IoTAP: Prácticas de diseño de la arquitectura y sus requisitos de calidad en el contexto de desarrollo de software basado en IoT

Universidad del Cauca

Monografía de Trabajo de Grado

Oscar Santiago López Erazo

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Maestría en Computación

Grupo de investigación IDIS

Áreas de Investigación: Arquitectura de software y Calidad de software

Popayán, junio de 2023

IoTAP: Prácticas de diseño de la arquitectura y sus requisitos de calidad en el contexto de desarrollo de software basado en IoT

Universidad del Cauca

Monografía de Trabajo de Grado

Oscar Santiago López Erazo

Director: PhD. Julio Ariel Hurtado Alegría

Co-Director: PhD. Francisco José Pino Correa

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Maestría en Computación

Grupo de investigación IDIS

Áreas de Investigación: Arquitectura de software y Calidad de software

Popayán, junio de 2023

TABLA DE CONTENIDO

LISTA DE TABLAS Y FIGURAS

AGRADECIMIENTOS

En primer lugar, agradezco a Dios por ayudarme a culminar esta etapa en la cual he podido aprender, reflexionar y mejorar en muchos aspectos de mi vida como el espiritual, académico y personal. Le dedico esta tesis a las personas que han influido en ese mejoramiento, a mi padre Oscar Enrique por estar siempre presente para mí, ayudándome a salir adelante con sus consejos y experiencias de vida, a mi madre María Amparo que siempre estuvo a mi lado brindándome su amor incondicional y su compañía, a mi hermana Juliana María quien siempre está para mi escuchando mis ideas y pensamientos, acompañándome y brindándome su comprensión, a mi perrito Puppy Andrés por ser mi compañero fiel, mi confidente y mi fuerza para seguir adelante, a mi amigo, director y profesor Julio Ariel quien me ayudo a encontrarme a mismo en los momentos que más dudaba sobre mí, apoyándome desde el principio cuando inicio mi carrera como Ingeniero, siendo mi mentor y mi guía formándome en cada aspecto como profesor e investigador al igual que como persona ayudándome a mejorar para aportar mis capacidades a la sociedad, a mi profesor Cesar Collazos por ayudarme y brindarme sus consejos para poder encontrar mi camino como investigador y al profesor Freddy Muñoz por ayudarme en la continua mejora de mi trabajo de investigación. A mis tutores Andrés Felipe Hurtado y Víctor Peñeñory quienes me brindaron la oportunidad de investigar en el Grupo LIDIS Laboratorio de Investigación para el Desarrollo de la Ingeniería de Software, adscrito al Departamento de Sistemas de la Universidad de San Buenaventura Cali, ayudándome a crecer como investigador y persona.

1. CAPITULO 1

Introducción

En este capítulo se presenta el problema planteado y abordado en el desarrollo del presente trabajo. Además, se evidencian los aportes, objetivos y la metodología del trabajo, descrita mediante un conjunto de fases, las cuales se desarrollan de forma clara y estructurada. Por último, se describe la estructura general del documento.

1.1. Planteamiento del problema

Con la rápida penetración de las tecnologías de la información y la comunicación (TIC) en las economías y en la vida cotidiana, y la necesidad de un mundo más interconectado, surgen paradigmas como el Internet de las cosas (IoT), una red de objetos físicos y digitales que están conectados entre sí a través de protocolos estandarizados [1]. La IoT está compuesta por una infraestructura general conformada por (i) hardware: elementos físicos y tangibles como las placas, los sensores y actuadores, (ii) conectividad de red: conexión que se realiza a través de internet, (iii) arquitectura: es la estructuración del producto software IoT y (iv) software: código, algoritmos, programas y rutinas que permiten realizar una tarea específica [2].

La IoT brinda nuevas oportunidades en muchos campos de aplicación, particularmente para la industria 4.0 e investigación, en tecnologías y su automatización. además de un impacto significativo en la vida cotidiana [2]. Su uso permite cerrar la brecha entre los objetos físicos y digitales encapsulados en un producto software. Para el 2020, se habían estimado 50 mil millones de dispositivos conectados al IoT [3], los cuales perciben, producen y comparten un gran volumen de datos, evidenciando la gran importancia de la calidad de estos productos IoT [4]. Por otra parte, los estudios generados por la International Data Corporation (IDC) informan que el 31,4 % de las organizaciones encuestadas habían lanzado soluciones IoT. Sin embargo, DZone, una empresa que se dedica a ofrecer información sobre las tendencias y preocupaciones de IoT, informó mediante una encuesta que el 87 % de los desarrolladores no le prestan la suficiente atención a este nuevo tipo de paradigmas [5].

El desarrollo de productos software en general cuenta con procesos y prácticas reconocidas por la industria de software, teniendo como fases críticas el análisis de los requerimientos, donde se determinan las funcionalidades, las propiedades y los atributos de calidad que debe tener un producto software, los cuales dan lugar a las primeras aproximaciones al diseño de la solución que se plantean en forma temprana en la arquitectura del software. Sin embargo, en el contexto del desarrollo productos software que utilizan el paradigma de la IoT [6], estos procesos y prácticas son aún muy incipientes. Este paradigma, hace posible que los niveles de automatización y autonomía de los productos software y servicios convencionales se incrementen de una forma rápida. Dentro de los productos software basados en IoT, existen grandes cantidades de productos que son tan nuevos y desconocidos para los desarrolladores, que requiere de mucha capacitación y el ensamble de estos productos se realiza de manera Ad hoc afectando la calidad de su construcción [5]. Esto se debe, a que la IoT, es considerada como parte de una arquitectura de información global emergente basada en internet, que presenta una gran variedad de interrogantes tales como privacidad, protección de datos, seguridad, ética,

interoperabilidad, que son notablemente diferentes en la internet convencional como consecuencia de su gran tamaño, volumen de datos generado y heterogeneidad [4].

La calidad de un producto software se define como el grado en el cual satisface los requisitos de los interesados aportando de esta manera valor al negocio [7]. Particularmente en la gestión de redes de dispositivos IoT, el valor al negocio puede llegar a comprometerse puesto que ésta tecnología tiene un gran impacto en términos de satisfacción del cliente, por su influencia en las relaciones de las organizaciones, clientes y proveedores. En el aspecto de organización, existe una alta complejidad en agregar sensores y datos sus datos a la variedad de los productos y procesos operativos de una empresa. Además, otro aspecto importante que se encuentra, es el de costos variables ya que a diferencia de los productos software convencionales, en los que los costos son relativamente bajos, cada dispositivo IoT adicional puede generar considerables costos continuos de mantenimiento, teniendo en cuenta que la IoT es una red real de cosas físicas que pueden crecer súbitamente. Por último, se encuentran las implicaciones sociales, ya que no se sabe con exactitud como las personas reaccionarán con la incursión de estos dispositivos en su vida cotidiana [5].

De acuerdo a lo anterior, los productos software basados en IoT cuentan con un conjunto de características y sub-características de la calidad del software, tales como funcionalidad, rendimiento, seguridad, mantenibilidad, entre otras, pero con unas particularidades propias del dominio de los sistemas ciberfísicos. La norma ISO 25010 plantea un modelo con el objetivo principal de definir la características y sub-características para un producto software en general [7]. Sin embargo, la mayoría de propuestas que abordan la calidad en dispositivos IoT, presentan dificultades debido a que : (i) la IoT es una fusión compleja de tecnologías con software estrechamente relacionado con el hardware [4], [8], [9], (ii) se requiere de metodologías específicas para el desarrollo y aseguramiento de la calidad de los productos software basados en la IoT [10]–[13], (iii) hay una mayor dependencia de la calidad de los productos de software IoT respecto al campo, contexto, situación, ámbito y dominio [6], [14], [15], (iv) existen deficiencias y retos en temas como la calidad del servicio, la heterogeneidad, los recursos limitados, la seguridad y privacidad del usuario, la confiabilidad del servicio, la interoperabilidad y la integración [2]–[5], [10], [16] y (v) los patrones de arquitectura de IoT son muy específicos del dominio de aplicación [17]. Por tanto, es importante tomar las acciones necesarias con el fin de lograr un grado de confianza de un producto software basado en IoT, tomando en cuenta buenas prácticas para la especificación de una arquitectura de software que responda a los requisitos de calidad de un producto software basado en IoT [14].

Para los productos software basados en IoT, la arquitectura es un aspecto relevante ya que ayuda en la abstracción de los componentes de hardware heterogéneos, garantiza operaciones fluidas mediante protocolos de red y aporta en la calidad de temas computacionales relacionados con la IoT [18]. Un ejemplo de esto son los componentes arquitectónicos como los módulos de código ejecutable, que tienen el objetivo de facilitar interfaces programables y estandarizadas. Alcanzar algunos atributos de calidad requiere de una adecuada implementación y compresión en la arquitectura con el fin de considerar los efectos que estos puedan ocasionar sobre las soluciones y las organizaciones. Como la arquitectura de software se tiene en cuenta en las etapas tempranas del ciclo de desarrollo software, es importante determinar la arquitectura correcta, cumpliendo los requisitos de calidad planteados y representándolos como escenarios concretos de atributos de calidad con el objetivo de contribuir en el éxito de cualquier proyecto [19].

En resumen, el principio de la arquitectura de software junto con buenas prácticas puede aportar en la evolución, modelado, desarrollo y ejecución de productos software complejos basados en IoT, fortaleciendo la calidad y funcionalidad requeridas, así como en el desarrollo de soluciones arquitectónicas de nueva generación para paradigmas como IoT [18].

La selección y el cumplimiento de los atributos de calidad durante el desarrollo de software basado en IoT es una tarea compleja [20], debido a que este tipo de sistemas vienen siendo aplicados en diferentes dominios en el contexto de una gran variedad procesos de negocio y tecnologías. Por lo anterior, se requiere tomar en cuenta los atributos de calidad de la arquitectura con el fin de lograr una adecuada funcionalidad de los sistemas (además del software), minimizar los riesgos y alcanzar una adecuada relación costo-beneficio durante su desarrollo [19].

Teniendo en cuenta lo anterior, es evidente que hay una gran necesidad por cubrir desde la ingeniería de software [5] en lo que respecta a la construcción de soluciones software basadas en la IoT y mucha relevancia en el diseño de soluciones que satisfagan los aspectos de calidad, por lo cual el presente proyecto plantea la siguiente pregunta de investigación alrededor de la obtención de una arquitectura en este contexto:

¿Cómo fortalecer el diseño de la arquitectura y sus requisitos de calidad en el contexto del desarrollo de software basado en IoT?

Por ello**,** se hace necesario aportar en la calidad de productos software basados en la IoT desde las etapas tempranas del desarrollo como lo es la arquitectura y sus requisitos de calidad. Es importante resaltar, como lo plantean Larrucea et al. [5], que se hace necesario investigar sobre las buenas prácticas de ingeniería de software para IoT, por lo cual esta tesis de maestría se enfocará en plantear y evaluar un conjunto de prácticas de arquitectura y sus requisitos relevantes que favorezcan la calidad de los productos software basados en IoT.

1.2. Objetivos

1.2.1. Objetivo general

Establecer un conjunto de prácticas[1] de arquitectura y sus requisitos de calidad para el desarrollo productos software[2] basados en IoT, considerando las características especiales de este paradigma, la norma ISO/IEC 25010 y los participantes directos e indirectos, con el fin favorecer la calidad de este tipo de productos.

1.2.2. Objetivos específicos

- Caracterizar, a partir de literatura, los atributos de calidad especiales del producto software basado en IoT, así como las prácticas de arquitectura de

[1] Práctica: se refiere a una directriz que establece como lograr determinadas características y sub-características de calidad dentro de un producto software basado en la IoT.

[2] Producto software: es una unidad lógica de compartición y empaquetado de software que tiene un desarrollo gestionado, un ciclo de vida de mantenimiento y atributos visibles para el cliente. Tomado de https://www.ibm.com/docs/es/cdfsp/7.6.1.2?topic=application-software-hierarchy

software existentes para alcanzarlos y los participantes directos e indirectos involucrados.

- Organizar a nivel conceptual y metodológico un conjunto de prácticas de arquitectura y sus requisitos de calidad que contribuyan al desarrollo del producto software basado en IoT a partir de la caracterización realizada anteriormente.
- Evaluar la comprensibilidad y utilidad percibida por desarrolladores de software de la región al utilizar las prácticas IoTAP en el desarrollo de un producto software basado en IoT mediante la aplicación de un estudio de caso en el dominio de la agricultura inteligente.

1.3. Metodología

Para el desarrollo del proyecto propuesto se adapta y sigue la metodología de investigación de la ingeniería de software por Finkelstein [21]. Esta metodología contiene tres fases: (i) Fase Proposicional y Analítica donde se complementa con los enfoques propuestos por Kitchenham [22] y Petersen et al. [23], (ii) de Valoración, donde se utilizará la metodología de métodos mixtos de investigación [24] bajo la perspectiva de investigación positivista mencionada en [25], que busca que los estudios de caso cuenten con evidencia para establecer proposiciones formales, midiendo variables, probando hipótesis y extrayendo inferencias de una muestra a una población establecida, y (iii) de Documentación y Divulgación. A continuación, se describen las fases de la Metodología de Ingeniería de Software.

1.4. Fase Proposicional y Analítica

Durante esta fase se realizará un proceso de investigación cualitativa el cual incluirá: (i) fase preparatoria; (ii) trabajo de campo; (iii) fase analítica y (iv) fase informativa. Para el desarrollo del estado del arte se realizó un mapeo sistemático siguiendo los enfoques planteados por Kitchenham [22], que propone una guía para revisiones sistemáticas apropiadas para investigadores de ingeniería del software y Petersen et al. [23], que propone una guía para realizar estudios de mapeo sistemático en ingeniería de software.

Para la caracterización de los atributos de calidad especiales y de las prácticas de ingeniería de software existentes, se llevará a cabo el siguiente proceso, en el cual se realizarán una serie de actividades con el fin de establecer el rumbo de la presente investigación. Para el primer objetivo, se ejecutarán tareas como la preparación del protocolo, con el objetivo de identificar y seleccionar: atributos especiales de calidad y prácticas de ingeniería de software existentes. Para el segundo objetivo, se tendrán en cuenta actividades como comprensión y análisis de la caracterización mencionada anteriormente, con el fin de clasificar y seleccionar el conjunto de prácticas.

1.5. Fase de Valoración

Para el tercer objetivo, se incluirá la metodología de métodos mixtos de investigación con el objetivo de complementar la etapa de evaluación y analizar con más detalle sus resultados. La investigación de métodos mixtos es el tipo de investigación en la cual se busca combinar técnicas, métodos, enfoques, conceptos

o lenguajes de investigación cualitativa y cuantitativa, con el objetivo de aprovechar las fortalezas de ambos tipos de indagación y minimizar sus debilidades [26][27]. Éstos pueden ser conjuntados de tal manera que las aproximaciones cuantitativa y cualitativa conserven sus estructuras y procedimientos originales ("forma pura de los métodos mixtos"). Alternativamente, estos métodos pueden ser adaptados, alterados o sintetizados para efectuar la investigación y lidiar con los costos del estudio ("forma modificada de los métodos mixtos") [24]. Para desarrollar esta fase se planeará, diseñará, ejecutará y documentará un estudio de caso en el que se desarrolle una solución IoT haciendo uso de las prácticas de arquitectura, apoyado por la metodología de estudio de casos realizada por Runeson y Host [25], que tiene como fin de guiar a los investigadores con prácticas recomendadas para estudios de casos de ingeniería de software así como listas de verificación derivadas y evaluadas empíricamente.

1.6. Fase de Documentación y Divulgación

Esta fase se abordará mediante de la generación de los productos derivados del proceso investigativo y la presentación de resultados por medio de informes de proyecto y generación de productos científicos como artículos de investigación.

1.7. Estructura del documento

A continuación, se describe la forma en la que se encuentra el trabajo desarrollado en el presente documento:

Capítulo 2- Marco Teórico: Se presenta el Marco conceptual para que el lector tengo una base comprender el trabajo. Además, se incluye el estado del arte, donde se muestran los trabajos más relevantes para el desarrollo del presente trabajo.

Capítulo 3 - Caracterización de los atributos de calidad especiales del producto software basado en IoT y de las prácticas de arquitectura de software

Capítulo 4 – Organización a nivel conceptual y metodológica del conjunto de prácticas con sus requisitos de calidad asociados

Capítulo 5 - Estudio de caso: Evaluar la comprensibilidad y utilidad percibida por desarrolladores de software de la región al utilizar las prácticas IoTAP en el desarrollo de un producto software basado en IoT

Capítulo 6- Conclusiones y trabajo futuro: Finalmente se presentan las conclusiones que surgieron como resultado de la investigación y se expone el trabajo que hace falta por desarrollar.

2. CAPITULO 2

En la actual época, la tecnología ha aportado de gran manera en la solución de distintos problemas o necesidades aportando calidad de vida para la humanidad. Dentro de la tecnología, se encuentran los productos que compuestos por software o más conocidos como **productos software** que se definen como una unidad lógica de compartición y empaquetado de software que tiene un desarrollo gestionado, un ciclo de vida de

mantenimiento y atributos visibles para el cliente. Particularmente en este marco teórico se establecen los aspectos relacionados a soportar la idea de trabajar sobre la arquitectura y los requisitos de calidad en el contexto de soluciones basadas en IoT (La referencia de esta definición se encuentra en la página 15-Pie de página).

2.1. Conceptos Fundamentales

2.2. Internet de las cosas

Dentro de los avances que se han tenido en las tecnologías de la información y las comunicaciones, uno de ellos es **Internet of Things (IoT)** que se define como una red de objetos físicos y digitales que están conectados entre sí a través de protocolos estandarizados [1]. La IoT está compuesta por una infraestructura general conformada por (i) hardware, (ii) conectividad de red, (iii) arquitectura y (iv) software [2]. Esta red también permite interconexiones entre humanos, servicios, sistemas, dispositivos y cosas con el objetivo de generar sociedades digitalizas y sistemas autónomos. Además, estos sistemas ayudan a recopilar, procesar e intercambiar datos útiles [18].

IoT también consiste en una red de dispositivos, actuadores y sensores que tienen como propósito recolectar datos con el fin de reaccionar a estos tomando decisiones, para posteriormente ejecutar acciones que mejoren el entorno en cuestión. Entre sus principales características, se encuentra la interacción de los dispositivos conectados a la red internet sin la necesidad de la intervención humana, siendo una de estas las razones más relevantes para adecuar el enfoque de desarrollo tradicional tomando en cuenta requerimientos características y específicos de estos productos software [6].

2.3. Calidad de Software en IoT

Existen dificultades entre las interconexiones mencionadas anteriormente que dificultan la compresión del producto software y puede llegar a comprometer la calidad de este mismo, por ello es necesario hacer hincapié en la **calidad de software**, que se define como la evaluación de un conjunto de variables y atributos, que se clasifican según los criterios de calidad internos y externos. Los criterios externos hacen referencia a la experiencia del usuario cuando este ejecuta el software en la fase de implementación y los criterios internos, hacen referencia a los aspectos internos del software, es decir, solo lo que pueden observar los desarrolladores [3]. La calidad se apoya sobre modelos que permiten realizar evaluaciones a dichos productos, uno de estos modelos es **ISO/IEC 25010**, el cual representa un modelo jerárquico que posee características de software, que cuenta con una clasificación especifica donde en la parte superior se encuentra de forma general, la calidad producto software. En el segundo nivel, se encuentra un grupo de ocho características o atributos de calidad que a su vez tienen asociadas un grupo de sub-características. Además, esta clasificación incluye una descripción de las características, la guía de aplicabilidad y sus respectivas métricas [14].

2.4. Arquitectura de Software

Para poder alcanzar la calidad en los productos software, se deben considerar los **atributos de calidad** relevantes, los cuales son definidos como cualidades medibles que permiten establecer en qué grado se cumplen los requerimientos de calidad [14]. Los sistemas, en general, poseen atributos de calidad como usabilidad, mantenibilidad, rendimiento y confiablidad, entre otros. En cualquiera de estos sistemas se pueden encontrar múltiples necesidades por cubrir respecto a estos atributos, siendo todos importantes en medidas diferentes que dependen del contexto, sin embargo, satisfacer un atributo de calidad puede afectar a otro atributo de calidad según las decisiones que se tomen durante el ciclo de vida de un producto software, debido a que hay relaciones de compromiso [28]. Dichos atributos impactan de gran forma en el producto software basado en IoT, donde uno de los activos de software más importante es la **arquitectura de software** que se define como una estructura que se encuentra organizada en componentes y sistemas que interactúan entre sí, permitiendo conformar sistemas complejos con una esperanza de vida mucho más larga. Las arquitecturas de software están determinadas por tres elementos relevantes: (i) piezas, (ii) las propiedades externamente visibles (ii) las relaciones y (iii) restricciones que se pueden representar mediante diferentes perspectivas llamadas vistas [29][30]. Las vistas aportan una descripción complementaria y consistente de la arquitectura desde un punto de vista específico, por ejemplo, la vista de componentes y conectores presenta las componentes ejecutables del sistema, los conectores que establecen los protocolos de interacción y las restricciones de tiempo de ejecución sobre los componentes y los conectores. Normalmente, estas vistas reflejan la forma en que se resuelven las inquietudes de los diferentes interesados en el software y sirven como un medio de comunicación de la arquitectura y su razón de ser a los diferentes usuarios del documento de arquitectura [31].

Además, existen decisiones de diseño que se toman para mejorar los atributos de calidad, denominadas **tácticas arquitecturales**. Una táctica mejora los problemas enfocándose a un atributo de calidad en específico. Estas mismas en ocasiones pueden ser aplicadas fácilmente utilizando las mismas estructuras y comportamiento compatible, como un patrón de arquitectura particular. Por otra parte, una táctica puede requerir cambios relevantes en el comportamiento y estructura del patrón o en algunos casos, estructuras y comportamientos totalmente nuevos [28].

Una **decisión de diseño arquitectónico**, se define como la descripción de modificaciones, adiciones, y sustracciones a la arquitectura de software, en sus reglas y restricciones de diseño, lógica y requisitos nuevos que acarrean nuevas decisiones consigo [32]. La arquitectura del sistema que se diseñara puede ser impactada debido a las tácticas que se seleccionen durante el diseño inicial de la arquitectura.

Dentro de la arquitectura existen métodos que han sido planteados por el Software Engineering Institute (SEI) perteneciente a la Carnegie Mellon University, entre ellos se encuentran (i) QAW, (ii) ADD, (iii) VaB y (iv) ATAM.

- El método Quality Attribute Workshop (QAW) planteado por Barbacci et al. [33], se enfoca en la captura, documentación y el análisis de los requerimientos que determinan la arquitectura de software, dichos requerimientos involucran atributos de calidad que son descritos a través de

escenarios que proporcionan información medible con el fin de analizar el comportamiento del artefacto o sistema esperado frente a un estímulo importante para el atributo de calidad analizado. Este taller ayuda a que los requisitos se recopilen de manera efectiva en grupo, sin embargo, puede ser engorroso y costoso de organizar. Por ello, se plantea el miniQAW, un taller más corto diseñado para facilitadores sin experiencia y fácil de adoptar [34].

- El método Attribute Driven Design (ADD) realizado por Wojcik et al. [35], plantea que una vez se han establecido y priorizado los escenarios, facilita aún más el diseño de la arquitectura mediante un enfoque recursivo de descomposición del sistema en componentes más pequeños. Durante el proceso se van tomando decisiones a nivel de diseño y escenarios, a través de la identificación y aplicación de tácticas y patrones orientados a satisfacer los requerimientos descritos por los escenarios. A todos estos elementos de diseño agrupados se le conoce como la estrategia arquitectónica.

- En Views and Beyond (VaB) planteada por Clements y Bass [36], se brindan estrategias documentales y visuales clave para comunicar la arquitectura de software resultado de la aplicación de ADD Y QAW. Por lo tanto, se obtienen distintas estructuras del sistema que se encuentran conformadas por las piezas y sus relaciones, desde diferentes puntos de vista. Estas estructuras de documentan a través de vistas que muestran una perspectiva particular del sistema.

- En Architecture Tradeoff Analysis Method (ATAM), planteado por Kazman, Klein y Clements [37], se encuentra un método que permite descubrir que tan bien una solución arquitectónica satisface los atributos de calidad establecidos y además que riesgos, puntos sensibles y compromisos están involucrados en la arquitectura propuesta respecto de las expectativas de calidad bajo evaluación.

Un aspecto clave en dentro de la arquitectura de software, es sobre qué debe incluir la documentación. según el SEI esta tarea es muy importante y por ello propone un marco para documentar y comunicar la arquitectura con base en vistas, puntos de vista y tipos de vista basado en el estándar ANSI/IEEE 1471-2000 [38], el cual contienen prácticas recomendadas para descripción de la arquitectura de los sistemas software. Los esquemas que se consideran con mayor frecuencia para representar la arquitectura son:

- En 4 + 1 Views [39], se describe la arquitectura dl software usando cinco vistas concurrentes: (i) lógica, (ii) procesos, (iii) física, (iv) desarrollo y (v) escenarios.

- Zachman [40] propone un marco que permite definir la arquitectura de sistemas de información mediante la creación de un marco descriptivo de disciplinas bastante independientes de los sistemas de la información especificando la arquitectura de estos sistemas basados en el marco neutral y objetivo.

- Analytic Principles and Tools for the Improvement of Architectures (APTIA) [41], se propone reutilizar casi por completo las técnicas de mejora y análisis existentes, con el fin de mejorar la arquitectura de un sistema existente.

2.5. Estado del Arte

2.6. Estudios sobre calidad de soluciones de software basados en IoT

Requerimientos no funcionales IoT

El paradigma IoT aumenta los niveles de autonomía y automatización de los sistemas y servicios tradicionales, lo que ha llevado su aplicación a diferentes ámbitos y dominios de aplicación. En el estudio de Ojo-González y Bonilla-Morales [6], realizan una revisión literaria apoyada con la norma ISO/IEC 25010, identificando los requerimientos no funcionales comunes para sistemas IoT como lo son la interoperabilidad, escalabilidad/flexibilidad, seguridad y sensibilidad al entorno, cabe resaltar que estos requerimientos no son los únicos que deben ser abordados, por ello se identifican aquellos requerimientos adicionales que reciben mayor atención si se tiene en cuenta los cuatro dominios de aplicación tales como: ciudades, hogares, agricultura y fábricas inteligentes. Sin embargo, los sistemas IoT que presentan sensibilidad al entorno o contexto son independientes del dominio, esta capacidad permite recolectar datos de múltiples sensores y actuar con base en dicha información. El desafío propuesto es independizar el diseño del dominio quien define qué aspectos del sistema se contemplan para cada una de las características.

IoT testing

Bures et al [10] presentan un estado del arte que desde las perspectiva de IoT, se enfoca en temas de seguridad donde se menciona que aún persisten dificultades denominadas brechas de seguridad que atentan contra el usuario. Respecto a privacidad y confianza del usuario, se encuentra que se puede dar un uso indebido a los datos personales recopilados, además se menciona que se debe tener en cuenta la reconstrucción del retrato digital del usuario a partir de varios flujos de datos. Con respecto a los bancos de pruebas de IoT, se habla sobre configuraciones independientes, arquitecturas distribuidas y simulación de dispositivos físicos debido al costo del entorno de prueba físico. En cuanto al aseguramiento de la calidad y técnicas de prueba, cubre el tema de las pruebas funcionales para IoT específicamente en pruebas de integración, basada en modelos y técnicas relacionadas. Dentro de los desafíos hallados se encuentra que el nivel de estandarización es insuficiente en temas como la legislación y técnicas de aseguramiento en temas relacionados a la seguridad, la confiabilidad, la interoperabilidad y la integración que se encuentran en este tipo de aplicaciones. También se menciona que todos los desafíos conllevan a una demanda para crear una metodología específica para el aseguramiento de calidad de aplicaciones IoT, además en el área de las metodologías de prueba y aseguramiento de calidad, existe poco trabajo realizado.

Aseguramiento de calidad

En la época actual ya no es aceptable lanzar productos IoT al mercado sin la adecuada calidad, por lo que, Ahmed et al. [12], realizan un mapeo sistemático para clasificar la evidencia hallada en la literatura a partir de 2009, cuando se realizó la primera publicación de estudios aseguramiento de la calidad de IoT. La búsqueda de trabajos en el presente estudio se hace en términos de cuál es el número de estudios publicados en la última década acerca del aseguramiento de calidad, países activos en investigación, que aspectos de calidad se han tratado en investigaciones anteriores, cuáles son las principales técnicas o conceptos de prueba que se han investigado previamente, dominios de aplicación específicos en el contexto de IoT desde la calidad y por último limitaciones/desafíos actuales en el aseguramiento de calidad. Los resultados de esta investigación evidencian la necesidad de realizar más trabajo en el contexto del aseguramiento de la calidad de IoT. Existen dos desafíos que aún deben ser solucionados, el primero la heterogeneidad en general ya que se encuentran varios tipos de soluciones que requieren métodos de prueba específicos y, en segundo lugar, centrarse en los niveles del sistema, incluida la capa física, los protocolos, el firmware y el software de un dispositivo en cuestión.

Es necesario desarrollar técnicas adecuadas para el aseguramiento de calidad en aplicaciones IoT, teniendo en cuenta el aseguramiento de calidad, definido como un patrón planificado y sistemático en el que se toman acciones necesarias con el fin de lograr un grado de confianza conveniente sobre un producto satisfaciendo los requisitos técnicos de calidad dispuestos mediante pruebas, para este tipo de softwares estas son muy importantes ya que tienen como objetivo encontrar y corregir errores. Por ello Foidl y Felderer [15], describen los requisitos de calidad que se encuentran cambiando en el ámbito de la IoT, agrupándolos en seis categorías: Entorno, Usuario, Cumplimiento/Acuerdo de nivel de servicio, Organización, Seguridad y Gestión de datos, según el autor debido a que el IoT aún está en sus inicios, existe una limitada información sobre artículos académicos sobre control de calidad y pruebas. Debido al gran volumen de datos generados respecto a los procesos de las aplicaciones IoT, la computación actual juega un papel importante en el aseguramiento de calidad. Manejar el gran volumen de datos generados por este tipo de aplicaciones se convierte en un gran desafío, además aplicar técnicas, algoritmos y métodos para extraer conocimientos valiosos en dicho ámbito, además de las arquitecturas disponibles para que satisfacer dichas cualidades.

Técnicas y metodologías IoT

El rápido crecimiento y adopción de la IoT ha hecho que exista un aumento en el uso de la computación ubicua, conectividad, máquinas y personas que usan esta tecnología. Según Marwah y Sirshar [13], existen muy pocos parámetros para verificar y garantizar la calidad de aplicaciones, dispositivos y maquinas que usen la IoT. También se presenta un estudio exhaustivo de las técnicas y metodologías utilizadas para la implementación de IoT proporcionado un amplio conocimiento en temas como estándares de aseguramiento de calidad y su respectiva comparación. Se evalúan técnicas de implementación contra parámetros de aseguramiento de calidad, con el fin de aumentar la confianza y satisfacción del cliente, a su vez la credibilidad permitiendo que compita a la par de otros productos. Como desafío, se

menciona que debería existir una metodología genérica para implementar IoT, altamente personalizable para un dominio específico [13]. Un aspecto importante que se menciona dentro del estudio es que el aseguramiento de la calidad en IoT es una nueva era de investigación.

Patrones y arquitecturas IoT

Debido a la amplia adopción de IoT en múltiples ámbitos como solución a necesidades específicas, Washizaki et al. [17], realizan una revisión sistemática con el fin de brindar una descripción general del panorama de la actualidad en temas como la arquitectura de sistemas y patrones de diseño para la IoT. Este propósito se logra mediante el análisis de un conjunto completo de patrones disponibles para IoT en la literatura. En este estudio, se encontraron 32 estudios embebidos entre los años 2014 al 2018, que tuvieron como finalidad ayudar a diseñar soluciones escalables y replicables mediante patrones de diseño, encapsulando los problemas y soluciones comunes reutilizándolos en contextos específicos. De los patrones extraídos en la presente revisión, se encuentra que aproximadamente el 57% de estos no son específicos de la IoT, lo que quiere decir que este tipo de sistemas se diseña comúnmente con arquitecturas y patrones de diseño convencionales. Como desafío, se encuentra que los patrones de arquitectura de IoT tienden a ser específicos del dominio, lo que implica que la naturaleza única de la adopción de IoT en dominios específicos aparece en el nivel de diseño de la arquitectura.

2.7. Propuestas metodológicas sobre calidad de soluciones de software basados en IoT

Ingeniería de software IoT

La industria y la investigación actual están influenciadas por una nueva generación de sistemas software escalables, altamente reactivos, y en algunos casos con recursos limitados muy característicos de la IoT. Estos sistemas por lo general se encuentran en ámbitos muy críticos, por ejemplo: la medicina, la automatización industrial y la gestión de energía. Larrucea et al [5], indican que en la actualidad existen una gran cantidad de sistemas IoT con una alta diversidad que es desconocida para los desarrolladores, trayendo como consecuencia que estos no estén preparados para ensamblar sistemas IoT de manera sistemática, afectando la calidad y las pruebas en aspectos como: (i) satisfacción de usuario, (ii) aspectos organizacionales, (iii) costos variables e (iv) implicaciones sociales. Además, se menciona que no ha surgido un conjunto consolidado de mejores prácticas de ingeniería de software para IoT, sin embargo, las técnicas de ingeniería del software pueden ser adaptadas, aprovechadas, re-pensadas para los desafíos que plantean este tipo de sistemas o si es necesario nuevos enfoques que impliquen tener en cuenta la gestión de configuración para softwares dinámicos y en continua re-configuración.

IoT Testing

El crecimiento de la IoT ha sido muy acelerado teniendo un gran potencial para solución de problemas de investigación. A pesar de este rápido crecimiento, en la actualidad se lanzan sistemas IoT sin las pruebas indicadas y adecuadas,

lo que causa serias dificultades en su calidad y de ninguna manera garantiza la satisfacción del usuario. Por ello Sirshar et al [9], revisan técnicas de prueba en usabilidad, confiabilidad, escalabilidad, compatibilidad de hardware-software, seguridad, rendimiento, conectividad, comparativa, piloto, funcional, red, interoperabilidad y APIs, usando inteligencia artificial y herramientas con el fin de generar casos de prueba. Este tipo de técnicas se clasifican por capas, las cuales son aplicación, servicio, red-puerta y sensor. La prueba de dispositivos y capas en IoT se vuelve una tarea desafiante, debido a que IoT es una arquitectura en la que el software está estrechamente entrelazado con el hardware, otro desafío que se menciona es la escalabilidad de este tipo de sistemas en contexto con la calidad de hardware, seguridad y problemas de privacidad.

Técnicas arquitectónicas IoT

Arakaki et al. [11], abordan cuán defectuoso podría ser un sistema de Internet de las cosas (IoT) si se implementa sin la calidad de la ingeniería. Para ello introduce los fundamentos de la arquitectura necesarios teniendo en cuenta referencias arquitectónicas sobre como diseñar y construir sistemas IoT mediante atributos de calidad presentes en la norma ISO/IEC 25010, con el fin de conseguir sistemas resilientes, flexibles, confiables y tolerantes a fallas. Además, el trabajo de Arakaki et al, muestran la infraestructura general de estos sistemas (actuadores, sensores y nube), indagando sobre cómo se debería evaluar la calidad teniendo en cuenta lo dicho anteriormente. Por ello, los autores plantean la necesidad de evaluar los atributos de calidad en los actuadores, sensores, nube y el sistema en general. Si se hace caso omiso a esta directriz se puede correr el riesgo de introducir defectos, desarrollando estos sistemas sin la calidad adecuada. El trabajo incluye un estudio de caso con el objetivo de mejorar aspectos de disponibilidad y mantenimiento aplicado al dominio de hogares inteligentes, utilizando un modelo de cadena de Markov. Se evidencia la necesidad de cómo construir un sistema IoT tomando en cuenta aspectos como los componentes fuertes (recursos de computación en la nube) y componentes débiles (sensores y actuadores).

Metodología para la evaluación de arquitecturas de aplicaciones IoT

Baños et al [14] desarrollan una metodología para evaluar la calidad de una aplicación IoT en términos de arquitecturas de software. La metodología está basada en los atributos de calidad propuestos por la ISO/IEC 25010. También, se aborda un conjunto de metodologías para evaluar las arquitecturas de estas aplicaciones, identificando cuales características y sub-características de calidad deben estar en un modelo de calidad. Para lograr este propósito, se identifican los atributos que permiten asegurar la calidad del software en aplicaciones IoT, se selecciona un grupo de metodologías de evaluación para arquitecturas IoT como referencia de la literatura y teniendo en cuenta los atributos mencionados anteriormente. Con base en las metodologías, los autores proponen una nueva metodología de evaluación de la calidad de software para este ámbito y, por último, valida dicha propuesta en aplicaciones IoT. Este trabajo también menciona que, dependiendo del campo, contexto o situación, algunas de estas características y sub-características toman más relevancia que otras. Además, no todos los modelos de evaluación de calidad

encontrados actualmente en la literatura pueden ser aplicados en el contexto de IoT.

2.8. Herramientas sobre calidad de soluciones de software basados en IoT

Calidad del servicio IoT

La determinación de características de calidad en aplicaciones IoT es muy importante ya que varía según los requisitos y funcionalidades del sistema, así como los dispositivos requeridos. Los requisitos de calidad no funcionales, son fundamentales en ese tipo de aplicaciones porque permiten la evaluación de estas mismas en términos de aplicabilidad y funcionalidad. Debido a esto, Kiruthika y Khaddaj [2], abordan la calidad de servicio (QoS) teniendo en cuenta la infraestructura del IoT compuesta por hardware, conectividad de red, arquitectura, algoritmos y software. De la misma forma, se clasifica la calidad del servicio en IoT en factores como: seguridad, rendimiento, usabilidad, fiabilidad, robustez, interoperabilidad y escalabilidad. Además, se abordan problemas de calidad del servicio considerando los factores fundamentales para diseñar y desarrollar modelos para sistemas IoT y, por último, teniendo en cuenta desafíos como, la estandarización de este tipo de sistemas, con el fin de mantenerse actualizados con la evolución de la tecnología.

Modelo de calidad IoT

Existen algunos enfoques convencionales de medición de la calidad de software que aún deben mejorarse y adaptarse para las necesidades de las aplicaciones IoT, por esta razón Tambotoh et al [3], realizan una descripción general del modelo de calidad del software para loT basado en ISO/IEC 25010 y los atributos de calidad de la información de COBIT 4.1 mediante la revisión de la literatura. En su trabajo, los autores examinan métricas correspondientes a los atributos del modelo y de calidad de la información mencionados anteriormente. Para esto, primero determinan y mapean las características del loT con sub-características y atributos en ISO/IEC 25010; luego, identifican los atributos de calidad que son consistentes con la calidad de la información en COBIT 4.1, y posteriormente, se crea una fórmula y su respectiva escala de medición de calidad y, finalmente, se propone un modelo de calidad para la IoT. El reto identificado es que, con millones de software y hardware conectados, se tenga en cuenta la calidad del servicio con el control de calidad en la IoT.

Kim [8], propone un modelo de calidad para aplicaciones IoT basado en la norma ISO/IEC 9126, el cual plantea cinco características de la IoT: (i) participación de dispositivos hardware, (ii) modelo de colaboración de dispositivos IoT, (iii) movilidad y conectividad, (iv) monitoreo remoto para dispositivos IoT y (v) recursos limitados (batería, memoria). Cada una de las 5 características de la IoT es descrita y mapeada con las características de calidad funcionalidad, confiabilidad, eficiencia y portabilidad propuestas por la ISO/IEC 9126. Durante el mapeo el autor planteo como los sub-factores de calidad afectan las características de la IoT. El modelo propuesto además aborda un conjunto de métricas para evaluar la calidad de aplicaciones IoT, considerando el mapeo realizado. El reto que se presenta en este estudio, es como medir la calidad de las aplicaciones IoT, ya que estas mismas son

una fusión compleja de tecnologías como redes inalámbricas, sensores y conectividad, teniendo en cuenta que es considerablemente diferente medir la calidad de softwares IoT a la de los convencionales.

Patrones y arquitectura IoT

Los dispositivos IoT contralados por aplicaciones poseen una conectividad, heterogeneidad y variedad muy alta ya que su función está en producir y compartir una gran cantidad de información. Por lo que Temkar y Bhaskar [4], adaptan el modelo de calidad ISO/IEC 25010 para cubrir las necesidades de aseguramiento de la calidad de las aplicaciones de IoT. Para ello mapean las características del modelo mencionado con las características de las aplicaciones IoT como dispositivos de hardware inteligente, colaboración entre software y hardware, conectividad de red (inalámbrica y móvil), monitoreo remoto de dispositivos IoT y recursos limitados de batería-memoria. El trabajo incorpora estas nuevas características de las aplicaciones IoT al modelo convencional e introducen un enfoque para diseñar la evaluación del aseguramiento de calidad mediante factores. Además, se detallan las sub-características de la norma con sus respectivas métricas para el respectivo aseguramiento de calidad. Todo lo anterior, plantea el desafío de concentrarse en el aseguramiento de la calidad de estas aplicaciones.

Medida de calidad IoT

Los sistemas IoT hoy en día tienen un gran impacto en nuestras vidas, debido a que automatizan ciertas tareas en la cotidianeidad, por lo cual estos sistemas deberían estar libres de defectos y ser de alta calidad. Abdallah et al [16], proponen un modelo de calidad para IoT que consiste en agregar un nuevo conjunto de características para precisar aún más en un modelo específico para esta tecnología utilizando como base la norma ISO/IEC 25010. Las características que se incluyen para lograr este propósito son: inteligencia, conectividad, enorme escala, censado, cambios dinámicos/naturaleza, heterogeneidad y seguridad. Este modelo se centró en todas las características relacionadas con estos sistemas, aportando factores de calidad que los miden. El reto que se presenta corresponde a que la calidad de estos sistemas debe medirse de manera diferente, teniendo en cuenta la presencia de entidades heterogéneas unidas conformando el sistema IoT.

Como resultado de toda la investigación realizada en los pasos anteriores, se obtiene una tabla resumen que contiene las quince referencias seleccionadas clasificadas por año, tipo de contribución, la validación realizada por los estudios clasificados, si emplean o no un modelo de referencia como la norma ISO/IEC 25010, ISO/IEC 9126 o COBIT 4.1 y por último el dominio de aplicación en el cual se realiza la propuesta seleccionada. Además, hemos incluido cómo se diferenciaría la propuesta de investigación de las presentes en el estado del arte.

2.9. Estudios adicionales investigados sobre calidad de soluciones de software basados en IoT

Según Rafique et al. [42], el desarrollo de productos software basados en internet de las cosas plantea desafíos enormes todo esto debido a la falta de marcos, herramientas y técnicas de desarrollo estándar que ayuden a los usuarios finales

con el objetivo de hacer menos complejos este tipo de productos. Por ello, este estudio plantea un marco de desarrollo de productos software basados en IoT, denominado como IADev, que usa el método ADD y Model Driven Development (MDD). Este marco se compone de dos partes principales, la primera que hace a referencia arquitectura iterativa mediante el diseño basado en atributos y la segunda, generación de modelos para guiar la transformación mediante MDD. IADev utiliza ADD para transformar los requisitos recolectados en una arquitectura al considerar todas las preocupaciones de parte de los interesados. Como desafío principal se encuentra que el desarrollo de un ciclo de vida estándar para un producto software basado en IoT es esencial cumpliendo con los requisitos de las partes interesadas, abordando todas las fases del desarrollo y satisfaciendo los paradigmas de implementación heterogéneos.

IoT es un paradigma que puede cambiar la vida cotidiana como la conoces en diferentes temas o contextos como la agricultura o la alimentación. Según Verdow et al. [43], se propone, desarrollo y aplica un marco de arquitectura para modelar productos software basados en IoT en el dominio de la agricultura y la alimentación. Este marco está compuesto por un conjunto de puntos de vista coherentes que sirven de guía para modelar arquitecturas individuales de estos productos. También, se menciona que debido a las diferentes inquietudes que deben abordarse para una gran cantidad de productos existentes, no se debe fijar tan solo un conjunto de puntos de vistas, sino que se pueden introducir múltiples puntos de vista. Un método sugerido para llevar esto acabo es VaB [43], que proporciona un mecanismo para adaptar o agregar los diferentes puntos de vista. El desafío que se menciona en el estudio descrito anteriormente es sobre como contribuir con la enorme heterogeneidad para la adopción a gran escala de la IoT.

La seguridad es un aspecto muy importante en la IoT, por ello Aman y Snekkenes [44], plantean seguridad adaptativa para productos software basados en IoT, que se describe como la forma de compensación autónoma y dinámica donde se toman decisiones en tiempo de ejecución, este atributo es deseable y clave para este paradigma. En el presente estudio, se plantea un enfoque basado en escenarios para evaluar y reconocer situaciones típicas de compensación. Además, se demuestra que es posible automatizar una respuesta de mitigación de compensación óptima para IoT mediante la evaluación de requisitos contextuales, QoS, preferencias del usuario, las capacidades de los objetos inteligentes y el riesgo enfrentado en tiempo de ejecución. En este estudio, ATAM sugirió un enfoque basado en escenarios para realizar un análisis a los enfoques de diseño que abordan varios atributos de QoS en arquitecturas software. Este estudio plantea que debido a la complejidad de las arquitecturas IoT, es un desafío reconocer, evaluar y modelar posibles situaciones de compensación utilizando seguridad adaptativa.

En la época actual ha cobrado una gran importancia el tema de la disminución de los recursos naturales, las tierras donde se puede cultivar y las condiciones impredecibles del clima y el medio ambiente. Por ello, en el presente estudio [45], se realiza una encuesta de soluciones de IoT para demostrar cómo se puede integrar IoT en el sector de la agricultura inteligente, con el fin de discutir la visión de los ecosistemas de agricultura inteligente disponibles para IoT mediante la evaluación de su arquitectura (dispositivos IoT, tecnologías de comunicación, almacenamiento y procesamiento de big data), sus aplicaciones y el cronograma de investigación. Por otro lado, se estudian cuáles son las tendencias y oportunidades

de las aplicaciones de IoT para la agricultura inteligente y también se plantean los problemas abiertos y los desafíos de la aplicación de IoT en la agricultura inteligente.

La llegada de IoT ha promovido nuevas ideas y dominios de investigación innovadores ya que el paradigma mencionado anteriormente apenas empieza a manifestarse. En este sentido se abren las posibilidades para experimentar en el dominio de la agricultura inteligente. En [46], Ray revisa varias aplicaciones asociadas con IoT, problemas y desafíos específicos relacionados con la implementación de IoT para mejorar la agricultura, centrándose en los requisitos específicos, analizando exhaustivamente las tecnologías y dispositivos de comunicación inalámbrica en aplicaciones agrícolas y ganaderas. También, se analizan varios estudios de casos para explorar las soluciones basadas en IoT existentes, hechas por varias organizaciones e individuos, categorizando de acuerdo con sus parámetros de implementación. Por último, se plantean las dificultades relacionadas con estas soluciones y se identifican los factores de mejora junto con la futura hoja de ruta de trabajo utilizando el IoT.

La agricultura es uno de los ejes más importantes de la economía global ya que satisface la gran demanda de productos alimenticios debido al incremento de la población mundial, con el objetivo de mejorar y modernizar las metodologías agrícolas tradicionales. IoT tiene el potencial de ser el mediador y/o facilitador clave para que se haga realidad la visión de la agricultura inteligente. En el presente estudio [47], se propone una arquitectura IoT centrada en el usuario para abordar los diversos problemas que se enfrentan en el dominio agrícola. Este sistema permite a los agricultores monitorear sus campos agrícolas en tiempo real, recibiendo recomendaciones para producir mejores cultivos. La arquitectura propuesta tiene como objetivo mejorar la cadena de suministro de alimentos permitiendo a los agricultores maximizar sus ganancias. La aplicabilidad de la arquitectura propuesta se evalúa utilizando múltiples casos de uso que abarcan los diferentes aspectos del proceso agrícola. También, se propone un marco novedoso para teléfonos inteligentes que facilita a los ingenieros de software desarrollar las aplicaciones e implementar varias funciones del sistema propuesto.

En [48], se propone una arquitectura IoT de bajo costo basada en tecnología de Red de Sensores Inalámbricos para monitoreo agrícola, con características especiales como la de ser desplegable en diferentes tipos de cultivos, en este caso particular el cultivo de cacao. Todo esto, con el objetivo de monitorear y almacenar información de diversos factores climáticos y de suelo que inciden en el óptimo crecimiento de los cultivos pertenecientes a pequeños y medianos productores agropecuarios mediante una aplicación multiplataforma (web, móvil, etc.). Esta propuesta corresponde a una herramienta integradora del campo agrotecnológico para los productores de cacao a través de la recolección, almacenamiento, gestión y visualización de datos de variables agrícolas. Como trabajo futuro, se realizará un estudio más profundo en temas como la comunicación y el consumo de energía. También, se pretende replicar el diseño con su respectiva implementación en cafetales.

La escasez de agua, la disponibilidad de tierras cultivables, las plagas y el cambio climático hacen que la agricultura se vea afectada y con el paso del tiempo deba satisfacer una gran demanda de alimentos, por ello debe ser un poco más óptima. IoT puede ayudar en ese sentido ya que puede brindar información oportuna y

precisa sobre alertas relacionadas con los cultivos, las precipitaciones, las plagas y la nutrición del suelo, lo que puede ayudar a mejorar el rendimiento. En [49], se muestra el uso de IoT en la agricultura con sus procesos asociados como la gestión del suministro de agua, control de plagas, gestión del rendimiento y seguridad. También, se describe el análisis de datos basado en la nube que actúa como back-end del ecosistema de sensores de IoT y ayuda a optimizar el uso de recursos, de igual forma mejorando el tamaño del rendimiento. Como desafío se plantea que para los países en desarrollo han estado esperando tecnologías para aumentar y mejorar la cantidad y calidad de su agricultura.

La agricultura es esencial para la economía de cualquier país, según las problemáticas actuales como que las técnicas que se utilizan no son eficientes, la mano de obra es cada vez mayor, el riego debe ser a tiempo y la aplicación de fertilizantes. Se plantea que IoT puede ayudar a mejorar la agricultura mejorando la eficiencia y la productividad mediante varios nodos de sensores que se utilizan para monitorear el nivel de acidez del suelo, la temperatura y otras variables. En el presente documento [50], se propone principalmente en el uso de IoT en la agricultura mediante la arquitectura propuesta que conduce al crecimiento exponencial de la agricultura y la economía. El desafío que se plantea, es que IoT necesita aprender de las técnicas actuales y a su vez automatizar tareas haciendo la vida más fácil a los humanos.

Lo sistemas software basados en IoT tienen una característica particular que permiten realizar operaciones en cualquier momento y lugar, esto se denomina como ubicuidad, con el objetivo de interconectar sistemas, dispositivos, humanos y servicios configurando sociedades digitalizadas. La arquitectura de software como modelo abstrae las complejidades de las fases del software como lo son: (i) modelado, (ii) diseño, (iii) desarrollo y (iv) evolución con el fin de diseñador sistemas complejos basados en IoT de manera eficiente y eficaz. En [18], se presenta un mapeo sistemático en el cual se resume conocimiento acerca de los principios y prácticas arquitectónicas con el objetivo de facilitar la transferencia de conocimientos que beneficia a la industria y a la academia sobre el papel que juega la arquitectura en los sistemas software basados en IoT.

Los anteriores tres trabajos dejan ver la relevancia de incluir y adaptar prácticas de la arquitectura en el contexto de la IoT. En esta tesis de maestría buscamos integrar las prácticas de arquitectura presentes en éstos y otros métodos de arquitectura. El objetivo es definir un conjunto de prácticas articuladas y consistentes con éste tipo de sistemas.

2.10. Aporte Significativo

Con respecto a las investigaciones, trabajos, estudios y brechas identificadas durante el proceso de revisión del estado del arte, la presente propuesta de trabajo de grado busca establecer un conjunto de prácticas de arquitectura y sus requisitos de calidad para el desarrollo de productos software basados en IoT teniendo en cuenta las características especiales de este paradigma apoyándose en la norma ISO/IEC 25010 favoreciendo la calidad en este tipo de productos. Este trabajo brindaría a la industria mecanismos metodológicos útiles para abordar el diseño arquitectónico de productos software intensivos en IoT. Desde el punto de vista científico permitiría avanzar en el conocimiento de la arquitectura de software y los retos que enfrenta en nuevos

escenarios como el que le ofrece la IoT. Además, desarrollar una buena arquitectura de software junto con buenas prácticas aportan en la evaluación, modelado, desarrollo y ejecución fortaleciendo la calidad y funcionalidad de los productos software basados en IoT. Una buena arquitectura también ayuda en la abstracción de los componentes hardware heterogéneos garantizando operación fluidas incluyendo los protocolos de red. El grupo IDIS viene avanzando significativamente en esta disciplina de la ingeniería de software, considerando nuevos escenarios como blockchain, computación en el borde y el software en la robótica y sus particularidades. Desde el punto de vista académico, el conjunto de prácticas puede ser aprovechado por cursos de diferentes programas de pregrado en los que aborden la IoT como tema central o como contexto de diseño (Sistemas, Electrónica, Automática y Mecatrónica).

3. CAPITULO 3

Caracterización de los atributos de calidad especiales del producto software basado en IoT y de las prácticas de arquitectura de software

Descripción Capitulo

Para el desarrollo metodológico de la presente propuesta se realizó un Mapeo Sistemático (MS) siguiendo los enfoques planteados por Kitchenham [22], que propone una guía para revisiones sistemáticas apropiadas para investigadores de ingeniería del software y Petersen et al. [23], que propone una guía para realizar estudios de mapeo sistemático en ingeniería de software. Para realizar la construcción del conocimiento fue necesario indagar sobre la temática seleccionada mediante la revisión detallada y cuidadosa de documentos relacionados con el tema de investigación propuesto. Este capítulo presenta los procedimientos cumplidos a fin de identificar y caracterizar los atributos de calidad especiales del producto software basado en IoT y de las prácticas de arquitectura de software.

3.1. Proceso metodológico para la caracterización de los atributos de calidad especiales del producto software basado en IoT y de las prácticas de arquitectura de software

A continuación, se sintetiza el estado del arte derivado de los resultados de este mapeo sistemático y posteriores derivados de la evolución de ésta propuesta de investigación.

Para la creación de la cadena de búsqueda, se identificaron los términos principales y sinónimos alternativos, con los cuales se lograsen obtener el mayor número de estudios primarios relevantes. Como resultado se establecieron las siguientes cadenas de búsqueda (Ver tabla 1):

“Software Quality”, “Quality Model”, “Quality Characteristics”, “Quality Assurance”, “Quality Assessment”, “Non Functional Requirement”, “IoT” e “Internet of things”.

Términos principales	*Términos alternativos*
Software Quality Internet of things Quality Model Quality Characteristics	(“Software Quality” OR “Quality Model” OR “Quality Characteristics” OR “Quality Assurance” OR “Quality Assessment” OR

Non Functional Requirement	“Non Functional Requirement”) AND (“IoT” AND “Internet of things”)

Tabla 1. Cadena de busqueda

Para la presente propuesta, se tuvo en cuenta el siguiente proceso para establecer el estado del arte correspondiente (Ver figura 1).

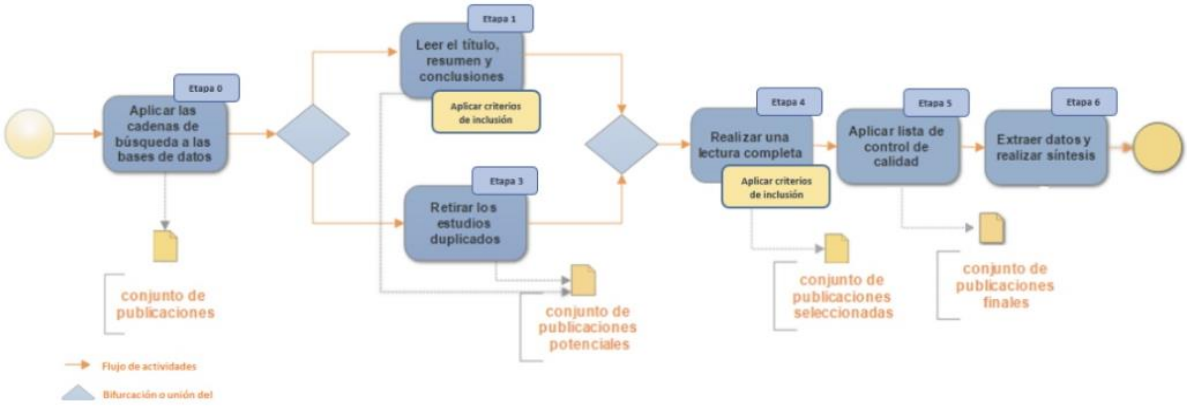


Figura 1. Estrategia de busqueda

La tabla 2 muestra el resumen de los elementos de la estrategia de búsqueda, incluidos los motores de búsqueda, el tipo de publicación, los criterios aplicados y el período de publicación.

Estrategias de búsqueda	
Bases de datos consultadas	Scopus, Science Direct e IEEE Xplore
Ítems objetivos	Artículos de investigación, Artículos de conferencias, Libros, Reportes técnicos y Trabajos de grado
Búsqueda aplicada a	Título, Abstract y Keywords
Lenguaje	Estudios escritos en inglés o español
Periodo de publicación	Desde el 2015 hasta 2021

Tabla 2. Estrategias de búsqueda

La búsqueda solo considera los estudios académicos y profesionales realizados en conferencias, revistas, reportes técnicos o trabajos de grado. Este trabajo excluyó blogs personales, páginas web u obras escritas en español, a excepción de algunos trabajos en español que se incluyeron debido a su contribución a la investigación como se muestra en la tabla 3. No se consideran trabajos que no se centren en la calidad de un producto software basado en IoT y, además, se excluyeron trabajos duplicados de la misma búsqueda en diferentes bases de datos y resúmenes de talleres.

Criterios de Inclusión/Exclusión	
Criterios de inclusión	• Términos incluidos en la cadena de búsqueda • Estudios realizados en artículos, conferencias, reportes técnicos o trabajos de grado • Estudios escritos en inglés y español • Publicaciones desde el 2000 hasta 2021. • Estudios que establezcan como asegurar la calidad de un producto software basado en IoT o estudios que planteen como fortalecer la calidad de un producto software basado en IoT .
Criterios de exclusión para título y abstract	• Estudios que no aborden la calidad de un producto software basado en IoT. • Estudios que solo presentan resúmenes o contenido en diapositivas. • El contenido de páginas web, blogs personales o folletos.
Criterios de exclusión para el texto completo	1. Estudios que presentan resúmenes de un workshop

Tabla 3. Resumen de criterios de inclusión y exclusión

Según [23], la evaluación de la calidad es más esencial en las revisiones sistemáticas para determinar el rigor y la relevancia de los estudios primarios. En los mapeos sistemáticos no es necesario realizar una evaluación de calidad, pero se sugiere hacer uso de la evaluación anteriormente mencionada. El uso de la clasificación de los tipos de investigación según la categoría de propuestas de solución mencionados en [51] contendría estudios sin evidencia empírica o científica. Los estudios en esta categoría generalmente no suelen ser incluidos en una revisión sistemática, aunque en mapeos sistemáticos son importantes para detectar tendencias de los temas en los que se trabaja.

En el presente estudio, se incluyeron hallazgos con y sin evidencia empírica como lo sugiere lo mencionado anteriormente. Para refinar los estudios seleccionados, se utiliza la lista de chequeo propuesta por [52] (Ver tabla 4), donde hay tres posibles respuestas (Si = un punto, No = cero puntos y parcialmente = medio punto), utilizando el primer cuartil (6/3 = 2) como punto de corte, si una publicación obtuvo menos de 2 se excluye de la lista de estudios finales para evitar trabajos de baja calidad.

Tras aplicar toda la estrategia de búsqueda descrita anteriormente, se encuentran los siguientes estudios y serán descritos a continuación. Además, se estructuraron estos mismos en las siguientes secciones: (i) estudios, (ii) propuestas metodológicas y (iii) herramientas sobre calidad de soluciones de software basados en IoT. Dentro de cada sección también se clasificaron las investigaciones por dominio en la Ingeniería de software.

#	Pregunta
QA1	¿El objetivo del estudio está lo suficientemente bien explicado?
QA2	¿Se explica claramente la idea, enfoque y limitaciones presentadas?
QA3	¿ Se tienen en cuenta las amenazas en contra de la validez?
QA4	¿Existe una descripción adecuada del contexto en el que el estudio se llevó a cabo?

QA5	¿Están los resultados de la investigación claramente establecidos?
QA6	¿Se diseñó el estudio para lograr estos objetivos?

Tabla 4. Lista de chequeo

A continuación, se muestra la tabla 5 con los resultados del proceso anterior y un resumen de los estudios seleccionados.

Cadena de búsqueda	*Resultados encontrados*		*Resumen del título, abstract y conclusiones*	*Seleccionados*
1	Scopus	518	105	15
	Science Direct	695		
	Ieee Xplore	103		
	Total	1316		

Tabla 5. Resumen de resultados tras aplicar la estrategia de búsqueda

3.2. Clasificación y selección de los atributos de calidad especiales del producto software basado en IoT

Tras aplicar toda la estrategia de búsqueda descrita anteriormente, se encuentran los siguientes estudios y serán descritos a continuación. Además, se estructuraron estos mismos en las siguientes secciones: (i) estudios, (ii) propuestas metodológicas y (iii) herramientas sobre calidad de soluciones de software basados en IoT. Dentro de cada sección también se clasificaron las investigaciones por dominio en la Ingeniería de software. Además, hemos incluido cómo se diferenciaría la propuesta de investigación de las presentes en el estado del arte. Los resultados pueden observarse en la Tabla 1, que se describe a continuación:

No.	***Referencia***	***Año***	***Tipo de contribución***	***Validación***	***Modelo de referencia***	***Dominio (Ingeniería de Software)***
1	[4]	2021	Herramienta	Escenarios de aplicación	ISO/IEC 25010	Aplicaciones específicas IoT
2	[6]	2021	Estudio	Validación Teórica	ISO/IEC 25010	Requerimientos no funcionales IoT
3	[11]	2020	Propuesta metodológica	Estudio de caso	ISO/IEC 25010	Técnicas arquitectónicas IoT
4	[17]	2020	Estudio	Validación Teórica	No especifica	Patrones y arquitecturas IoT
5	[10]	2019	Estudio	Validación Teórica	No especifica	IoT testing
6	[16]	2019	Herramienta	Validación Teórica	ISO/IEC 25010	Medida de Calidad IoT
7	[9]	2019	Propuesta metodológica	Validación Teórica	No especifica	IoT testing
8	[12]	2019	Estudio	Validación Teórica	ISO/IEC 25010	Aseguramiento de calidad IoT

No.	*Referencia*	*Año*	*Tipo de contribución*	*Validación*	*Modelo de referencia*	*Dominio (Ingeniería de Software)*
9	[14]	2017	Propuesta metodológica	Estudio de caso	ISO/IEC 25010	Metodología para la evaluación de calidad IoT
10	[3]	2017	Herramienta	Validación Teórica	ISO/IEC 25010 y COBIT 4.1	Modelo de calidad IoT
11	[8]	2016	Herramienta	Estudio de caso	ISO/IEC 9126	Modelo de calidad IoT
12	[5]	2016	Propuesta metodológica	No valida	No especifica	Ingeniería de software IoT
13	[15]	2016	Estudio	Validación Teórica	No especifica	Aseguramiento de calidad IoT
14	[2]	2016	Herramienta	Ilustración	No especifica	Calidad del servicio IoT
15	[13]	2015	Estudio	Validación Teórica	No especifica	Técnicas y metodologías IoT
16	Esta propuesta	2023	Conjunto de prácticas	Estudio de caso	ISO/IEC 25010	Arquitectura de Software

Tabla 6. Resumen caracterización artículos

Una vez se identificaron los estudios primarios se realizó el proceso para el desarrollo del primer objetivo de investigación que plantea la caracterización de los atributos de calidad especiales del producto software basado en IoT y de las prácticas de arquitectura de software. A continuación, se muestran los resultados obtenidos tras la caracterización de los estudios anteriormente mencionados:

Q1. ¿Cuáles atributos de calidad según la norma ISO/IEC 25010, especiales y dominios se han utilizado en el contexto de desarrollo de software basado en IoT?

De los quince artículos mencionados, en la tabla anterior, se realiza la clasificación de los mismos teniendo en cuenta los atributos de calidad enunciados por la norma ISO/IEC 25010. Los atributos de calidad son: (i) adecuación funcional, (ii) eficiencia de desempeño, (iii) compatibilidad, (iv) usabilidad, (v) fiabilidad, (vi) seguridad, (vii) mantenibilidad, (viii) portabilidad, y (ix) otros como conectividad, monitoreo, recursos limitados, escalabilidad, flexibilidad, sensibilidad al contexto, etc. (que hacen referencia a atributos especiales de la IoT que no son mencionados en la norma ISO/IEC 25010).

Estudios		**Atributo de calidad de la norma ISO/IEC 25010**								
No	*Referencia*	*Adecuación Funcional*	*Eficiencia de desempeño*	*Compatibilidad*	*Usabilidad*	*Fiablidad*	*Seguridad*	*Mantenibilidad*	*Portabilidad*	*Otros*
1	[4]	X	X	X	X	X	X	X	X	Conectividad Monitoreo Recursos Limitados
2	[6]			X			X			Escalabilidad Flexibilidad Sensibilidad al contexto

3	[11]	X	X	X	X	X	X	X	X	
4	[17]		X	X	X	X	X	X	X	Escalabilidad Replicabilidad Privacidad
5	[10]			X		X	X			Privacidad Confianza
6	[16]	X	X	X	X	X	X	X	X	Extendibilidad Escalabilidad Efectividad Correctitud Oportunidad Eficiencia de costo Robustez Reusabilidad Eficiencia Verificacion Validacion Inteligencia Conectividad Escala Censado Cambios Dinamicos Heterogeneidad Movilidad Conectividad Monitoreo Recursos Limitados
7	[9]		X	X	X	X	X			Escabilidad Compatibilidad HW-SW Conectividad Privacidad
8	[12]			X			X			Privacidad Sensibilidad al contexto Elasticidad Escalabilidad
9	[14]		X	X		X	X	X	X	
10	[3]	X	X	X	X	X	X	X	X	
11	[8]	X	X			X			X	Movilidad Conectividad Monitoreo Recursos limitados
12	[5]									
13	[15]						X			Entorno Usuario
14	[2]		X	X	X	X	X			Escalabilidad Robustez
15	[13]	X	X	X		X	X	X	X	Facilidad de uso Verificacion

										Validacion Extendibilidad Efectividad Oportunidad Rentabilidad
Total		6	10	12	7	11	13	7	8	

Tabla 7. Clasificacion de atrbutos de calidad de acuerdo a los articulos encontrados

Entre los atributos de calidad encontrados realizando la respectiva clasificación y selección de los mismos, se puede concluir que, los atributos **menos abordados** en los trabajos mencionados anteriormente que están asociados a la norma ISO/IEC 25010 en el producto software basado en IoT son **adecuación funcional (6), usabilidad (7), mantenibilidad (7) y portabilidad (8)**. Por el contrario, los **más abordados** son **eficiencia de desempeño (10), compatibilidad (12), fiabilidad (11) y seguridad (13).** En cuanto a los atributos espaciales que se sugiere abordar en el producto software basado en IoT, tras la clasificación y selección de los mismos son **escalabilidad, privacidad, sensibilidad al entorno, confianza, robustez, conectividad, movilidad, monitoreo de recursos, compatibilidad hardware-software y durabilidad.**

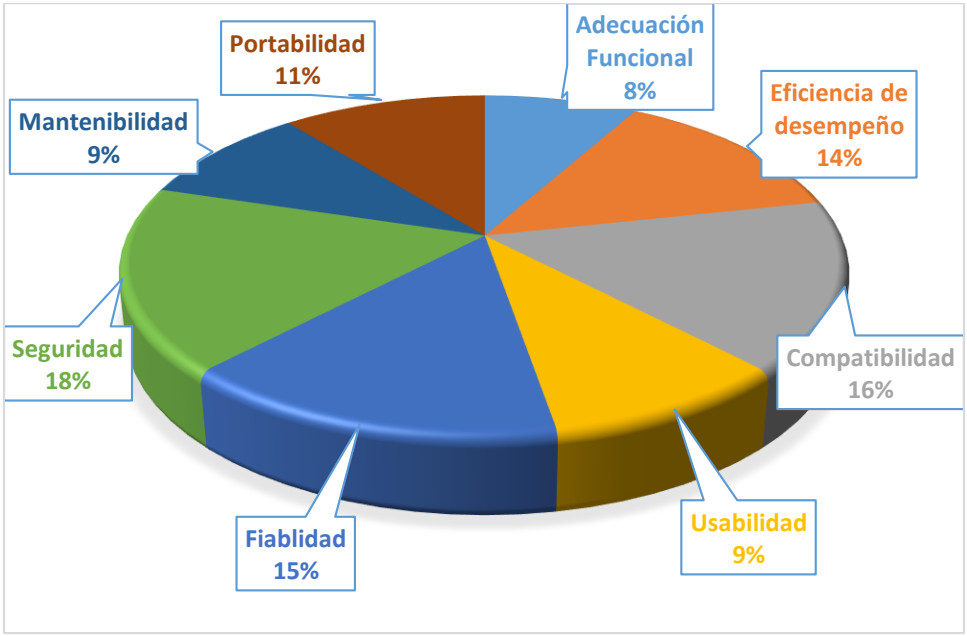


Figura 2. Resultados atributos usados NORMA ISO/IEC 25010

Por otra parte, también se realiza la clasificación y selección del dominio de aplicación de los productos software basados en IoT con el fin de establecer que dominios son los actualmente abordados en la industria.

Estudios		**Dominios de aplicación**					
No	Ref	Ciudades Inteligentes	Hogares Inteligentes	Agricultura	Fabricas Inteligentes	Cuidado de la salud	Otros
1	[4]						
2	[6]	X	X	X	X	X	

3	[11]		X				
4	[17]				X		Gestión Inteligente Blockchain Entrega de información de emergencia
5	[10]						
6	[16]						
7	[9]	X				X	
8	[12]	X	X			X	
9	[14]				X	X	Logística Vestibles Seguridad Industrial Agroindustria Medio Ambiente Gobierno
10	[3]						
11	[8]						
12	[5]						
13	[15]	X	X		X	X	Logística Transporte
14	[2]						
15	[13]	X					
Total		5	4	1	4	5	

Tabla 8. Clasificacion de los dominios usados en IoT de acuerdo a los articulos encontrados

Entre los dominios de aplicación de los productos software basados en IoT, se encontró que el dominio menos abordado en los trabajos mencionados anteriormente, e: **agricultura (1)**. Por otra parte, los dominios más abordados son **cuidado de la salud (5), ciudades (5), hogares (4) y fabricas inteligentes (4).** En la literatura, se sugiere tratar dominios como **logística, agroindustria, medio ambiente y transporte.**

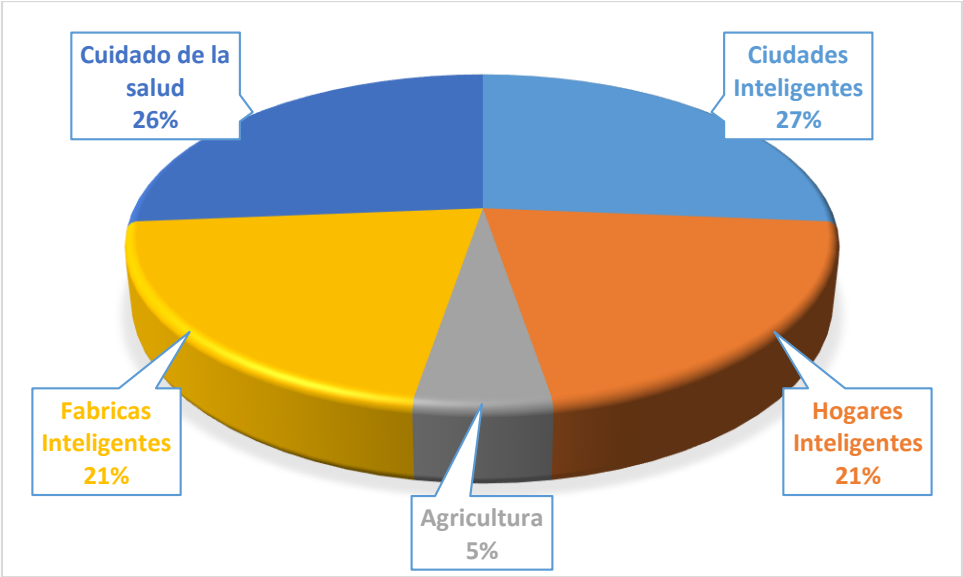


Figura 3. Resultados dominios usados

3.3. Clasificación y selección de las prácticas de arquitectura de software usadas en el producto software basado en IoT

De los quince artículos mencionados, en la tabla anterior, se realiza la clasificación de los mismos teniendo en cuenta las prácticas de arquitectura para el producto software basado en IoT mencionadas en los trabajos de forma general y también de manera específica las prácticas para estos productos en el dominio especifico de la agricultura.

Q2. ¿Cuáles prácticas de arquitectura de software se han utilizado en el contexto de desarrollo de software basado en IoT?

La arquitectura es un aspecto relevante ya que ayuda en la abstracción de los componentes de hardware heterogéneos, garantiza operaciones fluidas mediante protocolos de red y aporta en la calidad de temas computacionales relacionados con la IoT [18]. Alcanzar algunos atributos de calidad requiere de una adecuada implementación y compresión en la arquitectura con el fin de considerar los efectos que estos puedan ocasionar sobre las soluciones y las organizaciones.

Como la arquitectura de software se tiene en cuenta en las etapas tempranas del ciclo de desarrollo software, es importante determinar la arquitectura correcta, cumpliendo los requisitos de calidad planteados y representándolos como escenarios concretos de atributos de calidad con el objetivo de contribuir en el éxito de cualquier proyecto [19].
En resumen, el principio de la arquitectura de software junto con buenas prácticas pueden aportar en la evolución, modelado, desarrollo y ejecución de productos software complejos basados en IoT, fortaleciendo la calidad y funcionalidad requeridas, así como en el desarrollo de soluciones arquitectónicas de nueva generación para paradigmas como IoT [18]. En lo referente a las prácticas de arquitectura de software usadas en el producto software basadas en IoT, se encontraron las siguientes prácticas tras la revisión de los trabajos relacionados anteriormente.

1. Realizar configuraciones independientes con arquitecturas distribuidas utilizando simulación de dispositivos físicos [10].

2. Para el atributo de seguridad se aconseja verlo en capas como lo son capa de aplicación, transporte y percepción [12] [53].

3. Utilizar y verificar el cumplimiento/acuerdo de nivel de servicio, organización, seguridad y su forma de gestionar los datos [15].

4. Implementación de IoT con arquitecturas y metodologías personalizables para cada dominio [13].

5. Utilizar el listado de Estilos arquitectónicos, patrones de arquitectura y diseño propuesto por [17] que puede ayudar para la implementación de arquitecturas para el producto software basado en IoT reduciendo costos y tiempo.

6. Visualizar la solución IoT desde el elemento más simple hasta el todo: Atributos de calidad para sensores, Actuadores, nube y el sistema IoT en general [11].

7. Utilizar el método planteado por kazman, clement y bass en software architecture in practice con el fin de diseñar y seleccionar mecanismos que se aplican para obtener

la calidad final del sistema IoT. Lo anterior se logra utilizando el método sugerido y será aplicado mediante un ejemplo a continuación:

- Controlar la temperatura del ambiente, es decir, controlar valores obtenidos por el sensor
- Obligatorio: Evitar Valores Erróneos
- Usar Diagrama de táctica arquitectónica
- Tácticas: Aspectos de calidad + táctica + mecanismos técnicos

    1. Obtener valores
    2. Verificar consistencia de datos: Mayor Confiabilidad
    3. Preparar recursos para una eventual recuperación de una ejecución que falla
- Lógica a seguir: Se deben adoptar múltiples mecanismos de solución

Adoptar un método así es inusual en proyectos ad hoc, las consecuencias que se obtienen son mecanismos pobres [11].

8. Configurar pesos o ponderar las características/sub-características de acuerdo al dominio con el fin de que la calidad sea personalizable [14].

9. Entender la IoT como el conjunto de Hardware, conectividad, Arquitectura, algoritmos y software [2].

10. Apoyar la calidad de software mediante la calidad del proceso, servicio e información [3].

11. Definir mediante la participación de hardware y software, modelo de colaboración de dispositivos y dominio de aplicación así mismo cuales atributos serán protagonistas [4], [8], [16].

12. Realizar el mapeo de las características de la norma ISO/IEC 25010 junto con las características especiales de la IoT con el objetivo de ver las relaciones existentes entre las mismas.

13. Ponderar las relaciones anteriormente descritas con el objetivo de darle más énfasis a unas que otras según el dominio.

14. Combinar las dos prácticas (12 y 13) junto con las tácticas arquitecturales con el fin de incrementar la calidad y sus atributos.

15. Conformar la IoT como una serie de bloques funcionales para facilitar diversos beneficios del sistema como detección, identificación, actuación, comunicación y gestión. Los bloques anteriormente sugeridos son: aplicaciones, mantenimiento, servicio, comunicación, seguridad y dispositivo [46].

Como bien se sabe el dominio de aplicación de IoT es un aspecto que hay que tener en cuenta cuando se van a desarrollar productos que utilicen este paradigma. La llegada de IoT ha promovido nuevas ideas y dominios de investigación innovadores ya que el paradigma mencionado anteriormente apenas empieza a manifestarse. En este sentido se abren las posibilidades para experimentar en el dominio de la agricultura inteligente.

Por ello, se hace necesario realizar un análisis que permita desarrollar productos software basados en la IoT mediante la evaluación de su arquitectura (dispositivos IoT, tecnologías de comunicación, almacenamiento y procesamiento de big data), sus aplicaciones, funcionalidad y calidad requeridas. Además, se tiene en cuenta que en la anterior clasificación el dominio de la agricultura es uno de los menos abordados y es necesario aportar en dicho tema. En el siguiente apartado, se encontraron las siguientes prácticas tras la revisión de los trabajos adicionales en el dominio de la agricultura inteligente.

1. Como las condiciones meteorológicas son variables en todo el mundo ya que las propiedades de temperatura, humedad del suelo y ambiental, suelo y agua son muy diferentes se sugiere que la arquitectura de los productos software IoT permita conectarse a los centros meteorológicos nacionales con el fin de informar sobre las condiciones actuales del clima para optimizar algunas variables en la producción de cultivos [46].

2. La agricultura actual requiere de sistemas de riego que involucren IoT, con el fin de optimizar el uso del agua. Por ello, la literatura sugiere utilizar cuatro ítems esenciales en arquitecturas de productos software IoT que usen el riego inteligente como lo son (i) integración de datos meteorológicos, (ii) control del sistema mediante Wifi o Ethernet (Conexión inalámbricas LoRA), (iii) sincronización con sensores de humedad y (iv) reducción de facturas mensuales todo esto con el objetivo de conservar los recursos hídricos limitados [46].

3. El control de plagas y enfermedades es muy importante para agricultura inteligente ya que se puede aumentar la calidad de los productos agrícolas minimizando costos a través del control del uso de pesticidas y fertilizantes, para que esto ocurra se aconseja integrar, monitorear y gestionar la probabilidad/ocurrencia de las plagas en cultivos mediante el uso de IoT en sus respectivas arquitecturas [46].

4. La calidad del agua para la agricultura inteligente es una necesidad que requiere de bastante atención ya que muchos de tópicos que la componen requieren de una buena calidad del agua, por ello se aconseja monitorear el agua mediante parámetros químicos y físicos relacionados a sensores que se integran en la arquitectura como conductividad, oxígeno disuelto, turbidez, temperatura y pH. Así mismo como el monitoreo del suelo ya que es crucial para el dominio agrícola [46].

5. La gestión de la cadena de suministros es una tarea importante en la cual se debe hacer hincapié debido a que el agricultor debe obtener ganancias y a su vez eficiencia operativa, por ello se sugiere que la IoT monitorice procesos comerciales incluidos en la arquitectura con el fin de generar eficiencia en dicha cadena ayudando al agricultor [46].

6. Se sugiere que los productos software basados en IoT que trabajen sobre el dominio de la agricultura contengan un sistema de alerta incluido en la arquitectura, que permita controlar las condiciones climáticas y del suelo variables, con el fin de controlar variaciones que puedan influir en la eficiencia de la plantación o sugerirle al agricultor que acción se puede tomar según el análisis de los dispositivos conectados al sistema [45], [48].

4. CAPITULO 4

Organizar a nivel conceptual y metodológico un conjunto de prácticas de arquitectura y sus requisitos de calidad que contribuyan al desarrollo del producto software basado en IoT a partir de la caracterización realizada anteriormente.

Descripción Capitulo

En la presente sección, se realiza la organización de las prácticas halladas en la caracterización anteriormente realizada consultando diferentes autores como los Métodos SEI, ESI europeo, Fraunhofer, etc. Una vez, se encontró el autor indicado se procede a realizar la clasificación y selección de acuerdo a la definición propuesta por el Object Management Group (OMG) en el documento Kernel and Language for Software Engineering Methods (Essence v1.2 Pagina 83) [54] al igual que sus requisitos de calidad, seleccionado los atributos más sensibles para el software basado en IoT y que estén relacionados con las prácticas halladas en el anterior capitulo. Así mismo como, el planteamiento de un nuevo marco de trabajo en el cual se identifican algunas etapas relevantes para un perfil de calidad en productos software basados en IoT, el cual tendrá como objetivo permitir la organización y agrupamiento de las prácticas y la descripción de dos prácticas correspondientes a las actividades de diseño y refactorización.

4.1. Atributos de calidad identificados para productos software basados en IoT

Los atributos de calidad menos abordados y lo más abordados hallados en el anterior capitulo que están asociados a la norma ISO/IEC 25010 y con las prácticas de arquitectura son:

#	**Atributo de calidad**
A1	Adecuación funcional
A2	Usabilidad
A3	Mantenibilidad
A4	Portabilidad
A5	Escalabilidad
A6	Privacidad
A7	Sensibilidad al entorno
A8	Confianza
A9	Robustez
A10	Conectividad
A11	Movilidad
A12	Monitoreo de recursos
A13	Compatibilidad hardware-software
A14	Durabilidad
A15	Eficiencia de desempeño
A16	Compatibilidad
A17	Fiabilidad
A18	Seguridad

Tabla 9. Catálogo de atributos de calidad encontrados tras el proceso metodologico en el capitulo 3

Estos atributos de calidad son sensibles para el software basado en IoT por eso, es necesario enfocarse en este grupo ya que el aporte que se puede realizar a nivel técnico abriendo más las posibilidades para el correcto desarrollo y gestión de este tipo de productos software.

4.2. Prácticas de diseño de la arquitectura identificadas para productos software basados en IoT

Las seleccionadas tras el proceso de clasificación y selección en el capítulo 3 del presente trabajo de grado son:

#	**Práctica**
PC1	Implementación de IoT con arquitecturas y metodologías personalizables para cada dominio
PC2	Visualizar la solución IoT desde el elemento más simple hasta el todo: Atributos de calidad para sensores, Actuadores, nube y el sistema IoT en general
PC3	Utilizar el método planteado por kazman, clement y bass en software architecture in practice con el fin de diseñar y seleccionar mecanismos que se aplican para obtener la calidad final del sistema IoT.
PC4	Definir mediante la participación de hardware y software, modelo de colaboración de dispositivos y dominio de aplicación así mismo cuales atributos serán protagonistas
PC5	Realizar el mapeo de las características de la norma ISO/IEC 25010 junto con las características especiales de la IoT con el objetivo de ver las relaciones existentes entre las mismas.
PC6	Ponderar las relaciones anteriormente descritas con el objetivo de darle más énfasis a unas que otras según el dominio.
PC7	Combinar las dos prácticas (PC5 y PC6) junto con las tácticas arquitecturales con el fin de incrementar la calidad y sus atributos. También, se seleccionan las tácticas arquitecturales con el fin de incrementar la calidad con foco en atributos de calidad relevantes. Ver el uso de anotaciones.
PC8	Conformar la IoT como una serie de bloques funcionales para facilitar diversos beneficios del sistema como detección, identificación, actuación, comunicación y gestión. Los bloques anteriormente sugeridos son: aplicaciones, mantenimiento, servicio, comunicación, seguridad y dispositivo.
PC9	Entender la IoT como el conjunto de Hardware, conectividad, Arquitectura, algoritmos y software

Tabla 10. Prácticas de diseño de la arquitectura seleccionadas

Una vez se caracterizaron las prácticas y los atributos de calidad asociados, se procedió a realizar la selección de los mismos. El siguiente paso es la organización de los resultados que arrojó la presente investigación mediante la descripción realizada a continuación.

La aplicación de prácticas y atributos de calidad es transversal al proceso y al producto en sí mismo por eso es necesario indagar acerca de lo que es una práctica como su definición, su estructura y descripción entre otras.

4.3. Definición de práctica de acuerdo a la definición propuesta por el Object Management Group (OMG)

Descripción

Una práctica es un enfoque repetible para hacer algo con un objetivo específico en mente. Una práctica describe cómo manejar un aspecto específico de un esfuerzo de ingeniería de software, incluidas las descripciones de todos los elementos relevantes necesarios para expresar la guía de trabajo deseada que se requiere para lograr el propósito de la práctica. Una práctica puede definirse como una composición de otras prácticas.

Atributos

Reglas de consistencia: String [1]	Reglas sobre la consistencia de una Práctica particular. El formato para escribir estas reglas está fuera del alcance de esta especificación. Se recomienda utilizar texto sin formato u OCL.
Objetivo : String [1]	El objetivo de esta Práctica, expresado en una frase concisa y aislada. El contenido de este atributo debe ser una declaración breve y explícita que describa el objetivo que persigue la práctica. Se pueden dar explicaciones adicionales en el atributo "descripción" heredado de "ElementGroup"
Medidas : String [0..*]	Lista de unidades estándar utilizadas para evaluar el desempeño de la práctica y el logro de los objetivos.
Entrada : String [0..*]	Características esperadas de los elementos necesarios para iniciar la ejecución de una práctica.
Resultado: String [0..*]	Características esperadas de los elementos requeridos como productos después de la ejecución de una práctica.

Tabla 11. Descripcion del significado de una práctica

Semántica

Una práctica aborda un aspecto específico del desarrollo o trabajo en equipo. Proporciona la guía para caracterizar el problema, la estrategia para resolver el problema, y las instrucciones para verificar que efectivamente se ha abordado el problema. También describe qué evidencia de respaldo, si alguna, se necesita y cómo hacer que la estrategia funcione en la vida real.

Una práctica proporciona una forma de trabajo sistemática y repetible centrada en el logro de un objetivo. Cuando la práctica está compuesta por actividades, los criterios de realización derivados de las mismas se utilizan para verificar si el resultado producido alcanza el objetivo de la práctica. Para evaluar el desempeño de la práctica y el logro de los objetivos, se le pueden asociar medidas seleccionadas. Las medidas se estiman y recogen durante la ejecución de la práctica.

Como era de esperar, hay varios tipos diferentes de prácticas para abordar todas las diferentes áreas de desarrollo y trabajo en equipo, incluyendo (pero no limitado a):

- **Prácticas de desarrollo: como prácticas para desarrollar componentes, diseñar interfaces de usuario, establecer una arquitectura, planificar y evaluar iteraciones o estimar el esfuerzo.**

- **Prácticas sociales: como prácticas de trabajo en equipo, colaboración o comunicación.**

- **Prácticas organizacionales, como prácticas sobre hitos, revisiones de puertas de enlace o controles financieros.**

Excepto ejemplos triviales, una práctica no captura todos los aspectos de cómo realizar un esfuerzo de desarrollo. En cambio, la práctica aborda solo un aspecto de la misma. Para lograr una descripción completa, se pueden componer prácticas. El resultado de componer dos prácticas es otra práctica que captura todos los aspectos de las compuestas. De esta manera, se pueden crear prácticas más completas y poderosas, y eventualmente terminar con una que describa cómo se debe realizar un esfuerzo, es decir, un método.

- La definición de una práctica puede basarse en elementos definidos en un kernel. estos elementos, como los alfas, pueden usarse (y extendido) al definir elementos específicos de la práctica, como productos de trabajo.

- Una práctica puede ser una composición de otras prácticas. Todos los elementos de las otras prácticas se fusionan y el resultado se convierte en una nueva práctica (ver 9.4 para la definición de composición).

- Una práctica es cerrada en el sentido de que los elementos de la práctica solo pueden referirse a elementos que también forman parte de la práctica o a los grupos de elementos a los que se refiere esta práctica.

4.4. Modelo de Perfil de Calidad Planteado

Es importante plantear como un producto software avanza a través de su proceso y de su evolución como producto lo cual se sintetiza en la siguiente imagen con el fin de generar un marco de trabajo que permita agrupar y organizar las prácticas identificadas con el objetivo de seleccionar las más importantes para su posterior descripción.

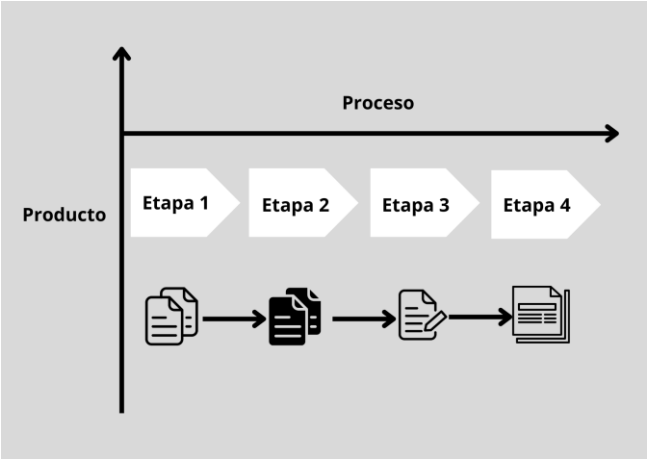


Figura 4. Producto VS Proceso de Software

La anterior imagen permitió el planteamiento de un nuevo marco de trabajo en el cual se identifican algunas etapas relevantes para un perfil de calidad en productos software basados en IoT, el cual tendrá como objetivo permitir la organización y agrupamiento de las prácticas. Las actividades correspondientes a cada etapa en el grafico son: (i) diseño (D), (ii) evaluación (E), (iii) refactorización (R), (iv) Perfil de calidad (QP), (v) Documentación y (vi) Catálogo (Ca). Para propósitos de la investigación se clasificarán las prácticas solo con (i) diseño (D), (ii) evaluación (E), (iii) refactorización (R) y (iv) Perfil de calidad (QP). Los tipos de práctica se determinan como: (i) excluyente (E), (ii) incluyente (I) y (iii) complementaria (C).

Versión Inicial del Modelo de perfil de calidad

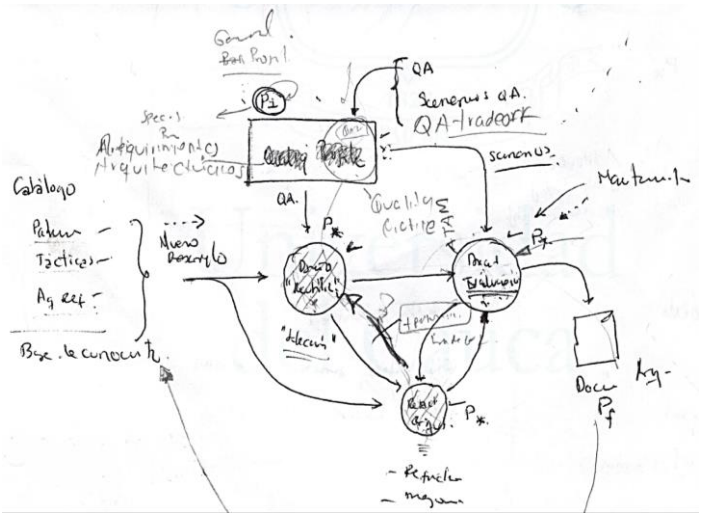


Figura 5. Modelo de perfil de calidad en su versión inicial

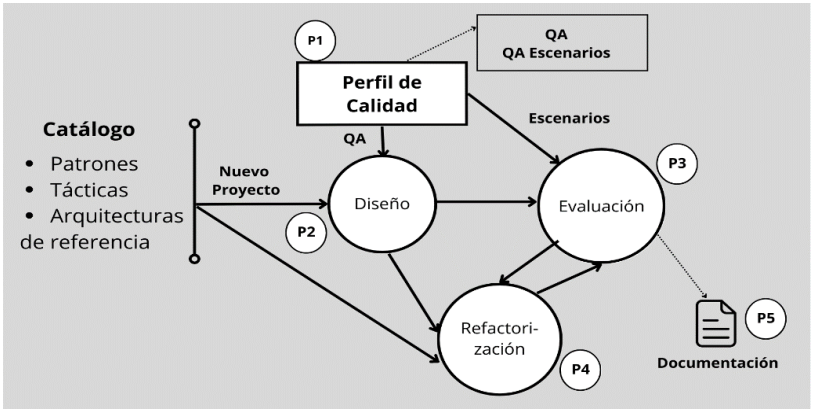


Figura 6. Modelo de perfil de calidad en su versión inicial

Primera iteración del modelo de perfil de calidad

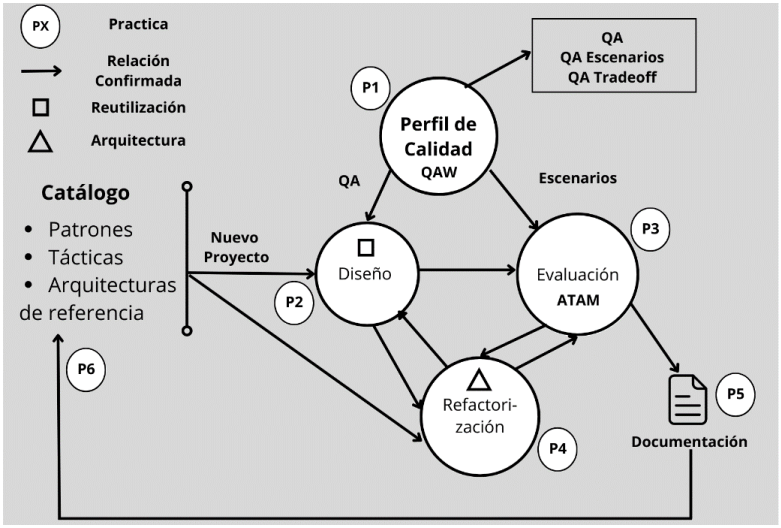


Figura 7. Iteración uno del modelo de perfil de calidad

Segunda iteración del modelo de perfil de calidad

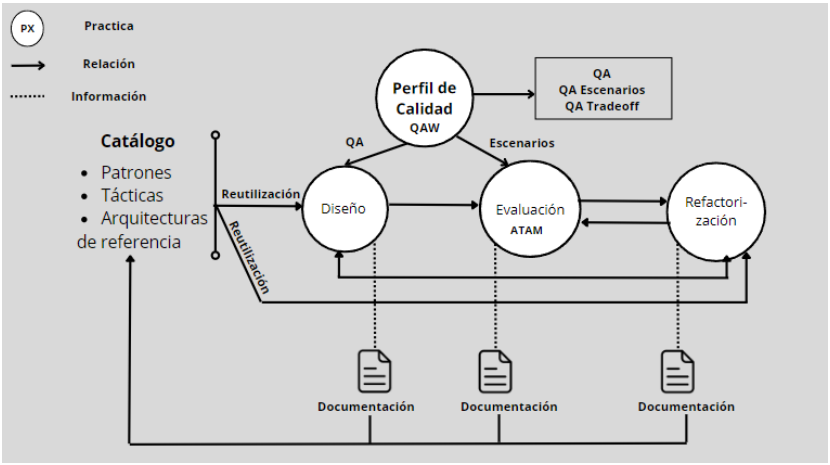


Figura 8. Iteración dos del modelo de perfil de calidad

Tercera iteración del modelo de perfil de calidad

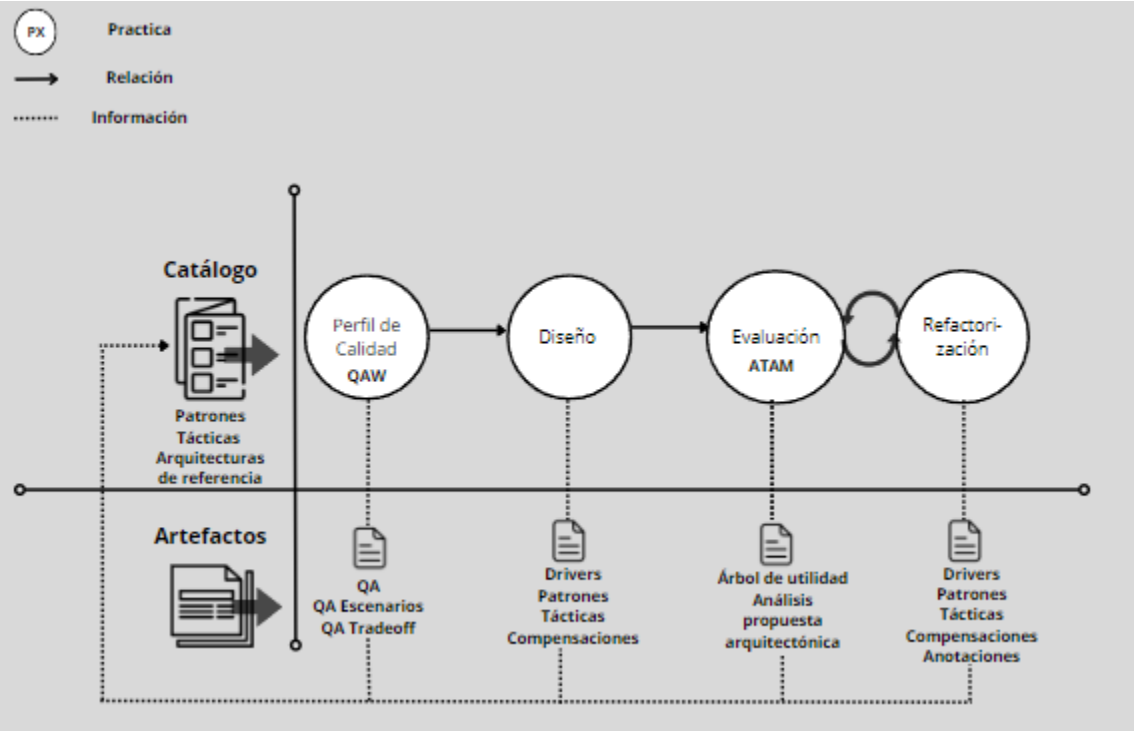


Figura 9. Iteración tres del modelo de perfil de calidad

4.5. Selección de prácticas de arquitectura de acuerdo al modelo de perfil de calidad

La literatura aporto un listado de prácticas de las cuales se filtraron algunas que se creen son relevantes para la presente investigación, teniendo como criterio la experiencia de los expertos y revisando una por una de acuerdo al perfil de calidad planteado por este trabajo en la sección anterior al igual que la clasificación de estas mismas en las actividades y tipos de práctica del perfil mencionado. El resultado final se presenta a continuación:

#	**Práctica**	**Actividad/Tipo**
PC1	Implementación de IoT con arquitecturas y metodologías personalizables para cada dominio	DI con PC2
PC2	Visualizar la solución IoT desde el elemento más simple hasta el todo: Atributos de calidad para sensores, Actuadores, nube y el sistema IoT en general.	DI con PC1
PC3	Utilizar el método planteado por kazman, clement y bass en software architecture in practice con el fin de diseñar y seleccionar mecanismos que se aplican para obtener la calidad final del sistema IoT.	DC con todas las de diseño
PC4	Definir mediante la participación de hardware y software, modelo de colaboración de dispositivos y dominio de aplicación así mismo cuales atributos serán protagonistas	DI con PC2
PC5	Realizar el mapeo de las características de la norma ISO/IEC 25010 junto con las características especiales de la IoT con el objetivo de ver las relaciones existentes entre las mismas.	QP

#	Práctica	Actividad/Tipo
PC6	Ponderar las relaciones anteriormente descritas con el objetivo de darle más énfasis a unas que otras según el dominio.	QP con PC59
PC7	Combinar las dos prácticas (PC5 y PC6) junto con las tácticas arquitecturales con el fin de incrementar la calidad y sus atributos. También, se seleccionan las tácticas arquitecturales con el fin de incrementar la calidad con foco en atributos de calidad relevantes. Ver el uso de anotaciones.	RC con PC5
PC8	Conformar la IoT como una serie de bloques funcionales para facilitar diversos beneficios del sistema como detección, identificación, actuación, comunicación y gestión. Los bloques anteriormente sugeridos son: aplicaciones, mantenimiento, servicio, comunicación, seguridad y dispositivo.	DI con PC1 y PC2
PC9	Entender la IoT como el conjunto de Hardware, conectividad, Arquitectura, algoritmos y software	DI con PC1 y PC2

Tabla 12. Prácticas de diseño de la arquitectura verificadas

A continuación, se realiza la clasificación de las prácticas de acuerdo a las etapas y actividades mencionadas anteriormente:

4.5.1. Selección de prácticas de arquitectura de acuerdo a las actividades del perfil de calidad (Relaciones)

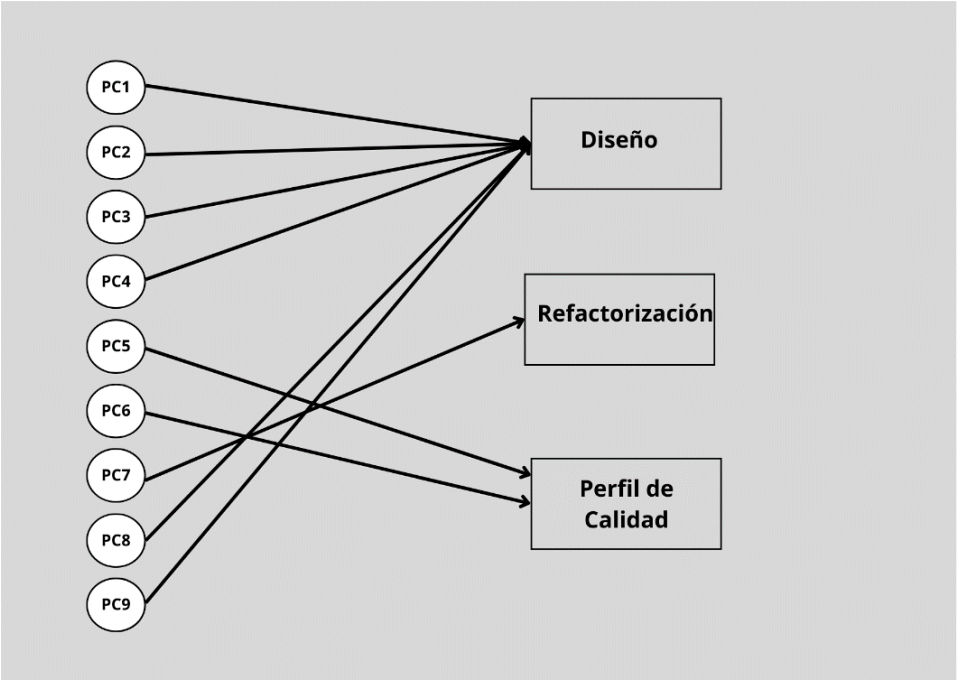


Figura 10. Relaciones entre las prácticas y etapas del perfil de calidad

4.5.2. Selección de prácticas de arquitectura de acuerdo al tipo de práctica del perfil de calidad

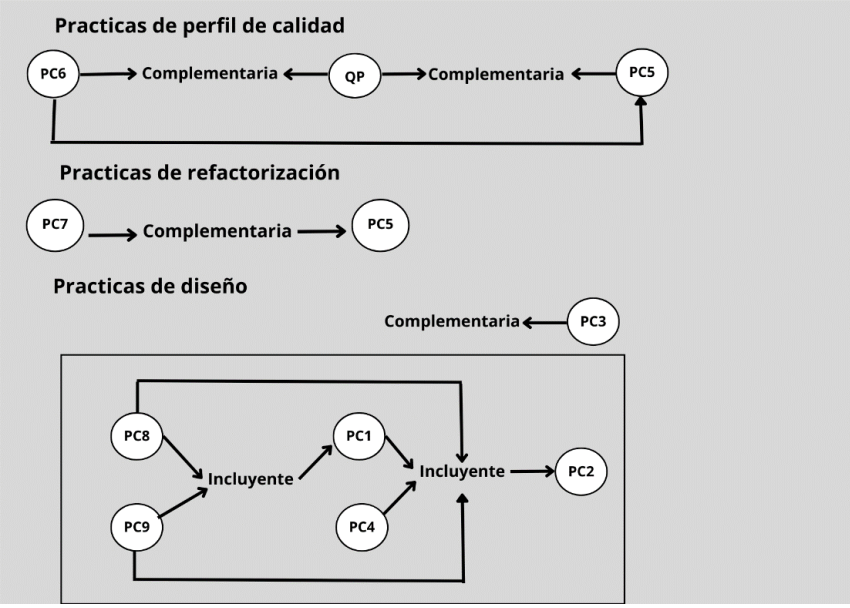


Figura 11. Relaciones entre las prácticas y actividades del perfil de calidad

4.6. Descripción de las prácticas de arquitectura seleccionadas de acuerdo a las etapas y actividades del perfil de calidad

Por lo tanto, se propone una plantilla en la cual se puedan describir las prácticas anteriormente planteadas con el fin de guiar a los desarrolladores de productos software basados en IoT incluyendo la calidad requerida a sus arquitecturas en los proyectos en los cuales se encuentran trabajando.

4.6.1. Práctica de diseño para alcanzar el perfil de calidad

Descripción de la práctica de diseño para la definición del perfil de calidad	
Descripción general	Visualizar la solución basada en IoT desde el elemento más simple hasta el todo con centro en las cualidades del sistema: - Relevancia de los atributos de Calidad para el sistema en general - Relevancia particular de atributos de calidad considerando aspectos específicos tales como tecnologías de sensores, actuadores, nube, el borde, entre otros.
Objetivo	Seleccionar los drivers, así como los patrones y las tácticas arquitecturales relacionados, con el fin de incrementar la calidad con foco en atributos de calidad relevantes.
Descripción específica	Asegurar que los módulos de código ejecutable, sensores, actuadores, nube y en general el producto software basado en IoT, alcancen los atributos de calidad requeridos para una adecuada implementación y compresión de la arquitectura con el fin de considerar los efectos que estos puedan

	ocasionar sobre las soluciones, las organizaciones y las compensaciones arquitectónicas en función de la calidad. Un buen diseño en la arquitectura de software junto con buenas prácticas pueden aportar en la evolución, modelado, desarrollo y ejecución de productos software complejos basados en IoT, fortaleciendo la calidad y funcionalidad.
Práctica asociada al modelo	P2 (Diseño)
Artefacto de entrada	Catálogo de patrones, Catálogo de Tácticas
Artefacto de salida	Arquitectura de propósito específico para IoT
Pasos	1. Identificar los atributos de calidad que serán drivers del diseño de la arquitectura de la solución basada en IoT integrando los diferentes involucrados. 2. Selección de los patrones y tácticas de arquitectura relevantes para el perfil de calidad establecido con base a los drivers identificados de acuerdo al dominio de aplicación escogido. 3. Para cada patrón (view) o táctica instanciarlos para establecer la solución considerando el dominio (servicios, redes, actuadores, sensores y dispositivos) que se asociaran para cumplir una tarea determinada. (PC1) (PC4). 4. Adicionar responsabilidades a las partes del sistema. Es decir, facilitar la identificación de las tareas que debe realizar cada de ellas, especificando comportamientos y facilitando su entendimiento. (PC8) (PC9) 5. Definir interfaces de los módulos, componentes y capas con el fin de mejorar en la abstracción de los componentes de software y hardware heterogéneos, definiendo la participación del software y hardware, estableciendo un modelo de participación, seleccionando cuales sensores, actuadores, tarjetas serán utilizados dependiendo de lo que se quiera medir. (PC4) 6. Se documenta lo obtenido en la selección de patrones y tácticas relevantes para el perfil de calidad establecido con respecto al diseño de la arquitectura, posteriormente, se envía dicha información al que tiene como objetivo albergar el conocimiento obtenido en estas prácticas. **(“En el modelo top-down se formula un resumen del sistema, sin especificar detalles. Cada parte nueva es entonces redefinida, cada vez con mayor detalle, hasta que la especificación completa es lo suficientemente detallada para validar el modelo”.)**
Recomendaciones	• Identificar correctamente las tareas de cada capa y bloque funcional. • Identificar correctamente el dominio de aplicación.

	• Identificar los atributos de calidad que debe tener presente a arquitectura según las limitaciones en su dominio. • Identificar los actores presentes en la arquitectura y su modelo de colaboración. • Documentar las vistas • Documentar las decisiones de diseño de la arquitectura
Herramientas de apoyo	Software Architecture in Practice (Len Bass Paul Clements and Rick Kazman)

Tabla 13. Práctica de diseño para alcanzar el perfil de calidad

4.6.2. Práctica de refactorización para alcanzar el perfil de calidad

Descripción de la práctica de refactorización para la definición del perfil de calidad	
Descripción general	Generar un nuevo diseño de la arquitectura siguiendo los mismos patrones o seleccionando unos nuevos dependiendo del impacto de las nuevas tácticas y de las fuerzas de los atributos de calidad.
Objetivo	Adecuar la arquitectura de acuerdo a los riesgos e inconformidades encontradas en la evaluación, tomando en cuenta las patrones, tácticas y compensaciones arquitectónicas en términos de atributos de calidad. (PC3) (PC5) (PC6) (PC7)
Descripción específica	Existe un problema específico con respecto a las arquitecturas del producto software basado en IoT y es que, en ocasiones, se deben tomar decisiones de compensación, es decir, si implementar el software de una manera que optimice un atributo de calidad en detrimento de otro. Por ello se cuenta con **ATAM**, clasificado como un método altamente estructurado para evaluar un diseño de arquitectura, permitiendo detectar, de manera temprana, riesgos técnicos, conflictos entre atributos, puntos clave del diseño y soluciones. Para mejorar estos atributos se utilizan las tácticas arquitecturales, que son medidas tomadas para mejorar estos mismos impactando de varias formas los patrones de arquitectura. En algunos casos, una táctica puede implementarse fácilmente usando las mismas estructuras (y un comportamiento compatible) como un patrón de arquitectura particular. Por otro lado, una táctica puede requerir cambios significativos en la estructura y comportamiento del patrón, o puede requerir estructuras y comportamiento completamente nuevos. Debido a la importancia de los atributos de calidad, es fundamental que se consideren durante el diseño inicial de la arquitectura del producto software basado en IoT.
Práctica asociada al modelo	P4 (Refactorización)
Artefacto de entrada	Limitaciones y riesgos de la arquitectura de propósito específico para IoT con las limitaciones que posee en función al dominio escogido
Artefacto de salida	Arquitectura de propósito específico para IoT refactorizada de acuerdo a su evaluación

Pasos	1. Evaluar las limitaciones, necesidades, preocupaciones y riesgos que poseen los drivers previamente seleccionados, por medio de ATAM, con el fin de detectar, de manera temprana, riesgos técnicos, conflictos entre atributos, puntos clave del diseño y soluciones, a partir del perfil de calidad establecido. 2. Se seleccionan nuevas tácticas para compensar las limitaciones, necesidades, preocupaciones y riesgos. A la luz de las nuevas tácticas se realiza el proceso de evaluación de la interacción entre los patrones y tácticas empleadas asociado al impacto en la arquitectura tratada. Esto se realiza usando las anotaciones de los diagramas de arquitectura con la información de la implementación de los patrones y tácticas abordadas. Esto debido a que implementar una táctica en una arquitectura puede impactar significativamente los patrones de arquitectura previamente seleccionados. 3. Se produce un nuevo diseño de la arquitectura siguiendo los mismos patrones o seleccionando unos nuevos dependiendo del impacto de las nuevas a tácticas y de las fuerzas de los atributos de calidad. 4. Se documenta lo obtenido del nuevo diseño con respecto a los patrones, tácticas y arquitecturas, se envía al catálogo de información que tiene como objetivo albergar el conocimiento obtenido en estas prácticas. **(“En el modelo bottom-up, las partes individuales se diseñan con detalle y luego se enlazan para formar componentes más grandes, que a su vez se enlazan hasta que se forma el sistema completo”.)**
Recomendaciones	• Se documentan las lecciones aprendidas recolectadas a través de los anteriores pasos. • Documentar la información obtenida en cada actividad. • Evaluar la arquitectura de software obtenida en la anterior práctica con el fin de producir un nuevo diseño de esta misma. • Identificar de manera temprana limitaciones, necesidades, preocupaciones y riesgos con el fin de generar nuevos diseños de arquitectura.
Herramientas de apoyo	• Software Architecture in Practice (Len Bass Paul Clements and Rick Kazman) • ATAM: Method for Architecture Evaluation (Rick Kazman, Mark Klein and Paul Clements)

	• Quality Attribute Workshops (QAWs) (Mario R. Barbacci, Robert Ellison, Anthony J. Lattanze, Judith A. Stafford, Charles B. Weinstock and William G. Wood) • How do architecture patterns and tactics interact? A model and annotation [28]. • Pattern-Driven Architectural Partitioning [55]. • Architecting Internet of Things Systems with Blockchain: A Catalog of Tactics [56]. • Software Architecture Quality Attributes of a Layered Sensor-Based IoT System [20]. • Implementing Reliability: the Interaction of Requirements, Tactics and Architecture Patterns [57]

Tabla 14. Práctica de refactorización para alcanzar el perfil de calidad

El ejemplo de la aplicación de las prácticas de diseño y refactorización para las arquitecturas basadas en IoT se encuentran en el **Anexo A.**

5. CAPITULO 5

Evaluar la comprensibilidad y utilidad percibida por desarrolladores de software de la región al utilizar las prácticas IoTAP en el desarrollo de un producto software basado en IoT mediante la aplicación de un estudio de caso en el dominio de la agricultura inteligente.

Descripción Capitulo

Para la presente investigación y en específico para la aplicación del estudio de caso, se toma como referente a Runeson y Höst [25], que proporcionan una introducción a la metodología de estudio de casos y pautas para los investigadores que realizan estudios de casos y los lectores que estudian los informes de dichos estudios. El contenido se basa en la propia experiencia de los autores al realizar y leer estudios de casos. El desarrollo de productos software, en el que también se encuentra incluida la IoT, se hace por parte de diferentes individuos, organizaciones, grupos, etc. que intervienen en esta actividad. La ingeniería de software es un área multidisciplinaria que involucra áreas donde normalmente se realizan estudios de casos, lo que quiere decir que muchas preguntas de investigación en ingeniería de software son adecuadas para la investigación de estudios de casos. Al realizar un estudio de caso, hay cinco pasos principales del proceso que se deben seguir: (i) Diseño del estudio de caso: se definen los objetivos y se planifica el estudio de caso, (ii) Preparación para la recolección de datos: se definen procedimientos y protocolos para la recolección de datos, (iii) Recopilación de datos: ejecución con recogida de datos sobre el caso estudiado, (iv) Análisis de los datos recopilados y (v) Reportes.

5.1. Estudio de caso

En esta sección se define el tipo de estudio de caso que se aplicara para la presente investigación, de acuerdo a los propósitos requeridos.

5.1.1. Definición del estudio de caso

El presente estudio de caso se define como **descriptivo** con el fin de retratar una situación o fenómeno determinado, en este caso, como se perciben los desarrolladores de software de la región la comprensibilidad y utilidad al utilizar las prácticas IoTAP. Se presenta en la Tabla 15 a continuación:

Ítem	**Descripción**
Pregunta de investigación	P1: ¿Cuál es el nivel de comprensibilidad de las prácticas IoTAP alcanzado por parte de ingenieros durante, después de su aplicación en un caso práctico? P2: ¿Cuál es el nivel de utilidad percibido por los ingenieros al utilizar las prácticas IoTAP?
Objetivo	Este estudio se enfoca en la evaluación de la comprensibilidad y utilidad percibida por desarrolladores de software de la región al utilizar las prácticas IoTAP en el desarrollo de un producto software basado en IoT.
Estrategia de selección de casos	Idoneidad y disponibilidad
Tipo de Caso según su objetivo	El presente estudio se define como un estudio de caso descriptivo.
Tipo de Caso según las Unidades de Análisis	Dado que sólo hay una unidad de análisis el estudio de caso se define como holístico.
Unidad de Análisis	Proyecto de diseño de la arquitectura y sus requisitos de calidad en el contexto de desarrollo de software basado en IoT para el sector agrícola.
Indicadores	**Percepción de la efectividad:** La efectividad de un método de alcance se refiere a la utilidad de los resultados del método en relación con las expectativas y necesidades de la empresa y, por lo tanto, la efectividad del método implica que las tareas que se realizan en el método son adecuadas para producir los resultados deseados. Para medir la efectividad de IoTAP se consideraron dos métricas cualitativas: facilidad de uso percibida y utilidad percibida. Para medir estas variables luego de aplicar las prácticas IoTAP, se adaptó y utilizó un instrumento de medición existente propuesto para la evaluación de los métodos de modelado de requisitos basados en percepciones y el Modelo de Aceptación de Tecnología (TAM).
Métricas	**Facilidad de uso percibida (PUE):** El grado en que una persona cree que usar las prácticas IoTAP no requerirá esfuerzo. Esta variable representa un juicio perceptivo del esfuerzo requerido para aprender y utilizar IoTAP (Davis, 1989). **Utilidad Percibida del Método (PUM):** El grado en que una persona cree que las prácticas IoTAP fueron claras, su nivel de detalle y su descripción facilitó su seguimiento, la realización de tareas y la adquisición de los artefactos requeridos. La utilidad percibida es una variable dependiente subjetiva, para medirla se utilizaron instrumentos sustentados en las propuestas: los métodos de modelado de Evaluación de requisitos basados en las

	percepciones de los usuarios (Abrahão, et al., 2011), el Modelo de Aceptación de Tecnología (TAM) (Davis, 1989) y The Method Evaluation Model (Moody, 2003) que se realizó después de aplicar el método.
Hipótesis	**Hipótesis alternativa 1:** Las prácticas IoTAP son percibidas útiles por un equipo de desarrollo de soluciones basadas en IoT para entender sus atributos de calidad y formular el diseño arquitectónico en un proyecto IoT en agricultura. **Hipótesis alternativa 2:** Las prácticas IoTAP resultan comprensibles a un equipo de desarrollo de soluciones basadas en IoT para entender sus atributos de calidad y formular el diseño arquitectónico en un proyecto IoT en agricultura. **Hipótesis nula 1:** Las prácticas IoTAP son percibidas como poco útiles por un equipo de desarrollo de soluciones basadas en IoT para entender sus atributos de calidad y formular el diseño arquitectónico. **Hipótesis nula 2:** Las prácticas IoTAP resultan incomprensibles a un equipo de desarrollo de soluciones basadas en IoT para entender sus atributos de calidad y formular el diseño arquitectónico.

Tabla 15. Práctica de refactorización para alcanzar el perfil de calidad

5.1.2. Diseño y planificación del estudio de caso

En esta fase, se diseñaron los procedimientos de investigación detallados para el estudio de caso y se hicieron los preparativos para la recolección de datos. Estos procedimientos y preparativos se presentan la Tabla 16,17 y 18 con el fin de realizar la ejecución de la presente propuesta.

Objetivo	**Indicadores**	**Métrica**	**Métodos para la recolección de datos**	**Instrumentos**
Este estudio se enfoca en la evaluación de la comprensibilidad y utilidad percibida por desarrolladores de software de la región al utilizar las prácticas IoTAP	Percepción de la efectividad	Facilidad de uso percibida (PUE)	Directa	Encuesta Protocolo de Observación
		Utilidad Percibida del Método (PUM)	Directa	Encuesta Protocolo de Observación

Tabla 16. Parámetros del estudio de caso de evaluación de las prácticas IoTAP

Instrumentos de recolección de datos

Los instrumentos utilizados se describen a continuación:

- **Encuesta acerca de la comprensibilidad y utilidad de las prácticas:** Encuesta estructurada que permite recolectar información de la percepción de los ingenieros acerca de la experiencia con el uso de las prácticas.

- **Protocolo de Observación:** El protocolo de observación, se realiza con el objetivo de registrar hechos durante el estudio, permite también realizar un registro escrito de diversas acciones, hechos, etc. y su organización se realiza de forma cronológica, con el fin de facilitar la revisión de los hechos anotados. Los científicos suelen desarrollar bitácoras durante sus investigaciones para explicar el proceso y compartir sus experiencias con otros expertos.

- **Protocolo de evaluación del artefacto:** El protocolo de evaluación del artefacto usa una rúbrica denominada como Software Architecture Document (SAD), que se utiliza con el fin de evaluar si los atributos de calidad se satisfacen con la arquitectura, proporciona también una descripción completa y general de la arquitectura del sistema, usando varias vistas arquitectónicas diferentes para representar los diferentes aspectos de un sistema. Su objetivo principal es transmitir y capturar las decisiones arquitectónicas significativas que se han hecho en un sistema.

- **Plantilla del Artefacto:** La evaluación de la presente propuesta de realizar con Architecture Trade-Off Analysis Method (ATAM) [37], un método para evaluar Arquitecturas de Software que permite constatar el grado de cumplimiento de una arquitectura con los atributos de calidad de un sistema. Sus principales artefactos son: (**i) el árbol de utilidad y (ii) análisis de una propuesta arquitectónica.**

Los anteriores instrumentos se podrán observar en el **Anexo B**.

Plan de recolección de datos

Evaluación con expertos

Sesión	Actividad	Duración
1	Charla inicial acerca del proyecto de investigación y su motivación	10 min
1	Explicación del ejemplo de aplicación de las prácticas IoTAP junto con su metodología	35 min
1	Evaluación por parte de los expertos mediante la encuesta definida acerca de la comprensibilidad y utilidad de las prácticas	10 min
1	Conclusiones, sugerencias e inquietudes acerca de la aplicación de las prácticas IoTAP en el ejemplo propuesto (Focus Group)	10 min

Tabla 17. Plan de recolección de datos con expertos

Estudio de caso aplicado

Sesión	Actividad	Duración
1	Entrenamiento en las prácticas y artefactos (SAD) de IoTAP y ATAM	40 min
2	Perfil de Calidad y diseño Preliminar de la arquitectura (workshop)	2 hora
3	Evaluación de la arquitectura mediante Lighweight ATAM de manera guiada (Workshop)	1 hora
4	Refactorización (Con anotaciones)	2 hora
5	Evaluación de la arquitectura mediante Lighweight ATAM de manera guiada (Workshop)	1 hora
6	Refactorización (Con anotaciones)	1 hora
7	Reunión donde se discute acerca de la comprensibilidad y utilidad de las prácticas IoTAP (Cuestionario de preguntas abiertas)	10 min
8	Conclusiones acerca de las actividades abordadas (Focus Group)	40 mi

Tabla 18. Plan de recolección de datos para el estudio de caso

Recursos de apoyo

Recursos	Propósito
Cuestionario Google Forms	Obtener el grado de la percepción de la facilidad de uso y utilidad tras la aplicación de las prácticas IoTAP
Sitio web en Google Sites	Facilitar la aplicación de los artefactos que se encuentra en el presente estudio de caso

Tabla 19. Recursos de apoyo

Expertos participantes

Experto	Descripción
Andrés Hurtado	Es ingeniero electrónico de la Universidad del valle (Sede Cali), profesor de la Universidad San Buenaventura (Sede Cali), se especializa en integrar tecnologías de la industria 4.0 ayudando a generar habilidades profesionales de la cuarta revolución industrial. Su fuerte es la IIoT o más bien conocida como IoT en ambientes industriales.
Flor Hernández	Es magister y estudiante de Doctorado de la Universidad del Cauca, profesora de la Universidad Nacional Abierta y a distancia (UNAD) y se especializa en temas como la arquitectura de software y la experiencia de usuario.
Leandro Antonelli	Es doctor, magister y profesor de la Universidad Nacional de La Plata y se especializa en el área de ingeniería de requerimientos, específicamente en gestión de requerimientos como ágiles y tradicionales.
Giovanni Zambrano	Es estudiante de Maestría en computación de la Universidad del cauca y se especializa en arquitecturas de software basadas en IoT adaptativas y en robótica en el ámbito de la agricultura 4.0 o agricultura de precisión.
Wilson Pantoja	Es estudiante de doctorado, magister, especialista y profesor de la Universidad del Cauca, se especializa en temas como arquitecturas de software y en la mejora de procesos de software.

Tabla 20. Expertos en arquitectura de software

5.1.3. Ejecución del estudio de caso

Evaluación con expertos

Sesión 1: Se realiza una sola sesión en la cual se abordan una serie de actividades con el fin de realizar la recolección de datos a los expertos participantes, las actividades se explican a continuación:

- Se realiza una charla introductoria al proyecto de investigación denominado como ***"IoTAP: Prácticas de diseño de la arquitectura y sus requisitos de calidad en el contexto de desarrollo de software basado en IoT"*** con el fin de explicar el fin y motivación del mismo.
- Se realiza la explicación de las prácticas IoTAP junto con su metodología a los expertos participantes con el objetivo de que conozcan cómo se aplican las mismas siguiendo la metodología planteada.

- Se realiza la evaluación de la aplicación de las prácticas IoTAP junto con su metodología por medio de la encuesta denominada como "***encuesta definida acerca de la comprensibilidad y utilidad de las prácticas"*** con el fin de conocer el resultado de las métricas planteadas anteriormente.

- Se realizan las conclusiones, sugerencias e inquietudes de las prácticas y la metodología planteadas por parte de los expertos con el fin de documentarlas en hallazgos para su posterior aplicación.

Estudio de caso aplicado

En el estudio de caso participan los expertos Giovanni Zambrano en las actividades de diseño y refactorización de arquitectura, Flor Hernández en la actividad de evaluación de la arquitectura.

- **Sesión 1:** El objetivo de la primera sesión es socializar, contextualizar y entrenar a los ingenieros en las prácticas IoTAP, artefactos como el Software Architecture Document (SAD) y los derivados de ATAM (el árbol de utilidad y análisis de una propuesta arquitectónica). Se realiza una presentación oral

con el fin de informar a los ingenieros participantes acerca de las actividades que se realizaran y a su vez orientar en el manejo de los artefactos necesarios para el presente estudio de caso.

- **Sesión 2:** La segunda sesión se realiza a modo de workshop, en el cual, bajo la supervisión del moderador, los ingenieros participantes realizan una aproximación hacia el modelo de perfil de calidad planteado y se realiza un diseño preliminar de la arquitectura promoviendo la participación activa de los participantes y se seleccionan los drivers, así como los patrones y las tácticas arquitecturales relacionados, con el fin de incrementar la calidad con foco en atributos de calidad relevantes.

- **Sesión 3:** En la tercera sesión se realiza la actividad de evaluación de la arquitectura preliminar mediante la adaptación de ATAM en un Lighweight ATAM, que permitirá evaluar el diseño de la arquitectura preliminar, detectando, de manera temprana, riesgos técnicos, conflictos entre atributos, puntos clave del diseño y soluciones.

- **Sesión 4**: En la cuarta sesión, se realiza la actividad de refactorización de la arquitectura, generando un nuevo diseño de la arquitectura siguiendo los mismos patrones o seleccionando unos nuevos dependiendo del impacto de las nuevas tácticas y de las fuerzas de los atributos de calidad y usando las anotaciones de los diagramas de arquitectura con la información de la implementación.

  En caso de ser requerido por los participantes del estudio de caso, se pueden repetir las sesiones 3 y 4 lo que en la lista de sesiones serian correspondientemente la 5 y 6.

- **Sesión 7:** Se realiza una reunión donde se discute acerca de la comprensibilidad y utilidad de las prácticas IoTAP.

- **Sesión 8:** Se realizan las conclusiones de la aplicación del estudio de caso a modo de Focus Group, que consiste en un debate guiado sobre un determinado tema, y que contribuye a discusiones abiertas para la presente investigación.

5.1.4. Análisis y resultados

Evaluación con expertos

Antes de realizar la ejecución del estudio de caso, se decidió hacer **una** sesión con expertos en arquitectura de software con el objetivo de evaluar las prácticas que se encuentran contenidas en la metodología diseñada mediante el uso del perfil de calidad planteado en el capítulo anterior. Se explican los pasos para ejecutar el perfil de calidad, la metodología y las prácticas asociadas, esto se puede ver en el Anexo A en la segunda versión del ejemplo de aplicación de las prácticas de diseño y refactorización para arquitecturas basadas en IoT. Después de la explicación de la metodología, se realiza la **encuesta acerca de la comprensibilidad y utilidad de las prácticas IoTAP** con el objetivo de evaluar las mismas y obtener resultados asociados, los resultados completos se pueden encontrar en el **Anexo C.** Las preguntas calificadas **de 1 a 5**, la respuesta "Totalmente en desacuerdo" corresponde a menor calificación, es decir, **uno** y la respuesta "Totalmente de acuerdo" corresponde la mayor calificación, es decir, **cinco**. Los resultados relevantes son enunciados a continuación:

Andrés Hurtado	
Promedio Total	4,2
Promedio Facilidad de uso	4,7
Promedio Utilidad	3,8

Tabla 21. Analisis estadistico respuestas experto 1

Flor Hernández	
Promedio Total	3,7
Promedio Facilidad de uso	4,1
Promedio Utilidad	3,4

Tabla 22. Analisis estadistico respuestas experto 2

Leandro Antonelli	
Promedio Total	3,9
Promedio Facilidad	3,8
Promedio Utilidad	4

Tabla 23. Analisis estadistico respuestas experto 3

Giovanni Zambrano	
Promedio Total	4,1
Promedio Facilidad de uso	4,3
Promedio Utilidad	4

Tabla 24. Analisis estadistico respuestas experto 4

Wilson Pantoja	
Promedio Total	4,5
Promedio Facilidad de uso	4,7
Promedio Utilidad	4,4

Tabla 25. Analisis estadistico respuestas experto 5

Análisis estadístico de variables			
Promedio Total Facilidad de uso	4,3	Desviación Facilidad de uso	0,38
Promedio Total Utilidad	3,9	Desviación Utilidad	0,36
Promedio Total participantes	4,1	Desviación promedio total	0,30

Tabla 26. Analisis estadistico general

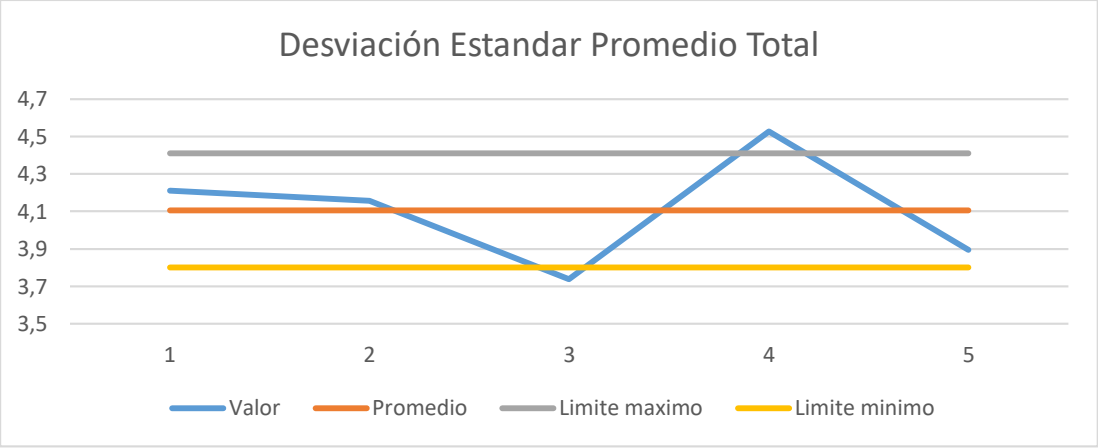


Figura 12. Desviación estandar promedio total

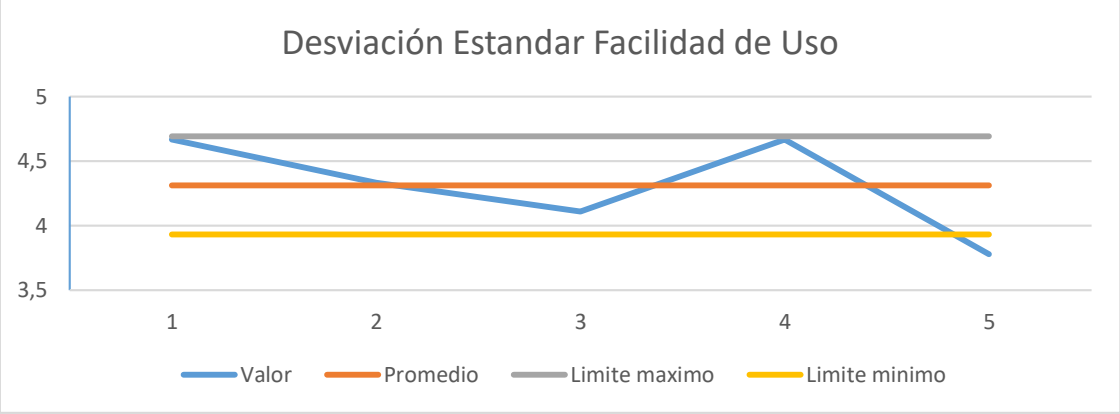


Figura 13. Desviación estandar facilidad de uso

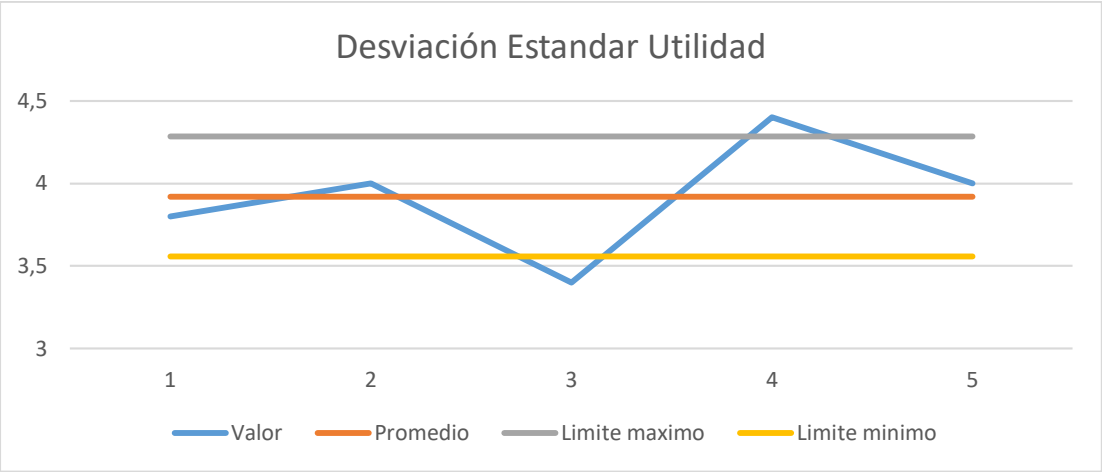


Figura 14. Desviación estandar utilidad

Se puede observar que los expertos perciben las prácticas IoTAP como fáciles de usar y también útiles para entender los atributos de calidad y formular el diseño arquitectónico en un proyecto IoT en agricultura. Estos puntajes son buenos, pero también se obtuvieron sugerencias por parte de los expertos para mejorar la metodología, el perfil de calidad y las prácticas, esto se menciona en los **hallazgos con respecto a la evaluación de los expertos**.

Hallazgos con respecto a la evaluación de los expertos

- Los expertos Andrés Hurtado y Giovanni Zambrano coinciden que sería importante trabajar en una aplicación tipo Dashboard que permita cubrir la metodología propuesta.
- El experto Andrés Hurtado sugiere realizar un diagrama a nivel de proceso en el cual se detallen los pasos, etapas y prácticas empleadas en la metodología.
- El experto Wilson Pantoja sugiere cambiar el nombre refactorización por refactorización de arquitectura en la metodología.
- El experto Wilson Pantoja sugiere cambiar el nombre de evaluación de arquitectura con ATAM por solamente evaluación de la arquitectura en la metodología.
- El experto Wilson Pantoja sugiere tener en cuenta el perfil del cultivo que contiene el lugar y el tipo de plantación que se tiene para poder medir los datos y variables correctas teniendo en cuenta el universo del discurso y las necesidades del cliente, esto hace parte de una de las contribuciones de la presente investigación en la cual se utiliza la sensibilidad al entorno para poder recuperar la información circundante en el contexto y actuar con base en ella.
- El experto Wilson Pantoja sugiere seguir trabajando en la propuesta de las prácticas debido a que en el Cauca hay una baja productividad en el agro por innovación y esto es una necesidad latente, la cual debe ser mejorada.
- Llevar las prácticas de arquitectura a los desarrolladores de Soluciones IoT de tal forma que las apliquen de una forma más ágil, y que no se genere deuda técnica arquitectónica.
- El contexto de la arquitectura de precisión es muy relevante para determinar aspectos funcionales y del perfil de calidad.

- Los expertos Giovanni Zambrano y Leandro Antonelli concluyen que el proceso de diseño y el modelo empleado son intuitivos y secuenciales y esto permite, tomarlo para el desarrollo de arquitecturas.
- Los expertos Flor Hernández y Wilson Pantoja concluyen que la propuesta esta personalizada para aplicar la arquitectura al dominio de la IoT, ahorrando tiempo al arquitecto para conocer un dominio tan específico.

Estudio de caso aplicado

Tras la aplicación del estudio de caso, se obtiene la información relevante de acuerdo a cada etapa planteada por el perfil de calidad planteado en el que se tienen en cuenta actividades como diseño, evaluación y refactorización de arquitectura. Se emplea la siguiente metodología diseñada para el presente estudio:

Ciclo de la Metodología

Figura 15. Ciclo de la metodología

Catálogo

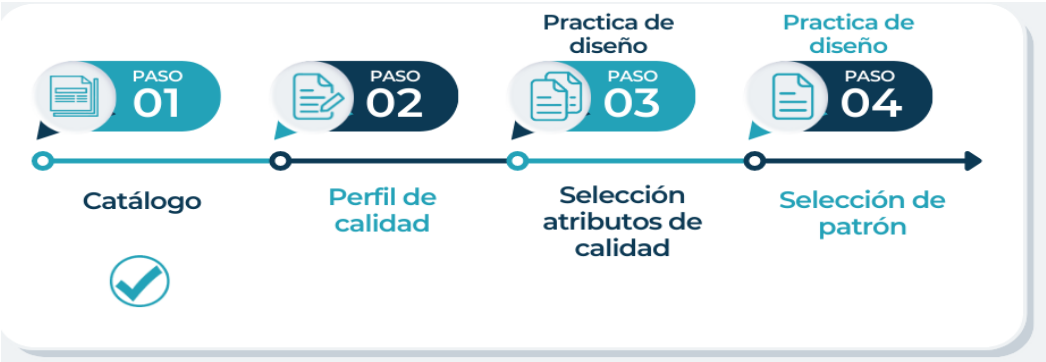


Figura 16. Paso 1 de la metodología correspondiente al catálogo

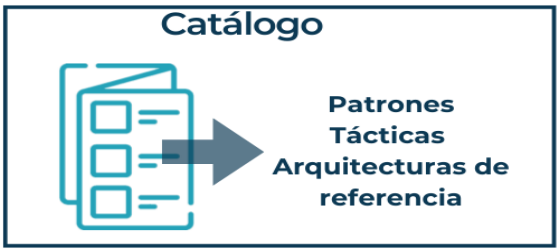


Figura 17. Catálogo de atributos de calidad, tacticas, patrones y arquitecturas de referencia

Se utilizan dos catálogos uno de atributos y otro de patrones para identificar cuáles son los más recurrentes en el dominio de la agricultura inteligente. A partir de la identificación se seleccionan los atributos y patrones relevantes para aplicaciones basadas en IoT, en este caso específico para robots semi-autónomos en agricultura. El resultado se podrá ver a continuación:

Atributos	***Patrones***
Eficiencia de desempeño	Capas
Compatibilidad	Basado en la nube
Usabilidad	Orientado a servicios
Confiabilidad	Micro-servicios
Seguridad	Restful
Mantenibilidad	Publicador/Suscriptor
Sensibilidad al entorno	Centrado en la información

Tabla 27. Catálogo de atributos de calidad y patrones de arquitectura

Perfil de calidad

Figura 18. Paso 2 de la metodología correspondiente al perfil de calidad

Se realiza una breve descripción del marco metodológico que engloba todo el proceso que tienen las actividades, incluidas las prácticas de diseño y refactorización con sus respectivos pasos. Este marco sirve de guía general de toda la metodología propuesta.

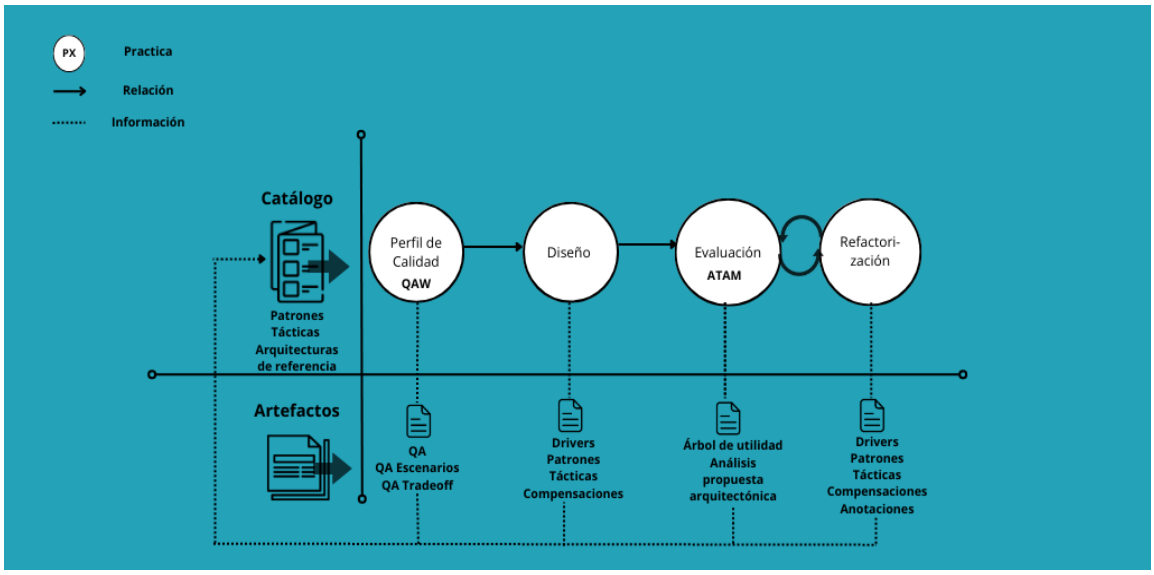


Figura 19. Perfil de calidad diseñado y empleado en la metodología

Selección de Atributos de Calidad

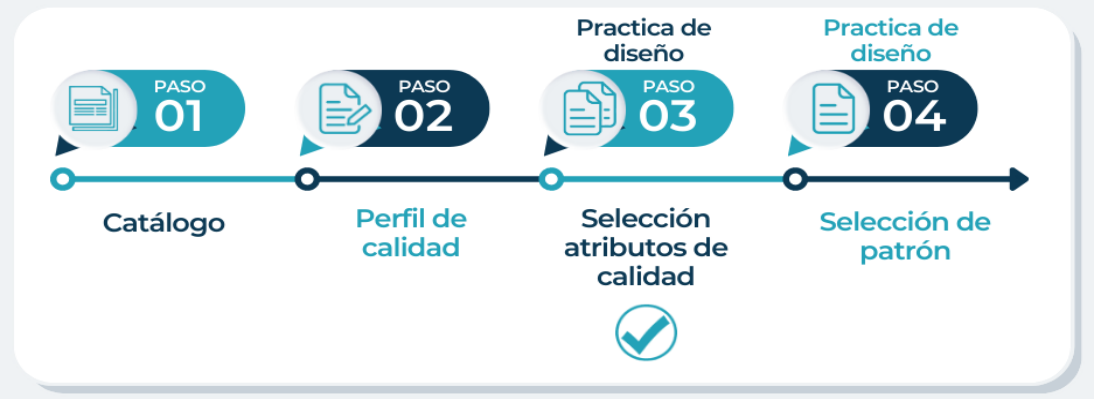


Figura 20. Paso 3 de la metodología correspondiente a la selección de atributos de calidad

El dominio a ser utilizado, es la agricultura inteligente, para el cual los atributos de calidad de la tabla son los más relevantes o sensibles para la arquitectura de software basados en IoT, puestos de manifiesto en las investigaciones consultadas y son los seleccionados para el desarrollo de la arquitectura a proponer. Los trabajos que componen el catálogo son los siguientes: [6], [17], [18], [58]–[61].

Atributos
Eficiencia de desempeño
Compatibilidad

Usabilidad
Confiabilidad
Seguridad
Mantenibilidad
Sensibilidad al entorno

Tabla 28. Atributos de calidad seleccionados del catálogo

Selección del patrón de arquitectura

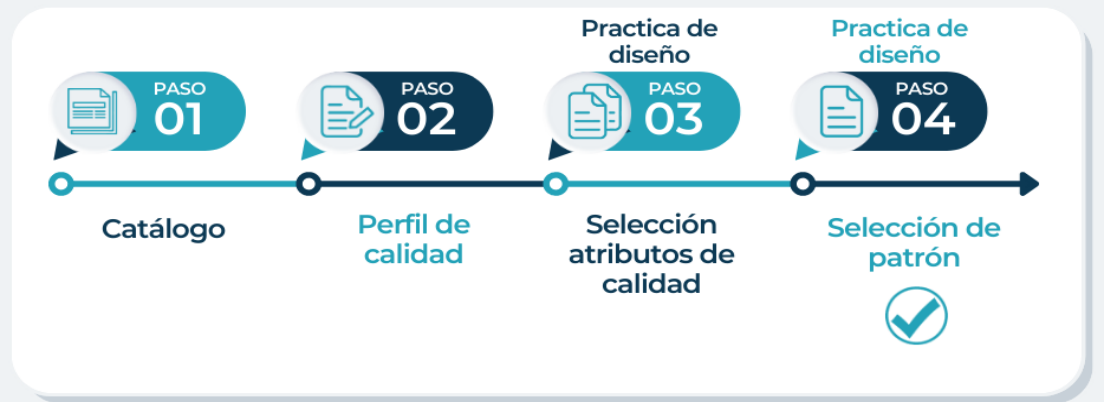


Figura 21. Paso 4 de la metodología correspondiente a la selección del patrón

El dominio a ser utilizado es la agricultura inteligente para el cual los patrones de arquitectura de la tabla son los más relevantes o sensibles para la arquitectura de software basados en IoT, puestos de manifiesto en las investigaciones consultadas y son los patrones seleccionados para el desarrollo de la arquitectura a proponer. Los trabajos que componen el catálogo son los siguientes: [6], [17], [18], [58]–[61].

Patrones
Capas
Basado en la nube
Orientado a servicios
Microservicios
Restful
Publicador/Suscriptor
Centrado en la información

Tabla 29. Patrones de arquitectura seleccionados del catálogo

Se va a contrastar la influencia de los atributos de calidad en los patrones de arquitectura seleccionados por medio de la calificación de los mismos. Se tiene en cuenta la escala de calificación que se muestra a continuación:

Aplicación practica de diseño IoTAP Dominio Agricultura							
Patron/Atributo	Eficiencia de desempeño	Compatibilidad	Usabilidad	Confiabilidad	Seguridad	Mantenibilidad	Sensibilidad al entorno
Capas	**	**	/	**	**	**	**
Basado en la nube	**	*	/	**	**	*	**
Orientado a servicios	**	**	/	**	*	**	**
Microservicios	**	**	/	*	*	*	**
Restful	/	*	/	**	*	**	*
Publicador/Suscriptor	**	**	*	*	**	**	**
Centrado en la informacion	*	*	/	**	*	*	**

Figura 22. Clasificación realizada por parte del arquitecto de software entre los atributos de calidad y los patrones de arquitectura

Aplicación practica de diseño IoTAP Dominio Agricultura								
Patron/Atributo	Eficiencia de desempeño	Compatibilidad	Usabilidad	Confiabilidad	Seguridad	Mantenibilidad	Sensibilidad al entorno	
Capas	2	2	0	2	2	2	2	12
Basado en la nube	2	1	0	2	2	1	2	10
Orientado a servicios	2	2	0	2	1	2	2	11
Microservicios	2	2	0	1	1	1	2	9
Restful	0	1	0	2	1	2	1	7
Publicador/Suscriptor	2	2	1	1	2	2	2	12
Centrado en la informacion	1	1	0	2	1	1	2	8

Figura 23. Resultados de la clasificación realizada por parte del arquitecto de software entre los atributos de calidad y los patrones de arquitectura

Se realiza la comparativa grafica de los siete patrones con el fin de determinar cuál es el más indicado para la investigación que se está apoyando:

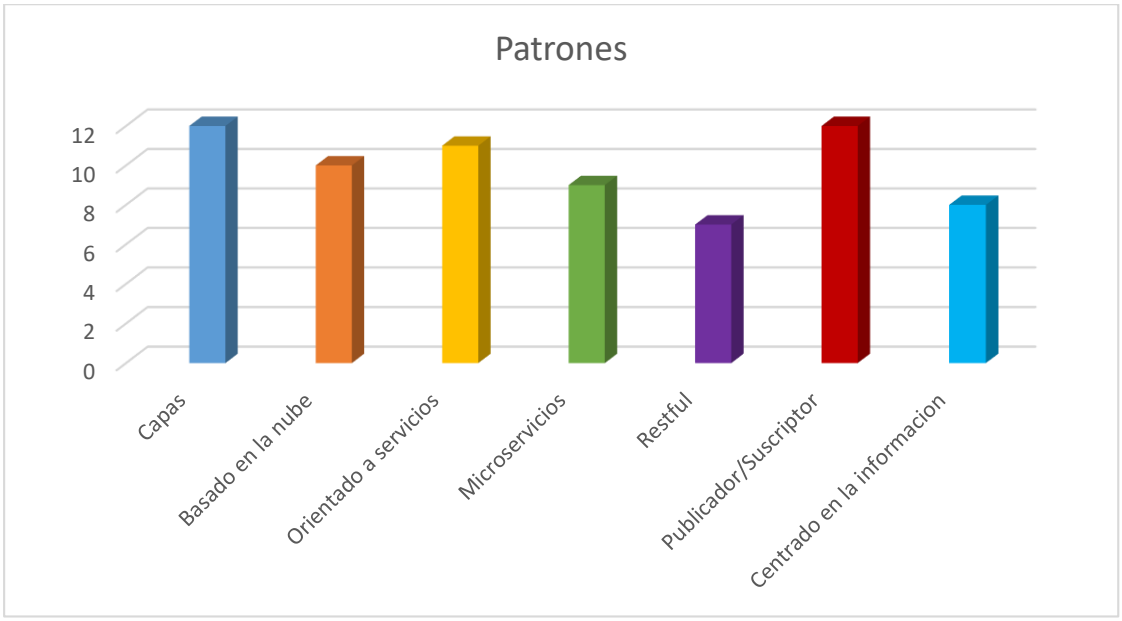


Figura 24. Resultados globales de la clasificación realizada por parte del arquitecto de software entre los atributos de calidad y los patrones de arquitectura

Según los hallazgos de calificar los atributos vs. los patrones nos arroja tres candidatos de arquitecturas de referencia más indicados para el desarrollo de la arquitectura como lo son: ***(i) capas, (ii) orientado a servicios y (iii) publicador/suscriptor***. El patrón seleccionado es el de **capas** y el porqué de la selección ***(Metas)*** y sus respectivos ***Drivers*** se detallan a continuación:

Drivers

- ***Eficiencia de desempeño:*** Esta arquitectura divide los elementos que la componen en niveles específicos y esto es relevante en el diseño de la arquitectura, capas como los sensores determinar una capa común que sería de fácil acceso por el usuario y los robots del sistema.

- ***Sensibilidad al entorno:*** Es muy relevante que el robot tenga la capacidad de adquirir información en tiempo real y poder percibir las pequeñas variaciones, que evaluadas con un histórico permita la toma de decisiones y la adaptabilidad en el desarrollo de la automatización de las tareas.

Otros Atributos de calidad relevantes

- ***Compatibilidad:*** Las capas delimitan los elementos o responsabilidades, esto permitiría al sistema adicionar más robots, teniendo en cuenta las capas delimitadas y permitiendo su interacción al hablar en el mismo lenguaje.

- ***Usabilidad:*** Esta está limitada por la capacidad de cómputo del sistema por lo cual se limita al hardware diseñado de los robots a ser implementados.

- ***Confiabilidad:*** las capas están organizadas por niveles de responsabilidad y aplicabilidad esto permite que podamos verificar el comportamiento del sistema y determinar los posibles errores y falencias del mismo en tiempo de ejecución y así poder generar respuestas a estos.

- ***Seguridad:*** es importante controlar el sistema para que no sea intervenido por agentes externos, al manipular información que tiene cierto grado de confidencialidad y que es pertinente para la automatización de los procesos, una modificación, podría hacer que el sistema respondiera erróneamente y causando un daño.

- ***Mantenibilidad:*** El sistema estaría organizado por niveles, estos subdividen su operación y delimitan el accionar, permitiendo evaluar el comportamiento de cada uno de ellos y determinar su funcionamiento para así evaluar un plan de mantenimiento para cada uno de ellos.

Las ***restricciones*** que se encuentran al utilizar este patrón se deben tener en cuenta ya que pueden afectar en el diseño de arquitectura, por ello [62][63], sugiere que deben tenerse en cuenta estas mismas en los diseños de arquitectura:

- ***Eficiencia de desempeño***: La comunicación por la red o internet es una de las tareas más tardadas de un sistema, incluso, en muchas ocasiones más tardado que el mismo procesamiento de los datos, por lo que el hecho de tener que comunicarnos de capa en capa genera un degrado significativo en el performance.

- ***Escalabilidad:*** Las aplicaciones que implementan este patrón por lo general tienda a ser monolíticas, lo que hace que sean difíciles de escalar, aunque es posible replicar la aplicación completa en varios nodos, lo que provoca un escalado monolítico.

- ***Complejidad de despliegue***: En este tipo de arquitecturas es necesarios desplegar los componentes de abajo arriba, lo que crea una dependencia en el despliegue, además, si la aplicación tiende a lo monolítico, un pequeño cambio puede requerir el despliegue completo de la aplicación.

- **Anclado a un Stack tecnológico**: Si bien no es la regla, la realidad es que por lo general todas las capas de la aplicación son construidas con la misma tecnología, lo que hace amarremos la comunicación con tecnologías propietarias de la plataforma utilizada.

- ***Tolerancia a los fallos***: Si una capa falla, todas las capas superiores comienzan a fallar en cascada.

- ***Modificabilidad/ Mantenibilidad:*** A veces no se logra la contención de un cambio y se requiere una cascada de cambios en varias capas.

En el contexto de diseño para arquitecturas de software basada en IoT***, con el fin de argumentar*** por que se escoge el patrón de capas y no se escogen los patrones resultantes como el patrón orientado a servicios y Publicador/Subscriptor***. Se decide utilizar el patrón de capas debido a su alta abstracción, encapsulamiento, capas de funcionalidad muy bien definidas, alta cohesión y reusabilidad. Se descartan*** los patrones como, el orientado a servicios ya que no es óptimo para la presente arquitectura debido al factor seguridad, no es muy relevante y en el sistema propuesto es necesario para velar por la seguridad de la información. Por otra parte, tampoco se escoge el patrón Publicador/Subscriptor ya que requiere una interfaz más elaborada, esto requeriría un aumento del hardware de los dispositivos o robots, así mismo se requiere que un sistema híbrido que permita la interacción entre los robots su autorganización y variación del sistema. **Para lograr** más eficiencia de desempeño y sensibilidad al entorno con respecto al correcto funcionamiento (en tiempo real) y a la toma de decisiones con base en la información recolectada del entorno**, se acepta que esto afectara** a la arquitectura en términos de componentes ***debido*** a que se debe garantizar que el tránsito de información no sea continuo**,** sino que cuando se perciba información relevante se actué con base en ella sin saturar el sistema.

Para el diseño de la arquitectura de software del este proyecto se toma como referencia la arquitectura por capas de 6 niveles compuesto de las siguientes capas, tomado del catálogo de arquitecturas de referencia:

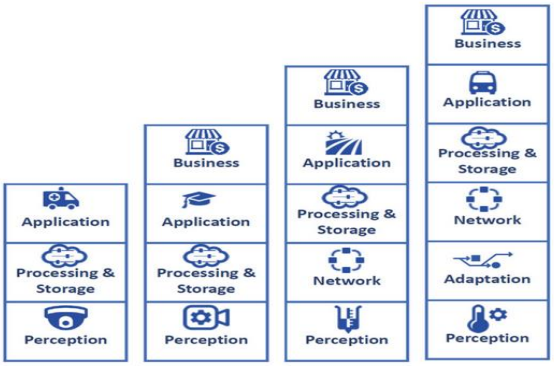


Fig. 3. Layered IoT architecture

Figura 25. Arquitecturas de referencia pertenecientes al catálogo

- **Capa de negocio:** Es la capa usada por los usuarios en la cual se llaman a las funcionalidades desde la capa de aplicación al usuario final.

- **Capa de aplicación:** Es la capa responsable de proporcionar los servicios inteligentes de IoT a los usuarios y se enfatiza en las necesidades a nivel de atributos de calidad que se requieren para un producto software basado en IoT.

- **Capa de servicio:** Se encarga de analizar y procesar los datos que provienen de la capa de conectividad. También es llamada capa de middleware debido a que contribuye con el intercambio de información entre objetos heterogéneos que no cuentan con requisitos específicos de software y hardware.

- **Capa de conectividad:** Es la capa responsable de procesar y transmitir los datos que se reciben de la capa de percepción. Cuenta con diferentes tipos de redes como WiFi, móvil, etc.

- **Capa de adaptación:** es la capa que tiene la responsabilidad de cambiar el comportamiento del sistema con base en los datos obtenidos a partir de la capa de percepción.

- **Capa de percepción:** Esta capa se encarga de percibir, detectar, recolectar y medir información del contexto circundante, por ejemplo: temperatura, humedad, etc.

Se realiza una vista inicial del diseño de la arquitectura incorporando las capas mencionadas anteriormente y podrá observarse a continuación:

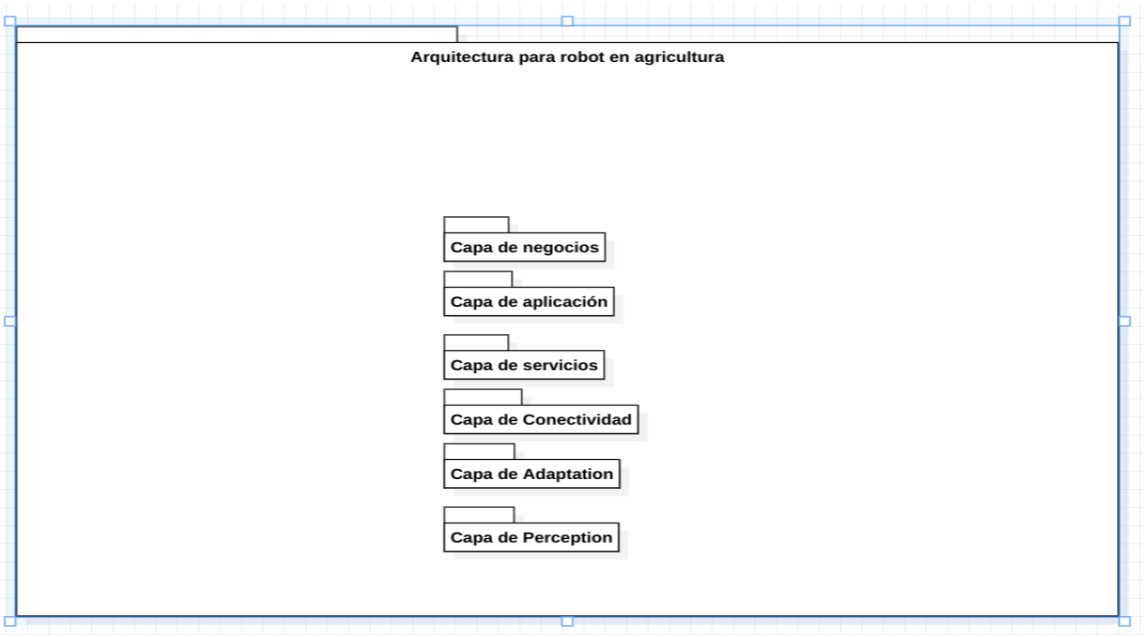


Figura 26. Vista inicial del diseño la arquitectura

Instanciación

(Descripción de manera gráfica y teórica de las vistas generadas a partir de las prácticas IoTAP: ***(i) lógica, (ii) procesos, (iii) implementación o (iv) despliegue***).

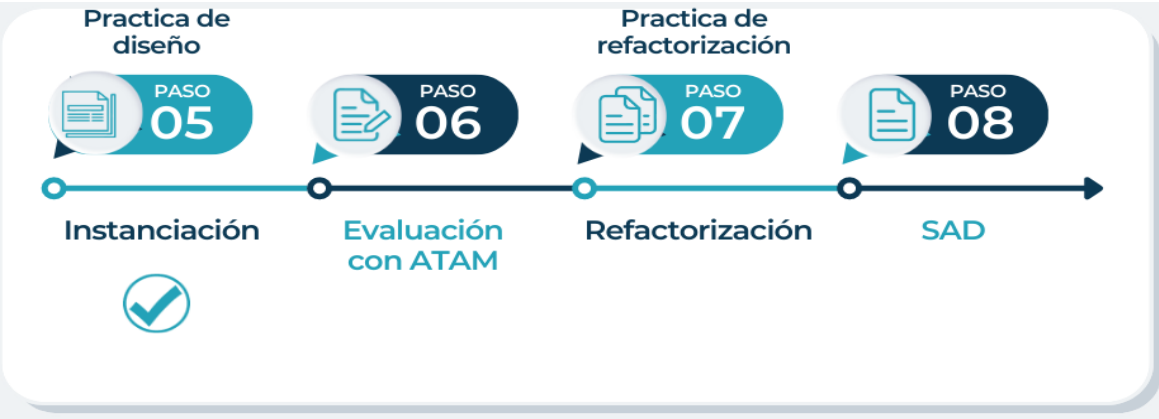


Figura 27. Paso 5 de la metodología correspondiente a la instanciación de la arquitectura

Como resultado de aplicar la práctica de diseño para arquitecturas basadas en IoT, se obtiene un diseño preliminar de la vista lógica de la arquitectura para ***"Arquitectura de software adaptativa para soportar la cooperación de mini robots semi-autónomos en el contexto de la agricultura de precisión".***

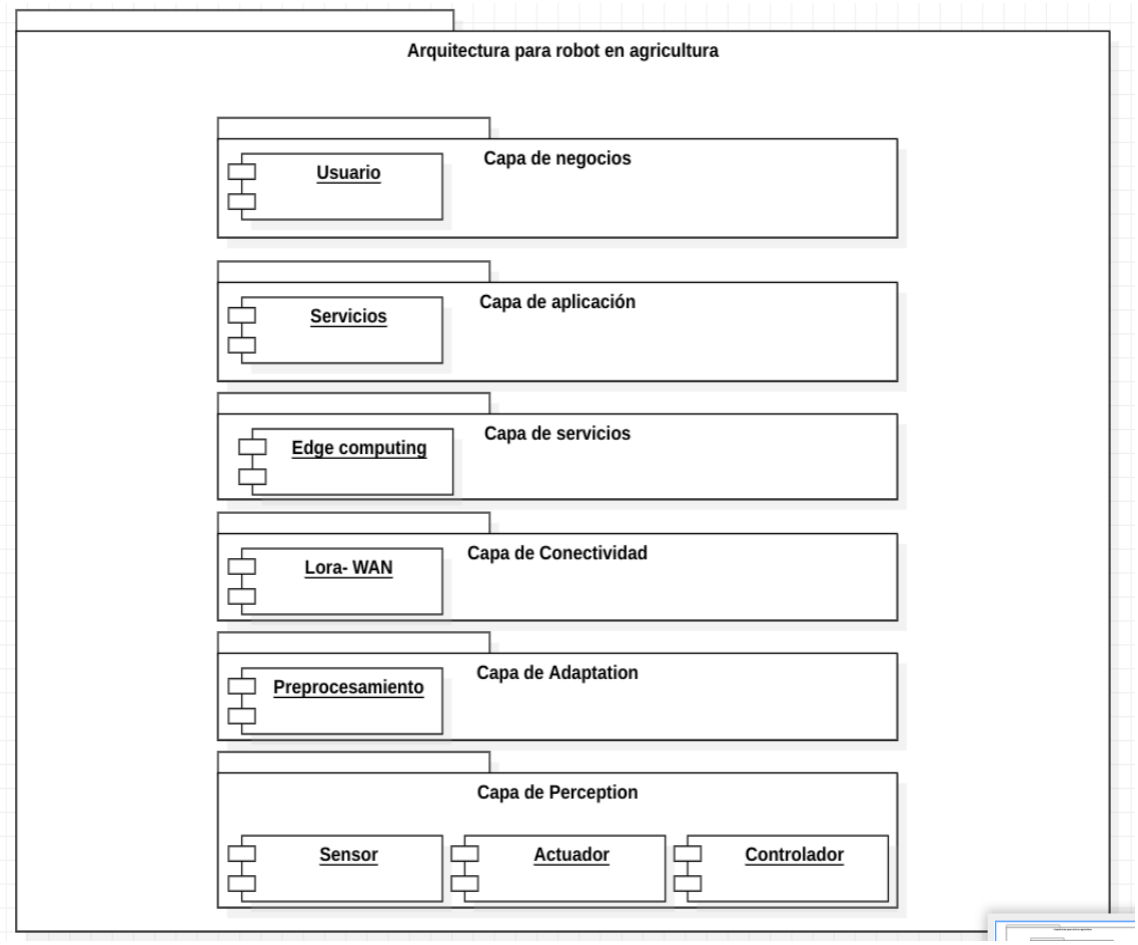


Figura 28. Arquitectura resultante de aplicar la práctica de diseño IoTAP

Evaluación con los artefactos de ATAM seleccionados

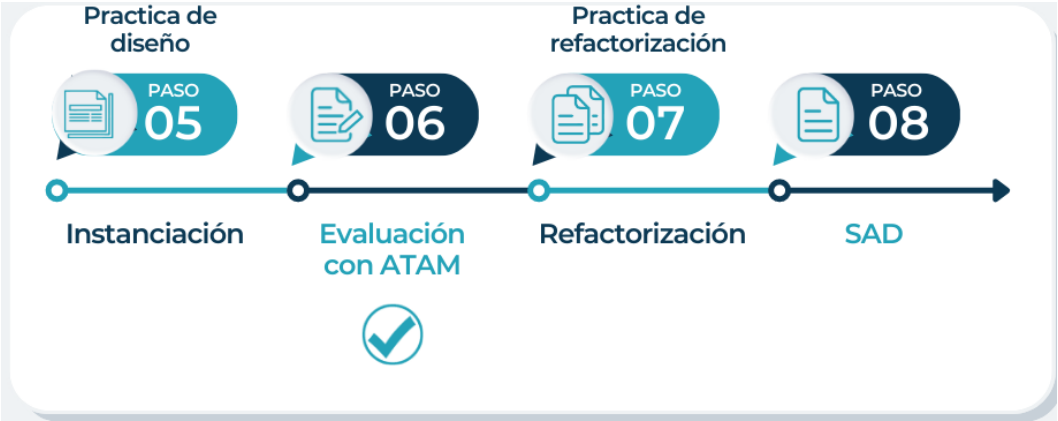


Figura 29. Paso 6 de la metodología correspondiente a la evaluacion con artefactos de ATAM seleccionados

De la evaluación con la metodología ATAM adaptada para la presente metodología se tienen en cuenta solo dos artefactos que son cruciales para la mejora continua de las

arquitecturas y corresponden a**: (i) Árbol de Utilidad y (ii) Análisis de una propuesta arquitectónica.**

Los atributos sensibles para la presente arquitectura son: **(i) Rendimiento, (ii) Modificabilidad y (iii) Mantenibilidad.** Estos se representan en el árbol de utilidad para identificar las preocupaciones y escenarios para la arquitectura, además se presentan las sensibilidades, riesgos y en caso de existir otros QA relevantes. Por otra parte, se realiza el **análisis de una propuesta arquitectónica** para examinar los escenarios, sensibilidades y riesgos identificados en árbol de utilidad.

- **Rendimiento:** Se observa que es un atributo sensible teniendo en cuenta que el sistema manipulará cantidades de datos considerables, en zonas rurales donde la cobertura se puede presentar de forma deficiente. Se debe garantizar que esa capa de conexión provea y reciba de las capas inferiores los elementos necesarios para un funcionamiento adecuado.

- **Modificabilidad:** la sensibilidad a cambios se presenta en un nivel considerable por tanto sería apropiado tener en cuenta la cohesión y acoplamiento, buscando de cada proceso de cierta manera cumpla su responsabilidad pero que haya una tolerancia a fallos que permita al sistema dar respuesta rápida y reponerse anta cualquier evento.

- **Mantenibilidad:** la separación de responsabilidades es importante y se debe prevenir niveles de dependencia altos, esto previene los cambios.

Árbol de utilidad

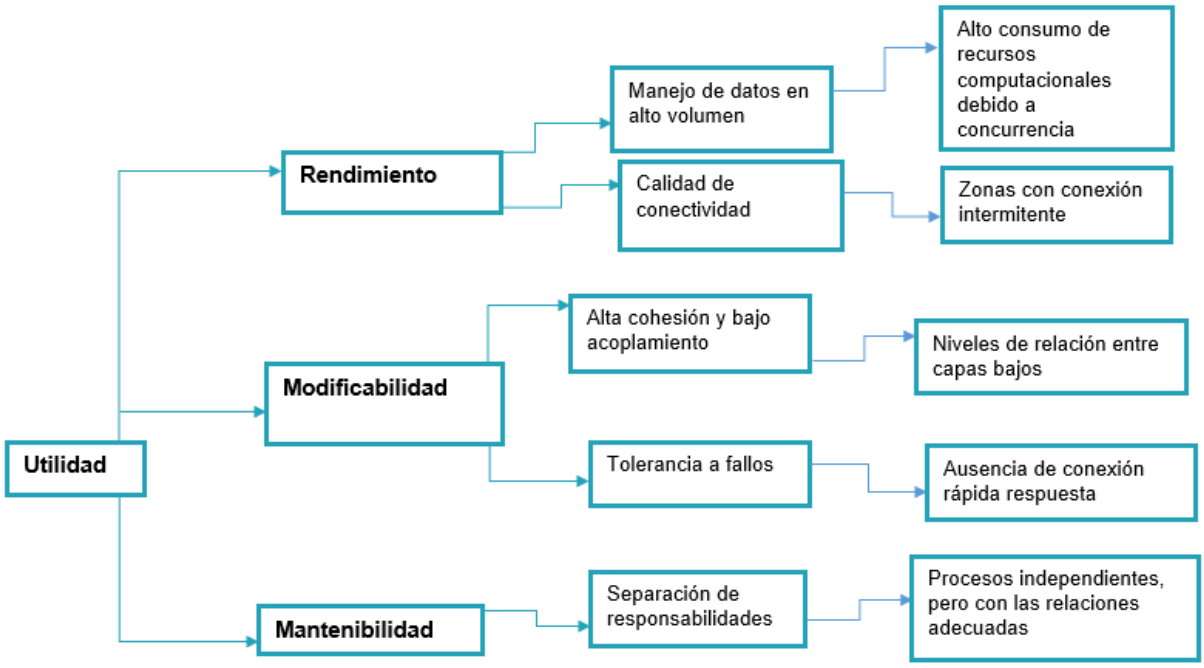


Figura 30. Arbol de utilidad perteneciente al metodo ATAM resultante de la evaluación realizada por la arquitecta de software

Análisis de una propuesta arquitectónica

Ítem	Descripción			
Escenario	Alto consumo de recursos computaciones debido a concurrencia			
Atributo	Rendimiento			
Ambiente	Operaciones normales			
Estimulo	Acceso concurrente a la información percibida.			
Respuesta	Se apaga la tarjeta			
Decisiones arquitecturales (DA)	DA	Riesgo	Sensibilidad	Compensación
	1 recursos eficientes	Alto	Alto	Optimización de algoritmos.
	2 transferencia de tarea	Alto	Medio	Respuesta y respaldo al proceso
Razonamiento	Los recursos del sistema deben ser utilizados óptimamente, para lograr la robustez del sistema y su correcto funcionamiento (en tiempo real), razón por la cual estoy de acuerdo debemos implementar la tácticas para lograr que el sistema pueda desarrollar las tareas en tiempo real y así lograr la toma de decisiones basado en los datos obtenidos del sistema, la capa impactada por la táctica es de adaptación y se verificará el impacto en las demás capas, es importante tener en cuenta que esta táctica puede impactar la arquitectura de software afectando el rendimiento de los agentes.			

Tabla 30. Resultado de abordar el escenario "Alto consumo de recursos computaciones debido a concurrencia" generado en el arbol de utilidad

Ítem	Descripción			
Escenario	Zonas con conexión intermitente			
Atributo	Rendimiento			
Ambiente	Operaciones normales			
Estimulo	Perdida de conexión			
Respuesta	Cero transmisión de datos			
Decisiones arquitecturales (DA)	DA	Riesgo	Sensibilidad	Compensación
	1 funcionamiento offline	Alto	alta	Proceso continuo, pero cambio en la eficiencia
	2 precargado en memoria temporal	Alto	alta	Afecta la eficiencia.
Razonamiento	Los datos obtenidos por los agentes es el insumo para poder representar el estado del cultivo, razón por la cual se debe garantizar el funcionamiento del sistema de forma offline guardando temporalmente la información, por lo cual estas tácticas permiten que los datos obtenidos se almacenen temporalmente y al momento de restablecer, lograr ser transmitidos en lo posible sin pérdida de información, así mismo se debe tener en cuenta que puede afectar el rendimiento de los agentes por lo cual se debe tener en cuenta este impacto a la arquitectura de software.			

Tabla 31. Resultado de abordar el escenario "Zonas con conexión intermitente" generado en el arbol de utilidad

Rationale

De acuerdo a lo encontrado anteriormente en la sección de casos y escenarios de calidad, se identifican las preocupaciones y/o sensibilidades principales para la arquitectura seleccionada, al igual que las tácticas, patrones y/o decisiones de diseño).

Driver candidato	Preocupación	Tácticas	Patrones satisfactores y/o decisiones de diseño
Eficiencia de desempeño	**Alto volumen de datos y conexión intermitente**	**Priorización de eventos Incrementar eficiencia de los recursos**	Recursos eficientes Transferencia de tarea Funcionamiento offline Precargado en memoria temporal

Tabla 32. Identificación del driver candidato, preocupaciones, tacticas y decisiones de diseño

Refactorización de la arquitectura de software

Figura 31. Paso 7 de la metodología correspondiente a la refactorización de la arquitectura

Impacto de las tácticas en las capas

Extiende el razonamiento realizado en vistas de casos y escenarios de calidad, se observa el impacto que pueden tener estas tácticas, dentro del patrón y como impactan con respecto a los atributos de calidad.

- **Capa de adaptación:** estas tácticas permiten que el sistema pudiera adaptarse a los cambios presentados como el perder conectividad de los robots y mantener las tareas en ejecución, permitiendo la implementación de un gestor de eventos que priorice la información y su verificación de cambios si son relevantes o con cambios significativos, así mismo como la conectividad en el momento, manejo de eventos que pueden ocurrir en el sistema, como el nivel de conectividad, manejo de información del contexto entre otros, estas tácticas pueden tener un impacto negativo en la eficiencia de desempeño del sistema, se evalúa si este degradación es mayor que enviar la información sin que ocurran los eventos.
- **Capa de percepción:** los datos tomados por lo robots son muy importantes y se pone de manifiesto en la toma de decisiones, razón por la cual estas tácticas permiten que no se pierda información al almacenarla en una memoria temporal, siempre verificando que puedan conectarse de nuevo al sistema y ser cargada, esto podría ocasionar en los robots la baja rendimiento de los recursos y su operatividad, se necesita de la optimización de los algoritmos para la toma de variables del entorno. debe tener en cuenta la configuración, calibración y precisión de los

sensores actuadores y tarjetas nivel de validación en los algoritmos para que la capa de adaptabilidad razone de forma óptima, con base en la información recolectada del contexto, si no tenemos en cuenta estas validaciones, los datos almacenados en la memoria temporal podrían afectar la eficiencia y rendimiento del sistema, estas tácticas afectaron la capacidad de la memoria temporal, pero al utilizar este filtro optimizar los datos obtenidos y evitará la saturación del sistema.

Filtro: si las variables no cambian no necesita transmitir.

En los atributos de calidad seleccionados para la presente arquitectura, no se presentan conflictos entre los mismo, al contrario, las tácticas apoyan y mejoran el impacto que ellos pueden tener en la arquitectura a desarrollar.

Se genera una ***nueva vista lógica***, la sigla ***PF*** corresponde al driver ***Performance*** o más conocido como ***eficiencia de desempeño***. A partir de las tácticas implementadas ***PF1 Incrementar eficiencia de los recursos y PF2 Priorización de eventos***, se denota con la sigla ***AI*** que se introdujeron las tácticas mencionadas anteriormente en el patrón de capas. Esto soportado por el trabajo [28], el cual propone un método ***para anotar diagramas de arquitectura con información de implementación de la táctica***. Es una manera clara de documentar la adición de tácticas a los documentos de arquitectura que también es fácil de usar, permite que los componentes arquitectónicos permanezcan visibles y puede anotar muchos estilos diferentes de diagramas. Se basa en los tipos de cambios en los patrones de arquitectura descritos anteriormente. El método de anotación consta de dos cosas: una lista de tácticas y los círculos que muestran la ubicación y el tipo de cambios en la arquitectura para una táctica. La lista de tácticas consta de entradas que asocian una táctica implementada con el círculo que muestra su implementación. Cada entrada en la lista de tácticas consta de un identificador único y el nombre de la táctica. El identificador consiste en una abreviatura del atributo de calidad, seguida de un número. El número identifica de forma única los cambios para esa táctica.

Table 5
Sample quality attribute abbreviations.

R	Reliability
S	Security
U	Usability
PF	Performance
PO	Portability
M	Maintainability
CP	Capacity
CF	Configurability
E	Extensibility

Table 6
Sample tactic list.

ID	Tactic
S1	Authorization (security ++)
S2	Encryption (security ++)
R3	Ping/Echo (reliability − −)
PF4	Concurrency (performance ~)

Table 7
Types of change to pattern participants.

Abbreviation	Type
I	Implemented in
R	Replicated
AI	Added, in the pattern
AO	Added, out of the pattern
M	Modified

Atypical circle
looks like this:

Fig. 4. Tactic implementation annotation style.

Fig. 5. Sample tactic implementation annotation.

Figura 32. Metodo de anotaciones para arquitecturas de software

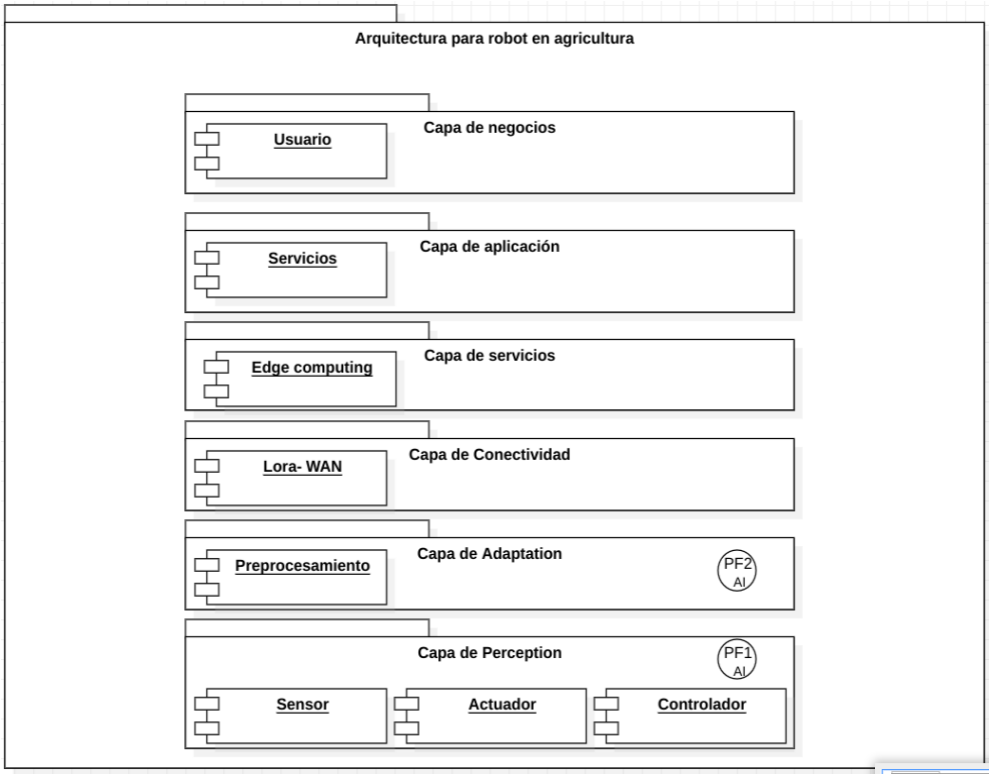


Figura 33. Resultado de aplicar el metodo de anotaciones

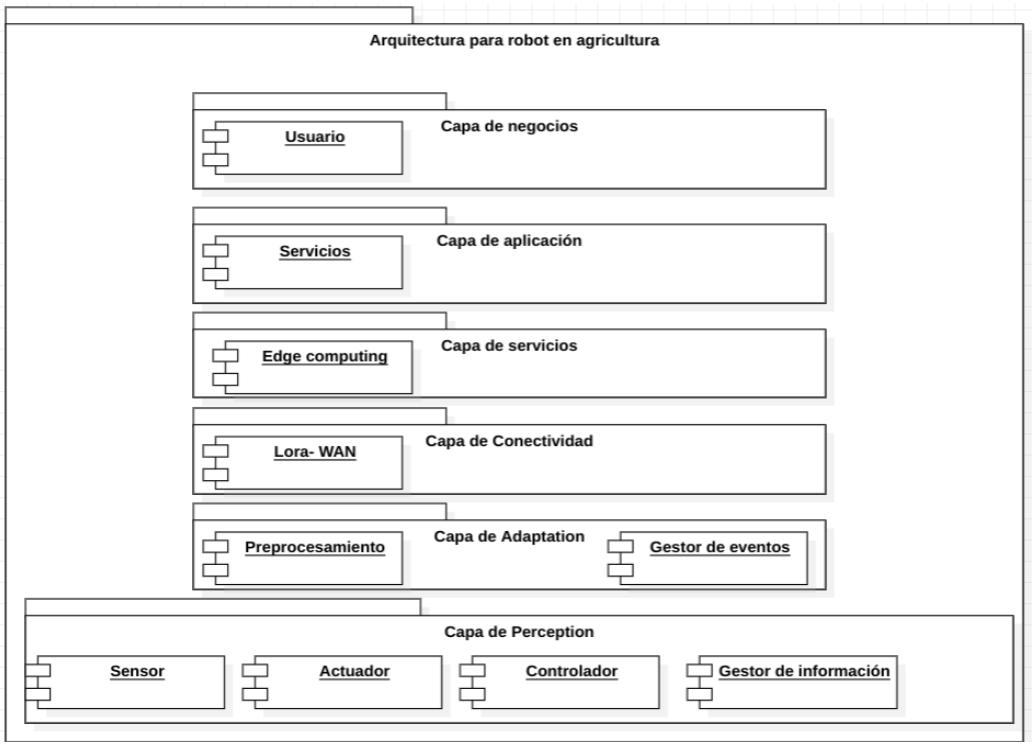


Figura 34. Arquitectura refactorizada con las tacticas sugeridas por la evaluación mediante el metodo de anotaciones

Se introducen los cambios en la arquitectura de software, en la capa ***de percepción el gestor de información y en la capa de adaptabilidad el gestor de eventos***, para instanciar el uso de las tácticas identificadas por parte de la evaluadora de arquitecturas ***Flor Hernández***, con el objetivo de mejorar la calidad de la arquitectura. La información de estas actividades se plasma en el **Software Architecture Document (SAD).**

Generación del software Architecture Document

Figura 35. Paso 8 de la metodología correspondiente al software architecture document

La información de estas actividades se plasma en el **Software Architecture Document (SAD)**, la totalidad del documento se podrá encontrar en el **Anexo D**.

Protocolo de Observación

Id	Fenómeno Observado	Práctica/Tarea	Artefacto	Reflexión
1	Atributo de calidad relevante denominado como "Sensibilidad al entorno" no es tan tenido en cuenta en la IoT	Práctica de diseño	Catálogo	Se debe prestar más atención a este atributo de calidad ya que es transversal cuando de adaptabilidad de habla.
2	Se observó que el proceso para calificar el archivo Excel que permite reconocer impacto de los drivers en el patrón debe mejorar e su explicación teniendo instrucciones en el mismo.	Práctica de diseño	Excel de calificación	Se debe contar con instrucciones dentro del documento que permita el proceso de calificación.
3	Cuando hay un empate en el archivo Excel entre patrones se debe analizar y evaluar porque un	Práctica de diseño	Excel de calificación	El criterio de desempate debe ser la comparación entre los mismos patrones permitiendo reconocer cual cumple mejor

	patrón es mejor que otro.			la tarea según los drivers escogidos.
4	El proceso se facilita mucho más si los participantes conocen sobre el tema de arquitecturas de software	Evaluación de la arquitectura con los artefactos seleccionados de ATAM	Artefactos ATAM	El método y las prácticas diseñadas facilitan el proceso de diseño y refactorización de una arquitectura siempre y cuando se seleccionen los artefactos de ATAM como el árbol de utilidad y los escenarios, es decir, acotar el método ATAM para que no sea tan engorroso.
5	El proceso de anotaciones de arquitectura de software complementa de muy buena forma los cambios realizados sobre la arquitectura	Práctica de refactorización	Anotaciones	El proceso de anotaciones facilito los cambios a realizar a la arquitectura de software permitiendo observar donde serán introducidas las tácticas con información asociada y su impacto en la estructura general.

Hallazgos estudio de caso aplicado

- Apreciaciones de la evaluadora Flor Hernández: ella menciona que la metodología es fácil de identificar ya que cuenta con pasos cortos, sencillos y claros evitando que la metodología sea engorrosa y difícil de entender. Además, las prácticas de ***(Diseño y refactorización)*** son fáciles de usar, comprensibles y útiles, adecuándose para apoyar el diseño de una arquitectura basada en IoT ya que mejora continuamente la calidad de la misma. Evaluar la arquitectura fue una tarea sencilla de realizar ya que el proceso se encuentra bien estructurado y también ayuda a entender el requerimiento de manera clara y concisa. En conclusión, la evaluadora percibe que la metodología y las prácticas diseñadas son comprensibles al igual que útiles cuando se realiza una arquitectura para productos software basados en IoT.

- Apreciaciones del arquitecto Giovanni Zambrano: él menciona que la metodología en general es muy práctica, muy amigable para la persona que la usa, debido a que tiene un alcance claro, ahorra tiempo y es fácil de usar al punto que recuerda los pasos enunciados por la metodología. También, considera que las prácticas (***Diseño y refactorización***) son comprensibles y útiles ya que las diferentes etapas como el diseño porque es cómodo ya que los pasos son claros, en la evaluación se pueden detectar debilidades, sensibilidades que pueden ser refactorizadas. La aplicación de la metodología permitió obtener una vista de la arquitectura de manera rápida ya que seguir los pasos planteados facilita el diseño de la misma, su evaluación y su refactorización.

El link de las grabaciones de la aplicación del estudio de caso se presenta a continuación: https://drive.google.com/drive/folders/1nRi-qqiIkOBnCDllvsVdQwDoBSNxth-f?usp=sharing

Análisis de resultados

- De acuerdo a la sesión con expertos y el desarrollo del caso, los usuarios potenciales y usuarios reales percibieron las prácticas de arquitectura como comprensibles y útiles para el diseño arquitectónico en el contexto de la IoT.

- Las prácticas enmarcadas en el perfil de calidad y la metodología diseñada facilitan el desarrollo de la arquitectura de software teniendo en cuenta los atributos de calidad ***(Attribute-Driven)*** relevantes para una aplicación basada en IoT incluyendo la sensibilidad al contexto como un atributo de calidad clave para este tipo de soluciones, realizando propuestas y mejora continua, refactorización ***(Tactic-Driven)*** que influye en el mantenimiento de este tipo de aplicaciones.

- Todos los expertos concluyen que la metodología es una secuencia de pasos con objetivos bien claros y concretos, guía para poder realizar el proceso de diseño y refactorización.

- La aplicación del estudio de caso permitió identificar fenómenos que ocurrieron tras la aplicación de las prácticas junto con la metodología, esto sirve para mejorar la propuesta a nivel de proceso adecuando de mejor forma los pasos usados para generar el diseño de una arquitectura.

- El estudio de caso permitió observar y validar de una forma realista el proceso evaluado por expertos, encontrando puntos por mejorar y ayudando a generar valor en el proyecto donde se aplicó el presente estudio.

- Las hipótesis alternativas 1 y 2 se cumplen ya que las prácticas se perciben útiles y comprensibles de acuerdo a los resultados por parte de los expertos y el estudio de caso aplicado.

- Las prácticas IoTAP proporcionan un marco de trabajo bien definido ya que poseen pasos que permiten su correcta aplicación, aportan en que, el diseño se haga más rápido, con un objetivo claro mejorando su calidad. A nivel de tareas y artefactos las prácticas son bien definidas para proporcionar una orientación útil que permita lograr un diseño inicial de la arquitectura requerida según el dominio de aplicación y las necesidades de los usuarios reales y potenciales.

- El perfil de calidad diseñado para la presente propuesta permitió enmarcar las prácticas IoTAP en actividades como el diseño, evaluación y refactorización para tener un orden, permitiendo la organización y agrupamiento de las mismas con el objetivo de establecer unos pasos bien claros y definidos que ayudaran en la obtención del objetivo principal que es una arquitectura basada en IoT teniendo en cuenta el dominio en donde se aplica.

- El catálogo de atributos de calidad, tácticas, patrones y arquitecturas de referencia permitió que el arquitecto de software empezara el proceso de aplicación de las prácticas IoTAP de una manera más fácil ya que contaba con un banco pequeño de conocimiento que le permitía consultar información relevante evitando buscar información dispersa y optimizando el tiempo para realizar el diseño inicial de la arquitectura que se requiere.

- En la actividad de diseño, se aborda la selección de atributos claves, selección de patrones teniendo en cuenta los drivers para la arquitectura que se diseña y, por último, la instanciación de la misma para generar el diseño inicial, se puede concluir que los pasos abordados facilitan el diseño de una manera más ágil debido a que es intuitivo y secuencial logrando la optimización del tiempo para realizar el diseño de una arquitectura basada en IoT.

- En la actividad de evaluación se toman los artefactos seleccionados del método ATAM (Árbol de utilidad y Análisis de una propuesta arquitectónica) que permiten realizar la evaluación de arquitecturas teniendo en cuenta los atributos sensibles para la misma, permitiendo la generación del árbol de utilidad con el objetivo de generar escenarios que se deben tener en cuenta o que son preocupantes, luego estos escenarios se analizan y se realiza un razonamiento, el cual permite establecer que driver es candidato para ser abordado mediante tácticas en la actividad de refactorización optimizando el tiempo.

- En la actividad de refactorización de arquitecturas basadas en IoT, se emplea el método de anotaciones que tiene una gran utilidad debido a que permite una clara forma de documentar la adición de tácticas a las vistas de las arquitecturas ayudando a que los componentes arquitectónicos permanezcan visibles permitiendo establecer el impacto que las tácticas tienen sobre las arquitecturas que se están diseñando y un recuento de los cambios que se realizan en las mismas.

- El Software Architecture Document permitió la consolidación de toda la información generada a través de las actividades (Diseño, evaluación y refactorización) y pasos enmarcados en el perfil de calidad para establecer la traza de las iteraciones que se realizan aplicando la metodología y las prácticas IoTAP a la arquitectura que se está diseñando.

- Determinar los pre requisitos de las prácticas es importante y plantear el perfil de la persona que use las prácticas es importante para que todo en conjunto funcione de una mejor manera

5.1.5. Amenazas de validez

- **Validez de constructo:** Para minimizar la subjetividad en los instrumentos que soportan la recolección de información del estudio de caso, durante su planificación es necesario que los instrumentos fueron sometidos a validaciones con la participación de los investigadores. Otra amenaza de este tipo identificada es la incorporación y manejo de nuevos elementos conceptuales y de lenguaje en el desarrollo del estudio. Con el fin de reducir esta amenaza como parte del estudio de caso, se planificó una actividad inicial en la cual se socializó y contextualizó a los participantes en los detalles del estudio para lograr una unificación de los conceptos y el lenguaje. La Escala de Likert es una escala de calificación que se emplea para preguntar a una persona sobre su nivel de acuerdo o desacuerdo con una declaración. Es ideal para medir reacciones, actitudes y comportamientos de una persona, en la presente propuesta se utiliza para la ***evaluación con los***

expertos indagando acerca de la percepción de la facilidad de uso y utilidad tras la aplicación de las prácticas IoTAP. La encuesta está estructurada por 19 preguntas que se ilustran en el referente Davis, [64] y usan la escala anteriormente mencionada. Para eliminar amenazas y sesgos se hace necesario utilizar las preguntas del referente debido a que están bien redactadas, con un objetivo claro, conciso y bien explicado ayudando a obtener información confiable.

- **Validez interna:** El tiempo invertido para la ejecución del estudio puede llegar a ser una amenaza de validez, al ser sesiones largas, los participantes en las etapas finales del estudio pueden percibir cansancio y esto puede influir en los resultados. Para tratar de contrarrestar esta amenaza, se definieron franjas horarias adecuadas para cada etapa en la ejecución del estudio de caso. La información que determina que las prácticas son comprensibles y útiles corresponden al ***estudio de caso*** ya que se realizaron una serie de preguntas abiertas cuando finalizaba cada actividad correspondiente en el perfil de calidad para conocer el punto de vista de los participantes y también sus apreciaciones, lo cual elimino la amenaza de obtener información sesgada ya que se pudo contrastar con la información obtenida por los expertos notando que la información cualitativa obtenida durante el desarrollo del estudio de caso apoya a la información recolectada en la evaluación con los expertos.

- **Validez externa**: En el desarrollo del estudio de caso se contará con expertos en arquitecturas de software, es un poco complejo encontrar este tipo de perfiles, por ello, esta situación se puede convertir en una amenaza de validez debido a que encontrar expertos en este tema es una tarea difícil, de tal forma que, habrá que suplir esta carencia. El método Attribute Driven Design (ADD) realizado por Wojcik et al. [35], es un enfoque para definir una arquitectura de software. Su proceso de diseño se basa en los requisitos de los atributos de calidad del software. ADD sigue un proceso de diseño recursivo que descompone un sistema o elemento del sistema aplicando tácticas arquitectónicas. En el anterior método se puede observar que se realiza el proceso de identificación de Atributos, tácticas y patrones. El ciclo común para el diseño de las arquitecturas es: (i) Drivers→ (ii) Tácticas→ (iii) Patrones. Mientras que en la presente propuesta el diseño de la arquitectura se aborda de la siguiente forma: (i) Drivers → (ii) (Patrones→Tácticas)*, esto apoyado por [28], aportando a que los diseños de arquitecturas para IoT sean más agiles y permitan optimizar el tiempo de su diseño. Según [18], la IoT debe basarse en prácticas y procesos estandarizados para desarrollar soluciones y sistemas que puedan usarse en todas las industrias, debido a que el papel de la arquitectura es fundamental para proporcionar un modelo del sistema IoT general en términos de componentes básicos, hardware, software e interconexiones para soportar la ejecución de operaciones. Por ello, es necesario proponer un ciclo diferente como el de esta propuesta, que permita obtener la arquitectura de una manera más temprana debido a que es necesaria desde las fases más tempranas en el ciclo de desarrollo de software. En cuanto al perfil de calidad diseñado, catálogo de atributos, las prácticas IoTAP y la metodología generados en la presente propuesta en el capítulo 3 y 4, estos apoyan a que el diseño de las arquitecturas se haga de una manera más

rápida, confiable y consistente permitiendo avanzar de una mejor forma en el desarrollo de aplicaciones IoT con un dominio especifico.

5.1.6. Limitantes

- El tiempo con el que contaban los expertos era limitado, por lo que en algunas ocasiones fue difícil realizar algunas tareas, por ello se decidió a calcular el tiempo que tomaba hacer cada fase de la ejecución del estudio de caso.
- El catálogo de atributos, tácticas, patrones y arquitecturas de referencia es aún muy pequeño, pero se espera mejorarlo tras su aplicación en diferentes ámbitos y dominios de aplicación en IoT.
- El contraste de la influencia de los atributos de calidad en los patrones de arquitectura puede mejorar debido a que se puede complementar con otros puntos de vista, referentes o técnicas que ayuden a su enriquecimiento.
- La base de conocimiento fue suficiente para el presente estudio de caso, sin embargo, debe complementarse un poco más para guiar de una mejor forma en la aplicación de las prácticas IoTAP, su metodología asociada y el perfil de calidad diseñado.
- La plantilla de análisis de una propuesta arquitectónica y la de contraste de la influencia de los atributos de calidad en los patrones de arquitectura son entendibles, sin embargo, debe ser mejor explicada con información adicional para guiar a la persona que la esté usando.
- Los catálogos son importantes para la presente propuesta debido a que las actividades del perfil de calidad se apoyan en ellos y en particular la de diseño, si bien se contaba con la información mínima que permitió la ejecución del estudio de caso, es necesario complementar el catálogo para que la actividad de diseño aporte un poco más de lo que hace actualmente.

6. CAPITULO 6

Descripción Capitulo

En este trabajo de grado de maestría se han propuesto prácticas de diseño y refactorización de la arquitectura y sus requisitos de calidad en el contexto de desarrollo de software basado en IoT. También la evaluación de la arquitectura por medio de los artefactos seleccionados de la metodología ATAM (Árbol de utilidad y Análisis de una propuesta arquitectónica). Todo este proceso enmarcado en el perfil de calidad diseñado que cuenta con las actividades generales como (i) diseño, (ii) evaluación y (iii) refactorización donde se aplican las prácticas en forma metodológica, es decir, mediante una serie de pasos que permiten abordar cada actividad para lograr el diseño. Posteriormente la evaluación y por último la refactorización de arquitectura teniendo en cuenta la evaluación. También, se cuenta con un catálogo de información que contiene tácticas, patrones, atributos de calidad y arquitecturas de referencia que permiten consolidar la información encontrada en la literatura y también la información resultante de aplicar la metodología y sus prácticas asociadas. Este trabajo fue elaborado a partir de: (i) Caracterización de los atributos de calidad especiales del producto software basado en IoT y de las prácticas de arquitectura de software, (ii) Organizar a nivel conceptual y metodológico un conjunto de prácticas de arquitectura y sus requisitos de calidad que contribuyan al desarrollo del producto software basado en IoT a partir de la caracterización realizada anteriormente y (iii) Evaluar la

comprensibilidad y utilidad percibida por desarrolladores de software de la región al utilizar las prácticas IoTAP mediante un estudio de caso.

Adicionalmente, con el fin de divulgar el presente trabajo de investigación se han escrito 2 artículos y se realizó un sitio web que contiene la información más importante del presente proyecto:

- ***IoTAP: Prácticas de diseño de la arquitectura y sus requisitos de calidad en el contexto de desarrollo de software basado en IoT*** el cual ha sido aceptado en la revista Ingenierías USBMed indexada como categoría C en el Publindex.

- ***SEMIoTICA - Security Scenarios Modeling for IoT-based Agriculture Solutions*** el cual ha sido enviado al evento Decisioning 2023 y fue aceptado para publicación.

- ***Prácticas para el desarrollo de la arquitectura de un producto software basado en IoT: una evaluación de expertos***, el cual ha sido aceptado como ponencia en el Sexto Congreso Andino de Computación, Informática y Educación - CACIED 2023.

- ***Sitio web con el resumen de todo el trabajo de grado*** incluidas las prácticas, catálogo, metodología, etc. https://sites.google.com/unicauca.edu.co/iotap/inicio.

6.1. Conclusiones, trabajo futuro y lecciones aprendidas

6.1.1. Conclusiones

- La IoT en general, es una red heterogénea de elementos muy compleja, específicamente en la agricultura algunos módulos o tecnologías son incapaces de interactuar, lo que aumenta aún más la complejidad del desarrollo de este tipo de productos al tener varias tecnologías y varios tipos de dispositivos.

- La caracterización realizada permitió identificar los atributos de calidad especiales del producto software basado en IoT, con el fin de encontrar cuales son los atributos más abordados, menos abordados y cuales no se tienen en cuenta cuando se realiza el desarrollo de este tipo de productos.

- La caracterización realizada permitió generar un catálogo de atributos de calidad usados en el diseño de arquitecturas de software basadas en IoT teniendo en cuenta su importancia y relevancia.

- Estos atributos de calidad son sensibles para el software basado en IoT por eso, es necesario enfocarse en el grupo identificado en el capítulo 4 ya que el aporte que se puede realizar a nivel técnico abre más las posibilidades para el correcto desarrollo y gestión de este tipo de productos software.

- La caracterización realizada permitió encontrar las prácticas de arquitectura de software más comunes en productos basados en IoT. Esto permitió

seleccionar las prácticas más usadas con respecto a la arquitectura de software de los productos anteriormente mencionados.

- Una arquitectura de software junto con buenas prácticas puede aportar en la evolución, modelado, desarrollo y ejecución de productos software complejos basados en IoT, fortaleciendo la calidad y funcionalidad requeridas.

- Las nuevas tecnologías como la IoT deben mejorar en todos los dominios de aplicación y en específico, en el dominio de agricultura inteligente con el fin de contribuir en la productividad de alimentos, monitorizando y proporcionado alimento limpio, reduciendo el trabajo humano y haciendo más eficiente la producción.

- El perfil de calidad planteado junto con la clasificación de las prácticas encontradas en la literatura permitió el diseño de la metodología donde se incluyen las prácticas diseñadas (Diseño y refactorización) y que enuncia los pasos necesarios para lograr el diseño de una arquitectura para software basado en IoT mediante las actividades de diseño, evaluación y refactorización.

- El catálogo de atributos de calidad, tácticas, patrones y arquitecturas de referencia puede ayudar a futuros arquitectos de software que deseen diseñar arquitecturas de software basadas en IoT teniendo en cuenta el perfil de calidad, las prácticas y la metodología diseñada para el presente estudio.

- Las prácticas diseñadas en el capítulo 4 permiten la aplicación del perfil de calidad y de la metodología planteada ya que poseen pasos propuestos bien delimitados que permiten llevar a cabo las actividades de diseño y refactorización.

- Con respecto a la evaluación se adaptó el método ATAM con el objetivo de hacerlo un poco más flexible y ágil permitiendo evidenciar preocupaciones y sensibilidades de manera más temprana por medio de sus artefactos clave como lo son el árbol de utilidad y el análisis de la propuesta arquitectónica.

6.1.2. Trabajo futuro

- Aún persiste una brecha que debe ser abordada y es principalmente en la reducción de costos de hardware y software en el producto software basado en IoT a la vez que se maximiza el rendimiento de dichos productos.

- En el tema de la estandarización es importante mejorar atributos de calidad como lo es la interoperabilidad entre diferentes aplicaciones, sistemas documentando pautas o prácticas específicas de la industria, agricultura entre otras para la implementación eficiente en la IoT.

- Uno de los temas más cruciales en el desarrollo del producto basado en IoT es la gestión de la energía ya que estos productos están compuestos por gran cantidad de elementos como lo son antenas, módulos, algoritmos,

sensores o actuadores entre otros lo que implica optimizar el uso de la energía.

- La sensibilidad al contexto es una característica especial de la IoT que debe ser investigada más a profundidad ya que a veces no es desafiante saber cómo gestionar todo el conjunto de datos recopilados por los sensores. Por ello, la sensibilidad al contexto podría ayudar a decidir qué, y cuales datos deben procesarse. En lo referente al dominio de la agricultura todavía se requiere trabajo en dicha característica especial.

- Un atributo de calidad especial para los productos software basados en IoT, es la tolerancia a fallos, que cada vez está más ausente en estos productos, el nivel de tolerancia fallos debe mantenerse alto si se requiere un sistema impecable que a pesar de un error técnico como que los módulos hardware pueden fallar porque la batería se agotó, una calibración defectuosa o una falla de comunicación se busca que todo el sistema siga funcionando.

- Otro atributo de calidad que debe abordarse y que se plantea como desafío es el mantenimiento ya que es necesario diseñar productos IoT de bajo mantenimiento que pueda realizar actividades automáticamente sin intervención humana de forma óptima, reduciendo el tiempo y costo de mantenimiento hasta un nivel de aceptación bajo.

- La confiabilidad como atributo de calidad aun plantea desafíos que deben ser solucionados en el dominio de la agricultura inteligente debido a que se realizan trabajos duros en dicho dominio, con la consecuencia de una rápida de degradación de componentes provocando fallas y minimizando la calidad del producto. Se debe asegurar que dichos componentes puedan soportar condiciones climáticas extremas como altas temperaturas y humedad y tormentas etc.

- En el tema de las arquitecturas, se hace necesario seguir trabajando en el atributo de calidad portabilidad ya que las arquitecturas actuales carecen de elementos que pueden ayudar a que sean a lo suficientemente solicitadas con aspectos particulares de los requisitos que plantea la agricultura. Por otra parte, la robustez como atributo de calidad también plantea brechas con respecto a las arquitecturas ya que deben ser tolerantes a fallos para que las aplicaciones puedan garantizarse.

- Se debe trabajar en la mejora continua de las prácticas, perfil, catálogo y metodología con el fin de generar una aplicación tipo Dashboard que me permita automatizar el proceso generado por los pasos de la metodología y también un perfil del lugar dependiendo del dominio aplicado para tener más claridad acerca de la medición de los datos y variables correctas teniendo en cuenta el universo del discurso y las necesidades del cliente.

6.1.3. Lecciones aprendidas

Arquitectura

- Para poder evaluar las prácticas de diseño y refactorización fue necesario crear el perfil de calidad incluyéndolas en actividades para poder diseñar una metodología que permitiría la aplicación de lo anterior mediante pasos bien definidos.

- Se debe tener en cuenta que para posteriores versiones de las prácticas, perfil y metodología es necesario aplicarlas en diferentes proyectos y contextos con el fin de mejorarlas complementándolas de forma que ayuden generando valor en la industria y proyectos.

- La creación del catálogo de atributos de calidad, tácticas, patrones y arquitecturas de referencia fue un gran punto de la presente investigación ya que representa una gran ayuda para industria y la academia aplicada al software basado en IoT.

- Los catálogos son importantes para la presente propuesta debido a que las actividades del perfil de calidad se apoyan en ellos, por ello es necesario complementar de mejor forma estos repositorios de conocimiento con el fin de generar una mejor versión de todo el conjunto de aportes como la metodología, las prácticas y el perfil de calidad.

- El contexto de arquitecturas es muy amplio lo que permitió aprender acerca de arquitecturas de referencia, tácticas, patrones, perfiles de calidad, atributos de calidad y modelos empleados para el diseño de arquitecturas.

Investigación

- A pesar que se obtuvo una buena percepción de comprensibilidad y utilidad todavía existen aspectos donde se puede enriquecer la presente propuesta como la mejora continua del perfil de calidad.

- La encuesta puede mejorar en cuestión de hacerla un poco más liviana para los evaluadores con el fin de hacerla un poco más práctica y sencilla.

- El diseño del estudio de caso puede mejorar incluyendo más expertos que puedan evaluar la propuesta con el fin de tener un panorama más amplio con respecto a las opiniones y apreciaciones de los mismos.

- La ejecución del estudio de caso puede mejorar ampliando la ventana de tiempo para realizar las diferentes actividades cuando se diseña una arquitectura de software basada en IoT.

- Se deben diseñar artefactos que permitan la participación activa de los participantes incluyendo preguntas abiertas, foros, discusiones con el objetivo de conocer ideas que puedan aportar y apoyar de una mejor forma la presente propuesta.

7. BIBLIOGRAFÍA

[1] L. Petrović, I. Jezdović, D. Stojanović, Z. Bogdanović, and M. Despotović-Zrakić, "Development of an educational game based on IoT," *Ijeec - Int. J. Electr. Eng. Comput.*, vol. 1, no. 1, 2017, doi: 10.7251/ijeec1701036p.

[2] J. Kiruthika and S. Khaddaj, "Software quality issues and challenges of internet of things," *Proc. - 14th Int. Symp. Distrib. Comput. Appl. Business, Eng. Sci. DCABES 2015*, pp. 176–179, 2016, doi: 10.1109/DCABES.2015.51.

[3] J. J. C. Tambotoh, S. M. Isa, F. L. Gaol, B. Soewito, and H. L. H. S. Warnars, "Software quality model for Internet of Things governance," *Proc. 2016 Int. Conf. Data Softw. Eng. ICoDSE 2016*, 2017, doi: 10.1109/ICODSE.2016.7936138.

[4] T. R and B. A, "Quality assurance of iot based home automation application using modified ISO/IEC 25010," *Int. J. Eng. Trends Technol.*, 2021.

[5] X. Larrucea, A. Combelles, J. Favaro, and K. Taneja, "Software Engineering for the Internet of Things House Advertisement," *IEEE Softw.*, vol. 33, no. 2, pp. c3–c3, 2016, doi: 10.1109/ms.2016.50.

[6] K. Ojo-Gonzalez and B. Bonilla-Morales, "Requerimientos no funcionales para sistemas basados en el Internet de las cosas (IoT): Una revisión," *I+D Tecnológico*, vol. 17, no. 2, 2021, doi: 10.33412/idt.v17.2.3303.

[7] iso25000, "ISO 25000," 2021. iso25000.com.

[8] M. Kim, "A Quality Model for Evaluating IoT Applications," *Int. J. Comput. Electr. Eng.*, vol. 8, no. 1, pp. 66–76, 2016, doi: 10.17706/ijcee.2016.8.1.66-76.

[9] M. Sirshar, M. Khan, K. Naeem, and T. Akbar, "Software Quality Assurance testing methodologies in IoT," no. December 2019, 2019, [Online]. Available:

www.preprints.org.
[10] M. Bures, T. Cerny, and B. S. Ahmed, “Internet of things: Current challenges in the quality assurance and testing methods,” *Lect. Notes Electr. Eng.*, vol. 514, pp. 625–634, 2019, doi: 10.1007/978-981-13-1056-0_61.
[11] R. Arakaki, V. T. Hayashi, and W. V. Ruggiero, “Available and Fault Tolerant IoT System: Applying Quality Engineering Method,” *2nd Int. Conf. Electr. Commun. Comput. Eng. ICECCE 2020*, no. June, pp. 12–13, 2020, doi: 10.1109/ICECCE49384.2020.9179341.
[12] B. S. Ahmed, M. Bures, K. Frajtak, and T. Cerny, “Aspects of Quality in Internet of Things (IoT) Solutions: A Systematic Mapping Study,” *IEEE Access*, vol. 7, no. c, pp. 13758–13780, 2019, doi: 10.1109/ACCESS.2019.2893493.
[13] M. Marwah, Q. Mateen, and M. Sirshar, “Software Quality Assurance in Internet of Things,” *Int. J. Comput. Appl.*, vol. 109, no. 9, pp. 16–24, 2015, doi: 10.5120/19217-0964.
[14] I. A. Baños, “Metodologia para Evaluar la Calidad de un Producto Software de una Implementacion de Internet de las Cosas,” 2017.
[15] H. Foidl and M. Felderer, “Data science challenges to improve quality assurance of internet of things applications,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9953 LNCS, pp. 707–726, 2016, doi: 10.1007/978-3-319-47169-3_54.
[16] M. Abdallah, T. Jaber, N. Alabwaini, and A. A. Alnabi, “A Proposed Quality Model for the Internet of Things Systems,” *2019 IEEE Jordan Int. Jt. Conf. Electr. Eng. Inf. Technol. JEEIT 2019 - Proc.*, pp. 23–27, 2019, doi: 10.1109/JEEIT.2019.8717516.
[17] H. Washizaki, S. Ogata, A. Hazeyama, T. Okubo, E. B. Fernandez, and N. Yoshioka, “Landscape of Architecture and Design Patterns for IoT Systems,” *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10091–10101, 2020, doi: 10.1109/JIOT.2020.3003528.
[18] A. Alreshidi and A. Ahmad, “Architecting software for the Internet of Thing based systems,” *Futur. Internet*, vol. 11, no. 7, 2019, doi: 10.3390/fi11070153.
[19] S. Moaven and J. Habibi, “A fuzzy-AHP-based approach to select software architecture based on quality attributes (FASSA),” *Knowl. Inf. Syst.*, vol. 62, no. 12, pp. 4569–4597, 2020, doi: 10.1007/s10115-020-01496-7.
[20] Z. Stojanov and D. Dobrilovic, “Software architecture quality attributes of a layered sensor-based IoT system,” *CEUR Workshop Proc.*, vol. 2984, pp. 66–74, 2021.
[21] A. Finkelstein and J. Kramer, “The Future of Software Engineering,” *ACM*, 2000, Accessed: Nov. 09, 2018. [Online]. Available: http://www0.cs.ucl.ac.uk/staff/A.Finkelstein/fose/future.html.
[22] B. Kitchenham, “Procedures for Performing Systematic Reviews,” 2004.
[23] L. K. Kai Petersen, Sairam Vakkalanka, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Inf. Softw. Technol.*, 2015, doi: http://dx.doi.org/10.1016/j.infsof.2015.03.007.
[24] R. H. Sampieri, C. F. Collado, and M. del P. B. Lucio, *Metodologia de la investigacion*, vol. 136, no. 1. 2014.
[25] P. Runeson and M. Höst, “Guidelines for conducting and reporting case study research in software engineering,” *Empir. Softw. Eng.*, vol. 14, no. 2, pp. 131–164, 2009, doi: 10.1007/s10664-008-9102-8.
[26] R. B. Johnson and A. J. Onwuegbuzie, “Mixed Methods Research: A Research Paradigm Whose Time Has Come,” *Educ. Res.*, vol. 33, no. 7, pp. 14–26, 2004, doi: 10.3102/0013189X033007014.
[27] J. E. McLaughlin, A. A. Bush, and J. M. Zeeman, “Mixed methods: Expanding research methodologies in pharmacy education,” *Curr. Pharm. Teach. Learn.*, vol. 8, no. 5, pp. 715–721, 2016, doi: 10.1016/j.cptl.2016.06.015.

[28] N. B. Harrison and P. Avgeriou, “How do architecture patterns and tactics interact? A model and annotation,” *J. Syst. Softw.*, vol. 83, no. 10, pp. 1735–1758, 2010, doi: 10.1016/j.jss.2010.04.067.
[29] P. Kruchten, H. Obbink, and J. Stafford, “The past, present, and future for software architecture,” *IEEE Softw.*, vol. 23, no. 2, p. 22, 2006, doi: 10.1109/MS.2006.59.
[30] M. J. Sanchez Grueso and J. A. Hurtado Alegría, “Un lenguaje de modelado para representar visualmente las decisiones de diseño arquitectónico y su rationale: Rationale,” *Inf. Técnico*, vol. 84, no. 2, pp. 155–174, 2020, doi: 10.23850/22565035.2622.
[31] M. L. Roldán, S. Gonnet, and H. Leone, “Operation-based approach for documenting software architecture knowledge,” *Expert Syst.*, vol. 33, no. 4, pp. 313–348, 2016, doi: 10.1111/exsy.12152.
[32] A. Jansen and J. Bosch, “Software architecture as a set of architectural design decisions,” *Proc. - 5th Work. IEEE/IFIP Conf. Softw. Archit. WICSA 2005*, vol. 2005, pp. 109–120, 2005, doi: 10.1109/WICSA.2005.61.
[33] W. G. W. Barbacci, Mario R., Robert Ellison, Anthony J. Lattanze, Judith A. Stafford, Charles B. Weinstock, “Workshops (QAWs), Quality Attribute Third Edition,” *Carnegie Mellon Softw. Eng. Inst.*, 2003.
[34] T. De Gooijer, “Discover quality requirements with the mini-QAW,” *Proc. - 2017 IEEE Int. Conf. Softw. Archit. Work. ICSAW 2017 Side Track Proc.*, pp. 196–198, 2017, doi: 10.1109/ICSAW.2017.52.
[35] R. Wojcik, P. Clements, F. Bachmann, P. Merson, R. L. Nord, and B. Wood, “Attribute-Driven Design (ADD),” *SEI Adm. Agent*, vol. 2, no. November, p. 55, 2006.
[36] P. Clements and L. Bass, “Using business goals to inform software architecture,” *Proc. 2010 18th IEEE Int. Requir. Eng. Conf. RE2010*, pp. 69–78, 2010, doi: 10.1109/RE.2010.18.
[37] R. Kazman, M. Klein, and P. Clements, “ATAM : Method for Architecture Evaluation,” *Cmusei*, vol. 4, no. August, p. 83, 2000, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.6014&rep=rep1&type=pdf.
[38] F. de M. Hernández Pérez and J. A. Hurtado Algeria, “Difficulties and challenges in the incorporation of architectural practices,” *Sist. y Telemática*, vol. 14, no. 38, pp. 74–86, 2016, doi: 10.18046/syt.v14i38.2290.
[39] P. Kruchten, “The 4 + 1 View of Software Architecture,” *IEEE Softw.*, vol. 12, no. 6, pp. 42–50, 1995.
[40] J. A. Zachman, “Framework for information systems architecture,” *IBM Syst. J.*, vol. 38, no. 2, pp. 454–470, 1999, doi: 10.1147/sj.382.0454.
[41] R. Kazman, “Tool support for architecture analysis and design,” *Int. Softw. Archit. Work. Proceedings, ISAW*, pp. 94–97, 1996, doi: 10.1145/243327.243618.
[42] W. Rafique, X. Zhao, S. Yu, I. Yaqoob, M. Imran, and W. Dou, “An Application Development Framework for Internet-of-Things Service Orchestration,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4543–4556, 2020, doi: 10.1109/JIOT.2020.2971013.
[43] C. Verdouw, H. Sundmaeker, B. Tekinerdogan, D. Conzon, and T. Montanaro, “Architecture framework of IoT-based food and farm systems: A multiple case study,” *Comput. Electron. Agric.*, vol. 165, no. April, p. 104939, 2019, doi: 10.1016/j.compag.2019.104939.
[44] W. Aman and E. Snekkenes, “Managing security trade-offs in the Internet of Things using adaptive security,” *2015 10th Int. Conf. Internet Technol. Secur. Trans. ICITST 2015*, pp. 362–368, 2016, doi: 10.1109/ICITST.2015.7412122.
[45] V. K. Quy *et al.*, “IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges,” *Appl. Sci.*, vol. 12, no. 7, 2022, doi: 10.3390/app12073396.

[46] P. P. Ray, “Internet of things for smart agriculture: Technologies, practices and future direction,” *J. Ambient Intell. Smart Environ.*, vol. 9, no. 4, pp. 395–420, 2017, doi: 10.3233/AIS-170440.

[47] A. Sinha, G. Shrivastava, and P. Kumar, “Architecting user-centric internet of things for smart agriculture,” *Sustain. Comput. Informatics Syst.*, vol. 23, pp. 88–102, 2019, doi: 10.1016/j.suscom.2019.07.001.

[48] J. C. Guillermo, A. García-Cedeño, D. Rivas-Lalaleo, M. Huerta, and R. Clotet, “IoT Architecture Based on Wireless Sensor Network Applied to Agricultural Monitoring: A Case of Study of Cacao Crops in Ecuador,” *Adv. Intell. Syst. Comput.*, vol. 893, pp. 42–57, 2019, doi: 10.1007/978-3-030-04447-3_3.

[49] P. M. Gupta, M. Salpekar, and P. K. Tejan, “Agricultural practices Improvement Using IoT Enabled SMART Sensors,” *2018 Int. Conf. Smart City Emerg. Technol. ICSCET 2018*, no. I, pp. 1–5, 2018, doi: 10.1109/ICSCET.2018.8537291.

[50] S. Verma, R. Gala, S. Madhavan, S. Burkule, S. Chauhan, and C. Prakash, “An Internet of Things (IoT) Architecture for Smart Agriculture,” *Proc. - 2018 4th Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2018*, pp. 1–4, 2018, doi: 10.1109/ICCUBEA.2018.8697707.

[51] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, “Requirements engineering paper classification and evaluation criteria: a proposal and a discussion,” *Requir. Eng.*, doi: https://doi.org/10.1007/s00766-005-0021-6.

[52] M. Zarour, A. Abran, J.-M. Desharnais, and A. Abdulrahman, “An investigation into the best practices for the successful design and implementation of lightweight software process assessment methods: A systematic literature review,” *J. Syst. Softw.*, 2014, doi: https://doi.org/10.1016/j.jss.2014.11.041.

[53] D. Navani, S. Jain, and M. S. Nehra, “The internet of things (IoT): A study of architectural elements,” *Proc. - 13th Int. Conf. Signal-Image Technol. Internet-Based Syst. SITIS 2017*, vol. 2018-Janua, pp. 473–478, 2018, doi: 10.1109/SITIS.2017.83.

[54] Object Management Group, “Kernel and Language for Software Engineering Methods (Essence),” *2007 4th IEEE Int. Work. Vis. Softw. Underst. Anal.*, no. Versión 1.2, p. 300, 2018, [Online]. Available: https://www.omg.org/spec/Essence/1.2.

[55] N. Harrison and P. Avgeriou, “Pattern-driven architectural partitioning balancing functional and non-functional requirements,” *2007 Second Int. Conf. Digit. Telecommun. ICDT’07*, pp. 21–26, 2007, doi: 10.1109/ICDT.2007.65.

[56] W. Yánez, R. Bahsoon, Y. Zhang, and R. Kazman, *Architecting Internet of Things Systems with Blockchain*, vol. 30, no. 3. 2021.

[57] N. B. Harrison and P. Avgeriou, “Implementing reliability: The interaction of requirements, tactics and architecture patterns,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6420 LNCS, pp. 97–122, 2010, doi: 10.1007/978-3-642-17245-8_5.

[58] H. Muccini and M. T. Moghaddam, *IoT architectural styles: A systematic mapping study*, vol. 11048 LNCS. Springer International Publishing, 2018.

[59] S. A. Halim, D. N. A. Jawawi, N. Ibrahim, M. Z. M. Zaki, and S. Deris, “Multi attribute architecture design decision for core asset derivation,” *J. Teknol.*, vol. 77, no. 9, pp. 75–87, 2015, doi: 10.11113/jt.v77.6187.

[60] G. Márquez, H. Astudillo, and R. Kazman, “Architectural tactics in software architecture: A systematic mapping study,” *J. Syst. Softw.*, vol. 197, 2023, doi: 10.1016/j.jss.2022.111558.

[61] L. Bass, P. Clements, R. Kazman, and an O. M. C. Safari, “Software Architecture in Practice, 4th Edition,” p. 464, 2021, [Online]. Available: https://learning-oreilly-com.libezproxy.open.ac.uk/library/view/software-architecture-in/9780136885979/ch01.xhtml.

[62] Oscar Blancarte, “Arquitectura en Capas,” 2021. https://reactiveprogramming.io/blog/es/estilos-arquitectonicos/capas (accessed May 29, 2023).
[63] EcuRed, “Arquitectura de capas en sistemas de información,” *2023*. https://www.ecured.cu/Arquitectura_de_capas_en_sistemas_de_información (accessed May 29, 2023).
[64] F. D. Davis, “Perceived usefulness, perceived ease of use, and user acceptance of information technology,” *MIS Q. Manag. Inf. Syst.*, vol. 13, no. 3, pp. 319–339, 1989, doi: 10.2307/249008.
[65] Platzi, “Patron en capas,” *Platzi*, 2023. https://platzi.com/tutoriales/1248-pro-arquitectura/5439-patron-arquitectonico-de-capas-layers/ (accessed May 26, 2023).

ANEXO A

Ejemplo de aplicación prácticas IoTAP Versión 1

Ejemplo de aplicación de práctica de diseño y refactorización para la definición del perfil de calidad

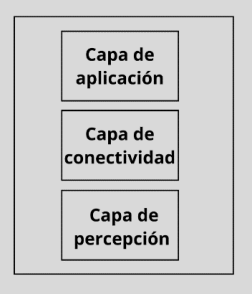
Paso 1: Identificar los atributos de calidad que serán drivers del diseño de la arquitectura de la solución basada en IoT integrando los diferentes involucrados.

En este caso, una empresa requiere medir la humedad de la tierra y su temperatura para poder plantar café en sus territorios. La temperatura y humedad de la tierra tienen un sensor, un umbral y un actuador asociado. En caso de no cumplirse dicho umbral el actuador avisa de lo que está sucediendo en los territorios y notifica la información obtenida del contexto. El dominio seleccionado para el presente ejemplo es el Smart Farming o agricultura inteligente en español de acuerdo al contexto planteado. Por ello se hace necesario aportar en atributos de calidad como: **(i) mantenibilidad, (ii) confiablidad, (iii) seguridad, (iv) usabilidad, (v) eficiencia de desempeño y (vi) compatibilidad.**

Paso 2: Selección de los patrones y tácticas de arquitectura relevantes para el perfil de calidad establecido con base a los drivers identificados de acuerdo al dominio de aplicación escogido.

El patrón de arquitectura que se escoge es el de capas debido a que es un patrón muy recurrente en este tipo de arquitecturas y en su dominio de aplicación, en los cuales se separan las aplicaciones de la conectividad y de la percepción.

Arquitectura Básica para IoT



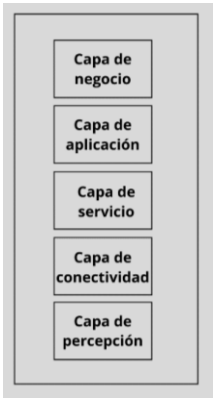
- Capa de aplicación: Es la capa responsable de proporcionar los servicios inteligentes de IoT a los usuarios y se enfatiza en las necesidades a nivel de atributos de calidad que se requieren para un producto software basado en IoT.

- Capa de conectividad: Es la capa responsable de procesar y transmitir los datos que se reciben de la capa de percepción.

- Capa de percepción: Es la capa que cuenta con los dispositivos como sensores, actuadores y dispositivos. Su tarea es percibir, recolectar y recopilar datos e información del contexto circundante.

Paso 3: Para cada patrón (view) o táctica instanciarlos para establecer la solución considerando el dominio (servicios, redes, actuadores, sensores y dispositivos) que se asociaran para cumplir una tarea determinada.

Los sensores, actuadores, tarjetas que se escogen se hace con base en que se requiere medir.



- Capa de negocio: Es la capa usada por los usuarios en la cual se llaman a las funcionalidades desde la capa de aplicación al usuario final.

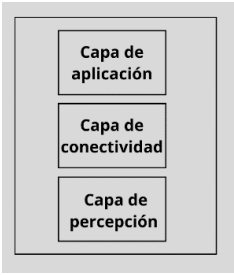
- Capa de aplicación: Es la capa responsable de proporcionar los servicios inteligentes de IoT a los usuarios y se enfatiza en las necesidades a nivel de atributos de calidad que se requieren para un producto software basado en IoT.

- Capa de servicio: Se encarga de analizar y procesar los datos que provienen de la capa de conectividad. También es llamada capa de middleware debido a que contribuye con el intercambio de información entre objetos heterogéneos que no cuentan con requisitos específicos de software y hardware.

- Capa de conectividad: Es la capa responsable de procesar y transmitir los datos que se reciben de la capa de percepción. Cuenta con diferentes tipos de redes como WiFi, móvil, etc.

- Capa de percepción: Esta capa se encarga de percibir, detectar, recolectar y medir información del contexto circundante, por ejemplo: temperatura, humedad, etc.

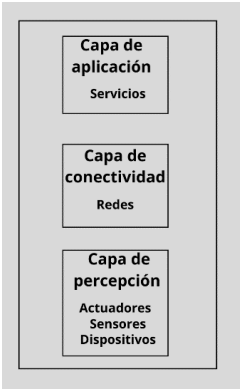
Paso 4: Adicionar responsabilidades a las partes del sistema. Es decir, facilitar la identificación de las tareas que debe realizar cada de ellas, especificando comportamientos y facilitando su entendimiento.

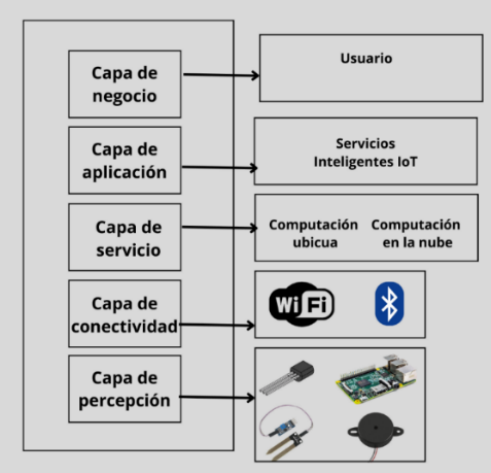


Se identifican los actores asociados por cada capa:

- Capa de aplicación: Servicios
- Capa de conectividad: Redes
- Capa de percepción: Actuadores, sensores y dispositivos

Una vez se hace esta clasificación general de que se utilizara en el producto, se obtiene el siguiente resultado:





- Capa de negocio: Utilizar el servicio anteriormente mencionado con el fin de saber el estado de la tierra.

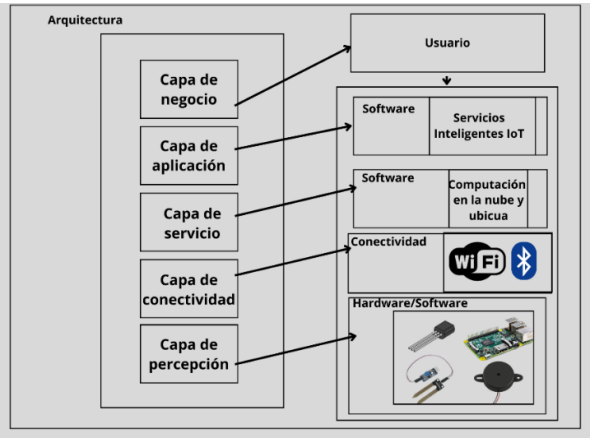
- Capa de aplicación: Servicio que permita monitorear la humedad y temperatura de la tierra detectando un umbral definido y reaccionado con base en él.

- Capa de servicio: Combinación entre computación ubicua y computación en la nube (Borde).

- Capa de conectividad: Se deben utilizar tecnologías como WiFi y Bluetooth como requerimiento del cliente.

- Capa de percepción: Medir la humedad de la tierra y su temperatura conectadas a una Raspberry Pi 3, cuando cualesquiera de las dos estén por debajo del umbral hacer sonar un buzzer.

Paso 5: Definir interfaces de los módulos, componentes y capas con el fin de mejorar en la abstracción de los componentes de software y hardware heterogéneos, definiendo la participación del software y hardware, estableciendo un modelo de participación, seleccionando cuales sensores, actuadores, tarjetas serán utilizados dependiendo de lo que se quiera medir.



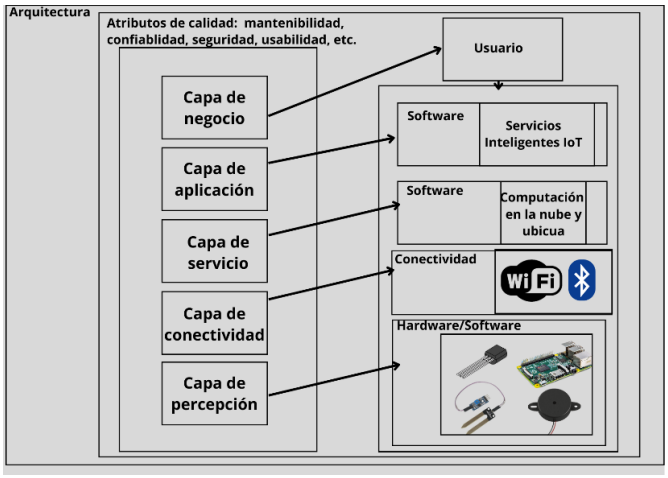
Paso 6: Se documenta todo el conocimiento adquirido a través de la selección de patrones y tácticas arquitecturales al igual que las lecciones aprendidas al aplicar la presente práctica.

Información aportada por el catálogo a través de la selección de patrones y tácticas

Uno de los objetivos del sector agrícola es incrementar la eficiencia y la eficacia en la productividad mediante la utilización adecuada de los recursos contribuyendo a que el impacto ambiental no se tan alto, optimizando y mejorando estos procesos agropecuarios monitoreando condiciones ambientales, contextuales y del suelo. De acuerdo a los drivers identificados en la práctica de diseño: (i) mantenibilidad, (ii) confiablidad, (iii) seguridad, (iv) usabilidad, (v) eficiencia de desempeño y (vi) compatibilidad es importante.

Con respecto a la primera es necesario tener en cuenta temas como la facilidad de programación, así como ayudando a que el producto software pueda ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. En la segunda, se debe abordar temas como tolerancia a fallos y disponibilidad, en la tercera administración de privilegios, en la cuarta interfaces intuitivas y retroalimentación, en la quinta topología de red, ubicación de sensores/actuadores, consumo energético y autonomía. Y por último en la sexta, sinergia entre dispositivos.

Cuando ya se identifiquen las partes y los bloques por cada capa, es necesario identificar las limitaciones, necesidades, preocupaciones y riesgos que poseen estos mismos en función al dominio escogido, esto por medio de ATAM, con el fin de evaluar y/o detectar, de manera temprana, riesgos técnicos, conflictos entre atributos, puntos clave del diseño y soluciones.



Ejemplo de aplicación prácticas IoTAP Versión 2

Motivación

Una práctica es un enfoque repetible para hacer algo con un objetivo específico en mente. Una práctica describe cómo manejar un aspecto específico de un esfuerzo de ingeniería de software, incluidas las descripciones de todos los elementos relevantes necesarios para expresar la guía de trabajo deseada que se requiere para lograr el propósito de la práctica. Como la arquitectura de software se tiene en cuenta en las etapas tempranas del ciclo de desarrollo software, es importante determinar la arquitectura correcta, cumpliendo los requisitos de calidad planteados y representándolos como escenarios concretos de atributos de calidad con el objetivo de contribuir en el éxito de cualquier proyecto.

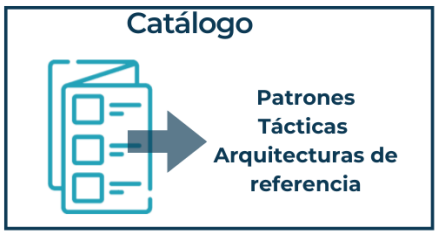
Ciclo del Método general

Contexto

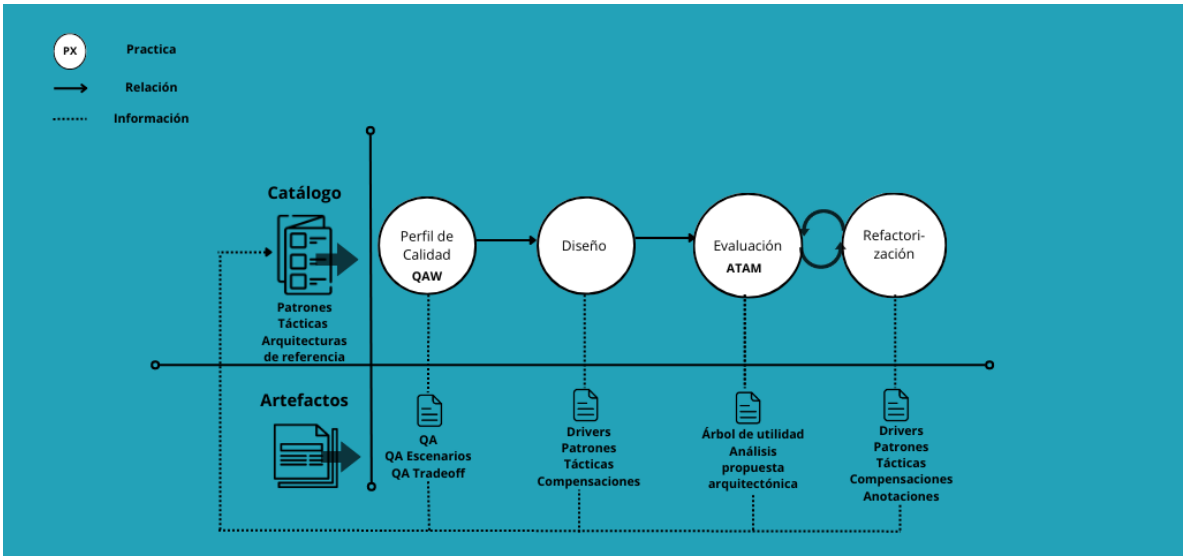
Uno de los objetivos del sector agrícola es incrementar la eficiencia y la eficacia en la productividad mediante la utilización adecuada de los recursos contribuyendo a que el impacto ambiental no se tan alto, optimizando y mejorando estos procesos agropecuarios monitoreando condiciones ambientales, contextuales y del suelo.

Por ello, una empresa requiere medir la humedad de la tierra y su temperatura para poder plantar café en sus territorios. La temperatura y humedad de la tierra tienen un sensor, un umbral y un actuador asociado. En caso de no cumplirse dicho umbral el actuador avisa de lo que está sucediendo en los territorios y notifica la información obtenida del contexto.

Aplicación del método



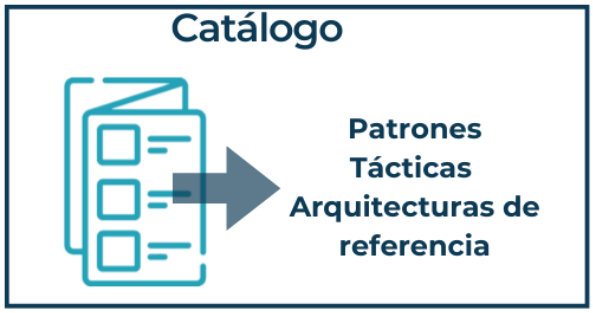
Catálogo: Estructura de almacenamiento donde se encuentra el conocimiento adquirido por medio de la búsqueda de referentes que contengan patrones, tácticas y arquitecturas de referencia para su uso en la presente metodología.



Perfil de calidad: Estructura ordenada donde se encuentra ordenada la metodología por actividades (Etapas) permitiendo llevar un orden por pasos y traza de la presente metodología.

¿Qué atributos se abordan?

El dominio seleccionado es Smart Farming o agricultura inteligente de acuerdo al contexto presentado.



Los drivers seleccionados para el dominio en cuestión son: **(i) mantenibilidad, (ii) confiablidad, (iii) seguridad, (iv) usabilidad, (v) eficiencia de desempeño y (vi) compatibilidad.** También se incluye **la sensibilidad al entorno** como driver ya que su presencia en las aplicaciones basadas en IoT es de suma importancia ya que ayuda en la percepción de la información que se encuentra en el contexto y también actúa con base en esa información.

Estos drivers y patrones son consultados en los referentes [1], [2], [3] y [4] que hacen parte del catálogo de información encontrados para esta investigación**.**

Se realiza la selección del patrón de arquitectura de software mediante la comparación entre los patrones más recurrentes en arquitecturas para IoT y los atributos de calidad (Drivers) previamente seleccionados con el fin de descubrir que patrón de arquitectura se ajusta más a mi necesidad y cómo impactan dichos drivers en la misma. Se realiza la calificación teniendo en cuenta que el valor más alto es 2 y el menor -2. A partir de los valores se generan algunos gráficos que me permiten determinar cuál es la arquitectura más indicada para las necesidades planteadas y su relevancia con los drivers.

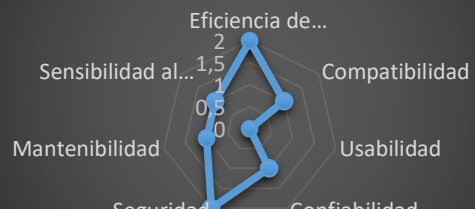
Item	Puntaje
**	2
*	1
/	0
~	-1
~~	-2

Aplicación practica de diseño IoTAP Dominio Agricultura							
Patron/Atributo	Eficiencia de desempeño	Compatibilidad	Usabilidad	Confiabilidad	Seguridad	Mantenibilidad	Sensibilidad al entorno
Capas	**	**	/	*	**	**	**
Basado en la nube	**	*	/	*	**	*	*
Orientado a servicios	**	**	/	**	*	**	**
Microservicios	**	*	/	*	*	*	**
Restful	*	*	/	*	*	**	*
Publicador/Suscriptor	**	**	/	*	*	**	*
Centrado en la informacion	*	*	/	**	*	*	*

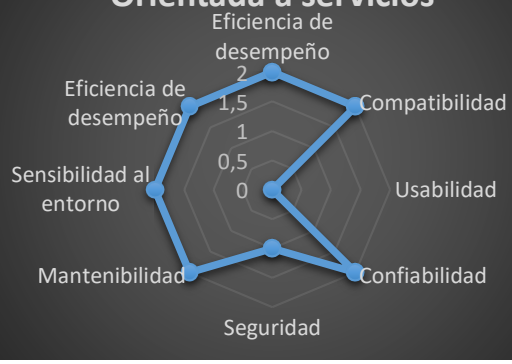
Aplicación practica de diseño IoTAP Dominio Agricultura							
Patron/Atributo	Eficiencia de desempeño	Compatibilidad	Usabilidad	Confiabilidad	Seguridad	Mantenibilidad	Sensibilidad al entorno
Capas	2	2	0	1	2	2	2
Basado en la nube	2	1	0	1	2	1	1
Orientado a servicios	2	2	0	2	1	2	2
Microservicios	2	1	0	1	1	1	2
Restful	1	1	0	1	1	2	1
Publicador/Suscriptor	2	2	0	1	1	2	1
Centrado en la informacion	1	1	0	2	1	1	1

Capas

Basado en la nube

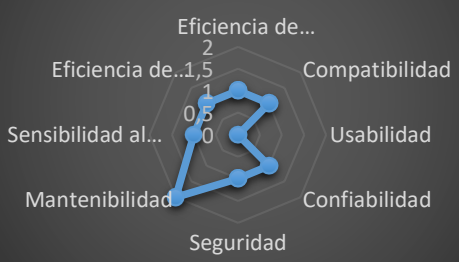


Orientada a servicios



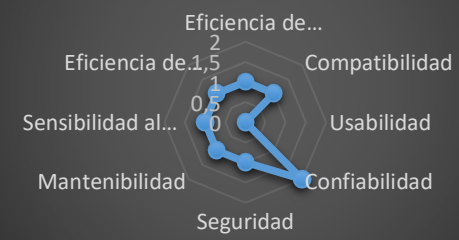
Microservicios

Restful



Publicador/Suscriptor

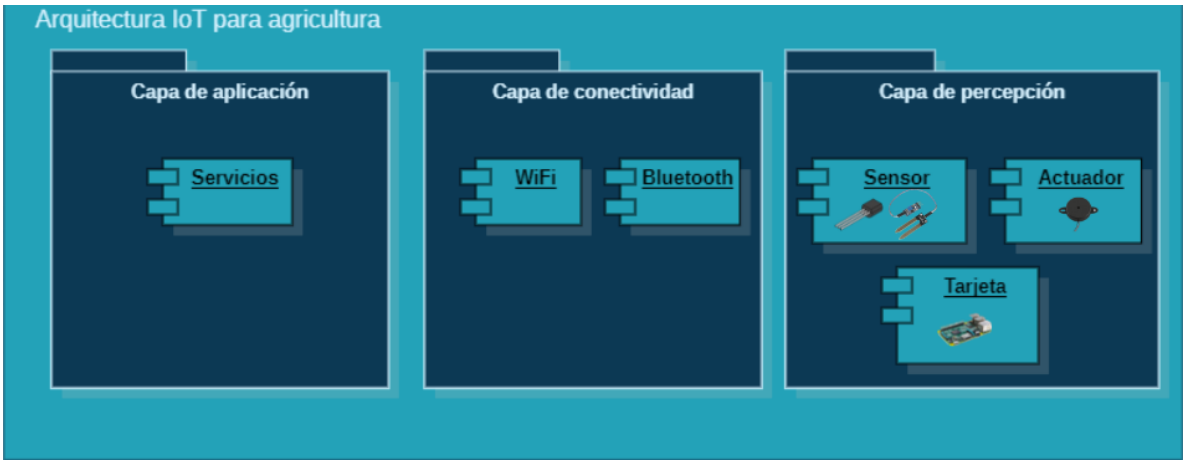
Centrado en la información



Patrones

¿Qué patrón se debería usar?

El patrón de arquitectura que se escoge es el de **capas** debido a que es un patrón muy recurrente en este tipo de arquitecturas y en su dominio de aplicación.

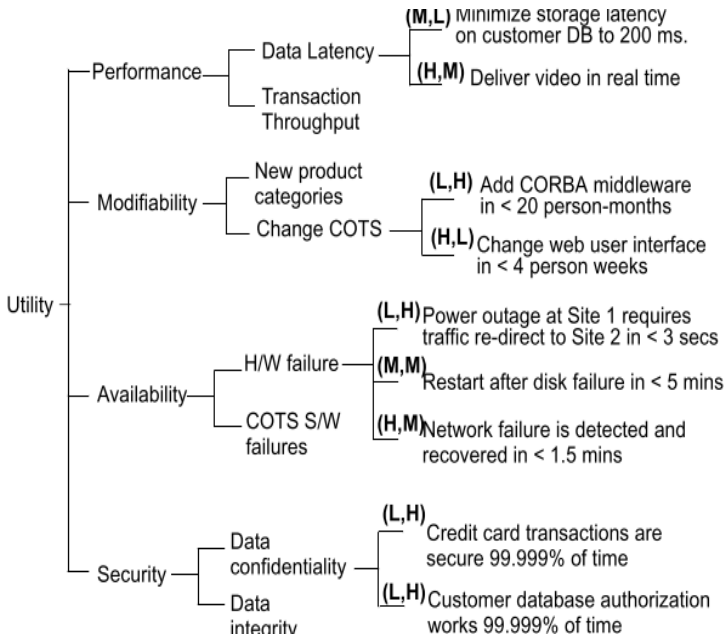


Debido a la especificidad de los sensores, actuadores y funciones **(medir temperatura y humedad)** es necesario incluir dos capas más **(negocio y servicio)** para suplir el requerimiento propuesto y dándole a cada capa su respectiva responsabilidad.

Para cada patrón **(view)** o táctica instanciarlos para establecer la solución, considerando el dominio.

¿Cómo se realiza la evaluación con ATAM?

De la evaluación con la metodología **ATAM** salen dos artefactos que son cruciales para la mejora continua de las arquitecturas y corresponden a: **(i) Árbol de Utilidad y (ii) Análisis de una propuesta arquitectónica.**



Scenario: S12 (Detect and recover from HW failure of main switch.)

Attribute: Availability

Environment: normal operations

Stimulus: CPU failure

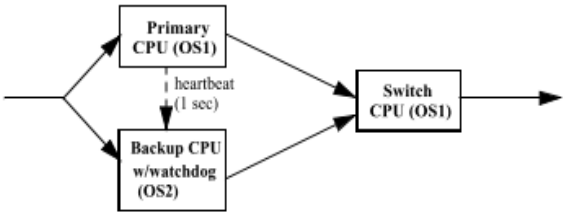
Response: 0.999999 availability of switch

Architectural decisions	**Risk**	**Sensitivity**	**Tradeoff**
Backup CPU(s)	R8	S2	
No backup Data Channel	R9	S3	T3
Watchdog		S4	
Heartbeat		S5	
Failover routing		S6	

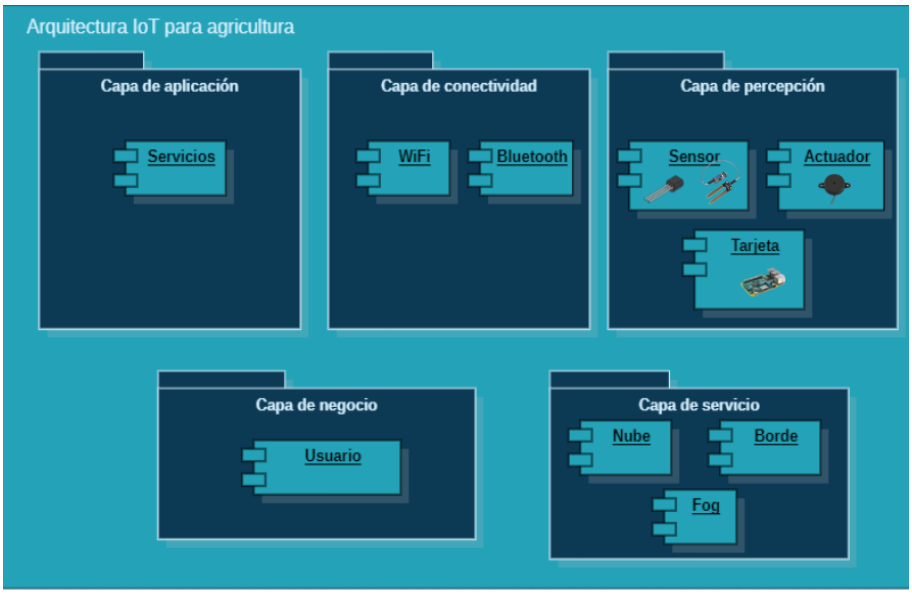
Reasoning:

- ensures no common mode failure by using different hardware and operating system (see Risk 8)
- worst-case rollover is accomplished in 4 seconds as computing state takes …
- guaranteed to detect failure with 2 seconds based on rates of heartbeat and watchdog …
- watchdog is simple and proven reliable
- availability requirement might be at risk due to lack of backup data channel … (see Risk 9)

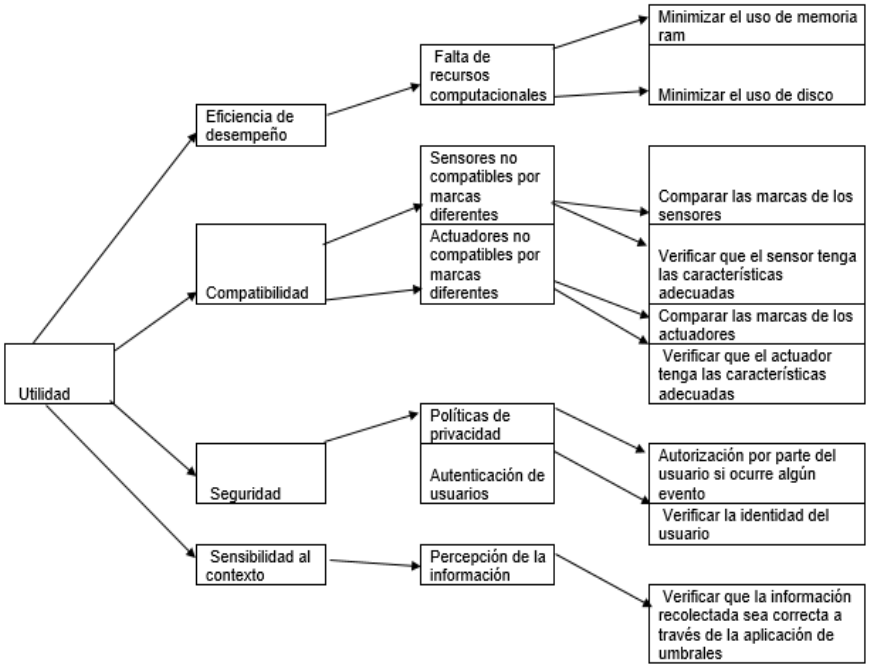
Architecture diagram:



Los atributos sensibles para la presente arquitectura son: **(i) seguridad, (ii) eficiencia de desempeño, (iii) compatibilidad y (iv) sensibilidad al contexto. Por ejemplo, el uso de WiFi y bluetooth en la capa de conectividad puede ocasionar problemas en el atributo de calidad eficiencia de desempeño. Por ello, se realizan los artefactos de ATAM para evidenciar sensibilidades y/o riesgos que puedan impactar en ella.**

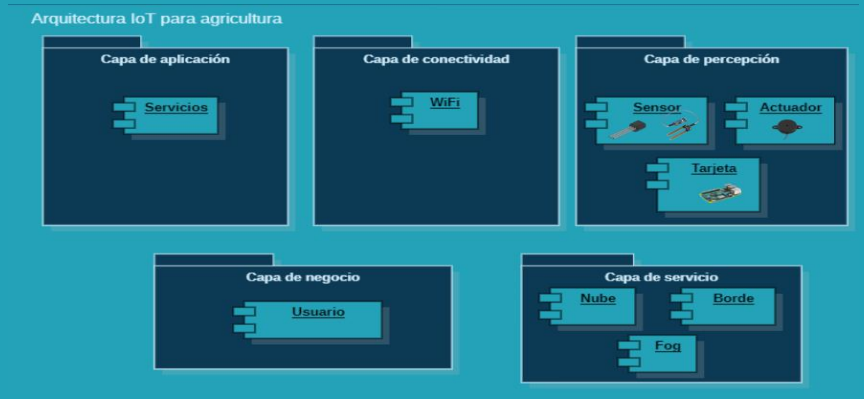


Se realiza el **árbol de utilidad** para identificar las preocupaciones y escenarios para la arquitectura, además se presentan las **sensibilidades, riesgos y en caso de existir otros QA relevantes.**

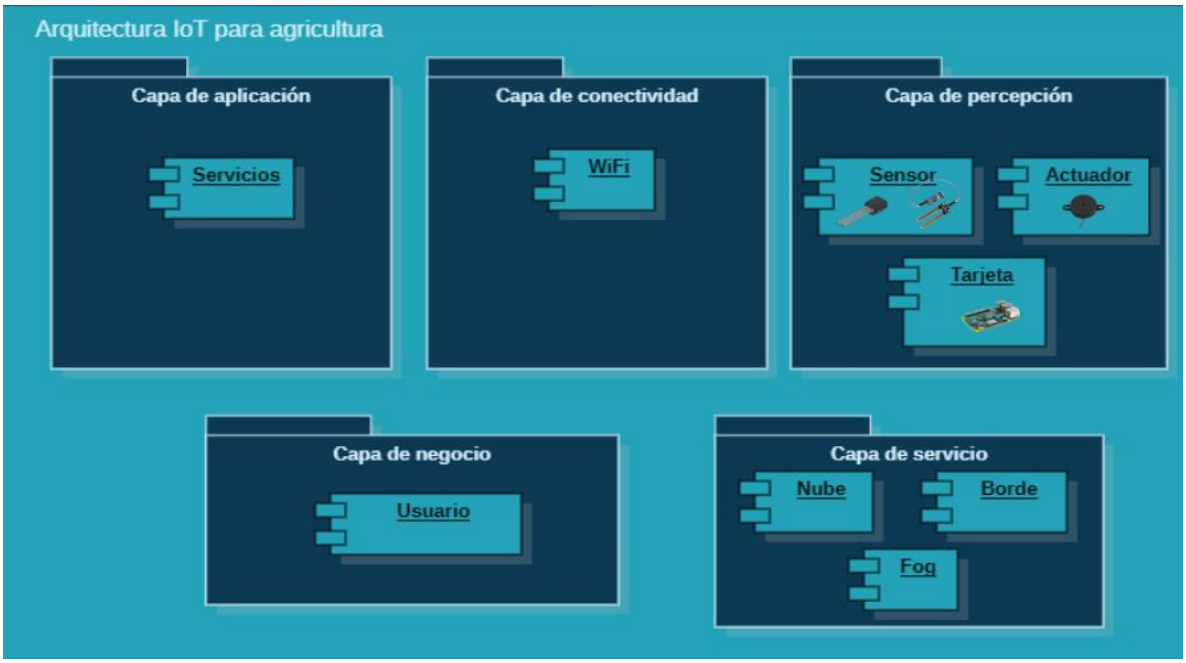


Se realiza el análisis de las **propuestas arquitectónicas** para identificar las preocupaciones y **escenarios** para la arquitectura. Se analizan los escenarios identificados por medio de la plantilla que se muestra a continuación:

Ítem	Descripción
Escenario	Minimizar el uso de memoria ram
Atributo	Eficiencia de desempeño
Ambiente	Operaciones sobre la información
Estimulo	Memoria ram limitada
Respuesta	Operaciones sobre la información incompletas
Decisiones arquitecturales (DA)	DA: 1 Utilizar la tecnología de procesamiento en el borde o en la nube para evitar el uso excesivo de memoria ram Riesgo: Alto Sensibilidad: Alta Compensación: Balancear la carga de procesamiento local hacia la nube. Limitar el uso de Bluetooth y usar solamente WiFi para la transmisión de datos. 2 3 4
Razonamiento	Se debe tener en cuenta que procesar un gran volumen de datos afecta a todo el sistema IoT ya que al carecer de buenas características se pueden llegar a fallos no deseados, por ello es importante realizar el balanceo de carga local hacia la nube limitando el uso de Bluetooth y realizando el envío de los datos por Wifi.

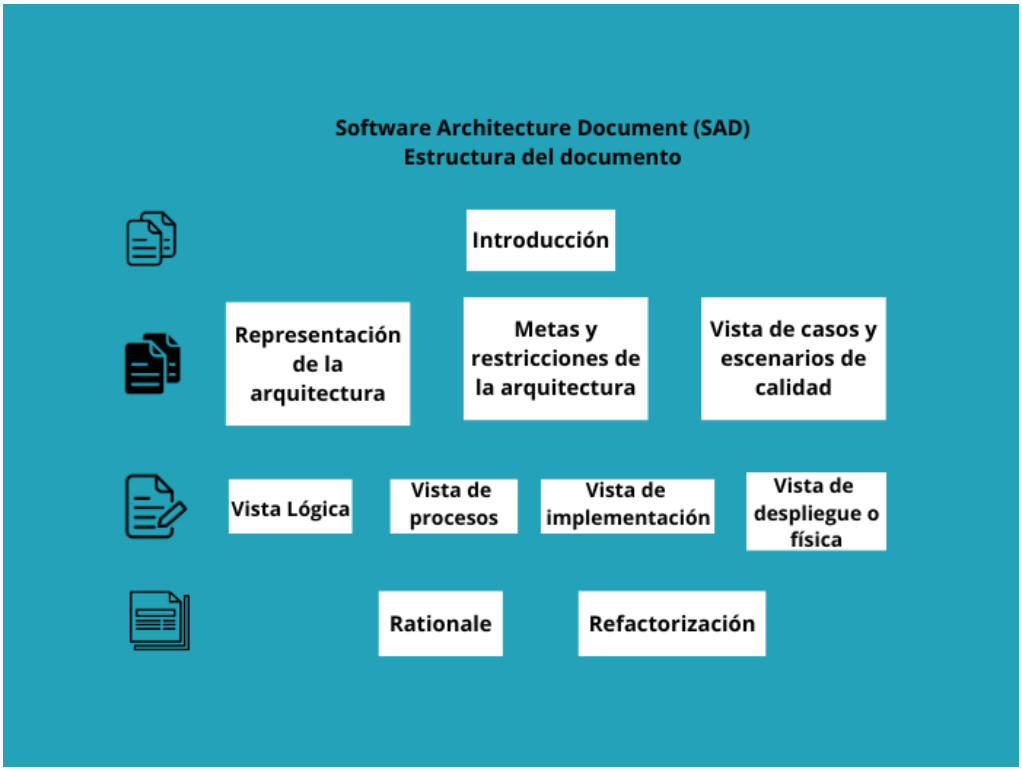


Se tienen en cuenta **las tácticas, anotaciones** para posteriormente realizar un análisis de las mismas, **actualizando o creando nuevas vistas** para la arquitectura.



Se seleccionan nuevas tácticas para compensar las **limitaciones**, **necesidades, preocupaciones y riesgos.** Se realiza el proceso de **evaluación de la interacción entre los patrones y tácticas empleadas asociado al impacto en la arquitectura tratada**. Esto debido a que implementar una táctica en una arquitectura puede impactar significativamente los patrones de arquitectura previamente seleccionados.

El producto de la aplicación del **modelo de perfil de calidad siguiendo cada actividad como Diseño, Evaluación, Refactoring** será el **documento SAD** junto con la información de la evaluación con ATAM.



ANEXO B

Encuesta

1. ¿El diseño de arquitecturas para las soluciones basadas en IoT es difícil de realizar sin el uso de las prácticas IoTAP?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

2. ¿El uso de las prácticas IoTAP propuestas me da un alcance claro sobre el diseño de las arquitecturas para las soluciones basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

3. ¿Las prácticas IoTAP propuestas abordan las necesidades del diseño de las arquitecturas para las soluciones basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

4. ¿El uso de las prácticas IoTAP propuestas ahorra tiempo cuando se diseñan las arquitecturas para las soluciones basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

5. ¿Las prácticas IoTAP propuestas permiten el diseño de las arquitecturas para las soluciones basadas en IoT más rápidamente?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

6. ¿Las prácticas IoTAP promueven aspectos que se apoyan el diseño de arquitecturas para las soluciones basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

7. ¿El uso de las prácticas IoTAP mejora la calidad del diseño de las arquitecturas para las soluciones basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

8. ¿El uso de las prácticas IoTAP facilita el diseño de las arquitecturas para las soluciones basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

9. ¿Las prácticas IoTAP propuestas son útiles para seguir los pasos requeridos para el diseño de las arquitecturas para las soluciones basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

10. ¿En general, las prácticas IoTAP me resultan útiles en el diseño de las arquitecturas de software basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

11. ¿A menudo me confundo cuando uso las prácticas IoTAP?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

12. ¿Con frecuencia cometo errores al usar las prácticas IoTAP?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

13. ¿Interactuar con las prácticas IoTAP suele ser frustrante?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

14. ¿Las prácticas IoTAP son rígidas e inflexibles para interactuar?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

15. ¿Me parece engorroso usar las prácticas IoTAP?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

16. ¿Es fácil para mí recordar cómo hacer los pasos de las prácticas IoTAP?:

    a. Totalmente en desacuerdo
    b. En desacuerdo

c. Ni de acuerdo ni en desacuerdo
d. De acuerdo
e. Totalmente de acuerdo

17. ¿Mi interacción con las actividades y artefactos de las prácticas IoTAP propuestas son fáciles de entender?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

18. ¿Las prácticas IoTAP propuestas me proporcionan una orientación útil para definir, realizar y establecer arquitecturas para las soluciones basadas en IoT?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

19. ¿En general, encuentro que las prácticas IoTAP propuestas son fáciles de usar?:

    a. Totalmente en desacuerdo
    b. En desacuerdo
    c. Ni de acuerdo ni en desacuerdo
    d. De acuerdo
    e. Totalmente de acuerdo

20. Lo positivo de las prácticas IoTAP:

    _____
    _____
    _____
    _____
    _____
    _____

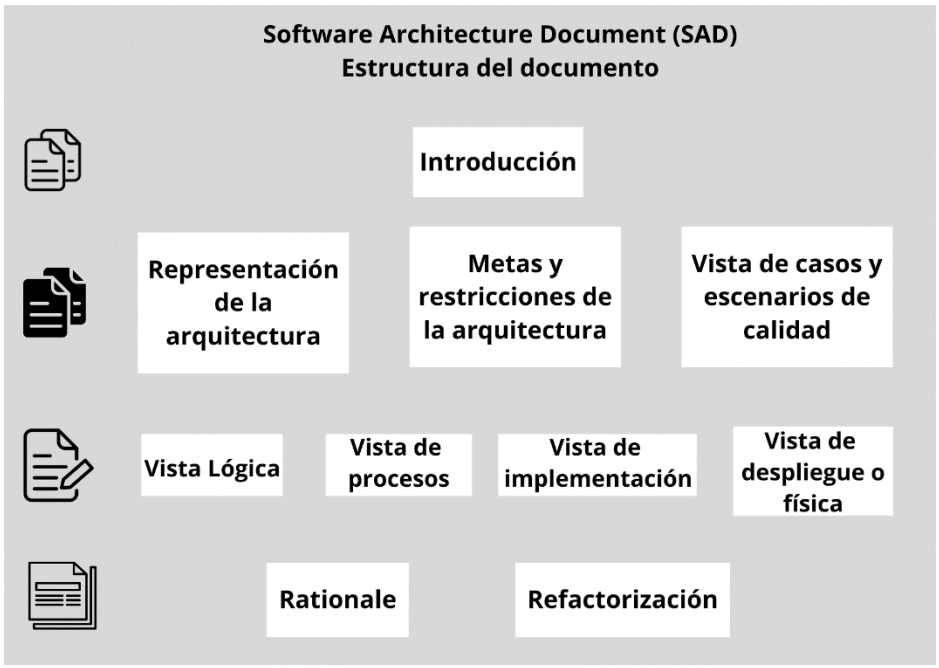
21. El negativo de las prácticas IoTAP:

    _____
    _____
    _____
    _____
    _____
    _____

Protocolo de Observación

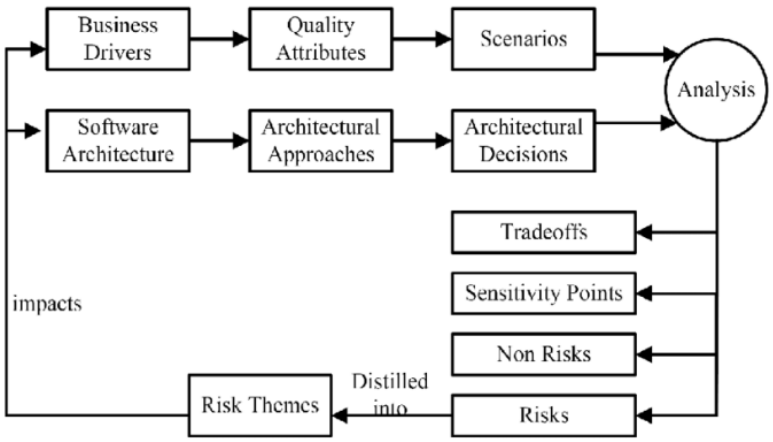
Id	Fenómeno Observado	Práctica/Tarea	Artefacto	Reflexión
#	Xxxxxxxxxxx	xxxxxxxxx		

Protocolo de evaluación del artefacto



Plantilla del Artefacto

ATAM es una metodología de evaluación que permite observar el grado de satisfacción de la arquitectura del sistema hacia los atributos de calidad, los cuales son definidos por los interesados del proyecto. Brinda un análisis iterativo y ayuda a detectar posibles falencias en una etapa temprana donde es relativamente económico corregir problemas. El enfoque de la metodología se basa en varios aspectos, el primero se obtiene de las pautas del negocio donde se realiza una recolección de escenarios que describen el comportamiento del sistema ante diferentes estímulos o circunstancias, los cuales se clasifican según al atributo de calidad al cual pertenezca; el segundo es la presentación de la arquitectura del sistema, la cual muestra la forma en la que está construido y cómo interactúan sus componentes, de este aspecto se derivan las propuestas arquitectónicas y las decisiones de diseño; por último se encuentra el análisis, donde se recopila la información del primer y segundo aspecto para evaluar el grado de satisfacción uno a uno de las propuestas arquitectónicas hacia los atributos de calidad, como resultado de este análisis tenemos, puntos de sensibilidad, riesgos, no riesgos y compensaciones [37]. En la siguiente figura se puede observar la manera en cómo interactúan los 3 aspectos mencionados:



Architecture Trade-Off Analysis Method (ATAM)
Estructura del método

Presentación	**1. Presentar el ATAM** **2. Presentar pautas del negocio** **3. Presentar la arquitectura**
Investigación y análisis	**4. Identificar propuestas arquitectónicas** **5. Generar árbol de utilidad de los atributos de calidad** **6. Analizar propuestas arquitectónicas (del paso 4)**
Pruebas	**7. Revisión del árbol de utilidad y priorización de escenarios** **8. Analizar propuestas arquitectónicas (reiterar actividades del paso 6, utilizando ranking de escenarios del paso 7)**
Informes	**9. Presentar resultados**

Pasos del método ATAM:

1. **Presentar el ATAM:** El líder del equipo de evaluación, realiza una descripción del método a los participantes, fija las expectativas y responde a las preguntas que puedan surgir.

2. **Presentar las pautas del negocio:** Un representante del proyecto describe las metas del negocio que se convertirán en las pautas primordiales de la arquitectura (principalmente en lo que refiere a atributos de calidad).

3. **Presentar la arquitectura:** El arquitecto describe la arquitectura, enfocándose principalmente en cómo la misma sigue las pautas del negocio

4. **Identificar las propuestas arquitectónicas**: Las propuestas arquitectónicas son identificadas por el arquitecto, pero no son analizadas.

5. **Generar el árbol de utilidad de los atributos de calidad:** Los atributos de calidad que comprometen la utilidad del sistema, son obtenidos especificados en escenarios (con estímulos y respuestas) y priorizados.

6. **Analizar las propuestas arquitectónicas**: Basados en los escenarios de mayor prioridad identificados en el paso 5, las propuestas arquitectónicas que cumplen con estos escenarios, son obtenidas analizadas.

7. **Revisión del árbol de utilidad**: Un gran conjunto de escenarios es obtenido por todos los interesados. Este conjunto luego debe ser priorizado.

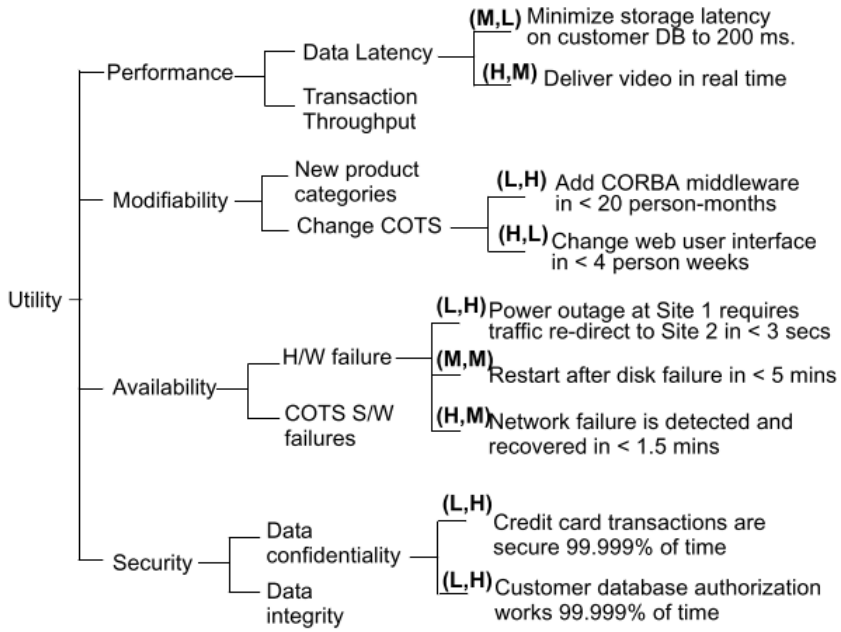
8. **Analizar las propuestas arquitectónicas**: Se reiteran las actividades del paso 6 utilizando el ranking de escenarios del paso 7. Este análisis puede llegar a descubrir nuevas propuestas arquitectónicas, riesgos, no riesgos, que deberán ser documentados.

9. **Presentar los resultados:** Basados en la información recogida durante el ATAM (propuestas, escenarios, árbol de utilidad, riesgos, no riesgos, etc.), el equipo de ATAM documenta y presenta los resultados a los interesados.

Artefactos del método ATAM

Árbol de utilidad

Para la generación de este artefacto es importante la participación del equipo de arquitectura y de los clientes representativos, se deben identificar las metas de calidad y priorizarlas de tal manera que se obtengan las más importantes para los interesados, esto se podrá observar en la siguiente figura:



Análisis de una propuesta arquitectónica

Para le generación de este artefacto se tienen en cuenta dos ámbitos, el primero es el escenario el cual describe de una forma corta la interacción de un interesado con el sistema y se analiza en 3 partes, la primera es el estímulo que explica lo que hace un interesado para interactuar con el sistema; la segunda el entorno, describe lo que pasa cuando se ejecuta el estímulo; y por último la respuesta, dice la forma en la que el sistema debería responder a través de la arquitectura al estímulo. El segundo ámbito es la propuesta arquitectónica que debería responder a ese escenario, esto se podrá observar en la siguiente figura:

Scenario: S12 (Detect and recover from HW failure of main switch.)

Attribute: Availability

Environment: normal operations

Stimulus: CPU failure

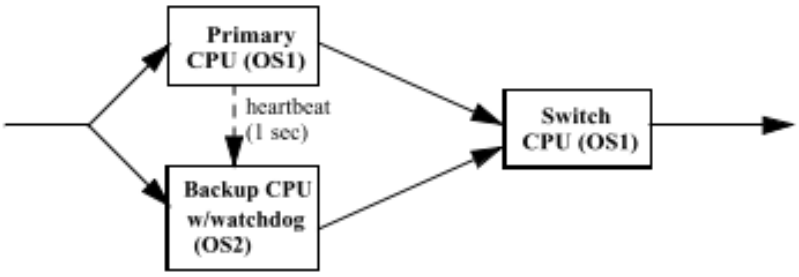
Response: 0.999999 availability of switch

Architectural decisions	**Risk**	**Sensitivity**	**Tradeoff**
Backup CPU(s)	R8	S2	
No backup Data Channel	R9	S3	T3
Watchdog		S4	
Heartbeat		S5	
Failover routing		S6	

Reasoning:

- ensures no common mode failure by using different hardware and operating system (see Risk 8)
- worst-case rollover is accomplished in 4 seconds as computing state takes …
- guaranteed to detect failure with 2 seconds based on rates of heartbeat and watchdog …
- watchdog is simple and proven reliable
- availability requirement might be at risk due to lack of backup data channel … (see Risk 9)

Architecture diagram:



ANEXO C

Resultados preguntas de la encuesta acerca de la comprensibilidad y utilidad de las prácticas IoTAP

Las preguntas 1,2,3,4,5,6,7,8 y 19 corresponden a preguntas sobre **facilidad de uso** y las preguntas 9,10,11,12,13,14,15,16,17,18 corresponden a **utilidad**.

Participante	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5
Andrés Hurtado	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
Giovanni Zambrano	De acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo
Flor Hernández	De acuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo	De acuerdo
Leandro Antonelli	De acuerdo	Totalmente en desacuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo	Ni de acuerdo ni en desacuerdo
Wilson Pantoja	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo

Participante	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10
Andrés Hurtado	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo	De acuerdo
Giovanni Zambrano	De acuerdo	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo
Flor Hernández	De acuerdo	De acuerdo	De acuerdo	De acuerdo	De acuerdo
Leandro Antonelli	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	De acuerdo
Wilson Pantoja	De acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo	Totalmente de acuerdo

Participante	Pregunta 11	Pregunta 12	Pregunta 13	Pregunta 14	Pregunta 15
Andrés Hurtado	Ni de acuerdo ni en desacuerdo	Ni de acuerdo ni en desacuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	En desacuerdo
Giovanni Zambrano	Ni de acuerdo ni en desacuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	En desacuerdo	En desacuerdo

Flor Hernández	Ni de acuerdo ni en desacuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	En desacuerdo
Leandro Antonelli	Ni de acuerdo ni en desacuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	En desacuerdo	En desacuerdo
Wilson Pantoja	Ni de acuerdo ni en desacuerdo	Ni de acuerdo ni en desacuerdo	En desacuerdo	Totalmente en desacuerdo	Totalmente en desacuerdo

Participante	Pregunta 16	Pregunta 17	Pregunta 18	Pregunta 19
Andrés Hurtado	De acuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo
Giovanni Zambrano	Ni de acuerdo ni en desacuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo
Flor Hernández	En desacuerdo	De acuerdo	De acuerdo	De acuerdo
Leandro Antonelli	De acuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo
Wilson Pantoja	Totalmente de acuerdo	De acuerdo	Totalmente de acuerdo	De acuerdo

Participante	Pregunta 20	Pregunta 21
Andrés Hurtado	Es una guía para poder hacer trazabilidad sobre el proceso de diseño	Ajustar un diagrama de global que integre todos los elementos
Giovanni Zambrano	El proceso de diseño y el modelo empleado son intuitivos y secuenciales y esto permite, tomarlo para el desarrollo de arquitecturas.	No tanto lo negativo, sino que sería muy interesante verla en un aplicación funcional.
Flor Hernández	Niveles de tiempo optimizados	Sin comentarios

Leandro Antonelli	Una secuencia de pasos con objetivos bien claros y concretos	Si tuviera experiencia en el desarrollo de arquitecturas IoT, el método podría parecer un tanto "burocrático"
Wilson Pantoja	Está personalizado para aplicar arquitectura al dominio de la IoT, ahorrando tiempo al arquitecto para conocer un dominio tan específico.	Ninguno

Comprensibilidad

Participante	Pr1	Pr 2	Pr 3	Pr4	Pr5	Pr 6	Pr 7	Pr8	Pr19
Andrés Hurtado	4	5	5	5	5	4	5	5	4
Giovanni Zambrano	4	4	5	5	4	4	4	5	4
Flor Hernández	4	4	5	4	4	4	4	4	4
Leandro Antonelli	4	1	4	3	3	5	5	5	4
Wilson Pantoja	4	5	5	5	5	4	5	5	4

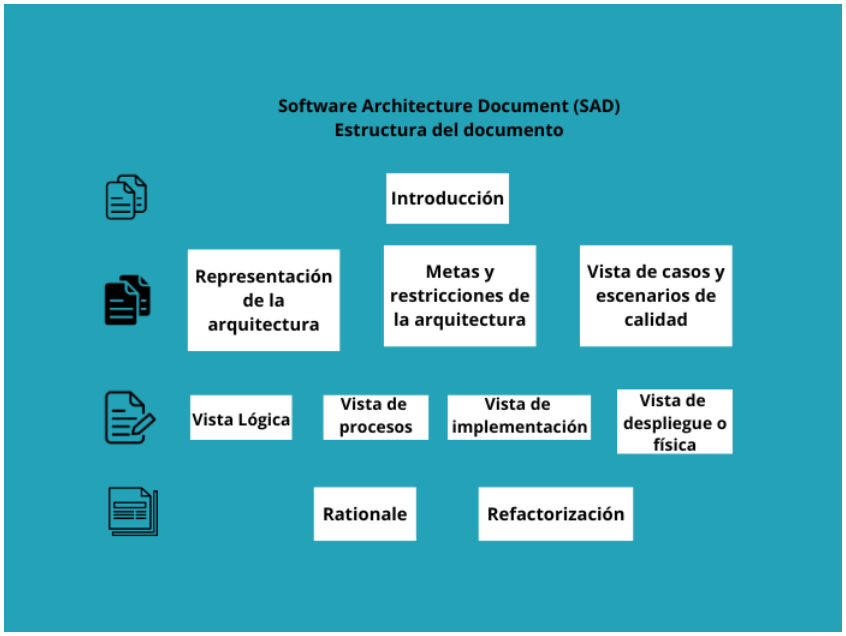
Utilidad

Participante	Pr9	Pr10	Pr11	Pr12	Pr13	Pr14	Pr15	Pr16	Pr17	Pr18
Andrés Hurtado	4	4	3	3	3	4	4	4	4	5
Giovanni Zambrano	5	5	3	3	4	4	4	3	4	5
Flor Hernández	4	4	3	2	3	4	4	2	4	4
Leandro Antonelli	5	4	3	3	4	4	4	4	4	5
Wilson Pantoja	5	5	3	3	4	5	5	5	4	5

ANEXO D

Ciclo de la Metodología

Estructura del Software Architecture Document (SAD)



La estructura del documento SAD será enunciada y explicada etapa por etapa a continuación:

Introducción

El proyecto "***Arquitectura de software adaptativa para soportar la cooperación de mini robots semi-autónomos en el contexto de la agricultura de precisión***" ha sido tomado como el estudio de caso para el proyecto de investigación ***"IoTAP: Prácticas de diseño de la arquitectura y sus requisitos de calidad en el contexto de desarrollo de software basado en IoT"*** de la Universidad del Cauca. El contexto de aplicación ha sido ***"enfocado en el diseño e implementación de una arquitectura que permite la adaptabilidad en un grupo de mini robots que desarrollan tareas en la agricultura de precisión orientados a la gestión del suelo, adquiriendo información relevante para la toma de decisiones y la automatización de las tareas realizando este trabajo de forma cooperativa".***

Contexto Del Problema

El mundo está retrocediendo en los esfuerzos para acabar con el hambre, la inseguridad alimentaria y la malnutrición, teniendo como meta del ODS 2, a falta de 8 años, causados por los conflictos bélicos, los fenómenos climáticos extremos y las perturbaciones económicas, el incremento de los alimentos nutritivos y las desigualdades [1]. Es un panorama que no mejorará si no se transforman los sistemas agroalimentarios, buscando mejorar el rendimiento de los cultivos, optimizando los recursos del suelo y de agua, para lograr alimentos nutritivos de un costo inferior que permita el acceso para todos, de forma sostenible e inclusiva. Es ahí donde la agricultura toma gran valor, como uno de los ejes estratégicos más importantes para poder lograr la seguridad alimentaria, esto requiere un cambio de los métodos utilizados comúnmente en la agricultura por prácticas agrícolas inteligentes que son conocidas como la agricultura 4.0 [24].

La tecnología e innovación como herramientas para lograr avanzar en estos esfuerzos, han sido el motor para la Agricultura 4.0 que está orientada por tendencias emergentes en tecnología como: la evolución de las mecanización agrícola cambiando el tamaño de los dispositivos, en lugar de utilizar maquinaria de gran tamaño y alto valor, en dispositivos más pequeños y de menor valor, modificando el uso de combustibles no renovables a emplear por energías renovables como lo son los dispositivos alimentados por paneles solares, minimizando el peso excesivo sobre el suelo, reduciendo los costos operativo y un valor de adquisición millonario, como fuente de cambio en la agricultura y así optimizar realmente los recursos en reduciendo los costos no solo económicos sino también ambientales [25].

Las tecnologías digitales como la internet de las cosas, aprendizaje automático y los sistemas robóticos autónomos como eje de la agricultura 4.0, se exploran significativamente en las granjas al aire libre y en granjas interiores, con un 69% y 31% de las investigaciones [3]. El empleo de grupos de robots que puedan realizar tareas de mecanización agrícola representa una evolución, que permite el aumento exponencial de las capacidades del campo, buscando la descomposición de una tarea realizada por la maquinaria y asignada a un grupo de robots. El desarrollo y control de estos sistemas robóticos representa un gran reto, que requiere tener en cuenta atributos de calidad como la adaptabilidad, escalabilidad, usabilidad, Configurabilidad, disponibilidad, autonomía entre otros y cómo lograr que el sistema sea cooperativo en donde varios robots necesitan interactuar entre sí para completar ciertas tareas [27].

Este trabajo pretende dar una posible solución a la siguiente pregunta:
¿Cuáles serían las consideraciones relevantes para estructurar soluciones basadas en mini robots para facilitar actividades cooperativas en el contexto de la agricultura de precisión?

En las investigaciones consultadas, se puede observar que van dirigidas a robots con uso específico y arquitecturas definidas con uso único, así mismo en el área de la agricultura 4.0 o agricultura de precisión, ven el uso de la robótica la forma de optimizar los recursos y obtener los mejores resultados, planteando retos y áreas de investigación, dentro de los cuales los enjambres, grupos o red de robots, plantean un campo de aplicación que puede repercutir en el desarrollo de nuevas tecnologías.

Propósito

La naturaleza proporciona ejemplos que nos muestran como cuando un organismo actúa independientemente no puede lograr un objetivo por sí mismo, pero en coordinación con otros organismos o compañeros de enjambre pueden resolver problemas complejos y completar el objetivo trazado que en la naturaleza está delimitado por la supervivencia de una especie [22]. Este paradigma es visto en aves, peces e insectos del mismo modo un grupo de robot básico puede formar un grupo que realiza una variedad de comportamientos que pueden ser superiores a las capacidades de los individuos empleando la comunicación, coordinación y cooperación es un muy buen modelo al comportamiento esperado por un enjambre de robots. Este principio del enjambre se puede emplear en varios niveles macro, micro e inclusive nanoescala con una gran cantidad de áreas de aplicación.

En la Agricultura de precisión es un conjunto de tecnologías que buscan optimizar la producción agrícola por medio de la variabilidad (espacial y temporal) de los factores de producción, gestionando los recursos disponibles como lo son el agua y el suelo [24]. La información de un cultivo en tiempo real variables de suelo, agua y ambientales, permite generar mapas de información que llevarán al agricultor a la toma de decisiones, ¿cómo potencializar los recursos disponibles para lograr un mejor rendimiento?, solo se puede llegar a decidir si se tiene la información necesaria, la arquitectura de software, permitirá controlar un grupo de robots que trabajando de manera semi autónoma obtendrán la información del área, compartiendo la información entre ellos y con un control centralizado.

Se plantea un entorno cooperativo de aprendizaje orientado a la realización de trabajo entre los robots para logra capturar, almacenar y distribuir las variables obtenidas del entorno, generando un repositorio particular de cada robot, un repositorio comunitario del robot con los robots cercanos y un repositorio en la nube, que permitirá el trabajo colaborativo con los interesados en la información obtenida, para generar aprendizaje y toma de decisiones, la información obtenida por medio de la tecnología del SEMI (Sistema de estaciones meteorológicas inteligentes) permitirá la trazabilidad de las variables que impactan el desarrollo del cultivo generando modelos, generando valor porque al conocer las variables y el cultivo se pueden generar modelos a partir de los cuales se puede mejorar y optimizar los cultivos, llegando a gestionar los recursos disponibles.

La tecnología busca construir sistemas que sean robustos, tolerantes a fallas y flexibles que los robots individuales que son construidos con un solo fin, buscando que los enjambres de

robots se puedan adaptar mejor su forma de actuar a los cambios en el entorno, en ahí donde se busca desarrollar una arquitectura que permita la interacción entre los robots y logren la coordinación, cooperación y adaptabilidad para que logren un objetivo común [8].

Alcance

La investigación ***"IoTAP: Prácticas de diseño de la arquitectura y sus requisitos de calidad en el contexto de desarrollo de software basado en IoT"*** tiene como objetivo apoyar en el diseño, evaluación y refactorización de la arquitectura de software en aplicaciones intensivas de IoT mediante sus prácticas planteadas y la metodología diseñada al proyecto de investigación ***"Arquitectura de software adaptativa para soportar la cooperación de mini robots semi-autónomos en el contexto de la agricultura de precisión"*** aportando en sus objetivos planteados***.***

Objetivo General

- Proponer un modelo arquitectónico adaptativo para facilitar las tareas de cooperación en contexto de un grupo de mini-robots semi-autónomos.

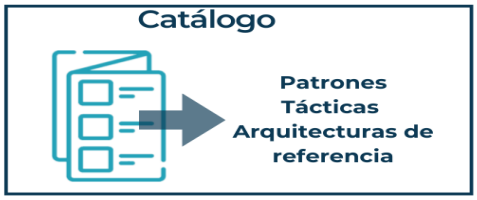
Objetivos específicos

- Caracterizar desde la literatura las arquitecturas de software, las cualidades, tácticas, patrones, los mecanismos de cooperación y colaboración, usados en sistemas de grupos de mini-robots autónomos.
- Formular un modelo de arquitectura adaptativa para sistemas basados en mini robots semi-autónomos y cooperativos, que permita balancear las cualidades estudiadas, considerando las tácticas y los patrones de arquitectura identificados en la literatura.

Roles

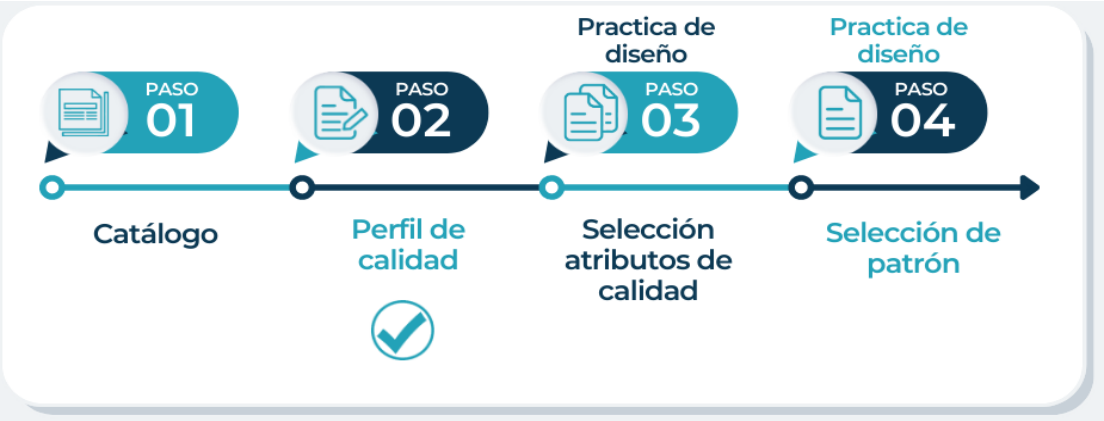
Participante	**Rol**	**Descripción de rol**
Giovanni Zambrano	*Arquitecto de software*	Participante que aplica las prácticas IoTAP de diseño y refactorización de arquitectura para productos software basados en IoT
Santiago López	*Arquitecto de software*	Diseñador de las prácticas IoTAP de diseño y refactorización de arquitectura para productos software basados en IoT
Julio Ariel Hurtado	*Arquitecto de software*	Diseñador de las prácticas IoTAP de diseño y refactorización de arquitectura para productos software basados en IoT
Flor Hernández	*Arquitecta de software*	Evaluadora de la arquitectura diseñada por medio de las prácticas IoTAP.

Metas y restricciones de la arquitectura

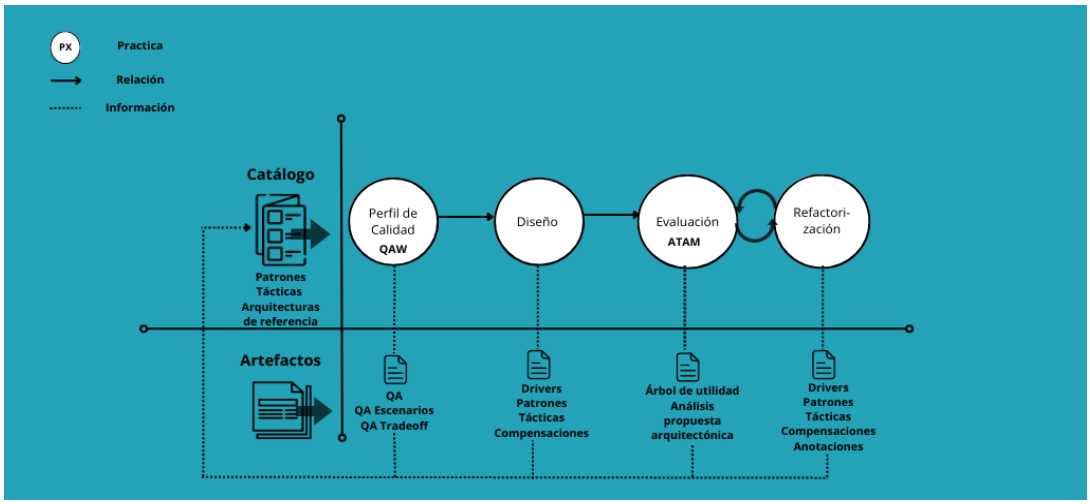


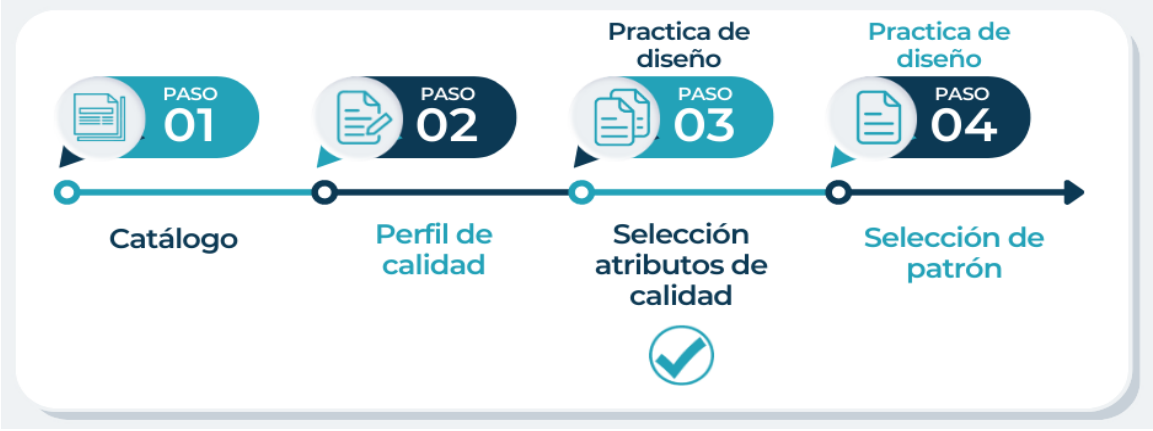
Se utilizan dos catálogos uno de atributos y otro de patrones para identificar cuáles son los más recurrentes en el dominio de la agricultura inteligente. A partir de la identificación se seleccionan los atributos y patrones relevantes para aplicaciones basadas en IoT, en este caso específico para robots semi-autónomos en agricultura. El resultado se podrá ver a continuación:

Atributos	***Patrones***
Eficiencia de desempeño	Capas
Compatibilidad	Basado en la nube
Usabilidad	Orientado a servicios
Confiabilidad	Microservicios
Seguridad	Restful
Mantenibilidad	Publicador/Suscriptor
Sensibilidad al entorno	Centrado en la información



Se realiza una breve descripción del marco metodológico que engloba todo el proceso que tienen las actividades, incluidas las prácticas de diseño y refactorización con sus respectivos pasos. Este marco sirve de guía general de toda la metodología propuesta.





El dominio a ser utilizado, es la agricultura inteligente, para el cual los atributos de calidad de la tabla son los más relevantes o sensibles para la arquitectura de software basados en IoT, puestos de manifiesto en las investigaciones consultadas y son los seleccionados para el desarrollo de la arquitectura a proponer. Los trabajos que componen el catálogo son los siguientes: [6], [17], [18], [58]–[61].

Atributos
Eficiencia de desempeño
Compatibilidad
Usabilidad
Confiabilidad
Seguridad
Mantenibilidad
Sensibilidad al entorno

El dominio a ser utilizado es la agricultura inteligente para el cual los patrones de arquitectura de la tabla son los más relevantes o sensibles para la arquitectura de software basados en IoT, puestos de manifiesto en las investigaciones consultadas y son los patrones seleccionados para el desarrollo de la arquitectura a proponer. Los trabajos que componen el catálogo son los siguientes: [6], [17], [18], [58]–[61].

Patrones
Capas
Basado en la nube
Orientado a servicios
Microservicios
Restful
Publicador/Suscriptor
Centrado en la información

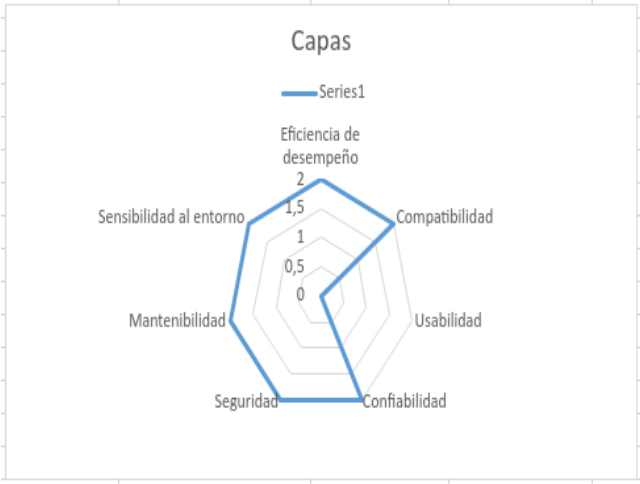
Se va a contrastar la influencia de los atributos de calidad en los patrones de arquitectura seleccionados por medio de la calificación de los mismos. Se tiene en cuenta la escala de calificación que se muestra a continuación:

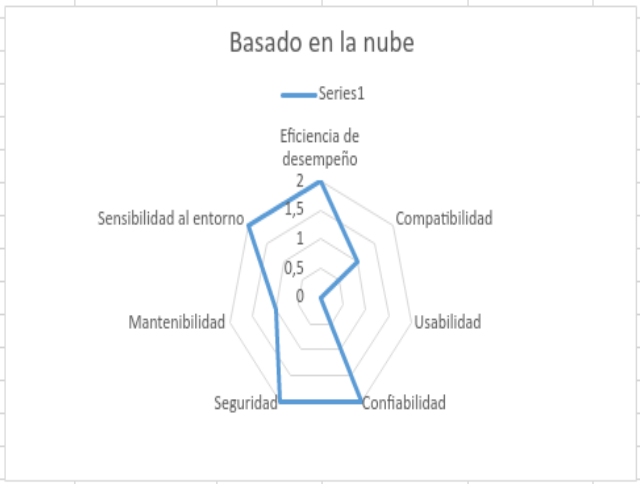
Item	Puntaje
**	2
*	1
/	0
~	-1
~~	-2

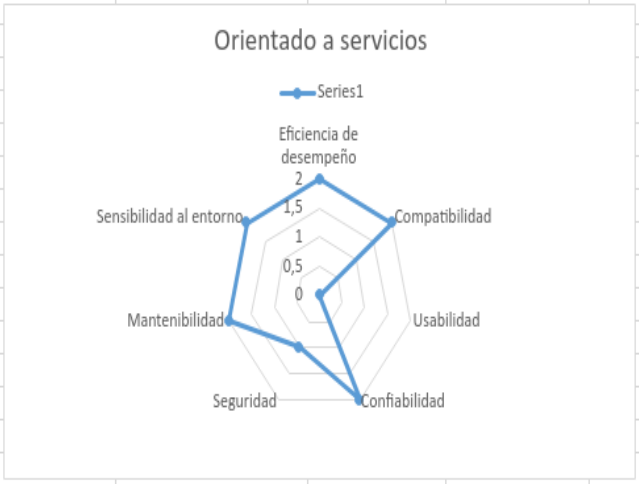
Aplicación practica de diseño IoTAP Dominio Agricultura							
Patron/Atributo	Eficiencia de desempeño	Compatibilidad	Usabilidad	Confiabilidad	Seguridad	Mantenibilidad	Sensibilidad al entorno
Capas	**	**	/	**	**	**	**
Basado en la nube	**	*	/	**	**	*	**
Orientado a servicios	**	**	/	**	*	**	**
Microservicios	**	**	/	*	*	*	**
Restful	/	*	/	**	*	**	*
Publicador/Suscriptor	**	**	*	*	**	**	**
Centrado en la informacion	*	*	/	**	*	*	**

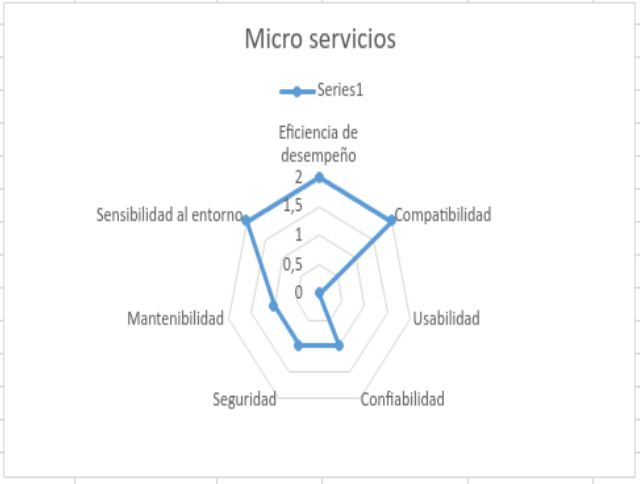
Aplicación practica de diseño IoTAP Dominio Agricultura								
patron/Atributo	Eficiencia de desempeño	Compatibilidad	Usabilidad	Confiabilidad	Seguridad	Mantenibilidad	Sensibilidad al entorno	
Capas	2	2	0	2	2	2	2	12
Basado en la nube	2	1	0	2	2	1	2	10
Orientado a servicios	2	2	0	2	1	2	2	11
Microservicios	2	2	0	1	1	1	2	9
Restful	0	1	0	2	1	2	1	7
Publicador/Suscriptor	2	2	1	1	2	2	2	12
Centrado en la informacion	1	1	0	2	1	1	2	8

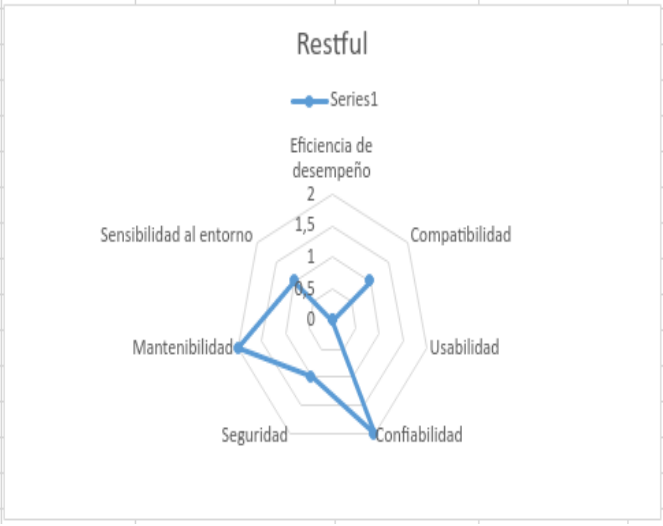
Gráficamente esta influencia estaría representada por los siguientes resultados:

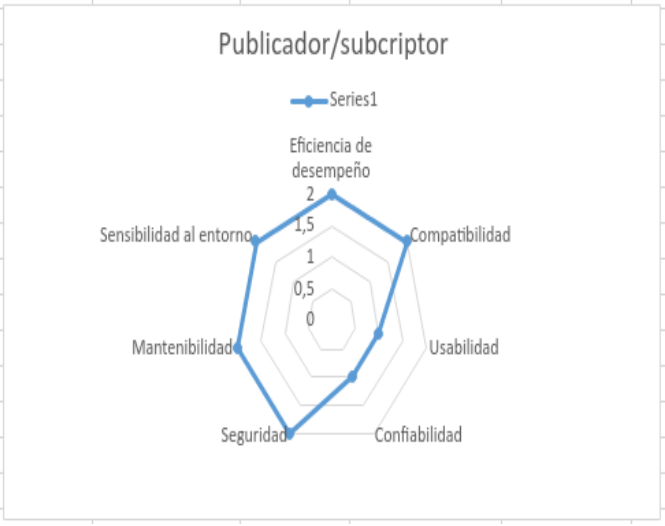


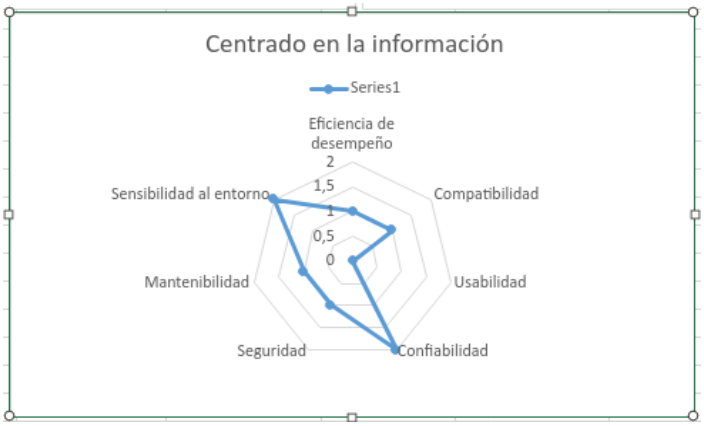




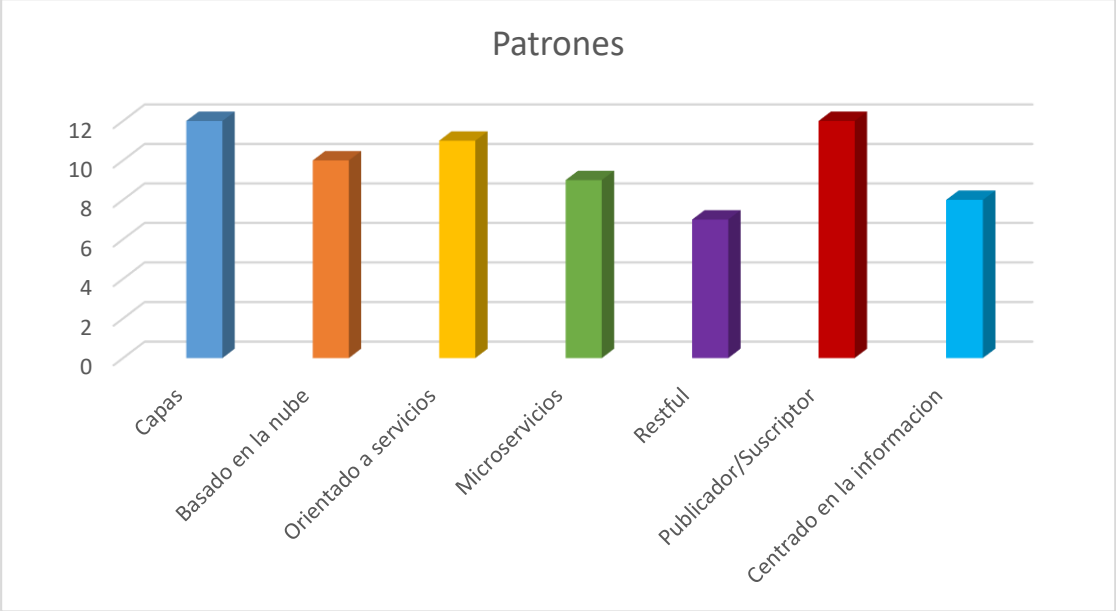








Se realiza la comparativa grafica de los siete patrones con el fin de determinar cuál es el más indicado para la investigación que se está apoyando:



Según los hallazgos de calificar los atributos vs. los patrones nos arroja tres candidatos de arquitecturas de referencia más indicados para el desarrollo de la arquitectura como lo son: (i) capas, (ii) orientado a servicios y (iii) publicador/suscriptor. El patrón seleccionado es el de **capas** y el porqué de la selección ***(Metas)*** y sus respectivos ***Drivers*** se detallan a continuación:

Drivers

- ***Eficiencia de desempeño:*** Esta arquitectura divide los elementos que la componen en niveles específicos y esto es relevante en el diseño de la arquitectura, capas como los sensores determinar una capa común que sería de fácil acceso por el usuario y los robots del sistema.

- ***Sensibilidad al entorno:*** Es muy relevante que el robot tenga la capacidad de adquirir información en tiempo real y poder percibir las pequeñas variaciones, que evaluadas con un histórico permita la toma de decisiones y la adaptabilidad en el desarrollo de la automatización de las tareas.

Otros Atributos de calidad relevantes

- ***Compatibilidad:*** Las capas delimitan los elementos o responsabilidades, esto permitiría al sistema adicionar más robots, teniendo en cuenta las capas delimitadas y permitiendo su interacción al hablar en el mismo lenguaje.

- ***Usabilidad:*** Esta está limitada por la capacidad de cómputo del sistema por lo cual se limita al hardware diseñado de los robots a ser implementados.

- ***Confiabilidad:*** las capas están organizadas por niveles de responsabilidad y aplicabilidad esto permite que podamos verificar el comportamiento del sistema y determinar los posibles errores y falencias del mismo en tiempo de ejecución y así poder generar respuestas a estos.

- ***Seguridad:*** es importante controlar el sistema para que no sea intervenido por agentes externos, al manipular información que tiene cierto grado de confidencialidad y que es pertinente para la automatización de los procesos, una modificación, podría hacer que el sistema respondiera erróneamente y causando un daño.

- ***Mantenibilidad:*** El sistema estaría organizado por niveles, estos subdividen su operación y delimitan el accionar, permitiendo evaluar el comportamiento de cada uno de ellos y determinar su funcionamiento para así evaluar un plan de mantenimiento para cada uno de ellos.

Las ***restricciones*** que se encuentran al utilizar este patrón se deben tener en cuenta ya que pueden afectar en el diseño de arquitectura, por ello [62] [63], sugiere que deben tenerse en cuenta estas mismas en los diseños de arquitectura:

- ***Eficiencia de desempeño***: La comunicación por la red o internet es una de las tareas más tardadas de un sistema, incluso, en muchas ocasiones más tardado que el mismo procesamiento de los datos, por lo que el hecho de tener que comunicarnos de capa en capa genera un degrado significativo en el performance.

- ***Escalabilidad:*** Las aplicaciones que implementan este patrón por lo general tienda a ser monolíticas, lo que hace que sean difíciles de escalar, aunque es posible replicar la aplicación completa en varios nodos, lo que provoca un escalado monolítico.

- ***Complejidad de despliegue***: En este tipo de arquitecturas es necesarios desplegar los componentes de abajo arriba, lo que crea una dependencia en el despliegue, además, si la aplicación tiende a lo monolítico, un pequeño cambio puede requerir el despliegue completo de la aplicación.

- **Anclado a un Stack tecnológico**: Si bien no es la regla, la realidad es que por lo general todas las capas de la aplicación son construidas con la misma tecnología, lo que hace amarremos la comunicación con tecnologías propietarias de la plataforma utilizada.

- ***Tolerancia a los fallos***: Si una capa falla, todas las capas superiores comienzan a fallar en cascada.

- ***Modificabilidad/ Mantenibilidad:*** A veces no se logra la contención de un cambio y se requiere una cascada de cambios en varias capas.

En el contexto de diseño para arquitecturas de software basada en IoT***, con el fin de argumentar*** por que se escoge el patrón de capas y no se escogen los patrones resultantes como el patrón orientado a servicios y Publicador/Subscriptor***. Se decide utilizar el patrón de capas debido a su alta abstracción, encapsulamiento, capas de funcionalidad muy bien definidas, alta cohesión y reusabilidad. Se descartan*** los patrones como, el orientado a servicios ya que no es óptimo para la presente arquitectura debido al factor seguridad, no es muy relevante y en el sistema propuesto es necesario para velar por la seguridad de la información. Por otra parte, tampoco se escoge el patrón Publicador/Subscriptor ya que requiere una interfaz más elaborada, esto requeriría un aumento del hardware de los dispositivos o robots, así mismo se requiere que un sistema híbrido que permita la interacción entre los robots su autorganización y variación del sistema. **Para lograr** más eficiencia de desempeño y sensibilidad al entorno con respecto al correcto funcionamiento (en tiempo real) y a la toma de decisiones con base en la información recolectada del entorno**, se acepta que esto afectara** a la arquitectura en términos de componentes ***debido*** a que se debe garantizar que el tránsito de información no sea continuo**,** sino que cuando se perciba información relevante se actué con base en ella sin saturar el sistema.

Representación de la arquitectura

El patrón en capas se define según [62] [65], como un patrón de arquitectura software que usa en la gran mayoría de sistemas y se centra en una distribución jerárquica de los roles y responsabilidades proporcionando una separación efectiva de las preocupaciones (cada cual se encarga de lo que le corresponde). Al pensar en un sistema en términos de capas, se imaginan los principales subsistemas de software ubicados de la misma forma que las capas de un pastel, donde cada capa descansa sobre la inferior. Sus principales beneficios son:

Beneficios:

- Abstracción, encapsulamiento, capas de funcionalidad muy bien definidas, alta cohesión, reusabilidad (las capas inferiores de la pirámide no tienen dependencias de las superiores por lo que es fácil utilizarlas) y débil acoplamiento (comunicación basada en abstracciones).

- La separación, reduce el riesgo y el impacto de los cambios tecnológicos. Por ejemplo, al cambiar el mecanismo de persistencia de la aplicación las capas que lo consumen no tienen por qué realizar cambios. Además, aumenta el rendimiento (a partir de cierta carga de trabajo) al distribuir las capas sobre múltiples capas físicas.

También aumenta la escalabilidad y la tolerancia a fallos.

- Este escenario es propicio para implementar una buena política de pruebas, permitiendo conmutar entre diferentes implementaciones de los interfaces (mock, servicios reales.
- Separación de responsabilidades: Permite la separación de preocupaciones (SoC), ya que cada capa tiene una sola responsabilidad.
- Fácil de desarrollar: Este estilo arquitectónico es especialmente fácil de implementar, además de que es muy conocido y una gran mayoría de las aplicaciones la utilizan.
- Fácil de probar: Debido a que la aplicación construida por capas, es posible ir probando de forma individual cada capa, lo que permite probar por separada cada capa.
- Fácil de mantener: Debido a que cada capa hace una tarea muy específica, es fácil detectar el origen de un bug para corregirlo, o simplemente se puede identificar donde se debe aplicar un cambio.
- Seguridad: La separación de capas permite el aislamiento de los servidores en subredes diferentes, lo que hace más difícil realizar ataques.

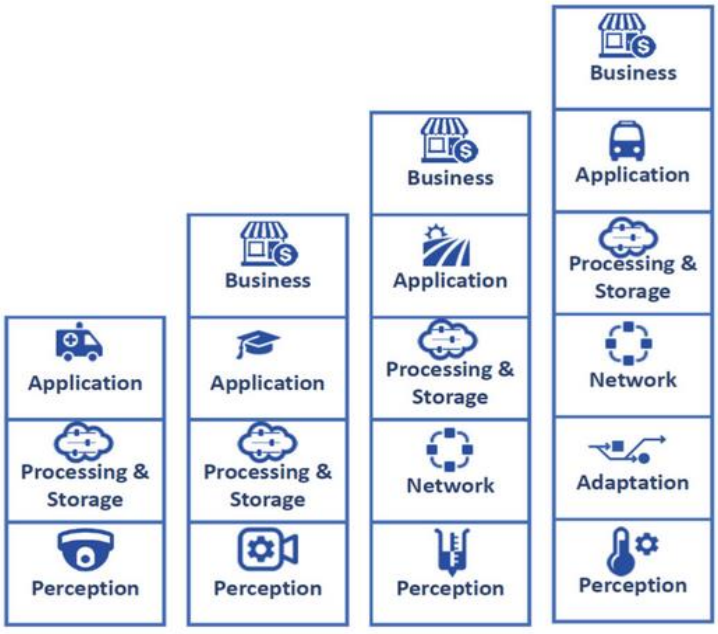


Fig. 3. Layered IoT architecture

Para el diseño de la arquitectura de software del este proyecto se toma como referencia la arquitectura por capas de 6 niveles compuesto de las siguientes capas, tomado del catálogo de arquitecturas de referencia:

- **Capa de negocio:** Es la capa usada por los usuarios en la cual se llaman a las funcionalidades desde la capa de aplicación al usuario final.

- **Capa de aplicación:** Es la capa responsable de proporcionar los servicios inteligentes de IoT a los usuarios y se enfatiza en las necesidades a nivel de atributos de calidad que se requieren para un producto software basado en IoT.

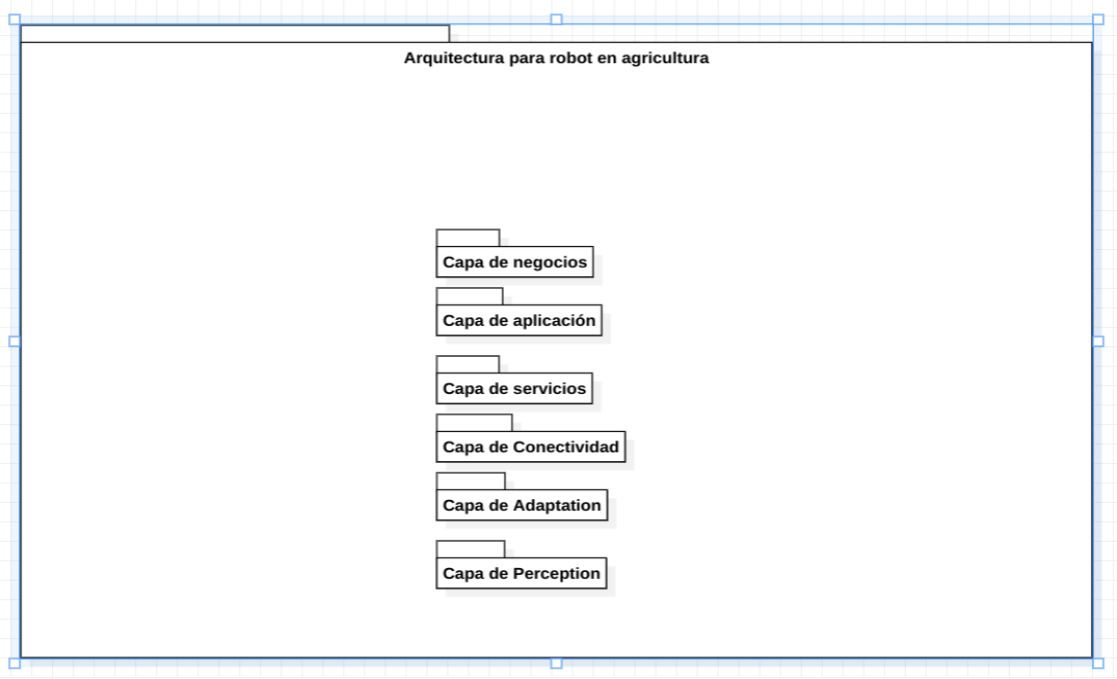
- **Capa de servicio:** Se encarga de analizar y procesar los datos que provienen de la capa de conectividad. También es llamada capa de middleware debido a que contribuye con el intercambio de información entre objetos heterogéneos que no cuentan con requisitos específicos de software y hardware.

- **Capa de conectividad:** Es la capa responsable de procesar y transmitir los datos que se reciben de las capas de percepción y adaptación. Cuenta con diferentes tipos de redes como WiFi, móvil, etc.

- **Capa de adaptación:** es la capa que tiene la responsabilidad de cambiar el comportamiento del sistema con base en los datos obtenidos a partir de la capa de percepción.

- **Capa de percepción:** Esta capa se encarga de percibir, detectar, recolectar y medir información del contexto circundante, por ejemplo: temperatura, humedad, etc.

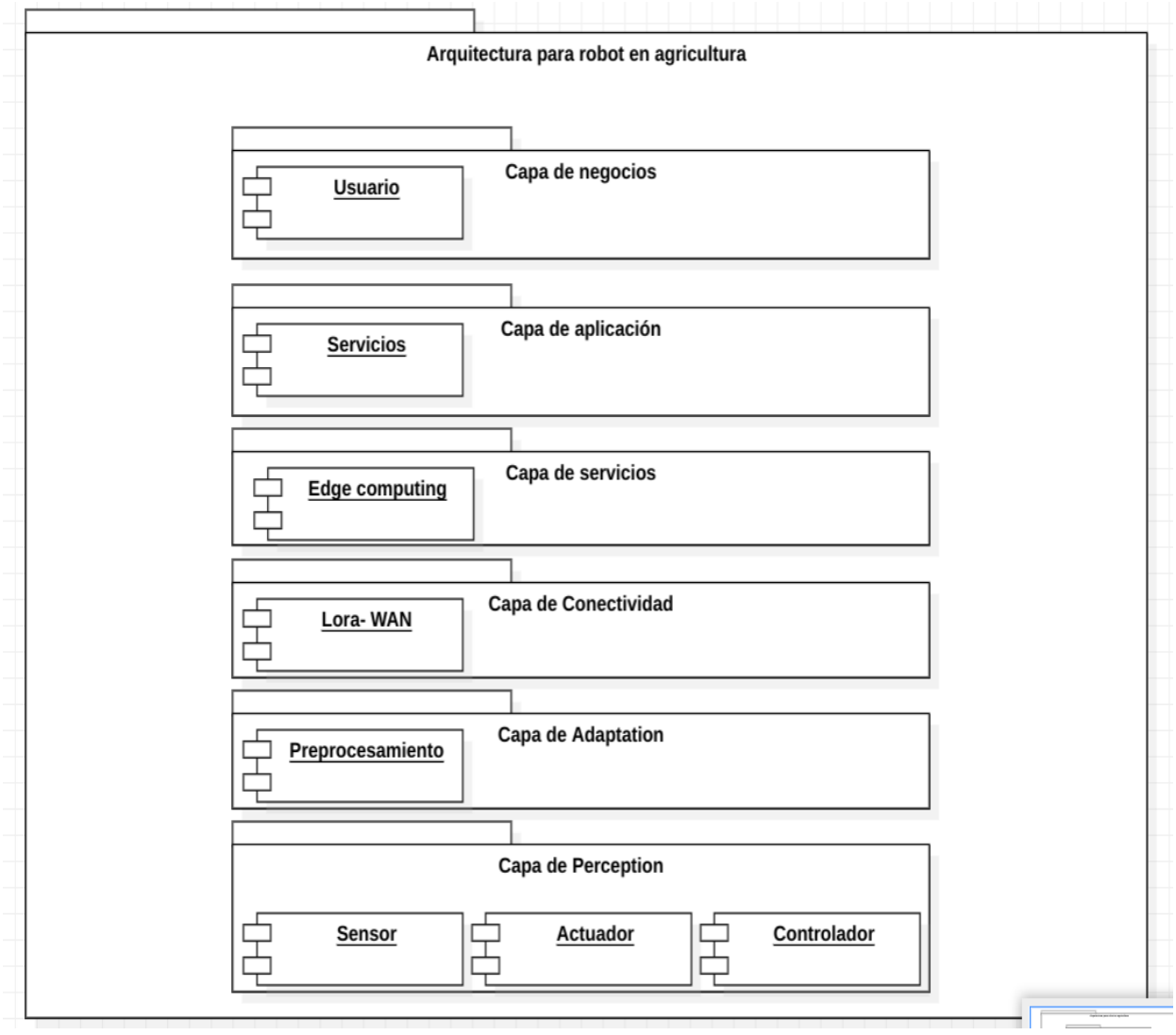
Se realiza una vista inicial del diseño de la arquitectura incorporando las capas mencionadas anteriormente y podrá observarse a continuación:



Vistas

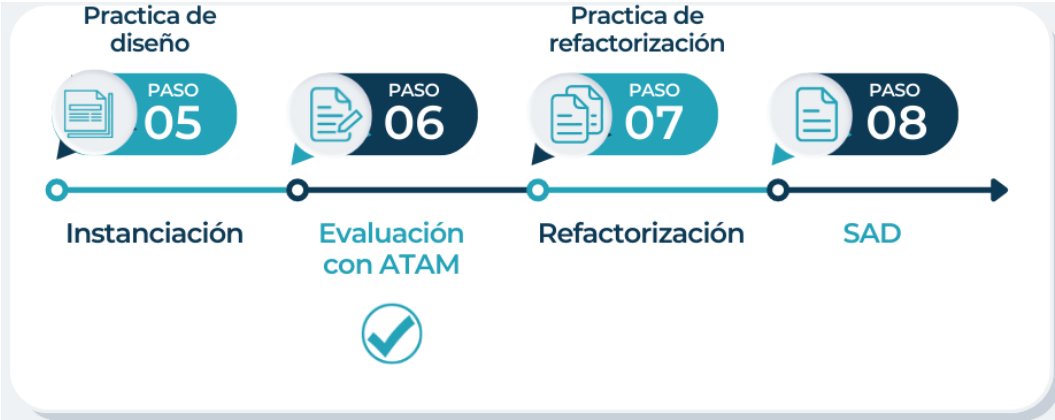
(Descripción de manera gráfica y teórica de las vistas generadas a partir de las prácticas IoTAP: ***(i) lógica, (ii) procesos, (iii) implementación o (iv) despliegue***).

Como resultado de aplicar la práctica de diseño para arquitecturas basadas en IoT, se obtiene un diseño preliminar de la vista lógica de la arquitectura para ***"Arquitectura de software adaptativa para soportar la cooperación de mini robots semi-autónomos en el contexto de la agricultura de precisión".***



Vista de casos y escenarios de calidad

(Se describen los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa y se reportan los artefactos resultantes seleccionados de la metodología ATAM: ***(i) árbol de utilidad y (ii) Análisis de una propuesta arquitectónica***).



De la evaluación con la metodología ATAM adaptada para la presente metodología se tienen en cuenta solo dos artefactos que son cruciales para la mejora continua de las arquitecturas y corresponden a**: (i) Árbol de Utilidad y (ii) Análisis de una propuesta arquitectónica.**

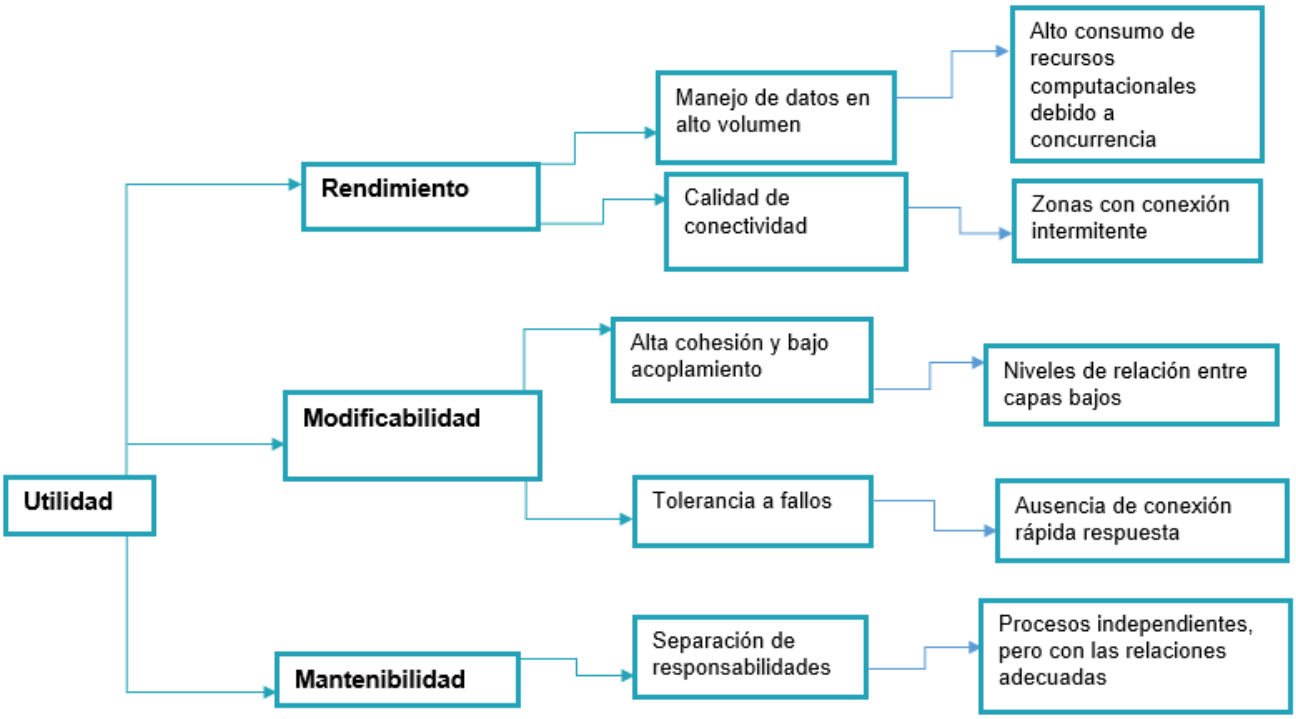
Los atributos sensibles para la presente arquitectura son: **(i) Rendimiento, (ii) Modificabilidad y (iii) Mantenibilidad.** Estos se representan en el árbol de utilidad para identificar las preocupaciones y escenarios para la arquitectura, además se presentan las sensibilidades, riesgos y en caso de existir otros QA relevantes. Por otra parte, se realiza el **análisis de una propuesta arquitectónica** para examinar los escenarios, sensibilidades y riesgos identificados en árbol de utilidad.

- **Rendimiento:** Se observa que es un atributo sensible teniendo en cuenta que el sistema manipulará cantidades de datos considerables, en zonas rurales donde la cobertura se puede presentar de forma deficiente. Se debe garantizar que esa capa de conexión provea y reciba de las capas inferiores los elementos necesarios para un funcionamiento adecuado.

- **Modificabilidad:** la sensibilidad a cambios se presenta en un nivel considerable por tanto sería apropiado tener en cuenta la cohesión y acoplamiento, buscando de cada proceso de cierta manera cumpla su responsabilidad pero que haya una tolerancia a fallos que permita al sistema dar respuesta rápida y reponerse anta cualquier evento.

- **Mantenibilidad:** la separación de responsabilidades es importante y se debe prevenir niveles de dependencia altos, esto previene los cambios.

Árbol de utilidad



Análisis de una propuesta arquitectónica

<table>
<tr><td>Ítem</td><td>Descripción</td></tr>
<tr><td>Escenario</td><td>Alto consumo de recursos computaciones debido a concurrencia</td></tr>
<tr><td>Atributo</td><td>Rendimiento</td></tr>
<tr><td>Ambiente</td><td>Operaciones normales</td></tr>
<tr><td>Estimulo</td><td>Acceso concurrente a la información percibida.</td></tr>
<tr><td>Respuesta</td><td>Se apaga la tarjeta</td></tr>
<tr><td>Decisiones arquitecturales (DA)</td><td>
<table>
<tr><td>DA</td><td>Riesgo</td><td>Sensibilidad</td><td>Compensación</td></tr>
<tr><td>1 recursos eficientes</td><td>Alto</td><td>Alto</td><td>Optimización de algoritmos.</td></tr>
<tr><td>2 transferencia de tarea</td><td>Alto</td><td>Medio</td><td>Respuesta y respaldo al proceso</td></tr>
</table>
</td></tr>
<tr><td>Razonamiento</td><td>Los recursos del sistema deben ser utilizados óptimamente, para lograr la robustez del sistema y su correcto funcionamiento (en tiempo real), razón por la cual estoy de acuerdo debemos implementar la tácticas para lograr que el sistema pueda desarrollar las tareas en tiempo real y así lograr la toma de decisiones basado en los datos obtenidos del sistema, la capa impactada por la táctica es de adaptación y se verificará el impacto en las demás capas, es importante tener en cuenta que esta táctica puede impactar la arquitectura de software afectando el rendimiento de los agentes.</td></tr>
</table>

<table>
<tr><td>Ítem</td><td>Descripción</td></tr>
<tr><td>Escenario</td><td>Zonas con conexión intermitente</td></tr>
<tr><td>Atributo</td><td>Rendimiento</td></tr>
<tr><td>Ambiente</td><td>Operaciones normales</td></tr>
<tr><td>Estimulo</td><td>Perdida de conexión</td></tr>
<tr><td>Respuesta</td><td>Cero transmisión de datos</td></tr>
<tr><td>Decisiones arquitecturales (DA)</td><td>
<table>
<tr><td>DA</td><td>Riesgo</td><td>Sensibilidad</td><td>Compensación</td></tr>
<tr><td>1 funcionamiento offline</td><td>Alto</td><td>Alta</td><td>Proceso continuo, pero cambio en la eficiencia</td></tr>
<tr><td>2 precargado en memoria temporal</td><td>Alto</td><td>Alta</td><td>Afecta la eficiencia.</td></tr>
</table>
</td></tr>
<tr><td>Razonamiento</td><td>Los datos obtenidos por los agentes es el insumo para poder representar el estado del cultivo, razón por la cual se debe garantizar el funcionamiento del sistema de forma offline guardando temporalmente la información, por lo cual estas tácticas permiten que los datos obtenidos se almacenen temporalmente y al momento de restablecer, lograr ser transmitidos en lo posible sin pérdida de información, así mismo se debe tener en cuenta que puede afectar el rendimiento de los agentes por lo cual se debe tener en cuenta este impacto a la arquitectura de software.</td></tr>
</table>

Rationale

De acuerdo a lo encontrado anteriormente en la sección de casos y escenarios de calidad, se identifican las preocupaciones y/o sensibilidades principales para la arquitectura seleccionada, al igual que las tácticas, patrones y/o decisiones de diseño).

Driver candidato	Preocupación	Tácticas	Patrones satisfactores y/o decisiones de diseño
Eficiencia de desempeño	**Alto volumen de datos y conexión intermitente**	**Priorización de eventos** **Incrementar eficiencia de los recursos**	Recursos eficientes Transferencia de tarea Funcionamiento offline Precargado en memoria temporal

Refactorización

Impacto de las tácticas en las capas

Extiende el razonamiento realizado en vistas de casos y escenarios de calidad, se observa el impacto que pueden tener estas tácticas, dentro del patrón y como impactan con respecto a los atributos de calidad.

- **Capa de adaptación:** estas tácticas permiten que el sistema pudiera adaptarse a los cambios presentados como el perder conectividad de los robots y mantener las tareas en ejecución, permitiendo la implementación de un gestor de eventos que priorice la información y su verificación de cambios si son relevantes o con cambios significativos, así mismo como la conectividad en el momento, manejo de eventos que pueden ocurrir en el sistema, como el nivel de conectividad, manejo de información del contexto entre otros, estas tácticas pueden tener un impacto negativo en la eficiencia de desempeño del sistema, se evalúa si este degradación es mayor que enviar la información sin que ocurran los eventos.

- **Capa de percepción:** los datos tomados por lo robots son muy importantes y se pone de manifiesto en la toma de decisiones, razón por la cual estas tácticas permiten que no se pierda información al almacenarla en una memoria temporal, siempre verificando que puedan conectarse de nuevo al sistema y ser cargada, esto podría ocasionar en los robots la baja rendimiento de los recursos y su operatividad, se necesita de la optimización de los algoritmos para la toma de variables del entorno. debe tener en cuenta la configuración, calibración y precisión de los sensores actuadores y tarjetas nivel de validación en los algoritmos para que la capa de adaptabilidad razone de forma óptima, con base en la información recolectada del contexto, si no tenemos en cuenta estas validaciones, los datos almacenados en la memoria temporal podrían afectar la eficiencia y rendimiento del sistema, estas tácticas afectaron la capacidad de la memoria temporal, pero al utilizar este filtro optimizar los datos obtenidos y evitará la saturación del sistema.

  Filtro: si las variables no cambian no necesita transmitir.

En los atributos de calidad seleccionados para la presente arquitectura, no se presentan conflictos entre los mismo, al contrario, las tácticas apoyan y mejoran el impacto que ellos pueden tener en la arquitectura a desarrollar.

Se genera una ***nueva vista lógica***, la sigla ***PF*** corresponde al driver ***Performance*** o más conocido como ***eficiencia de desempeño***. A partir de las tácticas implementadas, ***PF1 Incrementar eficiencia de los recursos y PF2 Priorización de eventos***, se denota con la sigla ***AI*** que se introdujeron las tácticas mencionadas anteriormente en el patrón de capas. Esto soportado por el trabajo [28], el cual propone un método para anotar diagramas de arquitectura con información de implementación de la táctica. Es una manera clara de documentar la adición de tácticas a los documentos de arquitectura que también es fácil de usar, permite que los componentes arquitectónicos permanezcan visibles y puede anotar muchos estilos diferentes de diagramas. Se basa en los tipos de cambios en los patrones de arquitectura descritos anteriormente. El método de anotación consta de dos cosas: una lista de tácticas y los círculos que muestran la ubicación y el tipo de cambios en la arquitectura para una táctica. La lista de tácticas consta de entradas que asocian una táctica

implementada con el círculo que muestra su implementación. Cada entrada en la lista de tácticas consta de un identificador único y el nombre de la táctica. El identificador consiste en una abreviatura del atributo de calidad, seguida de un número. El número identifica de forma única los cambios para esa táctica.

Table 5
Sample quality attribute abbreviations.

R	Reliability
S	Security
U	Usability
PF	Performance
PO	Portability
M	Maintainability
CP	Capacity
CF	Configurability
E	Extensibility

Table 6
Sample tactic list.

ID	Tactic
S1	Authorization (security ++)
S2	Encryption (security ++)
R3	Ping/Echo (reliability − −)
PF4	Concurrency (performance ~)

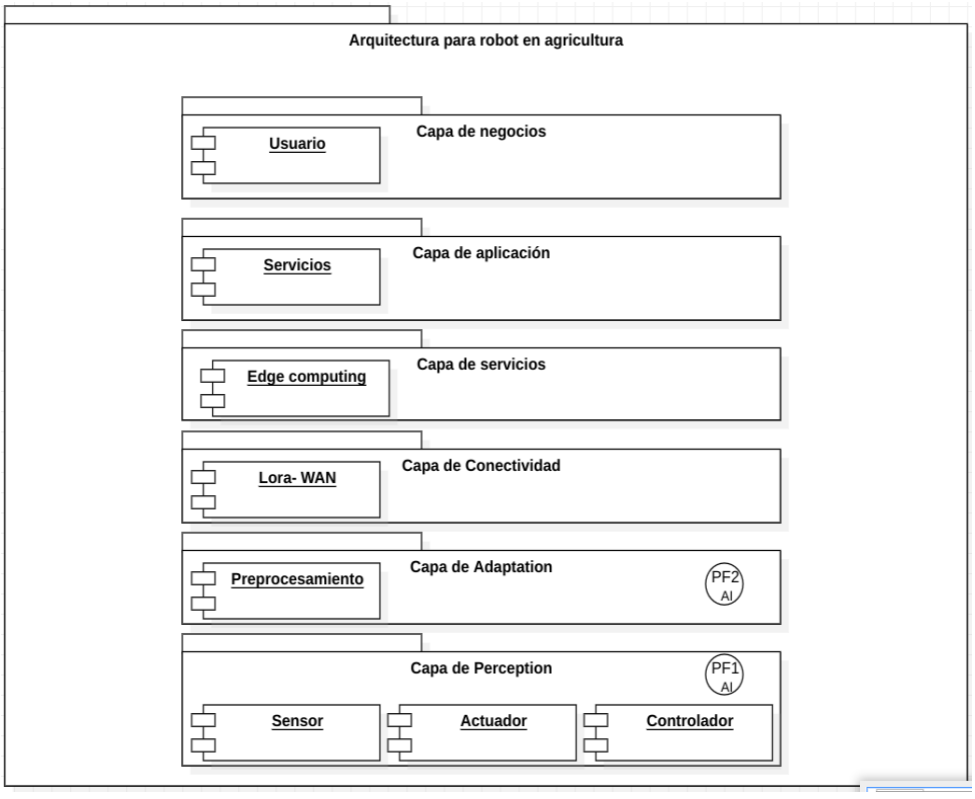
Table 7
Types of change to pattern participants.

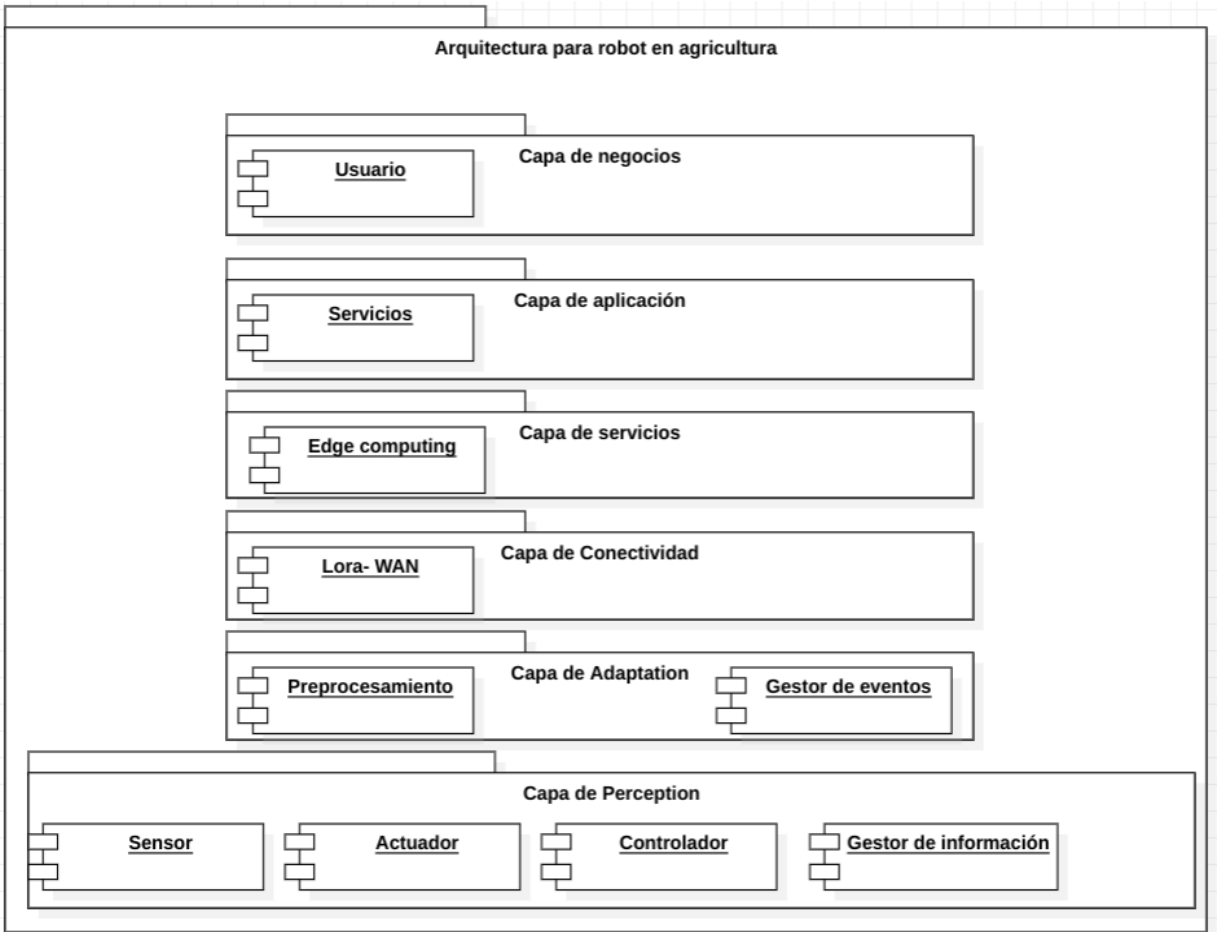
Abbreviation	Type
I	Implemented in
R	Replicated
AI	Added, in the pattern
AO	Added, out of the pattern
M	Modified

Atypical circle looks like this:

Fig. 4. Tactic implementation annotation style.

Fig. 5. Sample tactic implementation annotation.





Se introducen en la arquitectura de software, en la capa ***de percepción el gestor de información y en la capa de adaptabilidad el gestor de eventos***, para instanciar el uso de las tácticas identificadas por parte de la evaluadora de arquitecturas ***Flor Hernández***, con el objetivo de mejorar la calidad de la arquitectura. La información de estas actividades se plasma en el **Software Architecture Document (SAD).**