2.3.1.5 Propiedades de seguridad de la información que protegen. Las aplicaciones IoT deben salvaguardar los datos considerando una serie de propiedades como son la confidencialidad, integridad, disponibilidad, la autenticación, autorización, el no repudio, y la privacidad (Cirani, Ferrari, & Veltri, 2013; Heer et al., 2011). Las propiedades que cada propuesta aborda se muestran en la *Tabla 5*.

Tabla 5. Propiedades de seguridad protegidas por modelos y *frameworks*.

Propuesta	Seguridad	Privacidad	Confianza	Autenticación	Autorización	No repudio
P1	x					
P2	x	x				
P3	x					
P4				x	x	
P5	x	x				
P6			x			
P7				x	x	
P8				x	x	
P9				x		
P10			x			
P11		x				
P12			x			
P13	x	x				
P14	x					
P15	x	x		x	x	x
P16	x	x				
P17	x	x				
P18	x					
P19				x	x	x
P20	x	x				
P21	x	x				

Fuente: Autor.

La mayor tendencia es la protección de la seguridad y privacidad de los datos. Seguidamente, garantizar la autenticación y autorización, propiedades que van de la mano: mientras la autenticación establece quién está ingresando al sistema, la autorización determina qué permisos tiene ese usuario dentro del mismo.

2.3.1.6 Conclusiones y brecha de investigación. Investigadores han propuesto mecanismos, como modelos y *frameworks*, para abordar los problemas de ciberseguridad en las aplicaciones IoT. Dentro de estos mecanismos atienen una o varios de las propiedades de la información: confidencialidad, integridad, disponibilidad, autenticación y confianza. Pero los modelos y *frameworks* que se están proponiendo están atendiendo partes del domino de la ciberseguridad en IoT, ya sea por que atienden algunas de las propiedades de la información, o porque se enfoque en una de las capas una aplicación IoT y a proteger los recursos que en ellas están.

Las propuestas más completas son las hechas por Pacheco y Hariri (2016) y Pacheco *et al*. (2016), cuyo *frameworks* atienden los asuntos de seguridad en todas las capas de IoT: la capa de dispositivos, la de red y la capa de aplicaciones, y pretender guiar y ser un apoyo arquitectónico para los desarrollo de aplicaciones IoT. La limitación es que ambos *frameworks* están dirigidos a un dominio de aplicación específico: las infraestructuras ciberinteligentes y los vehículos inteligentes, respectivamente.

Existe una brecha de investigación y técnicos en mecanismos que ayuden y guíen en la implementación de la ciberseguridad en todo el dominio de IoT, el cual se consideren todas las capas de una aplicación IoT y esté orientada a aplicaciones IoT genéricas. Esta brecha puede ser resultado de que la ciberseguridad en IoT no está del todo claro, no sé sabe qué proteger, y al ser un dominio tan amplio, en el que intervienen muchas tecnologías heterogenias se hace difícil abordar la ciberseguridad de forma global en el dominio IoT. Este tipo de mecanismos, ya sean modelos, *frameworks* o guías, es necesario en el dominio de IoT, un paradigma que está ganando popularidad en la investigación, la industria y los gobiernos pero que su mayor reto es sus problemas de seguridad y privacidad. Esto genera que se retrase la implementación y el crecimiento del IoT.

2.3.2 Estado actual de la ciberseguridad en IoT. La seguridad ha sido un reto para todos los sistemas desde sus inicios, comenzando con los sistemas informáticos tradicionales (Kim & Solomon, 2016) y redes informáticas (Pawar & Anuradha, 2015), y se ha extendido a cada nuevo sistema que ha aparecido a través de los años, como: dispositivos móviles (NowSecure, 2016), las redes

inteligentes (Boroojeni, Amini, & Iyengar, 2016), *cloud computing* (Hussein & Khalid, 2016) y en los paradigmas de Internet de las cosas (Sicari, Rizzardi, Grieco, & Coen-Porisini, 2015) y *fog computing* (Stojmenovic, Wen, Huang, & Luan, 2015).

La seguridad es uno de los retos que enfrenta el Internet de las cosas (Mahalle, Anggorojati, Prasad, & Prasad, 2013). Entre los retos de seguridad que afrenta el IoT se encuentra principalmente la seguridad y la privacidad de los datos (I. Lee & Lee, 2015; Da Xu et al., 2014; Atzori et al., 2010) y la confianza de los recursos (Miorandi, Sicari, De Pellegrini, & Chlamtac, 2012; Sicari et al., 2015).

Garantizar la seguridad es relevante en todos los dominios de aplicación de IoT, por ejemplo: los riesgos del IoT en la salud, donde se manejan datos sensibles sobre los pacientes (Riazul Islam, Daehan Kwak, Humaun Kabir, Hossain, & Kyung-Sup Kwak, 2015; Supriya & Padaki, 2016) y en las ciudades inteligentes, donde se toman datos de personas y sus círculos sociales, se controlan las instalaciones de la ciudad e influye en las vidas de las personas (Elmaghraby & Losavio, 2014; K. Zhang et al., 2017). Asimismo, en los demás dominios como son: redes eléctricos inteligentes o *Smart Grids* (Dalipi & Yayilgan, 2016), la industria (Sadeghi, Wachsmann, & Waidner, 2015), las casas inteligentes (C. Lee, Zappaterra, Kwanghee Choi, & Hyeong-Ah Choi, 2014) y los vestibles (Thierer, 2014) por nombrar algunos de ellos.

Para realizar este del arte se hizo una revisión documental que incluyen artículos científicos, sitios web y reportes de seguridad de empresas reconocidos en el sector de la seguridad informática. Este estado del arte se organizó en dos secciones: los problemas de seguridad relacionado con el malware en IoT y los problemas de seguridad relacionados con los dispositivos IoT.

2.3.2.1 *Malware* en IoT. Así como los retos de seguridad, las técnicas utilizadas para los ataques cibernéticos han evolucionado al paradigma de Internet de las cosas. Una de estas técnicas son las familias de malware, como troyanos, gusanos y ransomware, también está migrando al entorno de IoT.

Un troyano es un programa de computadora que parece tener una función útil, pero también tiene una función oculta y potencialmente maliciosa que evade los mecanismos de seguridad, a veces explotando autorizaciones legítimas de una entidad del sistema que invoca el programa (Shirey, 2007). Ejemplos de troyano en IoT son: IRCTelnet, un troyano que apunta a dispositivos IoT basados en Linux, con el propósito de agregar esos dispositivos a una botnet y llevar a cabo ataques DDoS (Cimpanu, 2016); y NyaDrop, que atacan los puertos Telnet y abrir una puerta trasera en el dispositivo infectado y descargar el troyano Nya si

el dispositivo IoT utiliza una arquitectura MIPS de 32 bits (Malware Must Die, 2016).

Además, hay troyanos de hardware a nivel de dispositivo y de red (Subramani, Antonopoulos, Nosratinia, & Makris, 2016): en el nivel de dispositivos, troyanos de hardware en ICs criptográficos inalámbricos, permitiendo robar información confidencial y ocultar los datos filtrados como estructura "agregada" del perfil de transmisión, aprovechando las variaciones del proceso; y a nivel de la red, troyanos de hardware en redes inalámbricas, permitiendo robar información sensible en 802.11a/g y explotando el espacio no utilizado (Gap) entre estándares inalámbricos, punto de funcionamiento del dispositivo y especificaciones.

En cuanto a los gusanos, el cual se define como un programa autónomo que es autorreplicable y auto-propagable, que utiliza mecanismos de redes para difundirse (CNSS, 2010) se encuentran algunos ejemplos, como: ArduWorm, un gusano para Arduino Yun (Pastrana, Rodriguez-Canseco, & Calleja, n.d.); Darlloz, un gusano utilizado para la criptografía de minas, que apunta a arquitecturas Intel x86, ARM, MIPS y arquitecturas PowerPC (Hayashi, 2014); Hajime, un gusano para dispositivos IoT que se propaga en dispositivos que ejecutan servidores Telnet con credenciales predeterminadas inseguras (Edwards & Profetis, 2016); Un gusano desarrollado por investigadores que se extiende utilizando las lámparas inteligentes Philips Hue como una plataforma que utiliza sólo su conectividad inalámbrica integrada ZigBee y su proximidad física \ cite {ronen2016iot}; Y la radiación, un gusano está dirigido a dispositivos de televisión de circuito cerrado – CCTV (CyberX, 2016).

El *ransomware* es un tipo de malware que impide o limita el acceso de los usuarios a un sistema, ya sea bloqueando la pantalla del sistema o cifrando los archivos de los usuarios a menos que se pague un rescate para obtener la clave de descifrado (Trend Micro, 2015). En cuanto a la *ransomware* en IoT, ya está empezando a utilizar el concepto de *ransomware* de las cosas – RoT (Cobb, 2017), a la que Stephen Cobb llamar Jackware. El Jackware es una forma especializada de ransomware que busca tomar el control de un dispositivo cuyo propósito principal no es el procesamiento de datos o las comunicaciones digitales, por ejemplo, un coche, cuyo propósito es llegar de A a B (Cobb, 2016b). La primera víctima de Jackware en 2017 fue el sistema de llave electrónica de un hotel austriaco de cuatro estrellas (Schrott, 2017). Ransomware continuará proliferando, poniendo en riesgo los datos críticos, e incluso la vida, como los ataques ya conocidos por los hospitales y la infraestructura crítica (RSA, 2016).

En la actualidad, los troyanos para IoT pretenden crear botnets para llevar a cabo ataques de denegación de servicio distribuidos (DDos). Una botnet es una red de computadoras comprometidas, que es controlada por un tercero y usada para

transmitir malware o para lanzar ataques. En 2016, las botnets más conocidas y poderosas fueron Mirai y Leet. El Mirai es una botnet integrada por dispositivos IoT inseguros, al obtener acceso a ellos con credenciales de inicio de sesión comunes, y el tráfico generado excedió los 620 Gbps, otro ataque DDoS generó un tráfico de 1.1 Tbps (US-CERT, 2016).

El malware está aprovechando los errores de diseño, la configuración y la falta de gestión de las tecnologías utilizadas en las aplicaciones IoT y seguirá evolucionando y extendiéndose a nuevos sistemas que aparezcan en el futuro. Hasta que se resuelvan estos problemas de seguridad, en Internet y en IoT, todos los dispositivos serán vulnerables, aumentando los riesgos informáticos y consecuentemente los ataques informáticos, las filtraciones de información que afectan la seguridad y privacidad de los sistemas y usuarios y las pérdidas económicas.

2.3.2.2 Dispositivos IoT. El *Internet of Things* está integrado para una gran cantidad de dispositivos, que son vulnerables, aumentando el riesgo de afectar la seguridad y la privacidad de los usuarios y las empresas, y la integridad de los usuarios y las personas. Todo los dispositivos conectados a Internet, que es la filosofía de IoT, puede ser hackeado, como los coches (Currie, 2016; Miller & Valasek, 2015), smart TV (S. Lee & Kim, 2013), cámaras IP (Fox-Brewster, 2016), CCTV (Hioureas, 2015), dispositivos médicos (Finkle, 2016), juguetes infantiles (Gibbs, 2015), monitores de bebes (Ross, 2016), y timbres de puertas (Leyden, 2016), para nombrar algunos de los miles de dispositivos que están conectados a Internet. Los dispositivos vulnerables conectados pueden ser utilizan para el montaje de ataques contra cualquier tipo de destino y servicio de Internet y ataques contra las redes internas que alojan estos dispositivos conectados (Caltum & Segal, 2016).

Un estudio realizado por *Hewlett-Packard Enterprise* (2015) en el cual se probaron 10 dispositivos IoT, arrojo que: seis de cada 10 dispositivos que proporcionan interfaces de usuario eran vulnerables a una gama de problemas tales como XSS persistente y credenciales débiles; el 80% de los dispositivos, junto con sus componentes de aplicaciones en la nube y móviles, no han requerido contraseñas de suficiente complejidad y longitud; el 70% de los dispositivos, junto con su aplicación en la nube y en el móvil, permiten a un atacante identificar cuentas de usuario válidas a través de la enumeración de cuenta y el 70% de los dispositivos utilizan el servicio de red sin cifrar.

Samy Kamkar, un hacker ético, fue comisionado por *ForeScout Technologies*[8] para investigar los riesgos de seguridad que plantean los dispositivos IoT en entornos empresariales. Kamkar evaluó siete dispositivos y los hallazgos clave

[8] Para más información sobre la empresa visite su sitio web: www.forescout.com

fueron: (i) los siete dispositivos IoT pueden ser hackeados en tan sólo tres minutos, pero puede tomar días o semanas para remediar; (ii) si cualquiera de estos dispositivos se infectan, los hackers pueden plantar puertas traseras para crear y lanzar un ataque automatizado de DDoS de botnet IoT; y (iii) los ciberdelincuentes pueden aprovechar las técnicas de interferencia o falsificación para hackear sistemas de seguridad empresariales inteligentes, permitiéndoles controlar sensores de movimiento, cerraduras y equipos de vigilancia. Estos resultados se muestran en *IoT Enterprise Risk Report* (ForeScout Technologies, 2016).

Este mismo informe también indica que, cuando los dispositivos son hackeados exitosamente, estos dispositivos son una puerta de entrada a la red de la empresa más amplia y los ciberdelincuentes pueden descomponer aún más, lo que lleva a: (i) fisgonear y grabar de llamadas a través de teléfonos VoIP; (ii) con los sistemas HVAC[9] pueden forzar salas críticas (por ejemplo, salas de servidores) a sobrecalentar la infraestructura crítica y, en última instancia, causar daño físico; (iii) con sistemas de seguridad conectados, estos pueden ser deshabilitados para permitir robos físicos; (iv) a través de la televisión inteligente, pueden espiar a través de vídeo y micrófono; (v) utilizando impresoras conectadas, pueden acceder a información privada de la compañía y del usuario; y (vi) utilizando frigoríficos inteligentes, pueden obtener credenciales de usuario y utilizar bombillas inteligentes, pueden extraer credenciales de Wi-Fi para realizar nuevos ataques (ForeScout Technologies, 2016).

2.3.2.3 Conclusiones del estado del arte. El Internet de las cosas y sus dispositivos se ha convertido en un objetivo muy llamativo para los ciberdelincuentes, los cuales aprovechan la poca madurez de la ciberseguridad en todo el entorno de IoT, como por ejemplo, la poca o nula implementación de la seguridad en el diseño de productos y dispositivos IoT; las malas prácticas en la implementación de dichas tecnologías en hogares y organizaciones, como el cambio de las credenciales por defecto traídas por los dispositivos de fábrica y la mala gestión de las redes que soportan las aplicaciones de IoT. Estas condiciones tienen un impacto negativo en la privacidad y seguridad de los datos, la infraestructura de TI y la economía de las organizaciones y ciudadanos en general.

2.4 MARCO NORMATIVO Y ESTÁNDARES

En esta sección se presenta una revisión de las normas, políticas o estándares que influencian o aportan en este trabajo de investigación. En este trabajo se

[9] HVAC: *Heating, Ventilating, and Air Conditioning*, el cual es un sistema de climatización para espacios habitados.

tuvo en cuenta normas que fueran ampliamente conocidas y aceptadas a nivel internacional para proveer a los productos derivados de esta investigación una compaginación con buenas prácticas independientemente de la normatividad propia de cada país.

2.4.1 Estándar ISO/IEC 25.010:2011. La ISO/IEC 25010:2011 *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models* (2011) define un modelo de calidad del producto compuesto de ocho características (idoneidad funcional, eficiencia de rendimiento, compatibilidad, facilidad de uso, fiabilidad, seguridad, facilidad de mantenimiento y portabilidad) que se subdividen en sub-características relacionadas con las propiedades estáticas del software y las propiedades dinámicas de la computadora sistema. Este modelo es aplicable tanto a sistemas informáticos como a productos de software.

2.4.2 Estándar ISO/IEC 27.001:2013. La ISO/IEC 27001:2013 *Information technology -- Security techniques -- Information security management systems -- Requirements* (2013) es una norma internacional emitida por la Organización Internacional de Normalización - ISO y la Comisión Electrotécnica Internacional - CEI preparada para proporcionar los requisitos para establecer, implementar, mantener y mejorar continuamente un sistema de gestión de la seguridad de la información.

El objetivo principal de esta norma es proteger la confidencialidad, integridad y disponibilidad de la información en una empresa a través de la gestión de riesgos, que comienza con la evaluación de los riesgos y su posterior mitigación o tratamiento. Esta norma está dividida en 11 secciones y el Anexo A, que proporciona un catálogo de 114 controles –medidas de seguridad– distribuidos en 14 secciones.

2.4.3 Estándar ISO/IEC/IEEE 27017:2015. La ISO/IEC/IEEE 27017:2015 *Information technology -- Security techniques -- Code of practice for information security controls based on ISO/IEC 27002 for cloud services* (2015), es un estándar internacional que proporciona directrices para los controles de seguridad de la información aplicables a la provisión y uso de servicios en la nube, proporcionando controles e instrucciones de implementación tanto para los proveedores de servicios en la nube como para los clientes. Esta recomendación proporciona guías de implementación adicionales para los controles relevantes especificados en ISO/IEC 27002 y controles adicionales con directrices de implementación que se refieren específicamente a los servicios en la nube.

2.4.4 Estándar ISO/IEC 27032:2012. La ISO/IEC 27032:2012 *Information technology -- Security techniques -- Guidelines for cybersecurity* (2012) es un estándar internacional que proporciona una visión general de la ciberseguridad y su relación con otros tipos de seguridad. Asimismo, aborda una definición de las partes interesadas y una descripción de los roles de ciberseguridad, la orientación para abordar problemas comunes de la ciberseguridad y un marco para permitir a los interesados colaborar en la resolución de estos problemas.

Esta norma tiene el objetivo de proporcionar una guía de ciberseguridad, donde se destacan sus aspectos y dependencias en otros dominios de la seguridad como es la seguridad de la información, seguridad de la red, protección de infraestructura de información crítica – CIIP, por sus siglas en inglés *critical information infrastructure protection*, y cubre las prácticas de seguridad de referencia para los interesados en el ciberespacio.

2.4.5 Estándar ISO/IEC/IEEE 42010:2011. La ISO/IEC/IEEE 42010:2011 *Systems and software engineering -- Architecture description* (2011) es un estándar internacional que aborda la creación, análisis y mantenimiento de arquitecturas de sistemas a través del uso de descripciones de arquitectura. Esta norma estableció un modelo conceptual de la descripción de la arquitectura y los contenidos requeridos especificados de una descripción de la arquitectura. Asimismo, introduce puntos de vista Arquitectura, marcos de arquitectura y lenguajes de descripción de arquitectura para codificar convenciones y prácticas comunes de descripción de arquitectura y su contenido requerido.

2.4.5 Aportes de la normatividad a este trabajo. Estas normas contribuirán en este trabajo de investigación al dar los lineamientos y buenas prácticas en temas relacionados con la ciberseguridad y la ingeniería del software. La ISO/IEC 27001:2013, la ISO/IEC 2032:2012 y la ISO/IEC/IEEE 27017:2015 aportan el panorama general del área de la ciberseguridad, buenas práctica y controles necesarios para la seguridad de la información. La ISO/IEC 25010:2011, los requerimientos de calidad para el software y los sistemas, que, en este proyecto, solo se considerará el requerimiento de la seguridad.

2.5 MARCO CONTEXTUAL Y ANTECEDENTES

En esta sección se describe el contexto organizacional, comunitario, social o institucional en el que se realizará la investigación. Como antecedentes se describe algunos proyectos relevantes realizados en ese contexto que allanaron el camino para este proyecto. Los proyectos descritos están relacionados con la seguridad de la información en IoT.

2.5.1 Centro de Excelencia y Apropiación en Internet de las Cosas. El Centro de Excelencia y Apropiación en Internet de las Cosas – CEA-IoT[10] es una iniciativa promovida por el Ministerio de las TIC de Colombia, con el apoyo del Departamento Administrativo de Ciencia, Tecnología e Innovación - Colciencias, el cual busca posicionar a Colombia como líder regional en TIC.

El CEA-IoT es una alianza entre cinco universidades colombianas –Pontificia Universidad Javeriana Bogotá, Pontificia Universidad Javeriana Cali, Universidad Tecnológica de Bolívar, Universidad Santo Tomás y Universidad Autónoma de Bucaramanga–, líderes tecnológicos mundiales –como Microsoft, Intel y Hewlett-Packard– y las empresas colombianas para resolver las necesidades de los diferentes sectores productivos, potenciando el desarrollo económico del país desde la tecnología y la innovación a través del IoT, todo esto apalancado en la formación de talento humano especializado en IoT.

2.5.2 Fundación OWASP. La *Open Web Application Security Project* – OWASP[11] es una fundación, organización internacional y una comunidad abierta dedicada a permitir a las organizaciones concebir, desarrollar, adquirir, operar y mantener aplicaciones que se puedan confiar, y que aboga por acercarse a la seguridad de las aplicaciones como un problema de personas, procesos y tecnología porque los enfoques más efectivos para la seguridad de las aplicaciones incluyen mejoras en todas estas áreas (OWASP, 2017a).

OWASP es reconocido por apoyar la Guía de Pruebas de OWASP, un marco de pruebas completo desarrollado para ayudar a las personas a entender qué, por qué, cuándo, dónde y cómo probar las aplicaciones web.

Asimismo, esta organización apoya el Proyecto OWASP de Internet de las Cosas (IoT), el cual «está diseñado para ayudar a los fabricantes, desarrolladores y consumidores a comprender mejor los problemas de seguridad asociados con Internet de las Cosas ya permitir a los usuarios en cualquier contexto tomar mejores decisiones de seguridad Cuando construya, despliegue o evalúe tecnologías IoT» (OWASP, 2017b).

Este proyecto es de libre de uso y proporciona información sobre las áreas de ataque de IoT, las vulnerabilidades de IoT, el análisis de firmware, las debilidades del software ICS/SCADA, las guías de prueba de IoT, la guía de seguridad IoT, los principios de seguridad IoT, el marco de evaluación de IoT, orientaciones para desarrolladores, consumidores y fabricantes y los principios de diseño.

[10] Para más información sobre el centro pude ingresar a su sitio web: www.cea-iot.org
[11] Para mayor información sobre el proyecto, ingrese a su sitio web: www.owasp.org

2.6 CONSIDERACIONES FINALES DEL CAPÍTULO

El Internet de las cosas en un paradigma emergente que integra una gran variedad de tecnologías heterogenias, pero que los problemas de seguridad y privacidad son uno de sus mayores retos. La ciberseguridad es la encargada de proteger los activos de información a través de la gestión de las amenazas, pero esta área no ha alcanzado un nivel de madurez en el dominio del IoT, donde no existe guías que aclaren qué se debe proteger y de qué manera.

Los modelos son representación de un proceso o concepto del mundo real. A través de los modelos, se puede representar, a través de un lenguaje de modelamiento, el dominio de la ciberseguridad en IoT, identificando los elementos que la conforman y sus relaciones entre ellos. Este tipo de modelo puede llevarse a una arquitectura de aplicación IoT, ser implementada, probada y cuyo resultado sea una aplicación funcional.

Este tipo de desarrollo, en áreas como las telecomunicaciones, la telemática y recientemente, el internet de las cosas, no se guían por metodologías, técnicas y abstracciones para garantizar en mayor medida la calidad de la solución. Este tipo de elementos ya lo ha creado y mejorado la ingeniería del software, un área madura en el tema de desarrollos. La ingeniería del software no solo se aplica para la producción del software, sino que puede llevarse a otras áreas del conocimiento, como las mencionadas anteriormente, para el desarrollo de sistemas más complejos.

La ingeniería del software ha definido un ciclo para el desarrollo de software o sistemas. Este ciclo comienza con la gestión de los requisitos funcionales y de calidad del sistema. Entre los requisitos de calidad se encuentra el de la seguridad. Una buena gestión de los requisitos hace que se comprendan de mejor forma las necesidades reales, que estas se traduzcan en diseños que luego serán implementados y probados, resultado una aplicación funcional que cumpla con criterios de cualitativos y subjetivos de los usuarios, de forma eficiente y oportuna.

La ingeniería del software, la ciberseguridad y el modelamiento pueden aportar significativamente al internet de las cosas. En el punto de intersección de estas cuatro áreas se desarrolla este trabajo de investigación, y se espera contribuir significativamente con los resultados esperados.

3. ASPECTOS METODOLÓGICOS

En este capítulo se presenta los aspectos metodológicos que orientaron y sustentaron esta investigación. Este capítulo está dividido en cuatro secciones: En la sección 3.1, se describe el tipo de investigación que se desarrolló en este trabajo y su enfoque investigativo. En la sección 3.2, se presenta el universo objeto de estudio y la muestra con la cual se trabajó. En la sección 3.3, se enumeran las técnicas e instrumentos usados para la recolección de información. Finalmente, en la sección 3.4, se muestra la metodología usada y se detallan y explican las actividades realizadas durante esta investigación para alcanzar con los objetivos propuestos.

3.1 TIPO Y ENFOQUE DE INVESTIGACIÓN

El tipo de investigación en el que se enmarca este proyecto es la *investigación aplicada*, que se caracteriza por generar nuevo conocimento o usar el ya existente para dar respuesta a un problema o necesidad identificada (Colciencias, 2016). Esta investigación aplicada tiene un *enfoque cuantitativo*, el cual, según Hernandez Sampieri, Fernández Collado, & Baptista Lucio (2010):

> Parte de una idea, que va acotándose y, una vez delimitada, se derivan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica. De las preguntas se establecen hipótesis y determinan variables; se desarrolla un plan para probarlas (diseño); se miden las variables en un determinado contexto; se analizan las mediciones obtenidas (con frecuencia utilizando métodos estadísticos), y se establece una serie de conclusiones respecto de la(s) hipótesis (p. 4).

La hipótesis planteada en esta investigación se validó en el capítulo 6 de este documento, al validad el *framework* propuesto.

3.2 UNIVERSO Y MUESTRA

En esta investigación, el universo de estudio fue varios grupos de productos resultado de investigación como artículos, libros y capítulos de libros con los que se trabajó en diferentes actividades. Estos resultados de investigación fueron el conocimiento ya existente que se tomó para dar respuesta a un problema identificado en la sección 1.1, de acuerdo a la definición de lo que es una investigación aplicada.

En esta investigación no se trabajó con muestras, ya que no se aplico el principio de aleatoriedad para elegir los artículos a analizar. Para seleccionar de estos

universos de estudio se usó tecnicas propias de una revisión bibliográfica como el estado del arte y la revisión sistemática de la literatura. Los universos de estudio y el segmento de análisis se describen en la Tabla 6.

Tabla 6. Universo y segmento de análisis

Actividad	Universo	Segmento de análisis
Realización del estado del arte sobre frameworks y modelos de seguridad para aplicaciones IoT.	59 artículos	21 artículos
Revisión sistemática de la literatura	806 artículos	40 artículos

Fuente: Autor.

3.3 TÉCNICAS E INSTRUMENTOS

En esta sección se presentan las técnicas e instrumentos para la recolección de información usadas en esta investigación.

3.3.1 Técnicas. Las técnicas utilizadas para la recolección de información fueron dos revisiones bibliográficas. De acuerdo a la tipología de revisiones bibliográficas realizada por Grant & Booth (2009), en este trabajo de investigación se usó los siguientes tipos de revisiones:

- ***Estado del arte***. El estado es una metodología de investigación cualitativo-documental (Gómez Vargas, Galeano Higuita, & Jaramillo Muñoz, 2015) y una herramienta útil para revisar la situación actual de un objeto de estudio, hacer un inventario de la misma y crear nuevos escenarios de formación e investigación en el respectivo campo de interés (Patiño, 2016). El estado del arte tiene como finalidad reconocer y obtener conocimiento de puntos de partida, visualizando el comienzo y lo que se está desarrollando, para vislumbrar dónde se quiere llegar (Gómez Vargas et al., 2015).

- ***Revisión Sistemática de la literatura – RSL***. Una RSL es una forma de estudio secundario que utiliza una metodología bien definida para identificar, analizar e interpretar toda la evidencia disponible relacionada con una pregunta de investigación, área temática o fenómeno de interés de una manera que sea imparcial y, hasta cierto punto, repetible (Biolchini, Gomes Mian, Cruz Natali, & Horta Travassos, 2005; Kitchenham & Charters, 2007). Una RSL exige un método riguroso y explícito para la identificación, evaluación crítica y síntesis de la evidencia obtenida (Beltrán G., 2005). Los estudios individuales que contribuyen a una RSL se llaman estudios primarios.

3.3.2 Instrumentos. Los instrumentos utilizados para la recolección de información fueron:

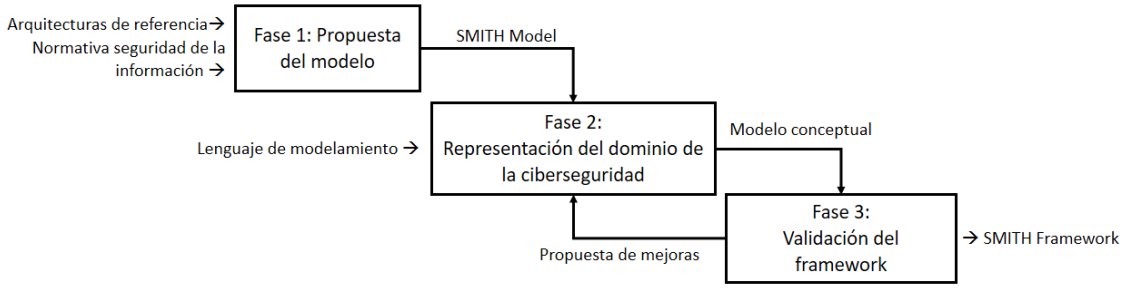
- ***Matriz bibliográfica***. Es un instrumento en el cual se hizo un inventario de los textos que conforman el universo y sobre el cual se aplicaron los filtros de selección. Una vez aplicado los filtros resultaron los textos de muestra. Este instrumento fue usando en el estado del arte y en la revisión sistemática de la literatura.

- ***Matriz analítica de contenido***. Este instrumento se usó para la extracción de información. En esta matriz relacionaron los textos de la muestra, escritos de forma vertical, con las categorías de análisis, escritas en horizontal. Este instrumento fue usando en el estado del arte y en la revisión sistemática de la literatura.

3.4 ACTIVIDADES REALIZADAS

Para el desarrollo de esta investigación se propuso una metodología compuesta por tres fases, la cual están directamente relacionadas con los objetivos específicos del proyecto, como se muestras en la Figura 6.

Figura 6. Metodología del proyecto de investigación.



Fuente: Autor.

En la Tabla 7 se especifican las actividades realizadas en el proceso de investigación y los resultados generados, las cuales están organizadas en las fases mencionadas anteriormente y sus subfases.

Tabla 7. Actividades realizadas en el proceso investigativo

Fase	Sub-fase	Actividad	Resultado
Formulación del modelo de gestión de ciberseguridad para aplicaciones IoT	Selección de arquitecturas de referencia (AR) de aplicaciones IoT que se analizaron	Revisión sistemática de la literatura (RSL) sobre arquitecturas de referencia para IoT	Listado de AR encontradas en la literatura
		Evaluación de arquitecturas de referencia para IoT	Listado de AR que se analizaron
	Identificación de niveles arquitecturales de una aplicación IoT genérica	Identificación de componentes y funcionalidades en común de las AR seleccionadas	Listados de funcionalidades y componentes de AR
		Selección de componentes y funcionalidades en común de las arquitecturas de referencia escogidas	Listado de componentes genéricos para IoT
		Especificación de una arquitectura general para aplicaciones IoT con los componentes seleccionados en cada capa	Arquitectura genérica para IoT propuesta
	Análisis de los requisitos de ciberseguridad que debe cumplir una aplicación IoT.	Revisión y selección de normatividad, metodologías, guías y buenas prácticas en ciberseguridad pertinentes para aplicaciones IoT	Fuentes para la extracción requisitos de seguridad
		Levantamiento de requisitos de seguridad de una aplicación IoT	Documento de requisitos de seguridad para aplicaciones IoT
	Construcción del modelo de gestión para la ciberseguridad para aplicaciones IoT.	Modelado conceptual general, utilizando la técnica seleccionada, de la estructura definida.	*SMITH Model*
		Modelado conceptual detallado que describe cada uno de los elementos del modelo general.	Descripción del *SMITH Model*
		Elaboración de una guía de uso del modelo construido.	Guía de buenas prácticas ciberseguridad para el aseguramiento de aplicaciones IoT
		Diseño de instrumento para verificación de cumplimiento los requisitos de seguridad de las aplicaciones de IoT, con base en el modelo construido	Instrumentos de evaluación
Representación del dominio de la ciberseguridad para aplicaciones IoT	Selección de lenguaje y herramientas de modelado	Selección del de modelamiento	Lenguaje de modelamiento unificado (UML)
		Selección de la herramienta de modelado.	Astah Professional
	Modelamiento del dominio de la ciberseguridad	Modelamiento del dominio de la ciberseguridad	Modelo IoT-CyDM
Validación del *framework*	Diseño de la técnica de validación del *framework*	Selección del método para validar el *framework* propuesto.	Método ATAM
		Creación los instrumentos para realizar la validación basado en el método seleccionado.	Definición del caso de uso
	Evaluación del *framework*	Aplicación de la técnica de validación diseñada.	Validación del *framework*.
		Análisis de los resultados.	
	Mejoramiento del *framework*	Creación del plan de mejoras del *framework* propuesto en caso de ser requerido.	

Fuente: Autor.

A continuación, se describen con más detalle las actividades realizadas:

3.4.1 Fase 1: Formulación del modelo de gestión de ciberseguridad para aplicaciones IoT. En esta fase se propuso un modelo para la gestión de la ciberseguridad en aplicaciones IoT, llamado *SMITH Model*, el cual se describe en el capítulo 4. Esta fase se nutre de las arquitecturas de referencia y de

normas, metodologías, directrices y mejores prácticas para la seguridad de la información.

Las arquitecturas de referencia permitieron identificar los componentes y capacidades genéricas de una aplicación IoT, y así, proponer la arquitectura genérica utilizada en el *SMITH Model*. Al evaluar las normas, metodologías, directrices y mejores prácticas para la seguridad de la información, se extrajeron requisitos de seguridad para aplicaciones IoT.

El resultado de esta fase es un marco conceptual que pretende ser una guía desde dos perspectivas: (i) Ser una guía para los desarrolladores en la construcción de las aplicaciones de IoT, en la que ellos puedan obtener información acerca de qué elementos de seguridad son necesarios para asegurar sus desarrollos en cada capa que componen una aplicación de IoT; (ii) para el análisis de elementos de ciberseguridad en las aplicaciones IoT: esta guía cuenta con unos instrumentos de ser una herramienta que busca identificar los elementos de seguridad que faltan en los desarrollos de IoT, y que podrían insertarse posteriormente.

Esta fase está compuesta por las siguientes sub-fases:

3.4.1.1 Selección de arquitecturas de referencia (AR) de aplicaciones IoT que serán analizadas. Esta subfase está compuesta por las siguientes actividades:

- *Revisión sistemática de la literatura (RSL) sobre arquitecturas de referencia para IoT*.

Por el alto componente metodológico de las RSL, en la sección 4.2.1 se describe las tres etapas de este proceso: planificación, conducción y reporte; las cuales abarcan la parte metodológica y los resultados los obtenidos.

- *Evaluación de arquitecturas de referencia (AR) para IoT*.

Para realizar esta evaluación se partió de las 40 arquitecturas de referencia obtenidas en la RSL, las cuales se muestran en la Tabla 18. El primer paso que se realizó fue filtrar las AR encontradas, eliminando aquellas que fueran orientadas a áreas específicas de IoT o desarrolladas para orientar requisitos de calidad como la seguridad o la resiliencia.

Luego de aplicar del proceso de depuración quedaron seis arquitecturas de referencia, como se muestran en la Tabla 8.

Tabla 8. Arquitecturas de referencia propuestas para IoT

Nombre	Autor(es)	Versión	Año	Área de aplicación	Fuente
Internet of Things – The inclusive model	CASAGRAS Project	--	2009	Genérico	CASAGRAS Project (2009)
Modelo de referencia de IoT	ITU-T	1.0	2012	Genérico	ITU-T (2012)
Internet of Things Architecture – IoT-A	IoT-A Project	3.0	2013	Genérico	Bauer, Boussard, Bui, De Loof, et al (2013)
Third Cycle Architecture de SmartSantander	SmartSantander Project	1.3	2013	Ciudad inteligente	Alex Gluhak et al. (2013)
Internet of Things Reference Model	Cisco	--	2014	Genérico	CISCO (2014)
Reference Architecture for Internet of Things	WSO2	0.9.0	2015	Genérico	WSO2 (2015)

Fuente: Autor.

Estas seis arquitecturas de referencia se evaluaron para determinar su completitud, de acuerdo a los elementos descritos en el RAModel. El *Reference Architecture Model* – RAModel, propuesto por Nakagawa, Oquendo, & Becker (2012), pretende presentar todos los elementos que podrían estar en una arquitectura de referencia, organizados por tipos y relaciones. Según los autores, el RAModel tiene cuatro formas de ser aplicado: (i) análisis de arquitecturas de referencia, para identificar elementos faltantes; (ii) análisis comparativo de arquitecturas de referencia, para seleccionar una entre un conjunto de ellas; (iii) como base para el establecimiento de nuevas arquitecturas de referencia; (iv) como apoyo al diseño de SPL (*Software Product Line*). El RAModel define 17 elementos, clasificados en cuatro grupos, tal como se muestra en la Tabla 9.

Tabla 9. Grupos y elementos del RAModel

Grupos y elementos del RAModel		Arquitecturas de referencia			
		AR1	AR2	…	ARn
Elementos del grupo de dominio	Legislación, estándar y regulación				
	Atributos de calidad				
	Cumplimiento del sistema				
Elementos del grupo de aplicación	Alcance				
	Requisitos funcionales				
	Datos del dominio				
	Restricciones				
	Riesgos				
	Objetivos y necesidades				
	Limitaciones				
	Elementos de software				

Elementos del grupo de infraestructura	Elementos de hardware				
	Mejores prácticas y lineamientos				
	Estilos arquitecturales				
Elementos del grupo de elementos transversales	Decisión				
	Terminología del dominio				
	Comunicación				

Fuente: Nakagawa *et al*. (2012).

Una vez se recolectó y analizó la documentación disponible de las arquitecturas de referencia, se procedió a identificar cuáles de los 17 elementos descritos por el RAModel se cumplían total o parcialmente. Para realizar la valoración se calificó con 1.0 para el cumplimiento total y 0.5 puntos al cumplimiento parcial, como se muestra en la Tabla 10. La evaluación completa se puede visualizar en el Anexo A – Evaluación de arquitecturas de referencia.

Tabla 10. Evaluación de las arquitecturas usando el RAModel

ID	Arquitectura de referencia	Cumplimiento de elementos del RAModel
AR1	Internet of Things – The inclusive model de CASAGRAS Project	8.0
AR2	Modelo de referencia de IoT de la ITU-T	9.0
AR3	Internet of Things Architecture de la IoT-A Project	16.0
AR4	Third Cycle Architecture de SmartSantander Project	9.0
AR5	Internet of Things Reference Model de Cisco	4.5
AR6	Reference Architecture for Internet of Things de WSO2	9.5

Fuente: Autor

Para elegir las arquitecturas de referencias que se estudiaron en este trabajo, se decidió considerar que aquellas que tuvieran la mitad más uno del total de elementos considerados en el RAModel. Para esta investigación, solo se consideró como arquitectura de referencia a aquellas arquitecturas que obtuvieron un puntaje igual o mayor a 9/17.

3.4.1.2 Identificación de los niveles arquitecturales de una aplicación IoT genérica. En esta subfase se estudiaron las arquitecturas de referencia – AR seleccionadas para conocer qué componentes y funcionalidades poseen, y partir de allí proponer una arquitectura genérica para aplicaciones IoT. Esta subfase está compuesta por las siguientes actividades:

- *Identificación de las componentes y funcionalidades en común de las AR seleccionadas*.

Luego de analizar la documentación de las AR seleccionadas, se identificó los componentes y funcionalidades que cada arquitectura considera.

- *Selección de componentes y funcionalidades en común de las arquitecturas de referencia escogidas*.

Una vez se identificó los componentes y funcionalidades se extrajeron las funcionalidades que tenían un común las AR estudiadas. Estas funcionalidades genéricas son las que se deben atender toda aplicación IoT para su funcionamiento.

- *Especificación de una arquitectura general para aplicaciones IoT con los componentes seleccionados en cada capa*.

Con las funcionalidades genéricas identificadas se propuso una arquitectura genérica para aplicaciones IoT, con el propósito para que el *SMITH Model* puede ser usado independientemente de lo que AR o arquitectura particular que se usa para la construcción de la aplicación IoT.

3.4.1.3 Análisis de los requisitos de ciberseguridad que debe cumplir una aplicación IoT. Esta subfase está compuesta por las siguientes actividades:

- *Revisión y selección de normatividad, metodologías, guías y buenas prácticas en ciberseguridad pertinentes para aplicaciones IoT e identificación de requisitos*.

Se identificó dos clases de fuentes del cual se podrían extraer los requisitos de seguridad para una aplicación IoT: el primero, los requisitos que consideraron las arquitecturas de referencia para proponer dicha arquitectura; el segundo, las normas, metodologías, directrices y buenas prácticas para implementar la seguridad de la información y/o ciberseguridad en sistemas.

En cuanto arquitecturas de referencia, se consideró la documentación de las cuatro arquitecturas de referencia que se estudiaron; en cuanto a las buenas prácticas se revisó el *OWASP IoT Project*, como se muestra en la Tabla 11.

Tabla 11. Fuentes para la extracción requisitos de seguridad

Fuente	Nombre del documento	Fuente
Modelo de referencia de la ITU-T	*Common requirements of the Internet of things*	ITU-T (2014b)
	Common requirements for Internet of things (IoT) applications	ITU-T (2014a)
Arquitectura de referencia del IoT Project	*Requirements — IoT-A: Internet of Things Architecture*	IoT-A Project (2016)
	Project Deliverable D4.2 - Concepts and Solutions for Privacy and Security in the Resolution Infrastructure	Serbanati *et al*. (2012)

Fuente	Nombre del documento	Fuente
Arquitectura del SmartSantander	*First Cycle Architecture Specification*	Muñoz *et al*. (2011)
	Second Cycle Architecture Specification	Jiménez *et al*. (Jiménez et al., 2012)
	Third Cycle Architecture Specification	Alex Gluhak *et al*. (2013)
Arquitectura de referencia de WSO2	*A Reference Architecture for the Internet of Things*	WSO2 (2015)
OWASP IoT Project	*Principles of IoT Security*	OWASP (2016b)

Fuente: Autor.

- *Levantamiento de requisitos de seguridad de una aplicación IoT*.

En esta actividad extrajeron los requisitos de seguridad para una aplicación IoT de las fuentes descritas en la Tabla 11. Estos requisitos se organizaron en cuatro grupos, como se muestra en la sección 4.4.

3.4.1.4 Construcción del modelo de gestión para la ciberseguridad para aplicaciones IoT. Esta subfase está compuesta por las siguientes actividades:

- *Modelado conceptual del dominio de ciberseguridad*

Para modelar el dominio de ciberseguridad para IoT, el primer paso fue entender qué elementos intervienen en dicho dominio y cómo estos se relacionan entre sí. De esta forma, ver de qué forma podría modelarse el dominio y de qué forma organizarse dichos elementos en el modelo.

Se identificaron tres grupos de elementos que intervienen: (i) la arquitectura IoT, (ii) las propiedades de la seguridad de la información y (iii) los elementos de ciberseguridad. La mejor forma de organizar estos grupos de elementos es a través de un cubo, donde cada represente los elementos de dicho grupo.

- *Elaboración de una guía de uso del modelo construido.*

Como complemento al *SMITH Model* se diseñó una guía con buenas prácticas de ciberseguridad que deberían considerar los desarrolladores en la etapa de diseño de una aplicación IoT. Las buenas prácticas de ciberseguridad se organizaron de acuerdo con cada una de las capas de la cara frontal del *SMITH Model*. El resultado de esta actividad fue la *Guía de buenas prácticas para el aseguramiento de aplicaciones IoT*.

- *Diseño de instrumento para verificación de cumplimiento los requisitos de seguridad de las aplicaciones de IoT, con base en el modelo construido*.

Para diseñar el instrumento de verificación, el cual es una lista de chequeo, se tomó cada una de las recomendaciones de seguridad descritas en la *Guía de buenas prácticas para el aseguramiento de aplicaciones IoT* y se identificaron las medidas que deben tenerse en cuenta para atender dichas recomendaciones de seguridad. Estas medidas se describieron en forma de pregunta, al cual se puede responder si se cumple o no dichas medidas, y así determinar qué elementos de ciberseguridad deben implementarse la aplicación IoT que se evalúa. Para comodidad en el uso de dicho instrumento, este se dividió en tres listas de chequeo, uno por cada una de las capas de la arquitectura genérica propuesta en este trabajo.

3.4.2 Fase 2: Representación del dominio de la seguridad para IoT. En esta fase se modeló el dominio de la ciberseguridad para Internet de las cosas usando el lenguaje unificado de modelado - UML. Este modelo se basó en el *SMITH Model* propuesto en la Fase 1, y está formado por seis componentes de ciberseguridad que se integran con los elementos del modelo de dominio IoT de la *IoT-A Project*. Esta fase está compuesta por las siguientes sub-fases:

3.4.2.1 Selección de lenguaje y herramientas de modelado. Esta subfase está compuesta por las siguientes actividades:

- *Selección del lenguaje de modelamiento*.

Como lenguaje para el modelamiento del domino de la ciberseguridad para aplicaciones IoT se seleccionó el lenguaje de modelamiento unificado, UML, porque es ampliamente usado en la ingeniería del software para el diseño de productos de software. Asimismo, UML es usando por arquitecturas de referencia como la IoT-A, una de las más conocidas en IoT.

- *Selección de la herramienta de modelado*.

Como herramienta para el modelado se eligió *Astah professional*, una herramienta robusta para la creación de diagramas UML, y que, además, ofrece una licencia de la versión *professional* de su producto para estudiantes.

3.4.2.2 Modelamiento del dominio de ciberseguridad para IoT. Para modelar el dominio de la ciberseguridad para aplicaciones IoT, como primer paso, se definió los conceptos claves del dominio y sus relaciones. En esa subfase se representó, a través del lenguaje UML, el dominio de ciberseguridad descrito en el *SMITH Model*. Los conceptos claves del dominio IoT y del dominio de ciberseguridad se extrajeron de este modelo.

Una vez se identificó los conceptos claves del dominio IoT se procedió a realizar el modelamiento de dicho dominio usando un diagrama de clases UML. Seguidamente se identificó los conceptos claves del domino de ciberseguridad en IoT y se propuso unos componentes de ciberseguridad abordaran dichos conceptos. Estos componentes se correlacionaron con los requisitos de seguridad descritos en el *SMITH Model* para verificar que estos requisitos atendían a dichos requisitos.

Una vez definidos los componentes que representan el dominio de ciberseguridad para aplicaciones IoT, se procedió a modelar dicho dominio, basándose en el modelo del dominio IoT propuesto anteriormente.

3.4.3 Fase 3: Validación del *framework* propuesto. En esta fase se validó el *framework* propuesto. Esta fase está compuesta por las siguientes sub-fases:

3.4.3.1 Diseño de la técnica de validación del *framework*. Esta subfase está compuesta por las siguientes actividades:

- *Selección del método para validar el framework propuesto*

Para seleccionar el método para validar el *framework* se hizo una búsqueda de métodos para la evaluación de arquitecturas. En la búsqueda se encontró que hay dos grupos de métodos de evaluación: los métodos de evaluación temprana y los métodos de evaluación tardía[12]. Se eligió los métodos de evaluación temprana porque su evaluación va acorde a la hipótesis y objetivo de esta investigación, que es la implementación de la ciberseguridad en la fase de diseño de una aplicación IoT. Para la validación del *framework* se eligió un método de evaluación temprana llamado ATAM, el cual se basa en escenarios.

- *Creación los instrumentos para realizar la validación basado en el método seleccionado*

En esta actividad se diseñó un caso de estudio al cual se le aplicó el instrumento de validación. Este instrumento de validación se basó en procedimiento usado por ATAM para la evaluación de arquitecturas. Este procedimiento consta de nueve pasos agrupados en cuatro fases: (i) presentación, (ii) investigación y análisis, (iii) pruebas y (iv) presentación de informe.

[12] Para más información sobre los métodos de evaluación de arquitecturas vea la sección 2.2.1.4 de este documento.

3.4.3.2 Evaluación del *framework*. En esta fase de aplicó el instrumento de validación creado a la arquitectura propuesta en el caso de estudio. Una vez terminado la validación se analizó los resultados y se presentó un informe.

3.4.3.3 Plan de mejoramiento del *framework*. El plan de mejoramiento del *framework* no aplicó, ya que la validación realizada confirmó la hipótesis planteada en este trabajo de investigación. Un plan de mejoramiento podrá surgir al aplicar otras validaciones descritas como trabajo futuro en el capítulo 7.

4. MODELO PROPUESTO DE GESTIÓN DE LA CIBERSEGURIDAD EN APLIACIONES IOT

En este capítulo se propone un modelo para guiar la implementación de elementos de ciberseguridad en aplicaciones IoT en la fase de diseño de estas, llamado *Security Management in Internet of THings Mode*l – *SMITH Model*. A través de este capítulo se muestra los pasos seguidos para proponer dicho modelo, cuáles son sus fundamentos y a partir de qué elementos se alimentó.

En la sección 4.1, se presenta la metodología usada para el desarrollo del *SMITH Model*, compuesto de cuatro fases: (i) la propuesta de una arquitectura IoT genérica; (ii) el levantamiento de los requisitos de seguridad para aplicaciones IoT; (iii) la construcción del modelo de gestión de ciberseguridad; y (iv) la validación del modelo propuesto.

En la sección 4.2, se muestra el proceso realizado en la revisión sistemática de la literatura sobre arquitecturas de referencia – AR para IoT. Se partió de 806 documentos extraídos de la literatura, llegando a 40 estudios primarios analizados. De estos estudios primarios se obtuvieron 40 AR, las cuales se depuró para obtener las que son genéricas para IoT. De la lista preliminar de AR resultante, se evaluó su completitud usando el RaModel, un modelo para identificar elementos faltantes en arquitecturas de referencia (Nakagawa et al., 2012).

En la sección 4.3, se propone una arquitectura genérica para aplicaciones IoT basada en las arquitecturas de referencia seleccionadas en la sección 4.2. El primer paso fue identificar los componentes y funcionalidades generales de cada AR; luego se extrajo las funcionalidades en común entre las AR y se agrupó estas funcionalidades en una arquitectura genérica por capas. El resultado es una arquitectura de tres capas que reúne las funcionalidades que debería tener una aplicación IoT; esta arquitectura considera tres conceptos que están tomando fuerza en la actualidad dentro de IoT: *cloud computing*, *fow computing* y *dew computing*[13].

En la sección 4.4, se realizó un levantamiento de requisitos de seguridad para aplicaciones IoT. Estos requisitos se tomaron de la documentación ofrecida por las arquitecturas de referencia, ya que dichas propuestas han realizado un trabajo definiendo los requisitos funcionales y de calidad en que se basarán, incluyendo los de seguridad. Estos requisitos se organizaron en cuatro grupos relacionados con la confidencialidad, la integridad, la disponibilidad, el no repudio.

[13] En la subsección 2.2.4 se describen cada uno de estas tecnologías.

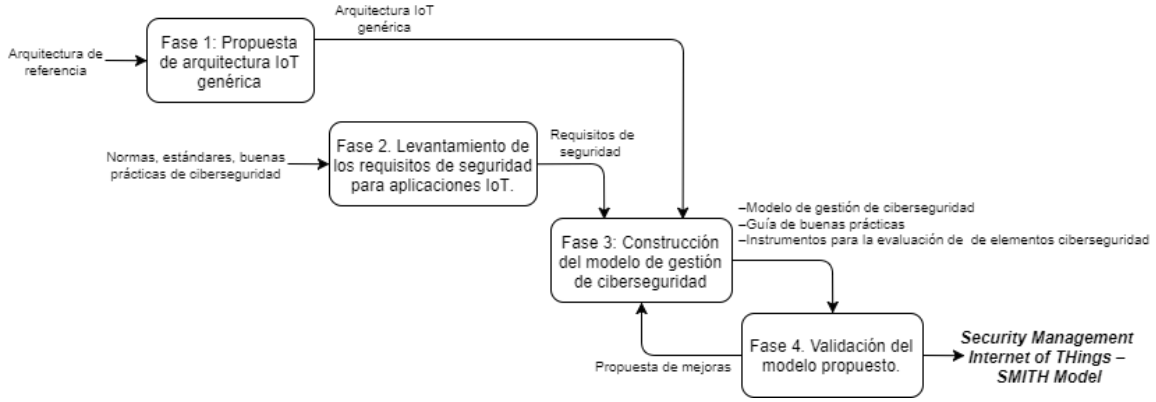
En la sección 4.5, se propone el *SMITH Model* el cual es una abstracción del dominio de la ciberseguridad en IoT y se representó a través de un cubo. Este modelo está compuesto por tres elementos: (i) la descripción del *SMITH Model*; (ii) la guía de buenas prácticas para el aseguramiento de aplicaciones IoT; (iii) los instrumentos de evaluación de elementos de ciberseguridad en aplicaciones IoT. En el Anexo B se presenta el modelo completo.

En la sección 4.6, se presentan las consideraciones finales de este capítulo.

4.1 METODOLOGÍA PARA EL DESARROLLO DE *SMITH MODEL*

Para el desarrollo del modelo de gestión de ciberseguridad se planteó una metodología de cuatro fases, como se muestran en la Figura 7.

Figura 7. Metodología de desarrollo del SMITH Model



Fuente: Autor.

Fase 1. Propuesta de arquitectura IoT genérica.

En esta fase se propuso una arquitectura IoT a partir del estudio de arquitecturas de referencia IoT propuestas. Para ello, se analizaron sus estilos arquitecturales identificando las funcionalidades y capacidades que consideraban. Una vez realizada este análisis se extrajeron las funcionalidades y capacidades comunes a todas las arquitecturas de referencia y se organizaron en un estilo arquitectural propuesto en este trabajo.

Fase 2. Levantamiento de los requisitos de seguridad para aplicaciones IoT.

En esta fase se revisó los documentos de las arquitecturas de referencia para IoT estudiadas y otros documentos de buenas prácticas para extraer los

requisitos de seguridad para una aplicación IoT. Estos requisitos se clasificaron en cuatro grupos:

- Grupo de requisitos para la confidencialidad de la información – GRC.
- Grupo de requisitos para la integridad de la información – GRI.
- Grupo de requisitos para la disponibilidad de la información – GRD.
- Grupo de requisitos para el no repudio – GRNP.

Fase 3. Construcción del modelo de gestión de ciberseguridad.

En esta fase realizó la abstracción del dominio de ciberseguridad para una aplicación IoT y se representó a través de un modelo en forma de cubo y se realizó la descripción de cada uno de sus vistas y sus respectivas capas. Asimismo, se propusieron los otros dos elementos que conforman este modelo: (i) la guía de buenas prácticas para el aseguramiento de aplicaciones IoT; y (ii) el instrumento para la evaluación de elementos de ciberseguridad en aplicaciones IoT.

Fase 4. Validación del modelo propuesto.

La validación del modelo propuesto se hará de tres formas:

1. *Validación técnica*. Se pueden considerar formas de validación como son mediante el uso de prototipos, validación de expertos en el área y pruebas en campo.
2. *Validación de usabilidad*. Este tipo de validación deja de un lado lo técnico y se concentra en cómo los usuarios interactúan con el modelo propuesto. El objetivo de esta validación es comprobar qué tan usable y amigable con los usuarios es la documentación del modelo, si es comprensible, si ofrece los suficientes recursos de ayuda, entre otras consideraciones.
3. La tercera forma de validación es representar, mediante algún lenguaje de modelamiento de software, el modelo propuesto. Una vez el modelo se representado a través del lenguaje seleccionado, se usa métodos diseñados en la ingeniería del software para su evaluación.

En este trabajo de investigación se usó la tercera forma de validación, ya que es la más pertinente para llevar a cabo dentro del tiempo de investigación disponible. Las otras dos formas de validación quedaron como trabajo futuro.

4.2 ARQUITECTURAS DE REFERENCIA PARA IOT

En esta sección se presenta el procedimiento de revisión sistemática de la literatura – RSL sobre arquitecturas de referencia – AR para IoT. En la

subsección 4.2.1 se describe el proceso realizado en la RSL a través de sus tres fases: planificación, conducción y reporte, y los resultados obtenidos.

En la subsección 4.2.2 se seleccionaron las AR que se analizaron para proponer la arquitectura genérica para IoT. Para esta selección se utilizó en RaModel (Nakagawa et al., 2012), un modelo de referencia para arquitecturas de referencia, con el cual se evaluó cuales AR cumplían con los elementos para considerarse realmente una arquitectura de referencia.

4.2.1 Revisión sistemática de la literatura. Se realizó una revisión de la literatura con el propósito de conocer las arquitecturas de referencia propuestas, para ello, se usó una metodología compuesta por tres fases: Planificación, conducción y reporte.

4.2.1.1 Planificación. En esta fase, se estableció un protocolo que fue usado como guía para conducir esta revisión de la literatura. En el protocolo de revisión se especificó siete pasos, a saber: (i) Objetivo de investigación; (ii) preguntas de investigación; (iii) términos de búsqueda; (iv) cadena de búsqueda; (v) fuentes de información; (vi) criterios de inclusión y exclusión; y (vii) extracción de datos.

(i) Objetivo de la investigación. El objetivo de esta revisión es identificar las arquitecturas de referencia propuestas para IoT.

(ii) Preguntas de investigación. En este paso, el objetivo de la investigación se traduce en preguntas de investigación. Con el objetivo de encontrar estudios primarios para entender y resumir evidencia sobre el dominio de seguridad de las arquitecturas de referencia para IoT, se propusieron las siguientes preguntas de investigación (PI).

- PI1: ¿Cuáles son las arquitecturas de referencia propuestas para IoT?
- PI2: ¿Cuáles son las arquitecturas de referencia para IoT más citadas?

(iii) Términos de búsqueda. Para la revisión se usó dos grupos de términos: en el grupo 1 se encuentran los términos relacionados con el IoT y en el Grupo 2 los términos relacionados con las arquitecturas de referencia. Los términos usados en la revisión de la literatura se muestran es la Tabla 12.

Tabla 12. Términos de búsqueda de revisión de la literatura

Grupo 1	Grupo 2
Internet of Things *IoT*	*Reference Architecture* *Reference Architectures* *Reference Architecture* *Reference Architectures*

Fuente: Autor.

(iv) Cadena de búsqueda. Con los grupos de términos identificados se definió una cadena para buscar los términos de búsqueda en los títulos, el resumen y las palabras claves de los artículos. La cadena de búsqueda la siguiente:

*TITLE-ABS-KEY (("*Internet of things*" OR IoT) AND ("*Reference Architectures*" OR "*Reference Architecture*" OR "*Reference Models*" OR "*Reference Model*"))*

(v) Fuentes de información. La búsqueda de información se realizó en las principales bases de datos digitales y en sitios web de las instituciones referentes en el área de las telecomunicaciones, como se muestra en la Tabla 13.

Tabla 13. Fuentes de información de la revisión de la literatura

Fuentes de información	
IEEE Xplore	ieeexplore.ieee.org/Xplore/home.jsp
ACM Digital Library	dl.acm.org
Scopus	www.scopus.com
ScienceDirect	www.sciencedirect.com/
SpringerLink	link.springer.com
Instituciones de referencia	ITU-T, Google Inc., Cisco, 7FP of the European Commission, IBM, Microsoft, Amazon

Fuente: Autor.

(vi) Criterios de inclusión y exclusión. El propósito de este paso es reducir el número de artículos recolectados en la fase de búsqueda, obteniendo artículos de calidad y relevantes para la investigación. Para definir la aceptación del artículo se definió los siguientes criterios de inclusión – CI:

- CI1: El artículo está escrito en inglés.
- CI2: El artículo se publicó desde el año 2005 al 2016.

- CI3: El artículo es de tipo artículo, libro, capítulo de libro o artículo de conferencia.
- CI4: El *Title-Keyword-Abstract* del artículo coincide con los criterios de búsqueda.
- CI5: El artículo define una arquitectura de referencia.

Los criterios de exclusión (CE) definidos para rechazar el artículo son los siguientes:

- CE1: ¿El artículo está disponible para la descarga?
- CE2: ¿El artículo describe una AR para IoT genérica?
- CE3: ¿La AR es de otros tipos de sistemas que no contiene funciones IoT?
- CE4: ¿Está el estudio primario relacionado con IoT, pero no propone o discute una RA?

(vii) Extracción de datos. Los datos que se extrajeron de casa uno de los artículos se muestra en Tabla 14.

Tabla 14. Extracción de datos de la revisión de la literatura

Datos recuperados	Descripción
Título	Título del artículo
Autores	Investigador(es) que escribieron el artículo
Año	Año en que se publicó el artículo
Arquitectura de referencia	Arquitectura de referencia mencionado en el artículo
Dominio de aplicación	Dominio de aplicación a la que está dirigido la arquitectura de referencia

Fuente: Autor.

4.2.1.2 Conducción. Se realizó una búsqueda manual utilizando la cadena de búsqueda en las bibliotecas digitales mencionadas en la Tabla 13. Los resultados de la búsqueda en cada una estas fuentes fueron compilados, organizados alfabéticamente en orden ascendente y se identificó los artículos repetidos, los cuales fueron descartados. Los resultados se muestran en la Tabla 15.

Tabla 15. Artículos recuperados en revisión de la literatura

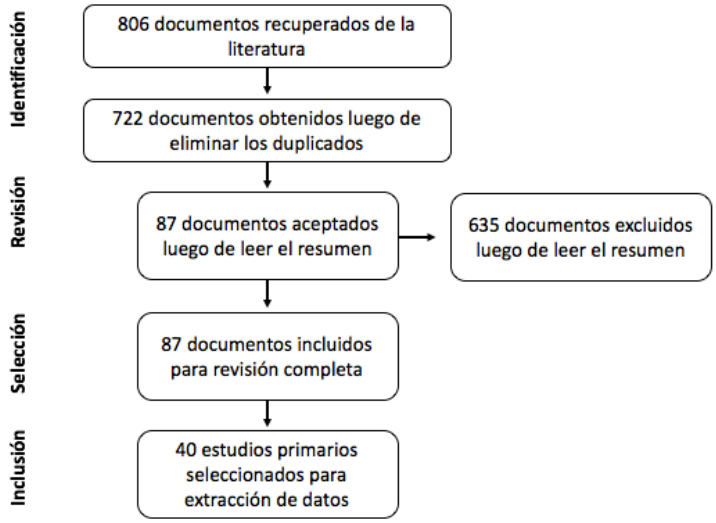
Librería digital	Artículos obtenidos	Artículos repetidos	Artículos totales
Scopus	125	5	120
ACM	2	1	1
IEEE Xplore	59	50	9
ScienceDirect	14	9	5
SpringerLink	606	19	587
Total	806	84	722

Fuente: Autor.

La información suministrada por las bases de datos sobre los *722 artículos obtenidos* se compiló y se procedió a eliminar la información que no era necesaria para la investigación. De cada artículo se obtuvo la siguiente información: un identificador de cada artículo, el título del artículo, autores, año de publicación, palabras claves, el editor, y el tipo de documento: artículo, artículo de conferencia, libro o capítulo de libro. Esta información se revisó por cada uno de los artículos y se aplicó un primer filtro, siguiendo los criterios de inclusión (CI) descritos anteriormente. Una vez aplicado el filtro 1, se rechazaron 635 artículos, quedando *87 artículos*. A estos 87 artículos se aplicó un segundo filtro teniendo en cuanta los criterios de exclusión (CE) mencionados previamente, del cual se obtuvieron *40 artículos primarios*.

En la Figura 8 se muestra una síntesis del proceso realizado en la conducción de la RSL. La información se organizó de acuerdo a la propuesta de Moher, Liberati, Tetzlaff y Altman (2009).

Figura 8. Flujo de información a través de las diferentes fases de la RSL



Fuente: Autor.

En la Tabla 16 se muestra los artículos primarios obtenidos. A cada artículo se le asignó un nuevo identificador (ID) para reseñarlos en este trabajo de investigación.

Tabla 16. Estudios primarios obtenidos de la revisión de la literatura

ID	Título	Año	Referencia
S1	*Architecting the Internet of Things: State of the Art*	2016	Abdmeziem, Tandjaoui, & Romdhani (2016)
S2	*A resilient Internet of Things architecture for smart cities*	2016	Abreu, Velasquez, Curado, & Monteiro (2017)
S3	*A reference architecture for improving security and privacy in internet of things applications*	2014	Addo, Ahamed, Yau, & Buduru (2014)
S4	*A Proposed Security Layer for the Internet of Things Communication Reference Model*	2015	Aldosari (2015)
S5	*Internet of things communication reference model*	2014	Alhamedi, Snasel, Aldosari, & Abraham (2014)
S6	*An IoT Protocol and Framework for OEMs to Make IoT-Enabled Devices Forward Compatible*	2015	Banda, Chaitanya, & Mohan (2015)
S7	*Enabling things to talk: Designing IoT solutions with the IoT architectural reference model*	2013	Bassi et al. (2013)
S8	*IoT reference model*	2013	Bauer, Bui, *et al*. (2013)
S9	*Privacy-Preserving Security Framework for a Social-Aware Internet of Things*	2014	Bernabe, Hernández, Moreno, & Gomez (2014)
S10	*A process for Generating Concrete Architectures*	2013	Boussard *et al*. (2013)
S11	*An analysis of Reference Architectures for the Internet of Things*	2015	Cavalcante, Alves, Batista, Delicato, & Pires (2015)
S12	*Test-enabled architecture for IoT service creation and provisioning*	2013	De, Carrez, Reetz, Tönjes, & Wang (2013)
S13	*IoT Lab: Towards co-design and IoT solution testing using the crowd*	2015	Fernandes *et al*. (2015)
S14	*Architecture Reference Model*	2014	Höller *et al*. (2014a)
S15	*The advantages of IoT and cloud applied to smart cities: ClouT user scenarios and reference architecture*	2015	Formisano et al. (2015)
S16	*IIoT Reference Architecture*	2016	Gilchrist (2016)
S17	*An Architectural Blueprint for a Real-World Internet*	2011	Alexander Gluhak *et al*. (2011)
S18	*Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things*	2013	Guo, Zhang, Wang, Yu, & Zhou (2013)
S19	*IoT Architecture – State of the Art*	2014	Höller *et al*. (2014b)
S20	*IoT Reference Architecture*	2014	Höller *et al*. (2014c)
S21	*Designing IoT architecture(s): A European perspective*	2014	Krco, Pokric, & Carrez (2014)
S22	*Internet of Things*	2013	G. M. Lee, Crespi, Choi, & Boussard (2013)
S23	*Data management for internet of things: Challenges, approaches and opportunities*	2013	Ma, Wang, & Chu (2013)
S24	*From the Internet of Things to the Internet of People*	2015	Miranda *et al*. (2015)
S25	*Social Networks and Internet of Things, an Overview of the SITAC Project*	2014	Monteiro, Oliveira, Bastos, Ramrekha, & Rodriguez (2014)
S26	*A Comprehensive Study of Security of Internet-of-Things*	2016	Nia & Jha (2016)
S27	*Architecture for Internet of Things Analytical Ecosystem*	2016	Ratkowski (2016)
S28	*An IoT based reference architecture for smart water management processes*	2015	Robles *et al*. (2015)

ID	Título	Año	Referencia
S29	*SmartSantander: The meeting point between Future Internet research and experimentation and the smart cities*	2011	L. Sanchez *et al.* (2014)
S30	*Internet of Things Service Systems Architecture*	2015	Schauer & Debita (2015)
S31	*HePA: Hexagonal platform architecture for smart home things*	2016	Seo, Kim, Yun, Huh, & Maeng (2015)
S32	*Standardizing the internet of things in an evolutionary way*	2014	Shen & Carugi (2014)
S33	*Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm*	2014	Shrouf, Ordieres, & Miragliotta (2014)
S34	*Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce*	2015	S. Singh & Singh (2015)
S35	*A Resource-Oriented Architecture for the Internet of Things (IoT)*	2016	Souza & Cardozo (2016)
S36	*IoT-A and FIWARE: Bridging the barriers between the cloud and IoT systems design and implementation*	2016	Stravoskoufos, Sotiriadis, & Petrakis (2016)
S37	*Reference model of Industrie 4.0 service architectures: Basic concepts and approach*	2015	Usländer & Epple (2015)
S38	*A reference architecture for IoT-based logistic information systems in agri-food supply chains*	2015	Verdouw, Robbemond, Verwaart, Wolfert, & Beulens (2015)
S39	*Reference Architectures for the Internet of Things*	2016	Weyrich & Ebert (2016)
S40	*Research on architecture of the internet of things for grain monitoring in storage*	2012	B. Xu, Zhang, & Yang (2012)

Fuente: Autor.

4.2.1.3 Reporte. En esta fase se leyó los 40 trabajos primarios obtenidos en la fase anterior y se procedió a extraer la información para responder a las preguntas en la Fase I:

PI1: ¿Cuáles son las arquitecturas de referencia propuestas para IoT?

Para responder esta pregunta, se buscó en cada uno de los estudios primarios las arquitecturas de referencia[14] que proponían, se basaban o mencionaban dichos trabajos. En la Tabla 17 se muestra las arquitecturas de referencia obtenidas en los estudios primarios.

Tabla 17 - Arquitecturas de referencia obtenidas de los estudios primarios

ID	Arquitectura de referencia propuesta, utilizada o mencionada
S1	*IEFT Protocol Suite*
	SENSEI Project
	BRIDGE
	CASAGRAS Project
	IDRA approach
	Server based approach

[14] En esta sección, el término 'Arquitectura de referencia' probablemente esté sujeto a la percepción del autor para referirse a una arquitectura IoT, y no propiamente que esta contribución cumpla con todos o la mayoría de los elementos podrían estar en una arquitectura de referencia. Para determinar la completitud de dichas arquitecturas, en la sección 4.2.1 se evaluarán usando el RAModel, determinándose cuales podrían considerarse realmente una arquitectura de referencia.

ID	Arquitectura de referencia propuesta, utilizada o mencionada
	EPC based approach
	Cloud approach
	Social Internet of Things (SIoT) architecture
	Virtual network approach
S2	Arquitectura de IoT resiliente para ciudades inteligentes
S3	Arquitectura de referencia para mejorar la seguridad y la privacidad
S4	*IoT-A Communication Reference Model*
S5	*IoT-A Communication Reference Model*
S6	*IoTivity Architecture*
	Industrial Internet Reference Architecture (IIRA)
S7	*IoT-A Project*
S8	*Architecture Reference Model (ARM) de IoT-A Project (IoT-ARM)*
S9	*Architecture Reference Model (ARM) de IoT-A Project (IoT-ARM)*
S10	*IoT-A Project*
S11	*Architecture Reference Model (ARM) de IoT-A Project (IoT-ARM)*
	Arquitectura de referencia de WSO2
S12	Basada en la arquitectura de IoT-A
S13	Proyecto basado en IoT-ARM del IoT-A Project
S14	IoT-A
S15	Arquitectura de referencia ClouT
S16	*IIoT Reference Architecture*
S17	Arquitectura de referencia para Real-World Internet (RWI)
S18	Framework conceptual para la IoT oportunista
S19	Modelo de referencia de la ITU-T
S20	IoT-A
S21	FI-WARE
	IoT-A
	IoT6
S22	Modelo de referencia de la ITU-T
	IoT-A
S23	Modelo de referencia en capas para la gestión de datos IoT
S24	*Internet of People Middleware*
S25	*SITAC Project*
S26	Modelo propuesto por Gubi *et al*. (2013)
	Modelo propuesto por Atzori *et al*. (2010)
	Modelo de CISCO (2014)
S27	Arquitectura para análisis de datos en IoT
S28	Arquitectura de referencia basada en IoT para procesos inteligentes de gestión del agua
S29	Modelo de referencia de SmartSantander
S30	IoT-A
S31	*Hexagonal Platform Architecture – HePA*
S32	IoT-A
S33	Arquitectura de referencia para fábrica inteligente basada en IoT
S34	Arquitectura de referencia IoT para *E-commerce*
S35	Arquitectura orientada a recursos para IoT
S36	*Architecture Reference Model* (ARM) de *IoT-A Project* (IoT-ARM)
S37	Modelo de RAMI 4.0
S38	Arquitectura de logística Agroalimentaria Inteligente
S39	*Reference Architecture Model Industrie* (RAMI 4.0)
	Industrial Internet Reference Architecture (IIRA)
	Internet of Things - Architecture (IoT-A)
	Arrowhead Framework
S40	Arquitectura IoT para el monitoreo de grano en almacenamiento

Fuente: Autor.

En la Tabla 18, se muestra las arquitecturas de referencias para IoT propuestas, mencionadas o en las que se basan algunos de los 40 estudios primarios estudiados.

Tabla 18 - Arquitecturas de referencia para IoT propuestos en la literatura

	Arquitectura de referencia		Fuente
1	Arquitectura de IoT resiliente para ciudades inteligentes		S237
2	Arquitectura de logística Agroalimentaria Inteligente		S38
3	Arquitectura de referencia basada en IoT para procesos inteligentes de gestión del agua		S28
4	Arquitectura de referencia ClouT		S15
5	Arquitectura de referencia de WSO2		S11
6	Arquitectura de referencia IoT para E-*commerce*		S34
7	Arquitectura de Referencia para fábrica inteligente basada en IoT		S33
8	Arquitectura de referencia para mejorar la seguridad y la privacidad		S3
9	Arquitectura de referencia para *Real-World Internet* (RWI)		S17
10	Arquitectura IoT para el monitoreo de grano en almacenamiento		S40
11	Arquitectura orientada a recursos para IoT		S35
12	Arquitectura para análisis de datos en IoT		S27
13	*Arrowhead Framework*		S39
14	BRIDGE		S1
15	*CASAGRAS Project*		S1
16	*Cloud approach*		S1
17	*EPC based approach*		S1
18	FI-WARE		S21
19	Framework conceptual para la IoT oportunista		S18
20	*Hexagonal Platform Architecture – HePA*		S31
21	*IDRA approach*		S1
22	*IEFT Protocol Suite*		S1
23	*IIoT Reference Architecture*		S16
24	*Industrial Internet Reference Architecture* (IIRA)		S6, S39
25	*Internet of People Middleware*		S24
26	*IoT-A Project*	*Internet of Things - Architecture (IoT-A)*	S7, S10, S12, S14, S20, S21, S22, S30, S32, S39
		Architecture Reference Model (IoT-ARM)	S8, S9, S11, S13, S36
		IoT-A Communication Reference Model	S4, S5
27	IoT6		S21
28	*IoTivity Architecture*		S6
29	Modelo de CISCO		S26
30	Modelo de referencia de la ITU-T		S19, S22
31	Modelo de referencia de SmartSantander		S29
32	Modelo de referencia en capas para la gestión de datos IoT		S23
33	Modelo propuesto por Atzori *et al*. (2010)		S26
34	Modelo propuesto por Gubi *et al*. (2013)		S26
35	*Reference Architecture Model Industrie* (RAMI 4.0)		S37, S39
36	*SENSEI Project*		S1
37	*Server based approach*		S1
38	*SITAC Project*		S25
39	*Social Internet of Things (SIoT) architecture*		S1
40	*Virtual network approach*		S1

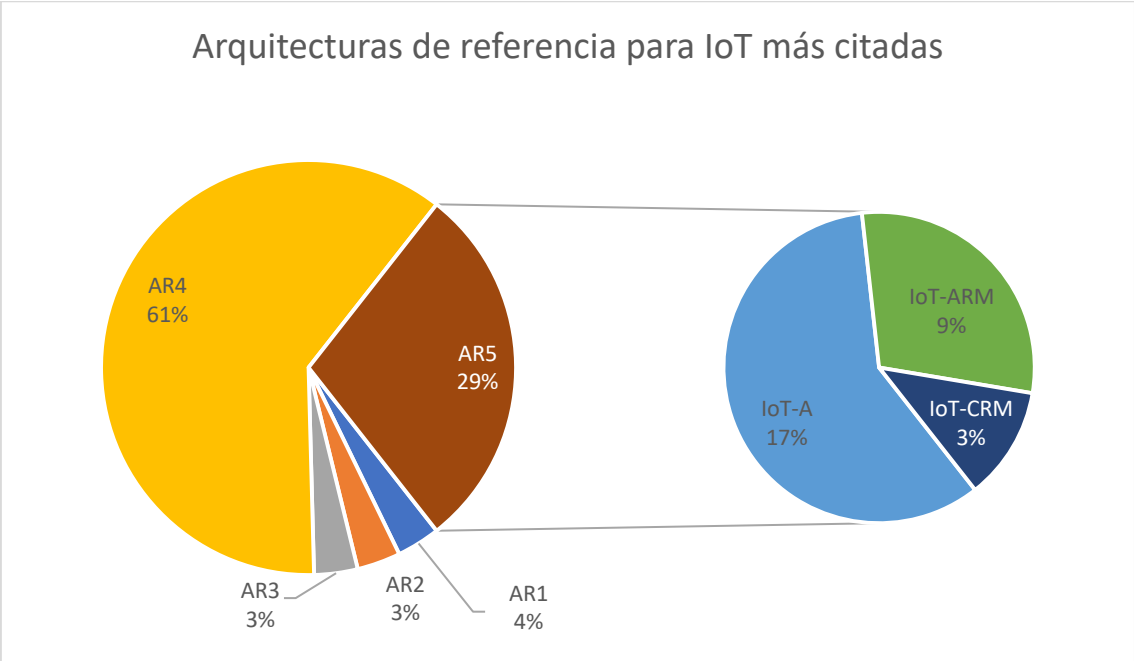
Fuente: Autor.

En la revisión de la literatura se encontraron 40 arquitecturas de referencia para IoT o alguna de sus áreas.

PI2: ¿Cuáles son las arquitecturas de referencia para IoT más citadas?

Para responder esta pregunta se basó en el número de estudios primarios que mencionan la arquitectura de referencia. En la Figura 9, se muestra la gráfica con la información tabulada sobre las arquitecturas de referencia para IoT encontradas en la literatura.

Figura 9 - Arquitecturas de referencia para IoT más citadas en la literatura



Fuente: Autor.

A continuación, se mencionan las leyendas usadas en la anterior figura:

- AR1: Modelo de referencia de la ITU-T
- AR2: *Industrial Internet Reference Architecture* (IIRA)
- AR3: *Reference Architecture Model Industrie* (RAMI 4.0)
- AR4: Otras arquitecturas de referencia propuestas en la literatura.
- AR5: IoT-A Project, cuyo segmento se divide en:
    - IoT-A: *Internet of Things – Architecture*
    - IoT-ARM: *Architecture Reference Model*
    - IoT-CRM: *Communication Reference Model*

De la Figura 9 se puede concluir que el IoT-A Project, y elementos que los componen como son: modelo y arquitectura de referencia, y el modelo de referencia de comunicaciones, son las contribuciones más representativas en la literatura. En conjunto, son mencionadas en 17 de los 40 trabajos primarios estudiados. Seguidamente se encuentran el modelo de referencia de la ITU-T, el *Industrial Internet Reference Architecture* (IIRA) y el *Reference Architecture Model Industrie (*RAMI 4.0) con dos menciones cada uno.

4.2.2 Arquitecturas de referencia seleccionadas. De acuerdo con el procedimiento descrito en la sección 3.4.1.1, las arquitecturas de referencia son las siguientes: (i) El modelo de referencia de la ITU-T, (ii) la arquitectura de referencia del *IoT-A Project*, (iii) la arquitectura de *SmartSantander Project* y (iv) la arquitectura para IoT de WSO2. De estas arquitecturas de referencia, la arquitectura más completa es la de la IoT-A Project, que obtuvo 16 de los 17 puntos, satisfaciendo 15 elementos completamente y dos elementos parcialmente. Las otras tres arquitecturas están muy parejas, con un puntaje entre 9 y 9.5.

4.3 ARQUITECTURA GENÉRICA PROPUESTA PARA APLICACIONES IOT

En esta sección se define una arquitectura genérica para aplicaciones IoT, con el propósito que el modelo de ciberseguridad propuesto en este capítulo sea interoperable con otras arquitecturas concretas de aplicaciones IoT, y no se diseñe basándose en una arquitectura específica y teniendo en cuenta solo sus elementos y características.

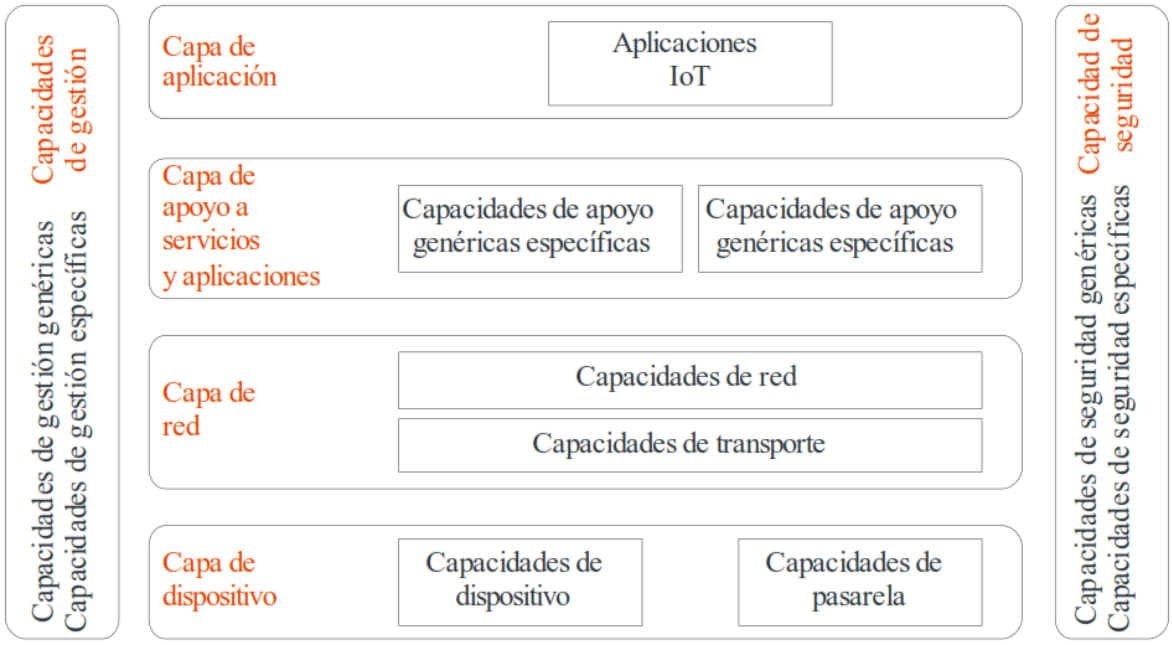
Para proponer esta arquitectura genérica se basó en arquitecturas de referencia para IoT propuestos en la literatura. Para ello, en la subsección 4.3.1 se seleccionaron las arquitecturas de referencia que serían analizados para proponer la arquitectura de referencia. Una vez se seleccionaron las arquitecturas de referencia, en la subsección 4.3.2 se identificaron sus capas y los elementos que las componen, y en la subsección 4.3.3 se seleccionaron qué elementos iban a formar la arquitectura genérica. Finalmente, en la subsección 4.3.4 se especifica la arquitectura genérica que se usará para construir el modelo de ciberseguridad.

4.3.1 Capas y componentes identificadas. Las cuatro arquitecturas de referencia seleccionadas en la subsección anterior se analizaron y se describió su estilo arquitectural para identificar sus componentes internos.

4.3.1.1 Análisis del modelo de referencia de la ITU-T. Este modelo fue propuesto por la Unión Internacional de Telecomunicaciones, ITU-T, por sus siglas en inglés, en 2012 en su recomendación Y.2060. Esta recomendación presenta los términos generales de IoT desde la perspectiva de la ITU-T, aclarando qué es IoT y sus actividades relativas (ITU-T, 2012).

Este modelo de referencia está compuesto por cuatro capas y dos capas trasversales que son las capacidades de gestión y de seguridad, como se muestra en la Figura 10.

Figura 10. Modelo de referencia de IoT de la ITU-T



Fuente: ITU-T (2012)

En la Tabla 19 se describen las capas, capacidades y funcionalidades del modelo de referencia.

Tabla 19. Capas y componentes del modelo de referencia de ITU-T

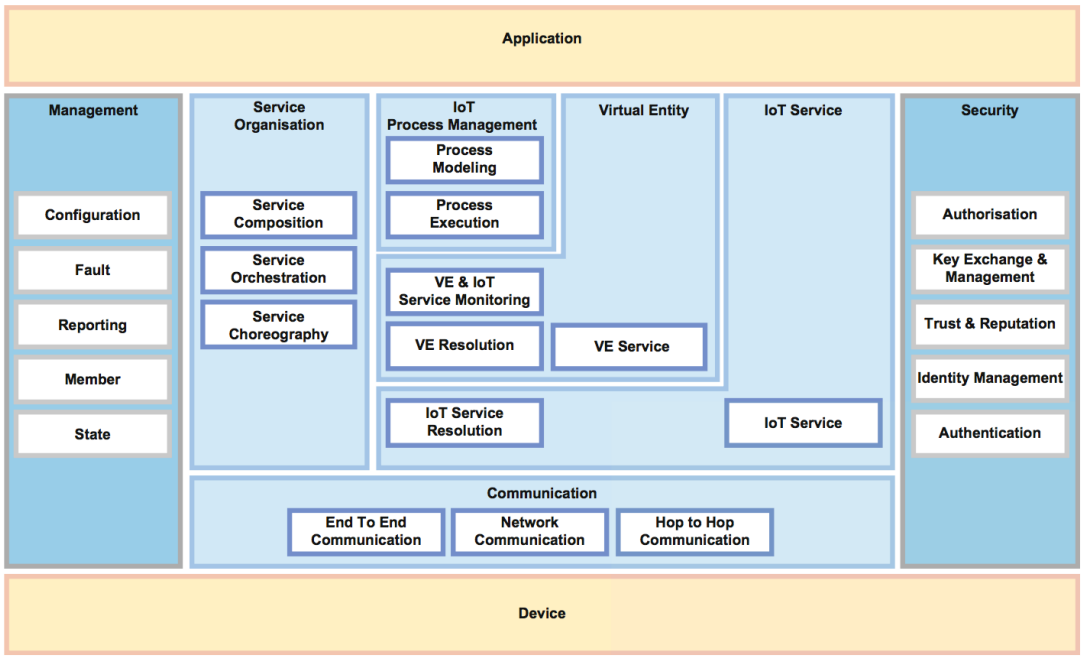
Capa	Capacidades	Descripción
Capa de aplicación		Contiene las aplicaciones IoT
Capa de soporte de servicios y aplicaciones	Capacidades de soporte genéricas	Pueden contener elementos como: • Procesamiento. • Almacenamiento. • Otras capacidades comunes para ser utilizadas por cualquier aplicación IoT.
	Capacidades de soporte específicas	Capacidades para atender necesidades particulares de una aplicación IoT

Capa	Capacidades	Descripción
Capa de red	Capacidades de red	Ofrece funciones de control de la conectividad en red, tales como: • Funciones de control de acceso y de recursos de transporte • Gestión de la movilidad • Autentificación, autorización y contabilidad (AAA).
	Capacidades de transporte	Suministra conectividad para el transporte de información y datos de servicios y aplicaciones IoT, así como la información de control y gestión.
Capa de dispositivo	Capacidades del dispositivo	• Interacción con la red de comunicaciones • Redes ah-hoc • Modo reposo y activo
	Capacidades de pasarela	• Soporte de interfaces múltiples para soportar dispositivos conectados mediante diferentes tipos de tecnologías. • Conversión de protocolo, para comunicaciones que usan diferentes protocolos, ya sean en la capa de dispositivo o entre diferentes capas.
Capas transversales	Capacidades de gestión	• Gestión de dispositivos, tales como: o Activación y desactivación de dispositivos remotos o Diagnostico o Actualización del firmware y/o del software • Gestión del estado de trabajo del dispositivo • Gestión de topología de red • Gestión del tráfico y la congestión
	Capacidades de seguridad	En la capa de aplicación: • Autentificación y autorización • Confidencialidad de datos de aplicación y protección de la integridad • Protección de la privacidad • Auditorias de seguridad • Antivirus En la capa de red: • Autentificación y autorización • Confidencialidad de datos de señalización y de datos de uso • Protección de la integridad de señalización En la capa de dispositivo: • Autentificación y autorización • Validación de la integridad del dispositivo • Control de acceso • Confidencialidad de datos • Protección de la integridad.

Fuente: Autor

4.3.1.2 Análisis de la arquitectura de referencia del IoT Project. El *IoT-A Project* es un proyecto financiado Unión Europea y está compuesto por varios elementos, entre ellos, el modelo de referencia. La arquitectura de referencia está compuesta por nueve grupos funcionales (FG, *Functionality Groups*), los cuales, a su vez, están compuestos por componentes funcionales (FC, *Functionality Components*), como se muestra en la Figura 11.

Figura 11. Arquitectura de referencia de IoT-A Project



Fuente: Bauer, Boussard, Bui, De Loof, *et al*. (2013)

De los nueve FG que contiene esta arquitectura, el FG de aplicación y el FG de dispositivo están fuera del alcance de la misma, las cuales se colorean en amarillo (Bauer, Boussard, Bui, De Loof, et al., 2013), como se muestra en la Tabla 20.

Tabla 20. Capas y componentes de la arquitectura de referencia de la IoT-A

Grupo funcional	Componente funcional	Descripción
Organización del servicio (*Service Organisation*) Es el FG central que actúa como un centro de comunicación entre otros FG	Servicio de Orquestación	Resuelve los servicios IoT que son adecuados para satisfacer las solicitudes de servicio procedentes del FC de ejecución de procesos o de los usuarios
	Composición del Servicio	Resuelve los servicios que se componen de los servicios IoT y otros servicios con el fin de crear servicios con una funcionalidad extendida.
		Tiene dos funciones principales: (1) soportar composiciones de servicio flexibles y (2) aumentar la calidad de la información.
	Servicio Coreografía	Ofrece un intermediario que maneja la comunicación de publicación/suscripción entre servicios.
Gestión de procesos IoT (*IoT Process Management*) Este FG proporciona los conceptos funcionales y las interfaces	Modelado de procesos	Proporciona un entorno para el modelado de procesos empresariales de IoT que serán serializados y ejecutados en el FC de Ejecución de Procesos.
		Proporciona las herramientas necesarias para modelar procesos usando la notación estandarizada.
	Ejecución de Procesos	Ejecuta los procesos que han sido modelados en el anterior FC, utilizando servicios de IoT que están orquestados en la capa de organización de servicio.
		Implementa modelos de procesos en los entornos de ejecución.
		Alinea los requisitos de la aplicación con las capacidades de servicio.

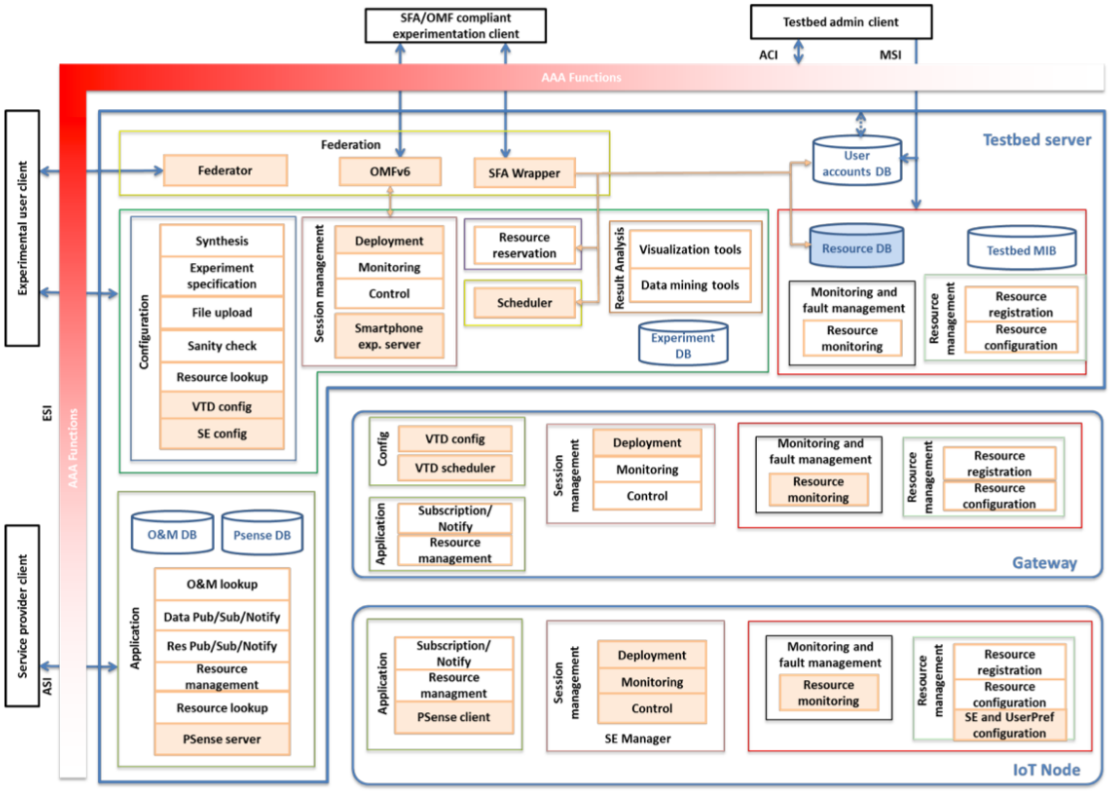
Grupo funcional	Componente funcional	Descripción
necesarias para la a integración de los sistemas de gestión de procesos tradicionales con el IoT-ARM		Una vez resuelve los servicios IoT, este FC puede invocar dichos servicios.
Entidad virtual (*Virtual Entity*) Contiene funciones para interactuar con el sistema IoT con base a entidades virtuales (EV), para descubrir y buscar servicios que puedan proporcionar información sobre VEs o que permitan la interacción con EVs, y para gestionar asociaciones, así como encontrar dinámicamente nuevas asociaciones y supervisar su validez	Resolución de EV	Proporciona las funcionalidades al usuario IoT para recuperar asociaciones entre EV y los servicios de IoT.
		Descubrimiento de asociaciones nuevas y en su mayoría dinámicas entre EV y servicios asociados, considerándose la ubicación, proximidad y otra información de contexto.
		Permite buscar servicios relacionados con EV.
		Permite gestionar asociaciones: insertar, borrar y actualizar asociaciones entre un EV y los servicios IoT que están asociados al EV
	Monitoreo del servicio EV e IoT	Es responsable de encontrar automáticamente nuevas asociaciones, las cuales se insertan en la FC resolución de EV. Se pueden derivar nuevas asociaciones basadas en asociaciones existentes, descripciones de Servicio e información sobre EV
	Servicio EV	Se encarga de los servicios de entidad, el cual representa un punto de acceso general a una entidad particular, ofreciendo medios para aprender y manipular el estado de la entidad a través de operaciones que permiten leer y/o actualizar los valores de los atributos de las entidades
Servicio IoT (*IoT Service*). Contiene servicios IoT, así como funcionalidades para el descubrimiento, la búsqueda y la resolución de nombres de los servicios IoT	Servicio IoT	Expone un recurso para hacerlo accesible a otras partes del sistema IoT.
		Principales funcione: (1) devolver información proporcionada por un recurso de forma síncrona, (2) aceptar información enviada a un recurso para almacenar la información o configurar el recurso o para controlar un dispositivo accionador y (3) suscribirse a información, es decir, devolver la información proporcionada por un recurso de forma asíncrona.
	Resolución de servicio IoT	Proporciona todas las funcionalidades necesarias para que el usuario encuentre y pueda ponerse en contacto con los servicios IoT.
Comunicación (*Communication*). Es una abstracción que modela la variedad de esquemas de interacción derivados de las muchas tecnologías que pertenecen a los sistemas IoT y proporciona una interfaz común al servicio FG de IoT.	Comunicación *hop-to-hop*	Proporciona la primera capa de abstracción de la tecnología de comunicación física del dispositivo para permitir el uso y la configuración de cualquier tecnología de capa de enlace diferente.
		Sus principales funciones son transmitir una trama desde el FC comunicación en red al FC comunicación *hop-to-hop* y desde un Dispositivo a la FC comunicación *hop-to-hop*.
		Los argumentos para la transmisión de trama se pueden establecer; ejemplos de argumentos incluyen: fiabilidad, integridad, cifrado y control de acceso.
		El FC de comunicación *hop-to-hop* también es responsable de enrutar un marco y gestionar la cola de tramas y establecer el tamaño y las prioridades de las colas de trama de entrada y salida.
	Comunicación en red	Se encarga de habilitar la comunicación entre redes a través de localizadores (direccionamiento) y resolución de ID.
		Incluye enrutamiento y convergencia a través de traducciones de protocolos de red.
		Trasmite un paquete desde los otros dos FC de este FG hacia el FC de comunicación en red.
		Los argumentos para la transmisión de paquetes pueden configurarse y ejemplos de argumentos incluyen: fiabilidad, integridad, cifrado, direccionamiento unicast/multicast y control de acceso.

Grupo funcional	Componente funcional	Descripción
		Entre otras funcionalidades.
	Comunicación *end-to-end*	Se encarga de la totalidad de la abstracción de comunicación de extremo a extremo, lo que significa que se encarga de la transferencia fiable, el transporte y las funcionalidades de traducción, apoyo proxies/pasarelas y de ajuste de parámetros de configuración cuando la comunicación atraviesa diferentes entornos de red.
		Los argumentos para el mensaje se pueden configurar y los ejemplos incluyen: fiabilidad, integridad, cifrado, control de acceso y multiplexación.
Seguridad (*Security*). Este FG es el responsable de garantizar la seguridad y la privacidad de los sistemas compatibles con IoT-A	Autorización	Esta FC es una interfaz para gestionar políticas y realizar decisiones de control de acceso basadas en políticas de control de acceso. Tiene dos funcionalidades por defecto: • Determinar si una acción está autorizada o no. • Administrar políticas, como agregar, actualizar o eliminar una directiva de acceso
	Gestión e intercambio de llaves	Este FC permite las comunicaciones seguras entre dos o más compañeros de IoT-A que no tienen conocimiento inicial entre sí o cuya interoperabilidad no está garantizada, asegurando la integridad y confidencialidad. A esta FC se le atribuyen dos funciones: • Distribuir las claves de forma segura. • Registrar las capacidades de seguridad.
	Confianza y Reputación	Este FC recopila las puntuaciones de reputación de los usuarios y calcula los niveles de confianza del servicio. Este FC tiene dos funciones predeterminadas: • Solicitar información de reputación. • Proporcionar información de reputación.
	Gestión de Identidad	Aborda las cuestiones de privacidad al emitir y administrar seudónimos e información accesoria a sujetos de confianza para que puedan operar (usar o proporcionar servicios) de forma anónima.
	Autenticación	Participa en la autenticación de usuarios y servicios. Las dos funcionalidades proporcionadas por este FC son: (1) para autenticar a un usuario basado en credenciales proporcionadas y (2) para verificar si una aseveración proporcionada por un usuario es válida o no válida.
Gestión (*Management*)	Configuración	Este FC es el responsable de inicializar la configuración del sistema, como recolectar y almacenar la configuración de FC y dispositivos; y también del seguimiento de los cambios de configuración y la planificación para la extensión futura del sistema. Sus funciones principales son: • Recuperar una configuración. • Configurar la configuración.
	Fallo	Su objetivo es identificar, aislar, corregir y registrar las fallas que ocurren en el sistema IoT.
	Informes	Uno de los muchos objetivos de informes concebibles es determinar la eficiencia del sistema actual.
	Miembro	Es responsable de la gestión de la membresía y la información asociada de cualquier entidad relevante (FG, FC, EV, *Service* IoT, dispositivo, aplicación, usuario) a un sistema IoT.
	Estado	Este FC monitorea y predice el estado del sistema IoT.

Fuente: Autor

4.3.1.3 Análisis de la arquitectura de SmartSantander. El diseño de esta arquitectura es cíclico, y en cada iteración se abordar nuevos requerimientos y especificaciones. Actualmente, esta arquitectura se encuentra en su tercer ciclo (Alex Gluhak et al., 2013), el cual se muestra en la Figura 12.

Figura 12. Arquitectura de referencia de SmartSantander Project



Fuente: Alex Gluhak *et al*. (2013)

En la *Tabla 21* se muestra los componentes y funcionalidades de la arquitectura de referencia.

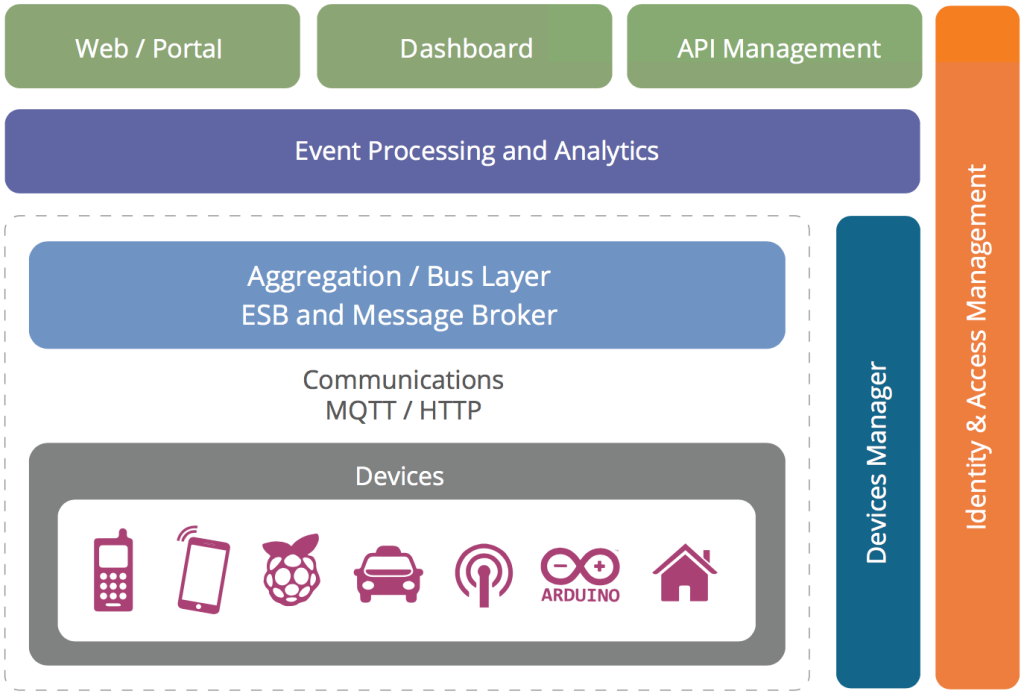
Tabla 21. Componentes y funcionalidades de la arquitectura de referencia de SmartSantander Project

Componente	Funcionalidad
Servidor de prueba	• Federación • Configuración • Gestión de sesión • Análisis de resultados • Reserva de recursos • Programador • Supervisión y gestión de fallos • Administración de recursos • Aplicación
Puerta de enlace	• Configuración • Aplicación • Gestión de sesión • Análisis de resultados • Reserva de recursos
Nodo IoT	• Aplicación • Gestión de sesión • Análisis de resultados • Reserva de recursos

Fuente: Autor

4.3.1.4 Análisis de la arquitectura de referencia de WSO2. Esta arquitectura está compuesta por un conjunto de componentes, con capas horizontales y trasversales (WSO2, 2015), como se muestra en la Figura 13.

Figura 13. Arquitectura de referencia para IoT de WSO2



Fuente: WSO2 (2015)

Las capas y sus funcionalidades se muestran en la *Tabla 22*.

Tabla 22. Capas y funcionalidades de la Arquitectura de referencia para IoT de WSO2

Capa	Funcionalidad
Capa de comunicaciones cliente/externas	En esta capa proporciona formas de comunicación fuera del sistema orientado al dispositivo. Estas formas incluyen tres enfoques principales: • Sitios web y portales que interactúan con los dispositivos y con la capa de procesamiento de eventos. • Paneles con vistas para la analítica y procesamiento de eventos. • APIs de comunicaciones de máquina a máquina. Estas API deben ser gestionadas y controladas a través de un sistema de gestión de API, la cual proporciona tres funciones principales: ○ Un portal enfocado al desarrollador, donde los desarrolladores pueden encontrar, explorar y suscribirse a las API del sistema. ○ Una puerta de enlace que gestiona el acceso a las API realiza comprobaciones de control de acceso. ○ La puerta de enlace publica datos en la capa de análisis, donde se almacena y procesa para proporcionar información sobre cómo se usan las API.

Capa	Funcionalidad
Capa de procesamiento y análisis de eventos	Toma eventos del bus y proporciona la capacidad de procesar y actuar sobre esos eventos. Tiene elementos como: • Almacenamiento de datos • Procesamiento de datos • Procesamiento de eventos
Capa de agregación/bus	En esta capa se agrupa e intermedia las comunicaciones, y su importancia radica en: • La capacidad de soportar un servidor HTTP y/o un corredor MQTT para hablar con los dispositivos. • La capacidad de agregar y combinar las comunicaciones de diferentes dispositivos y de encaminar las comunicaciones a un dispositivo específico. • La capacidad de puentear y transformar entre diferentes protocolos. A nivel de seguridad, esta capa debe realizar don funciones clave: • Actuar como un servidor de recursos OAuth2. • Actuar como un punto de aplicación de políticas (PEP) para el acceso basado en políticas.
Capa de comunicaciones	Esta capa se encarga de soportar la conectividad de los dispositivos, para que estos envíen/reciban datos de la nube o un servidor.
Capa de dispositivos	Tiene las siguientes características: • Comunicación directa o indirectamente a Internet. • Identificación del dispositivo.
Gestión de dispositivos	Posee las siguientes características: • Comunicación con los dispositivos a través de varios protocolos. • Control, ya sea individual o a un conjunto de dispositivos. • Administración remota del software y las aplicaciones desplegadas en el dispositivo. • Bloqueo y/o limpieza del dispositivo si es necesario. • Mantener la lista de identidades de dispositivos y asignarlos a los propietarios. • Mantener información sobre la disponibilidad y la ubicación del dispositivo. • Trabajar con la capa de gestión de identidad y acceso para gestionar los controles de acceso a dispositivos.
Gestión de identidad y acceso	Esta capa proporcionar los siguientes servicios: • Emisión y validación de token OAuth2. • Otros servicios de identidad, incluido el soporte SSO de SAML2 y OpenID Connect para identificar las solicitudes de entrada desde la capa Web • XACML PDP • Directorio de usuarios (por ejemplo, LDAP) • Gestión de políticas para el control de acceso.

Fuente: Autor.

4.3.2 Componentes y funcionalidades genéricas de aplicaciones IoT. Luego de describir las características de las capa o componentes de cada arquitectura de referencia, se analizó las funcionalidades en común que toda aplicación IoT debe tener de acuerdo con las arquitecturas de referencia estudiadas, las cuales se precisan a continuación:

- **Comunicación**. Se da en diferentes niveles de la aplicación IoT, ya sea en una misma capa, como es la entre comunicación dispositivos en la capa física o comunicación entre servicios en una capa superior; o la comunicación entre capas, como es el transporte de los datos recolectados en capa física hasta capa de aplicación, con el tratamiento realizado en cada capa para que sea

útil para la aplicación IoT y sus usuarios. Así mismo, la comunicación entre diferentes redes y formas de comunicación como es la *hop-to-hop*, la de red y la *end-to-end*; y la capacidad de agregar y combinar datos.

- **Procesamiento de datos**. El procesamiento de los datos generados en las capas inferiores, para almacenarla, analizarla y que pueda generar valor para la aplicación que es alimentada por dichos datos.
- **Procesamiento de eventos**. Esta funcionalidad provee a la aplicación IoT la capacidad de procesar y actuar sobre los eventos para iniciar actividades en tiempo real o casi real y tomar acciones autonómicas basadas en los datos de los dispositivos y del resto del sistema.
- **Analítica de datos**. Las aplicaciones IoT producen un gran volumen de datos que no pueden ser analizadas como se realiza en los sistemas tradicionales, por ende, esta funcionalidad brinda la capacidad de procesar estos volúmenes de datos, y a partir de ellos, generar información para la aplicación y para la toma de decisiones.
- **Almacenamiento de datos**. Provee un almacenamiento altamente escalable de los datos que han sido creados y procesados para ser consultados por los usuarios y servicios a través de las diferentes interfaces creadas para tal fin.
- **Gestión de servicios**. La creación y administración de los recursos, búsqueda, descubrimiento y exposición de servicios para hacerlos accesibles, verificar su disponibilidad y asignación publicación/suscripción entre servicios.
- **Gestión de dispositivos**. Son actividades relacionadas con la comunicación de los dispositivos, la operación de los dispositivos como es la supervisión de su estado de trabajo del dispositivo, su activación o desactivación de forma remota, solicitud de datos de monitorización realizado en entornos físicos, actualización de software y/o *firware*, el mantenimiento de la información sobre su disponibilidad y ubicación, entre otras gestiones relacionadas con los dispositivos.
- **Autenticación, autorización y contabilidad**. Provee la gestión de las identidades de los dispositivos, recursos y usuarios del sistema, y verifica a qué recursos pueden acceder y las acciones que estos pueden realizar sobre el sistema.
- **Gestión de la red**. Ya sea suministrando y controlando la conectividad, gestionando de movilidad y la topología de red, el transporte de paquetes y el intercambio de protocolos.
- **Visualización de los datos**. Presentar los datos e información generada a parte de ellos a los usuarios de las aplicaciones IoT, o permitir las interfaces para que otras aplicaciones IoT usen los datos generados por esta.

4.3.3 Análisis de funcionalidades. Las arquitecturas de referencia – AR estudiadas en esta sección difieren en su estilo arquitectural, ya sean representadas en componentes o capas, y estas características corresponde a la visión de sus autores y a las características que querían abordar. Cada AR

hace un nivel de abstracción diferente de las funcionalidades que debe tener una aplicación IoT, y las representa a través de diferentes elementos dentro de la arquitectura.

Para evitar la confusión de sobre qué elemento trabajar, ya que difieren en nombre o nivel de detalle entre una u otra AR, se realizó una abstracción de mayor nivel de las funcionalidades que debe abordar una aplicación IoT genérica. Estas funcionalidades son las comunes a todas las AR, y se obviaron algunas que eran muy propias de cada propuesta, como, por ejemplo, la funcionalidad de modelado de procesos que tiene la AR del *IoT-A Project*.

Mientras se realizaba esa abstracción de las funcionalidades, se percibió que dichas funcionalidades ubicadas a diferentes niveles de una aplicación IoT concuerdan con tres paradigmas de computación distribuida que se trabajan en la actualidad dentro del paradigma de IoT: *cloud computing*, *fog computing* y *dew computing*.

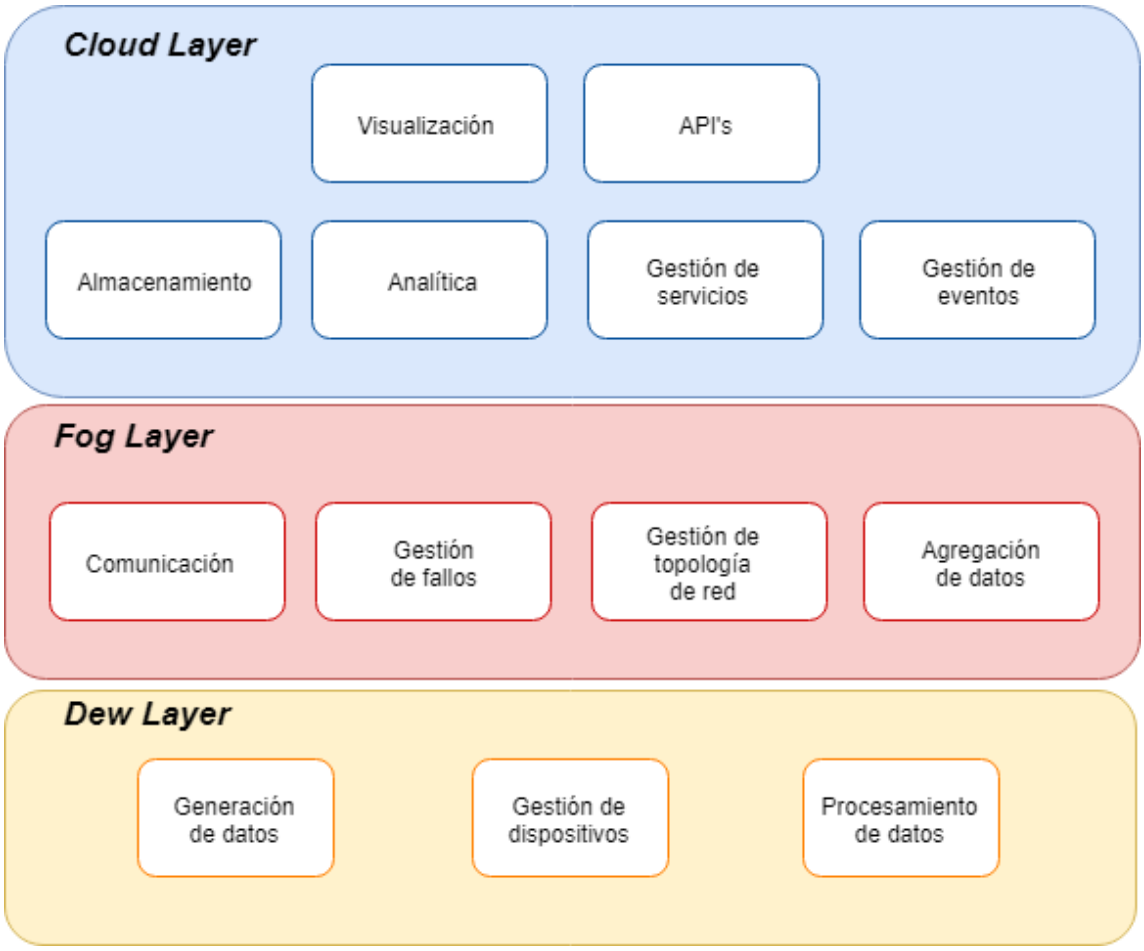
De esta forma, funcionalidades como el almacenamiento, el procesamiento y analítica de datos, el procesamiento de eventos, la visualización de contenidos tiene que ver con las funcionalidades que se manejan en el *cloud computing*. Asimismo, funcionalidades de comunicación, gestión de red, procesamiento y fusión de datos se asimilan al concepto de *fog computing*; y funcionalidades como la gestión de los dispositivos, la toma de decisiones respecto a ellos se asimila al concepto de *dew computing*.

En la subsección 2.2.4 se realiza una descripción de cada una de estas tecnologías de computación distribuida.

4.3.4 Diseño de arquitectura genérica de IoT. El objetivo de proponer una arquitectura genérica de IoT que contenga funcionalidades comunes de IoT, es ofrecer interoperabilidad del modelo propuesto en este capítulo con las diferentes arquitecturas de referencias y arquitecturas particulares para IoT.

Para esta arquitectura genérica se eligió un estilo arquitectural en capas. Esta arquitectura costa de tres capas: *Cloud Layer*, *Fog Layer* y *Dew Layer*; como se muestra en la Figura 14.

Figura 14. Arquitectura IoT genérica



Fuente: Autor

4.3.4.1 ***Cloud Layer***. En esta capa se concentra la mayor infraestructura de una aplicación IoT. En esta capa se tienen las siguientes funcionalidades y capacidades:

- **Visualización**. La aplicación IoT debe proveer varias formas de interacción del usuario con el sistema. Esta capacidad brinda al usuario la posibilidad de acceder a los recursos y servicios que ofrece la aplicación IoT, asimismo, la visualización de los datos generados por esta.
- **API's**. Esta capacidad permitirá a desarrolladores y otras aplicaciones acceder a recursos o servicios de la aplicación IoT y acceder a datos que pueden alimentar otras aplicaciones IoT. Todo eso siempre y cuando estén autorizados por el propietario de dichos recursos.
- **Almacenamiento**. Esta capacidad permite a la aplicación o plataforma IoT almacenar los datos generados por los dispositivos y los nuevos datos generados a partir de ellos. Gracias a las características ofrecidas por el *Cloud Computing* provee a la aplicación IoT de escalabilidad de recursos de almacenamiento.

- **Analítica**. Una aplicación IoT puede generar un gran volumen de datos que no pueden ser analizados siguiendo técnicas tradicionales. El *big data* permite tratar y analizar esta cantidad de información, identificando patrones y generando más información a partir de la información generada generando valor para la organización que use la aplicación IoT.
- **Gestión de servicios**. Esta capacidad permite la búsqueda, el descubrimiento y la resolución de nombre de servicios; expone los recursos disponibles para hacerlos accesibles a otras partes del sistema IoT y a los usuarios.
- Gestión de eventos.

4.3.4.2 ***Fog Layer***. En esta capa se encuentran las capacidades y funcionalidades relacionadas con la comunicación y transporte de datos. Entre ellas se tienen:

- **Comunicación**. Esta funcionalidad provee y gestiona la comunicación entre las capas *Dew* y *Cloud*, y se encarga de transportar información entre ellas. Ofrece funciones de control de conectividad de red y gestión de movilidad.
- **Gestión de fallos**. Su objetivo es identificar, aislar, corregir y registrar las fallas que ocurren en la aplicación IoT.
- **Gestión de topologías de red**. Esta capacidad provee la forma en que se transporta la información, ya sea una comunicación en red, *hop-to-hop* o *end-to-end*.
- Agregación de datos.

4.3.4.3 ***Dew Layer***. En esta capa se encuentran las capacidades y funcionalidades relacionadas con los dispositivos. Entre ellas se tienen:

- **Gestión de dispositivos**. Esta capacidad permite el control de dispositivo, entre las cuales se sienten funcionalidades como: La activación y desactivación de dispositivos de forma remota, la gestión del estado de trabajo del dispositivo, su diagnóstico, la actualización de firmware y/o software. Soportar las diferentes interfaces de los dispositivos conectados mediante diferentes tecnologías.
- **Gestión de los datos**. Esta funcionalidad contiene funcionalidades para la recolección de datos por parte de los nodos sensores y otros dispositivos.
- **Procesamiento de datos**. Aprovechando las capacidades de hardware de algunos dispositivos IoT, en esta capa se puede procesar información para la toma de decisiones que afecten a los nodos sensores y actuadores sin necesidad de enviar todos los datos a la capa superior, procesarlos, generar la toma de decisiones y enviar la orden de actuación a la capa inferior.

4.4 REQUISITOS DE SEGURIDAD PARA APLICACIONES IOT

En esta sección se especifican los requisitos de seguridad que debe considerar una aplicación IoT para garantizar la protección de los datos que se generen, transporten, procesen y almacenen al implementar dicha arquitectura. Los requisitos de seguridad, al igual que los demás requisitos de calidad, se deben considerar desde la fase de diseño de la solución IoT.

Una vez se revisaron las fuentes descritas en la Tabla 11, se determinó los requisitos de calidad relacionados con la seguridad que habían sido consideradas por cada fuente.

Basado en dichos requisitos de calidad, a continuación, se enumeran los requisitos que debe considerarse en el diseño y construcción de una aplicación IoT para reducir el riesgo que los datos sean comprometidos. Estos requisitos se clasifican en cuatro grupos:

4.4.1 Grupo de requisitos para la confidencialidad de la información (GRC). Este grupo de requisitos buscan garantizar la confidencialidad de la información, propiedad que impide que el acceso o divulgación. Este grupo de requisitos se clasifican en tres subgrupos: Requisitos de seguridad, requisitos de privacidad y requisitos de autorización y autenticación.

4.4.1.1 Requisitos de seguridad. Los requisitos de seguridad (RS) se muestran a continuación:

- RS1: Un sistema IoT debe proporcionar a los usuarios el acceso seguro a los recursos.
- RS2: Un sistema IoT debe proporcionar comunicación segura y de confianza entre los dispositivos, los recursos y servicios, para evitar que terceros puedan espiarlas.
- RS3: Un sistema IoT debe proporcionar una medida de seguridad y privacidad del dispositivo.
- RS4: Un sistema IoT debe evitar que un dispositivo se active sin el consentimiento del propietario.

4.4.1.2 Requisitos de privacidad. Los requisitos de privacidad (RP) se muestran a continuación:

- RP1: Un sistema IoT debe proporcionar una alternativa para que los usuarios usen sus servicios de forma anónima.

- RP2: Un sistema IoT debe soportar la comunicación anónima entre dispositivos y proporcionar la confidencialidad de comunicación.
- RP3: Un sistema IoT debe permitir que los usuarios tengan el control de cómo sus datos están expuestos a otros usuarios.
- RP4: Un sistema IoT debe proporcionar la privacidad de la ubicación.
- RP5: Un sistema IoT debe proteger la privacidad de los usuarios que acceden a información sobre entidades o servicios físicos.
- RP6: Un sistema IoT debe evitar el seguimiento del identificador del dispositivo por entidades no autorizadas.

4.4.1.3 Requisitos de autenticación y autorización. Los requisitos de autenticación y autorización (RAA) se muestran a continuación:

- RAA1: Un sistema IoT debe proporcionar diferentes permisos de acceso a la información.
- RAA2: Un sistema IoT debe admitir la limitación de acceso a la red a dispositivos específicos hasta que no presente credenciales apropiadas para unirse a la red.
- RAA3: Un sistema IoT debe apoyarse en mecanismos de control de acceso.
- RAA4: Un sistema IoT debe admitir la autenticación mutua de sujeto.

4.4.2 Grupo de requisitos para la integridad de la información (GRI). Los requisitos de integridad (RI) se muestran a continuación:

- GRI1: Un sistema IoT debe proveer la integridad de la comunicación.
- GRI2: Un sistema IoT debe validar la integridad de entidades virtuales, dispositivos, recursos y servicios.

4.4.3 Grupo de requisitos para la disponibilidad de la información (GRD). Los requisitos de disponibilidad (RD) se muestran a continuación:

- RD1: Un sistema IoT debe garantizar la disponibilidad de infraestructura.
- RD2: Un sistema IoT debe garantizar la disponibilidad de sus servicios.
- RD3: Un sistema IoT debe garantizar la disponibilidad de la red.
- RD4: Un sistema IoT debe garantizar la frescura de los datos.
- RD5: Los servicios IoT debe estar siempre accesible para los usuarios.

4.4.4 Grupo de requisitos para el no repudio (GRNP). Los requisitos de no repudio (RND) se muestran a continuación:

- RNP1: Un sistema IoT debe ofrecer una identificación única de los usuarios que solicitan datos a través de los servicios de descubrimiento/búsqueda.
- RNP2: Un sistema IoT debe apoyar la identificación de la ubicación del dispositivo.
- RNP3: Un sistema IoT debe garantizar el no repudio a nivel de recursos de red.

4.5 MODELO DE GESTIÓN DE CIBERSEGURIDAD PROPUESTO

En esta sección se presenta el modelo propuesto para la gestión de la ciberseguridad en aplicaciones IoT. Este modelo está compuesto por tres elementos: El *SMITH model*, que es una representación del dominio de la ciberseguridad para una aplicación IoT genérica; el segundo elemento es una guía de buenas prácticas de ciberseguridad que deben considerarse para asegurar una aplicación IoT, cuyas recomendaciones se organizan según la visión presentada en el *SMITH model*. Finalmente, se presenta el tercer elemento, un instrumento de evaluación buenas prácticas de ciberseguridad para aplicaciones IoT ya construidas, y de esta forma los desarrolladores puedan tomar medidas para reforzar la seguridad de sus soluciones IoT.

4.5.1 *SMITH Model*. El *Security Management in Internet of Things Model – SMITH Model* es una abstracción y representación del dominio de la ciberseguridad para las aplicaciones IoT. Este modelo surge como un aporte a un vacío de investigación que se encontró en el estado del arte realizado en este trabajo, el cual se presenta en la subsección 2.3.1.

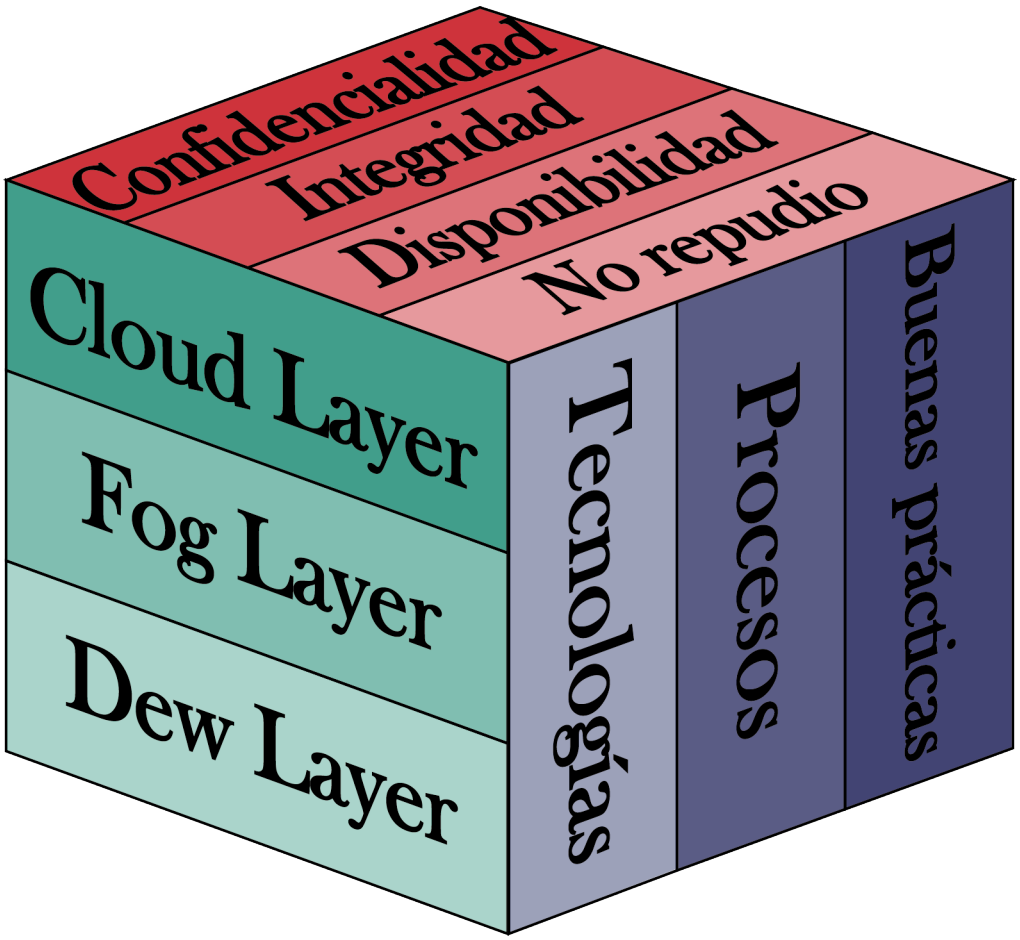
El *SMITH Model* apunta a ser una guía que presente a los desarrolladores de aplicaciones IoT las consideraciones de ciberseguridad que deben tenerse en cuenta desde la fase de diseño de una solución IoT, de forma fácil y concreta.

4.5.1.1 Diseño del *SMITH Model*. Para modelar el dominio de la ciberseguridad en IoT fue necesario entender cómo es el proceso para asegurar una aplicación IoT. Como primer paso, se identificó las dimesiones que interactúan en este dominio, de las cuales se identificaron tres: (i) los componentes y funcionalidades genéricos que integran una aplicación IoT; (ii) los requisitos de seguridad que deben tenerse en cuenta para asegurar la aplicación IoT; y (iii) los elementos que se usan en la ciberseguridad para proteger los activos de la información: tecnologías, procesos y prácticas.

Los componentes y funcionalidades de una aplicación se organizaron en tres capas: *Cloud layer*, *fog layer* y *dew layer*. Los requisitos de seguridad para una aplicación IoT se organizaron en cuatro grupos, cada uno de ellos en una capa: Confidencialidad, integridad, disponibilidad y no repudio. En cuanto a los elementos usados en la ciberseguridad se organizó en tres capas: tecnologías, procesos y prácticas.

La forma más adecuada de representar estas tres dimesiones fue a través de un cubo del cual se usaron las tres caras visibles. En cada una de estas caras, y que de ahora en adelante se denominan vistas, se organizaron las capas que integran cada una de estas dimesiones, como se muestra en la Figura 15.

Figura 15. *SMITH Model*



Fuente: Autor.

4.5.1.2 Descripción del *SMITH Model*. A continuación se describe el *SMITH Model*, sus vistas y las capas que las componen.

Vista frontal

La cara frontal del modelo se consideran las funcionalidades y capacidades de una aplicación IoT a través de una arquitectura IoT genérica propuesta por los autores del modelo[15], el cual está compuesta por tres capas: *Cloud Layer*, *Fog Layer* y *Dew Layer*.

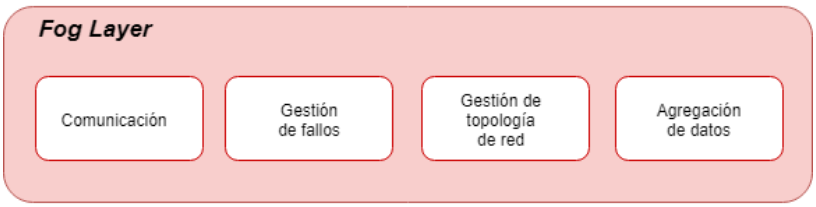
- ***Cloud Layer***. En esta capa se encuentra la mayor parte de la infraestructura de la aplicación IoT y cuenta con capacidades y funcionalidades como la visualización y accesibilidad a recursos, servicios y dato; las API para permitir y facilitar a otras aplicaciones y desarrolladores acceder a recursos de la aplicación IoT; el almacenamiento, el procesamiento de grandes volúmenes de datos, la gestión de servicios y eventos, como se muestra en la Figura 16.

Figura 16. Capacidades y funcionalidades de la *Cloud Layer*.

Fuente: Autor.

- ***Fog Layer***. En esta capa se encuentran las capacidades y funcionalidades relacionadas con la comunicación de datos, la gestión de fallos, la gestión de topología de red y la agregación de datos, como se muestra en la Figura 17.

Figura 17. Capacidades y funcionalidades de la *Fog Layer*.



Fuente: Autor.

[15] En la sección 4.3.5 de este documento se presenta y detalla la arquitectura genérica para IoT propuesta por los autores de este trabajo.

- ***Dew Layer***. Esta capa se encuentran las capacidades y funcionalidades relacionadas con los dispositivos como son la gestión de dispositivos, para la activación y desactivación de dispositivos de forma remota, la gestión del estado de trabajo del dispositivo, su diagnóstico, la actualización de firmware y/o software; también soporta las diferentes interfaces de los dispositivos conectados mediante diferentes tecnologías; la gestión de datos y el procesamiento de datos a nivel inferior, como se muestra en la Figura 18.

Figura 18. Capacidades y funcionalidades de la *Dew Layer*.

Fuente: Autor.

Vista superior

Esta cara está dividida en cuatro capas transversales las cuales están relacionadas con las propiedades de la información que deben protegidas a través de la ciberseguridad. Estas propiedades son la confidencialidad, la integridad, la disponibilidad y el no repudio. En cada una de las capas se considera un grupo de requisitos de calidad relacionados con la seguridad que deben ser considerados en una aplicación IoT para proteger de la información generada.

- **Capa de confidencialidad**. En esta capa se consideran el grupo de requisitos para garantizar la confidencialidad de la información. La confidencialidad es la propiedad que protege los datos de la divulgación o acceso a qué personas, procesos o aplicaciones no autorizados.

- **Capa de integridad**. En esta capa se consideran el grupo de requisitos para garantizar la integridad de la información. Esta propiedad busca proteger que los datos no han sido modificados por personas, procesos o sistemas no autorizadas, de esta forma se garantiza al receptor que los datos recibidos coinciden con los enviados por el emisor, permitiendo detectar si se produjo un cambio en los mismos durante su transmisión, ya sea un cambio, o que se añada o elimine parte del mensaje.

- **Capa de disponibilidad**. En esta capa se consideran el grupo de requisitos para garantizar la disponibilidad de la información. Esta

propiedad considera las medidas necesarias para que la información esté disponible para quienes estén autorizados a acceder a ella, ya sea persona, proceso o sistema.

- **Capa de no repudio**. En esta capa se consideran el grupo de requisitos para garantizar el no repudio. Esta propiedad lo que busca es proteger la participación de alguna de las partes que intervienen en la comunicación, ya sea en toda o una parte de ella.

  Para ello, el no repudio garantiza que se pueda probar el origen, el destino y la entrega del mensaje. De esta forma, en la prueba de origen el receptor puede demostrar ante terceros el origen de los datos recibidos; para la prueba de destino, el emisor de mensaje puede demostrar a terceros que los datos han sido entregados al emisor adecuado; finalmente, en la prueba de entrega, el emisor o receptor pueden demostrar ante terceros la fecha y hora en que se envió el mensaje.

Adicionalmente se enumeran un grupo de requisitos relacionados con la seguridad, el Grupo de requisitos para la infraestructura de seguridad – GRIS que reúne los requisitos que deben cumplir la infraestructura de seguridad de una aplicación IoT.

En la sección 4.4.2 se puede apreciar los grupos de requisitos de seguridad que deben considerarse en una aplicación IoT.

Vista lateral

En esta cara del modelo se consideran los elementos que intervienen en la ciberseguridad: las tecnologías, los procesos y las prácticas. Estos tres elementos son usados para proteger los activos de la información a través del tratamiento de amenazas que ponen en riesgo la información.

- **Tecnologías**. Este elemento es el más técnico y abarcas todas aquellas tecnologías hardware y software que se utilizan para asegurar los activos de la información, salvaguardando la confidencialidad, integridad, disponibilidad y no repudio de la información.

- **Procesos**. Otro factor de importante en la gestión de la ciberseguridad es establecer los procesos a realizar, esto se realiza a través de políticas organizacionales. Estos procesos o políticas se establecen desde un nivel más administrativo, con un alto compromiso de la gerencia para que estas

políticas se implementen de forma adecuada y sean acatadas por cada una de las personas que intervienen en ellos.

- **Buenas prácticas**. En la ciberseguridad las forma en que los usuarios usan e interactúan con la tecnología es muy relevante para proteger los activos de la información. Por esta razón, capacitar a los usuarios en las buenas prácticas de ciberseguridad es de vital importancia dentro de las organizaciones. En el ámbito de la ciberseguridad se dice que el usuario 'es el eslabón más débil de la cadena', que de nada sirve invertir recursos tecnológicos, hardware y software, para proteger los activos de información si las personas no están capacitadas para responder ante los incidentes de seguridad, aumentando el riesgo de ser víctimas de técnicas como la ingeniería social, y se conviertan en la puerta de entrada para que los cibercriminales ingresen a la infraestructura tecnológica de la organización.

4.5.2 Guía de buenas prácticas ciberseguridad para el aseguramiento de aplicaciones IoT. Como parte del *SMITH Model*, se propuso una guía que oriente a los desarrolladores de aplicaciones IoT sobre buenas prácticas de ciberseguridad deben considerar en la etapa de diseño de una aplicación IoT.

En esta guía se consideraron las recomendaciones de seguridad de OWASP en su *IoT Framework Assessment* (2016a), las cuales se organizaron de acuerdo con la vista frontal del modelo SMITH para facilitar la comprensión del panorama de la ciberseguridad de una aplicación IoT.

4.5.2.1 Buenas prácticas de ciberseguridad para *Cloud Layer*. En esta capa se encuentra la mayor infraestructura de la aplicación IoT y abarca funcionalidades como las que se describen en la Figura 17.

Figura 17. Capacidades y funcionalidades de la *Cloud Layer*

Fuente: Autor.

Comunicaciones cifradas

Esta capa debe admitir las comunicaciones cifradas, incluidos los certificados de seguridad, para identificarse con las otras componentes de la aplicación. De igual forma, se debe admitir certificados criptográficos para identificar otros componentes también, para la verificación de identidad bidireccional.

Autenticación

Los elementos de esta capa que requieran de autenticación deben permitir una autenticación compleja, incluida la autenticación de múltiples factores, evitando las credenciales predeterminadas. La interfaz también debe considerar algunos mecanismos de seguridad como de mitigación de fuerza bruta y enumeración contraria a la cuenta y debería permitir a los usuarios establecer fácilmente y restablecer de forma segura la información de la cuenta.

Credenciales de autenticación segura

Las credenciales de autenticación, en cualquier forma como son contraseñas, identificaciones del dispositivo, entre otros, deben ser apropiadamente codificados y cifrados antes de almacenarlos; y donde dichos mecanismos de almacenamiento también deben ser uniformemente fuertes.

Para conocer más sobre el almacenamiento de contraseñas puede consultar la *Password Storage Cheat Sheet* (2017c) en el siguiente enlace.

Almacenamiento cifrado y capacidad de clasificación de datos y segregación

Las aplicaciones IoT manejan un gran volumen de datos, y estos pueden contener información sensible sobre los usuarios, por esta razón, siempre que sea posible, el marco debe admitir el cifrado de datos que es almacenada, así como en cualquier mecanismo de exportación o copia de seguridad de dichos datos.

Este volumen de datos contempla una variedad de datos, algunos datos pueden ser muy sensibles y otros datos pueden ser benignos. Se debería proporcionar las capacidades para clasificar los datos y protegerlos dependiendo de la clasificación que se les dé. Se deben implementar controles que limiten el acceso y la exposición de datos confidenciales según la clasificación.

Capacidad de utilizar comunicaciones encriptadas entre componentes

Las comunicaciones entre los diferentes componentes de esta capa y otras capas deben utilizar un canal de comunicaciones cifrado para evitar que los datos se expongan en tránsito.

Informes y alertas de eventos de seguridad

En esta capa se concentra la mayor parte de la infraestructura de la aplicación IoT, y cuya capacidad hardware permite obtener grandes recursos de cómputo y almacenamiento, por lo tanto, es la capa con mayor capacidad de gestión de recursos y seguridad, sobre ella misma y la otras capas.

Esta debe contar con sólidas funciones de supervisión, generación de informes y alertas de eventos de seguridad. A esta capa se debe proveer de funciones que permitan detectar y reaccionar a la actividad maliciosa; además, debe poder segregar a los malos actores, limitar el acceso a partes maliciosas e integrarse fácilmente con los sistemas de registro y prevención de intrusión y registro de terceros.

Actualizaciones automáticas y verificación de actualización

Una buena práctica para reducir los riesgos de seguridad es mantener el software actualizado y permitir parches y actualizaciones es fundamental. Se debe identificar claramente la versión del software en funcionamiento y permitir parches y actualizaciones de software para componente de esta capa. Un proceso de actualización automática aumenta la probabilidad de que los sistemas se mantengan actualizados. Se sebe generar alertas automáticas de actualizaciones para los componentes que no se actualizan automáticamente.

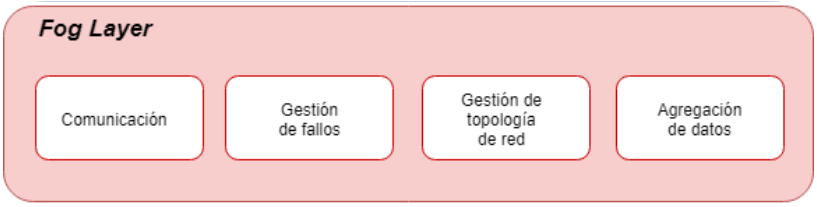
Utilice los últimos componentes de terceros actualizados

Para el *Cloud* hay muchos componentes, plataformas y servicios desarrollados por terceros que pueden usarse en nuestras propias soluciones, y respecto a esto, se debe implementar los desarrollos más actualizados. También se debe proveer la capacidad de mantenerse informado sobre nuevas versiones y actualizaciones de estos componentes, para su instalación a medida que se degradan las actualizaciones de seguridad que estén disponibles. En cuando a la hora de actualizar, se debe garantizar que las actualizaciones se distribuyan a través de un canal seguro y se verifiquen antes de la instalación, para descartar que se ha descargado la versión original ofrecida por el emisario.

4.5.2.2 Buenas prácticas de ciberseguridad para *Fog Layer*. En esta capa se encuentran las funcionalidades relacionadas con la comunicación. Esta capa

comunica la *Fog Layer* con la *Cloud Layer* y reúne funcionalidades como las que se muestran en la Figura 18.

Figura 18. Capacidades y funcionalidades de la *Fog Layer*.



Fuente: Autor.

Comunicaciones cifradas multidireccionales

Esta capa se encarga de la comunicación entre las capas *Cloud* y *Dew*, y se debe velar porque que las comunicaciones sean seguras para no degradar la seguridad de los mensajes en cualquier dirección siempre que sea posible. Se puede presentar que existan los canales de comunicaciones seguros y no seguros, en cuyo caso se debe prestar especial atención a la interceptación, manipulación e inyección de datos en puntos finales inseguros. También se debe proporcionar capacidades para segmentar y aislar las comunicaciones cuando sea posible.

Autenticación fuerte de componentes

Los componentes de esta capa deberían proporcionar mecanismos de autenticación tan sólidos como cualquier otro componente en la aplicación. Siempre que sea posible, los componentes de esta capa deben autenticarse de forma multidireccional para garantizar comunicaciones de confianza con las capas *Cloud* y *Dew*. Las capacidades criptográficas en la autenticación deberían ser un componente sólido de la solución de infraestructura.

Almacenamiento

La puerta de enlace puede servir como un único punto de falla, o ataque, en el ecosistema y debe almacenar solo la cantidad mínima de información, en un formato encriptado si es posible.

Registro y alerta

La puerta de enlace tendrá acceso a un volumen de tráfico y debería poder registrar y alertar en función del registro de eventos. El marco puede incluir integración con servicios de registro estándar o sistemas de detección de

intrusos. Incluso se podría admitir métodos alternativos para alertar en la puerta de enlace, como, por ejemplo, los SMS.

Capacidades de detección e informe de anomalías

Se debería permitir que se observe y monitoree el tráfico de comunicaciones y el comportamiento de los componentes. En esta capa se ubican los elementos adecuados para controlar el tráfico hacia y desde la nube y aquellos elementos que deben admitir la detección de anomalías o integrarse fácilmente con anomalías y sistemas de detección de intrusos. Con el manejo de elementos robustos incluso se podría admitir capacidades de prevención de intrusos para excluir a los actores sospechosos del ecosistema.

Utilice los últimos componentes de terceros actualizados

En esta capa, muchos de los elementos son desarrollados por terceros, y respecto a esto, se debe implementar los desarrollos más actualizados. También se debe proveer la capacidad de mantenerse informado sobre nuevas versiones y actualizaciones de estos componentes, para su instalación a medida que se degradan las actualizaciones de seguridad que estén disponibles. En cuando a la hora de actualizar, se debe garantizar que las actualizaciones se distribuyan a través de un canal seguro y se verifiquen antes de la instalación, para descartar que se ha descargado la versión original ofrecida por el emisario.

Actualizaciones automáticas y/o informes de versión

Una buena práctica para reducir los riesgos de seguridad es mantener el software actualizado. Se debe identificar claramente la versión del software en funcionamiento y permitir parches y actualizaciones de software. Un proceso de actualización automática aumenta la probabilidad de que los sistemas se mantengan actualizados.

4.5.2.3 Buenas prácticas de ciberseguridad para *Dew Layer*. En esta capa se encuentran los dispositivos IoT, que son heterogéneos y pueden diferir en sus capacidades de cómputo, almacenamiento, comunicación y energía; la cantidad de ellos puede variar dependiendo de la aplicación y reúne funcionalidades como las que se muestran en la Figura 19.

Figura 19. Capacidades y funcionalidades de la *Dew Layer*.

Fuente: Autor.

Las recomendaciones de ciberseguridad se clasificaron en tres categorías: recomendaciones generales, recomendaciones para los dispositivos y las recomendaciones relacionadas con las comunicaciones.

- **Recomendaciones generales**

A continuación, se presentan las recomendaciones generales.

Registro robusto de eventos

Se debería ofrecer un registro robusto de eventos realizadas en el sistema, incluido el registro de eventos de seguridad. Estos registros deben ser personalizables y deben informar eventos sensibles en un formato utilizable para usuarios finales, gerentes y operadores; los cuales también pueden ser útiles al proporcionar evidencia forense en caso de incidentes de seguridad.

Capacidades de identificación criptográfica

Es importante proveer a los componentes de esta capa de capacidades criptográficas para verificar la confianza de otros componentes de la aplicación IoT, como *gateways*, la *cloud* y dispositivos móviles, y la gestión del ciclo de vida criptográfico. Gestionar el ciclo de vida criptográfico implica respaldar la emisión y reedición de material criptográfico, el vencimiento de certificados criptográficos, un mecanismo de verificación de revocación, y un sistema de firma de material clave.

Cifrado de almacenamiento

Muchos dispositivos cuentan con la capacidad hardware de almacenar datos. Los datos confidenciales pueden ser susceptibles de robo o exposición a menos que se almacene con las consideraciones de seguridad adecuadas; estos datos pueden incluir lectura del sensor, configuración, credenciales de autenticación o claves criptográficas.

Autenticación local robusta y soporte de autenticación de múltiples factores

El uso de contraseñas es común en muchos servicios y recursos; estas contraseñas, en la medida de lo posible, deben ser complejas y se debe contar con autenticación de múltiples factores. Asimismo, los componentes desarrollados o usados que cuentes con mecanismo de autenticación deben informar o registrar intentos de autenticación fallidos y proporcionar un mecanismo de retardo o bloqueo exponencial para evitar ataques de fuerza bruta.

Capacidades M2M seguras

La comunicación maquina a maquita (M2M, *Machite-to-Machine*) debe basarse en la confianza a través de capacidades como la autorización, la verificación y la autenticación entre máquinas que hacen parte de la red. Estas capacidades, en la medida de lo posible, debe extenderse fuera de línea para evitar fallar al momento de perder la conexión. La confianza en las comunicaciones M2M podría gestionarse con la confianza transitiva, de este modo, un propietario podría certificar una cantidad de dispositivos que luego podrían autenticarse y confiar en función del propietario, independientemente del dispositivo o plataforma del ecosistema.

Interfaz web segura

Algunos dispositivos IoT tienen la capacidad hardware para ofrecer servicios web, y estos servicios, aunque no sean el servicio principal de la aplicación también deben asegurarse. Se debe garantizar que estas aplicaciones web implementen contramedidas de seguridad contra vulnerabilidades comunes tales como omisión de autenticación, scripts entre sitios y falsificación de solicitudes entre sitios. Las interfaces web deben presentarse sobre el protocolo TLS (HTTPS) y no deben usar certificados autofirmados o no válidos. También se debería limitar el acceso a la interfaz web para evitar el uso o abuso no autorizado.

Para más información, puede consultar este mismo apartado en la *Cloud Layer*.

Utilice pilas y protocolos de red establecidos y probados

En las aplicaciones se debe utilizar protocolos y pilas de protocolos de red que cuenten con buen soporte para evitar vulnerabilidades de seguridad comunes que pueden presentarse en otros más nuevos, que no han sido probados ampliamente o son exóticos. Para limitar la superficie de ataque se deben limitar

el número de protocolos al mínimo posible y desactivar los que no se estén usando.

Utilice los últimos componentes de terceros actualizados

En el ecosistema IoT hay muchos componentes, plataformas y servicios desarrollados por terceros que pueden usarse en las propias soluciones, y respecto a esto, se debe implementar los desarrollos más actualizados. También se debe proveer la capacidad de mantenerse informado sobre nuevas versiones y actualizaciones de estos componentes, para su instalación a medida que se degradan las actualizaciones de seguridad que estén disponibles. En cuando a la hora de actualizar, se debe garantizar que las actualizaciones se distribuyan a través de un canal seguro y se verifiquen antes de la instalación, para descartar que se ha descargado la versión original ofrecida por el emisario.

Se recomienda usar un lenguaje de programación seguro o sujeto a escrutinio

Los componentes deben estar escritos en lenguajes de programación que posean contramedidas de seguridad y demuestren un historial de seguridad sólida.

- **Recomendaciones relacionadas con los dispositivos**

A continuación, se presentan las recomendaciones relacionadas con los dispositivos IoT.

Actualizaciones automáticas y/o informes de versión

Una buena práctica para reducir los riesgos de seguridad es mantener el software actualizado y permitir parches y actualizaciones es fundamental. Se debe identificar claramente la versión del software en funcionamiento y permitir parches y actualizaciones de software. Un proceso de actualización automática aumenta la probabilidad de que los sistemas se mantengan actualizados.

Verificación de actualización

Las actualizaciones deben entregarse a través de un canal seguro y verificarse después de la descarga para garantizar que las actualizaciones sean legítimas. Los valores hash binarios de firma (y comprobación) y actualización entregados a través de un canal verificado y encriptado garantizan que las actualizaciones maliciosas no estén instaladas en un dispositivo. Tenga en cuenta que el acceso físico puede permitir que un atacante cargue un binario para colocarlo

directamente en un dispositivo, por lo que las actualizaciones deben verificarse antes de la instalación, en lugar de simplemente verificar una descarga.

Sin contraseñas predeterminadas

Se debe evitar las credenciales por defecto en todo el ecosistema, ya sean en componentes de autenticación local, así como componentes de autenticación para la nube, *gateway*, el dispositivo móvil u otros dispositivos del ecosistema. Las credenciales se deben establecer y restablecer cada cierto tiempo, y se debe establecer una política para el manejo de estas.

Las interfaces están deshabilitadas por defecto

Es recomendable desactivar la mayor cantidad posible de servicios y características de manera predeterminada del sistema, e ir habilitando dichas funciones según sea necesario para minimizar la superficie de ataque.

Capacidades de control y gestión del dispositivo

En cuanto a la gestión de dispositivos, se debería permitir la supervisión de su plataforma, y en lo posible, la administración de capacidades que permitan la detección de debilidades de seguridad o vulnerabilidades.

Funciones de seguridad fuera de línea

El marco debe asumir que el componente de borde puede perder conectividad y recurrir a características de seguridad locales en ausencia de recursos de red. Estas características de seguridad fuera de línea deben ser tan sólidas como las características en línea para evitar que los atacantes interrumpan las comunicaciones y degraden las medidas de seguridad.

Consideraciones de propiedad transitiva

Los dispositivos de IoT deben gestionarse ante el cambio de propietario. Se debería permitir que el dispositivo sea borrado, reiniciado o y que los datos sean destruidos para proteger la información del propietario anterior.

- **Recomendaciones relacionadas con las comunicaciones**

A continuación, se presenta las recomendaciones relacionadas con las comunicaciones entre los dispositivos y otros componentes de esta capa.

Cifrado de comunicaciones

Siempre que sea posible se debe proveer comunicaciones cifradas de extremo a extremo para garantizar que estas no puedan ser interceptadas o redirigidas.

Se debe rastrear y contener datos de fuentes potencialmente contaminadas o inseguras

Es posible que se requiera que los dispositivos de IoT procesen datos de canales que no se pueden proteger. El marco debería permitir alguna forma de etiquetado de datos o sanitización para rastrear y contener datos de fuentes no confiables.

4.5.3 Instrumento de evaluación. Este instrumento de evaluación permite a los desarrolladores IoT identificar qué elementos de ciberseguridad, básicos y de forma general, no están considerando en el desarrollo de sus aplicaciones. Esto con el fin de salvaguardar la información generada, transportada y almacenada en sus soluciones. En la *Tabla 23*, se muestra los controles de ciberseguridad para *Cloud Layer*.

Tabla 23. Controles de ciberseguridad para *Cloud Layer*.

Controles de ciberseguridad para *Cloud Layer*	
Control	**Cumple**
Comunicaciones cifradas	
¿La aplicación cuenta con cifrado en las comunicaciones?	
Autenticación	
¿Se contempla autenticación es servicios y componentes del sistema?	
¿Se incluye autenticación de múltiples factores?	
¿La interfaz incluye funciones de mitigación de fuerza bruta?	
¿Se cambiaron las credenciales por defecto?	
¿Se permite a los usuarios establecer fácilmente y restablecer de forma segura la información de la cuenta?	
Credenciales de autenticación segura	
¿Las credenciales de autenticación se codifican y se cifran apropiadamente antes de almacenarlos?	
Almacenamiento cifrado	
¿Se implementa el cifrado de datos en reposo?	
¿Se cifra cualquier mecanismo de exportación o copia de seguridad?	
Informes y alertas de eventos de seguridad	
¿Se implementan funciones de supervisión, generación de informes y alertas de eventos de seguridad?	
Actualizaciones automáticas y verificación de actualización	
¿Se implementa las actualizaciones automáticas?	
¿Se implementa la verificación de actualizaciones?	
Utilice los últimos componentes de terceros actualizados	
¿Componentes de terceros usados en la aplicación están actualizados?	

Controles de ciberseguridad para *Cloud Layer*	
Control	**Cumple**
¿Se reciben notificaciones cuando los terceros ofrecen actualizaciones de software y de seguridad de sus componentes?	
¿Las actualizaciones se distribuyen a través de un canal seguro y se verifiquen antes de la instalación?	

Fuente: Autor.

En la Tabla 24, se muestra los controles de ciberseguridad para *Fog Layer*.

Tabla 24. Controles de ciberseguridad para *Fog Layer*.

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
Comunicaciones cifradas multidireccionales	
¿Se cifran las comunicaciones?	
¿Se gestiona la seguridad al interactuar con canales de comunicación inseguros para prevenir, entre otras cosas, la interceptación, manipulación e inyección de datos?	
Autenticación fuerte de componentes	
¿Se proporcionan mecanismos de autenticación multidireccional con las otras capas de la aplicación?	
¿Las capacidades criptográficas en la autenticación son un componente sólido de la solución?	
Almacenamiento	
¿La puerta de enlace almacena solo la cantidad mínima de información, en un formato cifrado, de ser posible?	
Registro y alerta	
¿La puerta de enlace tiene acceso al volumen de tráfico y puede registrar y alertar en función del registro de eventos?	
Capacidades de detección e informe de anomalías	
¿Se permite que se observe y monitoree el tráfico de comunicaciones y el comportamiento de los componentes?	
¿Existen elementos para la detección de anomalías y la detección de intrusos?	
¿Estos elementos poseen la capacidad de prevención de intrusos para excluir a los actores sospechosos del ecosistema?	
Utilice los últimos componentes de terceros actualizados	
¿Se están usando las versiones más recientes de los componentes, plataformas y servicios desarrollados por terceros?	
¿Se provee la capacidad de mantenerse informado sobre nuevas versiones de software y actualizaciones de seguridad de los componentes a medida que estén disponibles?	
¿Las actualizaciones se distribuyen a través de un canal seguro y estas se verifican antes de la instalación?	
Actualizaciones automáticas y/o informes de versión	
¿El software se mantiene actualizado a las últimas versiones estables disponibles?	

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
¿El proceso de actualización se realiza automáticamente o se generan alertas para que se instalen manualmente en el menor tiempo posible a partir que estas estén disponibles?	

Fuente: Autor.

En la Tabla 25, se muestra los controles de ciberseguridad para *Dew Layer*.

Tabla 25. Controles de ciberseguridad para *Dew Layer*.

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
Registro fuerte de eventos	
¿Se realiza un registro de los eventos realizados en el sistema, incluido los eventos de seguridad?	
Capacidades de identificación criptográfica	
¿Se gestiona el ciclo de vida criptográfico para verificar la confianza de los elementos del sistema?	
Cifrado de almacenamiento	
¿Se cifran los datos sensibles almacenados en los dispositivos como lectura del sensor, configuración, credenciales de autenticación o claves criptográficas?	
Fuerte autenticación local y soporte de autenticación de múltiples factores	
¿Se cambiaron las credenciales por defecto de servicios y componentes usados?	
¿Se usan contraseñas robustas en los servicios de autenticación?	
En la medida de lo posible, ¿se cuenta con autenticación de múltiples factores?	
¿Se cuentan con mecanismos que informen y registren intentos de autenticación fallidos y se proporciona un mecanismo de retardo o bloqueo antes cierto número de intentos fallidos?	
Capacidades M2M seguras	
¿Se hace uso de capacidades como la autorización, la verificación y la autenticación para proporcionar comunicaciones M2M basadas en la confianza?	
En la medida de lo posible, ¿las capacidades como autorización, la verificación y la autenticación se extienden su funcionalidad a un estado *offline*?	
¿La confianza en las comunicaciones M2M se gestiona con la confianza transitiva?	
Interfaz web seguras	
¿Se implementen contramedidas de seguridad contra vulnerabilidades comunes tales como omisión de autenticación, scripts entre sitios y falsificación de solicitudes entre sitios, entre otros?	
¿Se hace uso del protocolo TLS (HTTPS) y no se usan certificados autofirmados o no válidos?	
¿Se limita el acceso a la interfaz web para evitar el uso o abuso no autorizado?	
Utilice pilas y protocolos de red establecidos y probados	
¿Los protocolos y pilas de protocolos usando cuentan con amplio soporte de prueba y actualizaciones?	
¿Se desactivan los protocolos que no son usados?	

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
Utilice los últimos componentes de terceros actualizados	
¿Se están usando las versiones más recientes de los componentes, plataformas y servicios desarrollados por terceros?	
¿Se provee la capacidad de mantenerse informado sobre nuevas versiones de software y actualizaciones de seguridad de los componentes a medida que estén disponibles?	
¿Las actualizaciones se distribuyen a través de un canal seguro y estas se verifican antes de la instalación?	
Se recomienda usar un lenguaje de programación seguro o sujeto a escrutinio	
¿Los componentes están escritos en lenguajes de programación que posean contramedidas de seguridad y demuestren un historial de seguridad sólida?	
Actualizaciones automáticas y/o informes de versión	
¿El software se mantiene actualizado a las últimas versiones estables disponibles?	
¿El proceso de actualización se realiza automáticamente o se generan alertas para que se instalen manualmente en el menor tiempo posible a partir que estas estén disponibles?	
Verificación de actualización	
¿Se provee que las actualizaciones se entreguen a través de un canal seguro y se verifiquen después de la descarga para garantizar que las actualizaciones sean legítimas?	
Sin contraseñas predeterminadas	
¿Las credenciales por defecto se han cambiado en todos los componentes y servicios usados en la aplicación?	
¿Se ha establecido una política para el manejo de las credenciales como es el establecerlas y restablecerlas cada cierto tiempo según sea necesario y otras consideraciones?	
Las interfaces están deshabilitadas por defecto	
¿Se desactivan las interfaces que no estén en uso?	
Capacidades de control y gestión del dispositivo	
¿Se permite la gestión de la plataforma del dispositivo y sus capacidades para la detección de debilidades de seguridad o vulnerabilidades?	
Funciones de seguridad fuera de línea	
¿Las características de seguridad siguen funcionando en modo *offline* y son tan sólidas como las características *online*?	
Consideraciones de propiedad transitiva	
Ante el cambio de propietario, ¿se permitir que el dispositivo sea borrado, reiniciado o y que los datos sean destruidos para proteger la información del propietario anterior?	
Cifrado de comunicaciones	
Siempre que sea posible, ¿se provee comunicaciones cifradas de extremo a extremo para garantizar que estas no puedan ser interceptadas o redirigidas?	
Rastrea y contiene datos de fuentes potencialmente contaminadas (inseguras)	
¿Se permite de alguna forma de etiquetado de datos o sanitización para rastrear y contener datos de fuentes no confiables?	

Fuente: Autor.

Nota: En el Anexo B se presenta el modelo y sus elementos propuesto en esta sección de una forma más a fin a la estructura de documento de estas características.

5. MODELO CONCEPTUAL DEL DOMINIO DE LA CIBERSEGURIDAD PARA APLICACIONES IOT

En este capítulo se representa, mediante un lenguaje de modelamiento, el dominio de la ciberseguridad para aplicaciones IoT. Esto corresponde al resultado del segundo objetivo específico definido en esta investigación. Esta representación corresponde a un modelo conceptual en el que se definieron los conceptos relevantes del dominio de la ciberseguridad y cómo estos se relacionan con el dominio de IoT. El propósito de este modelo es describir cómo interviene la ciberseguridad, y sus componentes, para asegurar una aplicación IoT.

Este modelo conceptual es una representación más formal del *Security Management in Internet of THings Model – SMITH Model* presentado en el capítulo 4 de este documento. Del *SMITH Model* se extrajo dos grupos de conceptos: los conceptos claves del dominio IoT y los conceptos claves del dominio de ciberseguridad. Con el primer grupo de conceptos se definió el modelo del dominio IoT. Con el segundo grupo de conceptos, se propuso seis componentes de ciberseguridad que se integraron con el modelo IoT definido. El modelo IoT integrado con los componentes de ciberseguridad conforman el modelo de ciberseguridad para aplicaciones IoT, llamado *IoT Cybersecurity Domain Model* – IoT-CyDM.

El *SMITH Model* e IoT-CyMD son dos representaciones del dominio de ciberseguridad para aplicaciones IoT. La diferencia entre ambos es que el *SMITH Model* es una abstracción libre, representada a través de un cubo, en el que la forma como sus capas se relacionan sigue una lógica que describe la relación de los elementos descritos en el dominio de la ciberseguridad. Además de la abstracción, este modelo tiene un gran componente narrativo. Por otro lado, IoT-CyDM es una representación del dominio de la seguridad, pero siguiendo las reglas de un lenguaje gráfico utilizado para describir sistemas, en este caso, el lenguaje de modelamiento unificado (UML – *Unified Modeling Language*).

Este capítulo está dividido en dos secciones. En la sección 5.1, se propone un modelo del dominio de IoT. Como primer paso, de identifican los conceptos claves del dominio IoT. Seguidamente, se representa dicho domino usando UML, se describe el dominio y se definen los conceptos claves que lo conforman.

En la sección 5.2, se presenta el modelo del dominio de ciberseguridad para aplicaciones IoT. Como punto inicial se propuso seis componentes de ciberseguridad que satisfacen los requisitos de seguridad que deben considerarse en una aplicación IoT. Seguidamente los componentes de ciberseguridad se integraron con el modelo IoT, resultando el modelo del dominio

de ciberseguridad propuesto y la definición de los seis componentes de ciberseguridad.

5.1 MODELO DEL DOMINIO IOT

El primer paso para modelar un dominio es definir los conceptos claves que intervienen en él y la relación entre ellos. Para definir los conceptos claves se debe considerar que estos conceptos sean independientes de tecnologías específicas y de los casos de uso; además, que los conceptos no cambien mucho en las próximas décadas.

En esta sección se propone un modelo del dominio IoT, cuyos conceptos claves se identificaron a partir del trabajo realizado en el *SMITH Model* y en la literatura. En la subsección 5.1.1 se presenta y se definen los conceptos claves identificados; en la subsección 5.1.2 se exhibe y describe el modelo del dominio IoT propuesto.

5.1.1 Concepto claves del dominio IoT. Para identificar los conceptos claves del dominio de IoT se tomó los requisitos de seguridad considerados en el *SMITH Model* y se identificó los diferentes elementos que se deben proteger en IoT. Estos conceptos claves se corroboraron con otras fuentes, como arquitecturas de referencias propuestas y la literatura científica. Los conceptos claves que intervienen en el dominio de IoT son:

5.1.1.1 Servicios. El concepto de Servicio se menciona o describe, de mayor o menor proporción, en trabajos relevantes de la literatura científica como son el de Gubbi et al. (2013), Borgia (2014), L. Da Xu et al. (2014), Atzori et al. (2010), Li, Xu, & Zhao (2015), Patel y Cassou (2015), Atzori et al. (2017).

5.1.1.2 Entidades. IoT es una infraestructura de red global donde las cosas, también conocidas en la literatura como objetos inteligentes o entidades, interactúan entre sí. Las entidades pueden recopilar información del entorno e interactuar/controlar el mundo físico (Borgia, 2014). Existen dos tipos de entidades: las físicas y virtuales (Atzori et al., 2017; L. Da Xu et al., 2014; Li et al., 2015; Miorandi et al., 2012; Van Kranenburg, 2008).

5.1.1.3 Recursos. Borgia (2014) y Cavalcante et al. (2016) coinciden es que existen recursos físicos y virtuales. Para Patel y Cassou (2015) un recurso es una representación conceptual de un sensor, un actuador, almacenamiento o una interface de usuario. En el trabajo realizado por Atzori et al.(2017) los objetos, servicios, servidores de datos, entre otros son recursos, lo cuales son

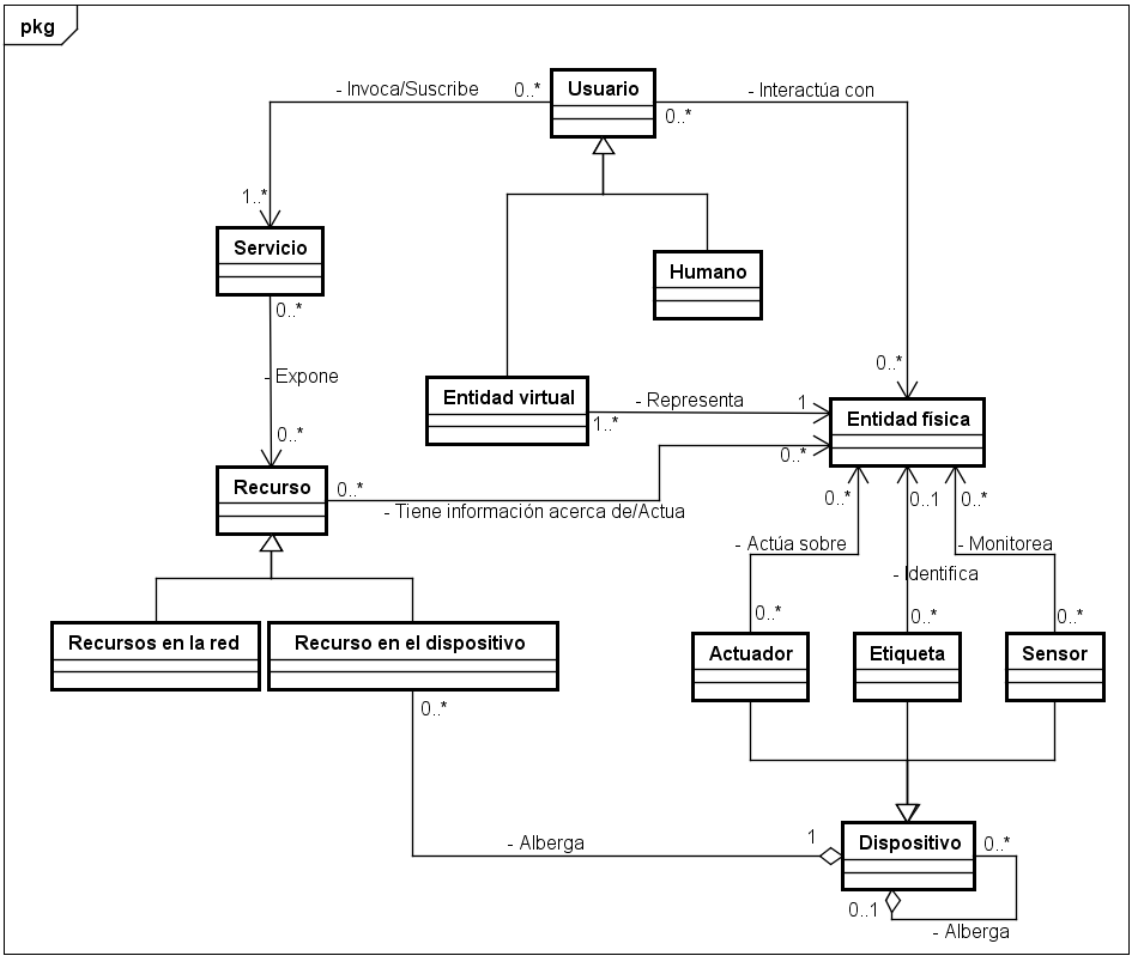
necesarios identificar. En Gubbi et al. (2013) se menciona diferentes tipos de recursos como son: recursos de red, recursos de dispositivo, de cómputo y almacenamiento y recursos en la nube.

5.1.1.4 Dispositivos. Un dispositivo es una entidad que provee recursos con la capacidad de interactuar con otros dispositivos (Patel & Cassou, 2015), los cuales pueden ser virtualizados (Atzori et al., 2017). Este concepto clave también se referencia en trabajos como los de Gubbi et al. (2013), Borgia (2014), L. Da Xu et al. (2014), I. Lee & Lee (2015), Li, Xu, & Zhao (2015). La mayoría de los trabajos mencionados en esta subsección coinciden en tres clases de dispositivos: los sensores, actuadores y etiquetas.

5.1.1.5 Usuarios. De acuerdo con Miorandi et al. (2012) el IoT es el «cambio de una Internet utilizada para interconectar dispositivos de usuario final a una Internet utilizada para interconectar objetos físicos que se comunican entre sí y/o con humanos para ofrecer un servicio dado» (p. 1498, mi traducción). Este concepto también es mencionado en trabajos como el de Gubbi et al. (2013), Borgia (2014), L. Da Xu et al. (2014), Atzori et al. (2010), Li, Xu, & Zhao (2015), Atzori et al. (2017).

5.1.2 Representación del dominio IoT. Una vez identificado los conceptos claves del dominio de IoT, estos conceptos y las relaciones se representaron usando un lenguaje semi-formal, en este caso, el lenguaje unificado de modelamiento – UML por sus siglas en inglés *Unified Modeling Language*. El modelo del dominio IoT propuesto se muestra en la Figura 19.

Figura 19. Modelo del dominio de IoT



Fuente: Autor.

En el dominio IoT, un usuario puede ser un humano o una entidad virtual. Este usuario puede invocar o suscribirse a uno o muchos servicios; y un servicio puede ser invocado o suscrito por cero o muchos usuarios. A su vez, un usuario interactúa con cero o muchas entidades físicas, y una entidad física puede ser interactuada por cero o muchos usuarios.

Un servicio expone cero o muchos recursos, y un recurso puede ser expuesto por cero o muchos servicios. Un recurso tiene información acerca de y/o actúa sobre cero o muchas entidades físicas, y una entidad física puede ser actuada por cero o muchos recursos.

Una entidad virtual representa una entidad en el mundo físico, y una entidad física puede ser representada por una o muchas entidades virtuales. Sobre las entidades físicas intervienen los dispositivos IoT, que pueden ser actuadores, etiquetas y sensores. Una entidad física puede ser actuada por cero o muchos actuadores, y un actuador puede actuar sobre cero o muchas entidades físicas. De igual forma, una entidad física puede ser identificada por cero o muchas

etiquetas, y una etiqueta puede identificar a cero o una entidad física. Una entidad física puede ser actuada por cero o muchos actuadores, y un actuador puede actuar sobre cero o muchas entidades físicas. Finalmente, una entidad física puede ser monitoreada por cero o muchos sensores, y un sensor puede monitorear a cero o muchas entidades físicas.

Los actuadores, etiquetas y sensores heredan características de una clase padre llamada Dispositivo. Un dispositivo puede albergar cero o muchos recursos en el dispositivo, y un recurso en el dispositivo ser alberga en un dispositivo. Los recursos sobre dispositivo al igual que los recursos sobre la red son una clasificación de recursos en el dominio IoT. Estas dos clases de recursos heredan características de la clase padre llamada Recurso.

A continuación, se definen cada uno de los términos claves que componen el modelo del dominio IoT propuesto:

Servicios. Un servicio es una entidad computacional independiente de la plataforma que se puede usar de una manera independiente de la plataforma (Bauer, Boussard, Bui, Carrez, et al., 2013, p. 348).

Entidades. Las entidades pueden ser de dos tipos:

- **Entidad física**. Una entidad física es una parte discreta e identificable del entorno físico que interesa al usuario para completar su objetivo y pueden ser casi cualquier objeto o entorno, incluyendo humanos o animales (Bauer, Boussard, Bui, Carrez, et al., 2013).

- **Entidad virtual**. Las entidades virtuales son artefactos digitales que representan entidades físicas; las entidades virtuales deben tener una única identificación que los identifica de manera unívoca (Bauer, Boussard, Bui, Carrez, et al., 2013). Las entidades virtuales puede ser aplicaciones de software, bases de datos y servicios.

Recursos. Los recursos son «componentes de software heterogéneos, generalmente específicos del sistema, que almacenan o procesan datos o información sobre una o más entidades físicas, o que proporcionan acceso a mediciones y actuaciones en el caso de sensores y actuadores, respectivamente» (Bauer, Boussard, Bui, Carrez, et al., 2013, p. 348).

- **Recursos en el dispositivo**. Los recursos en el dispositivo son típicamente recursos del sensor que proporcionan datos de detección o recursos del actuador; por lo tanto, se pueden ver como un "puente" entre el mundo digital y el físico. También pueden ser recursos de almacenamiento que están

limitados por la capacidad de almacenamiento del dispositivo (Bauer, Boussard, Bui, Carrez, et al., 2013).

- **Recursos en la red**. Este tipo de recursos no dependen de un hardware especial que permita una conexión directa con el mundo físico ya que se ejecutan en un servidor dedicado en la red o en la *cloud*. Los recursos en la red proporcionan servicios mejorados que requieren más recursos del sistema que los dispositivos típicos para IoT pueden proporcionar, tales como procesar datos y recursos de almacenamiento, que normalmente no sufren las limitaciones de sus contrapartes en el dispositivo (Bauer, Boussard, Bui, Carrez, et al., 2013).

Dispositivos. Un dispositivo es un componente físico técnico, hardware, con capacidades de comunicación a otros sistemas TIC. Un dispositivo se puede adjuntar o incrustar dentro de una entidad física, o monitorear una entidad física en su entorno (Bauer, Boussard, Bui, Carrez, et al., 2013, p. 343).

- **Sensores**. Un sensor se usa para determinar ciertas características físicas o químicas y transformarlas en una señal eléctrica para hacerlas procesables digitalmente, cerrando la brecha entre lo digital y lo físico (Smartex, 2016, p. 14). Los sensores proporcionan información, conocimiento o datos sobre la entidad física que supervisan conceptos (Bauer, Boussard, Bui, Carrez, et al., 2013).

- **Actuadores**. Los actuadores transforman las señales en diferentes formas de energía, como el movimiento o la presión, modificando el estado físico de una entidad física, como rotar o encender/apagar (Bauer, Boussard, Bui, Carrez, et al., 2013; Smartex, 2016).

- **Etiquetas**. Se usan para identificar entidades físicas, a las cuales las etiquetas generalmente están unidas físicamente, facilitando e incrementando la precisión del proceso de identificación (Bauer, Boussard, Bui, Carrez, et al., 2013).

Usuarios. Un usuario puede ser un humano o alguna entidad virtual que desee interactuar con un objeto físico particular (Bauer, Boussard, Bui, Carrez, et al., 2013, p. 348).

5.2 REPRESENTACIÓN DEL DOMINIO DE CIBERSERGURIDAD

En esta sección se presenta el modelamiento conceptual del dominio de la ciberseguridad para aplicaciones IoT, llamado *IoT Cybersecurity Domain Model* – IoT-CyDM. Para modelar los componentes de seguridad que integraron este

modelo conceptual se basó en la representación del dominio de ciberseguridad realizada en el *SMITH Model*, descrito en el capítulo 4 de este documento.

De igual forma como se procedió al modelar el dominio IoT, el primer paso para modelar el dominio de la ciberseguridad es definir los conceptos claves que intervienen en él y la relación entre ellos. Estos conceptos claves deben ser independientes de tecnologías específicas y de los casos de uso; además, que los conceptos no cambien mucho en las próximas décadas.

Esta sección está divida en dos subsecciones. En la subsección 5.2.1 se exponen los seis componentes de ciberseguridad propuestos para aplicaciones IoT. En la subsección 5.2.2 se modela el dominio de ciberseguridad para IoT y se define los componentes de ciberseguridad propuestos.

5.2.1 Componentes de ciberseguridad para IoT. Del *SMITH Model* se tomó los elementos de ciberseguridad representadas en la vista superior del modelo, compuesta por cuatro capas: confidencialidad, integridad, disponibilidad y no repudio.

Estos cuatro elementos de ciberseguridad contienen los requisitos de seguridad (RS) que atenderse para garantizar la seguridad de la información en una aplicación IoT. En los requisitos de seguridad se identificó los conceptos claves de seguridad, que se mencionan en la Tabla 26.

Tabla 26. Conceptos claves del dominio de ciberseguridad en IoT

Elemento de ciberseguridad	**Conceptos claves**
Confidencialidad	Acceso seguro, comunicación segura, medidas de seguridad y privacidad, control de activación, anonimato, control de datos expuestos, permisos de acceso, limitación de acceso, mecanismos de control, autenticación
Integridad	Integridad (de entidades virtuales, dispositivos, recursos y servicios)
Disponibilidad	Disponibilidad (de infraestructura, servicios, red) frescura de datos, accesibilidad
No repudio	Identificación única

Fuente: Autor.

Los conceptos claves mencionados en la Tabla 26 pueden ser atendidos en componentes que agrupen conceptos con características en común. Para la definición de estos componentes se basó en conceptos y mecanismos de seguridad ampliamente usados en la literatura técnica y científica. Estos

conceptos son la autenticación, autorización, disponibilidad, no repudio, cifrado de datos, gestión de identidad.

Durante la validación en la literatura de los conceptos anteriormente mencionados, se encontró que la autenticación, autorización, gestión de identidad, disponibilidad y no repudio son los términos adecuados y más genéricos para llamar a estos componentes. En cuando al cifrado de datos se encontró que hay un término más amplio y preciso que abarca todo el proceso del cifrado de los datos: La gestión de clave de cifrado o criptográficas[16] (Barker, 2016; Barker, Smid, Branstad, & Chokhani, 2013). La gestión de claves criptográficas implica dos cosas: (i) administrar del ciclo de vida de las claves criptográficas, que incluye: generar, usar, almacenar, archivar y eliminar claves; y (ii) proteger las claves de la pérdida o el uso indebido (Townsend Security, 2016).

De esta forma, el nombre definitivo de los componentes de ciberseguridad que integraron el modelo del dominio, son: Autenticación (AuthN), Autorización (AuthZ), gestión de claves criptográficas (CKM), gestión de identidad (IDM), Disponibilidad (AVBL) y No repudio (NRP).

Para validar que los estos componentes abordan las consideraciones de seguridad necesarias en una aplicación IoT se realizó una correspondencia entre requisitos de seguridad y los componentes propuestos, como se muestra en la Tabla 27. Para esta validación se utilizó los requisitos de seguridad considerados en el *SMITH Model*, que son: requisitos de seguridad (RS), requisitos de privacidad (RP), requisitos de autenticación y autorización (RAA), requisitos de integridad (RI), requisitos de disponibilidad (RD) y los requisitos de no repudio (RNP).

Tabla 27. Correlación entre requisitos y componentes de seguridad

Requisito	AuthN	AuthZ	CKM	IDM	AVBL	NRP
RS1: Un sistema IoT debe proporcionar a los usuarios el acceso seguro a los recursos.	X					
RS2: Un sistema IoT debe proporcionar comunicación segura y de confianza entre los dispositivos, los recursos y servicios, para evitar que terceros puedan espiarlas.			X			
RS3: Un sistema IoT debe proporcionar una medida de seguridad y privacidad del dispositivo.	X	X				
RS4: Un sistema IoT debe evitar que un dispositivo se active sin el consentimiento del propietario.				X		
RP1: Un sistema IoT debe proporcionar una alternativa para que los usuarios usen sus servicios de forma anónima.				X		
RP2: Un sistema IoT debe soportar la comunicación anónima entre dispositivos y proporcionar la confidencialidad de comunicación.			X	X		

[16] En inglés se usan indistintamente dos términos: el *Encryption Key Management* – EKM y el *Cryptographic Key Management* – CKM. En este trabajo se eligió usar el término gestión de clave criptográficas – CKM.

Requisito	AuthN	AuthZ	CKM	IDM	AVBL	NRP
RP3: Un sistema IoT debe permitir que los usuarios tengan el control de cómo sus datos están expuestos a otros usuarios.		X	X			
RP4: Un sistema IoT debe proporcionar la privacidad de la ubicación.			X			
RP5: Un sistema IoT debe proteger la privacidad de los usuarios que acceden a información sobre entidades o servicios físicos.			X	X		
RP6: Un sistema IoT debe evitar el seguimiento del identificador del dispositivo por entidades no autorizadas.		X		X		
RAA1: Un sistema IoT debe proporcionar diferentes permisos de acceso a la información.		X				
RAA2: Un sistema IoT debe admitir la limitación de acceso a la red a dispositivos específicos hasta que no presente credenciales apropiadas para unirse a la red.		X		X		
RAA3: Un sistema IoT debe apoyarse en mecanismos de control de acceso.	X	X				
RAA4: Un sistema IoT debe admitir la autenticación mutua de sujeto.			X	X		
RI1: Un sistema IoT debe proveer la integridad de la comunicación.			X			
RI2: Un sistema IoT debe validar la integridad de entidades virtuales, dispositivos, recursos y servicios.			X			
RD1: Un sistema IoT debe garantizar la disponibilidad de infraestructura.		X			X	
RD2: Un sistema IoT debe garantizar la disponibilidad de sus servicios.		X			X	
RD3: Un sistema IoT debe garantizar la disponibilidad de la red.		X			X	
RD4: Un sistema IoT debe garantizar la frescura de los datos.		X			X	
RD5: Los servicios IoT debe estar siempre accesible para los usuarios.		X			X	
RNP1: Un sistema IoT debe ofrecer una identificación única de los usuarios que solicitan datos a través de los servicios de descubrimiento/búsqueda.				X		
RNP2: Un sistema IoT debe apoyar la identificación de la ubicación del dispositivo.				X		
RNP3: Un sistema IoT debe garantizar el no repudio a nivel de recursos de red.				X		X

Fuente: Autor.

En la Tabla 27 se observa que los componentes de ciberseguridad considerados satisfacen los requisitos de seguridad que deben considerarse en una aplicación IoT. En la Tabla 28 se presentan los componentes de ciberseguridad propuestos, una breve descripción de su funcionalidad y los requisitos de seguridad consideraras en el *SMITH Model* que satisface dicho componente.

Tabla 28. Componentes de ciberseguridad propuestos

Nombre del componente de ciberseguridad	**Funcionalidad**	**Requisitos de seguridad considerados por *SMITH Model* que aborda**
Autenticación (AuthN)	Verifica la identidad asumida por un usuario (humano o entidad virtual), un servicio o recurso	• Confidencialidad de datos • Integridad de datos
Autorización (AuthZ)	Validación de permisos otorgados a usuario y entidades dentro del sistema	• Privacidad del servicio • Confidencialidad de datos • Integridad de datos

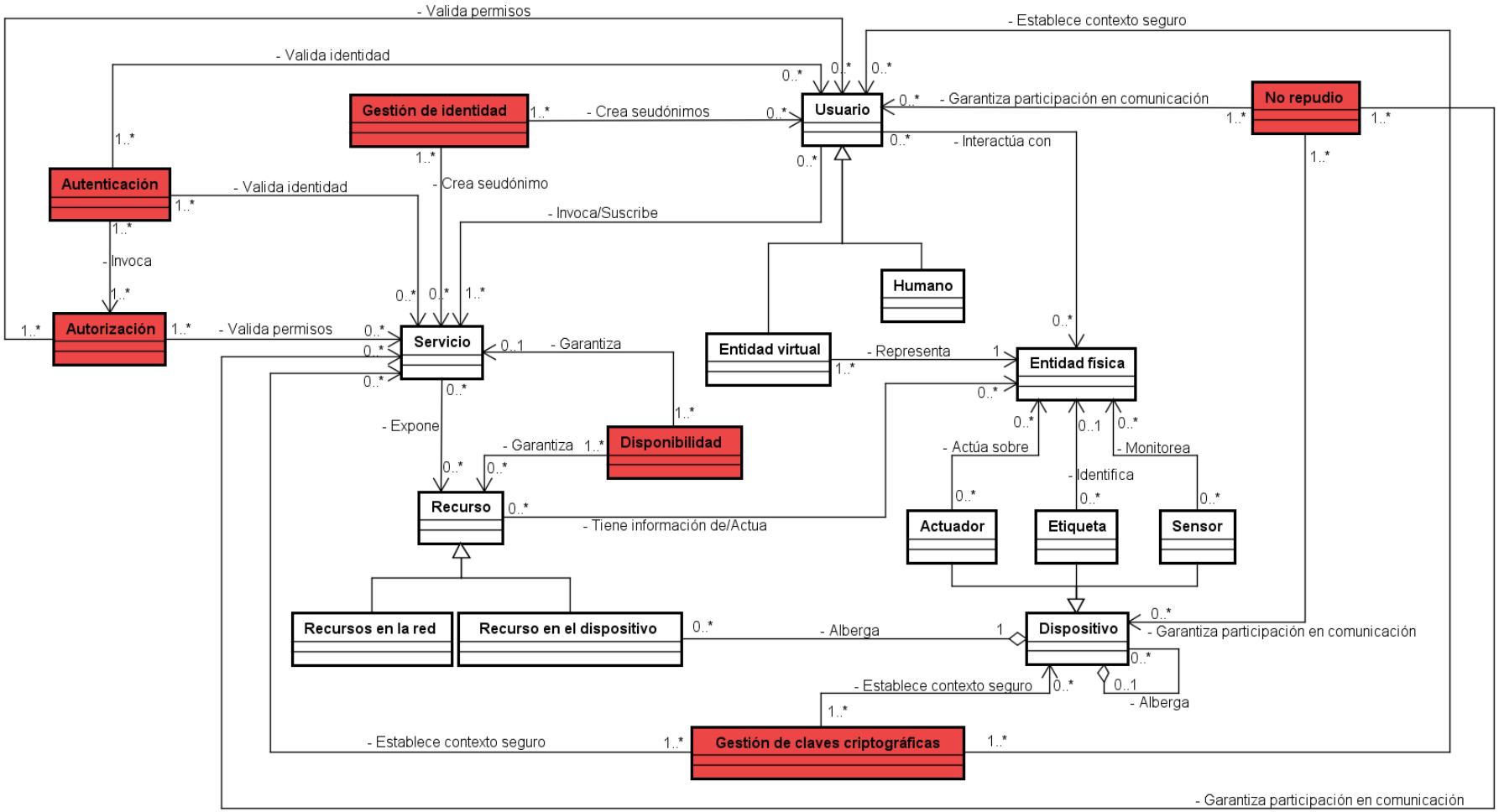
Gestión de clave criptográficas (CEM)	Administra el ciclo de vida de las claves y las protege de la pérdida o uso indebido	• Confidencialidad de comunicación • Integridad de comunicación
Gestión de identidad (IDM)	Gestión de identidades, seudónimos y políticas de acceso relacionadas	• Privacidad de usuario y servicio • No repudio en la comunicación
Disponibilidad (AVBL)	Provee políticas y controles para que la información esté disponible cuando se requiera	• Disponibilidad de servicios • Disponibilidad de recursos
No repudio (NRP)	Garantiza la participación de las partes en la comunicación y la comprobación del origen, destino y entrega del mensaje	• No repudio en la comunicación • No repudio a nivel de recursos de red

Fuente: Autor.

Estos seis componentes de ciberseguridad se usaron para modelar el dominio de ciberseguridad para aplicaciones IoT.

5.2.2 Modelo del dominio de ciberseguridad para IoT. El modelo de ciberseguridad propuesto, IoT-CyDM, está compuesto por seis componentes de ciberseguridad, como se muestra en la Figura 20.

Figura 20. IoT Cybersecurity Domain Model – IoT-CyDM.

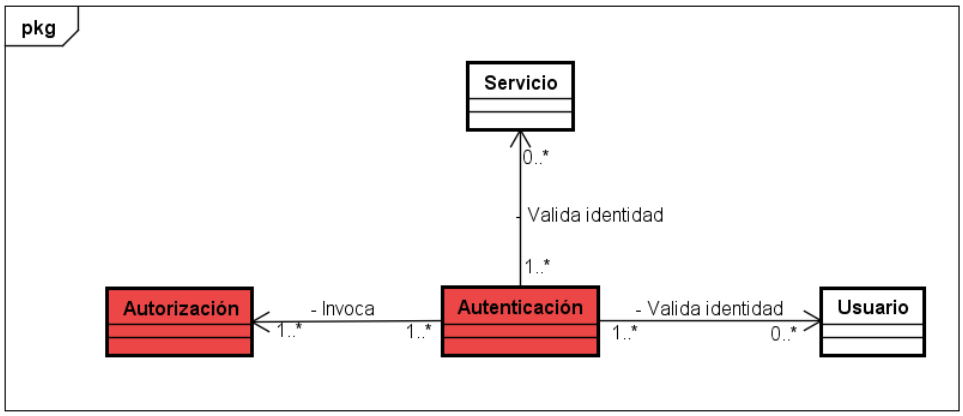


Fuente: Autor

5.2.2.1 Autenticación (AuthN). La autenticación es el proceso de verificar la identidad de un usuario, servicio o dispositivo para permitir el acceso a recursos restringidos. El proceso de autenticación puede realizarse utilizando métodos basados en algo conocido, como es una contraseña; métodos basados en algo que se posee, como un *token* o una tarjeta inteligente; o métodos basados en una característica física, como huellas, verificación de voz o escritura.

El componente AuthN tiene básicamente la funcionalidad de brindar un control de acceso basado en las credenciales proporcionadas por el usuario y los servicios, contribuyendo a la confidencialidad e integridad de los datos, como se ven en la Figura 21.

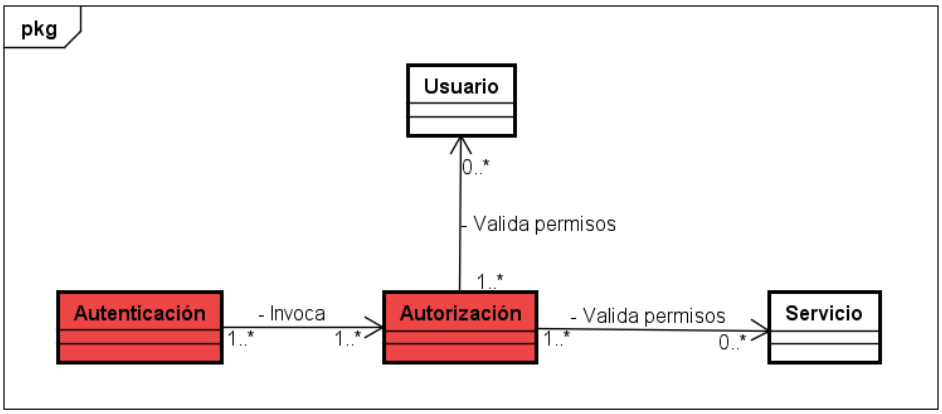
Figura 21. Relación del componente de Autenticación con los elementos de IoT



Fuente: Autor.

5.2.2.2 Autorización (AuthZ). La autorización hace referencia a la validación de los privilegios de acceso otorgados a un usuario o servicio o al acto de otorgar esos privilegios. La autorización complementa el proceso de autenticación, ya que una vez se verifica la identidad de un usuario se verifica qué privilegios posee sobre el sistema IoT y sus recursos. Este componente también de definen y gestionan las políticas de control de acceso. La relación de este componente con los elementos de seguridad se muestra en la Figura 22.

Figura 22. Relación del componente de Autorización con los elementos de IoT

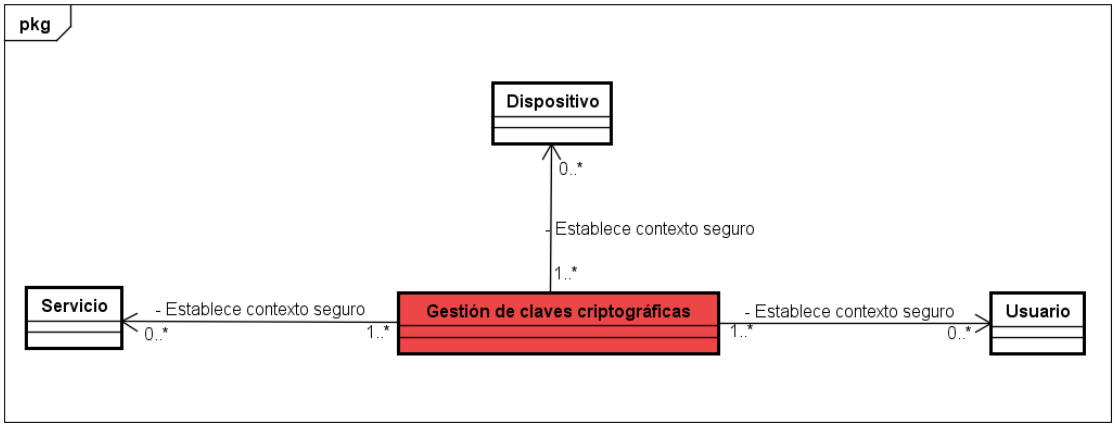


Fuente: Autor.

El componente AuthZ posee funcionalidades para la validación de los permisos del usuario, la administración de políticas de control de acceso y realizar decisiones de control de acceso basadas en dichas políticas. El componente AuthZ contribuye a la privacidad del servicio y la confidencialidad e integridad de los datos.

5.2.2.3 Gestión de claves criptográficas (CEM). El componente CEM permite comunicaciones seguras entre dos o más entidades que no tienen un conocimiento inicial entre sí o cuya interoperabilidad no está garantizada, asegurando la integridad y la confidencialidad. La relación de este componente con los elementos de seguridad se muestra en la Figura 23.

Figura 23. Relación del componente de Gestión de claves criptográficas con los elementos de IoT

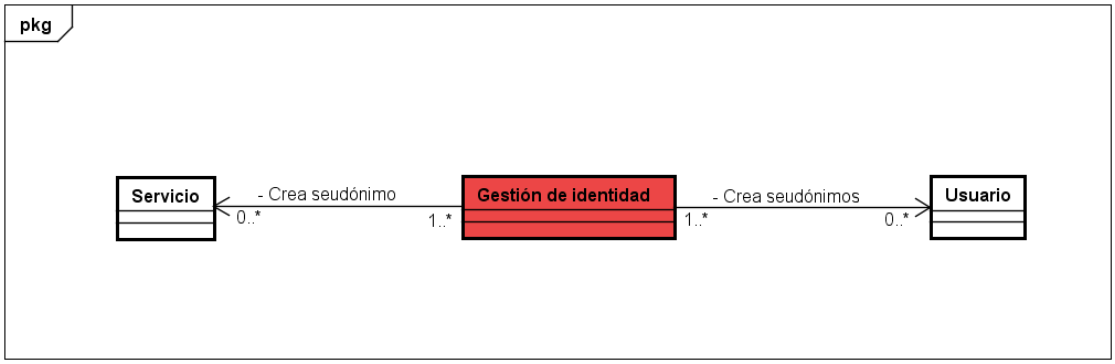


Fuente: Autor

De acuerdo con Serbanati *et al*. (2012), el componente de CEM tiene dos funciones: (i) distribuir las claves de forma segura: a petición, esta función descubre un marco de seguridad común soportado por el nodo emisor y un destino remoto, crea una clave (o par de claves) en este marco y luego las distribuye de forma segura. Se proporcionan parámetros de seguridad, incluido el tipo de habilitación de comunicaciones seguras; y (ii) registrar las capacidades de seguridad: los nodos y las puertas de enlace que desean beneficiarse de la mediación del CEM en el proceso de establecer conexiones seguras pueden hacer uso de la función de registrar las capacidades de seguridad. De esta forma, CEM registra sus capacidades y luego puede proporcionar claves en el marco correcto.

5.2.2.4 Gestión de identidad (IDM). La gestión de identidad aborda cuestiones de privacidad mediante la emisión y administración de seudónimos e información accesoria a sujetos de confianza para que puedan operar (usar o proporcionar servicios) de forma anónima. La relación de este componente con los elementos de seguridad se muestra en la Figura 24.

Figura 24. Relación del componente de Gestión de identidad con los elementos de IoT



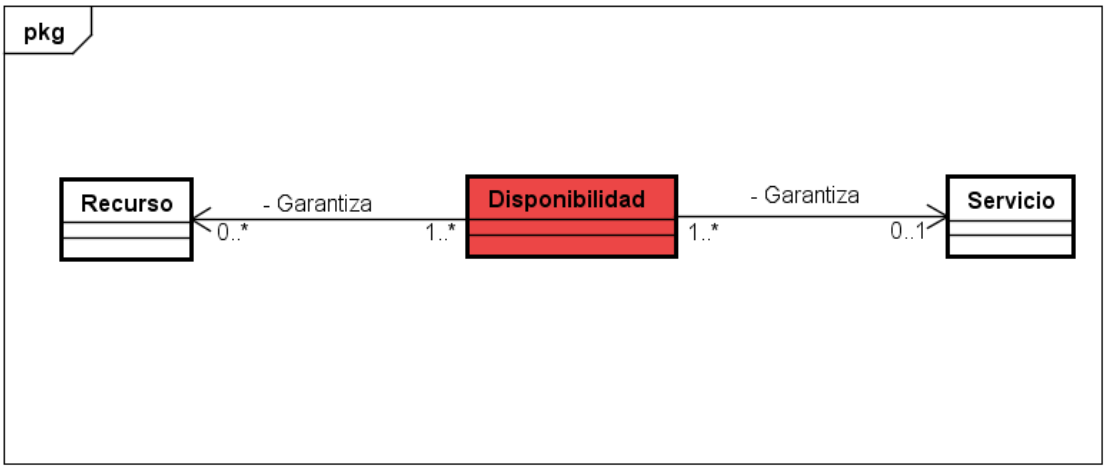
Fuente: Autor.

Su función es crear una identidad ficticia (identidad de raíz, identidad secundaria, pseudónimo o identidad de grupo) junto con las credenciales de seguridad relacionadas para que los usuarios y servicios utilicen durante el proceso de autenticación.

5.2.2.5 Disponibilidad (AVBL). Este componente provee las políticas y controles para garantizar que la información esté disponible cuando sea requerida. Para ello, se debe garantizar que los recursos que generan, transportan, almacenan, y presentan la información a los usuarios estén disponibles, y en caso de fallos, puedan recuperarse en el menor tiempo posible

sin una interrupción prologada del sistema. La relación de este componente con los elementos de seguridad se muestra en la Figura 25.

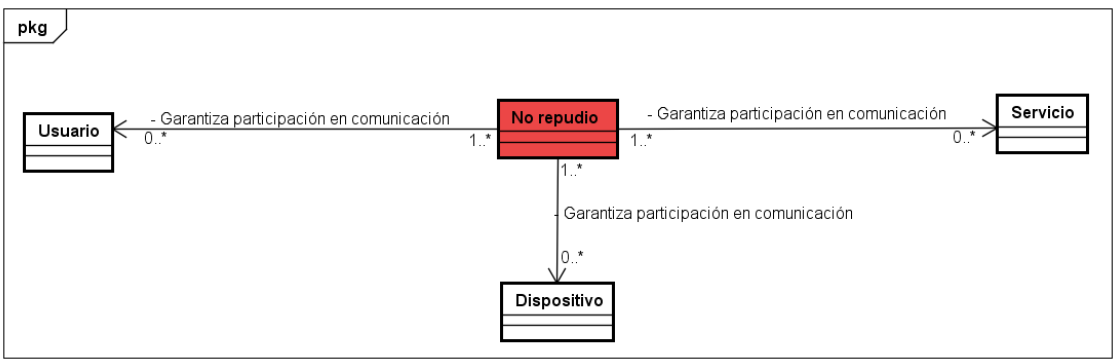
Figura 25. Relación del componente de Disponibilidad con los elementos de IoT.



Fuente: Autor.

5.2.2.6 No repudio (NRP). Este componente establece las condiciones para proteger la participación de alguna de las partes que intervienen en la comunicación, ya sea en toda o una parte de ella. Para ello, este componente garantiza que se pueda probar que el origen, el destino y la entrega del mensaje. De esta forma, en la prueba de origen el receptor puede demostrar ante terceros el origen de los datos recibidos; para la prueba de destino, el emisor de mensaje puede demostrar a terceros que los datos han sido entregados al emisor adecuado; finalmente, en la prueba de entrega, el emisor o receptor pueden demostrar ante terceros la fecha y hora en que se envió el mensaje. La relación de este componente con los elementos de seguridad se muestra en la Figura 26.

Figura 26. Relación del componente de No repudio con los elementos de IoT.



Fuente: Autor.

6. VALIDACIÓN DEL FRAMEWORK PROPUESTO

En este capítulo se presenta el proceso de validación del *framework* conceptual propuesto en este trabajo de investigación, llamado *SMITH Framework*. A su vez, este *framework* está compuesto por dos modelos, el *SMITH Model* y IoT-CyDM. El *SMITH Model* es una abstracción, representada a través de un cubo, del dominio de la ciberseguridad para aplicaciones IoT.

El modelo IoT-CyDM es una representación semiformal del *SMITH Model* a través de un lenguaje semiformal[17] para el describir arquitecturas de software, llamado lenguaje de modelamiento unificado – UML, por sus siglas en inglés, *Unified Modeling Language*. El UML es una especificación que define un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de los sistemas de objetos distribuidos (Object Management Group, 2005, 2017).

El objetivo de este capítulo fue validar si el *framework* propuesto podía orientar al equipo de desarrollo de aplicaciones IoT como cómo considerar la ciberseguridad en sus aplicaciones.

El resto del capítulo se divide de tres secciones. En la sección 6.1 se presenta el caso de estudio realizado para validar el *framework* propuesto. Este caso de uso consiste en el diseño de un sistema IoT para la seguridad física utilizando elementos de los modelos que integran el *framework*. En esta sección se explica la arquitectura conceptual y los requisitos del sistema IoT; también se realiza una presentación arquitectural del sistema compuesto por tres vistas: la vista conceptual, la vista funcional y la vista de servicios del sistema.

En la sección 6.2 se realiza la validación del diseño de la arquitectura de la aplicación IoT. El diseño de esta validación se basó en el método ATAM, un método para la evaluación temprana de arquitecturas de software basada en escenarios (Kazman et al., 1998). Finalmente, en la sección 6.3 se explica cómo se integran los dos modelos propuesto para conformar el *SMITH Framework*.

6.1 CASO DE ESTUDIO

El caso de estudio consiste en el diseño de un sistema de seguridad física basado en tecnologías IoT. Este sistema de seguridad física considera dos entornos: El entorno global y el entorno local de seguridad. El entorno global de

[17] Para más información sobre lenguajes de modelamiento consulte la subsección 2.1.5.2 de este documento.

seguridad hace referencia a los perímetros de un local o edificio y zonas exteriores. El entorno local de seguridad son las zonas de acceso restringido a zonas dentro del edificio o local (Oficina Nacional de Seguridad, 2016). El diseño del sistema de seguridad física se realizó teniendo en cuenta elementos del *SMITH Framework*.

El sistema IoT de seguridad física surge como una propuesta realizada a la Alcandía de Bucaramanga sobre como el IoT puede ayudar a la seguridad física de las instituciones públicas del municipio. Este caso de uso hace referencia a los Puntos Vive Digital (PVD) ubicados en Bucaramanga. Para determinar la arquitectura conceptual y los requisitos funcionales del sistema se hizo un levantamiento de información a través de revisión documental y una entrevista realizada al administrador de unos de los PVD para conocer el funcionamiento de estos.

En esta sección se presenta los alcances y limitaciones de caso de uso, la arquitectura conceptual del sistema, se describen sus requisitos funcionales y de calidad y se representa tres vistas arquitecturales del sistema: la vista conceptual, la vista funcional y la vista de servicios.

6.1.1 Alcance y limitaciones del caso de uso. Este caso de uso tiene dos objetivos: (i) validar si el *SMITH Framework* orienta al equipo de desarrollo de aplicaciones IoT sobre cómo considerar la ciberseguridad en aplicaciones IoT en la fase de diseño de estas; y (ii) validar que la arquitectura diseñada a partir del *SMITH Framework* atiende a los requisitos no funcionales relacionados con el atributo de calidad de seguridad. Lo anterior se fundamenta en que el objetivo de este trabajo de investigación es demostrar que se puede tomar medidas de seguridad desde la fase de diseño de una arquitectura de software. Por lo anterior, el proceso de desarrollo de la aplicación IoT llegó hasta la fase de diseño.

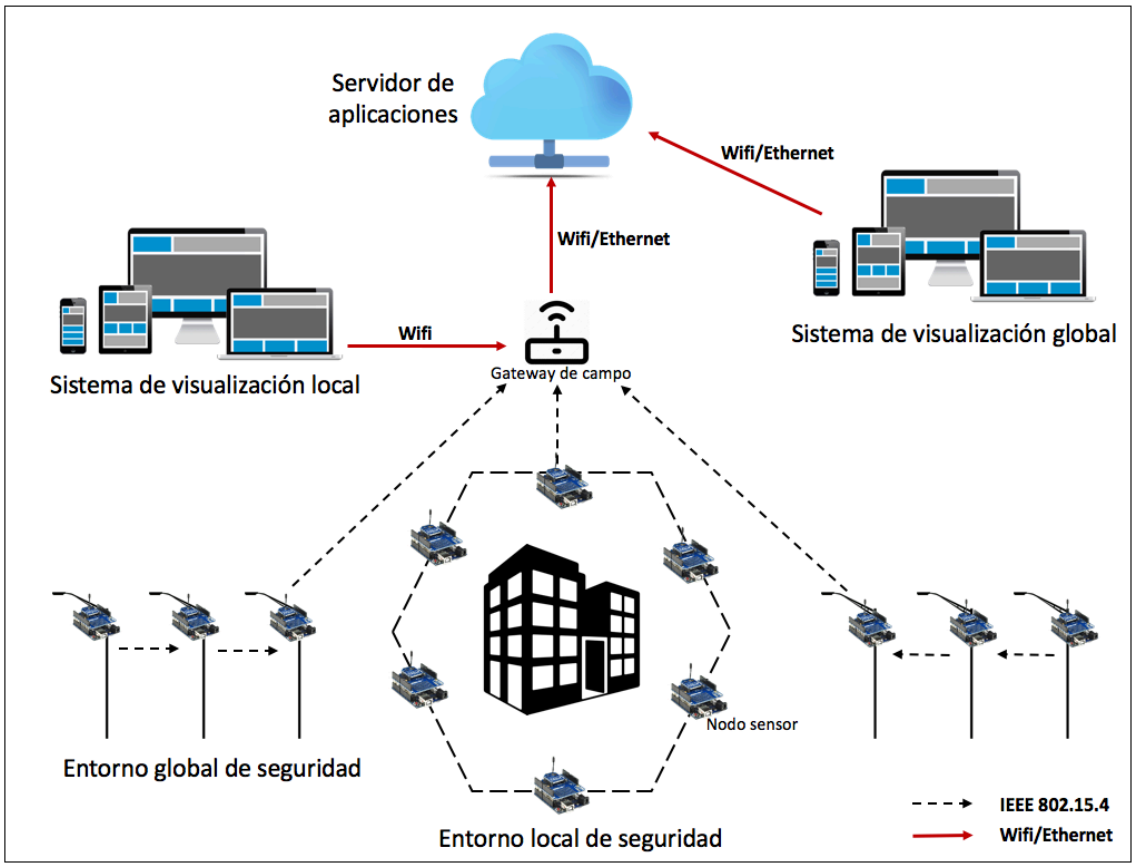
Otros aspectos del alcance y limitaciones del caso de uso son:

- Los elementos usados del *SMITH Model* fueron: (i) los requisitos de seguridad que debe considerar una aplicación IoT y (ii) la arquitectura genérica de aplicaciones IoT. Los elementos restantes del *SMITH Model* se usan en fases posteriores del proceso de desarrollo.
- Se usó el modelo IoT-CyDM para orientar en el diseño del modelo del dominio del sistema.

6.1.2 Arquitectura conceptual del sistema. La arquitectura conceptual del sistema IoT de seguridad perimetral se muestra en la Figura 27. Esta arquitectura se conforma por nodos sensores que integran los dos entornos de seguridad: el

global y el local; un *gateway* de campo que comunica estos entornos de seguridad con el servidor de aplicaciones y dos sistemas de visualización, uno local y otro global.

Figura 27. Arquitectura conceptual del sistema de seguridad perimetral



Fuente: Autor.

En la parte inferior de la arquitectura se encuentran los dos entornos de seguridad. El entorno global de seguridad está conformado por nodos sensores ubicados en las luminarias que detectan el nivel de luz natural o la presencia de personas para encender o apagarlas, así como en nodos con diversos tipos de sensores en la periferia del área física a proteger. El entorno local de seguridad está compuesto por nodos con sensores de presencia, presión, cercanía, biométricos entre otros.

Estos entornos de seguridad se comunican con el servidor de aplicaciones a través del *gateway* de campo. El *gateway* de campo tiene dos finalidades: la primera, es comunicar los entornos de seguridad con la capa superior de la aplicación, en el cual se procesa, almacena y se toman decisiones respecto a la aplicación y los eventos de seguridad, como es, por ejemplo, alertar sobre eventos de seguridad fuera de lo normal. La segunda finalidad es proveer de un

segundo sistema de visualización en caso de fallo en la conexión a internet. Ese sistema de visualización local provee de las funcionalidades básicas como almacenamiento de datos temporal, procesamiento de eventos y creación de alertas de seguridad.

En el servidor de aplicaciones se encuentra el software web que permite al administrador del sistema interactuar con el resto de la aplicación. Este software contiene módulos para el control de acceso, monitoreo y procesamiento de eventos, auditoria de las actividades del sistema y base de datos.

6.1.3 Requisitos del sistema. En esta subsección se presentan los requisitos funcionales (RF) y los requisitos no funcionales (RFN) relacionados con la calidad del sistema.

6.1.3.1 Requisitos funcionales. Los requisitos funcionales (RF) del sistema son:

- RF1: El sistema debe permitir gestionar roles de usuario (agregar, actualizar, eliminar y consulta) y asignarles los permisos correspondientes.
- RF2: El sistema debe permitir gestionar cuentas de usuario (agregar, actualizar, eliminar y consulta).
- RF3: El sistema debe permitir gestionar zonas físicas (agregar, actualizar, eliminar y consulta).
- RF4: El sistema debe permitir la gestión de nodos sensores (agregar, actualizar, eliminar y consulta) y su asignación a una zona física.
- RF5: El sistema debe permitir la autenticación de usuarios para el acceso a áreas restringidas.
- RF6: El sistema debe mostrar, en una vista de interfaz de usuario, el estado de las luminarias (encendido-apagado y funcionado-averiado) en tiempo real.
- RF7: El sistema debe permitir la consulta de eventos en el sistema.
- RF8: El sistema debe alertar al administrador del sistema y personal de seguridad en el momento en que los sensores detecten un cambio de las condiciones establecidas como normales.
- RF9: El sistema debe permitir configurar el envío de datos de los nodos sensores de forma asíncrona (periódico o basado en eventos) y síncrona (Reporte de datos por consulta).
- RF10: El sistema debe tener un módulo de auditoria donde se registre cada acción realizada en el sistema.
- RF11: El sistema debe permitir la generación de reportes sobre las actividades realizadas en el sistema y los inconvenientes de seguridad.

6.1.3.2 Requisitos de calidad. Para identificar los atributos de calidad del sistema se basó en la ISO 25010:2011 “Calidad del producto de software”. Para el atributo de calidad se tomó como base la lista de requisitos de seguridad ofrecido por el *SMITH Model*. A continuación, se presentan los requisitos no funcionales (RNF) agrupados según los atributos de calidad considerados en la norma:

Adecuación funcional

- RNF1: El sistema debe realizar las notificaciones con una tasa menor a un segundo.
- RNF2: El sistema debe tener una tasa lo más cercana a cero de falsos positivos y falsos negativos.

Usabilidad

- RNF3: El sistema debe contar con una baja curva de apropiación por parte de los usuarios.
- RNF4: El sistema debe ser operado y controlado de forma fácil por los usuarios, minimizar el número de interacciones para realizar una tarea.
- RNF5: El sistema debe contar con la validación de datos al ingresar datos al sistema para proteger el correcto funcionamiento del sistema por errores humanos.
- RNF6: El sistema debe mostrar la información de forma concisa y clara para reducir el tiempo de respuesta ante una notificación de alarma.

Fiabilidad

- RNF7: El sistema debe monitorizar las 24 horas y 7 días de la semana.
- RNF8: El sistema debe estar en la capacidad de auto configurar la red de sensores inalámbrica que se utilice para la monitorización.

Seguridad

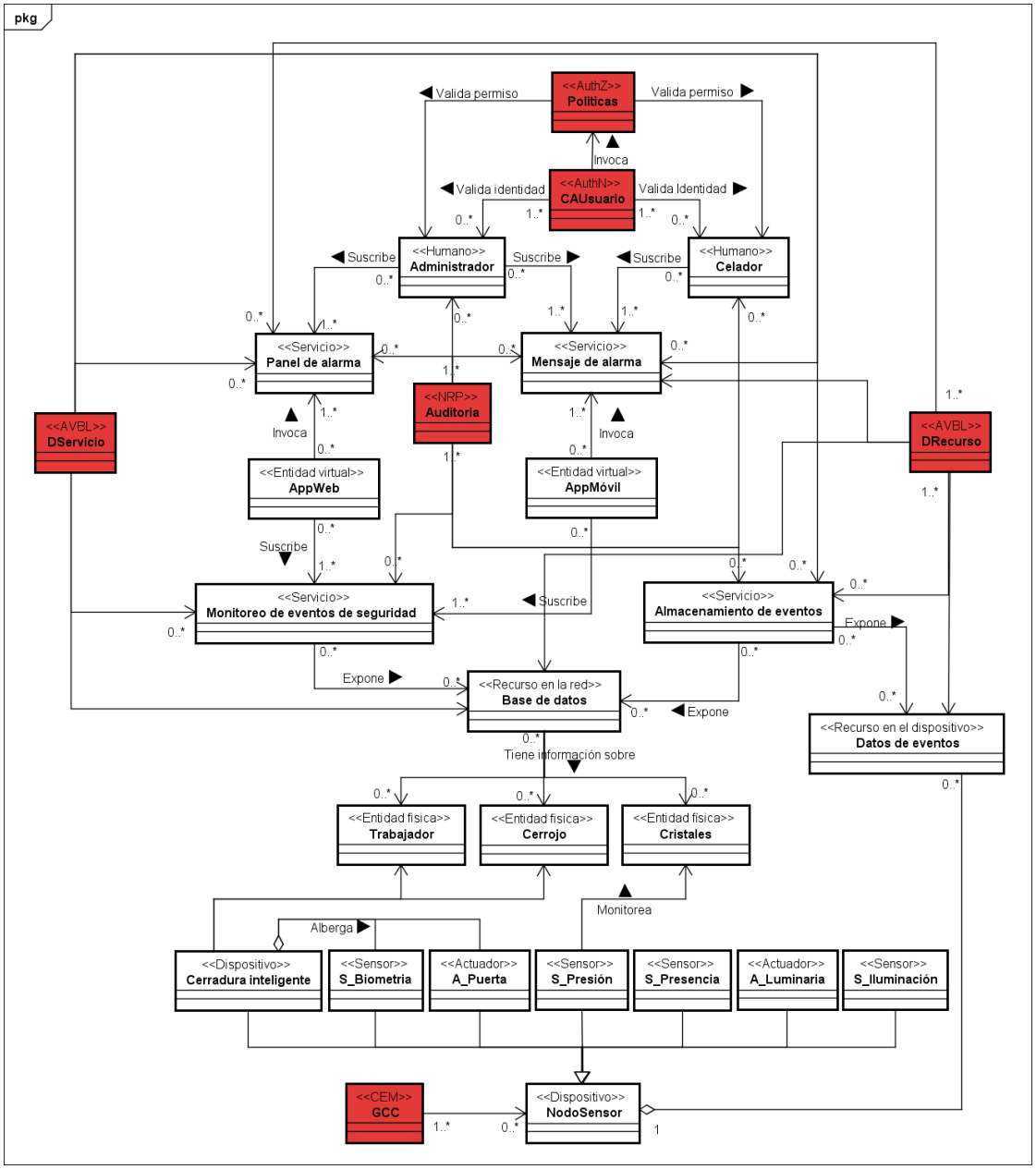
- RNF9: El sistema debe enviar los datos de forma cifrada para garantizar la confidencialidad de la información.
- RNF10: El sistema debe ofrecer mecanismos de autenticación y control de acceso.
- RNF11: El sistema debe permitir la asignación de roles para gestionar los controles de acceso y sus respectivos permisos en el sistema para garantizar la integridad de la información.

- RNF12: El acceso al sistema debe estar restringido por el uso de claves asignadas a cada uno de los usuarios y debe contar con doble factor de autenticación.
- RNF13: El sistema debe garantizar la participación de las partes en las comunicaciones realizadas en el sistema, permitiendo que se pueda probar el origen, el destino y la entrega del mensaje.

6.1.4 Presentación arquitectural del sistema. Para la presentación arquitectural del sistema IoT de seguridad perimetral se consideraron tres vistas arquitecturales: la vista conceptual, la vista funcional y la vista de servicios.

6.1.4.1 Vista conceptual. En esta vista se presenta a través de un diagrama de clases UML, como se ve en la Figura 28. Esta vista conceptual se basada en el modelo del dominio de ciberseguridad para aplicaciones IoT, IoT-CyDM, descrito en el capítulo 5 de este documento.

Figura 28. Vista conceptual del sistema



Fuente: Autor.

6.1.4.2 Vista funcional. Este punto de vista muestra detalles del sistema como los componentes de software y las interfaces de comunicación entre ellos. En la Figura 29 se muestra la vista funcional del sistema representada a través de un diagrama de componentes UML.

La vista funcional está estructurada en tres capas, siguiendo como base las funcionalidades genéricas de aplicaciones IoT consideradas en la arquitectura genérica propuesta por el *SMITH Model*.

En *Cloud Layer* se muestran los componentes relacionados con la gestión de servicios, gestión de eventos, almacenamiento y visualización. Los componentes relacionados con la gestión de servicios están el de control de acceso, auditoria y almacenamiento de eventos; en cuento a la gestión de eventos, el sistema considera el componente de monitorización de eventos. Los componente relacionados con la funcionalidad de visualización son: la apliación web y la aplicación móvil.

La *Fog Layer* comprende componentes con las funcionalidades de comunicación, gestion de fallos, gestión de topologías de red y agregación de datos. Estos componentes se relacionan con componentes de la capa superior e inferior, y en general, la función de esta capa es comunicar estas dos capas y garantizar que esa comunicación no se interrumpa.

Y finalmente, en la *Dew Layer* se muestran los componentes relacionados con la generación de datos, la gestión de dispositivos y el procesamiento de datos. Los componentes relacionados con la generación de datos se encuentra el sensor y el actuador. Estos dos junto con el componente del procesamiento integran el componente de nodo sensor el cual se relaciona con el componente de configuración de dispositvo y otros componentes de *Fog Layer*.

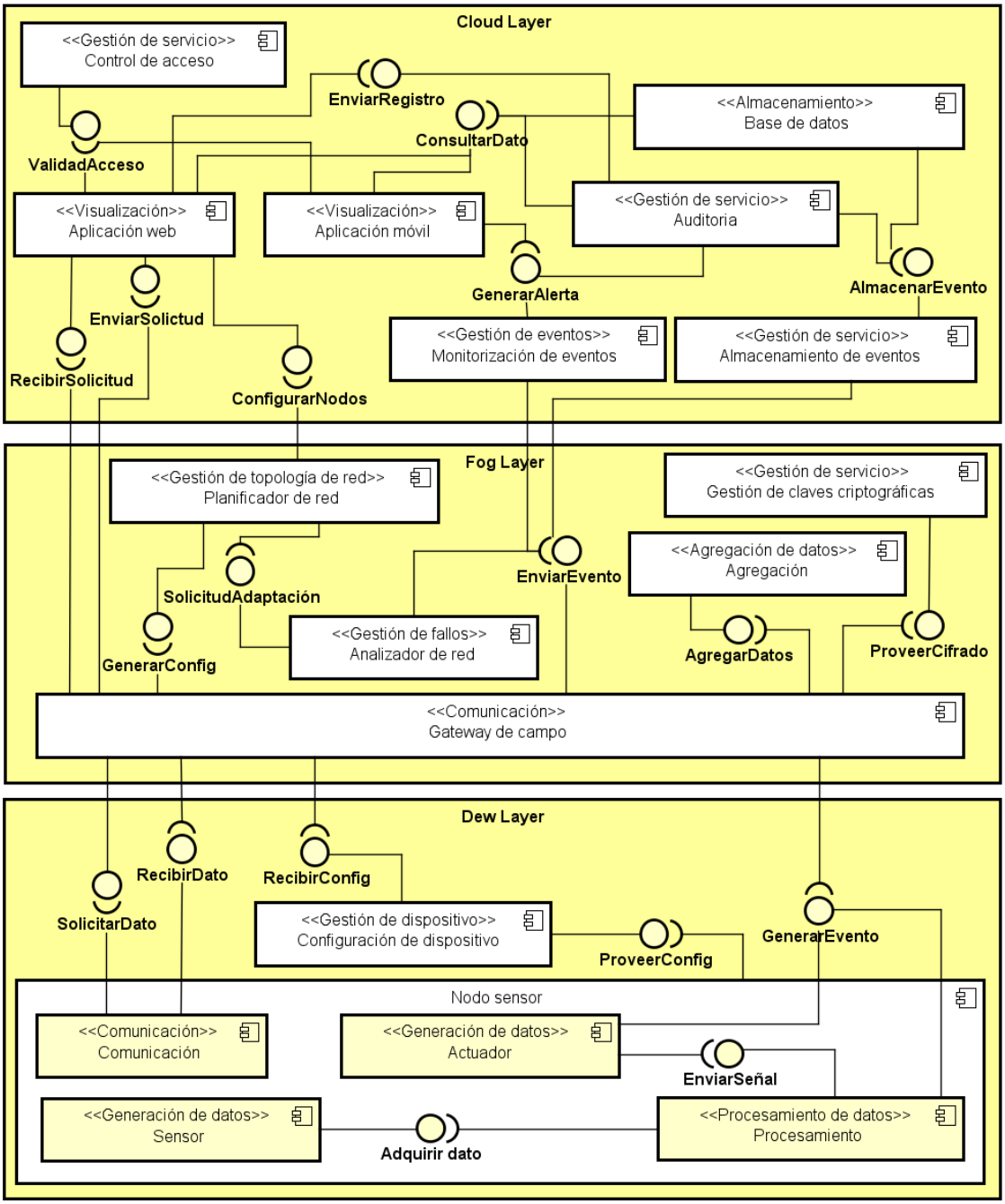
En la Tabla 29 se muestran los componentes del sistema consideradas en la vista funcional.

Tabla 29. Componentes del sistema

Capa	ID	Componente	Funcionalidad
Cloud Layer	C1	Control de acceso	Gestión de servicios
	C2	Aplicación web	Visualización
	C3	Aplicación móvil	Visualización
	C4	Base de datos	Almacenamiento
	C5	Auditoria	Gestión de servicios
	C6	Monitorización de servicios	Gestión de servicios
	C7	Almacenamiento de servicios	Gestión de servicios
Fog Layer	C8	Planificador de red	Gestión de topología de red
	C9	Gestión de claves criptográficas	Gestión de servicios
	C10	Agregación	Agregación de datos
	C11	Analizador de red	Gestión de fallos
	C12	Gateway de campo	Comunicación
Dew Layer	C13	Configuración de dispositivo	Gestión de dispositivo
	C14	Comunicación	Comunicación
	C15	Procesamiento	Procesamiento de datos
	C16	Sensor	Generación de datos
	C17	Actuador	Generación de datos

Fuente: Autor.

Figura 29. Vista funcional del sistema



Fuente: Autor.

6.1.4.3 Vista de servicios del sistema. La vista de servicios del sistema identifica a un alto nivel de abstracción los servicios mínimos que deben implementarse para la implementación del sistema IoT de seguridad perimetral. La vista de servicios se presenta en la Tabla 30.

Tabla 30. Vista de servicios del sistema

Servicio	Descripción	Componente	Requisito relacionado
ValidarAcceso	Valida la identidad y los permisos otorgados al usuario o servicio	Control de acceso	RNF10, RNF11, RNF12
EnviarRegistro	Provee los registros de las actividades realizadas por los usuarios	Auditoria	RNF13
ConsultarDato	Envía consultas a la base de datos	Aplicación web Aplicación móvil	RNF3, RNF4, RNF5, RNF6
AlmacenarEvento	Envía los datos generados por los eventos en la base de datos para su registro	Almacenamiento de eventos	RFN13
GenerarAlerta	Crea una alerta ante un evento de seguridad	Monitorización de eventos	RNF1, RNF2
EnviarSolicitud	Solicita datos de los nodos sensores útil para la aplicación o los usuarios	Aplicación web	RNF1
RecibirSolicitud	Recibe datos de los nodos sensores útil para la aplicación o los usuarios	Aplicación web	RNF1
ConfigurarNodos	Envía las configuraciones de los nodos sensores	Aplicación web	RNF1
AgregarDatos	Realiza la agregación de los datos antes de enviarlos a la nube	Agregación	RNF1
ProveerCifrado	Gestiona claves criptográficas y provee cifrado a las comunicaciones	Gestión de datos criptográficos	RNF9
EnviarEvento	Envía eventos de seguridad o fallos de la red	Gateway de campo	RNF1
SolicitudAdaptación	Genera solicitudes de adaptación de la red de nodos sensores	Analizador de red	RNF7, RNF8
GenerarConfig	Genera configuración de red para los nodos sensores	Planificador de red	RNF7, RNF8
SolicitarDato	Solicita datos a los nodos sensores como información de entidades físicas o estado de estos	Gateway de campo	RNF6
RecibirDato	Recibe datos a los nodos sensores como información de entidades físicas o estado de estos	Nodo sensor	RNF6
RecibirConfig	Recibe la configuración de dispositivo	Configuración de dispositivo	RNF7, RNF8
ProveerConfig	Envía la configuración del dispositivo a los nodos sensores	Configuración de dispositivo	RNF7, RNF8
GenerarEvento	Crea un evento de seguridad	Procesamiento	RNF1
EnviarSeñal	Envía señal para que el actuador actúe sobre la entidad física	Procesamiento	RNF1, RNF6
AdquirirDato	El sensor toma datos de la entidad física que monitorea	Nodo sensor	RNF1, RNF6

Fuente: Autor.

6.2 VALIDACIÓN DE LA ARQUITECTURA

La validación de la arquitectura propuesta en el caso de estudio se fundamentó en el *Architecture Tradeoff Analysis Method* – ATAM (Kazman et al., 1998), el cual es un método para la evaluación temprana de arquitecturas de software

basada en escenarios. El propósito de ATAM es evaluar las consecuencias de las decisiones arquitectónicas en relación con determinados atributos de calidad, permitiendo establecer si una arquitectura particular satisface los atributos de calidad.

Esta validación considera el proceso descrito por ATAM, que consta de nueve pasos agrupados en cuatro fases: (i) presentación, (ii) investigación y análisis, (iii) pruebas y (iv) presentación de informe. A continuación, se presenta la validación realizada a través de las nueve actividades del método ATAM.

6.2.1 Fase 1: Presentación. Esta fase se compone de tres pasos:

6.2.1.1 Paso 1: Presentación de ATAM. En esta prueba se usó ATAM para validar cómo la arquitectura propuesta considera los componentes de ciberseguridad representados en el modelo del dominio de ciberseguridad propuesto en este trabajo de investigación. Por esta razón, en esta validación sólo se tuvo en cuenta los atributos de calidad relacionados con la seguridad.

6.2.1.2 Paso 2: Presentación de los objetivos del negocio. El objetivo de realizar esta validación es verificar la seguridad otorgada a la arquitectura diseñada a partir del modelo de ciberseguridad para aplicaciones IoT, cuyo propósito es facilitar la implementación de componentes de ciberseguridad.

6.2.1.3 Paso 3: Presentación de la arquitectura. El diseño de la arquitectura para el sistema IoT de seguridad perimetral se basó en dos componentes del *SMITH Framework:* el modelo del dominio de ciberseguridad, IoT-CyDM, y la arquitectura genérica para aplicaciones IoT propuesta en el *SMITH model*.

A un nivel conceptual, la arquitectura consta de un servidor de aplicaciones, dos sistemas de visualización, uno global y otro local, y nodos sensores agrupados en dos entornos de seguridad: el entorno global, que hace referencia a los perímetros de la zona física a proteger, y el entorno local, que hace referencia a la zona a proteger. En la sección 6.1.3 de este documento se detalla la presentación arquitectural del sistema.

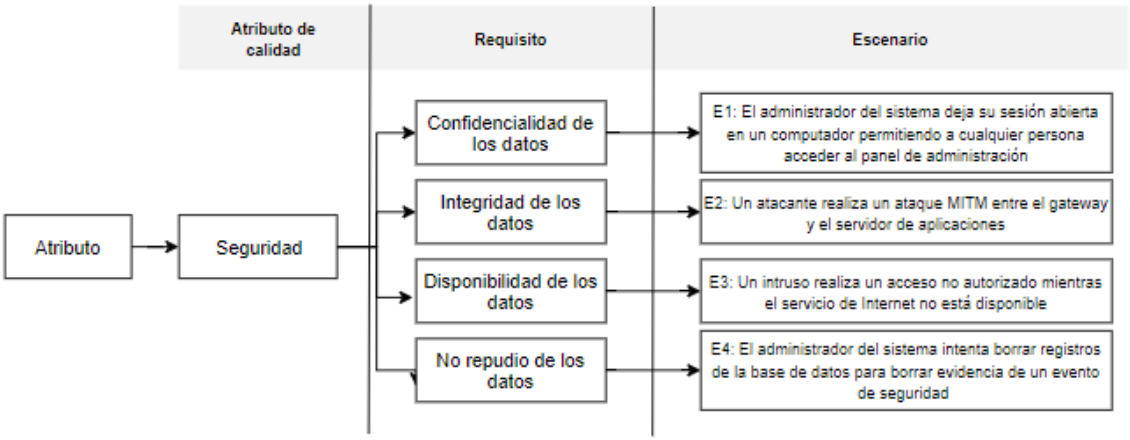
6.2.2 Fase 2: Investigación y análisis. Esta fase se compone de tres pasos:

6.2.2.1 Paso 4: Identificar las aproximaciones arquitecturales. El patrón arquitectónico usado en este sistema es en capas o *layers*. Este patrón ayuda a estructurar las funcionalidades del sistema componentes agrupados en capas específicas. La comunicación entre componentes se realiza entre capas vecinas

y a través de interfaces bien definidas. En la Figura 29 se muestra en detalle la arquitectura, sus componentes y sus interfaces.

6.2.2.2 Paso 5: Generar el árbol de utilidad de atributos de calidad. Un árbol de utilidad es un instrumento para para identificar, priorizar y refinar las metas de los atributos de calidad más importantes del sistema. De esta forma, la evaluación se concentra en los aspectos que son más críticos para el éxito del sistema. En la Figura 30 se muestra el árbol de utilidad con los requisitos de seguridad a evaluar y los escenarios de prueba.

Figura 30. Árbol de utilidad de atributos de calidad



Fuente: Autor.

6.2.2.3 Paso 6: Analizar las aproximaciones arquitecturales. Las entradas de este paso son los requisitos de calidad concretados en el paso 5 y las propuestas arquitectónicas utilizados en la arquitectura, del paso 4, midiéndose cuán adecuados son el uno para el otro. La meta de este paso es convencerse de que la propuesta instanciada de la arquitectura es la apropiada para satisfacer los requisitos del atributo de seguridad.

6.2.3 Fase 3: Pruebas. Esta fase se compone de dos pasos:

6.2.3.1 Paso 7: Lluvia de ideas y priorización de escenarios. En la lluvia de ideas se decidió no generar nuevos escenarios ni eliminar ninguno de los cuatro escenarios generados en los pasos anteriores, por esta razón se decidió analizarlos todos.

6.2.3.2 Paso 8: Analizar las aproximaciones arquitecturales. En esta validación se midió, a través de los escenarios, si la arquitectura instanciada

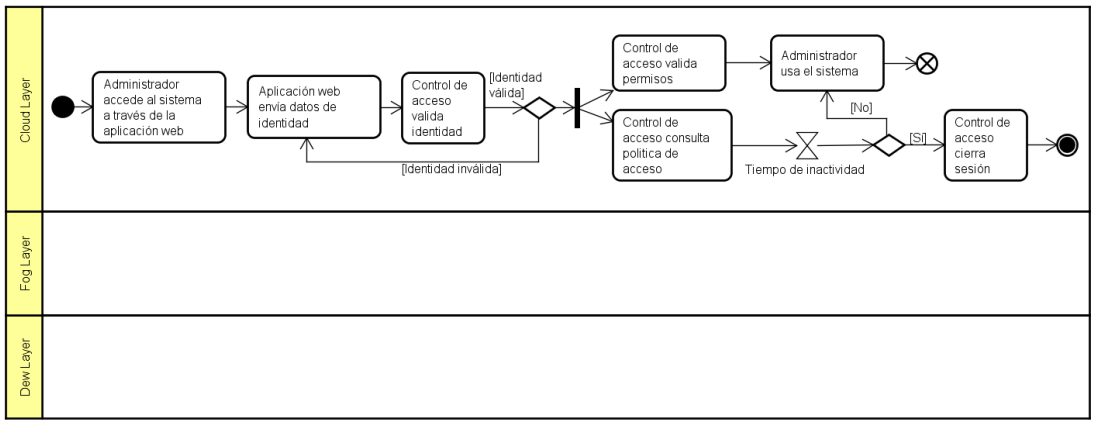
satisface los cuatro requisitos del atributo de seguridad considerados en el árbol del problema. Esta validación se hizo a través diagramas de actividad UML, que son un tipo de diagramas de comportamiento, que describe la funcionalidad del sistema en cierto escenario donde la interacción entre dos o más componentes es necesaria. A continuación, se presenta la validación de los escenarios propuestos:

Escenario 1.

En este escenario el administrador del sistema deja su sesión abierta en un computador permitiendo a cualquier persona acceder al panel de administración. El requisito por validar en este escenario es la confidencialidad de los datos. En este escenario intervienen los componentes de aplicación web y control de acceso.

En la *Figura 31* se representa el proceso realizado por el sistema durante el escenario propuesto.

Figura 31. Validación del escenario 1 - Confidencialidad de los datos



Fuente: Autor.

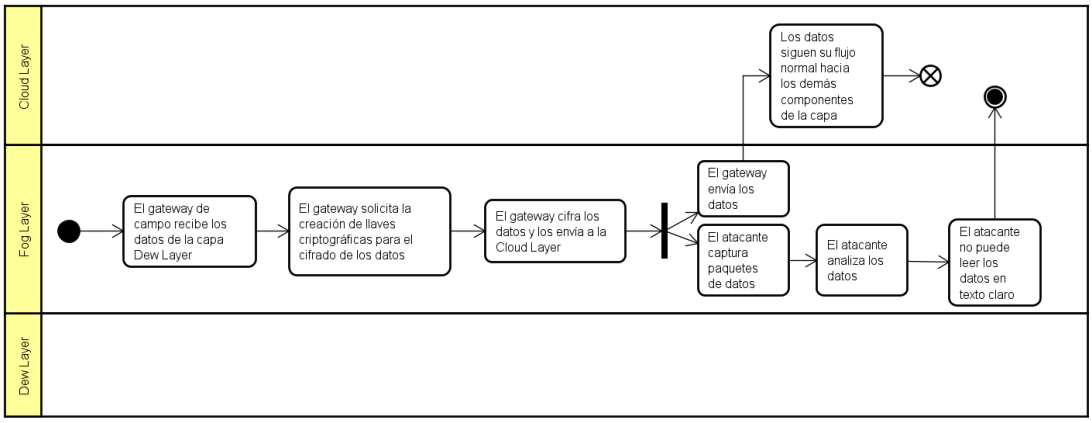
El componente de control de acceso implementa los componentes de autenticación y autorización considerados en el modelo IoT-CyDM. Este componente no solo valida la identidad y los permisos del usuario, sino que también define políticas asociadas al control de acceso como el tiempo de inactividad del sistema para cerrar la sesión de este.

Escenario 2.

En este escenario un atacante realiza un ataque de MITM[18] entre el *gateway* y el servidor de aplicaciones. El requisito por validar en este escenario es la integridad de los datos. En este escenario intervienen los componentes de *gateway* de campo y gestión de claves criptográficas.

En la *Figura 32* se representa el proceso realizado por el sistema durante el escenario propuesto.

Figura 32. Validación del escenario 2 - Integridad de los datos



Fuente: Autor.

El cumplimiento con el requisito de la integridad de los datos se debe a la incorporación del componente de gestión de claves criptográficas, el cual permite las comunicaciones seguras entre dos o más entidades que no tienen un conocimiento inicial entre sí o cuya interoperabilidad no está garantizada, asegurando la integridad y la confidencialidad de los datos[19].

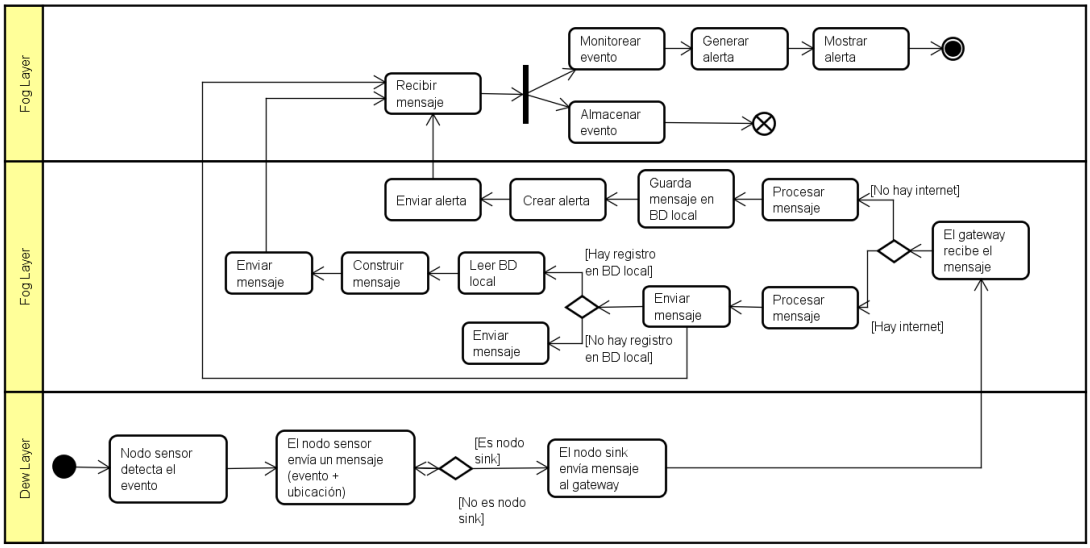
Escenario 3.

En este escenario, un intruso realiza un acceso no autorizado mientras el servicio de internet no está disponible. El requisito por validar en este escenario es la disponibilidad de los datos. En este escenario intervienen los componentes de sensor, procesamiento, *gateway* de campo, monitoreo de eventos, almacenamiento de eventos, base de datos, aplicación web y aplicación móvil.

[18] El ataque de hombre en el medio – MITM, *Man in the Middle*, en el que se adquiere la capacidad de leer, insertar y modificar mensajes, al introducirse en la comunicación entre dos partes.
[19] Funcionalidad descrita en el modelo IoT-CyDM presentado en el capítulo 5 de este documento.

En la Figura 33 se representa el proceso realizado por el sistema durante el escenario propuesto.

Figura 33. Validación del escenario 3 - Disponibilidad de los datos



Fuente: Autor.

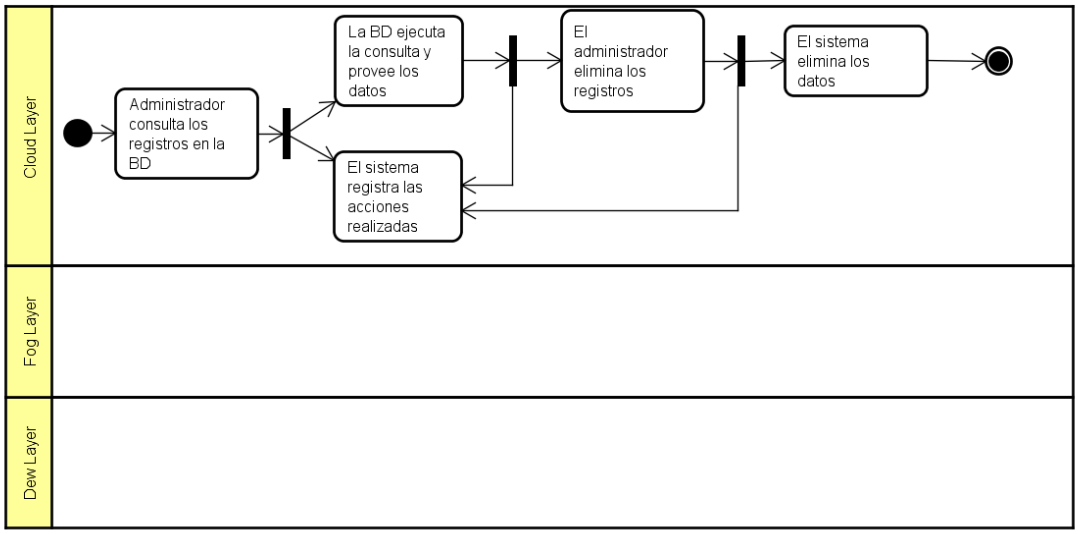
En este escenario se puede verificar que el componente de *gateway* de campo, aprovecha las capacidades del dispositivo en el que se aloja, no solo para comunicar la capa inferior con la superior, sino para almacenar y procesar información de forma local sin necesidad de ir al servidor de aplicación ubicado en la *Cloud Layer*. Esa medida de redundancia del sistema de visualización permite que el sistema siga su funcionamiento mientras se reestablece el servicio de internet, permitiendo así que la seguridad física no se vea afectada en cuando a la disponibilidad de los datos.

Escenario 4.

En este escenario el administrador del sistema elimina registros de la base de datos para borrar evidencia de un evento de seguridad. El requisito por validar en este escenario es no repudio de los datos. En este escenario intervienen los componentes de aplicación web, la base de datos y el componente de auditoria.

En la Figura 34 se representa el proceso realizado por el sistema durante el escenario propuesto.

Figura 34. Validación del escenario 4 - No repudio de los datos



Fuente: Autor.

En este escenario el requisito de seguridad relacionado con garantizar el no repudio de la información se satisface gracias a que la arquitectura diseñada incorpora un componente de auditoria que registra las acciones realizada por los usuarios y otras entidades del sistema, garantizando que se compruebe la acción de uno de ellos y su participación en la comunicación de alguna de las partes de ella.

6.2.4 Fase 4: Presentación de informe. El objetivo de esta validación era verificar que la arquitectura del sistema IoT de seguridad física descrita en el caso de estudio cumplía con los requisitos de seguridad. Esta arquitectura fue diseñada a partir del modelo IoT-CyDM, cuyo propósito es guiar a los desarrolladores de aplicaciones IoT, desde las buenas prácticas de la ingeniería del software, en la implementación de componentes de ciberseguridad en la fase de diseño de sus aplicaciones.

En la Tabla 31 y Tabla 32 se validó el cumplimiento de los requisitos funcionales y de calidad que debía atender la arquitectura diseñada en el caso de estudio. Los componentes se identifican con los ID asignados a cada uno de ellos en la Tabla 29.

Tabla 31. Validación de la atención de los requisitos funcionales

ID	Requisito	Estado	Componentes
RF1	El sistema debe permitir gestionar roles de usuario (agregar, actualizar, eliminar y consulta) y asignarles los permisos correspondientes.	Atendido	C2, C4
RF2	El sistema debe permitir gestionar cuentas de usuario (agregar, actualizar, eliminar y consulta).	Atendido	C2, C4

ID	Requisito	Estado	Componentes
RF3	El sistema debe permitir gestionar zonas físicas (agregar, actualizar, eliminar y consulta).	Atendido	C2, C4
RF4	El sistema debe permitir la gestión de nodos sensores (agregar, actualizar, eliminar y consulta) y su asignación a una zona física.	Atendido	C2, C4
RF5	El sistema debe permitir la autenticación de usuarios para el acceso a áreas restringidas.	Atendido	C2, C4
RF6	El sistema debe mostrar, en una vista de interfaz de usuario, el estado de las luminarias (encendido-apagado y funcionado-averiado) en tiempo real.	Atendido	C2, C4
RF7	El sistema debe permitir la consulta de eventos en el sistema.	Atendido	C2, C4, C7, C12, C13, C14, C15, C16, C17
RF8	El sistema debe alertar al administrador del sistema y personal de seguridad en el momento en que los sensores detecten un cambio de las condiciones establecidas como normales.	Atendido	C3, C6, C12, C14, C15, C16, C17
RF9	El sistema debe permitir configurar el envío de datos de los nodos sensores de forma asíncrona (periódico o basado en eventos) y síncrona (Reporte de datos por consulta).	Atendido	C2, C8, C12, C13
RF10	El sistema debe tener un módulo de auditoria donde se registre cada acción realizada en el sistema.	Atendido	C5
RF11	El sistema debe permitir la generación de reportes sobre las actividades realizadas en el sistema y los inconvenientes de seguridad.	Atendido	C2, C4, C5

Fuente: Autor.

Tabla 32. Validación de la atención de los requisitos de calidad

ID	Requisito	Estado	Componente
RNF1	El sistema debe realizar las notificaciones con una tasa menor a un segundo.	Atendido	C2, C3, C6, C12, C14, C15, C16, C17
RNF2	El sistema debe tener una tasa lo más cercana a cero de falsos positivos y falsos negativos.	Atendido	C2
RNF3	El sistema debe contar con una baja curva de apropiación por parte de los usuarios.	Atendido	C2, C3
RNF4	El sistema debe ser operado y controlado de forma fácil por los usuarios, minimizar el número de interacciones para realizar una tarea.	Atendido	C2, C3
RNF5	El sistema debe contar con la validación de datos al ingresar datos al sistema para proteger el correcto funcionamiento del sistema por errores humanos.	Atendido	C2, C3
RNF6	El sistema debe mostrar la información de forma concisa y clara para reducir el tiempo de respuesta ante una notificación de alarma.	Atendido	C2, C3
RNF7	El sistema debe monitorizar las 24 horas y 7 días de la semana.	Atendido	C8, C11, C12, C13
RNF8	El sistema debe estar en la capacidad de auto configurar la red de sensores inalámbrica que se utilice para la monitorización.	Atendido	C8, C11, C13
RNF9	El sistema debe enviar los datos de forma cifrada para garantizar la confidencialidad de la información.	Atendido	C9, C12
RNF10	El sistema debe ofrecer mecanismos de autenticación y control de acceso.	Atendido	C1, C4
RNF11	El sistema debe permitir la asignación de roles para gestionar los controles de acceso y sus respectivos permisos en el sistema para garantizar la integridad de la información.	Atendido	C1, C2
RNF12	El acceso al sistema debe estar restringido por el uso de claves asignadas a cada uno de los usuarios y debe contar con doble factor de autenticación.	Atendido	C1, C9
RNF13	El sistema debe garantizar la participación de las partes en las comunicaciones realizadas en el sistema, permitiendo que se pueda probar el origen, el destino y la entrega del mensaje.	Atendido parcialmente	C5

Fuente: Autor.

Los resultados obtenidos en las pruebas realizadas permiten determinar dos cosas: (i) los componentes incluidos en la arquitectura diseñada satisfacen el cumplimiento de los requisitos seguridad, identificados con los ID de RNF9 al RNF13, como se muestra en la Tabla 32. (ii) Que la arquitectura diseñada cumple con los requisitos funcionales del sistema y los requisitos relacionado con los demás atributos de calidad.

6.3 INTEGRACIÓN DEL FRAMEWORK

El *SMITH Framework* es el resultado de la integración de los dos modelos de ciberseguridad: el *SMITH Model* y el IoT-CyDM. El modelo IoT-CyDM es una representación semiformal del *SMITH Model*.

Los dos modelos pretender orientar al equipo de desarrollo en la implementación de la ciberseguridad en las aplicaciones IoT. Elementos del *SMITH Model* como los requisitos de seguridad y la arquitectura genérica de IoT, junto con el modelo IoT-CyDM realizan esta orientación en la fase de diseño de una aplicación IoT. Los otros elementos del *SMITH Model* como es la guía de buenas prácticas para el aseguramiento de aplicaciones IoT y la Evaluación de elementos de ciberseguridad en aplicaciones IoT se enfocan en orientan la fase de implementación y pruebas, respectivamente.

7. CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se presentan las conclusiones alcanzadas en esta investigación, se realiza una revisión de las contribuciones realizadas en este trabajo y se describe el trabajo futuro.

7.1 CONCLUSIONES

En este trabajo de investigación se propone un *framework* conceptual de ciberseguridad para aplicaciones IoT, llamado *Security Management in Internet of THings Framework – SMITH Framework*. Este *framework*, es una propuesta que busca contribuir a la solución del problema que se identificó en la literatura: que existen aplicaciones IoT inseguras por la falta de guías que orienten a los desarrolladores en la implementación del dominio de la ciberseguridad, desde la fase de diseño de estas y la evaluación de aplicaciones ya construidas.

El *SMITH Framework* contribuye de dos formas a la solución de este problema: la primera, a través del modelamiento del dominio de la ciberseguridad; y segundo, con una representación de este modelo mediante un lenguaje semiformal llamado UML. Estas contribuciones se traducen en dos modelos de ciberseguridad, los cuales integran el *SMITH Framework*: el *SMITH Model* y el modelo IoT-CyDM.

El *Security Management in Internet of THings Model – SMITH Model*, es una abstracción del dominio de la ciberseguridad, representada a través de un cubo, y su objetivo en orientar a los desarrolladores de aplicaciones IoT sobre qué se debe proteger en una aplicación IoT. Este modelo está compuesto por tres elementos principales: (i) el *SMITH Model* y la descripción de este; (ii) la guía de buenas prácticas para el aseguramiento de aplicaciones IoT; (iii) los instrumentos de evaluación de elementos de ciberseguridad en aplicaciones IoT.

El *IoT Cybersecurity Domain Model* – IoT-CyDM, es una representación semiformal del dominio descrito en el *SMITH Model* a través de un diagrama de clases UML. Este modelo identifica los conceptos claves del dominio IoT y del dominio de la ciberseguridad y la relación entre dichos estos. El resultado es un modelo del dominio de la ciberseguridad que oriente a ingenieros de software, arquitectos de software y arquitectos de soluciones en el diseño de modelos de dominio de una aplicación IoT en particular, sin importar el dominio de aplicación.

Este modelo será la base para que el diseño de la arquitectura de la solución IoT.

La validación del *SMITH Framework*, a través del método ATAM, permitió determinar que los componentes incluidos en la arquitectura diseñada para el caso de estudio satisfacen el cumplimiento de los requisitos de calidad relacionado con el atributo de la seguridad. Esto confirma la hipótesis planteada en este trabajo de investigación, la cual dice que "mediante un *framework*, compuesto por diferentes tipos de modelos, se puede orientar al equipo de desarrollo sobre cómo considerar ciberseguridad en las aplicaciones IoT".

Los resultados propuestos hacen un aporte a la brecha técnica e investigativa identificada en el estado del arte sobre *frameworks* de seguridad para aplicaciones IoT[20]. Las conclusiones de este estado del arte afirman que "existe una brecha de investigación y técnicos en mecanismos que ayuden y guíen en la implementación de la ciberseguridad en todo el dominio de IoT, el cual se consideren todas las capas de una aplicación IoT y esté orientada a aplicaciones IoT genéricas".

El *SMITH Framework* se orienta en facilitar la implementación de la ciberseguridad desde la fase de diseño de aplicaciones IoT. Además, este *framework* considera las características de la información: confidencialidad, integridad, disponibilidad y no repudio. Asimismo, los modelos propuestos abordan el dominio del IoT, y no se centran en un dominio de aplicación en particular, lo que permite que pueda ser usado para orientar la construcción de cualquier aplicación IoT.

7.2 REVISIÓN DE LAS CONTRIBUCIONES REALIZADAS

En esta sección se resume las principales contribuciones de este trabajo de investigación:

- **Modelo de gestión de ciberseguridad para aplicaciones IoT**

El modelo de gestión de ciberseguridad para aplicaciones IoT, llamado *SMITH Model* (ver capítulo 4), está compuesto por tres elementos: El modelo SMITH,

[20] Ver la sección 2.3.1 para más información.

que es una representación del dominio de la ciberseguridad para una aplicación IoT genérica; el segundo elemento es una guía de buenas prácticas de ciberseguridad que deben considerarse para asegurar una aplicación IoT, cuyas recomendaciones se organizan según la visión presentada en el modelo SMITH. Finalmente, se presenta el tercer elemento, un instrumento de evaluación buenas prácticas de ciberseguridad para aplicaciones IoT ya construidas, y de esta forma los desarrolladores puedan tomar medidas para reforzar la seguridad de sus soluciones IoT.

- **Modelo del dominio de ciberseguridad para aplicaciones IoT**

El modelo de domino de ciberseguridad para aplicaciones IoT, ver capítulo 5, llamado IoT-CyDM, es una representación del domino a través de un diagrama de clases UML que puede ser usado para instanciar modelos de aplicaciones IoT que serán instanciadas en una arquitectura IoT particular.

- **Caso de estudio sobre el estudio del *SMITH Framework* en el diseño de una aplicación IoT.**

Este caso de estudio de estudio que describe cómo usar los elementos del *SMITH Framework* para el diseño de una aplicación IoT que considere elementos de ciberseguridad, ver sección 6.1.

- **Estado del arte sobre *frameworks* de seguridad para aplicaciones IoT**.

Este estado del arte es una contribución a la literatura sobre *frameworks* de seguridad para aplicaciones IoT, ver sección 2.3.1. En este estado del arte se listan, clasifican y analizan dichas propuestas de seguridad y se identifica una brecha de investigación y técnica en mecanismos que ayuden y guíen en la implementación de la ciberseguridad en todo el dominio de IoT, el cual se consideren todas las capas de una aplicación IoT y esté orientada a aplicaciones IoT genéricas.

- **Metodología para la creación de modelos basado en requisitos de calidad.**

Esta metodología, conformada por cuatro fases, se propuso para la creación del *SMITH Model*, ver sección 4.1. Aunque esta metodología se usó para crear un modelo de seguridad, también podría ser usada para la creación de modelos basado en otros requisitos de calidad.

- **Revisión sistemática de la literatura sobre arquitecturas de referencia para aplicaciones IoT**

El objetivo de esta revisión sistemática de la literatura – RSL es identificar las arquitecturas de referencia – AR propuestas para aplicaciones IoT,, ver sección 4.2.1. En esta RSL es recuperaron 806 artículos y se aplicó una serie de filtros con criterios de calidad y exclusión. De este proceso resultado 40 estudios primarios, en los cuales se presentan o mencionan 40 arquitecturas de referencia para aplicaciones IoT.

Un trabajo siguiente a la RSL fue evaluar las 40 arquitecturas identificadas para determinar cuál de ellas realmente podrían considerarse como tal. Para esto, se hizo una depuración de las AR que fueran genéricas para IoT, descantando aquellas que fueran específicas para un dominio de aplicación. Seguidamente se usó el RaModel, un modelo para identificar elementos faltantes en arquitecturas de referencia (Nakagawa et al., 2012).

- **Arquitectura IoT genérica para aplicaciones IoT**

La arquitectura considera las funcionalidades comunes que debe tener una aplicación IoT, ver sección 4.3.4, y es el resultado del análisis de cuatro arquitecturas de referencia: (i) la del *IoT-A Project*, (ii) la propuesta por la ITU-T, (iii) la de *SmartSantander Project* y (iv) la propuesta por WSO2. Las funcionalidades genéricas identificadas se integraron en una arquitectura en capas, compuesta por tres capas: *Cloud layer*, *fog layer* y *dew layer*. Esta arquitectura tiene como propósito agrupar las funcionalidades básicas de una aplicación IoT con el fin de ofrecer interoperabilidad del *SMITH Model* con las diferentes arquitecturas de referencias y arquitecturas particulares para IoT.

- **Modelo del dominio IoT**

Este modelo del dominio IoT, es la base para el modelo del dominio de ciberseguridad para aplicaciones IoT, ver sección 5.1.2, y se considera que puede ser usado en otros contextos investigativos donde se requiera un modelo genérico y no basarse en una arquitectura de referencia en particular.

- **Artículos publicados**

Durante la revisión previa a la propuesta de este trabajo de investigación, se identificó que la literatura científica no había una división clara entre las aplicaciones de redes de sensores identificadas y las aplicaciones IoT. Por esta razón se propuso contribuir a la literatura en este tema. Este trabajo se presenta en dos artículos:

- **Rueda, J.**, & Talavera Portocarrero, J. (2017, diciembre 1). Similitudes y diferencias entre Redes de Sensores Inalámbricas e Internet de las Cosas: Hacia una postura clarificadora. Revista Colombiana De Computación, 18(2), 58-74. https://doi.org/10.29375/25392115.3218
- Manrique, J. A., **Rueda-Rueda, J. S.**, & Portocarrero, J. M. T. (2016). Contrasting Internet of Things and Wireless Sensor Network from a conceptual overview. In *Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on* (pp. 252–257).

7.3 TRABAJO FUTURO

Como trabajo futuro del *SMITH Framework* se proyecta validar, a través de prototipos, la guía de buenas prácticas de ciberseguridad para el aseguramiento de aplicaciones IoT considerada en el *SMITH Model*. De igual manera, enviar los instrumentos de evaluación, junto a un instrumento para la recolección de información, a desarrolladores de aplicaciones IoT en universidades, centros de investigación y empresas, y basado en la información recolectada, establecer mejoras a dichos instrumentos. En este mismo sentido, evaluar el modelo IoT-CyDM a través de casos de estudios que impliquen otros dominios de aplicación de IoT.

Estas validaciones buscan generar mejorar mejoras en el *framework* y los modelos que lo integran en dos aspectos: (i) los aspectos técnicos relacionados con el dominio IoT y aquellos relacionados con la ciberseguridad; y (ii) en aspectos de usabilidad, y que su lenguaje sea ameno y comprensible para los equipos de desarrollo de aplicaciones IoT.

REFERENCIAS

Abdmeziem, M. R., Tandjaoui, D., & Romdhani, I. (2016). Architecting the internet of things: state of the art. In *Robots and Sensor Clouds* (pp. 55–75). Springer.

Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile Software Development Methods: Review and Analysis. *CoRR*, *478*, 107. Retrieved from http://arxiv.org/abs/1709.08439

Abreu, D. P., Velasquez, K., Curado, M., & Monteiro, E. (2017). A resilient Internet of Things architecture for smart cities. *Annals of Telecommunications*, *72*(1–2), 19–30.

Adams, K. (2015). *Non-functional Requirements in Systems Analysis and Design*. Springer.

Addo, I. D., Ahamed, S. I., Yau, S. S., & Buduru, A. (2014). A reference architecture for improving security and privacy in Internet of Things applications. In *Mobile Services (MS), 2014 IEEE International Conference on* (pp. 108–115).

Aldosari, H. M. (2015). A Proposed Security Layer for the Internet of Things Communication Reference Model. *Procedia Computer Science*, *65*, 95–98.

Alhamedi, A. H., Snasel, V., Aldosari, H. M., & Abraham, A. (2014). Internet of things communication reference model. In *Computational Aspects of Social Networks (CASoN), 2014 6th International Conference on* (pp. 61–66).

Andolfi, F., Aquilani, F., Balsamo, S., & Inverardi, P. (2000). Deriving QNM from MSCs for performance evaluation of SA. In *ACM Workshop on Software Performance* (pp. 220–229).

Aquilani, F., Balsamo, S., & Inverardi, P. (2001). Performance analysis at the software architectural design level. *Performance Evaluation*, *45*(2–3), 147–178.

Ashton, K. (2009). That “Internet of Things” Thing. *RFID Journal*, 1. Retrieved from www.rfidjournal.com/articles/pdf?4986

Atamli, A. W., & Martin, A. (2014). Threat-Based Security Analysis for the Internet

of Things. In *2014 International Workshop on Secure Internet of Things* (pp. 35–43). IEEE. https://doi.org/10.1109/SIoT.2014.10

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: a survey. *Computer Networks*, *54*, 2787–2805. https://doi.org/10.1007/s10796-014-9492-7

Atzori, L., Iera, A., & Morabito, G. (2017). Understanding the Internet of Things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, *56*, 122–140. https://doi.org/10.1016/j.adhoc.2016.12.004

Babar, M. A., & Gorton, I. (2004). Comparison of scenario-based software architecture evaluation methods. In *11th Asia-Pacific Software Engineering Conference, 2004*.

Balsamo, S., Inverardi, P., & Mangano, C. (1998). An approach to performance evaluation of software architectures. In *Proceedings of the 1st international workshop on Software and performance* (pp. 178–190).

Banda, G., Chaitanya, K., & Mohan, H. (2015). An IoT protocol and framework for OEMs to make IoT-enabled devices forward compatible. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2015 11th International Conference on* (pp. 824–832).

Barker, E. (2016). *Recommendation for Key Management Part 1: General*. Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-57pt1r4

Barker, E., Smid, M., Branstad, D., & Chokhani, S. (2013). *A Framework for Designing Cryptographic Key Management Systems*. Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-130

Bassi, A., Bauer, M., Fiedler, M., Kramp, T., van Kranenburg, R., Lange, S., & Meissner, S. (Eds.). (2013). *Enabling Things to Talk*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-40403-0

Bauer, M., Boussard, M., Bui, N., Carrez, F., Jardak, C., De Loof, J., … Salinas, A. (2013). *Deliverable D1.5 – Final architectural reference model for the IoT v3.0*.

Bauer, M., Boussard, M., Bui, N., De Loof, J., Magerkurth, C., Meissner, S., …

Walewski, J. W. (2013). IoT Reference Architecture. In A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. van Kranenburg, S. Lange, & S. Meissner (Eds.), *Enabling Things to Talk: Designing IoT solutions with the IoT Architectural Reference Model* (pp. 163–211). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-40403-0_8

Bauer, M., Bui, N., De Loof, J., Magerkurth, C., Nettsträter, A., Stefa, J., & Walewski, J. W. (2013). IoT Reference Model. In *Enabling Things to Talk* (pp. 113–162). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-40403-0_7

Bayuk, J. L., Healey, J., Rohmeyer, P., Sachs, M. H., Schmidt, J., & Weiss, J. (2012). *Cyber Security Policy Guidebook*. Wiley Publishing.

Beltrán G., Ó. A. (2005). Revisiones sistemáticas de la literatura. *Revista Colombiana de Gastroenterología*, *20*(1), 10.

Bengtsson, P., & Bosch, J. (1998). Scenario-based software architecture reengineering. In *Fifth International Conference on oftware Reuse, 1998* (pp. 308–317). IEEE.

Bengtsson, P., & Bosch, J. (1999). Architecture level prediction of software maintenance. In *Software Maintenance and Reengineering, 1999. Proceedings of the Third European Conference on* (pp. 139–147).

Bengtsson, P., Lassing, N., Bosch, J., & van Vliet, H. (2004). Architecture-level modifiability analysis (ALMA). *Journal of Systems and Software*, *69*(1–2), 129–147.

Bergner, K., Rausch, A., Sihling, M., & Ternité, T. (2005). DoSAM--domain-specific software architecture comparison model. In *Quality of Software Architectures and Software Quality* (pp. 4–20). Springer.

Bernabe, J. B., Hernández, J. L., Moreno, M. V., & Gomez, A. F. S. (2014). Privacy-preserving security framework for a social-aware internet of things. In *International conference on ubiquitous computing and ambient intelligence* (pp. 408–415).

Biolchini, J., Gomes Mian, P., Cruz Natali, A. C., & Horta Travassos, G. (2005). *Systematic Review in Software Engineering*. Rio de Jainero.

Boehm, B. (n.d.). Evaluating a Software Architecture (pp. 19–42).

Boehm, B. W., Brown, J. R., & Kaspar, H. (1978). Characteristics of Software Quality.

Bohli, J.-M., Skarmeta, A., Moreno, M. V., García, D., & Langendörfer, P. (2015). SMARTIE project: Secure IoT data management for smart cities. In *Recent Advances in Internet of Things (RIoT), 2015 International Conference on* (pp. 1–6).

Borgia, E. (2014). The internet of things vision: Key features, applications and open issues. *Computer Communications*, *54*, 1–31. https://doi.org/10.1016/j.comcom.2014.09.008

Boroojeni, K. G., Amini, M. H., & Iyengar, S. S. (2016). Smart Grids: Security and Privacy Issues. Springer.

Boussard, M., Meissner, S., Nettsträter, A., Olivereau, A., Segura, A. S., Thoma, M., & Walewski, J. W. (2013). A Process for Generating Concrete Architectures. In *Enabling Things to Talk* (pp. 45–111). Springer.

Brooks, F. (1987). No Silver Bullet: Essence and Accidents of Software Engineering. *IEEE Computer*, *20*(4), 10–19.

Caltum, E., & Segal, O. (2016). Exploitation of IoT devices for Launching Mass-Scale Attack Campaigns.

Capgemini. (2018). *Cybersecurity talent — The big gap in cyber protection*.

Caracciolo, A., Lungu, M. F., & Nierstrasz, O. (2014). How Do Software Architects Specify and Validate Quality Requirements? In *European Conference on Software Architecture* (pp. 374–389). Springer.

CASAGRAS Project. (2009). *RFID and the Inclusive Model for the Internet of Things*.

Cavalcante, E., Alves, M. P., Batista, T., Delicato, F. C., & Pires, P. F. (2015). An analysis of reference architectures for the internet of things. In *Proceedings of the 1st International Workshop on Exploring Component-based Techniques for Constructing Reference Architectures* (pp. 13–16).

Cavalcante, E., Pereira, J., Alves, M. P., Maia, P., Moura, R., Batista, T., … Pires, P. F. (2016). On the interplay of Internet of Things and Cloud Computing: A systematic mapping study. *Computer Communications*, *89–90*, 17–33. https://doi.org/10.1016/j.comcom.2016.03.012

Chant, I. (2017). The Cybersecurity Talent Shortage Is Here, and It's a Big Threat to Companies. Retrieved January 10, 2018, from http://theinstitute.ieee.org/ieee-roundup/blogs/blog/the-cybersecurity-talent-shortage-is-here-and-its-a-big-threat-to-companies

Chen, Q., Abdelwahed, S., & Erradi, A. (2014). A model-based validated autonomic approach to self-protect computing systems. *IEEE Internet of Things Journal*, *1*(5), 446–460.

Cheung, R. C. (1980). A user-oriented software reliability model. *IEEE Transactions on Software Engineering*, (2), 118–125.

Chung, L., & do Prado Leite, J. C. S. (2009). On Non-Functional Requirements in Software Engineering. In *Conceptual Modeling: Foundations and Applications* (pp. 363–379). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-02463-4_19

Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). *Non-functional Requirements in Software Engineering*. Springer Science & Business Media.

Cimpanu, C. (2016). Problems Reappear for IoT Device Owners with Discovery of New DDoS Trojan.

Cirani, S., Ferrari, G., & Veltri, L. (2013). Enforcing Security Mechanisms in the IP-Based Internet of Things: An Algorithmic Overview. *Algorithms*, *6*(2), 197–226. https://doi.org/10.3390/a6020197

Cisco. (2015). *Mitigating the Cybersecurity Skills Shortage: Top Insights and Actions from Cisco Security Advisory Services*.

Cisco. (2016a). *Internet of Things at a Glance*.

Cisco. (2016b). The Internet of Things. It's not about things. It's about service.

Retrieved from https://www.jasper.com/sites/default/files/pdf/IoT_Infographic.pdf%7D

Cisco. (2017). *Cisco 2017 Annual Cybersecurity Report*.

CISCO. The Internet of Things Reference Model (2014). Retrieved from http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf

Clements, P., Garlan, D., Little, R., Nord, R., & Stafford, J. (2003). Documenting software architectures: views and beyond. In *Proceedings of the 25th International Conference on Software Engineering* (pp. 740--741). ACM. Retrieved from http://delivery.acm.org/10.1145/780000/776928/p740-clements.pdf?ip=200.69.124.106&id=776928&acc=ACTIVE SERVICE&key=4D9619BEF5D5941F.D0AFA4C1BA803950.4D4702B0C3E38B35.4D4702B0C3E38B35&__acm__=1520370891_ece2c328b7de31eaf77e2c65c0fa3758

CNSS. (2010). National Information Assurance (IA) Glossary. Committee on National Security Systems.

Cobb, S. (2016a). Cybersecurity skills gap: It's big and it's bad for security. Retrieved from https://www.welivesecurity.com/2016/12/16/cybersecurity-skills-gap-big-and-bad/

Cobb, S. (2016b). Jackware: When connected cars meet ransomware.

Cobb, S. (2017). RoT: Ransomware of Things.

Colciencias. (2016). *Tipología de proyectos calificados como de carácter científico, tecnológico e innovación* (Vol. 4). https://doi.org/10.1007/s13398-014-0173-7.2

Condry, M. W., & Nelson, C. B. (2016). Using Smart Edge IoT Devices for Safer, Rapid Response With Industry IoT Control Operations. *Proceedings of the IEEE*, *104*(5), 938–946.

Cortellessa, V., & Mirandola, R. (2000). Deriving a queueing network based performance model from UML diagrams. In *Proceedings of the 2nd international workshop on Software and performance* (pp. 58–70).

Currie, R. (2016). Developments in Car Hacking. *SANS Institute InfoSec Reading Room*, 1–34.

CyberX. (2016). Radiation IoT Cyber Security Campaign.

Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, *10*(4), 2233–2243.

Dalipi, F., & Yayilgan, S. Y. (2016). Security and Privacy Considerations for IoT Application on Smart Grids: Survey and Research Challenges. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)* (pp. 63–68). IEEE. https://doi.org/10.1109/W-FiCloud.2016.28

Davis, A. M. (1993). *Software Requirements: Objects, Functions and States*. Prentice-Hall, Inc.

De, S., Carrez, F., Reetz, E., Tönjes, R., & Wang, W. (2013). Test-Enabled Architecture for IoT Service Creation and Provisioning. In *The Future Internet Assembly* (pp. 233–245). https://doi.org/10.1007/978-3-642-38082-2_20

Deloitte. (2018). *The cybersecurity talent shortage: An emerging challenge for consumer products companies*.

Dobre, C., Mavromoustakis, C. X., Garcia, N., Ivanova Goleva, R., & Mastorakis, G. (Eds.). (2017). Glossary. In *Ambient Assisted Living and Enhanced Living Environments* (pp. xliii–xliv). Elsevier. https://doi.org/10.1016/B978-0-12-805195-5.00028-4

Dykstra, J. (2015). *Essential Cybersecurity Science: Build, Test, and Evaluate Secure Systems* (First Edit). O'Reilly Media.

Edwards, S., & Profetis, I. (2016). Hajime: Analysis of a decentralized internet worm for IoT devices.

Elmaghraby, A. S., & Losavio, M. M. (2014). Cyber security challenges in Smart Cities: Safety, security and privacy. *Journal of Advanced Research*, *5*(4), 491–497. https://doi.org/10.1016/j.jare.2014.02.006

Emm, D., Unuchek, R., & Kruglov, K. (2016). *Kaspersky Security Bulletin 2016. Review of the Year*.

Essery, Michael. (2016). Today 65% of Enterprises Already Using Internet of Things; Business Value found in Optimizing Operations and Reducing Risk.

Fernandes, J., Nati, M., Loumis, N. S., Nikoletseas, S., Raptis, T. P., Krco, S., … Ziegler, S. (2015). IoT Lab: Towards co-design and IoT solution testing using the crowd. In *Recent Advances in Internet of Things (RIoT), 2015 International Conference on* (pp. 1–6).

Finkle, J. (2016). J&J warns diabetic patients: Insulin pump vulnerable to hacking. Reuters.

Fiutem, R., & Antoniol, G. (1998). Identifying design-code inconsistencies in object-oriented software: A case study. In *Software Maintenance, 1998. Proceedings., International Conference on* (pp. 94–102).

Folmer, E., Van Gurp, J., & Bosch, J. (2004). Software architecture analysis of usability. In *International Workshop on Design, Specification, and Verification of Interactive Systems* (pp. 38–58).

ForeScout Technologies. (2016). *IoT Enterprise Risk Report*.

Formisano, C., Pavia, D., Gurgen, L., Yonezawa, T., Galache, J. A., Doguchi, K., & Matranga, I. (2015). The advantages of IoT and cloud applied to smart cities. In *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on* (pp. 325–332).

Forrester. (2017). *Predictions 2018: IoT Moves From Experimentation To Business Scale*.

Fowler, K. (2016). Cybersecurity. In *Enterprise Risk Management* (pp. 91–108). Elsevier. https://doi.org/10.1016/B978-0-12-800633-7.00007-9

Fox-Brewster, T. (2016). How Hacked Cameras Are Helping Launch The Biggest Attacks The Internet Has Ever Seen. Forbes.

Fundación Telefónica. (2016). *Ciberseguridad, la protección de la información en*

un mundo digital. Fundación Telefónica, Editorial Ariel S.A.

Gartner Inc. (2015). Gartner Says 6.4 Billion Connected “Things” Will Be in Use in 2016, Up 30 Percent From 2015.

Gartner Inc. (2016a). Gartner’s 2016 Hype Cycle for Emerging Technologies Identifies Three Key Trends That Organizations Must Track to Gain Competitive Advantage. Retrieved from www.gartner.com/newsroom/id/3412017

Gartner Inc. (2016b). Gartner Says By 2020, More Than Half of Major New Business Processes and Systems Will Incorporate Some Element of the Internet of Things.

Gartner Inc. (2016c). Gartner Says Worldwide IoT Security Spending to Reach $348 Million in 2016.

Gartner Inc. (2016d). Top 10 Strategic Technology Trends for 2017.

Ge, M., & Kim, D. S. (2015). A framework for modeling and assessing security of the internet of things. In *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on* (pp. 776–781).

Gibbs, S. (2015). Hackers can hijack Wi-Fi Hello Barbie to spy on your children. The Guardian.

Gilchrist, A. (2016). IIoT Reference Architecture. In *Industry 4.0* (pp. 65–86). Springer.

Gluhak, A., Hauswirth, M., Krco, S., Stojanovic, N., Bauer, M., Nielsen, R. H., … Corcho, O. (2011). An Architectural Blueprint for a Real-World Internet. In *Future Internet Assembly* (pp. 67–80).

Gluhak, A., Munoz, L., Sotres, P., Sanchez, L., Roux, P., Sanchez, B., … Hernandez, A. L. (2013). *Third Cycle Architecture Specification*.

Gokhale, S. S., & Trivedi, K. S. (2002). Reliability prediction and sensitivity analysis based on software architecture. In *Software Reliability Engineering, 2002. ISSRE 2003. Proceedings. 13th International Symposium on* (pp. 64–75).

Gómez Vargas, M., Galeano Higuita, C., & Jaramillo Muñoz, D. A. (2015). El

estado del arte: una metodología de investigación. *Revista Colombiana de Ciencias Sociales*, *6*(2), 423–442.

Grant, M. J., & Booth, A. (2009). A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal*, *26*(2), 91–108. https://doi.org/10.1111/j.1471-1842.2009.00848.x

Green, P. E. J. (2016). Introduction to Risk Management Principles. In *Enterprise Risk Management* (pp. 1–13). Elsevier. https://doi.org/10.1016/B978-0-12-800633-7.00001-8

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660.

Guo, B., Zhang, D., Wang, Z., Yu, Z., & Zhou, X. (2013). Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things. *Journal of Network and Computer Applications*, *36*(6), 1531–1539.

Hayashi, K. (2014). IoT Worm Used to Mine Cryptocurrency.

Heer, T., Garcia-Morchon, O., Hummen, R., Keoh, S. L., Kumar, S. S., & Wehrle, K. (2011). Security Challenges in the IP-based Internet of Things. *Wireless Personal Communications*, *61*(3), 527–542. https://doi.org/10.1007/s11277-011-0385-5

Hellaoui, H., Bouabdallah, A., & Koudil, M. (2016). TAS-IoT: Trust-Based Adaptive Security in the IoT. In *Local Computer Networks (LCN), 2016 IEEE 41st Conference on* (pp. 599–602).

Herjavec Group. (2017). *2017 Cybersecurity Jobs Report*.

Hernandez-Ramos, J. L., Pawlowski, M. P., Jara, A. J., Skarmeta, A. F., & Ladid, L. (2015). Toward a lightweight authentication and authorization framework for smart objects. *IEEE Journal on Selected Areas in Communications*, *33*(4), 690–702.

Hernandez Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. del P. (2010). *Metodología de la investigación* (Quinta edi). McGraw-Hill, Inc.

Hewlett Packard Enterprise. (2015). Internet Of things research study.

Hioureas, V. (2015, May). Does CCTV put the public at risk of cyberattack? Kaspersky Labs.

Höller, J., Tsiatsis, V., Mulligan, C., Karnouskos, S., Avesand, S., & Boyle, D. (2014a). Architecture Reference Model. In *From Machine-To-Machine to the Internet of Things* (pp. 167–197). Elsevier. https://doi.org/10.1016/B978-0-12-407684-6.00007-3

Höller, J., Tsiatsis, V., Mulligan, C., Karnouskos, S., Avesand, S., & Boyle, D. (2014b). IoT Architecture – State of the Art. In *From Machine-To-Machine to the Internet of Things* (pp. 145–165). Elsevier. https://doi.org/10.1016/B978-0-12-407684-6.00006-1

Höller, J., Tsiatsis, V., Mulligan, C., Karnouskos, S., Avesand, S., & Boyle, D. (2014c). IoT Reference Architecture. In *From Machine-To-Machine to the Internet of Things* (pp. 199–223). Elsevier. https://doi.org/10.1016/B978-0-12-407684-6.00008-5

Hopkin, P. (2017). *Fundamentals of Risk Management: Understanding, evaluating and implementing effective risk management*.

Huang, X., Craig, P., Lin, H., & Yan, Z. (2015). SecIoT: a security framework for the Internet of Things. *Security and Communication Networks*, *9*, 3083–3095. https://doi.org/10.1002/sec.1259

Hussein, N. H., & Khalid, A. (2016). A survey of Cloud Computing Security challenges and solutions. *International Journal of Computer Science and Information Security*, *14*(1), 52.

Hwang, H., & Park, Y. B. (2017). Safety - Critical Software Quality Improvement Using Requirement Analysis. In *2017 International Conference on Platform Technology and Service (PlatCon)* (pp. 1–4). IEEE. https://doi.org/10.1109/PlatCon.2017.7883725

IEEE. (1990). IEEE Standard Glossary of Software Engineering Terminology.

IEEE Computer Society. (2014). *Guide to the Software Engineering - Body of Knowledge.* (P. Bourque & R. E. Fairley, Eds.), *IEEE Computer Society* (V3 ed.). https://doi.org/10.1234/12345678

Intel. (2016). A Guide to the Internet of Things. How billion of online objects are making the web wiser.

Intel Security, & CSIS. (2016). *Hacking the Skills Shortage: A study of the international shortage in cybersecurity skills*.

Internet of Things Guide. (2016). Glossary Term.

Ionita, M. T., Hammer, D., & Obbink, H. (2002). Scenario-Based Software Architecture Evaluation Methods: An Overview. *Technical University*, 1–10.

Iorga, M., Feldman, L., Barton, R., Martin, M. J., Goren, N., & Mahmoudi, C. (2017). *The NIST Definition of Fog Computing*.

IoT-A Project. (2016). Requirements — IOT-A: Internet of Things Architecture. Retrieved from http://www.iot-a.eu/public/requirements/copy_of_requirements

ISACA. (2013). A simple definition of cybersecurity.

ISACA. (2016a). 2016 Cybersecurity Skills Gap. Retrieved from https://isaca.org.ar/2016/12/07/cybersecurity-skills-gap/

ISACA. (2016b). *Cybersecurity Fundamentals Glossary*.

ISACA. (2018). State of Cybersecurity Study: Security Budgets Increasing, But Qualified Cybertalent Remains Hard to Find. Retrieved May 31, 2018, from http://www.isaca.org/About-ISACA/Press-room/News-Releases/2018/Pages/State-of-Cybersecurity-Study-Security-Budgets-Increasing-But-Qualified-Cybertalent-Remains-Hard-to-Find.aspx

ISO/IEC/IEEE. (2010). *ISO/IEC/IEEE 24765:2010 Systems and software engineering - Vocabulary*.

ISO/IEC/IEEE. (2011). ISO/IEC/IEEE 42010:2011, Systems and software engineering — Architecture description.

ISO/IEC. (2012). ISO/IEC 27032:2012, Information technology -- Security techniques -- Guidelines for cybersecurity. Retrieved from https://www.iso.org/standard/44375.html

ISO/IEC. (2013). ISO/IEC 27001:2013, Information technology -- Security techniques -- Information security management systems -- Requirements.

ISO/IEC. (2015). ISO/IEC/IEEE 27017:2015, Information technology -- Security techniques -- Code of practice for information security controls based on ISO/IEC 27002 for cloud services.

ITU-T. (2012). *Overview of the Internet of things*. *Series Y: Global information infrastructure, internet protocol aspects and next-generation networks - Frameworks and functional architecture models*.

ITU-T. (2014a). *F.748.0: Common requirements for Internet of things (IoT) applications*.

ITU-T. (2014b). *Y.2066: Common requirements of the Internet of things*.

Jiménez, J. A., Russo, M., Krco, S., Bezanilla, R., Munoz, L., Galache, J. A., … Koutsoubelias, M. (2012). *Second Cycle Architecture Specification*.

Jóźwiak, L. (2017a). Advanced mobile and wearable systems. *Microprocessors and Microsystems*, *50*, 202–221. https://doi.org/10.1016/j.micpro.2017.03.008

Jóźwiak, L. (2017b). Advanced mobile and wearable systems. *Microprocessors and Microsystems*, *50*, 202–221. https://doi.org/10.1016/j.micpro.2017.03.008

Kaspersky Lab. (2016). *Kaspersky Security Bulletin 2016*.

Kaspersky Labs. (2015a). *Damage Control: The Cost of Security Breaches. It Security Risk Special Report Series*.

Kaspersky Labs. (2015b). *Global IT Security Risks Survey*.

Kazman, R., Bass, L., Abowd, G., & Webb, M. (1994). SAAM: A method for analyzing the properties of software architectures. In *Software Engineering, 1994. Proceedings. ICSE-16., 16th International Conference on* (pp. 81–90).

Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., & Carriere, J. (1998). *The Architecture Tradeoff Analysis Method*.

Kim, D., & Solomon, M. G. (2016). *Fundamentals of Information Systems Security*. Jones & Bartlett Learning. Retrieved from https://books.google.com.co/books?id=Yb4eDQAAQBAJ

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering version 2.3*.

Kotonya, G., & Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques* (1st ed.). Wiley Publishing.

Krco, S., Pokric, B., & Carrez, F. (2014). Designing IoT architecture (s): A European perspective. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on* (pp. 79–84).

Krishnamurthy, S., & Mathur, A. P. (1997). On the estimation of reliability of a software system using reliabilities of its components. In *Software Reliability Engineering, 1997. Proceedings., The Eighth International Symposium on* (pp. 146–155).

Kubat, P. (1989). Assessing reliability of modular software. *Operations Research Letters*, *8*(1), 35–41.

Laprie, J.-C. (1984). Dependability evaluation of software systems in operation. *IEEE Transactions on Software Engineering*, (6), 701–714.

Lassing, N. H., Rijsenbrij, D. B. B., & van Vliet, H. (1999). On software architecture analysis of flexibility, complexity of changes: Size isn't everything.

Lee, C., Zappaterra, L., Kwanghee Choi, & Hyeong-Ah Choi. (2014). Securing smart home: Technologies, security challenges, and security requirements. In *2014 IEEE Conference on Communications and Network Security* (pp. 67–72). IEEE. https://doi.org/10.1109/CNS.2014.6997467

Lee, G. M., Crespi, N., Choi, J. K., & Boussard, M. (2013). Internet of things. In *Evolution of Telecommunication Services* (pp. 257–282). Springer.

Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, *58*(4), 431–440.

Lee, S., & Kim, S. (2013). Hacking, surveilling, and deceiving victims on Smart TV. Black Hat.

Leyden, J. (2016). One Ring to pwn them all: IoT doorbell can reveal your Wi-Fi key. The Register.

Li, S., Xu, L. Da, & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, *17*(2), 243–259. https://doi.org/10.1007/s10796-014-9492-7

Lindvall, M., Tvedt, R. T., & Costa, P. (2003). An empirically-based process for software architecture evaluation. *Empirical Software Engineering*, *8*(1), 83–108.

Liu, L., Yin, L., Guo, Y., & Fang, B. (2014). EAC: a framework of authentication property for the IoTs. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2014 International Conference on* (pp. 102–105).

Lize, G., Jingpei, W., & Bin, S. (2014). Trust management mechanism for Internet of Things. *China Communications*, *11*(2), 148–156.

Lloyd's. (2017). *Counting the cost Cyber exposure decoded*.

Lloyd, W., & Connie, S. (2002). PASA: A Method for the Performance Assessment of Software Architectures. In *Proceedings of the Third International Workshop on Software and Performance (WOSP'2002), July* (pp. 24–26).

Loucopoulus, P., & Karakostas, V. (1995). *System Requirements Engineering*. McGraw-Hill, Inc.

Ma, M., Wang, P., & Chu, C.-H. (2013). Data management for internet of things: challenges, approaches and opportunities. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing* (pp. 1144–1151).

Mahalank, S. N., Malagund, K. B., & Banakar, R. M. (2016). Non Functional Requirement Analysis in IoT based smart traffic management system. In *2016 International Conference on Computing Communication Control and automation (ICCUBEA)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICCUBEA.2016.7860147

Mahalle, P. N., Anggorojati, B., Prasad, N. R., & Prasad, R. (2013). Identity authentication and capability based access control (iacac) for the internet of things. *Journal of Cyber Security and Mobility*, *1*(4), 309--348.

Malware Must Die. (2016). MMD-0058-2016 - Linux/NyaDrop - a linux MIPS IoT bad news.

Manrique, J. A., Rueda-Rueda, J. S., & Portocarrero, J. M. T. (2016). Contrasting Internet of Things and Wireless Sensor Network from a Conceptual Overview. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 252–257). IEEE. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2016.66

Maxwell, J. A. (2005). Conceptual framework: What do you think is going on. *Qualitative Research Design: An Interactive Approach*, *41*, 33–63.

Mead, N. R., & Stehney, T. (2005). Security quality requirements engineering (SQUARE) methodology. *ACM SIGSOFT Software Engineering Notes*, *30*(4), 1. https://doi.org/10.1145/1082983.1083214

Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing*.

Microsoft Colombia. (2016). Principales tendencias de seguridad en IoT.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. sage.

Miller, C., & Valasek, C. (2015). *Remote Exploitation of an Unaltered Passenger Vehicle*.

Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516.

Miranda, J., Mäkitalo, N., Garcia-Alonso, J., Berrocal, J., Mikkonen, T., Canal, C., & Murillo, J. M. (2015). From the Internet of Things to the Internet of People. *IEEE Internet Computing*, *19*(2), 40–47.

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009). Preferred Reporting Items for Systematic Reviews and Meta-Analyses: The PRISMA Statement.

PLoS Medicine, *6*(7), e1000097. https://doi.org/10.1371/journal.pmed.1000097

Molter, G. (1999). Integrating SAAM in domain-centric and reuse-based development processes. In *Proceedings of the 2nd Nordic Workshop on Software Architecture, Ronneby* (pp. 1–10).

Monteiro, C., Oliveira, M., Bastos, J., Ramrekha, T., & Rodriguez, J. (2014). Social Networks and Internet of Things, an Overview of the SITAC Project. In *International Wireless Internet Conference* (pp. 191–196).

Moore, B. J. (1994). Achieving software quality through requirements analysis. In *Proceedings of 1994 IEEE International Engineering Management Conference - IEMC '94* (pp. 78–83). IEEE. https://doi.org/10.1109/IEMC.1994.379948

Morán Delgado, G., & Alvarado Cervantes, D. G. (2010). *Métodos de investigación* (Primera ed). Pearson Education.

Mossburg, E., Gelinne, J., & Calzada, H. (2016). Beneath the surface of a cyberattack: A deeper look at business impacts.

Mostow, J. (1985). Towards Better Models of the Design Process. *AI Magazine*, *6*(1), 44–57.

Mozzaquatro, B. A., Jardim-Goncalves, R., Melo, R., & Agostinho, C. (2016). The application of security adaptive framework for sensor in industrial systems. In *Sensors Applications Symposium (SAS), 2016 IEEE* (pp. 1–6).

Muñoz, L., Sanchez, L., Galache, J. A., Gutierrez, V., Garcia, R., Poyato, P., … Ramdhany, R. (2011). *First Cycle Architecture Specification*.

Murphy, G. C., Notkin, D., & Sullivan, K. (1995). Software reflexion models: Bridging the gap between source and high-level models. *ACM SIGSOFT Software Engineering Notes*, *20*(4), 18–28.

Nakagawa, E. Y., Oquendo, F., & Becker, M. (2012). RAModel : A Reference Model for Reference Architectures. In *Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on* (pp. 297–301). IEEE. https://doi.org/10.1109/WICSA-ECSA.212.49

Namal, S., Gamaarachchi, H., MyoungLee, G., & Um, T.-W. (2015). Autonomic trust management in cloud-based and highly dynamic IoT applications. In *ITU Kaleidoscope: Trust in the Information Society (K-2015), 2015* (pp. 1–8).

Naur, P., & Randell, B. (1969). Software Engineering: Report of a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7-11 Oct. 1968, Brussels, Scientific Affairs Division, NATO.

Neisse, R., Fovino, I. N., Baldini, G., Stavroulaki, V., Vlacheas, P., & Giaffreda, R. (2014). A model-based security toolkit for the internet of things. In *Availability, Reliability and Security (ARES), 2014 Ninth International Conference on* (pp. 78–87).

Nia, A. M., & Jha, N. K. (2016). A comprehensive study of security of internet-of-things. *IEEE Transactions on Emerging Topics in Computing*.

NIST. (2011). ISO/IEC 25010:2011 - Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models.

NIST. (2013). Glossary of Key Information Security Terms.

NowSecure. (2016). 2016 NowSecure Mobile Security Report.

Object Management Group. (2005). Introducction to OMG's Unified Modeling Language. Retrieved from http://www.uml.org/what-is-uml.htm

Object Management Group. (2017). About the Unified Modeling Language Specification Versión 2.5.1. Retrieved from https://www.omg.org/spec/UML/About-UML/

Oficina Nacional de Seguridad. (2016). *Normas de la Autoridad Nacional para la Protección de la Información Clasificada*. Retrieved from http://www.buenjuicio.com/wp-content/uploads/2015/07/Normas_de_la_Autoridad_Nacional_para_la_Proteccion_de_la_Informacion_Clasificada.pdf

Oltski, J. (2017). *The Life and Times of Cybersecurity Professionals*.

OWASP. (2016a). IoT Framework Assessment. Retrieved November 29, 2017, from https://www.owasp.org/index.php/IoT_Framework_Assessment

OWASP. (2016b). Principles of IoT Security. Retrieved November 4, 2017, from https://www.owasp.org/index.php/Principles_of_IoT_Security

OWASP. (2017a). About The Open Web Application Security Project. Retrieved from www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project

OWASP. (2017b). OWASP Internet of Things (IoT) Project.

OWASP. (2017c). Password Storage Cheat Sheet. Retrieved November 29, 2017, from https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet

Owens, D. (2005). Documenting Software Architectures: Views and Beyond. *Technical Communication*, *52*(1), 75–77.

Pacheco, J., & Hariri, S. (2016). IoT Security Framework for Smart Cyber Infrastructures. In *Foundations and Applications of Self* Systems, IEEE International Workshops on* (pp. 242–247).

Pacheco, J., Satam, S., Hariri, S., Grijalva, C., & Berkenbrock, H. (2016). IoT Security Development Framework for building trustworthy Smart car services. In *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on* (pp. 237–242).

Pastrana, S., Rodriguez-Canseco, J., & Calleja, A. (n.d.). ArduWorm: A Functional Malware Targeting Arduino Devices.

Patel, P., & Cassou, D. (2015). Enabling high-level application development for the internet of things. *Journal of Systems and Software*, *103*, 62–84.

Patiño, R. G. (2016). El estado del arte en la investigación: ¿Análisis de los conocimientos acumulados o indagación por nuevos sentidos? *Revista Folios*, *2*(44).

Pawar, M. V, & Anuradha, J. (2015). Network Security and Types of Attacks in Network. *Procedia Computer Science*, *48*, 503–506.

Picco, G. Pietro. (2010). Software engineering and wireless sensor networks. In *Proceedings of the FSE/SDP workshop on Future of software engineering research - FoSER '10* (p. 283). New York, New York, USA: ACM Press.

https://doi.org/10.1145/1882362.1882421

Pohl, K. (2010). *Requirements Engineering: Fundamentals, Principles, and Techniques* (1st Editio). Springer Publishing Company.

Pressman, R. S. (2010). *Ingeniería del Software: un enfoque práctico* (Séptima ed). The McGraw-Hill.

Radomirovic, S. (2010). Towards a Model for Security and Privacy in the Internet of Things. In *Proc. First Int'l Workshop on Security of the Internet of Things* (p. 6).

Ratkowski, A. (2016). Architecture for Internet of Things Analytical Ecosystem. In *Dependability Engineering and Complex Systems* (pp. 385–393). Springer.

Refsdal, A., Solhaug, B., & Stølen, K. (2015). *Cyber-Risk Management*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-23570-7

Riazul Islam, S. M., Daehan Kwak, Humaun Kabir, M., Hossain, M., & Kyung-Sup Kwak. (2015). The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access*, *3*, 678–708. https://doi.org/10.1109/ACCESS.2015.2437951

Robles, T., Alcarria, R., de Andrés, D. M., Navarro, M., Calero, R., Iglesias, S., & López, M. (2015). An IoT based reference architecture for smart water management processes. *JoWUA*, *6*(1), 4–23.

Roman, G.-C. (1985). A taxonomy of current issues in requirements engineering. *IEEE Computer*, *18*(4), 14–23.

Ross, E. (2016). Baby monitors "hacked": Parents warned to be vigilant after voices heard coming from speakers. Independent.

Roy, B., & Graham, N. (2008). *Methods for Evaluating Software Architecture: A Survey*. Ontario, Canada.

Rozanski, N., & Woods, E. (2005). *Applying Viewpoints and Views to Software Architecture*.

RSA. (2016). 2016: Current State of Cybercrime.

Rueda R., J. S., & TalaveraP., J. M. (2017). Similitudes y diferencias entre Redes de Sensores Inalámbricas e Internet de las Cosas: Hacia una postura clarificadora Similarities and differences between Wireless Sensor Networks and the Internet of Things: Towards a clarifying position. *Revista Colombiana de Computación*, *18*(2), 58–74. https://doi.org/10.29375/25392115.3218

Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, *35*(3), 8–13. https://doi.org/10.1145/1764810.1764814

Sadeghi, A.-R., Wachsmann, C., & Waidner, M. (2015). Security and privacy challenges in industrial internet of things. In *Proceedings of the 52nd Annual Design Automation Conference on - DAC '15* (pp. 1–6). New York, New York, USA: ACM Press. https://doi.org/10.1145/2744769.2747942

Sanchez, L., Muñoz, L., Galache, J. A., Sotres, P., Santana, J. R., Gutierrez, V., … others. (2014). SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks*, *61*, 217–238.

Sanchez, S., Angel Sicilia, M., & Rodriguez, D. (2012). *Ingeniería del Sofware. Un enfoque desde la guía SWEBOK*. Alfaomega.

Schauer, P., & Debita, G. (2015). Internet of Things Service Systems Architecture.

Schrott, U. (2017). Austrian hotel experiences 'ransomware of things attack.'

Sefika, M., Sane, A., & Campbell, R. H. (1996). Monitoring compliance of a software system with its high-level design models. In *Proceedings of the 18th international conference on Software engineering* (pp. 387–396).

Seo, S., Kim, J., Yun, S., Huh, J., & Maeng, S. (2015). HePA: Hexagonal Platform Architecture for Smart Home Things. In *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on* (pp. 181–189).

Serbanati, A., Salinas Segura, A., Olivereau, A., Ben Saied, Y., Gruschka, N., Gessner, D., & Gomez-Marmol, F. (2012). *Project Deliverable D4.2 - Concepts and Solutions for Privacy and Security in the Resolution Infrastructure*.

Serna, J., Morales, R., Medina, M., & Luna, J. (2014). Trustworthy communications in Vehicular Ad Hoc NETworks. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on* (pp. 247–252).

Shaw, M. (1989). Larger Scale Systems Require Higher-Level Abstractions. *ACM Sigsoft Software Engineering Notes*, *14*(3), 143–146.

Shen, S., & Carugi, M. (2014). Standardizing the Internet of Things in an evolutionary way. In *ITU Kaleidoscope Academic Conference: Living in a converged world-Impossible without standards?, Proceedings of the 2014* (pp. 249–254).

Shirey, R. (2007). Internet Security Glossary, Version 2.

Shooman, M. L. (1976). Structural models for software reliability prediction. In *Proceedings of the 2nd international conference on Software engineering* (pp. 268–280).

Shrouf, F., Ordieres, J., & Miragliotta, G. (2014). Smart factories in Industry 4.0: A review of the concept and of energy management approached in production based on the Internet of Things paradigm. In *Industrial Engineering and Engineering Management (IEEM), 2014 IEEE International Conference on* (pp. 697–701).

Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks*, *76*, 146–164.

Singh, M., & Bhandari, P. (2016). Building a framework for network security situation awareness. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on* (pp. 2578–2583).

Singh, S., & Singh, N. (2015). Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce. In *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on* (pp. 1577–1581).

Skala, K., Davidovic, D., Afgan, E., Sovic, I., & Sojat, Z. (2015). Scalable Distributed Computing Hierarchy: Cloud, Fog and Dew Computing. *Open Journal of Cloud Computing (OJCC)*, *2*(1), 16–24.

Smartex. (2016). Glossary of terms and expressions used in connection with The Internet of Things with a final section of related 'Standards.' Retrieved from http://www.smartex.com/wp-content/uploads/2016/04/Internet-of-Things-Glossary-of-Terms-V8-draft.pdf

Smith, C. U. (1990). *Performance engineering of software systems*. Addison-

Wesley Longman Publishing Co., Inc.

Software Engineering Institute. (2016). Software Engineering Institute Glossary.

Sommerville, I. (2011). *Ingeniería del Software*. PEARSON.

Sommerville, I., & Sawyer, P. (1997). *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, Inc.

Souza, R., & Cardozo, E. (2016). A Resource-Oriented Architecture for the Internet of Things (IoT). In *Connectivity Frameworks for Smart Devices* (pp. 99–116). Springer.

Statista. (2018). Number of Internet of Things (IoT) devices connected worldwide in 2017 and 2018, by type (in millions).

Stoermer, C., Bachmann, F., & Verhoef, C. (2003). *SACAM: The software architecture comparison analysis method*.

Stojmenovic, I., Wen, S., Huang, X., & Luan, H. (2015). An overview of Fog computing and its security issues. *Concurrency and Computation: Practice and Experience*.

Stoneburner, G., Goguen, A. Y., & Feringa, A. (2002). *SP 800-30. risk management guide for information technology systems*.

Stravoskoufos, K., Sotiriadis, S., & Petrakis, E. (2016). IoT-A and FIWARE: bridging the barriers between the cloud and IoT systems design and implementation. In *Proc. 6th Int'l Conf. Cloud Computing and Services Science* (pp. 146–153).

Subramani, K. S., Antonopoulos, A., Nosratinia, A., & Makris, Y. (2016). Hardware-Induced Security & Privacy Vulnerabilities in the Internet of Things.

Supriya, S., & Padaki, S. (2016). Data Security and Privacy Challenges in Adopting Solutions for IOT. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 410–415). IEEE. https://doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2016.97

Tahir, R., Tahir, H., McDonald-Maier, K., & Fernando, A. (2016). A novel ICMetric based framework for securing the Internet of Things. In *Consumer Electronics (ICCE), 2016 IEEE International Conference on* (pp. 469–470).

Talavera Portocarrero, J. M. (2016). *RAMSES: Reference Architectue of Self-Adaptative Middleware for Wireless Sensor Networks*. Universidade Federal fo Rio de Janeiro.

Techopedia. (2017). What is Modeling Language? Retrieved November 2, 2017, from https://www.techopedia.com/definition/20810/modeling-language

Tekinerdogan, B. (2004). ASAAM: Aspectual software architecture analysis method. In *Software Architecture, 2004. WICSA 2004. Proceedings. Fourth Working IEEE/IFIP Conference on* (pp. 5–14).

Thierer, A. D. (2014). The Internet of Things & Wearable Technology: Addressing Privacy & Security Concerns Without Derailing Innovation. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2494382

Touhill, G. J., & Touhill, J. (2014). *Cybersecurity for Executives: A Practical Guide*. John Wiley & Sons, Inc.

Townsend Security. (2016). *Definitive Guide to Encryption Key Management Fundamentals*. Retrieved from https://info.townsendsecurity.com/definitive-guide-to-encryption-key-management-fundamentals

Trend Micro. (2015). Trend Micro Glossary: Ransomware.

Tuck, M. (2016). Internet of Things: Are We There Yet? (The 2016 IoT Landscape) – Matt Turck. Retrieved July 2, 2017, from http://mattturck.com/2016-iot-landscape/

Tuck, M. (2018). Growing Pains: The 2018 Internet of Things Landscape. Retrieved from http://mattturck.com/iot2018/

Tvedt, R. T., Lindvall, M., & Costa, P. (2002). A process for software architecture evaluation using metrics. In *Software Engineering Workshop, 2002. Proceedings. 27th Annual NASA Goddard/IEEE* (pp. 191–196).

US-CERT. (2016). Alert (TA16-288A) Heightened DDoS Threat Posed by Mirai and Other Botnets.

Usländer, T., & Epple, U. (2015). Reference model of industrie 4.0 service architectures. *At-Automatisierungstechnik*, *63*(10), 858–866.

Van Kranenburg, R. (2008). *The Internet og Things. A critique of ambient technology and the all-seeing network of RFID*. Amsterdam.

Vaquero, L. M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2008). A break in the clouds: Towards a Cloud Definition. *ACM SIGCOMM Computer Communication Review*, *39*(1), 50. https://doi.org/10.1145/1496091.1496100

Verdouw, C. N., Robbemond, R. M., Verwaart, T., Wolfert, J., & Beulens, A. J. M. (2015). A reference architecture for IoT-based logistic information systems in agri-food supply chains. *Enterprise Information Systems*, 1–25.

Weyrich, M., & Ebert, C. (2016). Reference architectures for the internet of things. *IEEE Software*, *33*(1), 112–116.

Williams, L. G., & Smith, C. U. (1998). Performance Engineering of Software Architectures. In *Proceeding on Workshop Software and Performance* (pp. 164–177).

WSO2. (2015). *A Reference Architecture for the Internet of Things*.

Xu, B., Zhang, D., & Yang, W. (2012). Research on architecture of the Internet of Things for grain monitoring in storage. In *Internet of Things* (pp. 431–438). Springer.

Yacoub, S. M., Cukic, B., & Ammar, H. H. (1999). Scenario-based reliability analysis of component-based software. In *Software Reliability Engineering, 1999. Proceedings. 10th International Symposium on* (pp. 22–31).

Yamamoto, Y., Morris, R. V., Hartsough, C., & Callender, E. D. (1982). The role of requirements analysis in the system life cycle. In *Proceedings of the June 7-10, 1982, national computer conference on - AFIPS '82* (p. 381). New York, New York, USA: ACM Press. https://doi.org/10.1145/1500774.1500821

Yang, J., & Fang, B.-X. (2011). Security model and key technologies for the Internet of things. *The Journal of China Universities of Posts and Telecommunications*, *18*(2), 109–112.

Yi, S., Li, C., & Li, Q. (2015). A Survey of Fog Computing: Concepts, Applications

and Issues. In *Mobidata ’15 Proceedings of the 2015 Workshop on Mobile Big Data* (pp. 37–42). ACM. https://doi.org/10.1145/2757384.2757397

York Risk Services Group. (2015). No Business is too small for a cyber-attack.

Zegzhda, D., & Stepanova, T. (2015). Achieving Internet of Things security via providing topological sustainability. In *Science and Information Conference (SAI), 2015* (pp. 269–276).

Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., & Shen, X. S. (2017). Security and Privacy in Smart City Applications: Challenges and Solutions. *IEEE Communications Magazine*, *55*(1), 122–129. https://doi.org/10.1109/MCOM.2017.1600267CM

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, *1*(1), 7–18. https://doi.org/10.1007/s13174-010-0007-6

ANEXO A – EVALUACIÓN DE ARQUITECTURAS DE REFERENCIA.

En esta sección se evalúa que tan completa son las arquitecturas de referencia seleccionadas y estudiadas en el capítulo 4. Para realizar esta medición se utilizó como fuente de medición el RaModel, un modelo de referencia que establece 17 elementos que una arquitectura debe considerar para considerarse como una arquitectura de referencia. Con los grupos y elementos considerados por el RaModel se creó un instrumento de medición, el cual se muestra en la Tabla 33. Para cuantificar el grado de completitud se usó la siguiente escala de medición: un puntaje de 1.0 para aquellos elementos que se cumplen totalmente; y un puntaje de 0.5 para aquellos elementos que se cumplen parcialmente.

Tabla 33 - Evaluación de arquitecturas de referencia usando el RAModel

Grupos y elementos del RAModel		**Arquitecturas de referencia**					
		AR1	**AR2**	**AR3**	**AR4**	**AR5**	**AR6**
Elementos del grupo de dominio	Legislación, estándar y regulación	1.0	1.0	1.0	-	-	-
	Atributos de calidad	-	1.0	1.0	1.0	-	1.0
	Cumplimiento del sistema	-	-	0.5	-	-	-
Elementos del grupo de aplicación	Alcance	1.0	1.0	1.0	1.0	-	1.0
	Requisitos funcionales	1.0	1.0	1.0	1.0	-	1.0
	Datos del dominio	-	-	1.0	1.0	-	-
	Restricciones	-	-	1.0	-	-	-
	Riesgos	-	-	1.0	-	-	1.0
	Objetivos y necesidades	1.0	1.0	1.0	-	1.0	1.0
	Limitaciones	0.5	-	0.5	-	-	-
Elementos del grupo de infraestructura	Elementos de software	-	-	1.0	1.0	1.0	-
	Elementos de hardware	1.0	1.0	1.0	1.0	1.0	1.0
	Mejores prácticas y lineamientos	0.5	-	1.0	1.0		0.5
	Estilos arquitecturales	1.0	1.0	1.0	1.0	1.0	1.0
Elementos del grupo de elementos transversales	Decisión	-	-	1.0	-	-	1.0
	Terminología del dominio	1.0	1.0	1.0	-	0.5	-
	Comunicación	-	1.0	1.0	1.0	-	1.0
Total		**8**	**9**	**16**	**9**	**4.5**	**9.5**

Fuente: Autor.

AR1 - Internet of Things – The inclusive model de CASAGRAS Project
AR2 - Modelo de referencia de IoT de la ITU-T
AR3 - Internet of Things Architecture – IoT-A Project
AR4 - Third Cycle Architecture de SmartSantander Project
AR5 - Internet of Things Reference Model de Cisco
AR6 - Reference Architecture for Internet of Things de WSO2

ANEXO B – MODELO DE GESTIÓN DE LA CIBERSEGURIDAD PARA APLICACIONES IOT

Security Management in Internet of THings Model – SMITH Model

Versión 1.0, 2018

1. Introducción

La seguridad es uno de los restos por resolver que tiene el Internet de las Cosas, IoT por sus siglas en inglés, *Internet of Things*. La variedad fabricantes y heterogeneidad de tecnologías hardware y software que hay en el ecosistema IoT hace que gestionar su seguridad sea más complejo, sumado a esto, la escasez de profesionales en el área de la ciberseguridad dificulta que las buenas prácticas de seguridad se implementen en las aplicaciones IoT.

Las aplicaciones IoT pueden generar un gran volumen de información, que pueden ser muy sensible, que debe ser protegida de ser acceda y modificada terceros no autorizados o que estos afecten su disponibilidad para los usuarios autorizados.

Con el fin de contribuir a la seguridad de IoT se han propuesto *frameworks* y modelos que abordan algunas áreas de la ciberseguridad, pero hay un vacío de propuestas que orienten al equipo desarrollador de aplicaciones IoT en la implementación de medidas de ciberseguridad desde la fase de diseño de esta. Por esta razón se propone el ***S****ecurity* ***M****anagement in* ***I****nternet of* ***TH****ings Model – SMITH Model*.

2. Propósito

El *SMITH Model* apunta a orientar a los desarrolladores de aplicaciones IoT en la implementación de las buenas prácticas de ciberseguridad desde la fase de diseño de estas.

SMITH Model pretende ser una guía de fácil lectura en que los desarrolladores de aplicaciones IoT encuentren las consideraciones de ciberseguridad que deben tener en cuenta independiente de las tecnologías que ellos usen. Por esta razón este modelo se basa en una arquitectura genérica de IoT diseñada a partir de arquitecturas de referencia para IoT propuestas por instituciones reconocidas en el área de la tecnología.

3. Alcance

El alcance que tiene el *SMITH Model* en su versión 1.0 tiene que ver con los elementos de ciberseguridad que se tratan en dicho modelo. Aunque en el modelo se representan los elementos de ciberseguridad necesarios, para la versión 1.0 no se describirán todos ellos.

Para la versión 1.0 se considera los elementos de la cara frontal, la cara superior y de la capa lateral solo se considera la capa de Tecnología. Las capas de Procesos y Prácticas se considerarán en una futura versión del modelo.

4. Documentos relacionados

El *SMITH Model* está compuesto por una serie de documentos que fundamentan y apoyan el modelo de ciberseguridad descrito en este documento, los cuales se mencionan a continuación:

- La *Guía de buenas prácticas para el aseguramiento de aplicaciones IoT*, que reúne recomendaciones de seguridad y se puede acceder desde el siguiente enlace.
- El documento de *Evaluación de elementos de ciberseguridad en aplicaciones IoT* presenta unas herramientas para determinar qué elementos y buenas prácticas de ciberseguridad son consideradas en la aplicación IoT, y se puede consultar en el siguiente enlace.
- La arquitectura genérica para aplicaciones IoT, el cual se puede consultar en el siguiente enlace.
- Los requisitos de seguridad para aplicaciones IoT, el cual se puede consultar en el siguiente enlace.

5. Descripción del *SMITH Model*

El *SMITH Model* es una representación del dominio de la ciberseguridad para las aplicaciones IoT. Los diferentes elementos que integran el dominio de ciberseguridad se representan a través de un cubo donde cada una de las caras, y sus respectivas capas, cuentan con los elementos que intervienen en este dominio, como se muestra en la Figura 1.

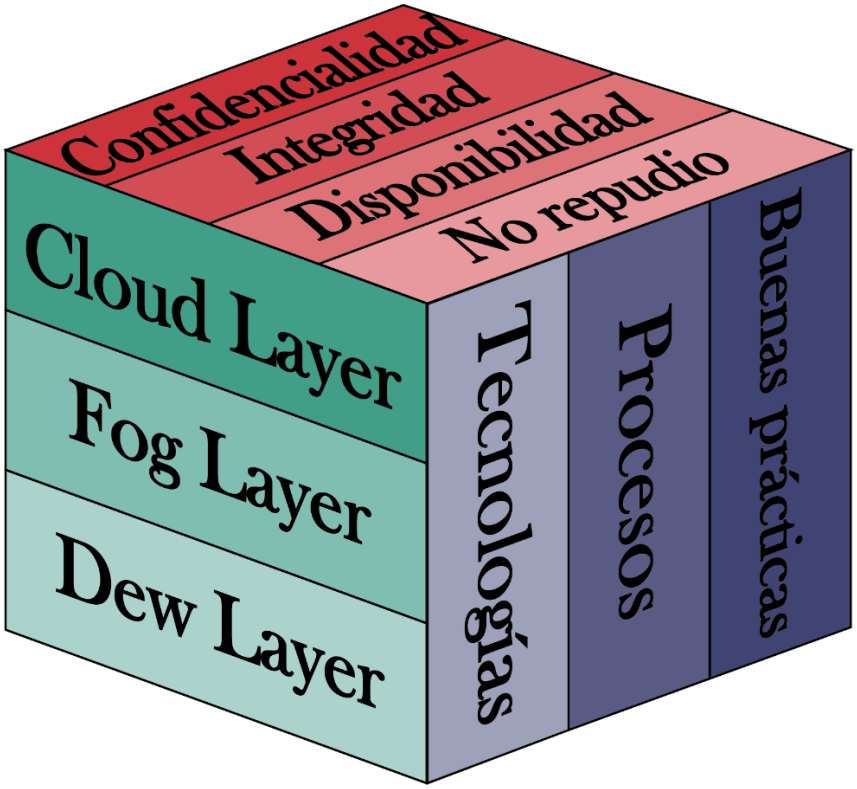


Figura 1. *SMITH Model*

5.1 Vista frontal

La cara frontal del modelo se consideran las funcionalidades y capacidades de una aplicación IoT a través de una arquitectura IoT genérica propuesta por los autores del modelo[21], el cual está compuesta por tres capas: *Cloud Layer*, *Fog Layer* y *Dew Layer*, como se muestra en la Figura 2.

5.1.1 ***Cloud Layer***. En esta capa se encuentra la mayor parte de la infraestructura de la aplicación IoT y cuenta con capacidades y funcionalidades como la visualización y accesibilidad a recursos, servicios y dato; las API para permitir y facilitar a otras aplicaciones y desarrolladores acceder a recursos de la aplicación IoT; el almacenamiento, el procesamiento de grandes volúmenes de datos, la gestión de servicios y eventos.

[21] En la sección 4.2.2 de este documento se presenta y detalla la arquitectura genérica para IoT propuesta por los autores de este trabajo.

Figura 2 - Capacidades y funcionalidades de la *Cloud Layer*.

5.1.2 *Fog Layer*. En esta capa se encuentran las capacidades y funcionalidades relacionadas con la comunicación de datos, la gestión de fallos, la gestión de topología de red y la agregación de datos, como se muestra en la Figura 3.

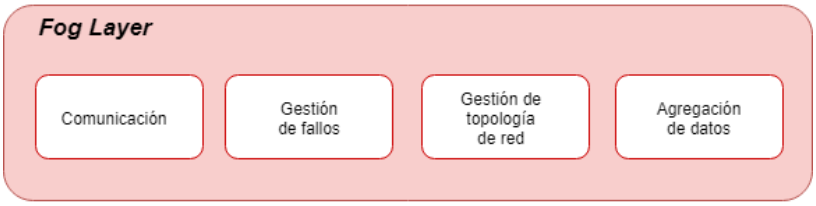


Figura 3 - Capacidades y funcionalidades de la *Fog Layer*.

5.1.3 *Dew Layer*. Esta capa se encuentran las capacidades y funcionalidades relacionadas con los dispositivos como son la gestión de dispositivos, para la activación y desactivación de dispositivos de forma remota, la gestión del estado de trabajo del dispositivo, su diagnóstico, la actualización de firmware y/o software; también soporta las diferentes interfaces de los dispositivos conectados mediante diferentes tecnologías; la gestión de datos y el procesamiento de datos a nivel inferior, como se muestra en la Figura 4.

Figura 4 - Capacidades y funcionalidades de la *Dew Layer*.

5.2 Vista superior

Esta cara está dividida en cuatro capas transversales las cuales están relacionadas con las propiedades de la información que deben protegidas a través de la ciberseguridad. Estas propiedades son la confidencialidad, la integridad, la disponibilidad y el no repudio. En cada una de las capas se considera un grupo de requisitos de calidad relacionados con la seguridad que

deben ser considerados en una aplicación IoT para proteger de la información generada.

5.2.1 Capa de confidencialidad. En esta capa se consideran el grupo de requisitos para garantizar la confidencialidad de la información. La confidencialidad es la propiedad que protege los datos de la divulgación o acceso a qué personas, procesos o aplicaciones no autorizados.

5.2.2 Capa de integridad. En esta capa se consideran el grupo de requisitos para garantizar la integridad de la información. Esta propiedad busca proteger que los datos no han sido modificados por personas, procesos o sistemas no autorizadas, de esta forma se garantiza al receptor que los datos recibidos coinciden con los enviados por el emisor, permitiendo detectar si se produjo un cambio en los mismos durante su transmisión, ya sea un cambio, o que se añada o elimine parte del mensaje.

5.2.3 Capa de disponibilidad. En esta capa se consideran el grupo de requisitos para garantizar la disponibilidad de la información. Esta propiedad considera las medidas necesarias para que la información esté disponible para quienes estén autorizados a acceder a ella, ya sea persona, proceso o sistema.

5.2.4 Capa de no repudio. En esta capa se consideran el grupo de requisitos para garantizar el no repudio. Esta propiedad lo que busca es proteger la participación de alguna de las partes que intervienen en la comunicación, ya sea en una toda o una parte de ella.

Para ello, el no repudio garantiza que se pueda probar el origen, el destino y la entrega del mensaje. De esta forma, en la prueba de origen el receptor puede demostrar ante terceros el origen de los datos recibidos; para la prueba de destino, el emisor de mensaje puede demostrar a terceros que los datos han sido entregados al emisor adecuado; finalmente, en la prueba de entrega, el emisor o receptor pueden demostrar ante terceros la fecha y hora en que se envió el mensaje.

Adicionalmente se enumeran un grupo de requisitos relacionados con la seguridad, el Grupo de requisitos para la infraestructura de seguridad – GRIS que reúne los requisitos que deben cumplir la infraestructura de seguridad de una aplicación IoT.

En el documento *Requisitos de seguridad para aplicaciones IoT* se puede apreciar los grupos de requisitos de seguridad que deben considerarse en una aplicación IoT. Puede acceder ingresando en el siguiente enlace.

5.3 Vista lateral

En esta cara del modelo se consideran los elementos que intervienen en la ciberseguridad: las tecnologías, los procesos y las prácticas. Estos tres elementos son usados para proteger los activos de la información a través del tratamiento de amenazas que ponen en riesgo la información.

5.3.1 Tecnologías. Este elemento es el más técnico y abarcas todas aquellas tecnologías hardware y software que se utilizan para asegurar los activos de la información, salvaguardando la confidencialidad, integridad, disponibilidad y no repudio de la información

5.3.2 Procesos. Otro factor de importante en la gestión de la ciberseguridad es establecer los procesos a realizar, esto se realiza a través de políticas organizacionales. Estos procesos o políticas se establecen desde un nivel más administrativo, con un alto compromiso de la gerencia para que estas políticas se implementen de forma adecuada y sean acatadas por cada una de las personas que intervienen en ellos.

5.3.3 Prácticas. En la ciberseguridad las forma en que los usuarios usan e interactúan con la tecnología es muy relevante para proteger los activos de la información. Por esta razón, capacitar el usuarios en las buenas prácticas de ciberseguridad es de vital importancia dentro de las organizaciones. En el ámbito de la ciberseguridad se dice que el usuario ‘es el eslabón más débil de la cadena’, que de nada sirve invertir recursos tecnológicos, hardware y software, para proteger los activos de información si las personas no están capacitadas para responder ante los incidentes de seguridad, aumentando el riesgo de ser víctimas de técnicas como la ingeniería social, y se conviertan en la puerta de entrada para que los cibercriminales ingresen a la infraestructura tecnológica de la organización.

Guía de buenas prácticas para el aseguramiento de aplicaciones IoT

Security Management in Internet of THings Model – SMITH Model

Versión 1.0, 2018

1. Introducción

La seguridad es uno de los restos por resolver que tiene el Internet de las Cosas, IoT por sus siglas en inglés, *Internet of Things*. La variedad fabricantes y heterogeneidad de tecnologías hardware y software que hay en el ecosistema IoT hace que gestionar su seguridad sea más complejo, sumado a esto, la escasez de profesionales en el área de la ciberseguridad dificulta que las buenas prácticas de seguridad se implementen en las aplicaciones IoT.

Las aplicaciones IoT pueden generar un gran volumen de información, que pueden ser muy sensible, que debe ser protegida de ser acceda y modificada terceros no autorizados o que estos afecten su disponibilidad para los usuarios autorizados.

Con el fin de contribuir a la seguridad de IoT se han propuesto *frameworks* y modelos que abordan algunas áreas de la ciberseguridad, pero hay un vacío de propuestas que orienten al equipo desarrollador de aplicaciones IoT en la implementación de medidas de ciberseguridad desde la fase de diseño de esta. Por esta razón se propone el ***S****ecurity* ***M****anagement in* ***I****nternet of* ***TH****ings Model – SMITH Model*.

Como parte del *SMITH Model* se propone la *Guía de buenas prácticas para el aseguramiento de aplicaciones IoT* que reúne buenas prácticas de ciberseguridad para aplicaciones IoT.

2. Propósito

La *Guía de buenas prácticas para el aseguramiento de aplicaciones IoT* pretende ser un documento que recopile recomendaciones de ciberseguridad, concreto y de fácil lectura, que oriente a los desarrolladores de aplicaciones IoT sobre qué consideraciones deben tener en cuenta para salvaguardar la información generada, transportada y almacenada en sus soluciones independientemente de las tecnologías que ellos usen.

3. Alcance

La *Guía de buenas prácticas para el aseguramiento de aplicaciones IoT* en un documento cuyo propósito es orientar a los desarrolladores sobre qué elementos de ciberseguridad deben atender en sus aplicaciones, por esta razón, no es un documento técnico, ni aborda los temas en profundidad.

Esta guía se enfoca en el qué se debe proteger y el cómo se debe proteger, pero no en las orientaciones técnicas de las mismas, pero sí se sugiere a los desarrolladores dónde pueden profundizar más sobre dichos temas.

4. Referencias y documentos relacionados.

Esta guía se basa en las recomendaciones de seguridad realizadas por OWASP en su documento *IoT Framework Assessment* (2016a).

El guía hace parte del *SMITH Model*, el cual está compuesto por una serie de documentos que fundamentan y apoyan dicho modelo, los cuales se mencionan a continuación:

- El ***S**ecurity **M**anagement in **I**nternet of **TH**ings Model – SMITH Model*, que presenta una abstracción del dominio de la ciberseguridad para aplicaciones IoT, el cual se puede acceder desde el siguiente enlace.
- El documento de *Evaluación de elementos de ciberseguridad en aplicaciones IoT* presenta unas herramientas para determinar qué elementos y buenas prácticas de ciberseguridad son consideradas en la aplicación IoT, y se puede consultar en el siguiente enlace.
- La arquitectura genérica para aplicaciones IoT, el cual se puede consultar en el siguiente enlace.
- Los requisitos de seguridad para aplicaciones IoT, el cual se puede consultar en el siguiente enlace.

5. Recomendaciones de ciberseguridad para aplicaciones IoT

Las recomendaciones de ciberseguridad están organizadas según la vista frontal del *SMITH Model*, la cual está compuesta en tres capas: *Cloud Layer*, *Fog Layer* y *Dew Layer*.

5.1 Buenas prácticas de ciberseguridad para *Cloud Layer*. En esta capa se encuentra la mayor infraestructura de la aplicación IoT y abarca funcionalidades como las que se describen en la Figura 1.

Figura 1 - Capacidades y funcionalidades de la *Cloud Layer*.

Comunicaciones cifradas

Esta capa debe admitir las comunicaciones cifradas, incluidos los certificados de seguridad, para identificarse con las otras componentes de la aplicación. De igual forma, se debe admitir certificados criptográficos para identificar otros componentes también, para la verificación de identidad bidireccional.

Autenticación

Los elementos de esta capa que requieran de autenticación deben permitir una autenticación compleja, incluida la autenticación de múltiples factores, evitando las credenciales predeterminadas. La interfaz también debe considerar algunos mecanismos de seguridad como de mitigación de fuerza bruta y enumeración contraria a la cuenta y debería permitir a los usuarios establecer fácilmente y restablecer de forma segura la información de la cuenta.

Credenciales de autenticación segura

Las credenciales de autenticación, en cualquier forma como son contraseñas, identificaciones del dispositivo, entre otros, deben ser apropiadamente codificados y cifrados antes de almacenarlos; y donde dichos mecanismos de almacenamiento también deben ser uniformemente fuertes.

Para conocer más sobre el almacenamiento de contraseñas puede consultar la *Password Storage Cheat Sheet* (2017c) en el siguiente enlace.

Almacenamiento cifrado y capacidad de clasificación de datos y segregación

Las aplicaciones IoT manejan un gran volumen de datos, y estos pueden contener información sensible sobre los usuarios, por esta razón, siempre que

sea posible, el marco debe admitir el cifrado de datos que es almacenada, así como en cualquier mecanismo de exportación o copia de seguridad de dichos datos.

Este volumen de datos contempla una variedad de datos, algunos datos pueden ser muy sensibles y otros datos pueden ser benignos. Se debería proporcionar las capacidades para clasificar los datos y protegerlos dependiendo de la clasificación que se les dé. Se deben implementar controles que limiten el acceso y la exposición de datos confidenciales según la clasificación.

Capacidad de utilizar comunicaciones encriptadas entre componentes

Las comunicaciones entre los diferentes componentes de esta capa y otras capas deben utilizar un canal de comunicaciones cifrado para evitar que los datos se expongan en tránsito.

Informes y alertas de eventos de seguridad

En esta capa se concentra la mayor parte de la infraestructura de la aplicación IoT, y cuya capacidad hardware permite obtener grandes recursos de cómputo y almacenamiento, por lo tanto, es la capa con mayor capacidad de gestión de recursos y seguridad, sobre ella misma y la otras capas.

Esta debe contar con sólidas funciones de supervisión, generación de informes y alertas de eventos de seguridad. A esta capa se debe proveer de funciones que permitan detectar y reaccionar a la actividad maliciosa; además, debe poder segregar a los malos actores, limitar el acceso a partes maliciosas e integrarse fácilmente con los sistemas de registro y prevención de intrusión y registro de terceros.

Actualizaciones automáticas y verificación de actualización

Una buena práctica para reducir los riesgos de seguridad es mantener el software actualizado y permitir parches y actualizaciones es fundamental. Se debe identificar claramente la versión del software en funcionamiento y permitir parches y actualizaciones de software para componente de esta capa. Un proceso de actualización automática aumenta la probabilidad de que los sistemas se mantengan actualizados. Se sebe generar alertas automáticas de actualizaciones para los componentes que no se actualizan automáticamente.

Utilice los últimos componentes de terceros actualizados

Para el *Cloud* hay muchos componentes, plataformas y servicios desarrollados por terceros que pueden usarse en nuestras propias soluciones, y respecto a

esto, se debe implementar los desarrollos más actualizados. También se debe proveer la capacidad de mantenerse informado sobre nuevas versiones y actualizaciones de estos componentes, para su instalación a medida que se degradan las actualizaciones de seguridad que estén disponibles. En cuando a la hora de actualizar, se debe garantizar que las actualizaciones se distribuyan a través de un canal seguro y se verifiquen antes de la instalación, para descartar que se ha descargado la versión original ofrecida por el emisario.

5.2 Buenas prácticas de ciberseguridad para *Fog Layer*. En esta capa se encuentran las funcionalidades relacionadas con la comunicación. Esta capa comunica la *Fog Layer* con la *Cloud Layer* y reúne funcionalidades como las que se muestran en la Figura 2.

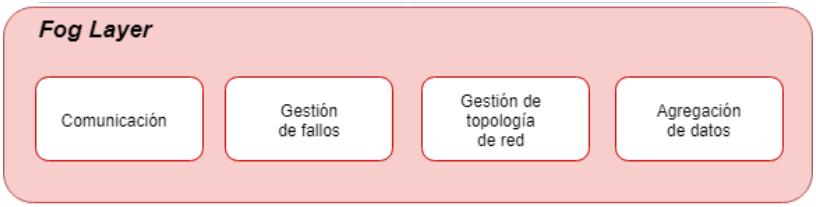


Figura 2 - Capacidades y funcionalidades de la *Fog Layer*.

Comunicaciones cifradas multidireccionales

Esta capa se encarga de la comunicación entre las capas *Cloud* y *Dew*, y se debe velar porque que las comunicaciones sean seguras para no degradar la seguridad de los mensajes en cualquier dirección siempre que sea posible. Se puede presentar que existan los canales de comunicaciones seguros y no seguros, en cuyo caso se debe prestar especial atención a la interceptación, manipulación e inyección de datos en puntos finales inseguros. También se debe proporcionar capacidades para segmentar y aislar las comunicaciones cuando sea posible.

Autenticación fuerte de componentes

Los componentes de esta capa deberían proporcionar mecanismos de autenticación tan sólidos como cualquier otro componente en la aplicación. Siempre que sea posible, los componentes de esta capa deben autenticarse de forma multidireccional para garantizar comunicaciones de confianza con las capas *Cloud* y *Dew*. Las capacidades criptográficas en la autenticación deberían ser un componente sólido de la solución de infraestructura.

Almacenamiento

La puerta de enlace puede servir como un único punto de falla, o ataque, en el ecosistema y debe almacenar solo la cantidad mínima de información, en un formato encriptado si es posible.

Registro y alerta

La puerta de enlace tendrá acceso a un volumen de tráfico y debería poder registrar y alertar en función del registro de eventos. El marco puede incluir integración con servicios de registro estándar o sistemas de detección de intrusos. Incluso se podría admitir métodos alternativos para alertar en la puerta de enlace, como, por ejemplo, los SMS.

Capacidades de detección e informe de anomalías

Se debería permitir que se observe y monitoree el tráfico de comunicaciones y el comportamiento de los componentes. En esta capa se ubican los elementos adecuados para controlar el tráfico hacia y desde la nube y aquellos elementos que deben admitir la detección de anomalías o integrarse fácilmente con anomalías y sistemas de detección de intrusos. Con el manejo de elementos robustos incluso se podría admitir capacidades de prevención de intrusos para excluir a los actores sospechosos del ecosistema.

Utilice los últimos componentes de terceros actualizados

En esta capa, muchos de los elementos son desarrollados por terceros, y respecto a esto, se debe implementar los desarrollos más actualizados. También se debe proveer la capacidad de mantenerse informado sobre nuevas versiones y actualizaciones de estos componentes, para su instalación a medida que se degradan las actualizaciones de seguridad que estén disponibles. En cuando a la hora de actualizar, se debe garantizar que las actualizaciones se distribuyan a través de un canal seguro y se verifiquen antes de la instalación, para descartar que se ha descargado la versión original ofrecida por el emisario.

Actualizaciones automáticas y/o informes de versión

Una buena práctica para reducir los riesgos de seguridad es mantener el software actualizado. Se debe identificar claramente la versión del software en funcionamiento y permitir parches y actualizaciones de software. Un proceso de actualización automática aumenta la probabilidad de que los sistemas se mantengan actualizados.

5.3 Buenas prácticas de ciberseguridad para *Dew Layer*. En esta capa se encuentran los dispositivos IoT, que son heterogéneos y pueden diferir en sus capacidades de cómputo, almacenamiento, comunicación y energía; la cantidad de ellos puede variar dependiendo de la aplicación y reúne funcionalidades como las que se muestran en la Figura 3.

Figura 3 - Capacidades y funcionalidades de la *Dew Layer*.

Las recomendaciones de ciberseguridad se clasificaron en tres categorías: recomendaciones generales, recomendaciones para los dispositivos y las recomendaciones relacionadas con las comunicaciones.

- **Recomendaciones generales**

A continuación, se presentan las recomendaciones generales.

Registro robusto de eventos

Se debería ofrecer un registro robusto de eventos realizadas en el sistema, incluido el registro de eventos de seguridad. Estos registros deben ser personalizables y deben informar eventos sensibles en un formato utilizable para usuarios finales, gerentes y operadores; los cuales también pueden ser útiles al proporcionar evidencia forense en caso de incidentes de seguridad.

Capacidades de identificación criptográfica

Es importante proveer a los componentes de esta capa de capacidades criptográficas para verificar la confianza de otros componentes de la aplicación IoT, como *gateways*, la *cloud* y dispositivos móviles, y la gestión del ciclo de vida criptográfico. Gestionar el ciclo de vida criptográfico implica respaldar la emisión y reedición de material criptográfico, el vencimiento de certificados criptográficos, un mecanismo de verificación de revocación, y un sistema de firma de material clave.

Cifrado de almacenamiento

Muchos dispositivos cuentan con la capacidad hardware de almacenar datos. Los datos confidenciales pueden ser susceptibles de robo o exposición a menos que se almacene con las consideraciones de seguridad adecuadas; estos datos pueden incluir lectura del sensor, configuración, credenciales de autenticación o claves criptográficas.

Autenticación local robusta y soporte de autenticación de múltiples factores

El uso de contraseñas es común en muchos servicios y recursos; se deben cambiar las credenciales de acceso por defecto; las contraseñas, en la medida de lo posible, deben ser complejas y se debe contar con autenticación de múltiples factores. Asimismo, los componentes desarrollados o usados que cuentes con mecanismo de autenticación deben informar y registrar intentos de autenticación fallidos y proporcionar un mecanismo de retardo o bloqueo exponencial para evitar ataques de fuerza bruta.

Capacidades M2M seguras

La comunicación maquina a maquita (M2M, *Machite-to-Machine*) debe basarse en la confianza a través de capacidades como la autorización, la verificación y la autenticación entre máquinas que hacen parte de la red. Estas capacidades, en la medida de lo posible, debe extenderse al estado *offline* para evitar fallar al momento de perder la conexión. La confianza en las comunicaciones M2M podría gestionarse con la confianza transitiva, de este modo, un propietario podría certificar una cantidad de dispositivos que luego podrían autenticarse y confiar en función del propietario, independientemente del dispositivo o plataforma del ecosistema.

Interfaz web segura

Algunos dispositivos IoT tienen la capacidad hardware para ofrecer servicios web, y estos servicios, aunque no sean el servicio principal de la aplicación también deben asegurarse. Se debe garantizar que estas aplicaciones web implementen contramedidas de seguridad contra vulnerabilidades comunes tales como omisión de autenticación, scripts entre sitios y falsificación de solicitudes entre sitios. Las interfaces web deben presentarse sobre el protocolo TLS (HTTPS) y no deben usar certificados autofirmados o no válidos. También se debería limitar el acceso a la interfaz web para evitar el uso o abuso no autorizado.

Para más información, puede consultar este mismo apartado en la *Cloud Layer*.

Utilice pilas y protocolos de red establecidos y probados

En las aplicaciones se debe utilizar protocolos y pilas de protocolos de red que cuenten con buen soporte para evitar vulnerabilidades de seguridad comunes que pueden presentarse en otros más nuevos, que no han sido probados ampliamente o son exóticos. Para limitar la superficie de ataque se deben limitar el número de protocolos al mínimo posible y desactivar los que no se estén usando.

Utilice los últimos componentes de terceros actualizados

En el ecosistema IoT hay muchos componentes, plataformas y servicios desarrollados por terceros que pueden usarse en las propias soluciones, y respecto a esto, se debe implementar los desarrollos más actualizados. También se debe proveer la capacidad de mantenerse informado sobre nuevas versiones y actualizaciones de estos componentes, para su instalación a medida que se degradan las actualizaciones de seguridad que estén disponibles. En cuando a la hora de actualizar, se debe garantizar que las actualizaciones se distribuyan a través de un canal seguro y se verifiquen antes de la instalación, para descartar que se ha descargado la versión original ofrecida por el emisario.

Se recomienda usar un lenguaje de programación seguro o sujeto a escrutinio

Los componentes deben estar escritos en lenguajes de programación que posean contramedidas de seguridad y demuestren un historial de seguridad sólida.

- **Recomendaciones relacionadas con los dispositivos**

A continuación, se presentan las recomendaciones relacionadas con los dispositivos IoT.

Actualizaciones automáticas y/o informes de versión

Una buena práctica para reducir los riesgos de seguridad es mantener el software actualizado y permitir parches y actualizaciones es fundamental. Se debe identificar claramente la versión del software en funcionamiento y permitir

parches y actualizaciones de software. Un proceso de actualización automática aumenta la probabilidad de que los sistemas se mantengan actualizados.

Verificación de actualización

Las actualizaciones deben entregarse a través de un canal seguro y verificarse después de la descarga para garantizar que las actualizaciones sean legítimas. Los valores hash binarios de firma (y comprobación) y actualización entregados a través de un canal verificado y encriptado garantizan que las actualizaciones maliciosas no estén instaladas en un dispositivo. Tenga en cuenta que el acceso físico puede permitir que un atacante cargue un binario para colocarlo directamente en un dispositivo, por lo que las actualizaciones deben verificarse antes de la instalación, en lugar de simplemente verificar una descarga.

Sin contraseñas predeterminadas

Se debe evitar las credenciales por defecto en todo el ecosistema, ya sean en componentes de autenticación local, así como componentes de autenticación para la nube, *gateway*, el dispositivo móvil u otros dispositivos del ecosistema. Las credenciales se deben establecer y restablecer cada cierto tiempo, y se debe establecer una política para el manejo de estas.

Las interfaces están deshabilitadas por defecto

Es recomendable desactivar la mayor cantidad posible de servicios y características de manera predeterminada del sistema, e ir habilitando dichas funciones según sea necesario para minimizar la superficie de ataque.

Capacidades de control y gestión del dispositivo

En cuanto a la gestión de dispositivos, se debería permitir la supervisión de su plataforma, y en lo posible, la administración de capacidades que permitan la detección de debilidades de seguridad o vulnerabilidades.

Funciones de seguridad fuera de línea

El marco debe asumir que el componente de borde puede perder conectividad y recurrir a características de seguridad locales en ausencia de recursos de red. Estas características de seguridad fuera de línea deben ser tan sólidas como las características en línea para evitar que los atacantes interrumpan las comunicaciones y degraden las medidas de seguridad.

Consideraciones de propiedad transitiva

Los dispositivos de IoT deben gestionarse ante el cambio de propietario. Se debería permitir que el dispositivo sea borrado, reiniciado o y que los datos sean destruidos para proteger la información del propietario anterior.

- **Recomendaciones relacionadas con las comunicaciones**

A continuación, se presenta las recomendaciones relacionadas con las comunicaciones entre los dispositivos y otros componentes de esta capa.

Cifrado de comunicaciones

Siempre que sea posible se debe proveer comunicaciones cifradas de extremo a extremo para garantizar que estas no puedan ser interceptadas o redirigidas.

Se debe rastrear y contener datos de fuentes potencialmente contaminadas o inseguras

Es posible que se requiera que los dispositivos de IoT procesen datos de canales que no se pueden proteger. El marco debería permitir alguna forma de etiquetado de datos o sanitización para rastrear y contener datos de fuentes no confiables.

Evaluación de elementos de ciberseguridad en aplicaciones IoT

Security Management in Internet of THings Model – SMITH Model

Versión 1.0, 2018

1. Introducción

La seguridad es uno de los restos por resolver que tiene el Internet de las Cosas, IoT por sus siglas en inglés, *Internet of Things*. La variedad fabricantes y heterogeneidad de tecnologías hardware y software que hay en el ecosistema IoT hace que gestionar su seguridad sea más complejo, sumado a esto, la escasez de profesionales en el área de la ciberseguridad dificulta que las buenas prácticas de seguridad se implementen en las aplicaciones IoT.

Las aplicaciones IoT pueden generar un gran volumen de información, que pueden ser muy sensible, que debe ser protegida de ser acceda y modificada terceros no autorizados o que estos afecten su disponibilidad para los usuarios autorizados.

Con el fin de contribuir a la seguridad de IoT se han propuesto *frameworks* y modelos que abordan algunas áreas de la ciberseguridad, pero hay un vacío de propuestas que orienten al equipo desarrollador de aplicaciones IoT en la implementación de medidas de ciberseguridad desde la fase de diseño de esta. Por esta razón se propone el ***S****ecurity* ***M****anagement in* ***I****nternet of* ***TH****ings Model – SMITH Model*.

Como parte del *SMITH Model* se presenta un instrumento para la *Evaluación de elementos de ciberseguridad en aplicaciones IoT* que enlista buenas prácticas de ciberseguridad que deben considerarse en aplicaciones IoT.

2. Propósito

Este instrumento para la evaluación de elementos de ciberseguridad en aplicaciones IoT enlista buenas prácticas de ciberseguridad que deben considerarse en aplicaciones IoT y su propósito es facilitar a los desarrolladores la evaluación de diferentes elementos de ciberseguridad para determinar cuáles no se han considerado en sus soluciones y puedan tomar acciones de mejora.

3. Instrumentos de evaluación

Estos instrumentos de evaluación permiten a los desarrolladores de aplicaciones IoT evaluar qué elementos de ciberseguridad, básicos y de forma general, no están considerando con el fin de salvaguardar la información generada en sus soluciones.

Este instrumento está organizado en tres listas de chequeo, una por cada capa que componen la vista frontal del *SMITH Model*.

3.1 Instrumento de evaluación para *Cloud Layer*

Controles de ciberseguridad para *Cloud Layer*	
Control	**Cumple**
Comunicaciones cifradas	
¿La aplicación cuenta con cifrado en las comunicaciones?	
Autenticación	
¿Se contempla autenticación es servicios y componentes del sistema?	
¿Se incluye autenticación de múltiples factores?	
¿La interfaz incluye funciones de mitigación de fuerza bruta?	
¿Se cambiaron las credenciales por defecto?	
¿Se permite a los usuarios establecer fácilmente y restablecer de forma segura la información de la cuenta?	
Credenciales de autenticación segura	
¿Las credenciales de autenticación se codifican y se cifran apropiadamente antes de almacenarlos?	
Almacenamiento cifrado	
¿Se implementa el cifrado de datos en reposo?	
¿Se cifra cualquier mecanismo de exportación o copia de seguridad?	
Informes y alertas de eventos de seguridad	
¿Se implementan funciones de supervisión, generación de informes y alertas de eventos de seguridad?	
Actualizaciones automáticas y verificación de actualización	
¿Se implementa las actualizaciones automáticas?	
¿Se implementa la verificación de actualizaciones?	
Utilice los últimos componentes de terceros actualizados	
¿Componentes de terceros usados en la aplicación están actualizados?	
¿Se reciben notificaciones cuando los terceros ofrecen actualizaciones de software y de seguridad de sus componentes?	
¿Las actualizaciones se distribuyen a través de un canal seguro y se verifiquen antes de la instalación?	

3.2 Instrumento de evaluación para *Fog Layer*

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
Comunicaciones cifradas multidireccionales	
¿Se cifran las comunicaciones?	
¿Se gestiona la seguridad al interactuar con canales de comunicación inseguros para prevenir, entre otras cosas, la interceptación, manipulación e inyección de datos?	

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
Autenticación fuerte de componentes (borde, plataforma, usuario)	
¿Se proporcionan mecanismos de autenticación multidireccional con las otras capas de la aplicación?	
¿Las capacidades criptográficas en la autenticación son un componente sólido de la solución?	
Almacenamiento	
¿La puerta de enlace almacena solo la cantidad mínima de información, en un formato cifrado, de ser posible?	
Registro y alerta	
¿La puerta de enlace tiene acceso al volumen de tráfico y puede registrar y alertar en función del registro de eventos?	
Capacidades de detección e informe de anomalías	
¿Se permite que se observe y monitoree el tráfico de comunicaciones y el comportamiento de los componentes?	
¿Existen elementos para la detección de anomalías y la detección de intrusos?	
¿Estos elementos poseen la capacidad de prevención de intrusos para excluir a los actores sospechosos del ecosistema?	
Utilice los últimos componentes de terceros actualizados	
¿Se están usando las versiones más recientes de los componentes, plataformas y servicios desarrollados por terceros?	
¿Se provee la capacidad de mantenerse informado sobre nuevas versiones de software y actualizaciones de seguridad de los componentes a medida que estén disponibles?	
¿Las actualizaciones se distribuyen a través de un canal seguro y estas se verifican antes de la instalación?	
Actualizaciones automáticas y/o informes de versión	
¿El software se mantiene actualizado a las últimas versiones estables disponibles?	
¿El proceso de actualización se realiza automáticamente o se generan alertas para que se instalen manualmente en el menor tiempo posible a partir que estas estén disponibles?	

3.3 Instrumento de evaluación para *Dew Layer*

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
Registro fuerte de eventos	
¿Se realiza un registro de los eventos realizados en el sistema, incluido los eventos de seguridad?	
Capacidades de identificación criptográfica	
¿Se gestiona el ciclo de vida criptográfico para verificar la confianza de los elementos del sistema?	
Cifrado de almacenamiento	
¿Se cifran los datos sensibles almacenados en los dispositivos como lectura del sensor, configuración, credenciales de autenticación o claves criptográficas?	

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
Fuerte autenticación local y soporte de autenticación de múltiples factores	
¿Se cambiaron las credenciales por defecto de servicios y componentes usados?	
¿Se usan contraseñas robustas en los servicios de autenticación?	
En la medida de lo posible, ¿se cuenta con autenticación de múltiples factores?	
¿Se cuentan con mecanismos que informen y registren intentos de autenticación fallidos y se proporciona un mecanismo de retardo o bloqueo antes cierto número de intentos fallidos?	
Capacidades M2M seguras	
¿Se hace uso de capacidades como la autorización, la verificación y la autenticación para proporcionar comunicaciones M2M basadas en la confianza?	
En la medida de lo posible, ¿las capacidades como autorización, la verificación y la autenticación se extienden su funcionalidad a un estado *offline*?	
¿La confianza en las comunicaciones M2M se gestiona con la confianza transitiva?	
Interfaz web seguras	
¿Se implementen contramedidas de seguridad contra vulnerabilidades comunes tales como omisión de autenticación, scripts entre sitios y falsificación de solicitudes entre sitios, entre otros?	
¿Se hace uso del protocolo TLS (HTTPS) y no se usan certificados autofirmados o no válidos?	
¿Se limita el acceso a la interfaz web para evitar el uso o abuso no autorizado?	
Utilice pilas y protocolos de red establecidos y probados	
¿Los protocolos y pilas de protocolos usando cuentan con amplio soporte de prueba y actualizaciones?	
¿Se desactivan los protocolos que no son usados?	
Utilice los últimos componentes de terceros actualizados	
¿Se están usando las versiones más recientes de los componentes, plataformas y servicios desarrollados por terceros?	
¿Se provee la capacidad de mantenerse informado sobre nuevas versiones de software y actualizaciones de seguridad de los componentes a medida que estén disponibles?	
¿Las actualizaciones se distribuyen a través de un canal seguro y estas se verifican antes de la instalación?	
Se recomienda usar un lenguaje de programación seguro o sujeto a escrutinio	
¿Los componentes están escritos en lenguajes de programación que posean contramedidas de seguridad y demuestren un historial de seguridad sólida?	
Actualizaciones automáticas y/o informes de versión	
¿El software se mantiene actualizado a las últimas versiones estables disponibles?	
¿El proceso de actualización se realiza automáticamente o se generan alertas para que se instalen manualmente en el menor tiempo posible a partir que estas estén disponibles?	
Verificación de actualización	
¿Se provee que las actualizaciones se entreguen a través de un canal seguro y se verifiquen después de la descarga para garantizar que las actualizaciones sean legítimas?	
Sin contraseñas predeterminadas	

Controles de ciberseguridad para *Dew Layer*	
Control	**Cumple**
¿Las credenciales por defecto se han cambiado en todos los componentes y servicios usados en la aplicación?	
¿Se ha establecido una política para el manejo de las credenciales como es el establecerlas y restablecerlas cada cierto tiempo según sea necesario y otras consideraciones?	
Las interfaces están deshabilitadas por defecto	
¿Se desactivan las interfaces que no estén en uso?	
Capacidades de control y gestión del dispositivo	
¿Se permite la gestión de la plataforma del dispositivo y sus capacidades para la detección de debilidades de seguridad o vulnerabilidades?	
Funciones de seguridad fuera de línea	
¿Las características de seguridad siguen funcionando en modo *offline* y son tan sólidas como las características *online*?	
Consideraciones de propiedad transitiva	
Ante el cambio de propietario, ¿se permitir que el dispositivo sea borrado, reiniciado o y que los datos sean destruidos para proteger la información del propietario anterior?	
Cifrado de comunicaciones	
Siempre que sea posible, ¿se provee comunicaciones cifradas de extremo a extremo para garantizar que estas no puedan ser interceptadas o redirigidas?	
Rastrea y contiene datos de fuentes potencialmente contaminadas (inseguras)	
¿Se permite de alguna forma de etiquetado de datos o sanitización para rastrear y contener datos de fuentes no confiables?	

Arquitectura genérica para aplicaciones IoT

Security Management in Internet of THings Model – SMITH Model

Versión 1.0, 2017

Arquitectura genérica para aplicaciones IoT

El objetivo de proponer una arquitectura genérica de IoT que contenga funcionalidades comunes de IoT, es ofrecer interoperabilidad del modelo propuesto en este capítulo con las diferentes arquitecturas de referencias y arquitecturas particulares para IoT.

Para esta arquitectura genérica se eligió un estilo arquitectural en capas. Esta arquitectura costa de tres capas: *Cloud Layer*, *Fog Layer* y *Dew Layer*; como se muestra en la Figura 1.

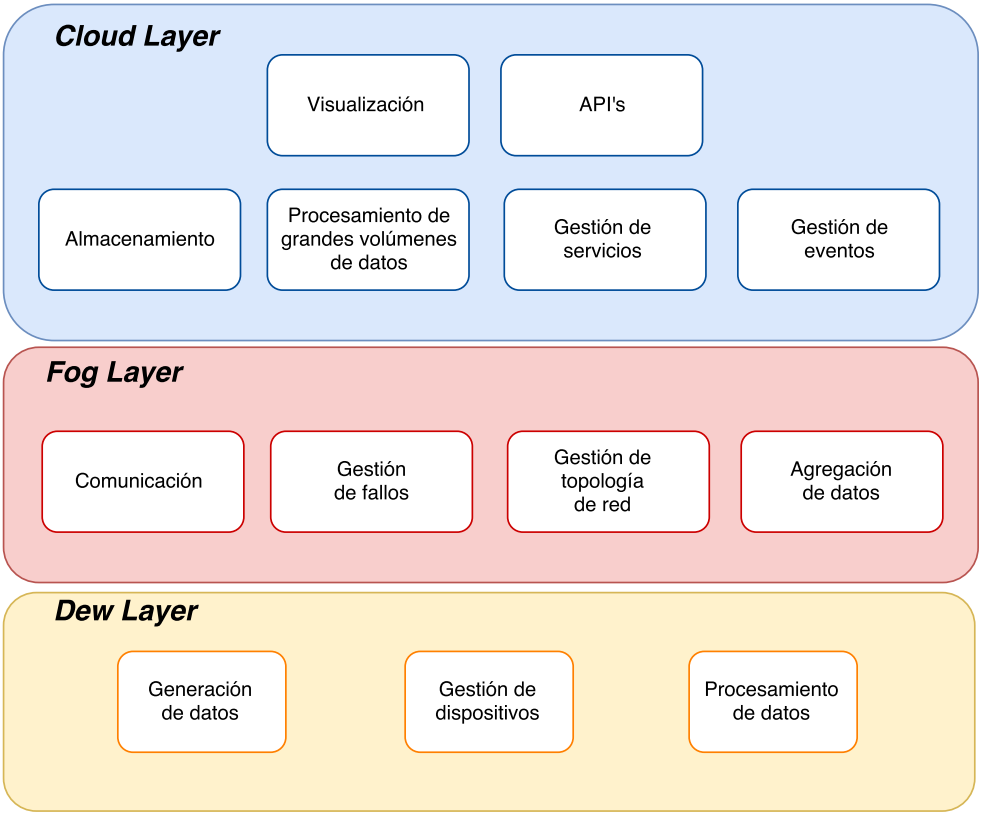


Figura 1. Arquitectura genérica para aplicaciones IoT

1 ***Cloud Layer***. En esta capa se concentra la mayor infraestructura de una aplicación IoT. En esta capa se tienen las siguientes funcionalidades y capacidades:

- **Visualización**. La aplicación IoT debe proveer varias formas de interacción del usuario con el sistema. Esta capacidad brinda al usuario la posibilidad de acceder a los recursos y servicios que ofrece la aplicación IoT, asimismo, la visualización de los datos generados por esta.
- **API's**. Esta capacidad permitirá a desarrolladores y otras aplicaciones acceder a recursos o servicios de la aplicación IoT y acceder a datos que

pueden alimentar otras aplicaciones IoT. Todo eso siempre y cuando estén autorizados por el propietario de dichos recursos.

- **Almacenamiento**. Esta capacidad permite a la aplicación o plataforma IoT almacenar los datos generados por los dispositivos y los nuevos datos generados a partir de ellos. Gracias a las características ofrecidas por el *Cloud Computing* provee a la aplicación IoT de escalabilidad de recursos de almacenamiento.
- **Procesamiento de grandes volúmenes de datos**. Una aplicación IoT puede generar un gran volumen de datos que no pueden ser analizados siguiendo técnicas tradicionales. El *big data* permite tratar y analizar esta cantidad de información, identificando patrones y generando más información a partir de la información generada generando valor para la organización que use la aplicación IoT.
- **Gestión de servicios**. Esta capacidad permite la búsqueda, el descubrimiento y la resolución de nombre de servicios; expone los recursos disponibles para hacerlos accesibles a otras partes del sistema IoT y a los usuarios.
- Gestión de eventos.

2 *Fog Layer*. En esta capa se encuentran las capacidades y funcionalidades relacionadas con la comunicación y transporte de datos. Entre ellas se tienen:

- **Comunicación**. Esta funcionalidad provee y gestiona la comunicación entre las capas *Dew* y *Cloud*, y se encarga de transportar información entre ellas. Ofrece funciones de control de conectividad de red y gestión de movilidad.
- **Gestión de fallos**. Su objetivo es identificar, aislar, corregir y registrar las fallas que ocurren en la aplicación IoT.
- **Gestión de topologías de red**. Esta capacidad provee la forma en que se transporta la información, ya sea una comunicación en red, *hop-to-hop* o *end-to-end*.
- Agregación de datos.

3 *Dew Layer*. En esta capa se encuentran las capacidades y funcionalidades relacionadas con los dispositivos. Entre ellas se tienen:

- **Gestión de dispositivos**. Esta capacidad permite el control de dispositivo, entre las cuales se sienten funcionalidades como: La activación y desactivación de dispositivos de forma remota, la gestión del estado de trabajo del dispositivo, su diagnóstico, la actualización de

firmware y/o software. Soportar las diferentes interfaces de los dispositivos conectados mediante diferentes tecnologías.

- **Gestión de los datos**. Esta funcionalidad contiene funcionalidades para la recolección de datos por parte de los nodos sensores y otros dispositivos.
- **Procesamiento de datos**. Aprovechando las capacidades de hardware de algunos dispositivos IoT, en esta capa se puede procesar información para la toma de decisiones que afecten a los nodos sensores y actuadores sin necesidad de enviar todos los datos a la capa superior, procesarlos, generar la toma de decisiones y enviar la orden de actuación a la capa inferior.

Requisitos de seguridad para aplicaciones IoT

Security Management in Internet of THings Model – SMITH Model

Versión 1.0, 2017

Requisitos de seguridad para aplicaciones IoT

Los requisitos de seguridad presentados en este documento se basaron en los documentos de las arquitecturas de referencia de la ITU-T, la IoT-A, SmartSantander y la WOS2. Basado en estas fuentes se extrajeron los requisitos de calidad relacionados con la seguridad.

A continuación, se enumeran los requisitos que debe considerarse en el diseño y construcción de una aplicación IoT para reducir el riesgo que los datos sean comprometidos. Estos requisitos se clasifican en cinco grupos: el grupo de requisitos para la confidencialidad de la información – GRC, el grupo de requisitos para la integridad de la información – GRI, el grupo de requisitos para la disponibilidad de la información – GRD y el grupo de requisitos para el no repudio – GRNP.

1. Grupo de requisitos para la confidencialidad de la información (GRC). Este grupo de requisitos buscan garantizar la confidencialidad de la información, propiedad que impide que el acceso o divulgación. Este grupo de requisitos se clasifican en tres subgrupos: Requisitos de seguridad, requisitos de privacidad y requisitos de autorización y autenticación.

1.1 Requisitos de seguridad. Los requisitos de seguridad (RS) se muestran a continuación:

- RS1: Un sistema IoT debe proporcionar a los usuarios el acceso seguro a los recursos.
- RS2: Un sistema IoT debe proporcionar comunicación segura y de confianza entre los dispositivos, los recursos y servicios, para evitar que terceros puedan espiarlas.
- RS3: Un sistema IoT debe proporcionar una medida de seguridad y privacidad del dispositivo.
- RS4: Un sistema IoT debe evitar que un dispositivo se active sin el consentimiento del propietario.

1.2 Requisitos de privacidad. Los requisitos de privacidad (RP) se muestran a continuación:

- RP1: Un sistema IoT debe proporcionar una alternativa para que los usuarios usen sus servicios de forma anónima.
- RP2: Un sistema IoT debe soportar la comunicación anónima entre

dispositivos y proporcionar la confidencialidad de comunicación.

- RP3: Un sistema IoT debe permitir que los usuarios tengan el control de cómo sus datos están expuestos a otros usuarios.
- RP4: Un sistema IoT debe proporcionar la privacidad de la ubicación.
- RP5: Un sistema IoT debe proteger la privacidad de los usuarios que acceden a información sobre entidades o servicios físicos.
- RP6: Un sistema IoT debe evitar el seguimiento del identificador del dispositivo por entidades no autorizadas.

1.3 Requisitos de autenticación y autorización. Los requisitos de autenticación y autorización (RAA) se muestran a continuación:

- RAA1: Un sistema IoT debe proporcionar diferentes permisos de acceso a la información.
- RAA2: Un sistema IoT debe admitir la limitación de acceso a la red a dispositivos específicos hasta que no presente credenciales apropiadas para unirse a la red.
- RAA3: Un sistema IoT debe apoyarse en mecanismos de control de acceso.
- RAA4: Un sistema IoT debe admitir la autenticación mutua de sujeto.

2. Grupo de requisitos para la integridad de la información (GRI). Los requisitos de integridad (RI) se muestran a continuación:

- GRI1: Un sistema IoT debe proveer la integridad de la comunicación.
- GRI2: Un sistema IoT debe validar la integridad de entidades virtuales, dispositivos, recursos y servicios.

3. Grupo de requisitos para la disponibilidad de la información (GRD). Los requisitos de disponibilidad (RD) se muestran a continuación:

- GRD1: Un sistema IoT debe garantizar la disponibilidad de infraestructura.
- GRD2: Un sistema IoT debe garantizar la disponibilidad de sus servicios.
- GRD3: Un sistema IoT debe garantizar la disponibilidad de la red.
- GRD4: Un sistema IoT debe garantizar la frescura de los datos.
- GRD5: Los servicios IoT debe estar siempre accesible para los usuarios.

4. Grupo de requisitos para el no repudio (GRNP). Los requisitos de no repudio (RND) se muestran a continuación:

- RNP1: Un sistema IoT debe ofrecer una identificación única de los usuarios que solicitan datos a través de los servicios de descubrimiento/búsqueda.
- RNP2: Un sistema IoT debe apoyar la identificación de la ubicación del dispositivo.
- RNP3: Un sistema IoT debe garantizar el no repudio a nivel de recursos de red.