

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL
ECUADOR SEDE ESMERALDAS**



TÍTULO:

**MODELO DE OBJETO PARA AGENTIFICACION DEL INTERNET
DE LAS COSAS**

LÍNEA DE INVESTIGACIÓN:

AUTOMATISMOS Y SISTEMAS INTELIGENTES

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO DE SISTEMAS Y COMPUTACIÓN

AUTOR:

BRYAN ESPINOZA ROBLES

ASESOR:

PABLO PICO VALENCIA (PhD)

ESMERALDAS, 2021

TRIBUNAL DE GRADUACIÓN

Trabajo de tesis aprobado luego de haber dado cumplimiento a los requisitos exigidos por el reglamento de Grado de la PUCESE previo a la obtención del título de INGENIERO DE SISTEMAS Y COMPUTACIÓN.

.....
PhD. Pablo Pico Valencia
Tutor de Tesis

.....
Mgt. Evelin Flores García
Lector 1

.....
Mgt. Gustavo Chango Sailema
Lector 2

.....
Mgt. Susana Patiño Rosado
Coordinadora de la Escuela de Sistemas y Computación

AUTORIA

Yo, **Espinoza Robles Bryan Alberto**, con número de cédula de identidad **0804511731** manifiesto que mediante la presente investigación sobre el tema **“MODELO DE OBJETO PARA AGENTIFICACIÓN DEL INTERNET DE LAS COSAS”** los resultados obtenidos como tesis de grado, previo a la obtención del título de **“INGENIERO EN SISTEMAS Y COMPUTACIÓN”** son de total responsabilidad del autor, y que se ha respetado las fuentes de información consultadas, realizando las citas correspondientes y los resultados alcanzados son totalmente personales, únicos y legítimos. Al mismo tiempo declaro que todo el contenido incluyendo resultados, discusión, conclusiones, recomendaciones y otros efectos legales y académicos que se desglosan, son y serán exclusiva responsabilidad legal y académica del autor y de la PUCESE.

Espinoza Robles Bryan Alberto

C.I. 0804511731

AGRADECIMIENTO

Agradezco a Dios por darme la vida, salud y por darme la oportunidad de terminar cada etapa de mi vida con éxito, por haberme guiado a lo largo de mi carrera universitaria, por ser mi fortaleza y las bendiciones recibidas cada día.

Agradezco a mis padres por estar conmigo en cada paso como persona, por inculcarme cada principio y valores, por los regaños y paciencia como hijo por cada apoyo y sacrificio no solo conmigo sino también con mis demás hermanas y hermano para convertirnos en personas capaces de poder cumplir todo lo que uno se propone y que, ante los muchos obstáculos, el que persevera alcanza.

Agradezco a cada miembro de mi familia que es inmensa y les guardo un infinito amor y cariño.

A todos mis compañeros de escuela, colegio, universidad con quienes compartí muchas vivencias y aprendizaje.

Bryan Espinoza.

DEDICATORIA

Le dedico este trabajo a mis padres Alberto Espinoza Coterá y Alicia Robles Palma por haber confiado en mí y ser un apoyo importante durante el transcurso de toda mi vida, mi inmensa gratitud y cariño a ustedes que son las personas que más amo en el mundo.

Bryan Espinoza

Índice de contenidos

TRIBUNAL DE GRADUACIÓN	I
AUTORIA.....	II
AGRADECIMIENTO	III
DEDICATORIA.....	IV
RESUMEN.....	IX
ABSTRACT	X
INTRODUCCIÓN	1
PRESENTACIÓN DEL TEMA DE INVESTIGACIÓN	1
PLANTEAMIENTO DEL PROBLEMA.....	1
JUSTIFICACIÓN	2
DELIMITACIÓN DE OBJETIVOS.....	3
Objetivo general.....	3
Objetivos específicos	3
CAPÍTULO I: MARCO TEÓRICO	4
1.1. MARCO TEÓRICO-CONCEPTUAL	4
1.1.1. Internet de las cosas	4
1.1.2. Aspectos importantes del IoT.....	5
1.1.3. Arquitectura de IoT basada en capas	6
1.1.4. Software para IoT (middlewares).....	8
1.1.5. Protocolos de comunicación de IoT.....	9
1.1.6. Aplicaciones de IoT	11
1.1.7. Aplicación del IoT en el área de salud	12
1.1.2. Paradigma basado en agentes.....	14
1.1.2.1. Agente software.....	14
1.1.2.2. Características de un agente	14
1.1.2.3. Arquitectura de agentes	15
1.1.2.4. Sistema multiagentes	16
1.1.2.5. Marco de desarrollo JADE	16
1.1.3. Agentificación del IoT	18
1.1.3.1. Agentificación basada en agentes embebidos.....	19
1.1.3.2. Agentificación basada en MASs.....	21
1.2. ANTECEDENTES.....	22
1.3. BASES LEGALES.....	25
CAPÍTULO II: METODOLOGÍA.....	27
2.1. DELIMITACIÓN DE LA INVESTIGACIÓN	27
2.2. TIPO DE INVESTIGACIÓN.....	27
2.2. MÉTODOS DE INVESTIGACIÓN	28
2.3. VARIABLES SUJETAS A INVESTIGACIÓN	29
2.4. TÉCNICA DE INVESTIGACIÓN E INSTRUMENTOS DE RECOLECCIÓN DE DATOS	30

2.5. NORMAS ÉTICAS	30
CAPITULO III: RESULTADOS	31
3.1. REVISIÓN DE LA LITERATURA	31
3.1.1. Sistemas de IoT en salud.....	31
3.1.2. Sistemas de IoT basados en agentes en salud	34
3.1.3. Taxonomía de agentes.....	34
3.2. PROPUESTA.....	36
3.2.1. Descripción del sistema	36
3.2.2. Prototipo del objeto de IoT	37
3.2.2.1. Hardware del objeto de IoT	38
3.2.2.1. Diseño del circuito	38
3.2.2.2. Lectura de datos de sensores	40
3.2.2.3. Ecosistema de agentes para objeto de IoT.....	41
3.2.2.4. Integración, agentificación del IoT	46
3.2.2.5. Modelo de datos	47
3.2.2.6. Prueba del objeto propuesto	48
3.3. EVALUACIÓN DEL PROTOTIPO	49
3.3.1. Evaluación de la usabilidad.....	49
3.3.2. Evaluación de la funcionalidad	53
3.3.3. Portabilidad	55
CAPÍTULO IV: DISCUSIÓN.....	57
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES	59
5.1. CONCLUSIONES	59
5.2. RECOMENDACIONES.....	60
BIBLIOGRAFÍA.....	62

Índice de figuras

Figura 1 Ecosistema de IoT [4]. _____	4
Figura 2 Arquitectura de 5 capas de IoT [8] _____	7
Figura 3 Arquitectura del marco de desarrollo de agentes JAVA (JADE) [28]. _____	17
Figura 4 Arquitectura de una entidad agente de cosa para agentificar el IoT [31] _____	19
Figura 5 Arquitectura para agentificar el IoT usando sistemas multiagentes [33] _____	22
Figura 6 Taxonomía de agentes para el cuidado de la salud IoT. _____	36
Figura 7 Escenario asistencia domiciliaria basado en IoT. _____	37
Figura 8 Diagrama de bloques del prototipo de objeto de IoT construido. _____	37
Figura 9 Circuito de Arduino con los sensores y comunicación con la Raspberry PI. _	38
Figura 10 Diagrama del esquema de las conexiones de los componentes _____	38
Figura 11 Construcción del prototipo final _____	39
Figura 12 Colocación de los sensores para la recolección de datos _____	39
Figura 13 Lectura de datos desde el entorno de Raspberry PI 3. _____	40
Figura 14 Modelo de sistema multi- agentes de IoT _____	42
Figura 15 Diagrama de Secuencia de la comunicación de los agentes _____	43
Figura 16 Organización de las plantillas para crear los agentes JADE. _____	44
Figura 17 Plataforma JADE y agentes del sistema. _____	46
Figura 18 Modelo general del Sistema para agentificación de objetos para el IoT _____	47
Figura 19 Funcionamiento de los agentes resultado final en NetBeans _____	48
Figura 20 Efectividad del prototipo _____	50

Índice de tablas

Tabla 1 Variables para la calidad del sistema	29
Tabla 2 Selección de estudios de IoT aplicados al cuidado de la salud	32
Tabla 3. Principales estudios en los que se propone agentificar el IoT para el desarrollo de sistemas de monitoreo de la salud.	35
Tabla 4 Comparación de las arquitecturas para los agentes	41
Tabla 5 Valores por cumplimiento.....	54
Tabla 6 Evaluación del Funcionamiento	54

RESUMEN

Hoy en día, el Internet de las Cosas (IoT por sus siglas en inglés) se está convirtiendo en una tecnología prometedora. Cada día está cambiando la forma de entender el mundo llegando a modernizar y simplificar el estilo de vida de las personas a través de objetos o dispositivos conectados al ciberespacio que interactúan entre sí para automatizar procesos. No obstante, se hace necesario la disposición de un modelo de objeto que permita el modelado de comportamientos inteligentes de manera que se optimicen los recursos de IoT y permitan crear sistemas más inteligentes en beneficio de la sociedad.

El propósito de este trabajo fue determinar un modelo de objeto de IoT basado en hardware abierto que permita embeber entidades como son los agentes inteligentes, y modelar de una manera proactiva, inteligente y colaborativa, las capacidades de las que actualmente carecen los objetos de IoT.

La investigación está enfocada al área de la salud que es de gran importancia para el IoT a través del cuidado, monitoreo, diagnóstico y recomendación del estado del paciente en tiempo real a través de agentes. Se utilizaron métodos de investigación bibliográfico y analítico, los cuales posibilitaron la recopilación de información sobre ecosistemas de IoT y agentes. Como resultado se obtuvo un modelo de sistema para agentificar objetos, determinando una taxonomía o estructura jerarquizada y arquitectura para agentes. Posteriormente, se diseñó dicho sistema utilizando dispositivos embebidos Arduino y Raspberry. Para el desarrollo de los agentes software se usó la plataforma JADE (*Java Agent Development Framework*). Para comprobar el funcionamiento del modelo de objeto propuesto se realizaron pruebas instalando el sistema sobre el cuerpo de una persona. De esta manera, fue posible medir métricas de usabilidad lo cual permitió medir la calidad del sistema. El prototipo de objeto de IoT propuesto es usable, transportable y eficaz, por lo que puede ser empleado para el desarrollo de futuras investigaciones en el ámbito del cuidado de la salud de las personas usando IoT.

Palabras claves: Internet de las cosas, agentes inteligentes, Jade, dispositivos embebidos, código abierto, Arduino, cuidado de la salud.

ABSTRACT

Today, the Internet of Things (IoT) is becoming a promising technology. Every day, the way we understand the world is changing and modernizing and simplifying people's lifestyles through objects or devices connected to the network that interact with each other to automate processes. However, there is a need for an object model that allows the modelling of intelligent behavior in a way that optimizes IoT resources and allows the creation of more intelligent systems for the benefit of society.

The purpose of this work was to determine an open hardware based IoT object model that allows to embed entities such as intelligent agents, and model in a proactive, intelligent and collaborative way, the capabilities that IoT objects currently lack.

The research is focused on the area of health that is of great importance for IoT through the care, monitoring, diagnosis and recommendation of the patient's condition in real time through agents. Bibliographic and analytical research methods were used, which enabled the collection of information on IoT ecosystems and agents. As a result, the system model was obtained to identify objects, determining a taxonomy or hierarchical structure and architecture for agents. Later, the system was designed using Arduino and Raspberry embedded devices. The JADE (*Java Agent Development Framework*) platform was used for the development of the software agents. In this way it was possible to measure usability metrics which allowed to measure the quality of the system. The proposed prototype object of IoT was usable, transportable and effective and thus, it can be used for the development of future research in the field of health care of people using IoT, and especially in the care of the diabetic patient.

Keywords: Internet of things, intelligent agents, Jade, embedded devices, open source, Arduino, health care.

INTRODUCCIÓN

Presentación del tema de investigación

En el presente trabajo se plantea determinar un modelo de objeto de Internet de las Cosas (*Internet of Things*, IoT por sus siglas en inglés) basado en hardware abierto y una entidad de tipo de agente software ligero orientado a la gestión de los recursos del objeto, modelado de una manera proactiva, inteligente y colaborativa. De esta forma, la futura generación de objetos conectados a Internet podrá optimizar el uso de sus recursos asociados en ecosistemas heterogéneos donde participen.

En la actualidad muchos de los objetos de IoT, comercializados como objetos inteligentes, carecen de mecanismos sofisticados de inteligencia. En su mayoría, el componente de inteligencia incorporado consiste en definir un conjunto de parámetros sobre los cuales se basan para modificar su comportamiento. No obstante, en los ecosistemas de IoT actuales y futuros, se requiere que dichos objetos puedan ser configurados de acuerdo con las necesidades de las aplicaciones. Ello implica que los objetos del futuro IoT deban distribuirse como productos de código abierto de manera que su *firmware* pueda ser reprogramado en base a las necesidades y el dominio de la aplicación que se está implementando. Asimismo, se requiere que los objetos estén en la capacidad de ejecutar tareas autónomas con poca o nula intervención humana, así como establecer colaboración entre dispositivos.

Planteamiento del problema

El IoT es un campo en auge que está en proceso de desarrollo. De manera global, la sociedad está tendiendo hacia un modelo tecnológico de próxima generación que beneficia la infraestructura de Internet, desplegando dispositivos inteligentes que permite que toda persona, cosa, lugar, servicio o red se encuentre interconectado, basándose en las tecnologías relacionadas con las redes de comunicaciones inalámbricas, electrónica digital, análisis de datos, programación e Inteligencia Artificial [1].

Esta investigación surge con el fin de proporcionar y hacer de manera más fácil la vida de las personas y allá es donde apunta la tecnología del Internet, puesto que

desde ya se cuenta con entornos inteligentes como hogares, ciudades, universidades, industrias y hospitales. En el ámbito del cuidado de la salud, el IoT juega un papel importante ya que ha permitido el desarrollo de sistemas médicos especializados en el cuidado de personas mayores, monitoreo y control remoto de la salud, diagnóstico oportuno, entre otros. Sin embargo, países en desarrollo requieren de tecnologías económicas de IoT que sean configurables de acuerdo con la realidad del medio. En este sentido, muchos de los objetos de IoT que se comercializan hoy en día tiene altos costos y muchos de ellos son cajas negras que no permiten configuraciones específicas. Ello incurre a la problemática de no poder modelar acciones complejas en ecosistemas de IoT inteligentes y socio-colaborativos.

Justificación

La sociedad se está moviendo hacia una estructura de estar continuamente conectados, lo que lleva a posicionar al IoT como una de las tecnologías emergentes en la actualidad. En esta línea, se tiene previsto que las soluciones exitosas junten una gran cantidad de objetos inteligentes identificados por una dirección IP (*Internet Protocol*, por sus siglas en inglés) única para brindar una mayor experiencia a los usuarios finales a través de protocolos de comunicación estándares. Por consiguiente, se espera que el número de objetos conectados a Internet crezca considerablemente y la cantidad de datos que dichos objetos pueden capturar de entornos reales deban ser almacenados y procesados en tiempo real de una forma eficiente e inteligente [2].

El IoT es una tecnología futura de gran importancia para crear un mundo conectado. Se la reconoce como un campo de las comunicaciones y del Internet aplicable en una amplia gama de entornos tales como: la industria 4.0, ciudades inteligentes, hogar inteligente, salud inteligente, agricultura inteligente, entre otros.

Aplicaciones reales del IoT en problemas enfocados en el perfeccionamiento de la salud pública, la educación, gestión del tránsito, el estacionamiento, el transporte público y privado, la cultura, etc., son algunos de los múltiples usos que se le puede dar a los objetos de IoT. No obstante, se hace necesario la disposición

de objetos que se comporten de una forma realmente inteligente, proactiva, colaborativa y autónoma.

Los objetos de IoT comercializados en la actualidad son vendidos como objetos inteligentes o “*smart objects*”. Sin embargo, dichos objetos están aún lejos de comportarse de manera autónoma, proactiva, colaborativa e inteligente. Es por ello, que se requiere crear un modelo de objeto de IoT basado en hardware y software *open source* que permita embeber entidades como son los agentes software, y así adquirir las capacidades de las que actualmente carecen los objetos de IoT.

Delimitación de objetivos

Objetivo general

Construir un modelo de objeto basado en agentes para gestionar los recursos de Internet de las Cosas (IoT) aplicando tecnología de hardware y software libre.

Objetivos específicos

- a. Establecer una taxonomía de objetos basado en agente para gestionar el IoT.
- b. Describir una arquitectura de un modelo de objeto agentificado.
- c. Desarrollar un prototipo de objeto de IoT agentificado basado en hardware y software libre.
- d. Evaluar el modelo de objeto propuesto en un ecosistema de Internet de los Agentes en el área del cuidado de la salud.

CAPÍTULO I: MARCO TEÓRICO

1.1.Marco teórico-conceptual

1.1.1. Internet de las cosas

El IoT se define como una infraestructura de red global dinámica para la sociedad de la información, utiliza servicios proporcionados por los protocolos de Internet y transmisión para comunicarse con dispositivos tanto físicos y virtuales a través de identidades, atributos y sensores capaz de recoger datos para monitorizar y controlar las cosas del mundo real a distancia sin necesidad de la interacción de las personas [3].

Como se ilustra en la Figura 1 , el IoT es un sistema de dispositivos conectados que tienen la capacidad de interactuar entre ellos mediante sensores que a su vez envían o transmiten información exacta.



Figura 1 Ecosistema de IoT [4].

Los recursos del IoT apuntan a crear un mundo donde todos los objetos (conocidos en IoT como objetos o cosas inteligentes) localizados en entornos de la vida cotidiana están conectados a Internet y se comunican entre ellos con un mínimo de intervención humana. El objetivo final del IoT es crear un lugar mejor para los seres humanos, donde los objetos que les rodean saben lo que les gusta, lo que

quieren y lo que necesitan. De esta manera, se busca que el IoT actúe sin instrucciones explícitas.

1.1.2. Aspectos importantes del IoT

Para obtener un conocimiento más claro sobre los componentes del IoT, como el significado real y su respectiva funcionalidad, en los siguientes apartados se presentan seis aspectos esenciales para alcanzar un nivel de eficiencia y de calidad de servicio de los dispositivos interconectados.

- La **identificación** es fundamental para que exista control en las redes de IoT. Existen técnicas de identificación disponibles para el IoT, entre ellos el código de productos electrónicos (EPC) y los códigos ubicuos (uCode) [5]. Sin embargo, el direccionamiento de los objetos de IoT es de alto riesgo para diferenciar entre la identificación del objeto y su dirección. El ID de objeto se refiere a su nombre, como "T1" para un sensor de temperatura particular y la dirección del objeto se refiere a su dirección dentro de una red de comunicaciones. También, los métodos de direccionamiento de los objetos de IoT incluyen IPv6 e IPv4 [5].
- **La detección** permite identificar y recopilar las cuantificaciones claves a la hora del análisis y el direccionamiento para identificar las cosas en Internet. Si bien los sensores son una gran ayuda para reunir información clave para monitorear y diagnosticar las "cosas", éstas tienden a carecer de la capacidad de controlar o reparar esas "cosas" cuando existe algún problema o incidente [6].
- **Las tecnologías de comunicación** en IoT enlazan objetos múltiples para ofrecer servicios inteligentes únicos. Por lo general, las redes de IoT deben trabajar con bajo consumo de energía en circunstancias de enlaces de comunicación ruidosos y con pérdida. Algunos ejemplos de protocolos de comunicación utilizados para la IoT son, *Bluetooth*, *Wifi*, *IEEE 802.15.4*, *Z-wave* y *LTE-Advanced*. Además, algunas tecnologías de comunicación definidas utilizan identificación por radiofrecuencia (RFID), comunicación de campo cercano (NFC) y ancho de banda ultra ancho (UWB). Sin

embargo, RFID es la primera tecnología utilizada para realizar el concepto máquina-a-máquina (M2M) etiqueta y lector RFID [5].

- **Unidades de centrales de procesamiento** hace referencia a microcontroladores, microprocesadores, sistema en chip (SOCs), matriz de puertas lógicas programable en campo (FPGAs) y aplicaciones de software incorporan el "cerebro" y la capacidad de procesamiento computacional del IoT. Hasta la fecha se crearon varias plataformas de hardware para ejecutar aplicaciones de IoT tales como: Arduino, UDOO, Friendl y ARM, Z1, WiSense, Mulle y T-Mote Sky, Intel Galileo, Raspberry PI, Gadgeteer, BeagleBone, Cubieboard [5]. Además, muchas plataformas de software se manejan para proporcionar funcionalidades de IoT, entre estas plataformas, los sistemas operativos son trascendentes, ya que se inicia durante todo el tiempo de activación de un dispositivo [5].
- **Los servicios** relacionados con la identidad son los servicios más elementales y significativos que se utilizan en otros tipos de servicios. Cada aplicación que precisa traer objetos del mundo real al mundo virtual tiene que identificar esos objetos. Los servicios de agregación de información recopilan y este a la vez comprende las mediciones sensoriales que deben procesarse e informarse a la aplicación de IoT [5].
- **La semántica** en IoT tiene como concepto la capacidad de extraer conocimiento inteligentemente por diferentes artefactos para ofrecer los servicios requeridos. La extracción de conocimiento incluye revelar y utilizar recursos e información de modelado. También, contiene el reconocimiento y el estudio de datos para dar sentido a la decisión correcta de facilitar el servicio exacto. La semántica permite al IoT ejecutar procesos basados en el significado de los datos y no únicamente en la sintaxis de éstos [7]. Esto extiende el espectro de búsquedas realizadas en ecosistemas de IoT de manera que las búsquedas sean más precisas [6].

1.1.3. Arquitectura de IoT basada en capas

La arquitectura de IoT especifica el estudio del modelo de capas que se encuentran visualizado en la Figura 2. En esta figura se muestra cómo la arquitectura se encuentra organizada para un correcto funcionamiento. En este contexto, las

especificaciones de las 5 capas son muy importantes. A continuación, se detallan cada una de ellas.

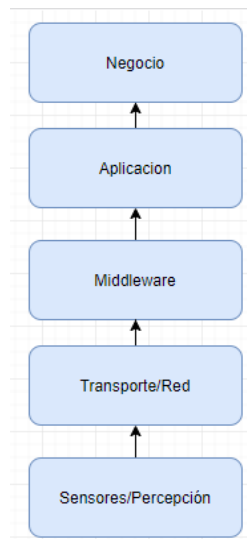


Figura 2 Arquitectura de 5 capas de IoT [8]

- **La capa de percepción** tiene como principal función recopilar y procesar información capturada por los nodos sensoriales (objetos o cosas). Dispositivos como sensores y actuadores están involucrados en esta capa. En esta capa se capturan datos como la temperatura, humedad, la aceleración, entre otros [9].
- **La capa de red / transporte** se encarga de transferir los datos de los sensores mediante redes 3G, 4G, WIFI, Bluetooth, Zigbee, desde la capa de percepción al nivel de capa de servicios o aplicación y viceversa [8].
- **La capa middleware o procesos**, acumula, examina y procesa los datos masivos obtenidos por el sensor o capa de percepción y la trasmite a través de la red a un dispositivo inteligente o sistema que procesa y determina la acción a realizar [8].
- **La capa de aplicación** se centra en la prestación de servicios según las necesidades del cliente. IoT evoluciona debido a las diversas aplicaciones disponibles para los usuarios en función de los datos procesados. Algunas aplicaciones como mayor interés son la domótica o hogar inteligente, la ciudad inteligente, la salud inteligente, el entorno inteligente, etc. se extienden con una evaluación cuidadosa de las peticiones en el mercado [9].

- **En la capa de negocio** defiende el sistema completo de IoT. Los modelos de negocios se inician con la evaluación del rendimiento de las aplicaciones y servicios existentes. El éxito de IoT no solo depende de las tecnologías utilizadas, sino también de cómo se entregan los servicios que provee a los clientes [9].

1.1.4. Software para IoT (middlewares)

Los middlewares pueden facilitar un proceso de desarrollo integrando dispositivos heterogéneos de computación y comunicaciones, y apoyando la interoperabilidad dentro de las diversas aplicaciones y servicios. A continuación, se describen los tipos de middlewares basados en eventos, orientados a servicios, basados en máquinas virtuales, basados en agentes y basados en bases de datos [10].

- **Middlewares basados en eventos.** Se encargan de que tanto los componentes, las aplicaciones y todos los demás objetos que intervienen interactúen a través de los eventos, que son un conjunto de datos escritos cuyos valores específicos describen su comportamiento y se extienden desde los componentes de la aplicación emisora (productores) a los componentes de la aplicación receptora (consumidores). Entre los más conocidos se encuentra middlewares como: Hermes, EMMA, GREEN, RUNES, PRISMAS, SensorBus, Mires [10].
- **Middlewares orientados a servicios.** El paradigma de diseño orientado a servicios crea software o aplicaciones en forma de servicios. La informática orientada a servicios (SOC) se basa en enfoques de arquitectura orientada a servicios (SOA) [11] y se ha utilizado tradicionalmente en sistemas de TI corporativos. Las características de SOC, como la neutralidad tecnológica, el desacoplamiento, la reutilización del servicio, la capacidad de composición del servicio y la capacidad de descubrimiento del servicio [92], también son potencialmente beneficiosas para las aplicaciones de IoT. En esta categoría, se destacan los siguientes middlewares: *Hydra, Sensewrap, MUSIC, TinySOA, SOCRADES, SENSEI, ubiSOAP, Servilla, KASOM, CHOReos MOSDEN, Xively, CarrIoT, Echelon* [10].

- **Middlewares basados en máquinas virtuales (VM).** El diseño de middleware orientado a VM proporciona soporte de programación para un entorno de ejecución seguro para aplicaciones de usuario mediante la virtualización de la infraestructura. Las aplicaciones se dividen en pequeños módulos separados, que se inyectan y distribuyen por toda la red. Entre los middleware basados en este diseño se encuentran los siguientes: *Maté, VM*, Melete, MagnetOS, Squawk, Sensorware, Extended Maté, DVM, DAViM, SwissQM, TinyVM, TinyReef* [10].
- **Middlewares basados en agentes.** En la estrategia de middleware basado en agentes, las aplicaciones se dividen en programas modulares para suministrar la inyección y distribución a través de la red utilizando agentes móviles. Los middlewares que implementan este enfoque son: *Impala, Smart messages, ActorNet, Agilla, Ubiware, UbiROAD, AFME, MAPS, MASPOT, TinyMAPS* [10].
- **Middleware orientado en base de datos.** Estos middlewares usan una red de sensores que es muy familiar con un sistema de base de datos relacional virtual. A través de una aplicación puede consultar la base de datos usando un lenguaje de consulta similar a SQL, que permite la formulación de consultas complejas. Los siguientes middlewares son los más relevantes: *SINA, COUGAR, IrisNet, Sensation, TinyDB, GSN, KSpot, HyCache* [10].

1.1.5. Protocolos de comunicación de IoT

En el dominio del IoT se han planteado varios protocolos de comunicación a través de los cuales los objetos pueden intercambiar de manera lógica y segura datos y mensajes. Entre los protocolos de comunicación más importantes y ampliamente empleados en escenarios de IoT figuran, el protocolo de transferencia de hipertexto, el protocolo de aplicación restringida y el protocolo basado en cola de mensajes transporte de telemetría.

El protocolo de transferencia de hipertexto (HTTP/ HTTPS, por sus siglas en inglés) se describe como un protocolo muy potente que a su vez tiene un alto grado de complejidad en el tamaño del código, superando la memoria de los dispositivos embebidos y los recursos de la red [12]. Los recursos del uso de aplicaciones web actuales se requiere de HTTP/HTTPS para la obtención de información y

actualización del mismo, destinado a la transferencia segura de datos de hipertexto, está basado en Transferencia de Estado Representacional (REST, por sus siglas en inglés) que es una arquitectura capaz de poder hacer el proceso de proporcionar información mediante el Identificador de Recursos Uniforme (URI por sus siglas en inglés); que realiza un control de petición-respuesta asíncrona permitiendo a los clientes acceder a los recursos alojados en los servidores, al mismo tiempo se crea una web distribuida mostrando una arquitectura escalable y flexible [12].

El protocolo de aplicación restringida (CoAP por sus siglas en inglés), se describe como un protocolo de transferencia que posibilita trabajar con dispositivos de recursos limitados, a diferencia de HTTP, este trabaja sobre Protocolo de datagramas de usuario (UDP por sus siglas en inglés) en lugar de usar el estándar de Protocolo de control de transmisión (TCP por sus siglas en inglés). La capa de mensaje define cuatro métodos de petición (*GET*, *POST*, *PUT*, *DELETE*) y los códigos de respuesta son similar a los de HTTP [12].

El protocolo CoAP puede trabajar y ser introducido a HTTP para promover la integración de los dispositivos embebidos [12]. Así mismo que HTTP, CoAP se basa en la arquitectura REST, pero la diferencia es que una “cosa” actúa como cliente y como servidor, es decir que a través del intercambio de mensajes asíncronos un cliente a partir de código de métodos solicita un recurso de un servidor por medio de un URI, el servidor responde con un código de respuesta, la semántica de *request/response* de CoAP incluye los métodos de petición [13].

Finalmente, el protocolo basado en cola de mensajes transporte de telemetría (MQTT, por sus siglas en inglés), es un protocolo de comunicación M2M (machine-to-machine), patrón de tipo mensaje de publicación/suscripción, está diseñado para comunicaciones M2M ligeras en redes restringidas, utiliza TCP/IP como protocolo de transporte y TLS/SSL para la seguridad, por lo tanto, la comunicación entre el cliente y el bróker o corredor está orientada a la conexión. Otra gran característica de MQTT son sus tres niveles de Calidad de Servicio (QoS) para una entrega fiable de los mensajes. Es más adecuado para grandes redes de pequeños dispositivos que necesitan ser monitoreados o controlados desde un servidor back-end en Internet. No está diseñado para la transferencia de dispositivo a dispositivo ni para la multidifusión de datos a muchos receptores. Es un sistema muy básico de protocolo de mensajería que ofrece sólo unas pocas opciones de control [14].

1.1.6. Aplicaciones de IoT

Un sin número de aplicaciones de la vida diaria son inteligentes, pero aún es una utopía que se comunican o comportan como tal. Sin embargo, su integración con otras tecnologías como la Inteligencia Artificial y computación en la nube han hecho originado una gama de innovadoras aplicaciones en diferentes dominios [15]. A continuación, se describen algunos ejemplos introducidos en el mundo real.

- Para un sistema de monitoreo **de tráfico inteligente**, la realización de un sistema apropiado para la identificación automática de vehículos y otros factores de tráfico es muy importante el uso de tecnologías de IoT en lugar de utilizar métodos comunes de procesamiento de imágenes. El sistema inteligente de monitoreo de tráfico brinda una buena experiencia de transporte al aliviar la congestión. Proporciona características como detección de robo, notificación de accidentes de tráfico, menos contaminación ambiental [15].
- La creación de **un ambiente inteligente** para la predicción de desastres naturales como inundaciones, incendios y terremotos es ya viable gracias a las tecnologías innovadoras de IoT. En la actualidad hay un correcto control de la contaminación del aire en el medio ambiente.
- El IoT ya facilita también soluciones de bricolaje y ante esto se han creado **sistemas de casas inteligentes** (puede ser creado por cualquier persona, aunque no sea profesional en el tema) para la automatización del hogar con las cuales se puede controlar de forma remota los aparatos según las necesidades. El monitoreo apropiado de los medidores de servicios públicos, el suministro de energía y agua sirve de gran ayuda al ahorrar recursos y detectar sobrecargas inesperadas o fugas de agua [15].
- **Sistema de agricultura inteligente**, este se encarga de observar la nutrición del suelo, la luz, la humedad, etc. y optimiza la experiencia de la vivienda ecológica mediante el ajuste automático de la temperatura para maximizar la producción. El riego y la fertilización precisa puede mejorar la calidad del agua y a ahorrar los fertilizantes correspondientemente [15].
- **La gestión de recursos como la energía** es una necesidad para lograr un medio ambiente sostenible y *Smart Grid* incorpora un mecanismo

fundamental para su ejecución. Además, la difusión de las fuentes de energía renovables ha generado una profunda modernización del sistema tradicional de distribución eléctrica y en el camino de la distribución de energía. La **red inteligente** se define como un sistema inteligente de distribución eléctrica que entrega flujos de energía de los productores a los consumidores de forma bidireccional [16].

1.1.7. Aplicación del IoT en el área de salud

Previamente, se ha evidenciado la aplicación del IoT en varias áreas del conocimiento. Otra de las áreas en las que se ha aplicado con cierto éxito es el ámbito del cuidado de la salud, esto es, en la gestión de hospitales, emergencias médicas, monitoreo de condiciones fisiológicas de pacientes, cuidado y vida asistida para personas mayores y pacientes crónicos.

En relación a la gestión de los hospitales, éstos se equipan con dispositivos portátiles inteligentes y flexibles integrados con etiquetas RFID que se conceden a los pacientes al instante de su arribo, a través de los cuales no solo los médicos; sino las enfermeras, también pueden observar el ritmo cardíaco, la presión arterial, la temperatura y otras condiciones de los pacientes dentro o fuera del hospital [17].

Todos los días existen muchas emergencias médicas como paros cardíacos entre otros problemas de salud, pero las ambulancias tardan un tiempo considerable en llegar donde el paciente, en este caso para solventar este problema usan la tecnología IoT en las amortías de aviones no conducidos por personas que pueden volar a la escena con el kit de emergencia y así dar un control adecuado, los médicos pueden realizar un seguimiento del problema a pacientes y pueden enviar el avión no tripulado para proporcionar atención médica rápida hasta que llegue la ambulancia [17].

Asimismo, diferentes organizaciones de salud comenzaron a recopilar una gran cantidad de datos producidos por dispositivos IoT, tanto sector médico y sanitario está vigorosamente afectado por IoT gracias a la utilización de dispositivos de detección avanzados que permiten el monitoreo en tiempo real de parámetros médicos y funciones vitales (i.e., temperatura, presión arterial, frecuencia cardíaca, nivel de colesterol). Estos datos recopilados se envían a través de tecnologías de

comunicación estándar (i.e., *Bluetooth*, *ZigBee*, *WirelessHART*, *ISA100*) y se ponen a disposición del personal médico para el diagnóstico y control de la salud de los pacientes a través de las redes de área corporal (BAN) formadas por dispositivos portátiles conectados entre sí, permiten a los médicos continuar la monitorización remota del paciente fuera del hospital [16].

Finalmente, el IoT también puede proporcionar beneficios para mejorar la calidad de vida de los ciudadanos, por ejemplo, ayudar a tener una vida independiente (inclusión electrónica) y tratar indicaciones de estilo de vida para el bienestar de la persona. La inclusión electrónica tiene como objetivo principalmente segmentar problemas específicos de personas, como el envejecimiento o la población discapacitada para permitir ser activos en la sociedad. El monitoreo de la condición de los ancianos, la emulación de la consulta médica en el hogar, asistentes personales disponibles en pantallas de ordenadores o televisión que estimulen a las personas a hacer ejercicio, y los asistentes móviles, son algunos de los servicios o aplicaciones claves en esta área, y pueden ser desarrollados usando tecnologías IoT.

Al monitorear las señales fisiológicas en tiempo real capturadas a través de sensores, el sistema puede llegar a realizar acciones como: activar alarmas médicas (i.e., en caso de caídas detectadas), sugerir una posible hospitalización, diagnosticar demencia (i.e., Parkinson, Alzheimer) en la etapa más temprana observando las desviaciones de los comportamientos normales, entre otras [16].

Como ejemplo de un dispositivo IoT para la vida independiente de una persona adulta está el caso de un prototipo de dispositivos portables equipados con sensores que se basan en determinar la ubicación del adulto tanto en el interior como el exterior de la casa. Además de detectar caídas o movimientos brusco y signos vitales significativos, cualquiera de estas situaciones se activará como una alarma que será enviada a personas que estén a cargo (i.e., médico, familiares). Dichos datos son generalmente enviados a una nube o teléfono móvil para su procesamiento, los cuales sirven posteriormente para emitir alertas a los cuidadores una vez detectadas anomalías [18].

1.1.2. Paradigma basado en agentes

1.1.2.1. Agente software

Se define un agente software como una entidad autónoma, que tiene la capacidad de decidir las acciones que se llevarán a cabo en el ambiente y las interacciones que se establezcan con otros agentes, según sus percepciones y estados internos [19].

En Inteligencia Artificial (IA), un agente inteligente es una entidad autónoma que observa a través de sensores y actúa sobre un ambiente que utiliza actuadores (agente) y dirige su actividad hacia el logro de objetivos (racional). Los agentes inteligentes también pueden aprender o usar el conocimiento para lograr sus objetivos [19]. Por otro lado, existe otra definición donde se dice que el agente es un sistema informático capaz de realizar una acción autónoma y flexible para lograr los objetivos diseñados [20].

1.1.2.2. Características de un agente

La integración de varias herramientas de diseño en un solo sistema holístico, local o distribuido puede requerir que los agentes cumplan con ciertas características. Algunas de estas características son las siguientes:

- **Racionalidad.** Un agente racional es aquel que puede tomar la decisión correcta en cada situación sobre la base de un conjunto de criterios, utilizados para medir el nivel de rendimiento en términos del éxito del comportamiento del agente [19].
- **Autonomía.** Un agente puede tomar la iniciativa y ejercer un grado no trivial de control sobre sus propias acciones. Así, la autonomía es la capacidad de compensar el conocimiento previo parcial o incorrecto, y generalmente se logra mediante el aprendizaje [19].
- **Aprendizaje.** Esta característica se refiere a que los agentes tienen la capacidad de aprender su entorno en base a experiencias al observar ya sea el entorno en que lo rodea y actuar en consecuencia para predecir el futuro [19].
- **Reflexivo.** Los agentes reflexivos operan por medios simples que se compone de un intérprete de información recibida de sus sensores para hacer

coincidir las reglas de condición-acción (IF-THEN) basadas en la información de percepción actual. del entorno percibido [19].

- **Colaborativo.** La colaboración habilita a los agentes para que sean capaces de cooperar entre aquellos que tengan un mismo interés para alcanzar ciertos objetivos que por sí solos no pueden alcanzar [19].

1.1.2.3.Arquitectura de agentes

La arquitectura determina la organización de los mecanismos necesarios para la creación de un agente que pueda reaccionar ante eventos como estímulos. Además, este tipo de sistemas definen los agentes en un conjunto de módulos que interactúan entre sí para lograr una correcta funcionalidad. Son tres las arquitecturas estudiadas para la realización del agente. Estas arquitecturas se describen como:

- **Arquitectura deliberativa.** Una arquitectura de agente deliberativo tiene un alto grado de complejidad. Se emplean a través de un modelo para la representación del conocimiento de forma simbólica. Para lograr sus objetivos sigue una planificación paso a paso, por este motivo al agente a la hora de razonar demora en tomar una decisión porque recopila y evalúa todos los comportamientos posibles basados en estados mentales (creencias, deseos, intenciones y planes) [21] permitiendo establecer el mejor comportamiento que se ajuste para lograr sus objetivos [22].
- **Arquitectura reactiva.** Una arquitectura de un agente reactivo es de menos complejidad que la deliberativa al momento de implementarse, se caracteriza por no tener un modelo simbólico para su razonamiento, al momento de lograr su objetivo ejecuta un comportamiento basado en percepciones, puede que no sea el más eficiente porque toma decisiones rápidamente, realizando poca o ninguna búsqueda a la hora de reaccionar o dar una respuesta [22].
- **Arquitectura híbrida.** La arquitectura híbrida combinan los aspectos del comportamiento reactivo y deliberativo, aprovecha la velocidad del comportamiento reactivo en reaccionar a los eventos que tengan lugar en el

entorno y que no requiera un mecanismo de razonamiento complejo sino que tenga la capacidad de tener un modelo simbólico de razonamiento [23].

1.1.2.4.Sistema multiagentes

Los Sistemas Multiagentes (*Multiagent Systems*, MASs por sus siglas en inglés) son sistemas distribuidos y asíncronos. Es una tecnología utilizada para resolver problemas muy difíciles que no pueden ser resueltos por un solo agente, solucionando el problema dividiéndolo en subproblemas que son resueltos por agentes de menor complejidad que colaboran entre sí [24].

Las características más importantes de los MASs, incluyendo la eficiencia, el bajo costo, la flexibilidad y la fiabilidad la convierten en una solución eficaz para resolver tareas complejas. Su eficiencia proviene de la división del trabajo inherente al MAS por la cual una tarea compleja se divide en múltiples tareas más pequeñas, cada una de las cuales se asigna a un agente distinto. Naturalmente, los gastos generales asociados, por ejemplo, el procesamiento y el consumo de energía, se amortizan a través de los múltiples agentes, lo que a menudo resulta en una solución de bajo costo comparado con un enfoque en el que todo el complejo problema debe ser resuelto por una sola entidad poderosa.

Cada agente puede resolver la tarea asignada con cualquier nivel de conocimiento predefinido, lo que introduce una gran flexibilidad. El sistema distribuido y la naturaleza de la resolución de problemas adoptada en los MASs también imparte una alta fiabilidad, en caso de fallo del agente, la tarea puede ser fácilmente reasignados a otros agentes [25].

Entre las principales ventajas de usar sistemas multiagentes figuran las siguientes: (i) tienen en cuenta la naturaleza específica de la aplicación y ambiente, (ii) pueden ser compuestas e investigadas las iteraciones locales entre individuo y (iii) se organizan en subcapas o componentes a la hora de presentarse dificultades de modelar y de calcular.

1.1.2.5. Marco de desarrollo JADE

El marco de desarrollo de agentes en el lenguaje Java (JADE) es un marco de software que consiste en un middleware para permitir la creación y ejecución de

sistemas multiagentes distribuidos [26]. Fue iniciado por Telecom Italia y la Universidad de Parma en 1998, motivados por validar las especificaciones de la Fundación para Agentes Físicos Inteligentes (FIPA, por sus siglas en inglés), a través de una implementación concreta y conforme, cabe mencionar que puso gran énfasis en la usabilidad y la simplicidad de sus API, por su uso no sólo del equipo de programadores sino también por otros programadores sin amplia experiencia en las especificaciones FIPA y la teoría de sistemas multiagentes [27].

Con el fin de crecer y darse a conocer en comunidades del FIPA y de los sistemas de agentes se usó en productos comerciales, en 2000 JADE se convirtió en Código Abierto distribuido por Telecom Italia bajo la LGPL (GNU Licencia de la Licencia Pública General Menor); además es utilizado por una comunidad heterogénea de usuarios como herramienta tanto para apoyar las actividades de investigación como para crear aplicaciones reales [27].

JADE trabaja conjunto con FIPA, adicionalmente tal y como se visualiza en la Figura 3, proporciona un gráfico para desplegar y depurar un MASs, es decir múltiples agentes que utilizan un Lenguaje de Comunicación FIPA (FIPA-ACL) y el agente para intercambiar mensajes ya sea dentro de su propia plataforma o con una exterior distribuida. Cada instancia de ejecución de JADE se llama contenedor. Un contenedor es un hilo independiente que a su vez contiene un conjunto de contenedores [28].

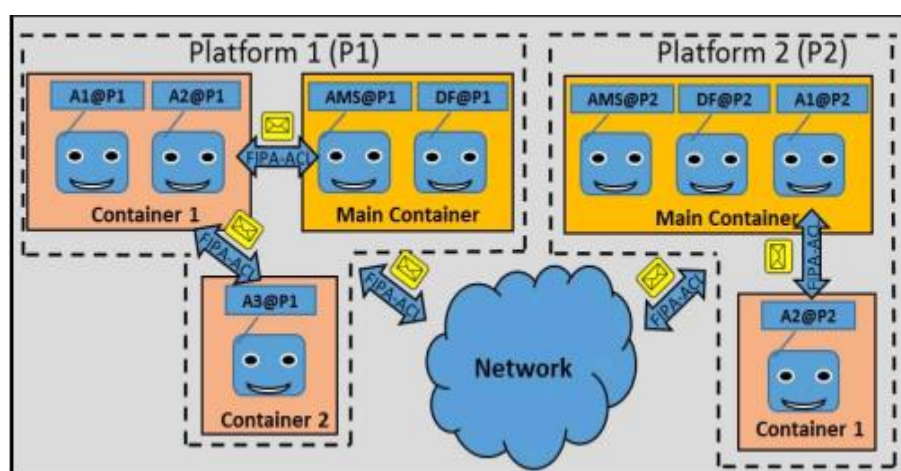


Figura 3 Arquitectura del marco de desarrollo de agentes JAVA (JADE) [28].

Los agentes se comunican mediante el intercambio de mensajes asíncronos, un modelo de comunicación casi universalmente aceptado para las interacciones distribuidas y poco acopladas, es decir, entre entidades heterogéneas. Para comunicarse, un agente sólo envía un mensaje a un destino. Los agentes se identifican con un nombre, por lo que la operación de envío no necesita la referencia del objeto de destino y como consecuencia directa, no hay dependencia temporal entre los agentes comunicantes. El emisor y el receptor pueden estar disponibles en diferentes momentos [27].

El conjunto de todos los contenedores se llama plataforma y proporciona una capa homogénea que oculta a los desarrolladores de aplicaciones su complejidad y la diversidad de los niveles subyacentes (hardware, sistema operativo, tipo de la red, una máquina virtual Java (por sus siglas en inglés Java Virtual Machine, (JVM)), cada plataforma debe contener un contenedor principal que a su vez contiene dos agentes necesarios que son: un Sistema de Gestión de Agentes (AMS) y un Facilitador de Directorio (DF) [27].

AMS proporciona un ID único para cada agente bajo su plataforma, para ser usado como una agente dirección de comunicación, mientras que el DF está anunciando los servicios que cada agente bajo su plataforma puede ofrecer, a fin de facilitar el intercambio de servicios de los agentes, de manera que cada agente pueda obtener su objetivo específico [28]. Además, JADE incluye tanto las bibliotecas (es decir, las clases de Java) necesarias para desarrollar agentes de aplicación, y el tiempo de ejecución del entorno proporciona los servicios básicos que deben estar activo en el dispositivo antes de que los agentes puedan ser ejecutados [27].

1.1.3. Agentificación del IoT

El proceso de agentificación del IoT origina por la necesidad de que los objetos de IoT puedan escalar su comportamiento; es decir, son solamente entidades pasivas en las que el usuario es quien mayoritariamente controla el estado de los objetos, a ser entidades proactivas que puedan tomar por si mismos sus propias decisiones. En este sentido, en [29] se planteó este nuevo paradigma el cual puede ser implementado a través de dos enfoques, esto es, agentificación a través del embebido de agentes sobre los objetos de IoT y el desarrollo de sistemas

multiagentes que controlan los recursos de los objetos de IoT desde servidores externos a los propios dispositivos.

1.1.3.1. Agentificación basada en agentes embebidos

El proceso de embeber un agente en la arquitectura del dispositivo físico puede tomar dos formas. El primero de estos reside en una integración directa y se produce cuando el objeto es de un tipo de hardware abierto y admite la actualización de software que dirige su comportamiento. Esto implica cargar un firmware que puede actualizarse dinámicamente. Sin embargo, la segunda forma de encajar agentes en un objeto de IoT es mediante componentes alternativos; es decir, integrando un conjunto de sensores/actuadores, formando así nodos pasarelas o llamados *gateway* de IoT a través de un componente SBC (computadoras de una sola placa) [30].

Las computadoras de una sola placa (SBC) están integradas con sensores y TCP/IP incorporado y las funcionalidades de seguridad se usan regularmente para realizar productos de IoT (por ejemplo, Arduino Yun, Raspberry PI, BeagleBone Black etc.) En la Figura 4, se visualiza la arquitectura de una entidad agente de cosa (*agent of things*) que está asociada directamente con un objeto de IoT para proveer los datos requeridos por los clientes [31].

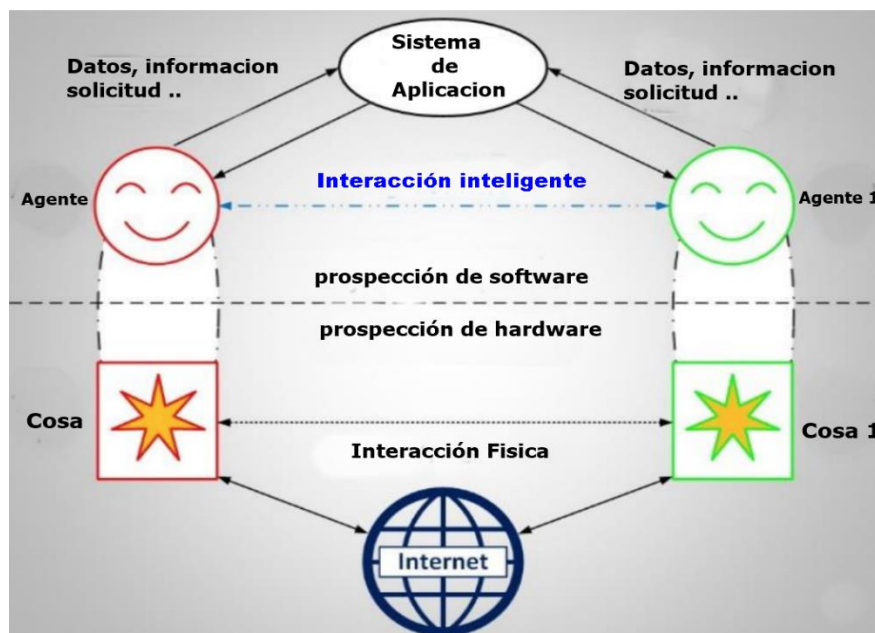


Figura 4 Arquitectura de una entidad agente de cosa para agentificar el IoT [31]

Se hace ahínco en que la placa única u ordenador de placa simple (SBC) de bajo coste están generalmente equipadas con tecnología de información y comunicación, estos objetos pueden almacenar su contexto, están conectados en red, pueden acceder a servicios de Internet e interactúan entre ellos [5]. Entre los principales componentes SBC existentes en el mercado y ampliamente extendidas entre los desarrolladores de IoT figuran los siguientes: Raspberry Pi, BeagleBone, Uddo. Sin embargo, en el mercado existen muchas otras más.

- **Raspberry Pi.** Es una placa de computadora pequeña o también conocida como microcomputadora, potente, bajo costo, y orientada a la educación, funciona de la misma manera que una PC, se requiere de recursos como teclado, una unidad de visualización y una fuente de alimentación, es una excelente plataforma para comunicarse con muchos dispositivos. Los componentes que conforman el sistema, sus unidades centrales y de procesamiento de gráficos, hardware de audio y comunicaciones junto con 256 MB (Modelo A) - 512 MB (Modelo B) de chip de memoria, están integrados en un solo componente. El procesador de Raspberry Pi es un sistema en un chip de 32 bits y 700 MHz, que se basa en la arquitectura ARM11 y se puede aumentar la velocidad para alcanzar mayor potencia [32]. La Raspberry Pi, usa un sistema operativo de Linux conocida como Raspbian. Hay una amplia gama de proyectos experimentales en los que el SBC se usa para monitorear las condiciones fisiológicas de personas [32].
- **BeagleBone Board** es una computadora de placa única basada en procesadores Texas Instruments de baja potencia, que utiliza el núcleo ARM Cortex-A8, ejecuta en un sistema operativo como Linux / Android 4.0. La principal diferencia entre este y Arduino es que puede ejecutar un pequeño sistema operativo, convirtiéndolo prácticamente en un miniordenador que puede ejecutar programas en estos sistemas operativos. BeagleBone está diseñado para trabajar a un nivel mucho más alto y tiene mucha mayor capacidad de procesamiento que Arduino [32].
- **Udoo Board** es una pequeña PC que se usa con Android y Linux OS, con una placa integrada compatible con Arduino. Es una poderosa placa de diseño de prototipos para el perfeccionamiento y de software. Udoo reúne

una microcomputadora con los puertos de comunicación más usuales (Ethernet, WiFi, USB, HDMI, SATA, entrada / salida digital y analógica) y un microcontrolador con un pin out estándar para aplicaciones de creación rápida de prototipos. Por lo tanto, Udoos es una plataforma de hardware abierta y de bajo costo equipada con un procesador ARM i.MX6 Freescale y una sección compatible con Arduino Due basada en el procesador ATMEL SAM3X ARM. En resumen UDOO busca juntar los mejores elementos de Raspberry Pi y Arduino en una sola microcomputadora [32]

- **Arduino.** En realidad, no es un componente SBC, pero constituye una plataforma informática física de código abierto ampliamente usada en IoT que está basada en una placa de microcontrolador simple y un ambiente de desarrollo para escribir software para la placa. Recibe información de una variedad de sensores y puede controlar luces, motores y otros actuadores. El microcontrolador se puede programar usando el lenguaje de programación Arduino y el entorno de desarrollo integrado Arduino (IDE) [32].

1.1.3.2. Agentificación basada en MASs

Este segundo enfoque para realizar la integración de IoT y agentes se centra en la creación de uno o varios MASs cuyos agentes están especializados en gestionar los objetos conectados, así como los servicios de IoT y los recursos mancomunados. Básicamente el modelo se centra en la adaptación, capacidades sociales y de colaboración. Se usan middlewares especializados para poder proporcionar una interfaz de comunicación de bajo nivel con los objetos de IoT se comunica a los agentes que forman el MAS con los dispositivos físicos que conforman la red de objetos IoT [30].

En la Figura 5, se ilustra un esquema general orientado para agentificar el IoT mediante sistemas multiagentes. En dicha arquitectura se observa cómo los agentes que controlan el IoT se ejecutan en uno o varios servidores externos. Asimismo, es posible observar que para acceder al estado o los recursos de los objetos de IoT es necesario incluir un middleware especializado para IoT, esto es, un middleware que adopte los servicios web para abstraer las comunicaciones con los dispositivos de IoT. De esta manera, las acciones ejecutadas por los objetos de IoT están

controlados directamente por los agentes software. Por tanto, es posible que los objetos de IoT, adquieran comportamientos proactivos, sociales e inteligentes.

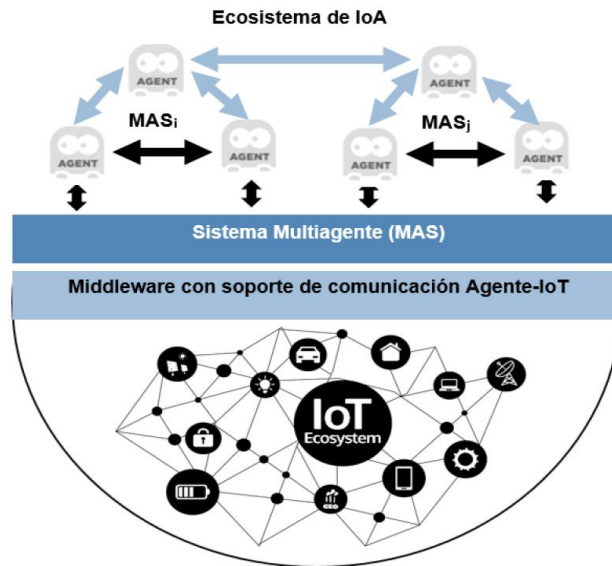


Figura 5 Arquitectura para agentificar el IoT usando sistemas multiagentes [33]

1.2. Antecedentes

En la actualidad existe una visión ampliamente compartida sobre el gran potencial de Internet de las Cosas (IoT) para impulsar la innovación en muchos dominios de la salud. Por otra parte, distintos estudios de prospectiva identifican al sector sanitario como el mercado más importante de IoT en el futuro [34].

Los primeros indicios sobre la investigación en el área de salud como antecedente al IoT trata del estudio de grandes volúmenes de datos de pacientes que se recogen día a día en los hospitales mediante el uso de dispositivos médicos autónomos, incluidos los signos vitales. Dichos datos a veces se almacenan en hojas de cálculo, que no forman parte de los registros de salud electrónicos de los pacientes y, por lo tanto, es difícil para los cuidadores combinar y analizar dichos datos. Un posible enfoque a la solución para superar estas limitaciones es la interconexión de dispositivos médicos a través de Internet mediante una plataforma distribuida de IoT. Este enfoque permite que los datos de diferentes fuentes se combinen para diagnosticar mejor el estado de salud del paciente e identificar posibles acciones de anticipación (medicina preventiva). Se consideran enfoques heurísticos comunes, como los sistemas de puntuación de alerta temprana

ponderados, y se analiza la posibilidad de emplear algoritmos inteligentes. Como resultado, la investigación propone posibles direcciones para combinar los datos del paciente en las salas de hospital para mejorar la eficiencia, permitir la optimización de los recursos y minimizar el deterioro de la salud del paciente. Se concluye que un enfoque centrado en el paciente es crítico, y que el paradigma de IoHT continuará proporcionando soluciones más óptimas para el manejo de pacientes en salas de hospital [35].

De la misma manera en otra investigación [36] se expuso un caso de estudio aplicado al IoT en el cuidado de la salud, tiene como objetivo principal monitorizar en tiempo real la diabetes a través de la respiración, está basado por un diagnóstico de sensores del sistema de Internet de los Agentes, tiene como problemática que cuando un cuerpo tiene muy poco insulina, significa que las células del cuerpo no pueden tomar suficiente azúcar (glucosa) de la sangre. Así que, las cetonas que son químicos que aparecen en el cuerpo cuando la grasa corporal se utiliza como energía en lugar de la glucosa, si se mide este nivel de acetona se puede ayudar a controlar y monitorizar la condición de los pacientes diabéticos, ante un gran número de cetonas significa que la diabetes está fuera de control.

Para poder cumplir con el objetivo se construyó un prototipo llamado “Diab-check” que es un sistema de Internet de las Cosas (IoT) para la prueba de aliento y monitorización de la condición de los pacientes diabéticos, consiste en hardware y software, se usaron sensores de gas (FIGARO TGS822), temperatura (DT11) y humedad además de un microcontrolador como lo es Arduino Uno. El agente, se implementó mediante la plataforma JADE por ser un software libre, estable y de código abierto. Como resultado se realizó un diagnóstico en base al motor de inferencia del Lenguaje Prolog, donde se definieron varias reglas de inferencia para hallar un diagnóstico más preciso. En conclusión la arquitectura empleada controla los dispositivos y materiales conectados, proporciona un diagnóstico, además, asegura la comunicación e interacciones entre diferentes cosas, agentes y sistemas externos [36].

En la misma línea de investigación, [37], y con un enfoque muy parecido aplicando agentes inteligentes en un ecosistema de IoT en la salud y desarrollado en la plataforma JADE, el objetivo fue el desarrollo de un sistema de monitoreo inteligente para monitorear el ritmo cardíaco de los pacientes de forma automática.

La problemática con la que se hallaron fue que el aumento de las enfermedades cardíacas causan millones de muertes cada año, en especial a las personas que están en una etapa de envejecimiento y el alto costo de la atención médica, además de los problemas de seguridad ante el robo o pérdida de datos sensibles para los pacientes, si los datos automáticos no se verifican ni se gestiona correctamente, la información personal y médica de los pacientes puede ser manipulada, utilizada o comprometida en la falta de no tener una vigilancia en tiempo real.

Por todas estas razones se creó un prototipo para un dispositivo de control remoto de la frecuencia cardíaca que se puede llevar puesto, el hardware que se usó para la creación fue una Raspberry, un sensor *Heart Rate* y un convertidor de analógico a la digital (ADS1115). Se usó una arquitectura de IoT de 5 capas y como resultado se evidenció que el ritmo cardíaco del paciente puede ser monitoreado por el médico sin tener que visitar al paciente; esto permitirá conocer la frecuencia cardíaca del paciente en condiciones reales (en casa, en el trabajo, etc.), se concluyó que este sistema puede reducir los costos de la atención de la salud al reducir visitas al consultorio del médico, hospitalizaciones y pruebas de diagnóstico de procedimiento y por último ofrecer una mejor seguridad de los datos a través de un encriptación [37].

Se ha podido evidenciar en el entorno de la atención médica los pocos avances significativos en el área de tecnologías de redes, hardware y software, es decir, que ante un gran número de problemas en este entorno el aporte no ha sido el esperado, por estas razones en [38], los autores se han enfocado en estas tres áreas mencionadas anteriormente permitiendo aplicar tecnología para abordar mucho de estos problemas como es el caso del monitoreo de pacientes en tiempo real. El objetivo de esta investigación estuvo enfocado en diseñar y desarrollar soluciones tecnológicas innovadoras para ofrecer un entorno de atención médica más proactivo y confiable, contiene componentes como sistemas de múltiples agentes (MASs) que están diseñados para realizar actividades de Monitoreo Remoto de Pacientes (RPM) de forma autónoma.

Como metodología se seleccionó un marco para evaluar si el dominio de la RPM podía generar aplicaciones personalizadas que pudieran lograr el objetivo declarado de ser receptivas y flexibles dentro del entorno. El resultado obtenido fue que las aplicaciones pueden ampliar el marco de las siguientes formas: identificación,

recopilación, almacenamiento, recuperación, visualización, monitoreo, detección de anomalías, notificación de recursos y reconfiguración dinámica también se observó que su diseño ayudó a una monitorización más proactiva del paciente. A través de estos sistemas experimentales, se divisaron anomalías en tiempo real con agentes enviando notificaciones rápidamente a los proveedores de salud. Se concluye que el costo-beneficio de la construcción de un sistema de software IoT en la salud demostró el valor del enfoque, permitiendo generar aplicaciones en este dominio de manera más oportuna [38].

1.3. Bases legales

El Estado nacional instaura el marco legal y también las unidades de gestión y/o controles necesarios para el funcionamiento de las Tecnología de la Información en el sector salud. En Especial da importancia la función del Ministerio de Salud Pública (MSP), sus respectivas dependencias, institutos y entidades descentralizadas, así como determinados actores dentro del marco de la informática, la comunicación y la interoperabilidad entre otros [39].

La Ley Orgánica de Salud establece la responsabilidad del MSP, mediante el artículo 6, de participar con el organismo competente, en la investigación y el desarrollo de la ciencia o tecnología en salud, salvaguardando la vigencia de los derechos humanos y respetando los principios bioéticos [39]. En este sentido, es importante señalar que el trabajo no se ha enfocado en probar el prototipo desarrollado con pacientes reales, las pruebas se realizaron solo en laboratorio.

Por otro lado, el artículo 32 determina que la salud es un derecho que avala el Estado, cuya elaboración se vincula al ejercicio de otros derechos, entre ellos, el derecho a la alimentación, educación, cultura física, trabajo, seguridad social, ambientes sanos y otros que sustentan el buen vivir [40]. El prototipo desarrollado en este trabajo es una alternativa para que los pacientes con patologías cardiorrespiratorias puedan automonitorearse y de esta manera gestionar mejor su salud y la calidad de vida desde sus hogares.

Finalmente, el artículo 385 menciona que el sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en el marco del respeto al ambiente, la naturaleza, la vida, las culturas y la soberanía, tendrá como finalidad: desarrollar

tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejoren la calidad de vida y contribuyan a la realización del buen vivir [40]. A través de esta investigación académica se ha desarrollado una línea que puede ser mejorada en el campo de la salud integrando nuevos componentes electrónicos que conlleven a monitorear otras enfermedades que requieren monitoreo continuo.

CAPÍTULO II: METODOLOGÍA

En este apartado se describen los aspectos metodológicos relacionados con el estudio que en líneas generales dan detalles sobre cómo se realizó la investigación. Asimismo, se describen las variables e indicadores que determinaron el funcionamiento del prototipo desarrollado.

2.1. Delimitación de la investigación

La investigación no es considerada como una investigación de campo, motivo por el cual no existe una delimitación espacial. Sin embargo, en lo concerniente a la delimitación temporal, se puede afirmar que la investigación estuvo delimitada al año 2020. Durante todo el año se realizaron tareas de construcción del prototipo y pruebas de laboratorio.

Por otro lado, es importante señalar que la investigación también estuvo delimitada a estudiar solamente dos condiciones fisiológicas que están relacionadas con las enfermedades cardio respiratorias, esto es, temperatura corporal y ritmo cardio respiratorio.

2.2. Tipo de investigación

Para el desarrollo de esta investigación se trabajó bajo un enfoque mixto que brindó una integración sistemática de los métodos cualitativos y cuantitativo con el fin de lograr obtener una perspectiva más amplia y profunda para el desarrollo y construcción del modelo de objeto de IoT agentificado, planteado en este estudio, y así cumplir con los objetivos propuestos.

El estudio fue de tipo cualitativo debido a que se analizaron las tecnologías compatibles con el desarrollo de dispositivos hardware de IoT, los sensores compatibles para medir datos de condiciones fisiológicas (cardio respiratorias), y los estándares inalámbricos para crear un dispositivo de IoT capaz de medir dichos datos de manera fiable. Además, permitió explorar las características de los agentes software y los aspectos a tomar en cuenta para su implementación. Se analizó con detalles el alcance y las limitaciones de la herramienta JADE; ampliamente usada para el desarrollo de agentes y sistemas multiagentes.

Por otro lado, el estudio fue de tipo cuantitativo considerando que llevó a cabo un diseño experimental de un prototipo de objeto de IoT agentificado orientado al cuidado de la salud, a partir del cual se pudo monitorear las condiciones fisiológicas (medidas mesurables por medio de sensores) de las personas que participaron en el ensayo, y en base a ello se pudo también establecer un diagnóstico para mejorar la calidad de vida de pacientes que padezcan de enfermedades cardio respiratorias.

2.2. Métodos de investigación

La investigación planteada es de carácter experimental debido a que el modelo de objeto de IoT agentificado, en conjunto con el modelo de despliegue y comunicación, se probó en un ambiente real controlado, orientado al cuidado de la salud de una persona en base a un conjunto de variables relacionadas con el monitoreo de la temperatura corporal y la medición del ritmo cardio respiratorio.

También se utilizó una metodología exploratoria con un enfoque documental o bibliográfico con el cual se recogieron y analizaron datos de estudios relacionados sobre prototipos de dispositivos de IoT orientados a monitorear la salud, para en base a ellos plantear una taxonomía de objetos basado en agentes que ayudaran a gestionar las acciones a ejecutar en IoT (i.e., diagnóstico). Este método permitió también determinar los procesos sobre los cuales se basaría el funcionamiento del prototipo creado.

Finalmente, se aplicó también el método inductivo y deductivo. El primero, que parte de lo específico a lo general, permitió partir de generalizaciones sobre el uso de agentes en IoT para aplicarlas en el dominio de la salud considerando un novedoso paradigma como es el IoT agentificado, conocido también como el Internet de los Agentes [33]. El segundo, que va de lo general a lo específico, con el objetivo de extraer conclusiones lógicas y válidas a partir de un conjunto de datos proporcionados por las bases teóricas analizadas y en base a los datos de la evaluación del prototipo de objeto de IoT agentificado.

2.3. Variables sujetas a investigación

Para direccionar la investigación, se plantearon tres variables sujetas a estudio. Como se puede observar en la Tabla 1, se ha evaluado la usabilidad del sistema en términos de facilidad para que una persona use el objeto de IoT, el funcionamiento del sistema para validar que cumple con el objetivo planteado, y la portabilidad a fin de verificar si el objeto creado puede ser empleado en distintos escenarios (i.e., hogar, parque).

Tabla 1 Variables para la calidad del sistema

Variables	Indicadores	Tipo de variable	Entidades
Usabilidad	<ul style="list-style-type: none">• Nivel de facilidad para instalación• Nivel de facilidad ejecución de tareas de usuario• Nivel de facilidad de uso del prototipo	Cuantitativa	Paciente
Funcionalidad	<ul style="list-style-type: none">• Grado de cumplimiento del objetivo• Eficacia del sensado• Eficacia de la comunicación entre agentes• Grado del cumplimiento de los comportamientos de agentes acorde a lo programado	Cualitativa	Sistema de IoT agentificado
Portabilidad	<ul style="list-style-type: none">• Facilidad de transporte• Grado de ubicuidad• Mecanismo de alimentación de energía• Grado de facilidad para reemplazo de sensores	Cualitativa	Prototipo

2.4. Técnica de investigación e instrumentos de recolección de datos

En esta investigación se usaron fichas de observación para registrar los indicadores (métricas) que permitieron medir las variables sujetas a estudio en relación con el prototipo desarrollado. Estas métricas fueron analizadas en investigaciones anteriores en las cuales evaluaban aspectos que comprobaran el correcto funcionamiento de un prototipo en un ecosistema de IoT para la salud.

Las fichas con la que se realizaron las pruebas se pueden observar en el apartado de Anexo A (evaluación de usabilidad), Anexo B (evaluación de funcionalidad) y Anexo C (evaluación de portabilidad) del prototipo desarrollado.

2.5. Normas éticas

La información obtenida es fruto de la investigación que se obtuvo con responsabilidad y veracidad de manera que se cumplió con los lineamientos del Reglamento de Grados de la Pontificia Universidad Católica Sede Esmeraldas.

Se respetó la Ley de propiedad intelectual que refiere a la protección de las creaciones intelectuales, de acuerdo a los términos establecidos por las leyes ecuatorianas, entre ellas, la Ley Orgánica de Regulación y Control del Poder del Mercado y la Ley Orgánica de Comunicación [41]. Se respetaron cada una de las ideas y posturas de los autores citados. Además, es importante señalar que, para evitar el uso de herramientas de software no autorizadas, se usó tecnologías de software y hardware libre.

Finalmente, las pruebas realizadas únicamente en laboratorio se realizaron con el consentimiento informado de las personas que apoyaron el desarrollo de los experimentos. Los datos fueron anonimizados y se usaron exclusivamente para fines científicos y académicos.

CAPITULO III: RESULTADOS

En este apartado se presenta una taxonomía dirigida a establecer una recomendación sobre cómo organizar los agentes software para el desarrollo de sistemas orientados al cuidado de la salud, específicamente el apoyo a personas con problemas cardio respiratorios. Dichos agentes, corresponden a unidades que operan en ecosistemas de IoT. Además, se describe la arquitectura y un prototipo de un objeto de IoT enfocado en el diagnóstico del estado de pacientes de acuerdo con condiciones como la temperatura corporal y su condición cardio respiratorio, ambas, medidas mediante sensores fisiológicos. La gestión de dichos datos se realiza a través de agentes que se comunican entre sí para realizar recomendaciones.

3.1.Revisión de la literatura

3.1.1. Sistemas de IoT en salud

En la Tabla 2, se muestran los estudios seleccionados enfocados a la gestión de entornos de IoT enfocados en el ámbito de la salud.

Se realizó un análisis profundo de las 10 investigaciones listadas en la Tabla 1, a partir del cual se determinó cuáles eran los principales sistemas propuestos en la literatura para el desarrollo de aplicaciones de agentes e IoT en el dominio de la salud. En términos generales, se analizó el tipo de condiciones fisiológicas que se monitorearon en las aplicaciones propuestas por investigadores de la línea en la que se enfoca la investigación, y también, la tecnología de IoT que emplearon para medir las condiciones fisiológicas de los pacientes.

Es importante señalar que los estudios plantearon diferente forma de gestionar los equipos médicos y enfatizaron en la monitorización en tiempo real, gestión de los datos, diagnósticos y recomendaciones orientadas a pacientes. Asimismo, se describieron metodologías de diseño, diagrama de bloques y la descripción de las arquitecturas de los sistemas propuestos. Finalmente, y en vista de que el área de la salud es muy delicada y susceptible, se evidenció que los sistemas contemplan en su mayoría mediciones de rendimiento, por lo que es indispensable la utilización de la tecnología de la IoT de una forma rápida y segura en relación con la transmisión de datos.

Tabla 2 Selección de estudios de IoT aplicados al cuidado de la salud

N°	Año	Título	Monitoreo	Tecnología	Fuente	Ref.
1	2016	<i>IOT Based Vitality Measurement System</i>	Temperatura, Presión Arterial, ECG.	E Monitoreo de la salud y red de sensores inalámbricos en un Sistema Arduino	IJSCM	[42]
2	2017	<i>Health Monitoring Systems using IoT and Raspberry Pi-A Review</i>	Temperatura, Presión Arterial, Ritmo cardíaco	Sistema de vigilancia de la salud basado en Raspberry-Pi usando IoT	IEEE	[43]
3	2017	<i>HEMAN: Health Monitoring and Nours an IoT based e-Health Care System for Remote Telemedicine</i>	Temperatura, Presión Arterial, Ritmo Cardíaco, Pulso Oxígeno	Sistema Arduino de atención sanitaria basado en la IoT	IEEE	[44]
4	2018	<i>Health Monitoring System using Internet of Things</i>	ECG, latidos del corazón, respiración, temperatura y presión arterial	Sistema de un controlador Arduino Mega de vigilancia de la salud IoT	IEEE	[45]
5	2019	<i>IoT Health care Monitoring and Tracking: A Survey</i>	Temperatura, Presión Arterial, Ritmo cardíaco	IoT basado en Raspberry PI para el monitoreo de la salud	IEEE	[46]

6	2018	<i>Internet of Things Based Patient Health Monitoring System Using Wearable Biomedical Device</i>	Temperatura, Ritmo cardiaco	Sistema embebido en un Arduino con dispositivos de IoT en la atención médica	IEEE	[47]
7	2017	<i>Monitoring Patient's Health with Smart Ambulance system using Internet of Things (IOTs)</i>	Temperatura, Ritmo cardiaco	Un Sistema Arduino y sensores inteligente de salud y ambulancias mediante IoT	IEEE	[48]
8	2017	<i>Real Time Patient Monitoring System based on Internet of Things</i>	ECG, latidos del corazón, respiración, temperatura y presión arterial, Presión de Oxígeno	Sistema basado en IoT usando controlador Arduino y sensores para la salud	IEEE	[49]
9	2016	<i>Design of a Low-Cost Smart Chair for Telemedicine and IoT based Health Monitoring</i>	Presión oxígeno, latidos del corazón, temperatura	Sistema de vigilancia de la salud basado en la telemedicina a través de Arduino y sensores biomédicos	IEEE	[50]
10	2017	<i>IoT Clinic-Internet based Patient Monitoring and Diagnosis System</i>	ECG, latidos del corazón, respiración, temperatura y presión arterial, Presión de Oxígeno	Sistema de diagnóstico médico remoto basado en un microcontrolador STM32F429	IEEE	[51]

Por otro lado, la Tabla 2 muestra también que las condiciones más comunes de ser monitoreadas en el área de la salud fueron realizadas para verificar que el estado del paciente no tenga riesgo alguno. Los escenarios donde estas aplicaciones han sido implementados principalmente en hogares de ancianos, de manera que las personas enfermas no necesiten visitar al médico e ir al hospital con regularidad para conocer la condición de su enfermedad (i.e., tomar temperatura, conocer ritmo cardiaco, entre otras). Para monitorear algunas de estas condiciones, se usaron sensores ECG (latidos del corazón), respiración, temperatura y presión arterial, presión de oxígeno. Estos sensores se integraron principalmente con controladores como Raspberry Pi y Arduino. A partir de esta integración se crearon sistemas para el cuidado de la salud basado en IoT.

3.1.2. Sistemas de IoT basados en agentes en salud

Para el análisis de ecosistemas de IoT dirigidos por agentes se seleccionaron 5 estudios como se evidencia en la Tabla 3. En dichos estudios se plantearon sistemas multiagentes para el cuidado de la salud usando IoT. En general, se presentan como resultados sistemas que proporcionan servicios afines con la salud utilizando una red y algún tipo de conexión entre agentes inteligentes. En la Tabla 3 se encontró que las condiciones de monitoreo son generalmente a través de agentes dedicados al diagnóstico de un paciente. En términos generales, los agentes más usados son los agentes usuarios, agente diagnóstico, agentes de medición y los agentes de decisión que hace de coordinador teniendo como objetivo principal administrar a los demás agentes para dar un correcto diagnóstico del estado del paciente. Asimismo, la tecnología más usada para implementar los agentes fue el marco de trabajo JADE siendo la herramienta más usada para crear agentes inteligentes para el cuidado de la salud.

3.1.3. Taxonomía de agentes

Para la formulación de una taxonomía de agentes, que no pretender ser un estándar en el desarrollo de agentes para la salud, se consideró el análisis de la Tabla 2 y Tabla 3. De esta manera, se planteó una taxonomía de agentes y sus funciones como se muestra en la Figura 6.

Tabla 3. Principales estudios en los que se propone agentificar el IoT para el desarrollo de sistemas de monitoreo de la salud.

N°	Año	Título	Monitoreo	Tecnología	Fuente	Ref.
1	2008	<i>A multi-agent prototype system for medical diagnosis</i>	Diagnóstico de tratamientos a seguir a través de agentes de decisión y diagnóstico	JADE y Java Expert System Shell (JESS)	IEEE	[52]
2	2009	<i>A Multi-Agent System (MAS) Based Scheme for Health Care and Medical Diagnosis System</i>	Un agente llamado Agente de Usuario (VA) se encarga de tomar las entradas del usuario, es decir, los síntomas de los pacientes con la ayuda de una interfaz de usuario	JADE	IEEE	[53]
3	2018	<i>Towards the Internet of agents: an analysis of the Internet of things from the intelligence and autonomy perspective</i>	Agente para medición inteligente del nivel de glucosa e informa la condición al paciente	Teórico	IEEE	[54]
4	2016	<i>Application of intelligent agents in healthcare: Review</i>	Planificación de procesos médicos, supervisión de la salud de los pacientes, el hospital y la gestión de los recursos clínicos.	JADE	Springer	[55]
5	2015	<i>Asistencia domiciliaria adaptada al contexto: una aproximación basada en multiagentes</i>	Agentes para Monitorizar la presión arterial y de pulso de un paciente	JADE	ResearchGate	[56]



Figura 6 Taxonomía de agentes para el cuidado de la salud IoT.

Una breve descripción de los agentes que conforman la taxonomía de la Figura 6 se describe a continuación:

- **Agente Usuario.** Agente encargado de dar funcionamiento al sistema al usuario. Es un intermediario entre el usuario y el sistema de agentes.
- **Agente Medición.** Agente especializado en obtener los datos de los sensores fisiológicos que se encuentra posicionado en el cuerpo del paciente.
- **Agente Coordinador.** Agentes que son la vía de comunicación entre los datos obtenidos por los sensores o también llamados agentes de medición y el de diagnóstico.
- **Agente Diagnóstico.** Agente orientado en realizar un diagnóstico, en base a los antecedentes de salud o parámetros de medición, del paciente, y además, ayuda a comprender el estado del paciente.

3.2. Propuesta

3.2.1. Descripción del sistema

El escenario de estudio estuvo alineado al cuidado de la salud basado en tecnologías de IoT, específicamente orientado al monitoreo de la temperatura y las condiciones cardio respiratorias de pacientes en su domicilio. Un esquema general del sistema propuesto se ilustra en la Figura 7. En este sentido, la atención médica se personaliza con herramientas o tecnologías emergentes como es el IoT para el cuidado

predictivo y preventivo de la salud las personas, diagnóstico que es llevado a cabo por entidades inteligentes como son los agentes software.

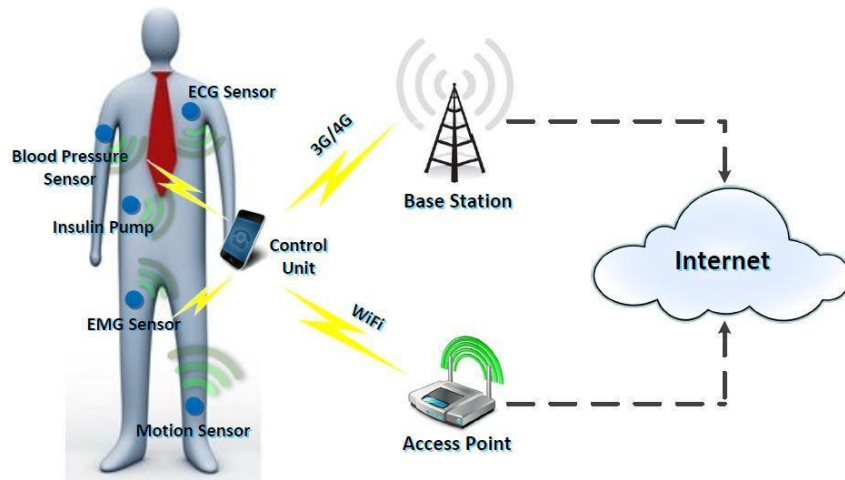


Figura 7 Escenario asistencia domiciliar basado en IoT.

3.2.2. Prototipo del objeto de IoT

Por el hecho de que en la propuesta se busca diseñar un objeto de IoT (hardware) agentificado (software), implica en primer lugar que en el proyecto se prototipara el diseño del circuito el microcontrolador que conecte los sensores a emplear (temperatura corporal, y ritmo cardiaco), y segundo lugar, también se ha requerido prototipar el sistema de agentes a partir del cual se diagnosticará el estado del paciente. El esquema general del objeto de IoT se ilustra en la Figura 8.

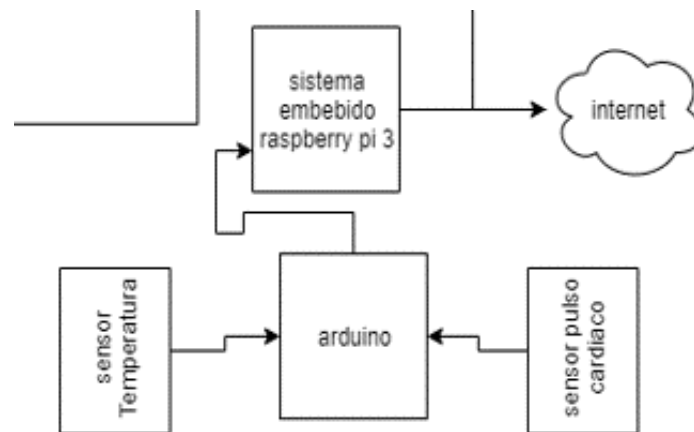


Figura 8 Diagrama de bloques del prototipo de objeto de IoT construido.

3.2.2.1. Hardware del objeto de IoT

3.2.2.1. Diseño del circuito

En base al esquema del prototipo de la Figura 8, se ha diseñado un sistema embebido que integró los sensores de temperatura corporal (DS18B20) y ritmo cardiaco (AD8232), en un microcontrolador de tipo Arduino UNO. El Arduino, establece conexión con una Raspberry Pi 3, siendo éste el que habilita la conexión a Internet y la ejecución de los agentes que dirigen el comportamiento del sistema.

Para diseñar el circuito, se elaboró un esquema usando un programa *open source* especializado en la realización de esquemas eléctricos en proyectos con Arduino, denominado *Fritzing*. En la Figura 9 y Figura 10, se muestra el esquema del circuito.

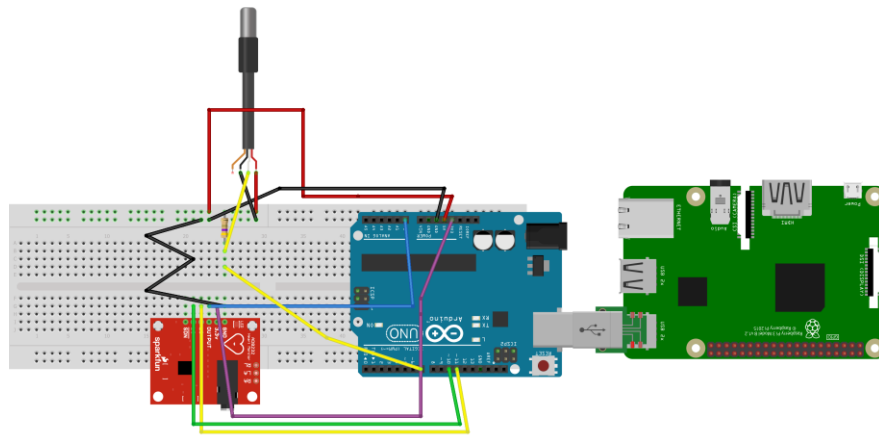
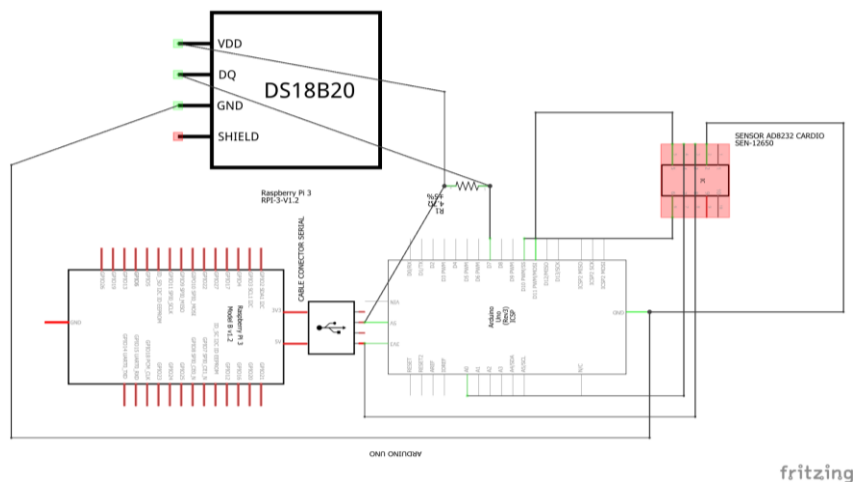


Figura 9 Circuito de Arduino con los sensores y comunicación con la Raspberry PI.



Una vez implementado el diseño del circuito se creó el dispositivo físico de IoT. Una vista del circuito creado con las tecnologías hardware usadas se muestra en la Figura 11.

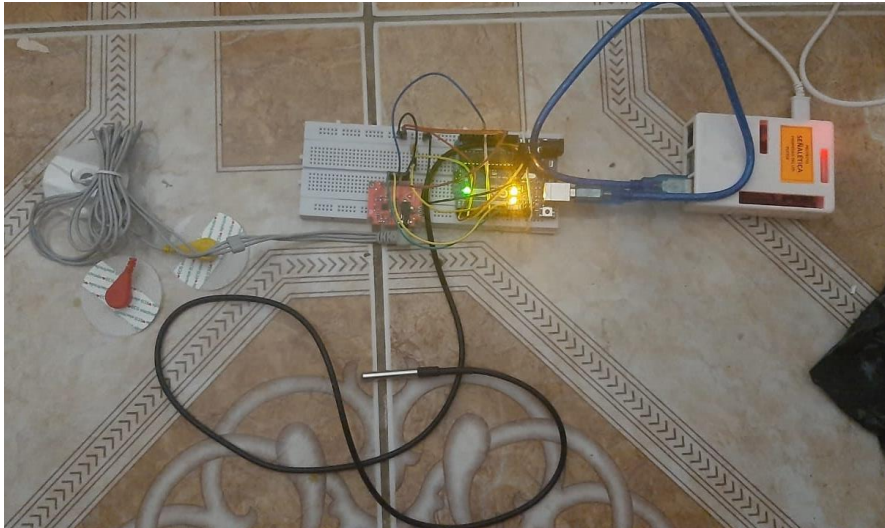


Figura 11 Construcción del prototipo final

Por otro lado, una vista de los sensores colocados en el cuerpo del paciente se muestra en la Figura 12. En la figura se observa cómo el sensor de ritmo cardiaco AD8232 se incorpora en el tórax de uno de los individuos que participó en el experimento.

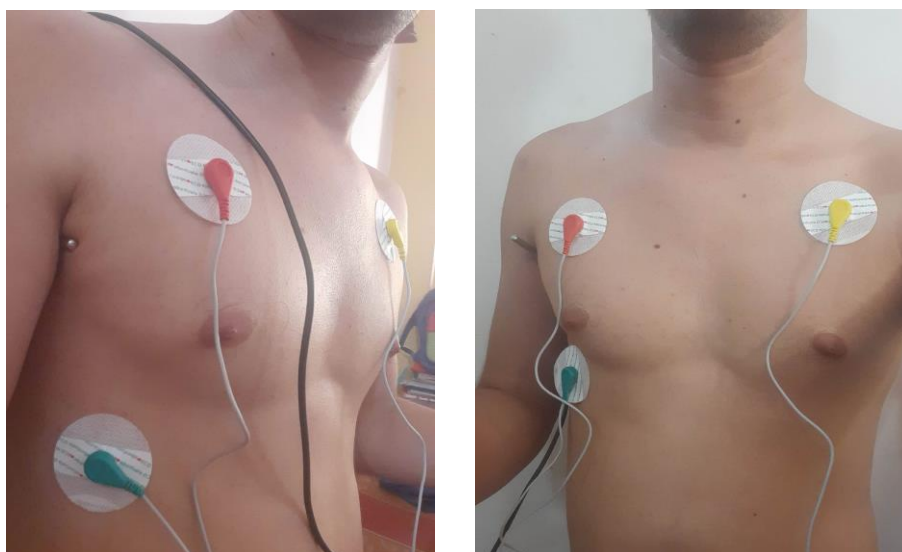


Figura 12 Colocación de los sensores para la recolección de datos

3.2.2.2. Lectura de datos de sensores

A través del IDE que implementa el lenguaje de programación de Arduino se realizó la lectura de los sensores. El código del Algoritmo 1 muestra la lectura de datos de la temperatura en el pin 7 del Arduino almacenándolo en una variable temperatura. Todo este montaje se hace en torno al protocolo 1-Wire que se usa para la comunicación de valores digitales y *DallasTemperatura* que es la Biblioteca para uso de sensores de medición de la temperatura. El mismo código permitió leer la información de cardio.

Algoritmo 1 Lectura de los valores de los sensores temperatura y cardio.

```
1. #include <OneWire.h>
2. #include <DallasTemperature.h>
3. const int DS18B20_PIN = 7; // Pin 7 del sensor
4. OneWire oneWire(DS18B20_PIN); // Creacion del objeto OneWire
5. DallasTemperature ds18b20(&oneWire); // Creamos objeto sensor
6. float temperatura; // Variable para almacenar la temperatura
7. void setup(void)
8. {
9.   Serial.begin(9600); // Inicializa el puerto serie
10.  ds18b20.begin(); // Inicializa el DS18B20 o cardio
11. }
```

Tras realizar la ejecución del Algoritmo 1, los datos de los sensores fueron medidos mediante el uso del microcontrolador Arduino UNO. En la Figura 13, muestra la lectura de dichos datos desde el entorno de Raspberry PI, Raspbian.

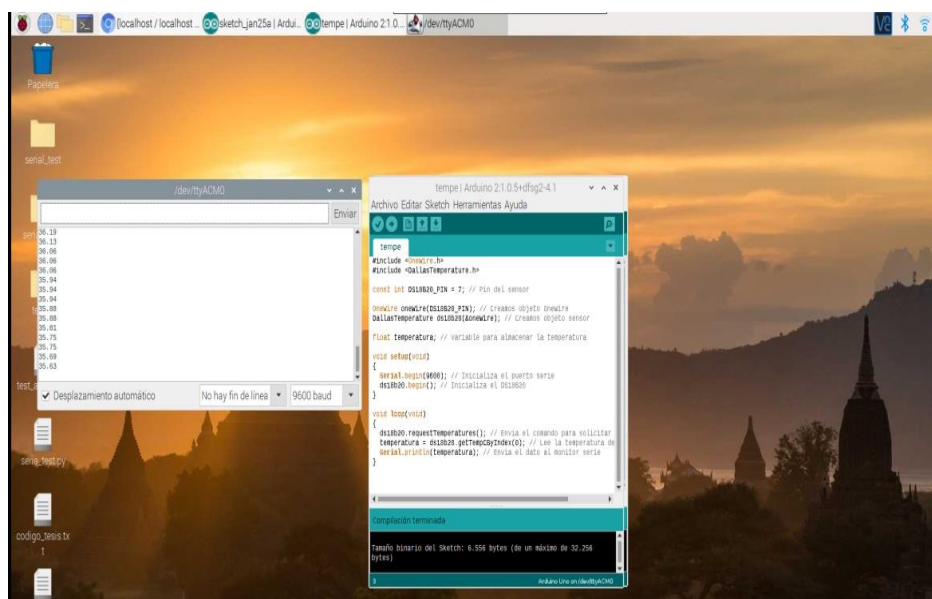


Figura 13 Lectura de datos desde el entorno de Raspberry PI 3.

3.2.2.3. Ecosistema de agentes para objeto de IoT

Arquitectura de agente para IoT

Para determinar la arquitectura idónea para agentificar el objeto de IoT creado, se tomaron en cuenta varias características de los agentes reactivos, deliberativos y las arquitecturas híbridas. Una comparación de éstas tres arquitecturas en términos de su complejidad, razonamiento, tiempo de respuesta y los aspectos de implementación, se resume en la Tabla 4.

Tabla 4 Comparación de las arquitecturas para los agentes

Características	Deliberativa	Reactiva	Híbrida
Modelo	Modelo simbólico complejo	Modelo simbólico practico	Modelo simbólico practico
Complejidad	Alta	Alta	Baja
Planificación	Realiza tareas para alcanzar sus objetivos	No realiza tareas para alcanzar sus objetivos	Realiza tareas para alcanzar sus objetivos
Razonamiento	Decisiones se toman utilizando mecanismos de razonamiento lógico	No razona	Razonamiento lógico
Tiempo de respuesta	Lento	Rápido	Rápido
Estructura	Complejo	Simple	Simple
Visión	A largo plazo	A corto plazo	A corto plazo

En base a los resultados de la comparación realizada, se decidió emplear la arquitectura reactiva puesto que los agentes reactivos reúnen todos los conocimientos de las acciones a realizar en un sistema dirigido por agentes. Además, dicha arquitectura no necesita construir un modelo de representación mental de su entorno ya que operan en base a un mecanismo evento-reacción. Por otra parte, los agentes reactivos tienen la capacidad de reacción basada en los acontecimientos de su ambiente, aspecto que se gestiona en IoT. Y finalmente, es importante mencionar que los agentes reactivos son una alternativa eficiente debido a que no requieren recursos de cómputo sofisticados para ejecutarse. Todas estas características ubican a los agentes reactivos como entidades idóneas para el desarrollo de un sistema como el propuesto.

Modelo de agente para IoT

Para la parte de diseño del sistema de agentes, una vez recogida toda información sobre las acciones a ejercer cada agente y la arquitectura a usarse, se propuso un sistema en el que intervienen cinco tipos de agentes como se muestra en la Figura 14. Estos agentes corresponden a: AgenteUsuario, AgenteDiagnosticador, Agente Coordinador, AgenteTemperatura y AgenteCardio o considerados como Agentes de Medición. Dicho modelo de sistema multiagentes se ajusta a un sistema embebido implementado a través de un componente de placa única como es Raspberry PI.

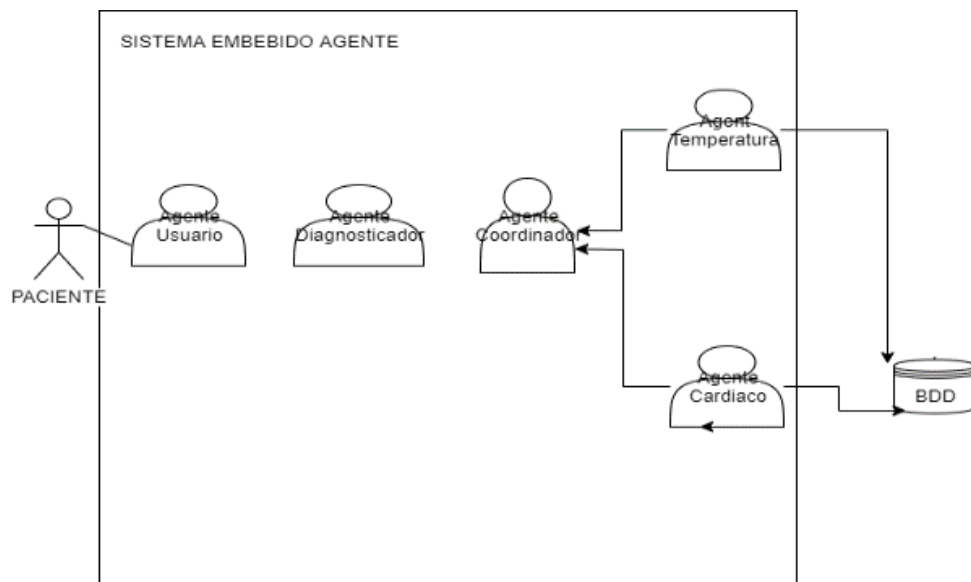


Figura 14 Modelo de sistema multi- agentes de IoT

Los agentes de la Figura 14 corresponden a agentes descritos en la taxonomía previamente definida en la Figura 6. En este sentido, El **Agente Usuario** tiene como tarea iniciar el funcionamiento del sistema. El **Agente Coordinador** realiza la petición de los datos procesados al **Agente Medición** que a su vez lee y accede a los datos de los sensores tanto de temperatura y electrocardiograma, ambos almacenados en su respectiva base de datos. Posteriormente, se envían los datos al **Agente Coordinador** para que éste almacene una muestra con los datos a trabajar y, por último, éste envía al **Agente Diagnóstico** para que realice las tareas de formular un diagnóstico y recomendación al paciente.

En la Figura 15, se muestra un diagrama secuencia que detalla cómo los agentes se comunican entre sí. Los agentes pueden llevar a cabo sus tareas y cumplir sus objetivos siguiendo este esquema de comunicación. Es importante señalar que todos los agentes, implementados en JADE, se comunican siguiendo el estándar FIPA, por lo que éstos pueden interoperar a futuro incluso con otras plataformas de agente si se lo requiere. Dicho diagrama sigue las acciones descritas previamente donde se indicó los tipos de agente, sus tareas y cómo se interrelacionaban para cumplir sus objetivos, y consecuentemente, el objetivo del sistema.

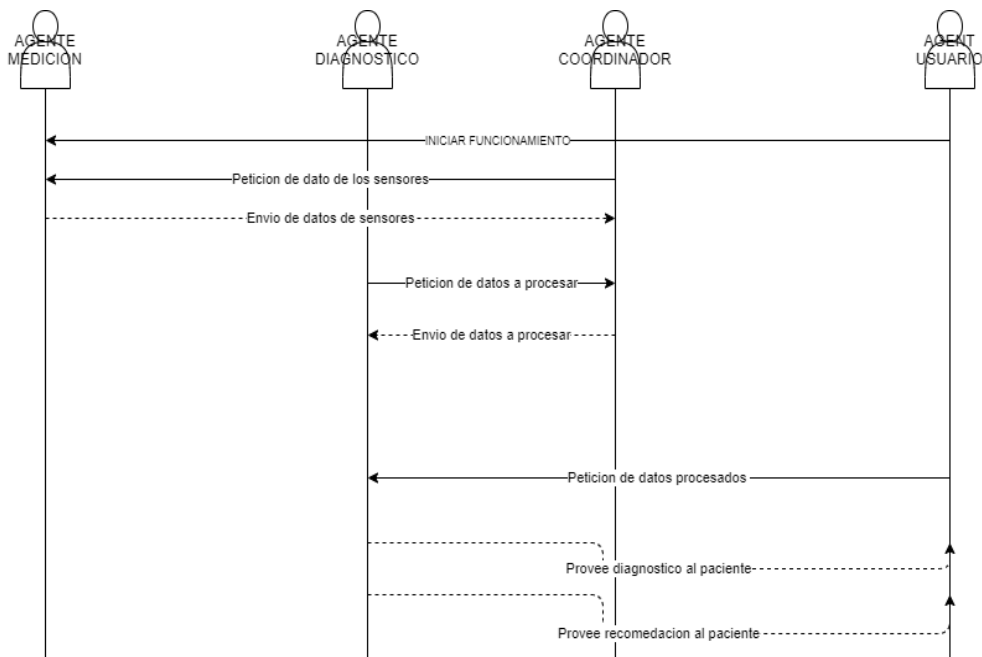


Figura 15 Diagrama de Secuencia de la comunicación de los agentes

Desarrollo de agentes en JADE

A través de NetBeans, que es un entorno de desarrollo para sistemas Java, se importó las dependencias para usar el marco de trabajo JADE.

Siguiendo la sintaxis para el desarrollo de agentes JADE, se programaron 5 clases, a partir de las cuales fue posible crear las instancias de los agentes antes descritos, y que formaron parte del sistema multiagentes. La forma en cómo se estructuraron las plantillas para la creación de los agentes involucrados en el sistema se muestra en la Figura 16.

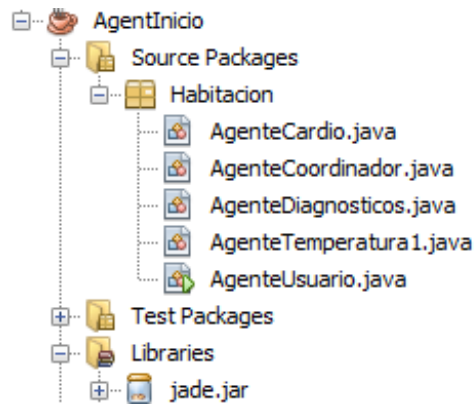


Figura 16 Organización de las plantillas para crear los agentes JADE.

Plantillas para la creación de agentes en JADE

Con la finalidad de ilustrar el comportamiento de uno de los agentes, en el Algoritmo 2 se presenta el código de la plantilla para crear el Agente Usuario. En términos generales, se incluyen las librerías (líneas 3-8), y se crea una clase principal donde se crea un agente en la plataforma JADE (líneas 13-37), sobre dicha plataforma, se crea las cinco instancias de agentes antes descritas (líneas 43-47).

Algoritmo 2. Platilla para la creación de agente usuario.

```

1. package Habitacion; // ecosistema dentro del hogar
2.
3. import jade.core.Runtime;
4. import jade.core.Profile;
5. import jade.core.Agent;
6. import jade.core.ProfileImpl;
7. import jade.wrapper.*;
8. import java.net.InetAddress;
9.
10.
11. public class AgenteUsuario {
12. public static void main(String args[]) throws StaleProxyException {
13.     Properties pp = new Properties();
14.     pp.setProperty(Profile.MAIN, Boolean.FALSE.toString());
15.     pp.setProperty(Profile.MAIN_HOST, "127.0.0.1");// conexión con el servidor
16.     pp.setProperty(Profile.MAIN_PORT, Integer.toString(8080));// Puerto por default
17.     Profile p = new ProfileImpl(pp);
18.     AgentContainer ac =jade.core.Runtime.instance().createAgentContainer(p);
19.     Runtime rt = Runtime.instance();
20.     rt.setCloseVM(true);
21.
22.     //Buscar dirección en el ipconfig
23.     String IP_local = "";
24.     try {
25.         InetAddress direccion = InetAddress.getLocalHost();
26.         IP_local = direccion.getHostAddress();//ip como String
27.     } catch (Exception e) {
28.         e.printStackTrace();
29.     }
30.
31.     //Asignar IP hallada
32.     Profile profile = new ProfileImpl(IP_local, 8080, "main");
33.     AgentContainer mainContainer =

```

```

34.         jade.core.Runtime.instance().createMainContainer(profile);
35.
36.     // creación de contenedores a los agentes
37.     AgentController rma = null;
38.     AgentController rme = null;
39.     AgentController rmi = null;
40.     AgentController rmo = null;
41.     AgentController rmu = null;
42.     try {
43.         rma = mainContainer.createNewAgent("temp1", "Habitacion.AgenteTemperatura1",
            new Object[0]);
44.         rme = mainContainer.createNewAgent("temp2", "Habitacion.AgenteTemperatura1", new
            Object[0]);
45.         rmi = mainContainer.createNewAgent("coord", "Habitacion.AgenteCoordinador", new
            Object[0]);
46.         rmo = mainContainer.createNewAgent("card", "Habitacion.AgenteCardio", new
            Object[0]);
47.         rmu = mainContainer.createNewAgent("diag", "Habitacion.AgenteDiagnosticos", new
            Object[0]);
48.
49.         // inicio de contenedores
50.         rma.start();
51.         rme.start();
52.         rmi.start();
53.         rmo.start();
54.         rmu.start();
55.     } catch (StaleProxyException e) {
56.         e.printStackTrace();
57.     }
58. }
59. }

```

Las plantillas complementarias están accesibles en el repositorio del proyecto disponible en: <https://github.com/Proyecto-bryan-espinoza/proyectoIoA>.

Instancias de agentes en JADE

A partir de la plantilla previamente creada, a continuación, se procedió a crear las instancias de los agentes en JADE. Para ello, se lanzó la plataforma JADE la cual es capaz de gestionar y coordinar los agentes que formen parte de un sistema multiagentes.

En la Figura 17, se visualiza la plataforma en el entorno de *JADE* con las cuatro instancias de agentes creados, y que dirigen el sistema de diagnóstico médico, esto es, AgenteCardio, AgenteCoordinador, AgenteDiagnostico y AgenteTemperatura. Todos los agentes se ejecutaron en la misma plataforma de agentes y la comunicación la realizaron mediante el uso de mensajes que siguen la sintaxis del lenguaje de comunicación de agentes (ACL) propuesto por el estándar FIPA.

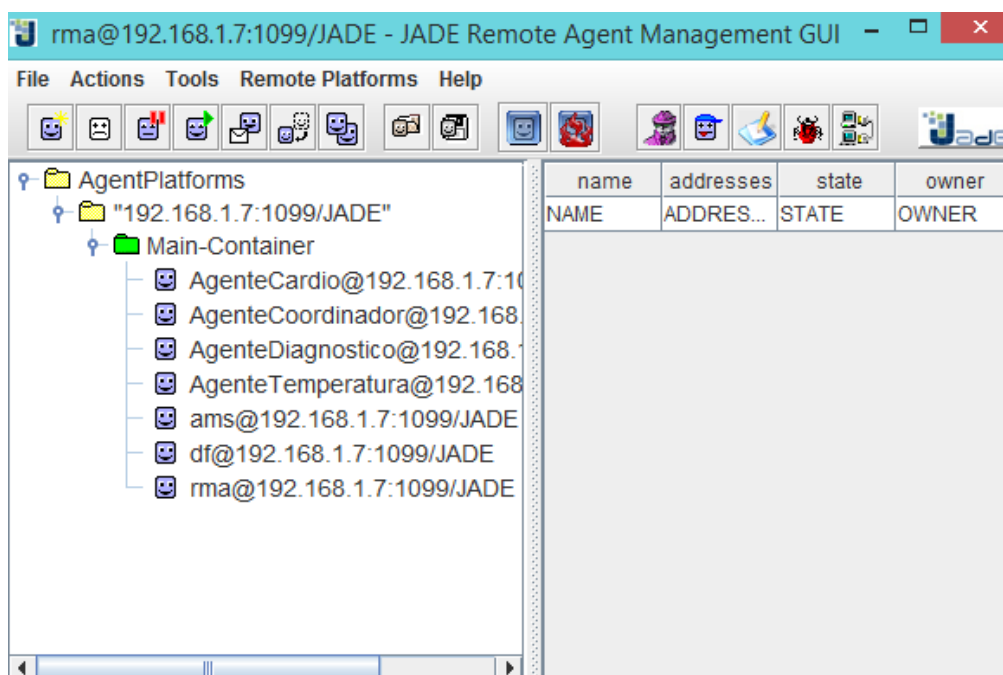


Figura 17 Plataforma JADE y agentes del sistema.

3.2.2.4. Integración, agentificación del IoT

Para la construcción de prototipo se creó un modelo como se visualiza en la Figura 18. En dicho modelo se integra el circuito del objeto de IoT con el sistema de agentes desarrollado. También se integra el modelo de datos usado por el sistema para guardar los datos obtenidos por el objeto de IoT planteado.

En términos generales, se planteó el proceso de agentificación partiendo de integrar los agentes en un ecosistema de IoT para el cuidado de la salud. Se procedió a embeber en una Raspberry en conjunto con Arduino, agentes sobre los objetos de medición o sensores como son el de temperatura y cardio, creándose así, un sistema multiagentes desarrollado en JADE —agentes de Medición (Temperatura y Cardio), Coordinador, Usuario y Diagnóstico— que a través de la misma plataforma se ejecutaron dentro de un contenedor. A partir de dichos agentes se pudo establecer comunicación, logrando gestionar los recursos de IoT de manera inteligente, proactiva y colaborativa que en pocas palabras son cosas conectados a una red con autonomía propia es decir sin la necesidad de una persona.

Además, fue posible almacenar los datos capturados por el objeto de IoT, procesados por los agentes, en una base de datos relacional MySQL.

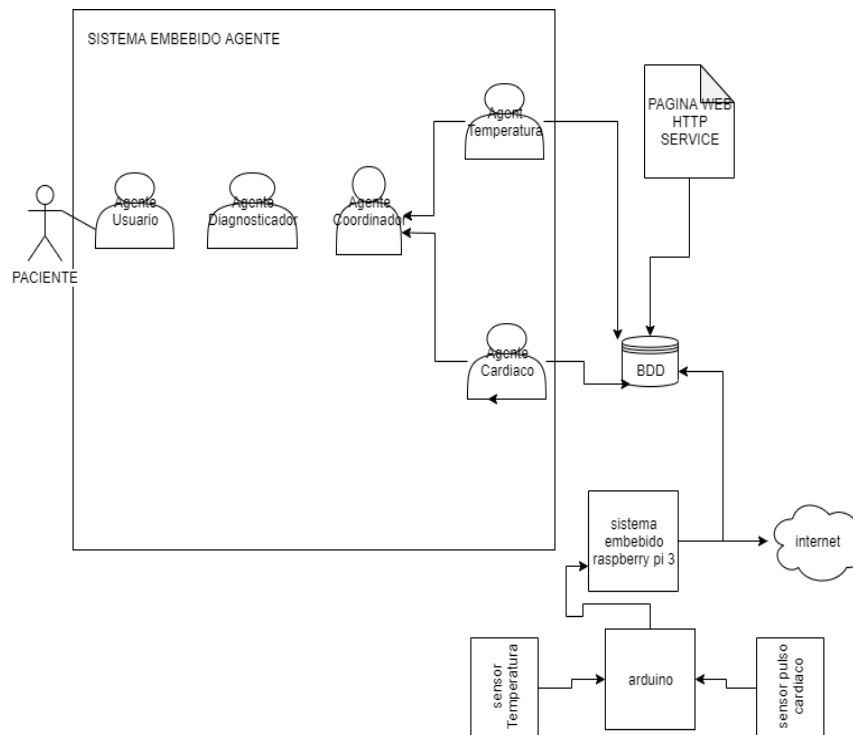


Figura 18 Modelo general del Sistema para agentificación de objetos para el IoT

3.2.2.5. Modelo de datos

Para el proceso de análisis de los datos con los que los agentes que forman parte del sistema operaron, se construyó un servidor dentro del Raspberry PI para subir los datos de los sensores y almacenarlos en una tabla llamada *datos_temp_cardio*. En términos generales, la tabla incluyó los campos: id BIGINT (20) PRIMARY KEY AUTO_INCREMENT, tiempo DATETIME, y el campo temperatura DECIMAL, cardio DECIMAL.

Para subir los datos se utilizó Python como lenguaje de programación en el cual se importó varias librerías para conectar, en primer lugar, al Arduino por puerto serie, el conector específico para la conexión de la base de datos con Python y JSON para recibir los datos en JSON. Después se insertaron los datos medidos en la BD. El código para realizarlo se muestra en el Algoritmo 3.

Algoritmo 3. Subida de los datos de los sensores a la base de datos *PhpmyAdmin*.

```
1. #!/usr/bin/env python
2.
3. import serial
4. import json
5. import MySQLdb
6. arduino = serial.Serial('/dev/ttyACM0',9600)//comunicación serial arduino con
python
7. print arduino.readline()
8. print arduino.readline()
9. while True:
10.     character= arduino.readline()
11.     MyJson = character
12.     db = MySQLdb.connect("localhost","root","pucese11","datos") //conexión a la
base de datos
13.     if character != '\n':
14.         try:
15.             data=json.loads(character)
16.             print data
17.             print data['cardio']
18.             curs = db.cursor()
19.             curs.execute("INSERT INTO
datos(fecha,temperatura,cardio)values(now(),"+str(data['temperatura'])+", "+str(da
ta['cardio'])+")")
20.             db.commit()//confirmar cambios
21.         except ValueError           print ""
22. arduino.close()
```

3.2.2.6. Prueba del objeto propuesto

La puesta en marcha del sistema desarrollado permitió la captura de datos en tiempo real de la temperatura y de la condición cardiorrespiratoria del paciente que usó el objeto de IoT. En la Figura 19, se muestra los datos capturados por el objeto y se muestra el diagnóstico correspondiente. Estos datos se visualizan en el Anexo C.

```
INFORMACIÓN: MTP addresses:|
http://equipo-bryan:7778/acc
ene 26, 2020 7:34:15 PM jade.core.AgentContainerImpl joinPlatform
INFORMACIÓN: -----
Agent container Main-Container@192.168.56.1 is ready.
-----
-----
Temperatura: 36C° Cardio: 110
-->El diagnostico del paciente es Temperatura normal y frecuencia cardiaca normal
-----
Temperatura: 36C° Cardio: 115
-->El diagnostico del paciente es Temperatura normal y frecuencia cardiaca normal
-----
```

Figura 19 Funcionamiento de los agentes resultado final en NetBeans

El diagnóstico que el agente realiza en base a las dos condiciones fisiológicas del paciente se realiza a partir de las reglas descritas en el Algoritmo 4.

Algoritmo 4. Reglas seguidas por el agente para diagnosticar el estado del paciente.

```
1. //Temperatura baja y frecuencia cardiaca anormal
2. if ((comp[0] >= 35 && comp[0] < 33) && (comp[1] >= 120 && comp[1] < 86))
3. {
4.     estado = "Temperatura baja y frecuencia cardiaca anormal";
5. }
6. //Temperatura normal y frecuencia cardiaca normal
7. else if ((comp[0] >= 37 && comp[0] < 36) && (comp[1] >= 84 && comp[1] < 70))
8. {
9.     estado = "Temperatura normal y frecuencia cardiaca normal";
10. }
11. //Temperatura alto y frecuencia cardiaca buena
12. else if ((comp[0] >= 40 && comp[0] < 39) && (comp[1] >= 68 && comp[1] < 62))
13. {
14.     estado = "Temperatura levemente alta y frecuencia cardiaca levemente buena";
15. }
16. //Temperatura muy alto y frecuencia cardiaca excelente
17. else if ((comp[0] >=42 && comp[0] < 41) && (comp[1] >= 60 && comp[1] < 50))
18. {
19.     estado = "Temperatura muy alta y frecuencia cardiaca muy buena";
20. }
```

3.3. Evaluación del prototipo

Para la evaluación el dispositivo se conectó inalámbricamente (Wifi 802.11) a un *router* en el que se usó protocolos de comunicación como es el HTTP/HTTPS que realiza un paradigma cliente-servidor y una topología de red tipo estrella. Se creó un agente coordinador y una vez ya ejecutado el prototipo para el cuidado de la salud, se pudo observar mediante la métrica de usabilidad, funcionamiento el correcto uso, y funcionamiento del sistema del modelo para agentificar objetos de IoT. Dicha evaluación se realizó en función de la coherencia del comportamiento de los agentes, de la calidad de los datos de los sensores y del diagnóstico formulado.

3.3.1. Evaluación de la usabilidad

Para verificar que la facilidad de manejo del prototipo de una forma adecuada fue necesario realizar las pruebas de usabilidad y con ayuda de la norma ISO 9241, se definió que la usabilidad contempla las métricas de efectividad, eficiencia y satisfacción con la que un producto permite lograr objetivos específicos a usuarios centrado en la calidad de uso, es decir, se describe cómo el usuario realiza tareas específicas en escenarios específicos con efectividad.

Las tareas para la efectividad, eficiencia y satisfacción son las siguientes:

- T1. Arranque del sistema.
- T2. Conexión a la red domestica (por cable o wifi).
- T3. Colocación correcta de los sensores de cardio como de temperatura.
- T4. Verificar si el servidor esta activo.
- T5. Ejecutar el software empleado al sistema multiagentes.

Efectividad

Se entiende que cualquier software debe tener unos objetivos claros y éstos deben ser alcanzables. Se usó la métrica de porcentaje de tareas completadas con éxito al primer intento. Para el cálculo de la efectividad se aplicó la ecuación detallada en (1).

$$Efectividad = \frac{\text{número de tareas completadas con éxito}}{\text{número de tareas realizadas}} * 100\% \quad (1)$$

Una vez calculada la efectividad del sistema propuesto, se evidenció en la Figura 20 que los primeros cinco intentos fueron muy fructíferos puesto que el porcentaje de éxito en la mayoría fue del 80 % hasta alcanzar 100%, generando un porcentaje de efectividad aceptable en la que muestra que los usuarios no tendrán gran dificultad a la hora de completar este tipo las tareas sujetas a estudio, todas ellas básicas para utilizar el prototipo o sistema.

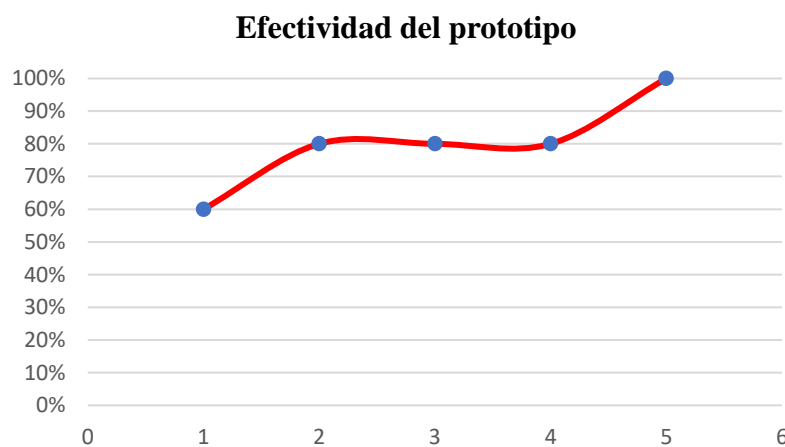


Figura 20 Efectividad del prototipo

Eficiencia

Mide los recursos empleados para lograr determinados objetivos. Los indicadores de eficiencia incluyen el tiempo de finalización de las tareas y el tiempo de aprendizaje. Para este aspecto se utilizó la métrica de tiempo invertido en el primer intento como se muestra en (2).

$$Eficiencia = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{NR} \quad (2)$$

Donde:

- N= número total de tareas propuestas.
- R= número total de usuarios.
- n_{ij} = el resultado de la tarea i por el usuario j; si el usuario completa la tarea con éxito $n_{ij}=1$, caso contrario $n_{ij}=0$.
- t_{ij} = el tiempo empleado por usuario j para completar la tarea i. Si la tarea no se completa con éxito, entonces se mide el tiempo hasta el momento en que el usuario abandona la tarea.

Una vez que los usuarios aprendieron el funcionamiento básico del prototipo para calcular la eficiencia del sistema que es la que hace referencia al tiempo de media que es necesario para calcular cada tarea se obtuvo que la eficacia fue de 0.023 tareas por segundo como se muestra en (3).

$$Eficiencia = \frac{\frac{3}{27} + \frac{4}{35} + \frac{4}{36} + \frac{4}{38} + \frac{5}{40}}{5 \times 5} = 0.023 \text{ tareas / s} \quad (3)$$

Satisfacción

Es el grado con que el usuario se siente satisfecho mediante actitudes positivas al usar el prototipo y alcanzar los objetivos. Se usó como métrica un cuestionario para su respectiva medición.

Con relación a su evaluación, el aspecto de satisfacción de desempeño llevado a cabo después de realizar las tareas antes mencionadas (T1-T5) permitió medir el nivel de comodidad del prototipo. El cuestionario empleado contempló un

cuestionario estandarizado para medir la satisfacción. Dicho cuestionario corresponde al cuestionario *Post-Tareas SEQ* [57].

El cuestionario SEQ fue elegido ya que es fácil de responder. Una vez que finalizó cada tarea el usuario puntuó con una X del 1 al 7 el nivel de complejidad (Muy Difícil- Muy fácil) de la tarea evaluada siguiendo el modelo que se adjunta en el ANEXO A.

La puntuación de cada una de las tareas por parte de los usuarios fue considerada alta por lo que se determinó que el sistema ofrece al usuario una satisfacción a la hora de su uso. Con la finalidad de detallar de una manera más profunda sobre los aspectos evaluados, a continuación, se describen los resultados obtenidos en cada una de ellas:

- **En general ¿Cuál fue el grado de complejidad del 1 al 7 al iniciar o arrancar el prototipo?**

Resp. Muy fácil. Argumentando a la respuesta del usuario, la elección de muy fácil se dio porque la complejidad de encender el sistema no es nada del otro mundo dado que solo se tuvo que conectar a una fuente de poder o toma corriente. El proceso consistió básicamente en encender una computadora por lo que no se requirió más que presionar ningún botón de arranque y en este caso el usuario ni tuvo gran inconveniente.

- **En general ¿Cuál fue el grado complejidad del 1 al 7 para conectar el prototipo a Internet ya sea por cable o wifi 2?**

Resp. Fácil. En todo sistema operativo se cuenta en la actualidad con maneras de conexión a Internet. Por parte del usuario se hizo que conectara a Internet de los dos modos posibles, esto es, tanto por cable como inalámbrico (Wifi). En ambos casos la conexión fue exitosa y por ende fue fácil. La única demora fue hallar la ubicación para conectarse vía Wifi por el simple hecho de que el usuario no estaba familiarizado al sistema operativo de Raspbian.

- **En general ¿Cuál fue el grado de complejidad del 1 al 7 la colocación correcta de los sensores tanto de cardio como de temperatura?**

Resp. Algo fácil. Se pudo observar que la complejidad no fue un problema debido a que el sensor para medir el ritmo cardiaco en el cuerpo debe ser exacto, en los primeros intentos, la ubicación de los 3 electrodos fue colocados por el usuario de forma incorrecta. Sin embargo, una vez comprendido eso se colocó de manera correcta y por lado del sensor de temperatura no hubo mayor problema de ubicación.

- **En general ¿Cuál fue el grado de complejidad del 1 al 7 para verificar si el servidor esta activo?**

Resp. Algo fácil. Esta tarea es de alta relevancia porque con ayuda del servidor, los datos de los sensores se pudieron almacenarse en la base de datos. Cabe recalcar que el prototipo está diseñado para una vez arrancar el sistema y conectarse al Internet, el servidor permanecer activo. Para el usuario fue importante saber que primero se debe contar con una conexión a Internet, luego abrir el navegador y en la zona de enlace colocar localhost, y fue redireccionado a una página que muestre información del servidor que se está ejecutando. Si no redirecciona es porque esta inactivo. La tarea fue algo fácil puesto que al principio al no tener la práctica de una o dos veces no se llegaba al objetivo de verificar; pero luego con la experiencia que obtuvo el usuario se pudo realizar con éxito.

- **En general ¿Cuál fue el grado de complejidad al ejecutar el software empleado del sistema multiagentes?**

Resp. Muy fácil. La tarea en si no tuvo complejidad alguna debido a que el software creado se alojó un archivo ejecutable en el escritorio del sistema. Por lo tanto, su ejecución basta con ejecutar dicho programa y el usuario pudo ubicar fácilmente el programa a ejecutar.

3.3.2. Evaluación de la funcionalidad

Se evaluó la funcionalidad para evidenciar si el prototipo cumple con el objetivo planteado, si realiza su trabajo sin presentar anomalías y si presenta resultados de calidad de manera oportuna según los requerimientos del usuario. A continuación, se evidencia la Tabla 5 los rangos de aceptación usados tanto para el funcionamiento como la portabilidad del prototipo.

Tabla 5 Valores por cumplimiento

Valor	% Cumplimiento	Significado	Rango
1.0	90-100	Excelente	A
0.8	79-89	Satisfactorio	B
0.6	50-69	Aceptable	C
0.4	30-49	Deficiente	D
0.0	0-29	Inaceptable	E

En la Tabla 6 se visualiza la evaluación del cumplimiento de los aspectos que definen la funcionalidad del sistema, y de acuerdo con los valores de cumplimiento el resultado fue excelente, por ende, el sistema tiene un alto nivel de funcionamiento.

Tabla 6 Evaluación del Funcionamiento

N°	Aspectos a Evaluación del Funcionamiento	SI/NO
1	¿Captura el sensor de temperatura datos en tiempo-real?	SI
2	¿Capturan el sensor de cardio datos en tiempo real?	SI
3	¿Los dispositivos usados tienen características de un sistema IoT para el cuidado de la salud?	SI
4	¿Los sensores fisiológicos se encargan de medir y detectar variaciones significativas de las constantes vitales de una persona?	SI
5	¿Se evidencia las peticiones de datos por parte del agente coordinador a la base de datos en la que se almacena lo datos recogidos por los agentes de medición?	SI
6	¿Responde el agente coordinador a las solicitudes realizadas por parte del agente diagnosticador?	SI
7	¿Las tareas realizadas por el agente Diagnosticador se realizó con éxito?	SI
8	¿Se ejecutaron de manera correcta los agentes en el dispositivo prototipado?	SI

Las observaciones realizadas al sistema propuesto, en la que se evidencia el cumplimiento de ocho aspectos que involucran al IoT y el sistema multiagentes, detallan en base a sus respuestas a las preguntas 1, 2, 3 y 4 que el objetivo o tareas se pudieron realizar y constatar sin problema alguno.

Estas primeras cuatro interrogantes se realizaron con el fin de verificar que en la parte física del prototipo trabajaje en base a lo solicitado como fue el funcionamiento de los sensores en tiempo real en conjunto con la Raspberry PI,

dando como resultados datos de temperatura y cardio que se monitorearon con un *delay* de un segundo. Por otro lado, los componentes o sensores integrado son usados para medir o diagnosticar el IoT en el cuidado de la salud, Raspberry, sensores como el de temperatura (DS18B20) y cardio (AD8232), permitieron medir las condiciones fisiológicas del paciente. A partir de esta información, los agentes pudieron llevar a cabo sus acciones en respuesta a los datos recibidos.

Las interrogantes siguientes como son 5, 6, 7 ,8 se centraron los agentes software creados en la plataforma de JADE. Dichos agentes, cumplieron con las tareas que se les establecieron. Tareas como la petición de los datos medidos y recolectados por los agentes de medición, diagnóstico del paciente y la comunicación de los datos entre agentes fue exitoso. Esto se debe a que JADE es una herramienta para el desarrollo de agentes que ha pasado la fase experimental y es un software con un alto grado de madurez y popularidad entre la comunidad de desarrolladores de agentes. En resumen, el sistema de agentes se comportó acorde a los requerimientos y necesidades.

3.3.3. Portabilidad

La evaluación de la portabilidad busca determinar qué tan manipulable por parte de un usuario y además determinar la factibilidad para que el dispositivo pueda ser llevado llevar de un lado a otro sin interrumpir su funcionamiento normal.

A continuación, se visualiza la evaluación y de acuerdo con los valores de cumplimiento el resultado fue aceptable, esto significa que por ende puede ser manipulado por el usuario.

- **¿Funciona el dispositivo prototipado sin necesidad de estar conectado a una fuente de alimentación?**

Resp: No. La respuesta es porque el prototipo de objeto agentificado funciona por medio de un cargador que es de vital importancia. Entonces, debe estar conectado a una fuente de alimentación y consecuentemente, el prototipo no está diseñado para poder trabajar sin corriente alterna AC. Una

alternativa sería usar una batería portable que no sea compleja de transportar.

- **¿Puede ser el dispositivo prototipado llevado a escenarios distintos del hogar del usuario?**

Resp: Sí. Argumentando a la respuesta, el prototipo se puede llevar de uno a otro sector del hogar siempre y cuando exista una fuente de alimentación en los puntos del hogar donde se lo use y siempre que haya conectividad inalámbrica cercana para conectarse a Internet.

- **¿Es el dispositivo prototipado de fácil transportación?**

Resp: Sí. El prototipo puede transportarse de una dirección a otra sin problema alguna ya que cuenta con dispositivos electrónicos pequeños que no son pesados.

- **¿Se ejecuta el dispositivo prototipado en una plataforma distinta para el cual fue construido?**

Resp: No. Según el diseño con el cual se construyó el prototipo cuenta con una Raspberry PI de bajo costo que trabaja bajo un sistema operativo Linux como es Raspbian. por lo tanto, el dispositivo no está diseñado para operar en otra plataforma que no sea la antes mencionada.

- **¿Se puede reemplazar los objetos usados para medir (sensores) por otros nuevos?**

Resp: Sí. Los sensores si se pueden reemplazar por otros nuevos cuando así lo desee el usuario siempre y cuando sean de las mismas características que el anterior. Se puede usar otro tipo de sensores; pero el circuito debe cambiar y el sistema de lectura de datos también.

CAPÍTULO IV: DISCUSIÓN

Existe en la actualidad problemas en las aplicaciones que se han enfocado en la utilización del Internet de las Cosas en términos de la comunicación entre objetos y así lograr la monitorización y ejecución de procesos automáticos donde recogen gran cantidad de datos de pacientes acerca de los signos vitales (i.e., control de la respiración, glucosas, frecuencia cardiaca). Generalmente los datos son almacenados en hojas cálculo y es difícil que los cuidadores puedan combinarlos y analizarlos. Para contrarrestar este problema, se han ya propuesto mejorar la eficiencia y la automatización de los procesos de monitoreo y diagnóstico basado en agentes inteligentes creados en la plataforma de JADE, de tal manera que se pueda apoyar al paciente para el cuidado de la enfermedad que padece, en casos particular, pacientes con problemas de respiración, glucosa, frecuencia cardiaca [34], [35], [36], [38].

El desarrollo de este trabajo de investigación presenta un modelo innovador para el llevar a cabo proceso de monitorización, almacenamiento y análisis de datos de dos condiciones fisiológicas. Mediante la utilización de agentes inteligentes que se comunican entre sí y el uso de datos recogidos de sensores fisiológico como son el de temperatura y cardio, se pudo emitir un diagnóstico de pacientes. Es importante determinar el número de agentes adecuados para evitar que el sistema tenga una alta complejidad y disminuya así su nivel de eficiencia.

Se monitoreo en tiempo real y se almacenaron los datos sensados en una base de datos, permitiendo una información de mejor calidad e integración de manera que estén mejor organizados. Además de la eficiencia de recursos para guardar gran cantidad de datos y dar un análisis más exacto a través de entidades reactivas, fue posible que el objeto de IoT respondiera de forma autónoma ante situaciones específicas. Ello ha habilitado al prototipo de objeto de IoT que llegue a comportarse de manera activa, escalando de la típica posición de los objetos pasivos.

Además, los resultados de la evaluación de las variables usabilidad, funcionamiento y portabilidad fueron favorables para medir la calidad del sistema generando el cumplimiento de los objetivos propuestos. Por parte del funcionamiento se evaluaron ocho aspectos o característica del sistema que se

mencionan en la Tabla 6 y que determinaron el manejo correcto de la integración de la parte física y de los agentes de IoT sin presentar anomalías o error alguno. De acuerdo con los valores de cumplimiento, el resultado fue excelente por lo tanto el sistema cuenta con un alto nivel de funcionamiento. Es un sistema sencillo pero funcional.

Asimismo, se evaluaron cinco aspectos de portabilidad para determinar la capacidad del prototipo al cambiar de entorno. En cuanto con los valores de cumplimiento, el resultado obtenido tras evaluar el prototipo desarrollado en este estudio fue aceptable, esto quiere decir, que el prototipo puede ser manipulado sin problemas por parte del usuario.

Finalmente, se evaluó la usabilidad del sistema con la ayuda de la ISO 9241 en la que se definieron las métricas de efectividad, eficiencia y satisfacción. Para el cálculo de las métricas se determinaron cinco tareas centrada en la calidad de uso que fueron resueltas por el usuario y tuvieron como resultados un aceptable porcentaje en la mayoría con una efectividad del 80%, una eficiencia de 0.023 tareas/s y por último una satisfacción considerada alta en la que se utilizó un cuestionario *Post-Tareas SEQ*. Consecuentemente, se demostró un nivel óptimo de interacción y uso del prototipo por parte del usuario. Además, es importante señalar que el usuario no debe tener altos conocimientos de tecnología, luego de un entrenamiento previo puede ser factiblemente usado.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Esta investigación presenta un análisis de estudios previos enfocados en ecosistemas de Internet de las Cosas y en agentes para gestionar los recursos del IoT, donde se determinó una estructura organizada, es decir una taxonomía de agentes de acorde al ecosistema propuesto en el área de la salud. Se tomaron en cuenta agentes de medición, diagnóstico, coordinador y usuario. Sin embargo, podrían integrarse nuevos agentes acorde a la complejidad del sistema a desarrollar y del objeto de IoT a agentificar.

Se describió una arquitectura reactiva para obtener la capacidad de razonamiento basada en la representación del medio ambiente de forma rápida y a corto plazo e ideal para poder analizar los datos y reaccionar según el diagnóstico del estado del paciente. Gracias a los agentes reactivos, fue factible desarrollar comportamientos para leer, procesar y llegar a conclusiones de forma automática en base a los cambios del entorno donde se desenvuelven los agentes, esto es, acorde a las condiciones fisiológicas, temperatura y estado cardiorrespiratorio del paciente, as posibles reacciones a sus acciones y saber sus posibles comportamientos. Estos agentes pueden llegar a interoperar con otros ecosistemas que trabajen con información del paciente a futuro.

Se diseñó un modelo de agente y comunicación para desarrollar un prototipo para la monitorización y diagnósticos de personas a través de datos de sensores fisiológicos leídos desde un Arduino y procesados en un Raspberry PI 3 que empela un el sistema operativo Raspbian, cumpliendo así, con los requisitos de la investigación de usar tecnología de bajo costo y software libre para embeber agentes en objetos de IoT, que pueden ser usados para formar ecosistemas más complejos. Los agentes, al comunicarse usando el estándar FIPA tienen la posibilidad de interoperar con otros agentes que se comuniquen usando el mismo estándar.

Se evaluó el modelo propuesto dentro del ecosistema de agentes para el cuidado de la salud integrando la parte electrónica que incorpora el hardware abierto, a través de sensores fisiológicos, microcontroladores y software libre para la creación de los agentes; permitiendo un correcto diagnóstico en base a la monitorización en

tiempo real de la persona o paciente. Con los recursos tecnológicos empleados se comprobó que el objeto de IoT diseñado funcionó según lo planificado. Por tanto, el prototipo puede servir de base para crear futuros objetos de IoT agentificados para el área de la salud y otras a fines.

Finalmente, se midió la facilidad que brinda el sistema a los usuarios mediante la evaluación de la usabilidad con ayuda de la norma ISO 9241, evaluando las tres características que definen a la usabilidad, obteniendo como resultado una tendencia del 80% de efectividad en las tareas realizadas por el usuario, una eficiencia para verificar el tiempo invertido por tareas del (0.023 tarea/s), y un nivel de satisfacción considerada alta. Por lo tanto, se concluye que el sistema es usable por parte de los que interactuaron con el prototipo desarrollado. Además, el sistema ofrece un excelente funcionamiento y una aceptable portabilidad de este, de manera que se puede mejorar para que integre nuevos sensores, y para que sea más comercial. Los resultados se obtuvieron en experimentos de laboratorio, no se llevó a un contexto de hospital.

5.2. Recomendaciones

Considerar esta investigación para ampliar futuras investigaciones donde usen tecnologías emergentes para ofrecer una mejor calidad de vida a personas que padecen enfermedades que requieren supervisión y monitoreo periódico.

Utilizar este tipo de tecnologías emergentes dentro de hospitales en donde se encuentra gran cantidad de personas que quiere tener una atención más rápida y eficiente. Esto podría evitar que las salas de monitoreo o tomas de condiciones fisiológicas de los pacientes no colapsen.

Determinar las mejores herramientas de hardware y software libres para realizar un proceso óptimo de resultados enfocados a los ecosistemas de IoT en la salud. Un área bastante delicada en donde los resultados deben ser lo más exacto posibles para evitar que el paciente se agrave antes de recurrir a una casa de salud.

Para la lectura de los datos de los sensores de temperatura y electrocardiograma se sugiere colocar en su correcta posición para tener datos con mayor exactitud. Se recomienda desarrollar videos que enseñen a los usuarios a emplear el sistema.

Analizar y elegir las mejores herramientas IoT en el campo de la salud para lograr un mayor rendimiento en la comunicación de dispositivos y presentar reportes en tiempo real. También, se recomienda usar una plataforma para el desarrollo de agentes más ligera de manera que el tiempo para procesar los datos y establecer conclusiones sea en un menor tiempo.

BIBLIOGRAFÍA

- [1] P. Pico-Valencia, J. A. Holgado-Terriza, y X. Quinonez-Ku, «A brief survey of the main Internet-based approaches. An outlook from the Internet of things perspective», *Proc. - 3rd Int. Conf. Inf. Comput. Technol. ICICT 2020*, pp. 536-542, 2020.
- [2] R. Girau, S. Martis, y L. Atzori, «Lysis: A platform for iot distributed applications over socially connected objects», *IEEE Internet Things J.*, vol. 4, n.º 1, pp. 40-51, 2017.
- [3] L. Liu, «IoT and a sustainable city», *Energy Procedia*, vol. 153, pp. 342-346, 2018.
- [4] J. P. Cisneros Santoyo, «Diseño de un control inteligente usando técnicas de lógica difusa, aplicado a un balastro electrónico», p. 99, 2016.
- [5] A. Al-fuqaha, S. Member, M. Guizani, M. Mohammadi, y S. Member, «Internet of Things : A Survey on Enabling», *IEEE Commun. Surv. Tutorials*, vol. 17, n.º 4, pp. 2347-2376, 2015.
- [6] A. Rayes, S. Salam, y T. Road, *Internet of Things – From Hype to Reality* .
- [7] P. Pico-Valencia, J. A. Holgado-Terriza, y J. A. Senso, «Towards an Internet of Agents model based on Linked Open Data approach», *Auton. Agent. Multi. Agent. Syst.*, vol. 33, n.º 1-2, pp. 84-131, 2019.
- [8] A. Cobos, «Diseño e implementación de una arquitectura IoT basada en tecnologías Open Source.», p. 133, 2016.
- [9] D. Navani, S. Jain, y 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.
- [10] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, y S. Cla, «Middleware for Internet of things: A survey», *IEEE Internet Things J.*, vol. 3, n.º 1, pp. 70-95, 2016.
- [11] R. Perrey y M. Lycett, «Service-oriented architecture», *Proc. - 2003 Symp. Appl. Internet Work. SAINT 2003*, pp. 116-119, 2003.
- [12] A. Cama y D. Cama, «Las redes de sensores inalámbricos y el Internet de las cosas», *Inge-Cuc*, vol. 8, n.º 1, pp. 162-172, 2012.
- [13] D. Hanes, G. Salgueiro, P. Grossetete, R. Barton, y J. Henry, *IoT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things*, n.º

3491. 2017.
- [14] Nitin Naik, «Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP», 2017.
- [15] M. U. Farooq y M. Waseem, «A Review on Internet of Things (IoT)», vol. 113, n.º 1, pp. 1-7, 2020.
- [16] E. Borgia, «The Internet of Things vision : Key Features , Applications and Open Issues», *Comput. Commun.*, 2014.
- [17] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, y K. Mankodiya, «Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare», *Futur. Gener. Comput. Syst.*, vol. 78, pp. 659-676, 2018.
- [18] C. N. y P. M. Barillaro Sebastián, De Luca Graciela, Valiente Waldo, Carnuccio Esteban, García Gerardo, Volker Mariano, Giulianelli Daniel, «Diseño de sistema IoT de monitoreo y alarma para personas mayores», *XVIII Work. Investig. en Ciencias la Comput.*, pp. 712-716, 2016.
- [19] A. Garro, «Intelligent Agents and Environment», n.º iii, pp. 1-6, 2018.
- [20] A. Fougères y E. Ostrosi, «Intelligent agents for feature modelling in computer aided design», *J. Comput. Des. Eng.*, n.º November, 2017.
- [21] K. On Chin, K. S. Gan, R. Alfred, P. Anthony, y D. Lukose, «Agent Architecture: An Overview», *Trans. Sci. Technol.*, vol. 1, n.º 1, pp. 18-35, 2014.
- [22] M. Wu, W. Cao, J. Peng, J. She, y X. Chen, «Balanced Reactive-Deliberative Architecture for Multi-Agent System for Simulation League of RoboCup», vol. 7, pp. 945-955, 2009.
- [23] A. Leite y R. Girardi, «The Journal of Systems and Software A hybrid and learning agent architecture for network intrusion detection», *J. Syst. Softw.*, vol. 130, pp. 59-80, 2017.
- [24] M. W. Khan, J. Wang, M. Ma, L. Xiong, P. Li, y F. Wu, «Optimal energy management and control aspects of distributed microgrid using multi-agent systems», *Sustain. Cities Soc.*, vol. 44, pp. 855-870, 2019.
- [25] A. Dorri, S. S. Kanhere, y R. Jurdak, «Multi-Agent Systems: A Survey», *IEEE Access*, vol. 6, pp. 28573-28593, 2018.
- [26] F. Bergenti, E. Iotti, S. Monica, y A. Poggi, «Agent-Oriented Model-Driven

- Development for JADE with the JADEL Programming Language», *Comput. Lang. Syst. Struct.*, 2017.
- [27] F. Bellifemine, G. Caire, A. Poggi, y G. Rimassa, «JADE: A software framework for developing multi-agent applications. Lessons learned», *Inf. Softw. Technol.*, vol. 50, n.º 1-2, pp. 10-21, 2008.
- [28] A. R. Sadik y B. Urban, «A Holonic Control System Design for a Human & Industrial Robot Cooperative Workcell», *Proc. - 2016 Int. Conf. Auton. Robot Syst. Compet. ICARSC 2016*, n.º September 2018, pp. 118-123, 2016.
- [29] P. Pico-valencia y J. A. Holgado-terriza, «Agentification of the Internet of Things : A systematic literature review», vol. 14, n.º 10, 2018.
- [30] P. Pico-Valencia y J. A. Holgado-Terriza, «Agentification of the Internet of Things: A systematic literature review», *Int. J. Distrib. Sens. Networks*, vol. 14, n.º 10, 2018.
- [31] A. M. Mzahm, M. S. Ahmad, y A. Y. C. Tang, «Enhancing the Internet of Things (IoT) via the Concept of Agent of Things (AoT)», *J. Netw. Innov. Comput.*, vol. 2, n.º October, pp. 101-110, 2014.
- [32] M. Maksimović, V. Vujović, N. Davidović, V. Milošević, y B. Perišić, «Raspberry Pi as Internet of Things hardware : Performances and Constraints», n.º July, pp. 6-12, 2015.
- [33] P. A. Pico Valencia, «HACIA EL INTERNET DE LOS AGENTES: UN MODELO SEMÁNTICO PARA AGENTIFICAR ECOSISTEMAS DE INTERNET DE LAS COSAS», Universidad de Granada, 2018.
- [34] J. García Guajardo, «El hospital “en la mano”», *I+S Rev. la Soc. Española Informática y Salud*, n.º 129, pp. 19-21, 2018.
- [35] C. André, C. F. Pasluosta, B. Esko, D. Bandeira, y R. Righi, «Artificial Intelligence In Medicine Internet of Health Things : Toward intelligent vital signs monitoring in hospital wards», n.º September 2017, 2018.
- [36] S. Kouah y I. Kitouni, «Internet of things agents diagnosis architecture: Application to healthcare IoT system», *CEUR Workshop Proc.*, vol. 2326, pp. 62-69, 2018.
- [37] M. Ahmid, O. Kazar, S. Benharzallah, L. Kahloul, y A. Merizig, «An Intelligent and Secure Health Monitoring System Based on Agent», *2020 IEEE Int. Conf. Informatics, IoT, Enabling Technol. ICIoT 2020*, pp. 291-296, 2020.
- [38] C. O. Fernandes y C. J. P. De Lucena, «A Software Framework for Remote Patient

- Monitoring by Using Multi-Agent Systems Support», *JMIR Med. Informatics*, vol. 5, n.º 1, p. e9, 2017.
- [39] M. E. Buitrón, E. Gea, y M. V. García, «Tecnologías en información y comunicación sanitaria», *Revistapuce*, pp. 273-289, 1970.
- [40] Ecuador, «Constitución del Ecuador», *Regist. Of.*, n.º 20 de Octubre, p. 173, 2008.
- [41] Q. Sánchez, «Los Derechos de Autor (Propiedad Intelectual) en relación a la Ley Orgánica de Comunicación y a la Ley Orgánica de Control de», pp. 1-86, 2011.
- [42] T. Gupta, B. Wadhwa, Y. Sharma, O. Juneja, R. Butola, y S. Karkra, «IOT Based Vitality Measurement System», vol. 5, n.º 5, pp. 549-554, 2016.
- [43] V. Pardeshi, S. Sagar, S. Murmurwar, y P. Hage, «Health monitoring systems using IoT and Raspberry Pi - A review», *IEEE Int. Conf. Innov. Mech. Ind. Appl. ICIMIA 2017 - Proc.*, n.º Icimia, pp. 134-137, 2017.
- [44] C. Raj, C. Jain, y W. Arif, «HEMAN: Health monitoring and nous: An IoT based e-health care system for remote telemedicine», *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, vol. 2018-Janua, pp. 2115-2119, 2018.
- [45] V. Patil, S. S. Thakur, y V. Kshirsagar, «Health Monitoring System Using Internet of Things», *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. ICICCS 2018*, n.º Iciccs, pp. 1523-1525, 2019.
- [46] S. S. Mishra y A. Rasool, «IoT health care monitoring and tracking: A survey», *Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019*, n.º Icoei, pp. 1052-1057, 2019.
- [47] Z. U. Ahmed, M. G. Mortuza, M. J. Uddin, M. H. Kabir, M. Mahiuddin, y M. J. Hoque, «Internet of Things Based Patient Health Monitoring System Using Wearable Biomedical Device», *2018 Int. Conf. Innov. Eng. Technol. ICIET 2018*, pp. 1-5, 2019.
- [48] H. N. Saha, N. F. Raun, y M. Saha, «Monitoring patient's health with smart ambulance system using Internet of Things (IOTs)», *2017 8th Ind. Autom. Electromechanical Eng. Conf. IEMECON 2017*, pp. 91-95, 2017.
- [49] H. M. Abdul-Jabbar y J. K. Abed, «Real Time Pacemaker Patient Monitoring System Based on Internet of Things», *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 745, n.º 1, pp. 28-30, 2020.

- [50] G. R. D. Ganesh, K. Jaidurgamohan, V. Srinu, C. R. Kancharla, y S. V. S. Suresh, «Design of a low cost smart chair for telemedicine and IoT based health monitoring: An open source technology to facilitate better healthcare», *11th Int. Conf. Ind. Inf. Syst. ICIIIS 2016 - Conf. Proc.*, vol. 2018-Janua, pp. 89-94, 2016.
- [51] S. Divakaran, L. Manukonda, N. Sravya, M. M. Morais, y P. Janani, «IOT clinic-Internet based patient monitoring and diagnosis system», *IEEE Int. Conf. Power, Control. Signals Instrum. Eng. ICPCSI 2017*, pp. 2858-2862, 2018.
- [52] Q. Yang y J. S. Shieh, «A multi-agent prototype system for medical diagnosis», *Proc. 2008 3rd Int. Conf. Intell. Syst. Knowl. Eng. ISKE 2008*, pp. 1265-1270, 2008.
- [53] S. Gupta y S. Pujari, «A multi-agent system (MAS) based scheme for health care and medical diagnosis system», *2009 Int. Conf. Intell. Agent Multi-Agent Syst. IAMA 2009*, pp. 4-6, 2009.
- [54] P. Pico-Valencia, J. A. Holgado-Terriza, D. Herrera-Sánchez, y J. Sampietro, «Towards the Internet of agents: An analysis of the Internet of things from the intelligence and autonomy perspective», *Ing. e Investig.*, vol. 38, n.º 1, pp. 121-129, 2018.
- [55] S. Iqbal, W. Altaf, M. Aslam, W. Mahmood, y M. U. G. Khan, «Application of intelligent agents in health-care: review», *Artif. Intell. Rev.*, vol. 46, n.º 1, pp. 83-112, 2016.
- [56] A. Armentia, U. Gangoiti, R. Priego, E. Estévez, E. Irisarri, y M. Marcos, «Asistencia domiciliaria adaptada al contexto: una aproximación basada en multi-agentes», *XXXVI Jornadas de Automática*, pp. 198-205, 2015.
- [57] F. V. VALENCIA AGUIRRE, «MEJORAMIENTO DE LA USABILIDAD DEL MÓDULO DE ALUMNOS DEL PORTAL WEB MIESPE DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE A TRAVÉS DE APLICACIONES MÓVILES MULTIPLATAFORMA», *Espe*, p. 132, 2017.
- [58] S. K. Gawali y M. K. Deshmukh, «ScienceDirect ScienceDirect ScienceDirect temperature function for a long-term district heat demand forecast Assessing the feasibility of using the heat demand-outdoor Autonomy in IoT Technologies The Energy Energy Autonomy in IoT Technologies», *Energy Procedia*, vol. 156, n.º September 2018, pp. 222-226, 2019.

ANEXOS

ANEXO A

Formato Ficha con respecto a la métrica a evaluar de la usabilidad en el aspecto de satisfacción del prototipo.

Usuario: _____

Fecha: _____

Completar el siguiente cuestionario y marque con una cruz en la casilla que crea apropiada según la tarea realizada.

- 1. En general ¿Cuál fue el grado de complejidad del 1 al 7 al iniciar o arrancar el prototipo?**

Muy difícil	Difícil	Algo difícil	Normal	Algo fácil	Fácil	Muy fácil
1	2	3	4	5	6	7

- 2. En general ¿Cuál fue el grado complejidad del 1 al 7 para conectar el prototipo a Internet ya sea por cable o wifi?**

Muy difícil	Difícil	Algo difícil	Normal	Algo fácil	Fácil	Muy fácil
1	2	3	4	5	6	7

3. En general ¿Cuál fue el grado de complejidad del 1 al 7 la colocación correcta de los sensores tanto de cardio como de temperatura?

Muy difícil	Difícil	Algo difícil	Normal	Algo fácil	Fácil	Muy fácil
1	2	3	4	5	6	7

4. En general ¿Cuál fue el grado de complejidad del 1 al 7 para verificar si el servidor esta activo?

Muy difícil	Difícil	Algo difícil	Normal	Algo fácil	Fácil	Muy fácil
1	2	3	4	5	6	7

5. En general ¿Cuál fue el grado de complejidad al ejecutar el software empleado del sistema multiagentes?

Muy difícil	Difícil	Algo difícil	Normal	Algo fácil	Fácil	Muy fácil
1	2	3	4	5	6	7

ANEXO B

Ficha del formato de evaluación del prototipo de objeto de IoT agentificado en base a aspectos relacionados con el funcionamiento de los sensores, agentes y el sistema integrado.

N°	Aspectos a evaluación de la funcionalidad del prototipo	Sí/No
1	¿Captura el sensor de temperatura datos en tiempo-real? Observación:	
2	¿Capturan el sensor de cardio datos en tiempo real? Observación:	
3	¿Los dispositivos usados tienen características de un sistema IoT para el cuidado de la salud? Observación:	
4	¿Los sensores fisiológicos se encargan de medir y detectar variaciones significativas de las constantes vitales de una persona? Observación:	
5	¿Se evidencia las peticiones de datos por parte del agente coordinador a la base de datos en la que se almacena lo datos recogidos por los agentes de medición? Observación:	
6	¿Responde el agente coordinador a las solicitudes realizadas por parte del agente Diagnosticador? Observación:	
7	¿Las tareas realizadas por el agente Diagnosticador se realizó con éxito? Observación:	
8	¿Se ejecutaron de manera correcta los agentes en el dispositivo prototipado? Observación:	

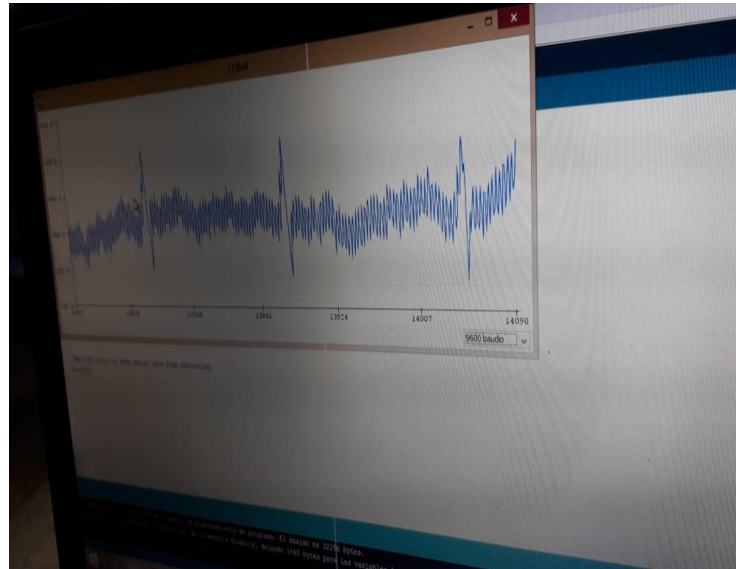
ANEXO C

Ficha del formato de evaluación de la portabilidad del prototipo de objeto de IoT agentificado.

Nº	Aspectos a evaluación de la portabilidad	SI/NO
1	¿Funciona el dispositivo prototipado sin necesidad de estar conectado a una fuente de alimentación? Observación:	
2	¿Puede ser el dispositivo prototipado llevado a escenarios distintos del hogar del usuario? Observación:	
3	¿Es fácil transportación? Observación:	
4	¿El dispositivo prototipado se ejecuta en una plataforma distinta para el cual fue construido? Observación:	
5	¿Se puede remplazar los objetos usados para medir(sensores) por otro nuevos? Observación:	

ANEXO D

Visualización de la variable temperatura y señal electrocardiograma medidos por los sensores.



Anexo 1 Monitorización en tiempo real las señales de sensor de electrocardiograma



Anexo 2 Resultados de la lectura del sensor de temperatura