



**UTMACH**

**FACULTAD DE INGENIERÍA CIVIL**

**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

**Desarrollo de una aplicación wearable con reconocimiento de gestos con técnicas machine learning para su uso en entornos Smart Home**

**CARRILLO GONZAGA EDISSON MANUEL  
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**APOLO YAGUANA JEAN PIERRE  
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**MACHALA  
2022**



**UTMACH**

**FACULTAD DE INGENIERÍA CIVIL**

**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

**Desarrollo de una aplicación wearable con reconocimiento de gestos con técnicas machine learning para su uso en entornos Smart Home**

**CARRILLO GONZAGA EDISSON MANUEL  
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**APOLO YAGUANA JEAN PIERRE  
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**MACHALA  
2022**



**UTMACH**

**FACULTAD DE INGENIERÍA CIVIL**

**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

**PROPUESTAS TECNOLÓGICAS**

**Desarrollo de una aplicación wearable con reconocimiento de gestos con técnicas machine learning para su uso en entornos Smart Home**

**CARRILLO GONZAGA EDISSON MANUEL  
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**APOLO YAGUANA JEAN PIERRE  
INGENIERO EN TECNOLOGIAS DE LA INFORMACION**

**HERNANDEZ ROJAS DIXYS LEONARDO**

**MACHALA  
2022**

# Tesis Carrillo

*por* Tesis Carrillo Carrillo

---

**Fecha de entrega:** 15-sep-2022 02:50p.m. (UTC-0500)

**Identificador de la entrega:** 1900705921

**Nombre del archivo:** Documento\_para\_TURNITIN\_Carrillo.pdf (2.26M)

**Total de palabras:** 13205

**Total de caracteres:** 69429

# Tesis Carrillo

---

## INFORME DE ORIGINALIDAD

---

3%

INDICE DE SIMILITUD

3%

FUENTES DE INTERNET

0%

PUBLICACIONES

0%

TRABAJOS DEL  
ESTUDIANTE

---

## FUENTES PRIMARIAS

---

1	<a href="https://es.scribd.com">es.scribd.com</a> Fuente de Internet	<1 %
2	<a href="https://hdl.handle.net">hdl.handle.net</a> Fuente de Internet	<1 %
3	<a href="https://www.coursehero.com">www.coursehero.com</a> Fuente de Internet	<1 %
4	<a href="https://estadictica.blogspot.com">estadictica.blogspot.com</a> Fuente de Internet	<1 %
5	<a href="https://doku.pub">doku.pub</a> Fuente de Internet	<1 %
6	Submitted to Universidad Carlos III de Madrid Trabajo del estudiante	<1 %
7	<a href="https://repositorio.utmachala.edu.ec">repositorio.utmachala.edu.ec</a> Fuente de Internet	<1 %
8	Submitted to Universidad Técnica de Machala Trabajo del estudiante	<1 %
9	<a href="https://es.slideshare.net">es.slideshare.net</a> Fuente de Internet	<1 %

---

10	<a href="http://globaljournals.org">globaljournals.org</a> Fuente de Internet	<1 %
11	<a href="http://moam.info">moam.info</a> Fuente de Internet	<1 %
12	<a href="http://repositorio.unsa.edu.pe">repositorio.unsa.edu.pe</a> Fuente de Internet	<1 %
13	<a href="http://bibdigital.epn.edu.ec">bibdigital.epn.edu.ec</a> Fuente de Internet	<1 %
14	<a href="http://es.affdu.com">es.affdu.com</a> Fuente de Internet	<1 %
15	<a href="http://sedici.unlp.edu.ar">sedici.unlp.edu.ar</a> Fuente de Internet	<1 %
16	<a href="http://sfelections.sfgov.org">sfelections.sfgov.org</a> Fuente de Internet	<1 %
17	<a href="http://artetatuaje.blogspot.com">artetatuaje.blogspot.com</a> Fuente de Internet	<1 %
18	<a href="http://nosqlcenter.es">nosqlcenter.es</a> Fuente de Internet	<1 %
19	<a href="http://www.cepe.unam.mx">www.cepe.unam.mx</a> Fuente de Internet	<1 %
20	<a href="http://www.elmeridianodecordoba.com">www.elmeridianodecordoba.com</a> Fuente de Internet	<1 %
21	<a href="http://www.formacionprofesional.info">www.formacionprofesional.info</a> Fuente de Internet	<1 %

22

[www.kelsey-seybold.com](http://www.kelsey-seybold.com)

Fuente de Internet

<1 %

---

23

[www.neuropower.com](http://www.neuropower.com)

Fuente de Internet

<1 %

---

24

[www.que.es](http://www.que.es)

Fuente de Internet

<1 %

---

Excluir citas

Activo

Excluir coincidencias < 10 words

Excluir bibliografía

Activo

## CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

Los que suscriben, CARRILLO GONZAGA EDISSON MANUEL y APOLO YAGUANA JEAN PIERRE, en calidad de autores del siguiente trabajo escrito titulado Desarrollo de una aplicación wearable con reconocimiento de gestos con técnicas machine learning para su uso en entornos Smart Home, otorgan a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tienen potestad para otorgar los derechos contenidos en esta licencia.

Los autores declaran que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

Los autores como garantes de la autoría de la obra y en relación a la misma, declaran que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asumen la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.



---

CARRILLO GONZAGA EDISSON MANUEL

0706980562



---

APOLO YAGUANA JEAN PIERRE

0706805868

## **DEDICATORIA**

El presente proyecto está dedicado principalmente a Dios, que me ha permitido avanzar en este trayecto de nuestras vidas guiándonos durante el camino hacia nuestra formación profesional, siendo el soporte en tiempos difíciles, así como nuestro protector en todo momento.

De igual manera, a mis padres que han sido el pilar incondicional, dándome el ejemplo de lograr nuestras metas con esfuerzo y dedicación. A mi familia y amigos que han estado presentes para darnos una mano cuando la necesitábamos.

Finalmente, a mis docentes, que son la fuente de conocimiento de la que hemos requerido para llegar tan lejos, por el tiempo que emplean en instruirnos y guiarnos, por la paciencia de socorrer a nuestras dudas y el amor puesto a tan noble profesión.

**Jean Pierre Apolo Yaguana**

Dedico el presente trabajo primeramente a Dios y a mis padres que han sido una pieza fundamental en toda mi vida y que a pesar de las circunstancias siempre han estado presentes para mí y mis hermanos. También le dedico a mi familia, a aquellos que están y a los que nos dejaron en estos años difíciles de pandemia. De igual forma a mis amigos y compañeros, aquellos que han estado conmigo desde que inicie mi carrera o incluso desde antes. Por último, a mis docentes, por sus consejos, sus enseñanzas y por todo lo que han hecho durante estos 5 años de carrera.

**Edisson Manuel Carrillo Gonzaga**

## **AGRADECIMIENTO**

Agradezco a mis padres el apoyo brindado a lo largo de mi carrera universitaria, que ha sido fundamental para lograr su culminación y realización del presente trabajo de titulación. A mis familiares y amigos por ser una parte fundamental en nuestro desarrollo personal y académico. A nuestro tutor Ing. Dixys Leonardo Hernández Rojas, Msc por brindarme su apoyo al resolver nuestras dudas durante el trabajo de titulación y su preocupación en el desarrollo del mismo. A los docentes involucrados en el trabajo de titulación, por su dedicación al instruirnos y guiarnos paso a paso en este trabajo, con el fin de lograr un desarrollo adecuado.

**Jean Pierre Apolo Yaguana**

Agradezco a Dios y a mis padres, por su sacrificio para darme la mejor educación posible en todos sus ámbitos, por enseñarme a no rendirme ante las dificultades, por ser un pilar fundamental en mi vida. Agradezco a mi familia y amigos que me han acompañado en diferentes etapas de mi vida, siempre dándome apoyo. A nuestro tutor Ing. Dixys Leonardo Hernández Rojas, Msc por guiarnos durante la realización del presente trabajo, así mismo al resto de docentes que nos han impartido su conocimiento en distintas áreas y que nos fueron de utilidad para desarrollar el presente trabajo.

**Edisson Manuel Carrillo Gonzaga**

## RESUMEN

Tras la llegada de la pandemia de Covid-19, se evidenció un crecimiento acelerado en la compra de dispositivos Smart Home, una de las razones principales es que las personas tuvieron que pasar mucho más tiempo en sus hogares, lo que llevó a encontrar soluciones eficientes con respecto a la productividad, seguridad y entretenimiento. El crecimiento se mantiene constante en Latinoamérica al igual que el resto del mundo, razón por la cual se proyecta que para 2025, se alcance un aproximado de 1200 millones de dispositivos conectados a la red, de los cuales el 65% corresponderá a dispositivos smart home. Esto se traduce a entornos smart home más robustos, para lo cual es necesario contar con mejores opciones para el control de los mismos.

En la actualidad el control de dispositivos smart home se lleva a cabo mayormente por comandos de voz e interfaces gráficas para lo cual se debe tener a la mano el dispositivo que recibe los comandos como son las bocinas inteligentes, dispositivos móviles, entre otros. Además de ello, el costo de estos dispositivos no siempre está al alcance de todo público, el precio varía dependiendo de las capacidades del mismo. Por estas razones se propuso desarrollar una aplicación wearable con capacidad de reconocimiento de gestos, mediante aprendizaje automático con técnicas machine learning, para el control de dispositivos integrados a una Smart Home.

Durante el proceso de desarrollo de la aplicación wearable se utilizó la metodología IoT Methodology que consta de 3 fases principales, brainstorm, build y tune, cada una de estas fases cuenta a su vez con 3 subfases que complementan la metodología con diferentes tareas que se desarrollaron durante todo el proceso de creación de la aplicación wearable. Esta metodología es una metodología ágil, lo cual permitió ir mejorando cada uno de los diferentes elementos del prototipo con cada una de las iteraciones que se realizaron.

Se realizó una recopilación de las herramientas, materiales y técnicas que serían de utilidad durante el proceso, además, se definió una arquitectura de cuatro capas y los requisitos funcionales de cada una de ellas para el diseño e implementación del prototipo. En la capa de Aplicación, se ubicó la nube de Alexa, en la capa de Servicio a Oracle Cloud y Node-Red, en la capa de Red a Bluetooth y las peticiones HTTP, en la capa de Dispositivos a los sensores de Hexiwear y los diferentes dispositivos smart home también denominados actuadores.

Una vez completado el prototipo, se elaboraron dos escenarios de prueba, el primero se utilizó para la evaluación del rendimiento de tres modelos machine learning: KNN, Decision Tree y Random Forest, para lo cual se utilizaron las métricas Precision, Recall, F1-Score y Cross-Validation. El segundo escenario de prueba se utilizó para evaluar la latencia y rendimiento de los

diferentes dispositivos del prototipo, para ello se midieron diferentes tiempos en diferentes etapas de ejecución del prototipo para así obtener un tiempo total, estas pruebas se realizaron cien veces por cada una de las acciones de cada uno de los cinco dispositivos smart home.

Como conclusión, se realizó una búsqueda bibliográfica como punto de partida del presente trabajo, se diseñó el prototipo usando la metodología IoT Methodology y utilizando una arquitectura de cuatro capas. Se obtuvo un modelo machine learning haciendo una comparación de rendimiento entre los algoritmos KNN, Decision Tree y Random Forest siendo este último el que obtuvo mejores métricas de resultados. Se elaboró el firmware del dispositivo Hexiwear con la capacidad de recolectar los datos del sensor acelerómetro integrado en el wearable. Se realizaron pruebas de evaluación del prototipo con el que se midieron el rendimiento de los modelos machine learning, el tiempo de ejecución de diferentes acciones y la funcionalidad del prototipo, obteniendo resultados satisfactorios.

Con esta investigación se pretende ser un punto de partida para futuros trabajos de investigación y mejorar aquellos que ya han sido desarrollado en el ámbito del Internet de las Cosas, además de ofrecer nuevas alternativas eficientes y económicas para el control de smart home a los usuarios finales, y ser un medio de accesibilidad para usuarios que poseen discapacidad visual, auditiva o en su lenguaje y que no pueden hacer uso de los dispositivos de control tradicionales que se basan en voz o interfaces gráficas.

## **PALABRAS CLAVE**

Hexiwear, Iot, Machine Learning, Smart Home, Wearable

## **ABSTRACT**

After Covid-19 pandemic arrival, there was an accelerated growth in the Smart Home devices purchase, one of the main reasons is that people had to spend much more time in their homes, which led to finding efficient solutions regarding productivity, security and entertainment. Growth remains constant in Latin America as well as in the rest of the world, which is why it is projected that approximately 1.2 billion devices connected to the network will be reached by 2025, 65% will correspond to smart home devices. This translates into more robust smart home environments, for which it is necessary to have better options to control them.

Currently, the control of smart home devices is carried out mostly by voice commands and graphical interfaces, for which the device that receives the commands must be at hand, such as smart speakers, mobile devices, among others. In addition to this, the cost of these devices is not always available to everyone, the price varies depending on its capabilities. For these reasons, it was proposed to develop a wearable application with gesture recognition capacity, through automatic learning with machine learning techniques, to control devices integrated into a Smart Home.

During the development process of the wearable application, the "IoT Methodology" methodology was used, which consists of 3 main phases, brainstorm, build and tune, each of these phases in turn has 3 subphases that complement the methodology with different tasks that were developed throughout the process of creating the wearable application. This methodology is an agile methodology, which allowed to improve each of the different elements of the prototype with each of the iterations that were carried out.

A compilation of the tools, materials and techniques that would be useful during the process was made, in addition, a four-layer architecture and the functional requirements of each of them were defined for the design and implementation of the prototype. In the Application layer, the Alexa cloud was located, in the Service layer to Oracle Cloud and Node-Red, in the Network layer to Bluetooth and HTTP requests, in the Device layer to the Hexiwear sensors and the different smart home devices also called actuators.

Once the prototype was completed, two test scenarios were developed, the first was used to evaluate the performance of three machine learning models: KNN, Decision Tree and Random Forest, for which the Precision, Recall, F1-Score metrics were used. and Cross Validation. The second test scenario was used to evaluate the latency and performance of the different devices of the prototype, for this different times were measured in different stages of execution of the

prototype in order to obtain a total time, these tests were performed one hundred times for each of the actions of each of the five smart home devices.

In conclusion, a bibliographic search was carried out as a starting point for this work, the prototype was designed using the "IoT Methodology" methodology and using a four-layer architecture. A machine learning model was obtained by comparing the performance between the KNN algorithms, Decision Tree and Random Forest, the latter being the one that obtained the best results metrics. The Hexiwear device firmware was developed with the ability to collect the accelerometer sensor data integrated in the wearable. Evaluation tests were carried out on the prototype with which the performance of the machine learning models, the execution time of different actions and the functionality of the prototype, obtaining satisfactory results.

This research is intended to be a starting point for future research work and to improve those that have already been developed in the field of the Internet of Things, in addition to offering new efficient and economical alternatives for the control of smart homes to end users. , and be a means of accessibility for users who have visual, hearing or language disabilities and who cannot use traditional control devices based on voice or graphic interfaces.

## **KEYWORDS**

Hexiwear, Iot, Machine Learning, Smart Home, Wearable

## ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS .....	12
ÍNDICE DE FIGURAS .....	13
GLOSARIO.....	14
INTRODUCCIÓN .....	15
i. Declaración y formulación del Problema .....	15
ii. Objeto de estudio y Campo de acción .....	17
iii. Objetivos.....	17
iv. Hipótesis y variables o Preguntas de investigación.....	17
v. Justificación.....	19
vi. Distribución del trabajo de integración curricular .....	19
1. CAPITULO I. MARCO TEÓRICO .....	21
1.1. Antecedentes de la Investigación .....	21
1.2. Antecedentes Teóricos.....	24
1.3. Antecedentes Contextuales.....	34
1.3.1. Ámbito de aplicación .....	35
1.3.2. Establecimiento de requerimientos .....	35
2. CAPITULO II. DESARROLLO DEL PROTOTIPO.....	36
2.1. Definición del prototipo .....	36
2.2. Metodología de desarrollo del prototipo .....	37
2.2.1. Enfoque, alcance y diseño de investigación.....	37
2.2.2. Unidades de análisis .....	38
2.2.3. Técnicas e instrumentos de recopilación de datos .....	38
2.2.4. Técnicas de procesamiento de datos para la obtención de resultados.....	38
2.2.5. Metodología o métodos específicos .....	38
2.2.6. Herramientas y/o Materiales .....	39
2.3. Desarrollo del prototipo.....	39

2.3.1. Brainstorm.....	40
2.3.2. Build.....	44
2.3.3. Tune .....	52
2.4. Ejecución del prototipo.....	52
3. CAPITULO III. EVALUACIÓN DEL PROTOTIPO.....	56
3.1. Plan de evaluación del prototipo .....	56
3.1.1. Pruebas .....	56
3.1.2. Requerimientos.....	58
3.1.3. Diagrama o estructura de los escenarios .....	60
3.2. Resultados de la evaluación.....	62
4. CONCLUSIONES .....	66
5. RECOMENDACIONES.....	67
6. REFERENCIAS BIBLIOGRÁFICAS.....	69
7. ANEXOS .....	72

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Variables y dimensionamiento.....	18
<b>Tabla 2:</b> Categorización de variables. ....	18
<b>Tabla 3:</b> Preguntas de investigación.....	21
<b>Tabla 4:</b> Criterios de inclusión y exclusión.....	22
<b>Tabla 5:</b> Ventajas y desventajas de smart home.....	28
<b>Tabla 6:</b> Comparativa entre VGR y SGR.....	30
<b>Tabla 7:</b> Ventajas y desventajas de Hexiwear.....	34
<b>Tabla 8:</b> Técnicas e instrumentos de recopilación de datos. ....	38
<b>Tabla 9:</b> Metodología o métodos específicos.....	39
<b>Tabla 10:</b> Herramientas y/o Materiales. ....	39
<b>Tabla 11:</b> Roles del equipo de trabajo.....	40
<b>Tabla 12:</b> Requisitos funcionales por capa para el desarrollo del prototipo .....	41
<b>Tabla 13:</b> Requisitos no funcionales .....	42
<b>Tabla 14:</b> Descripción de gestos y acciones de los dispositivos smart home .....	44
<b>Tabla 15:</b> Elementos por capa de la arquitectura .....	44
<b>Tabla 16:</b> Pruebas para la evaluación del prototipo .....	56
<b>Tabla 17:</b> Requerimientos para la evaluación del prototipo.....	59
<b>Tabla 18:</b> Modelo de dispositivos utilizados para las pruebas de prototipo .....	60
<b>Tabla 19:</b> Pruebas y métricas que se obtienen para el Escenario 1.....	61
<b>Tabla 20:</b> Pruebas y métricas que se obtienen para el Escenario 2.....	62
<b>Tabla 21:</b> Rendimiento del algoritmo KNN.....	63
<b>Tabla 22:</b> Rendimiento del algoritmo Decision Tree.....	64
<b>Tabla 23:</b> Rendimiento del algoritmo Random Forest.....	65
<b>Tabla 24:</b> Latencia y funcionalidad de acciones .....	66

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Declaración del Problema .....	15
<b>Figura 2:</b> Proceso de búsqueda. ....	22
<b>Figura 3:</b> Resultados por área de estudio. ....	23
<b>Figura 4:</b> Trabajos desarrollados por año a partir de 2016 .....	23
<b>Figura 5:</b> Concurrencia y relación entre Palabras Claves .....	24
<b>Figura 6:</b> Antecedentes teóricos.....	25
<b>Figura 7:</b> Arquitectura de prototipo .....	36
<b>Figura 8:</b> IoT Methodology.....	40
<b>Figura 9:</b> Mapa de contexto del prototipo .....	42
<b>Figura 10:</b> Flujo de funcionamiento del Prototipo .....	43
<b>Figura 11:</b> Implementación del firmware del wearable .....	45
<b>Figura 12:</b> Algoritmo para la creación de datasets.....	46
<b>Figura 13:</b> Algoritmo para la extracción de características de los datasets. ....	46
<b>Figura 14:</b> Visualización de picos para cada gesto. ....	47
<b>Figura 15:</b> Algoritmo para la extracción de características del ruido. ....	48
<b>Figura 16:</b> Algoritmo para la creación del modelo ML .....	48
<b>Figura 17:</b> Método principal del entrenamiento.....	49
<b>Figura 18:</b> Servidor HTTP. ....	49
<b>Figura 19:</b> Autenticación y re-autenticación de Alexa. ....	49
<b>Figura 20:</b> Configuración de nueva cuenta de Alexa.....	50
<b>Figura 21:</b> Ventana de inicio de sesión de Alexa.....	50
<b>Figura 22:</b> Flujo para el control del dispositivo Amazon Echo Dot. ....	51
<b>Figura 23:</b> Recepción de datos en aplicación Hexiwear. ....	51
<b>Figura 24:</b> Método para la recolección de datos. ....	52
<b>Figura 25:</b> Funcionamiento del gesto “Mano hacia arriba”. ....	53
<b>Figura 26:</b> Funcionamiento del gesto “Mano hacia abajo”.....	53
<b>Figura 27:</b> Funcionamiento del gesto “Mano hacia la derecha”. ....	54
<b>Figura 28:</b> Funcionamiento del gesto “Mano hacia la izquierda”.....	54
<b>Figura 29:</b> Funcionamiento del gesto “Mano hacia adelante”. ....	55
<b>Figura 30:</b> Matriz de confusión.....	57
<b>Figura 31:</b> Escenario 1: Medición de rendimiento de algoritmos Machine Learning .....	60
<b>Figura 32:</b> Escenario 2: Realización de gestos y cambio de estado de dispositivos.....	61

## GLOSARIO

### F

**Firmware:** Es el software que controla la parte lógica de un dispositivo electrónico.

**Framework:** Un framework o marco de trabajo es un conjunto de herramientas, técnicas y estructuras para el desarrollo de un software determinado.

### M

**Metadatos:** Los metadatos son datos que resumen de manera detallada información relevante acerca de un archivo, recurso, entre otros.

**Microcontrolador:** Un microcontrolador es un circuito integrado destinado a realizar operaciones en sistemas embebidos.

### P

**Plataforma de hardware:** El termino se refiere a una arquitectura de procesador o computador que cuentan con diversos componentes para el desarrollo de aplicaciones más complejas.

### S

**Sensores:** Los sensores son dispositivos de entrada que proveen un dato de salida con respecto a una unidad física.

### W

**Wearable:** Los wearables son una categoría de dispositivos electrónicos que pueden ser usados como accesorios en prendas de vestir o en el cuerpo del usuario.

# INTRODUCCIÓN

El internet de las cosas ha pasado a tomar un papel fundamental en la sociedad gracias al desarrollo de nuevas tecnologías que se entrelazan entre sí como son el 5G, una red móvil de alta velocidad que permitirá interconectar una mayor cantidad de dispositivos IoT debido a que cada vez, más usuarios, máquinas y humanos participan activamente en internet en lo social y laboral[1]. Smart home es una rama de IoT ambientada a interconectar dispositivos que se usan en el hogar. La llegada de la pandemia de Covid-19, fue contraproducente para algunas industrias, pero para otras fue un impulso, tal es caso de smart home que a pesar de la crisis económica que trajo consigo la pandemia tuvo un crecimiento considerable a nivel mundial[2]. En el caso de Latinoamérica se estima que se llegará en 2025 a 1200 millones de dispositivos IoT, un 65% de ellos serán dispositivos smart home [3]. El gran crecimiento de dispositivos conlleva a buscar una nueva manera de interconectar y controlar los dispositivos de una manera eficiente, es por ello que se desarrolló la presente investigación con el objetivo de desarrollar una aplicación wearable con capacidad de reconocimiento de gestos, mediante aprendizaje automático con técnicas machine learning, para el control de dispositivos integrados a una Smart Home.

## i. Declaración y formulación del Problema

### Declaración del problema



Figura 1: Declaración del Problema

La aparición del virus del Covid-19 cambió el estilo de vida de las personas, obligándolas a pasar mucho más tiempo en casa y preocupándose por lo que sucede en ella. Wright[2] hace énfasis que, a pesar de la crisis económica a casa del virus y el distanciamiento social, la demanda de dispositivos smart home creció durante 2021, siendo televisores, equipos de ejercitación y equipos de seguridad los más demandados. En Latinoamérica el panorama es muy similar al resto del mundo en donde se mantiene un crecimiento constante en el uso de dispositivos smart home. La empresa especializada en datos, Statista[3], proyecta que para 2025 los dispositivos IoT conectados alcanzarán una cifra de 1200 millones, pero de estos un 64% serán dispositivos smart home, wearables y automóviles. La conectividad ha mejorado en Ecuador durante los últimos años, gracias a ello empezaron a surgir empresas con soluciones IoT enfocadas a la seguridad, pero otro uso importante de esta tecnología en el país está en el área de salud. El diario El Telégrafo[4], en su investigación sobre IoT, menciona que los relojes con medidores de parámetros de salud, funciones de mensajes y notificaciones de llamadas son muy comunes en el país, especialmente aquellas que practican actividades físicas.

Con la expansión acelerada que se ha evidenciado de los dispositivos IoT, se pueden encontrar en el mercado diversos dispositivos de smart home que se controlan con voz, interruptores o apps móviles, pero muchas veces se busca simplificar mucho más estas tareas con dispositivos que son cotidianos para las personas como los relojes o pulseras. Los smartphones poseen todas las características para controlar un entorno Smart Home, pero aún siguen siendo engorrosos al tener que acceder al dispositivo y luego a una app específica para realizar acciones, además que se debe tener las manos desocupadas y acudir al sitio en el que esté el smartphone, lo toma más tiempo, mientras que con un wearable sería una tarea más sencilla. Existen varios wearables que permiten el control de actividad física y de salud de las personas, pero son pocos los que permiten el control de Smart home y aquellos que poseen estas capacidades son compatibles solo con ciertos dispositivos de Smart home y aún menos aquellos que tienen capacidad de reconocer gestos.

## **Formulación del problema**

### **Problema principal**

¿Cómo desarrollar una aplicación wearable con reconocimiento de gestos e integración con un hogar inteligente en la ciudad de Machala?

## **Problemas específicos**

- ¿Cómo controlar un entorno Smart Home con una aplicación wearable?
- ¿Cómo obtener los datos de movimiento de la mano a través de sensores?
- ¿Cómo crear un firmware para el funcionamiento de un wearable?
- ¿Cómo integrar la aplicación wearable para el control de Smart Home?

## **ii. Objeto de estudio y Campo de acción**

### **Objeto de estudio**

- Aplicación wearable con reconocimiento de gestos para entornos Smart Home

### **Campo de acción**

- Desarrollo de una aplicación wearable con aprendizaje autónomo supervisado y predictivo para Smart Home.

## **iii. Objetivos**

### **Objetivo General**

- Desarrollar una aplicación wearable con capacidad de reconocimiento de gestos, mediante aprendizaje automático con técnicas machine learning, para el control de dispositivos integrados a una Smart Home.

### **Objetivos específicos**

- Realizar una búsqueda bibliográfica para la elaboración del estado del arte y marco teórico.
- Diseñar el prototipo utilizando una metodología para el desarrollo de aplicaciones wearable.
- Implementar un modelo de aprendizaje automático para reconocimiento de gestos.
- Elaborar el firmware(aplicación) con las funciones necesarias para el wearable.
- Realizar las pruebas de funcionamiento del prototipo en un entorno Smart Home.

## **iv. Hipótesis y variables o Preguntas de investigación**

### **Hipótesis**

El desarrollo de la aplicación wearable, con técnicas machine learning para el reconocimiento de gestos facilita a los usuarios el control de sus dispositivos conectados al entorno Smart Home.

## Declaración y categorización de variables

### Hipótesis principal (o preguntas de investigación)

- El desarrollo de la aplicación wearable, con técnicas machine learning para el reconocimiento de gestos facilita a los usuarios el control de sus dispositivos conectados al entorno Smart Home.

### Variables y dimensionamiento (o categorización)

Variables	Conceptos
<b>Variable Independiente:</b>  Aplicación wearable con técnicas machine learning	Desarrollo de la aplicación wearable con técnicas machine learning se refiere al desarrollo de la aplicación para un dispositivo wearable que logre reconocer gestos mediante un aprendizaje con técnicas machine learning.
<b>Variable Dependiente:</b>  Control de dispositivos conectados al entorno Smart Home	El control de dispositivos conectados al entorno Smart Home hace referencia a la implementación del modelo machine learning para el reconocimiento de gestos, que a su vez permite el control de aparatos conectados al entorno Smart Home.

Tabla 1: Variables y dimensionamiento

Variables	Categorías	Indicadores	Técnicas
<b>Variable Independiente:</b>  Aplicación wearable con técnicas machine learning	1. Internet of Things 2. Machine learning 3. Plataforma de hardware	1. Diseño de arquitectura IoT. 2. Diseño de modelo machine learning 3. Extracción de datos para aprendizaje del modelo. 4. Entrenamiento del modelo machine learning. 5. Modificación de firmware del dispositivo wearable.	1. Recopilación de información para definir la arquitectura Iot. 2. Análisis de técnicas para el diseño del modelo machine learning. 3. Carga de datos para reconocimiento de gestos. 4. Programación de firmware.
<b>Variable Dependiente:</b>  Control de dispositivos conectados al entorno Smart Home	1. Experiencia de usuario 2. Funcionamiento	1. Comodidad del dispositivo wearable 2. Conexión estable del wearable con el entorno Smart Home. 3. Gestos intuitivos. 4. Reconocimiento de gestos.	1. Análisis del modelo machine learning. 2. Pruebas de rendimiento.

Tabla 2: Categorización de variables.

## **v. Justificación**

El Internet de las Cosas ha ganado terreno en el campo de la automatización, llevando al surgimiento de nuevas sub áreas como lo es Smart Home, que consiste en automatizar los objetos con los que las personas conviven en sus casas, como son la iluminación, electrodomésticos y otros aparatos eléctricos. Los entornos Smart Home se controlan principalmente por medio de aplicaciones, sean estas móviles, web o incluso desde otros objetos como relojes o bocinas. Los smartphones poseen todas las características para controlar un entorno Smart Home, pero aún sigue siendo un problema tener que acceder al dispositivo y luego a una app específica para realizar acciones, además que se debe tener las manos desocupadas y acudir al sitio en el que esté el smartphone tomando mucho tiempo. Otro punto importante son los costos, porque un móvil puede tener los sensores necesarios para llevar a cabo estas tareas y a pesar de que es muy común que las personas tengan uno, los wearables siguen siendo una solución menos costosa, como mencionan Faisal et al[5]. quienes afirman que los altos costos pueden ser reducidos por la implementación de soluciones en plataformas wearable. Es por ello que con el presente proyecto se busca desarrollar una aplicación wearable con capacidad de reconocimiento de gestos, mediante aprendizaje automático con técnicas machine learning, para el control de dispositivos integrados a una Smart Home. Además, se espera que este trabajo sea de gran ayuda para investigaciones futuras que busquen mejorar sistemas de automatización de hogares que beneficien a todas las personas interesadas y aquellas que por diferentes condiciones no pueden hacer uso de otros medios como su voz o a través de interfaces graficas como son las personas mudas o ciegas.

## **vi. Distribución del trabajo de integración curricular**

El trabajo se encuentra distribuido en una introducción y tres capítulos que se detallan a continuación:

**INTRODUCCIÓN:** En este primer apartado se declaró y formulo el problema, el objeto de estudio, campo de acción, los objetivos, hipótesis y justificación que apoyan la realización de la investigación.

**CAPITULO I:** Este capítulo muestra los antecedentes de la investigación que contó con una revisión sistemática de literatura para la posterior elaboración del marco teórico y los antecedentes contextuales.

**CAPITULO II:** El segundo capítulo se dedicó al desarrollo del prototipo, para lo cual se hizo una definición del prototipo. Se continuó con la metodología de desarrollo del prototipo con la cual se describieron el enfoque, alcance y diseño de la investigación, se describieron las unidades de

análisis, las técnicas e instrumentos de recopilación de datos, las técnicas de procesamiento de datos, métodos específicos y herramientas y/o materiales. Por último, se desarrolló y ejecutó el prototipo.

CAPITULO III: Para el tercer capítulo se realizó la evaluación del prototipo a través de un plan de evaluación para finalmente mostrar evidencias de los resultados de la evaluación.

# 1. CAPITULO I. MARCO TEÓRICO

## 1.1. Antecedentes de la Investigación

La revisión bibliográfica sobre los temas de investigación relevantes en este trabajo se realizó usando la metodología de Revisión Sistemática de la Literatura, cuyo objetivo principal dentro de los trabajos de investigación es identificar, evaluar y resumir los resultados de estudios anteriores que puedan aportar a la investigación actual[6].

### a) Preguntas de investigación

A continuación, se detallan las preguntas que se establecieron para la búsqueda de información sobre el desarrollo de aplicaciones wearable con reconocimiento de gestos para Smart Home.

Pregunta	Descripción y motivación
1. ¿Cuáles son los principales modelos de Machine Learning usados para el reconocimiento de gestos de la mano?	La pregunta busca descubrir los modelos, diseños, experiencias, conceptos sobre Machine Learning.
2. ¿Qué resultados se ha conseguido con la plataforma Hexiwear dentro de entornos Smart Home?	La pregunta busca evidenciar experiencias, beneficios, resultados, conceptos sobre la plataforma Hexiwear.
3. ¿Cuáles son las herramientas y/o técnicas usadas para la creación de una aplicación wearable para el reconocimiento de gestos para Smart Home?	La pregunta busca descubrir los frameworks, plataforma de hardware, lenguajes, algoritmos, diseños que se utilizarán.

Tabla 3: Preguntas de investigación.

### b) Palabras claves y Cadena(s) de búsqueda

Para la búsqueda de información en las distintas bases de datos, se utilizaron cadenas de búsqueda tanto en inglés y español con posibles términos que se pueden encontrar en títulos, palabras claves, metadatos, entre otros.

Español:

- (("hogar Inteligente" OR "Internet de las cosas" OR iot) AND wearable AND ("aprendizaje automático" OR "aprendizaje profundo" OR "reconocimiento de gestos"))

Inglés

- (("smart home" OR "internet of things" OR iot) AND wearable AND ("machine learning" OR "deep learning" OR "gesture recognition"))

### c) Criterios de inclusión y exclusión

Los criterios de inclusión y exclusión fueron escogidos de acuerdo a los resultados que se buscan lograr con la investigación y los requisitos que debe cumplir el presente trabajo de titulación.

#	Criterios de inclusión
1	Estudios primarios
2	Estudios publicados después de 2016
3	Estudios Relacionados a Smart Home
4	Estudios Relacionados a Machine Learning
5	Estudios Relacionados a Hexiwear o plataformas similares
#	Criterios de exclusión
1	Estudios secundarios
2	Estudios duplicados
3	Estudios anteriores a 2016
4	Estudios cuyo documento es inaccesible
5	Estudios en idiomas diferentes al inglés o español
6	Estudios con conclusiones o resultados similares
7	Estudios de menos de 3 hojas

Tabla 4: Criterios de inclusión y exclusión.

### d) Proceso y resultados de la búsqueda

#### Proceso de búsqueda

El proceso de búsqueda se realizó con las cadenas de búsqueda en diferentes bases de datos bibliográficas y la depuración contó con cuatro pasos que se detallan en la Figura 2, para lo cual además se hizo uso de la herramienta Parsifal con la finalidad de agilizar el proceso de la revisión sistemática de la literatura.

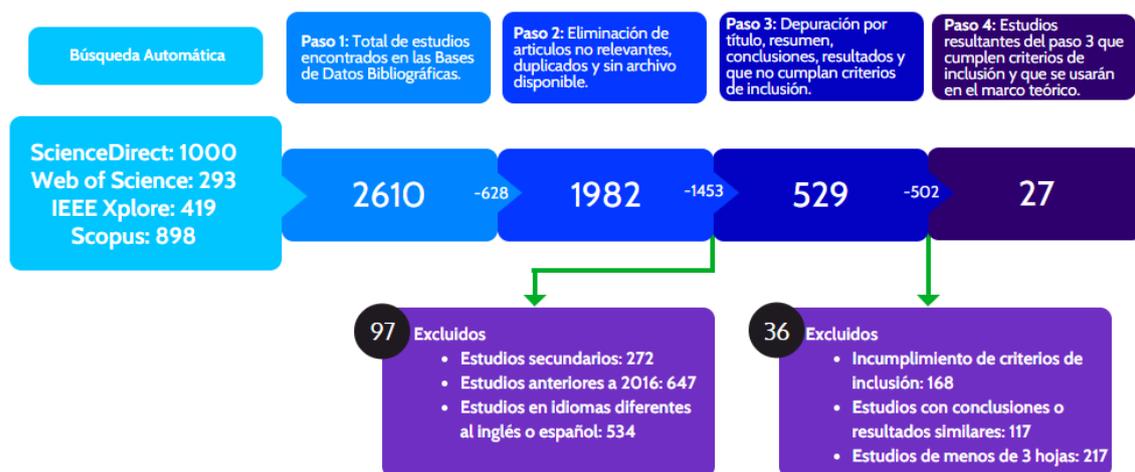
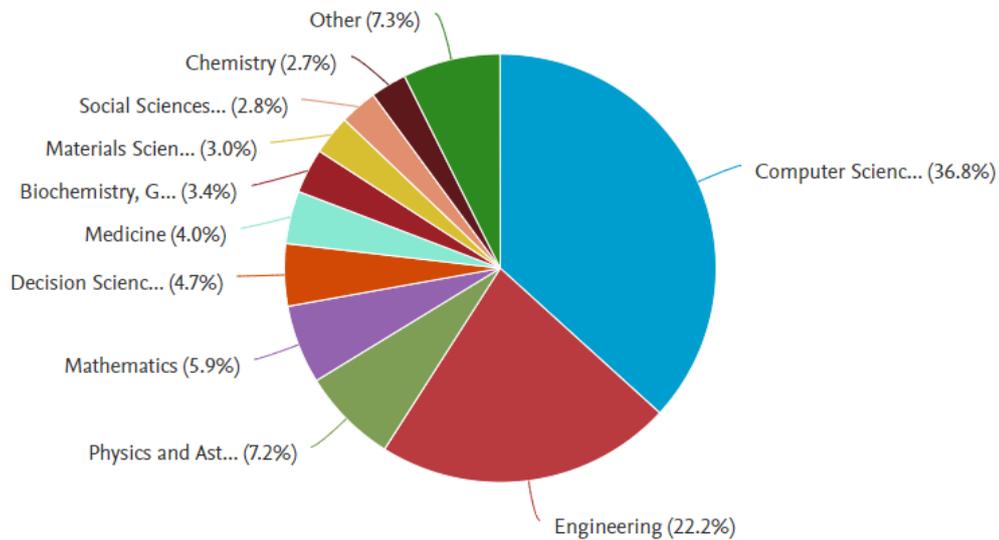


Figura 2: Proceso de búsqueda.

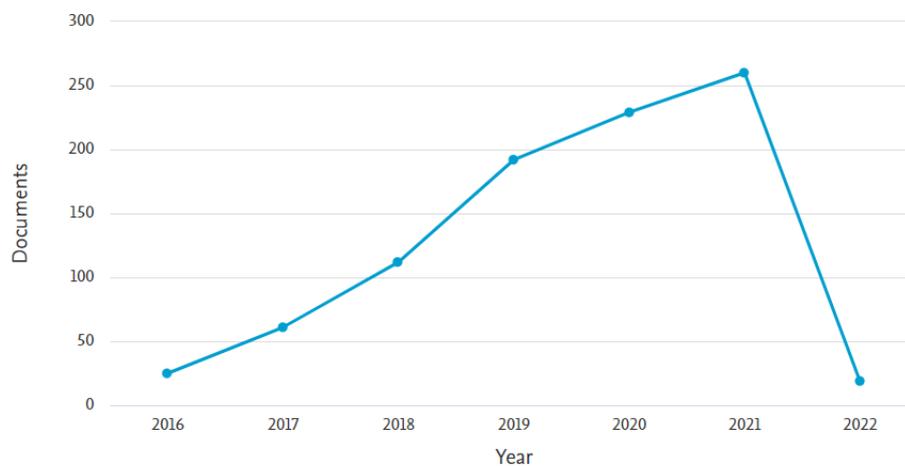
## Resultados finales de búsqueda

Los resultados de búsqueda se distribuyeron de acuerdo al área a la cual está enfocada. En la Figura 3 se muestra el número de artículos área de estudio, ocupando las ciencias computacionales el 36.8%, seguido por ingeniería con el 22.2% de los resultados de búsqueda.



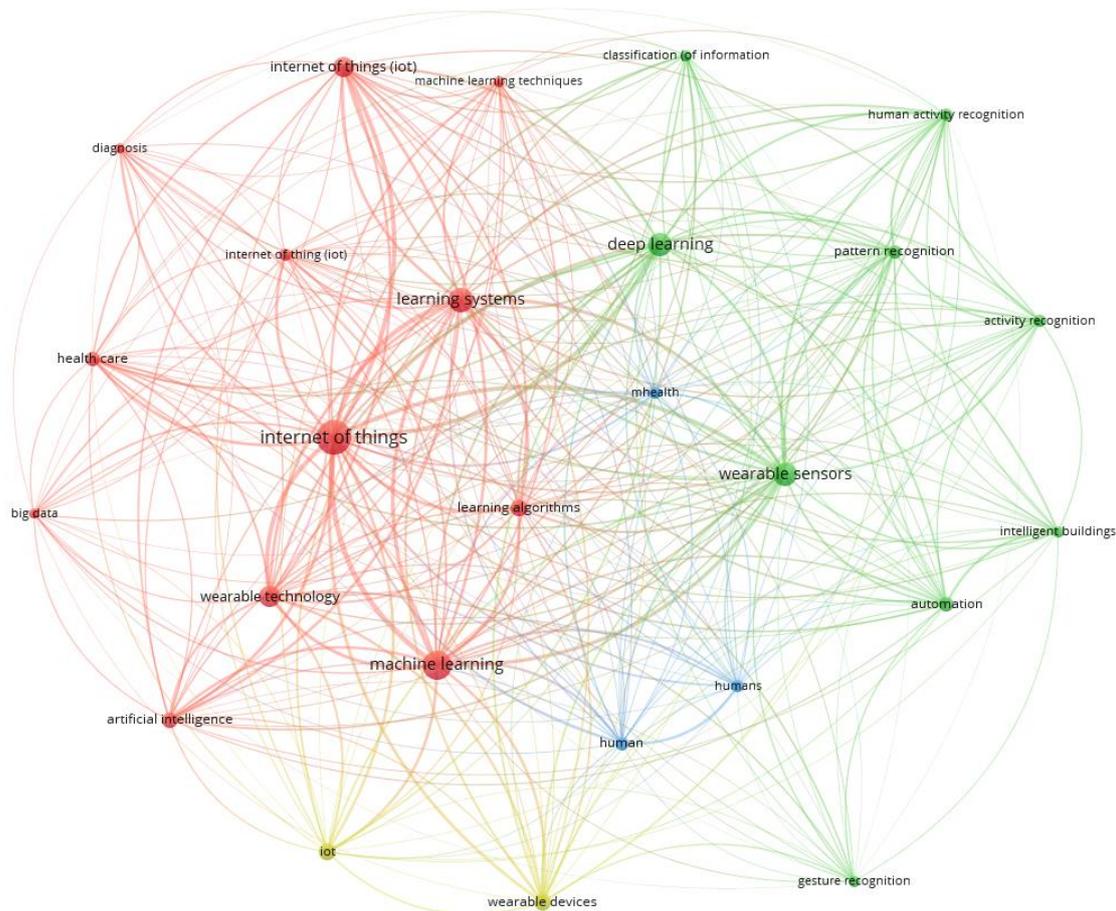
**Figura 3:** Resultados por área de estudio.

En la Figura 4 se observa un resumen de los documentos por año que fueron obtenidos a través de la cadena de búsqueda. Esta gráfica demuestra el nivel de desarrollo que ha tenido el tema con el pasar de los años, manteniendo un crecimiento constante, cabe recalcar que la cantidad de artículos para 2022 se debe que la presente revisión sistemática de literatura se desarrolló en los primeros días del año.



**Figura 4:** Trabajos desarrollados por año a partir de 2016

Se utilizó la herramienta para observar la concurrencia y relación que existe entre las palabras claves de los resultados obtenidos en la revisión sistemática de la literatura como se muestra en la Figura 5. Se puede observar que Internet of Things, seguida de Machine Learning, Learning Systems y Wearable Sensors con los términos más concurridos en la investigación, también se observa que existe una relación constante entre las diferentes palabras claves.



**Figura 5:** Concurrencia y relación entre Palabras Claves

## 1.2. Antecedentes Teóricos

Luego de la realización de la investigación y recopilación bibliográfica se realizó una clasificación de temas y subtemas como se muestra en la Figura 4. Esta estructura se desarrolla en base a los puntos más relevantes dentro de la investigación y aquella información que será de ayuda en el resto de componentes del presente trabajo.

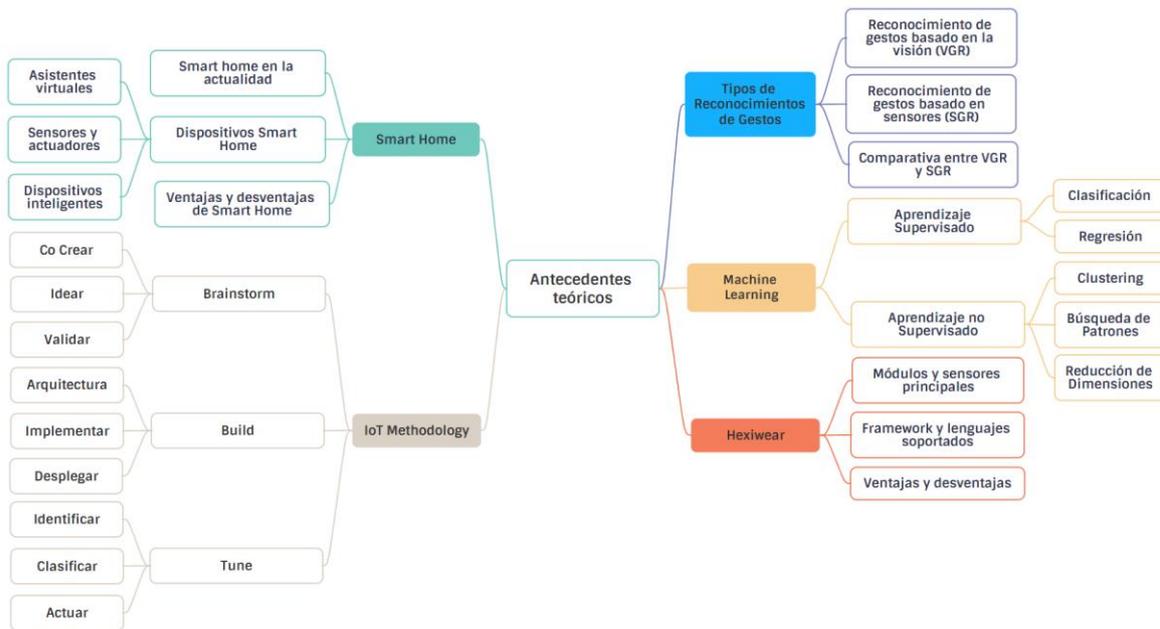


Figura 6: Antecedentes teóricos.

### 1.2.1. Smart Home

La primera vez que apareció el término interconexión de dispositivos fue en los años 70s, mientras que Internet de las Cosas fue mencionado por Kevin Ashton en 1999 [7], desde entonces los conceptos relacionados a internet de las cosas se han incrementado a medida que nuevas tecnologías son adoptadas en el mercado, ya sea por innovación o popularidad. Estos se han ido desglosando hacia determinadas áreas según las necesidades; una de estas es la domótica, que busca la seguridad y comodidad de los usuarios que implementan dichas tecnologías. Tomando como ejemplo el estudio de Ferraris et al.[8], se puede decir que la integración de un Smart Home se basa en la aplicación de productos y/o servicios interconectados en una red doméstica, mediante la cual se permite el control de dispositivos inteligentes. Pero referirse a un entorno Smart Home es tener en cuenta que es solo el usar dispositivos inteligentes para realizar ciertas actividades cotidianas, si no, se debe considerar que el uso que se dé a los mismo sea el adecuado para lograr aprovechar las bondades, citando algunas como: controles de seguridad y vigilancia remota, monitorización en tiempo real, uso remoto de ciertos dispositivos, tracking y demás[9].

#### Smart home en la actualidad

Desde los inicios de la era tecnológica, hemos sido testigos del avance que han logrado las diferentes áreas tecnológicas relacionadas a la automatización e Internet de las cosas. En un principio, estas ideas solían ser percibidas como pensamientos utópicos o muy alejados a esa época, sin embargo, la investigación constante llevada de la mano con el desarrollo tecnológico ha permitido adoptar poco a poco estos conceptos hasta convertirlos en una parte indispensable en la

vida cotidiana de miles de usuarios. De manera que el imaginar residencias completamente automatizadas ya no es un pensamiento lejano, son una realidad que ha sobrepasado las expectativas de los consumidores[10].

Actualmente, se cuenta con un mercado muy amplio con distintas alternativas enfocadas a las necesidades de todo público interesado en implementar un entorno domótico en su hogar. Las opciones abarcan desde dispositivos tan sencillos como bombillas inteligentes hasta sistemas domóticos integrados los cuales facilitan monitorizar y controlar el entorno Smart Home[11].

Sin duda, la domótica ha sido una rama sumamente importante dentro del campo de internet de las cosas, debido a la innovación y facilidad que generan sus aplicaciones se ha convertido en una de las tendencias más populares en cuanto a mejoras domésticas.

### **1.2.1.1. Herramientas para smart home**

Dentro del ámbito de internet de las cosas, existe una gran variedad de opciones en cuanto a dispositivos y herramientas que permitan implementar un entorno Smart Home. Estos pueden ser utilizados según las necesidades del usuario y la infraestructura de la residencia en donde se va a aplicar. Esto es algo que se debe tener en cuenta al seleccionar los equipos para el entorno Smart Home, ya que cada dispositivo tiene sus restricciones de uso, además de ciertas condiciones requeridas para un funcionamiento óptimo.

Entre las principales herramientas para Smart Home, se pueden destacar las siguientes:

#### **Asistentes virtuales**

Con el aumento de interés en el área domótica se han presentado varias alternativas, entre las principales destacan Alexa y Google Assistant, las cuales fueron desarrolladas en un inicio con el fin de ayudar en actividades cotidianas como recordatorios, alarmas, proveer información climática entre otras. Si bien, la idea de tener un asistente virtual al alcance de la mano, que se pueda controlar desde un Smartphone u otro dispositivo compacto fue una novedad aún no se conseguía darles un uso tan predominante; sin embargo, con el tiempo estos evolucionaron para transformarse en sistemas domóticos que son capaces de realizar tareas programadas de manera autónoma que a su vez mejoran la calidad de vida de sus usuarios[8].

Estas herramientas funcionan conectadas a la Red de área local inalámbrica del Smart Home, por si solas son capaces de cumplir con ciertas actividades; pero en el caso de un Smart Home, estos se conforman de varios dispositivos y sistemas que se encuentran conectados en la misma red. De

esta manera los asistentes son capaces de enviar solicitudes de orden a los dispositivos ya sea para realizar una acción o para el intercambio de datos[12].

## **Sensores y actuadores**

Los sistemas Smart Home se constituyen por un gran conjunto de elementos para llevar a cabo las tareas para los que han sido desarrollados. Si bien es cierto que, algunas de las funciones que se pueden realizar en un entorno Smart Home suelen ser de baja complejidad, sin duda la mayoría requiere el uso de componentes que provean datos y/o realicen las ordenes necesarias para el cumplimiento de las mismas. Estos componentes son sensores y actuadores, en el caso de los sensores son dispositivos que perciben características de su entorno y ya sea de manera analógica o digital las convierten en información que posteriormente será procesada por el sistema, algunos de los sensores más utilizados para sistemas domóticos son: sensores de luz, sonido, temperatura, humedad, posicionamiento, proximidad, giroscopio, acelerómetro, entre otros; por otro lado, los actuador a diferencia del sensor, son quienes reciben las órdenes y realizan la acción, por lo general, estos actuadores son motores, relés u otro tipo de componente electrónico y/o mecánico que controle los dispositivos finales[13].

## **Dispositivos inteligentes**

Una parte fundamental de los entornos domóticos radica en la variedad de dispositivos que se encuentran conectados conformando el sistema para Smart Home. Estos dispositivos se los puede definir como aparatos electrónicos que trabajan de manera autónoma interconectados a una o más redes, en las cuales realizan la transferencia de datos para enviar o recibir solicitudes a realizar[14]. Estos dispositivos, además, pueden tener una identidad única, poseer atributos físicos, una personalidad virtual y utilizar interfaces de comunicación inteligentes para la transferencia de datos[15].

### **1.2.1.2. Ventajas y desventajas de smart home**

A continuación, se presentan las diferentes ventajas y desventajas puntuales sobre los entornos Smart Home[16, 17].

<b>Ventajas</b>	<b>Desventajas</b>
Ayudan en tareas cotidianas, mejorando la calidad de vida del usuario.	Dependencia al uso excesivo de la tecnología.
Acceso y control remoto al entorno Smart Home.	Sumisión a lujos innecesarios.
Facilita la gestión del ambiente en la residencia.	Requiere conocimiento previo para configurar los componentes y dispositivos al entorno Smart Home.
Permite monitorizar las diferentes áreas del hogar, con el fin de minorizar accidentes o tragedias.	Incentiva la ociosidad en los miembros del hogar e interrumpe las actividades diarias.
Mejora la optimización de consumo energético de manera autónoma.	Monitorización de actividades privadas.
Aporta una gestión en la salud de los usuarios.	Ciertas aplicaciones son intrusivas e invaden la privacidad.
Monitorizar y notificar estados de salud de los individuos, para prevenir riesgos sanitarios.	Genera intranquilidad en los miembros del hogar.

**Tabla 5:** Ventajas y desventajas de smart home.

## 1.2.2. IoT Methodology

La presente investigación se desarrollará con el uso de la metodología IoT Methodology conocida también como IoTM, una metodología del tipo iterativa creada por Consultancy for Internet of Things team cuyo ciclo de vida consta de 3 fases principales que a su vez contienen 3 sub-fases cada una[18]:

### 1.2.2.1. Brainstorm:

- **Co-Crear:** Con los integrantes del equipo de trabajo y demás interesados se define el dominio del problema y se proponen soluciones creativas a través de una lluvia de ideas haciendo uso de las diferentes tecnologías y recursos disponibles.
- **Idear:** Se conceptualizarán las ideas que se obtuvieron de la etapa anterior a través de conceptos reales y demostraciones. Para ello se definirá como será la solución, su funcionamiento y como se verá cuando se implemente.
- **Validar:** Aquí se comprenderá el potencial de las estrategias propuestas, su aplicabilidad en el ecosistema en general, además se definirán los roles que corresponden a cada miembro del equipo según sus habilidades.

### 1.2.2.2. Build:

- **Arquitectura:** Se elaborará la arquitectura adecuada para el proyecto IoT, también se analizarán los riesgos que puedan existir a futuro y las oportunidades para un desarrollo sostenible.
- **Implementar:** Se desarrollará el prototipo el cual podrá ser probado y medido para asegurar que su funcionamiento y usabilidad sean adecuados para el usuario.

- **Desplegar:** Esta etapa consiste en el lanzamiento de la versión final y sus futuras actualizaciones. En el presente proyecto no se tomará en cuenta esta etapa debido a que se busca crear un prototipo.

### **1.2.2.3. Tune:**

- **Identificar:** En esta etapa se identificarán los problemas existentes en el prototipo y se buscará prevenir otras fallas que no han sido confirmadas.
- **Clasificar:** Se clasificará cada una de las fallas, teniendo en cuenta su nivel de riesgo, Muy alto, Alto y Bajo.
- **Actuar y ajustar:** Una vez clasificados los fallos, se procederá a corregirlos y realizar los ajustes correspondientes para evitar que estos puedan volver a aparecer o afectar otros componentes.

## **1.2.3. Tipos de reconocimientos de gestos**

### **1.2.3.1. Reconocimiento de gestos basado en la visión (VGR)**

El reconocimiento de gestos basado en la visión es uno de los tipos de reconocimientos de gestos que se han utilizado para estudiar el movimiento humano. Para Sharma & Singh[19] el reconocimiento basado en la visión es más amigable con los usuarios pues no es necesario que porten ningún dispositivo en su cuerpo para la recopilación de datos, pero en cambio es necesario el uso de una cámara para poder obtener las imágenes que posteriormente serán procesadas a través de otras tecnologías para identificar la acción que realiza la persona.

### **1.2.3.2. Reconocimiento de gestos basado en sensores (SGR)**

Otro tipo de reconocimiento de gestos es el basado en sensores, en este caso se necesitará de equipamiento dependiendo la zona del cuerpo de donde se desean recolectar los datos, en el caso del reconocimiento de gestos de la mano, puede ser un guante o un accesorio, por ejemplo, un reloj. A diferencia del reconocimiento basado en visión no se requiere la intervención humana pues los puntos de inicio y fin de cada gesto son detectados automáticamente, pero en cambio para el reconocimiento de gestos basado en sensores le resulta más difícil la detección de secuencias de gestos[20].

### 1.2.3.3. Comparativa entre VGR y SGR

A continuación, se describe una comparativa entre VGR y SGR, los principales tipos de reconocimientos de gestos.

VGR	SGR
Requiere de una cámara, pero no requiere dispositivos integrados al cuerpo humano.	Requiere de dispositivos como wearables con distintos sensores que recopilen los datos.
Requiere tecnologías adicionales para el análisis de datos.	Requiere tecnologías adicionales para el análisis de datos.
Puede detectar varios gestos a la vez o en secuencia.	Es más complicado el reconocimiento de gestos en secuencia.
En algunos casos no resulta muy portable, y la cámara debe tener una resolución aceptable para mejores resultados.	Es mucho más portable porque los sensores pueden estar integrados en un wearable de uso cotidiano.
Los costos de los dispositivos con este tipo de reconocimiento son más elevados.	El costo depende del tipo de sensores utilizados para el reconocimiento de gestos.

Tabla 6: Comparativa entre VGR y SGR.

### 1.2.4. Machine learning

El uso de Machine Learning se ha expandido a diferentes áreas al tratarse de una disciplina que forma parte de la Inteligencia Artificial. Machine Learning parte de una serie de algoritmos que posibilita el reconocimiento de patrones por parte de diferentes dispositivos, por ejemplo, en el estudio de Qaroush et al.[21] sobre el reconocimiento del lenguaje de señas en tiempo real, se usa algoritmos machine learning para reconocer 28 letras aisladas del alfabeto árabe

#### 1.2.4.1. Aprendizaje supervisado

El aprendizaje supervisado consiste en un modelo de aprendizaje automático en el cual se realiza un entrenamiento con datos que previamente han sido verificados como correctos, para ello se separa los datos en dos subconjuntos, el primero es el de entrenamiento y el segundo es de prueba[22]. Al final del entrenamiento se usan los datos de prueba y los resultados obtenidos se comparan con los resultados esperados para lo cual se usan métricas tales como la matriz de confusión, F1 score, precisión, AUC, entre otros.

#### Clasificación

Los algoritmos de clasificación se distinguen de los de regresión porque la salida esperada es categórica, es decir, solo podrá tener dos estados, por ejemplo: verdadero o falso, si o no, entre otros[23].

## **Regresión**

Las técnicas de regresión lineal buscan los valores de coeficientes óptimos para una función del tipo  $y=ax+b$ , para así comprobar si existe una relación entre una variable de entrada y una de salida, es por ello que se utilizan generalmente para la predicción de variables continuas[23].

### **1.2.4.2. Aprendizaje no supervisado**

Roohi et al.[24] explican que este tipo de aprendizaje está basado en la manera en que los seres humanos obtienen conocimiento, a través de la exploración y el descubrimiento de patrones, pues es precisamente esto que lo diferencia del aprendizaje supervisado, en el cual es necesario realizar un entrenamiento. Una de las ventajas sobre el aprendizaje supervisado, es que el aprendizaje es más flexible y autónomo cuando se habla del aprendizaje no supervisado.

## **Clustering**

El clustering es una técnica utilizada para analizar datos y agruparlos para obtener detalles específicos sobre los datos. La eficiencia de este tipo de técnicas de machine learning son muy efectivos con grandes cantidades de datos, por ejemplo, K-means, un algoritmo de clustering, requiere muy poca memoria, pero aun así cumple con sus tareas de una forma rápida[25].

## **Reconocimiento de Patrones**

El reconocimiento de patrones es un tipo de Machine Learning en donde se busca la coincidencia entre la información recopilada previamente en una base de datos con los nuevos datos de entrada. En este tipo de aprendizaje automático se proporcionan patrones conocidos y algunos desconocidos, para lo cual el algoritmo debe encontrar con cuales la semejanza es máxima, todo esto con ayuda de medidas de compatibilidad, como F1, el área bajo la curva, precisión, entre otras[26].

## **Reducción de Dimensiones**

La reducción de dimensiones es una técnica utilizada cuando se posee una gran cantidad de variables de entrada o también llamadas dimensiones, pues su objetivo es la reducción de las mismas. En representaciones de datos de alta dimensionalidad, la reducción de dimensiones transforma el conjunto original en un conjunto con baja dimensionalidad conservando el significado original de los datos[27].

### **1.2.5. Hexiwear**

Hexiwear es una plataforma de desarrollo elaborado por MikroElektronika, es utilizado para aplicaciones Iot, ya que ha sido pensado para establecer conexiones con dispositivos cercanos o servidores en la nube. Su diseño compacto y de cómoda portabilidad para el usuario, así como su amplia gama de módulos adicionales lo convierten en una herramienta potente para ser usado en sistemas complejos para Smart Home. Sus principales funcionalidades radican en darle al usuario la posibilidad de controlar ciertas tareas, recibir notificaciones específicas, monitorizar estadísticas de salud entre otros, todo desde su muñeca[28]. Este provee al usuario cierta documentación para su uso y desarrollo de apps, de igual manera se facilita el acceso a su software de código abierto, con el objetivo de incentivar a la comunidad a continuar aportando al crecimiento y desarrollo de esta plataforma. El hecho de tener acceso al firmware del dispositivo y contar con más módulos, da paso a un sin fin de mejoras en su practicidad.

#### **1.2.5.1. Módulos y sensores principales**

Como se mencionó con anterioridad, el kit de desarrollo de Hexiwear incluye una serie de módulos que le permiten al dispositivo añadir ciertas funciones al mismo[26–28]. Estos módulos son los siguientes:

##### **Acelerómetro**

Este sensor capta la aceleración gravitacional del dispositivo, provee los datos en los ejes X, Y y Z y usa como medida los metros sobre segundos al cuadrado  $m/s^2$ .

##### **Giroscopio**

Mide los giros y rotaciones del dispositivo, usa coordenadas en los ejes X, Y y Z.

##### **Magnetómetros**

Permite percibir y medir la intensidad de campos magnéticos tanto estáticos como dinámicos.

##### **Sensores de humedad y temperatura**

Estos componentes obtienen señales eléctricas del ambiente, las cuales son procesadas en información. La humedad se calcula en gramos por metro cúbico( $g/m^3$ ), sin embargo, es expresada en porcentajes (%). La temperatura puede ser calculada y expresada comúnmente en grados Kelvin (K), Celsius ( $^{\circ}C$ ) y/o Fahrenheit ( $^{\circ}F$ ).

## **Sensor digital de presión**

Mediante este logra se medir la presión real respecto al sensor y otros determinados elementos que procesan esta información.

## **Sensor de ritmo cardiaco**

Su común funcionamiento radica en la emisión de luz en zonas convenientes para la lectura de estos datos, de esta manera cierta cantidad de luz es reflejada debido al flujo sanguíneo y esta se alterna dependiendo del ritmo cardiaco del usuario.

### **1.2.5.2. Framework y lenguajes soportados**

#### **Python**

Python es un lenguaje de programación de alto nivel utilizado para el desarrollo de aplicaciones de propósito general. Una de sus características principales es que no se necesita traducir el código escrito a lenguaje de máquina para que se lo pueda ejecutar. Para lograr la ejecución de las aplicaciones se utiliza un programa que lo interprete. Además, es un lenguaje multiplataforma y de código abierto que facilita el desarrollo y es sencillo de entender. El lenguaje Python ha ido mejorando su compatibilidad, hasta el punto en que es utilizado para diferentes desarrollos sobre machine learning, redes neuronales e inteligencia artificial[32]. De igual manera al proveer al usuario un libre acceso para desarrollo, se ha adoptado para aplicaciones en dispositivos de bajos recursos o con hardware limitado, con la finalidad de realizar aplicaciones para propósitos específicos.

#### **Zerynth**

Es una implementación de software que parte del lenguaje de programación Python, generalmente utilizado para microcontroladores. Como una de sus características permite la combinación de los lenguajes Python y el lenguaje C, para el desarrollo de aplicaciones IoT y cloud computing[33].

#### **Kinetis SDK**

Es un conjunto integrado de software orientado a los microcontroladores Kinetis desarrollados por NXP, este incluye varios elementos necesarios para el desarrollo integral de aplicaciones, entre estos se cuenta con el sistema de inicio y configuración, controladores para el uso de periféricos, middleware y kernels del sistema operativo en tiempo real (RTOS). Este también cuenta con documentación para uso inicial y de su API[33].

### 1.2.5.3. Ventajas y desventajas

A continuación, se presentan las ventajas y desventajas con respecto al uso de Hexiwear.

Ventajas	Desventajas
Portabilidad y comodidad para el usuario.	El diseño puede llegar a ser engorroso para cierto grupo de usuarios en específico.
Sus características lo hacen un dispositivo muy completo para llevar a cabo múltiples tareas.	El hacer uso de varios sensores reduce considerablemente el tiempo de duración de la batería.
El uso de módulos lo convierte en un accesorio con potencial para entornos Iot.	Al ser un dispositivo abierto, no es resistente al agua ni salpicaduras.
Permite monitorizar el estado de salud del usuario.	El sensor de ritmo cardiaco no es completamente preciso por lo que tiene un margen de error considerable.
Versatilidad al aceptar conexiones a dispositivos cercanos o servidores en la nube.	Para un usuario que no tiene conocimiento previo sobre este tipo de dispositivos, su uso completo le representa complejidad.
Facilita la programación y desarrollo de nuevas funciones al ser de código abierto.	Tiene un costo más elevado en comparación con otros Smart Watch tradicionales con funcionalidad limitada pero suficiente para el usuario promedio.
Cuenta con las funciones que un Smart Watch ofrece.	

Tabla 7: Ventajas y desventajas de Hexiwear.

### 1.3. Antecedentes Contextuales

Se han realizado diversos estudios sobre el reconocimiento de gestos con wearables como relojes e incluso anillos inteligentes. Iyer et al.[34] en 2016 realizan un trabajo enfocado en el reconocimiento de gestos en la mano con dispositivos wearable en IoT haciendo uso del reconocimiento de patrones, para ello se utilizaron datos de entrenamiento realizados por 52 personas, cada una de ellas realizaba 5 gestos distintos. Luego del entrenamiento el modelo fue probado con un data set independiente y en diferentes escenarios, dando como resultados un valor de 0.96 de precisión, un recall de 0.92 y un F1-Score de 0.92.

Majid & Tariq [33] en el año 2020 realizaron una aplicación wearable de Internet de las Cosas usando Hexiwear, en este estudio se concluyó que Hexiwear es una solución versátil para uso en implementaciones smart home debido a sus módulos de expansión y su bajo costo comparado con otros dispositivos de características similares. Se probaron aplicaciones de imágenes, de calidad del aire y detección de movimiento, brindando todas ellas resultados satisfactorios.

Abid et al[35]. en 2021, elaboraron un estudio sobre una plataforma wearable basada en sensores para el reconocimiento de gestos en personas con síndrome autista utilizando algoritmos Machine

Learning, para ello se recolectaron los datos del acelerómetro y giroscopio en los ejes x, y, z a una frecuencia de 50Hz. A partir de ello, 24 gestos fueron probados con 10 personas y cada una de ellas repetía el gesto de 7 a 12 veces. Con estas pruebas se compararon cuatro clasificadores, KNN, árbol de decisión, bosques aleatorios y redes neuronales. KNN, bosques aleatorios y redes neuronales presentaron precisiones mayores al 90%.

### **1.3.1. Ámbito de aplicación**

Se desarrolló una aplicación wearable con capacidad de controlar dispositivos smart home a través de gestos, para ello se hace uso del aprendizaje autónomo y otras tecnologías. Con este dispositivo se benefician todas aquellas personas que desean un método de control de dispositivos diferente y más económico que los actuales que están basados en comandos de voz o interfaces gráficas. Además, se ofrece como una alternativa de accesibilidad para aquellas personas que poseen discapacidad visual, auditiva o de lenguaje y que no pueden hacer uso de los medios de control de smart home tradicionales.

### **1.3.2. Establecimiento de requerimientos**

La aplicación wearable es capaz de recolectar los datos de los sensores giroscopio y acelerómetro del dispositivo wearable, estos datos se envían a través de bluetooth al dispositivo móvil para posteriormente ser procesados en la nube. Para el procesamiento de los datos se cuenta con un modelo de aprendizaje autónomo el que decide que gesto es el que está realizando la persona para enviar la orden al dispositivo smart home que tiene configurada dicho gesto para realizar alguna acción.

## 2. CAPITULO II. DESARROLLO DEL PROTOTIPO

### 2.1. Definición del prototipo

Como se describe en la Figura 7, cuando una persona tiene el reloj en su muñeca y realiza un gesto, los datos recolectados por los sensores de acelerómetro y giroscopio son enviados a través de bluetooth al dispositivo móvil que hace la función de gateway, este se encarga de procesar los datos y realizar la predicción con el modelo machine learning, una vez que se obtiene la predicción, esta es enviada al servidor a través de peticiones HTTP, el servidor cuenta con una instancia Node-Red instalada, y está instancia a su vez tiene configurado un nodo con una API de Alexa. Con el dato recibido del gateway, el servidor toma la decisión de a que dispositivo smart home enviar la acción que tiene predefinida el gesto realizado por el usuario completando así su tarea.



Figura 7: Arquitectura de prototipo

El prototipo se desarrolló utilizando una arquitectura de 4 capas que se detallan a continuación:

**Capa dispositivos.** – Esta capa está compuesta por los sensores y dispositivos que recolectaran los datos (sensores del dispositivo hexiwear) y los actuadores y dispositivos smart home como TVs, bocinas, luces, entre otros.

**Capa de servicio.** – En esta capa se establece la comunicación a dispositivos específicos desde el servidor, para ello se utilizó el nodo node-red-contrib-alexa-cakebaked que proporciona una API para el control de dispositivos vinculados al asistente virtual Alexa de Amazon desde una instancia de Node-RED alojada en el servidor.

**Capa de red.** – Se estableció una capa de comunicación, para la cual se usa Bluetooth Low Energy para comunicación entre el dispositivo hexiwear y el gateway para la recepción de datos. Para la comunicación entre el gateway y el servidor en cambio se utilizaron peticiones HTTP.

**Capa de Aplicación.** – Aquí se encuentra tanto el diseño de la aplicación wearable instalado en el dispositivo Hexiwear y la aplicación Android que se encarga de recolectar los datos para su posterior procesamiento con Machine Learnig y así enviar una orden al servidor según el gesto realizado por el usuario.

## **2.2. Metodología de desarrollo del prototipo**

### **2.2.1. Enfoque, alcance y diseño de investigación**

#### **Enfoque de la investigación**

El enfoque fue cuantitativo, debido a que la investigación se basó en la medición de variables cuyos valores son de datos numéricos y los resultados también se apoyen en datos numéricos y estadísticos.

#### **Alcance de la investigación**

El alcance de la investigación al inicio fue exploratorio, puesto que se realizará una recopilación bibliográfica previa y posteriormente pasó a descriptivo, debido a que se estudiaron características del fenómeno, es decir los gestos de la mano del ser humano, pasando a medir variables a través de sensores que posteriormente fueron puestas bajo análisis.

## **Diseño de la investigación**

El diseño de la investigación fue cuasi-experimental, porque el sujeto de estudio no se eligió de manera aleatoria, si no se eligió previo al inicio de la investigación, que en este caso el sujeto fueron los gestos que una persona puede realizar con su mano y la interacción con dispositivos smart home.

### **2.2.2. Unidades de análisis**

#### **Población (universo)**

La población utilizada en el presente estudio fueron los datos arrojados por el sensor acelerómetro del dispositivo wearable. El 70% de los datos serán tomados como datos de entrenamiento y el 30% restante serán datos de prueba.

#### **Muestra**

No se obtendrá una muestra para el presente trabajo.

### **2.2.3. Técnicas e instrumentos de recopilación de datos**

<b>Técnicas</b>	<b>Instrumento</b>
Observación	Guía de observación y análisis de control para las pruebas del prototipo (ver Anexo 2)
Análisis de datos	Guía de análisis de datos (ver Anexo 3)

**Tabla 8:** Técnicas e instrumentos de recopilación de datos.

### **2.2.4. Técnicas de procesamiento de datos para la obtención de resultados**

Tras la revisión bibliográfica de trabajos similares se concluyó que la técnica de procesamiento de datos más adecuada para llevar a cabo la investigación fue Machine Learning con aprendizaje supervisado, específicamente algoritmos de clasificación, todo esto debido a la naturaleza de la investigación y los resultados a los que se pretendía llegar.

### **2.2.5. Metodología o métodos específicos**

La metodología que se utilizó es la IoT Methodology conocida también como IoTM. Esta metodología consta de 3 fases principales y cada una contiene 3 subfases. En la siguiente tabla se enumeran las fases y sub-fases de la metodología IoTM. Cada uno de los elementos de la metodología se encuentran con más detalles en la fundamentación teórica del presente trabajo.

Fase	Sub-fases
Brainstorm	Co-Crear
	Idear
	Validar
Build	Arquitectura
	Implementar
	Desplegar
Tune	Identificar
	Clasificar
	Actuar y ajustar

**Tabla 9:** Metodología o métodos específicos.

### 2.2.6. Herramientas y/o Materiales

Después de realizar investigaciones en diferentes fuentes bibliográficas se procedió a definir las herramientas y materiales que se utilizarán a lo largo del desarrollo del prototipo, además se realizó una clasificación por categorías como se muestra en la Tabla 10.

Categoría	Herramientas y/o materiales
Software	<ul style="list-style-type: none"> <li>• Repositorio WolkSense Hexiwear</li> <li>• Kinetis SDK</li> <li>• Node-RED</li> <li>• Visual Studio Code</li> </ul>
Hardware	<ul style="list-style-type: none"> <li>• Kit de desarrollo Hexiwear</li> <li>• Dispositivos smart home, por ejemplo: focos, interruptores, tomacorrientes, entre otros.</li> </ul>
Lenguajes	<ul style="list-style-type: none"> <li>• Python</li> <li>• C++</li> <li>• Java</li> </ul>
Datos	<ul style="list-style-type: none"> <li>• Dataset de gestos</li> </ul>
Algoritmos	<ul style="list-style-type: none"> <li>• Algoritmos de Clasificación, por ejemplo, KNN, árbol de decisión, random forest, entre otros.</li> </ul>

**Tabla 10:** Herramientas y/o Materiales.

### 2.3. Desarrollo del prototipo

El desarrollo del prototipo se realizó utilizando la metodología IoT Methodology, con sus tres fases principales que son Brainstorm, Build y Tune, las dos primeras se desarrollan durante este segundo capítulo mientras que la tercera se desarrolla en el tercer capítulo. En la Figura 9 se observa la metodología con sus respectivas fases y subfases.



**Figura 8:** IoT Methodology. Fuente: [15]

### 2.3.1. Brainstorm

#### Co-Crear

En esta primera etapa se desarrollaron algunas actividades con la finalidad de recolectar información que ayude a describir el funcionamiento y finalidad del prototipo, así como las personas que estarán inmersas en el proceso de desarrollo del mismo. En la tabla 11 se describe las personas responsables del desarrollo del prototipo.

Nombre	Rol
Ing. Dixys Hernández	Supervisor
Jean Pierre Apolo	Desarrollador
Edisson Carrillo	Desarrollador

**Tabla 11:** Roles del equipo de trabajo

Otro apartado importante dentro de esta etapa fue la de establecer un listado de requisitos como se muestra en la tabla 12 con la finalidad de tener una idea clara de lo que se debe cumplir al desarrollar el prototipo.

Capa	Requisitos funcionales
Aplicación	<ul style="list-style-type: none"> <li>-Permitir la conexión del wearable.</li> <li>-Receptar los datos del giroscopio.</li> <li>-Predecir en tiempo real los gestos.</li> <li>-Enviar el resultado de la predicción al servidor.</li> <li>-Desarrollar un firmware para la recolección y envío de los datos del acelerómetro de Hexiwear.</li> </ul>
Servicio	<ul style="list-style-type: none"> <li>-Crear una máquina virtual con IP pública.</li> <li>-Instalar node red en la máquina virtual.</li> <li>-Crear un servidor HTTP en Node Red.</li> <li>-Instalar Alexa Cakebaked.</li> <li>-Iniciar sesión en Alexa Cakebaked con una cuenta Amazon Alexa.</li> <li>-Recibir los resultados de las predicciones.</li> <li>-Enviar órdenes a los dispositivos vinculados a la cuenta de Alexa.</li> </ul>
Red	<ul style="list-style-type: none"> <li>-Lectura de datos de los sensores de Hexiwear a través de puerto serie en ámbito de pruebas y por Bluetooth Low Energy para prototipo final.</li> <li>-Enviar peticiones GET desde la capa de aplicación al servidor con la acción a activar.</li> </ul>
Dispositivos	<ul style="list-style-type: none"> <li>-Configurar el Wearable Hexiwear.</li> <li>-Adquirir dispositivos compatibles con Alexa.</li> <li>-Vincular los dispositivos a la misma cuenta vinculada a Alexa Cakebaked.</li> <li>-Configurar los dispositivos en Node Red.</li> </ul>

**Tabla 12:** Requisitos funcionales por capa para el desarrollo del prototipo

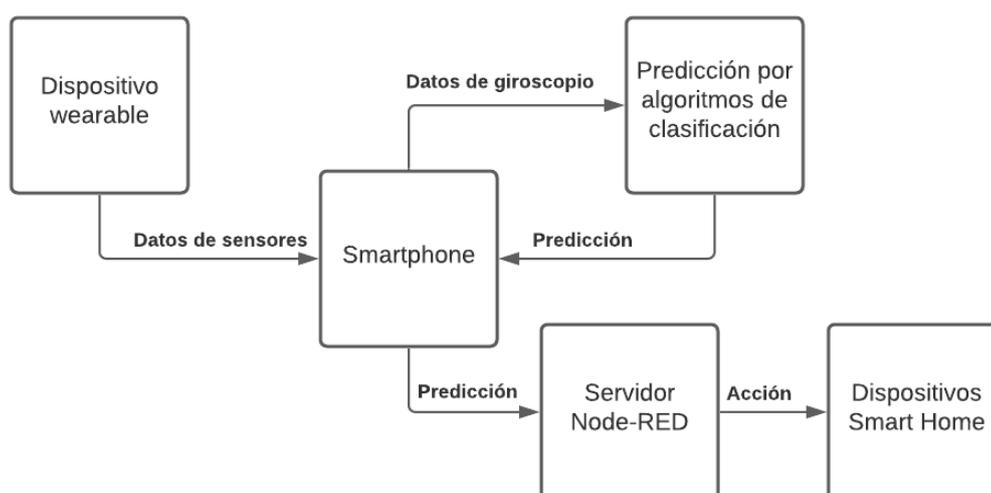
Para la definición de requisitos no funcionales, se utilizó el Modelo de Calidad para la Evaluación de Aplicaciones IoT propuesto por Kim [36], este modelo define cuatro características principales que son Funcionalidad, con idoneidad, exactitud, interoperabilidad, seguridad como subcaracterísticas; fiabilidad con madurez, tolerancia a fallos, recuperabilidad, conectividad, movilidad como subcaracterísticas; Eficiencia con eficiencia de recursos y eficiencia de tiempo como subcaracterísticas; Portabilidad con inestabilidad, conformidad, sustentabilidad y reemplazabilidad como subcaracterísticas.

Característica	Subcaracterística	Requisito no funcional
Funcionalidad	Exactitud	El sensor de Aceleración debe brindar la información en unidad de aceleración (m/s <sup>2</sup> ).
	Interoperabilidad	Las acciones y gestos deben ejecutarse en el orden en que fueron llamados.
	Seguridad	Los datos no se deben almacenar en la nube, para ello la predicción se realiza en la capa de aplicación.
Fiabilidad	Madurez	El prototipo debe mostrar la menor cantidad de fallos en su etapa final.
	Tolerancia a Fallos	El prototipo debe ser capaz de detectar fallos y seguir operando una vez afrontados.
	Recuperabilidad	El servidor debe tener capacidad de reiniciar los servicios en caso de detectar un fallo grave para corregirlos.
	Conectividad	Se debe usar tecnologías de conexión rápidas para el envío y recepción de datos en las diferentes capas.
	Movilidad	El dispositivo Wearable puede ser usado con normalidad por el usuario sin que se genere la detección de gestos falsos además de utilizar baterías.

Característica	Subcaracterística	Requisito no funcional
Eficiencia	Eficiencia de recursos	El firmware del wearable debe ser capaz de usar los recursos disponibles de una forma eficiente sin afectar el rendimiento.
	Eficiencia de tiempo	Desde la realización del gesto hasta la generación de la acción en el dispositivo Smart Home debe haber un tiempo menor a 3 segundos.
Portabilidad	Conformidad	La aplicación wearable debe hacer uso de tecnologías comunes para el desarrollo de aplicaciones wearable a nivel internacional.
	Sustentabilidad	El uso de la batería debes ser eficiente aprovechando las ventajas de la conexión Bluetooth Low Energy y evitando sobrecargas de tareas innecesarias, brindando una autonomía aceptable.
	Reemplazabilidad	Los dispositivos Smart Home pueden ser reemplazados por otros de características similares sin afectar el funcionamiento de la aplicación wearable.

**Tabla 13:** Requisitos no funcionales

Como último apartado de la fase de Co-Creación, se desarrolló el mapa de contexto del prototipo con la ayuda de los requisitos establecidos previamente. Este mapa de contextos que se muestra en la Figura 8, ayuda a comprender de forma visual el contexto y funcionamiento del prototipo, el cual recolecta los datos del giroscopio y acelerómetro del dispositivo wearable para enviarlos a través de bluetooth hacia un smartphone que cuenta con la aplicación que contiene el machine learnig para realizar la predicción, este a su vez envía el resultado hacia el servidor que contiene la instancia de Node-RED encargada de enviar las diferentes órdenes a los dispositivos smart home.



**Figura 9:** Mapa de contexto del prototipo

## Idear

Para esta etapa se crearon los primeros bosquejos del prototipo, para ello se definió que serían 4 los dispositivos que se iban a controlar, siendo estos un tomacorriente inteligente, un dispositivo Amazon Echo Dot, una bombilla inteligente y una televisión, tal como se muestra en la Figura.

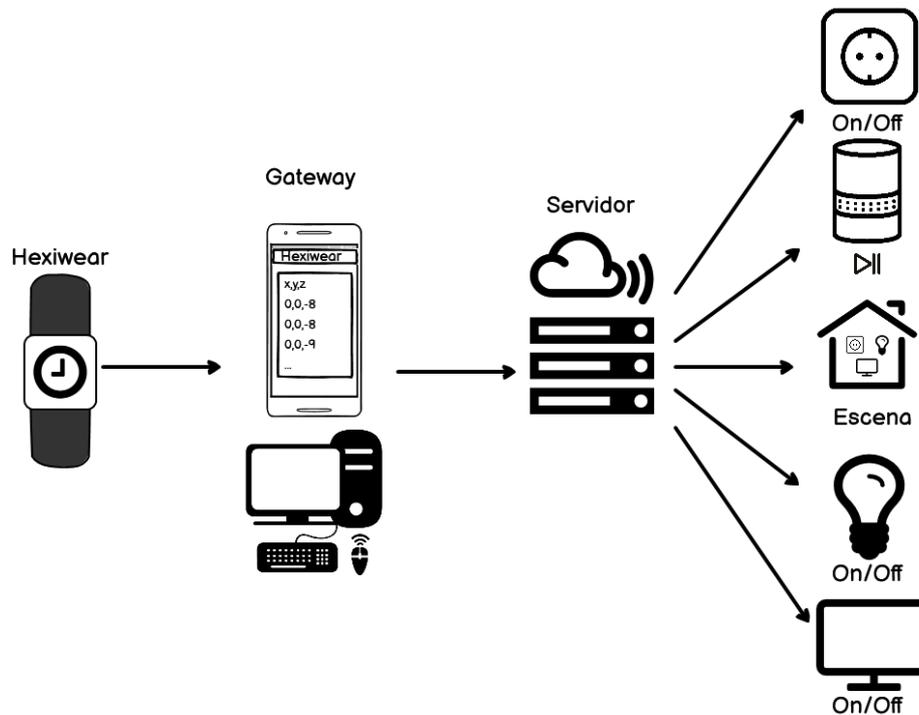


Figura 10: Flujo de funcionamiento del Prototipo

Al haber obtenido el bosquejo final se consideró definir cuáles son los gestos que se usarán para controlar cada uno de los dispositivos. En la Tabla 14 se detalla de manera ilustrativa cada uno de los gestos con los dispositivos smart home que controlarán.

Acción	Encender / Apagar	Encender / Apagar	Activar escena
Dispositivo	 Bombilla Sengled (Bth)	 TCL Android TV	 Todos los dispositivos
Gesto			

Acción	Reproducir / Pausar	Encender / Apagar
Dispositivo	 Echo Dot (Gen 3)	 Smart Plug Govee
Gesto		

**Tabla 14:** Descripción de gestos y acciones de los dispositivos smart home

## Validar

El proceso de validación se realizó a través de una reunión entre los miembros del equipo de trabajo, entre los cuales se revisó los bocetos e instrumentos elaborados durante esta primera etapa. Esta etapa culminó con la información descrita en esta primera fase de la metodología IoT Methodology y sirvió de base para el desarrollo de las siguientes etapas del proyecto.

### 2.3.2. Build

#### Arquitectura

La arquitectura que se escogió fue el modelo de referencia IoT de 4 capas, compuestas por la capa de aplicación, capa de servicio, capa de red y capa de dispositivos, en la Tabla 15 se enlistan los elementos que compondrán cada una de las capas.

Capa	Elementos
Aplicación	<ul style="list-style-type: none"> <li>• Nube de Alexa</li> </ul>
Servicio	<ul style="list-style-type: none"> <li>• Oracle Cloud</li> <li>• Node-Red</li> </ul>
Red	<ul style="list-style-type: none"> <li>• Bluetooth</li> <li>• Peticiones HTTP</li> <li>• WiFi</li> <li>• Smartphone</li> <li>• Echo Dot</li> </ul>
Dispositivos	<ul style="list-style-type: none"> <li>• Reloj Hexiwear</li> <li>• Smart TV</li> <li>• Smart Plug</li> <li>• Foco Inteligente</li> </ul>

**Tabla 15:** Elementos por capa de la arquitectura

## Implementar

En esta subfase se realiza la implementación de cada uno de los elementos del prototipo, para ello se ha dispuesto dividir en tres secciones: implementación del firmware del wearable Hexiwear, creación de los datasets, implementación y entrenamiento del aprendizaje automático e Implementación de instancia de Node-RED.

### Implementación del firmware del wearable

Para la implementación del firmware se desarrolló código en C++ haciendo uso del IDE y librerías de Hexiwear en MBED Compiler. El código como se muestra en la Figura 11, envía los datos del acelerómetro del dispositivo a través del puerto serie, pero previo a esto los valores que son dados en gravedades (g), se multiplican por el valor de la gravedad terrestre ( $9.80665\text{m/s}^2$  aproximadamente) para obtener el valor de la aceleración en x, y, z.

```
4
5 // Método principal
6 int main() {
7
8     accel.accel_config();
9     while (true) {
10        //Obtención y envío de valores x, y, z por el puerto serie
11        accel.acquire_accel_data_g(accel_data);
12        pc.printf("%f,%f,%f\r\n",accel_data[0]*9.80665,accel_data[1]*9.80665,accel_data[2]*9.80665);
13        led1 = !led1;
14        //Usar para modificar la frecuencia de envío de datos
15        //Thread::wait(100);
16    }
17 }
18
```

Figura 11: Implementación del firmware del wearable

### Creación de los datasets

Se armó un data set por cada uno de los gestos que se realizaron, para ello se hizo uso de código en Python para recolectar solo la parte entera de los valores de x, y, z en un numpy array que posteriormente es almacenado en un archivo CSV. El código que se muestra en la Figura 12 es el encargado de recolectar los datos de x, y, z. Cada gesto se realizó varias veces hasta completar el tamaño del dataset, el cual se define multiplicando los Hertz a los que se envían los datos por el puerto serie y el tiempo estimado en el que se repetirá cada gesto.

```

5 def captureSerialData():
6     sample_rate = 100 #hertz
7     time = 10 #segundos
8     sampleSize = sample_rate * time #tamaño de la muestra
9
10    port = 'COM3' #Declaración del puerto serie
11    acc = np.zeros((sampleSize, 3), dtype = float) #Inicialización de array para almacenar datos
12    hexi = serial.Serial(port,9600) #Apertura dell puerto sserie con su Baud rate
13    hexi.flushInput()
14    print('Empezando captura')
15    try:
16        for i in range(sampleSize):
17            data = hexi.readline() #lectura de los valores del puerto serie
18            Data = data[:-2].split(b',') #separación de datos
19            print(Data)
20            acc[i,0] = float(Data[0])
21            acc[i,1] = float(Data[1])
22            acc[i,2] = float(Data[2])
23            print(sampleSize - i)
24        return acc
25    except Exception as e:
26        print(e)
27        print("Error al recolectar datos")
28        return acc
29
30
31 if __name__ == '__main__':
32     accData = captureSerialData() #grafico de datos recolectados
33     plt.plot(accData)
34     np.savetxt('right_down.csv', accData, delimiter=',', fmt='%d') #guardado de datos en CSV
35     plt.show()

```

Figura 1212: Algoritmo para la creación de datasets

## Implementación y entrenamiento de aprendizaje automático

Para la implementación del algoritmo se dividió en tres funciones principales, la primera de ellas extraer\_caracteristicas, que requiere como parámetros la ruta del archivo que contiene cada dataset, el nombre del gesto y el valor de clasificación que se le desea asignar. Lo primero que realiza el algoritmo es obtener los picos en un eje determinado según el gesto que se desea detectar, cada uno de estos se almacena en un vector para posteriormente determinar la curtosis, varianza y simetría. Estas tres medidas ayudan a obtener valores de referencia en la distribución de los datos en un rango que se determina por cada pico, restando -30 y sumando +45 a cada uno de los picos, -30 y +45 se debe ajustar para obtener mejores resultados. En la Figura 13, se puede observar con más detalle el algoritmo utilizado.

```

#Para cada pico se extraen las características, tomando en cuenta
#30 valores anteriores al pico
#y 45 valores posteriores al pico
for peak in peaks:
    frame = data[peak-30:peak+45, :]
    frame_var = np.var(frame, axis = 0)
    frame_skew = stat.skew(frame, axis = 0)
    frame_kurt = stat.kurtosis(frame, axis = 0)

    caracteristicas = np.hstack((frame_var, frame_skew, frame_kurt))
    if np.size(total_caracteristicas) == 0:
        total_caracteristicas = caracteristicas
    else:
        total_caracteristicas = np.vstack((total_caracteristicas, caracteristicas))
y = np.ones(len(peaks)) * clasificacion
return total_caracteristicas, y

```

Figura 1313: Algoritmo para la extracción de características de los datasets.

En la Figura 14 se puede visualizar los picos de cada gesto, en donde para “Y”, en el gesto hacia la derecha los picos tienen a ser negativos, y en el gesto hacia la izquierda los picos tienen a ser positivos. Para “X” en cambio, los picos tienden a ser positivos en el gesto hacia arriba y negativos en el gesto hacia abajo. Par el gesto hacia adelante se toma como referencia el eje “Z” y tiende a ser negativo.

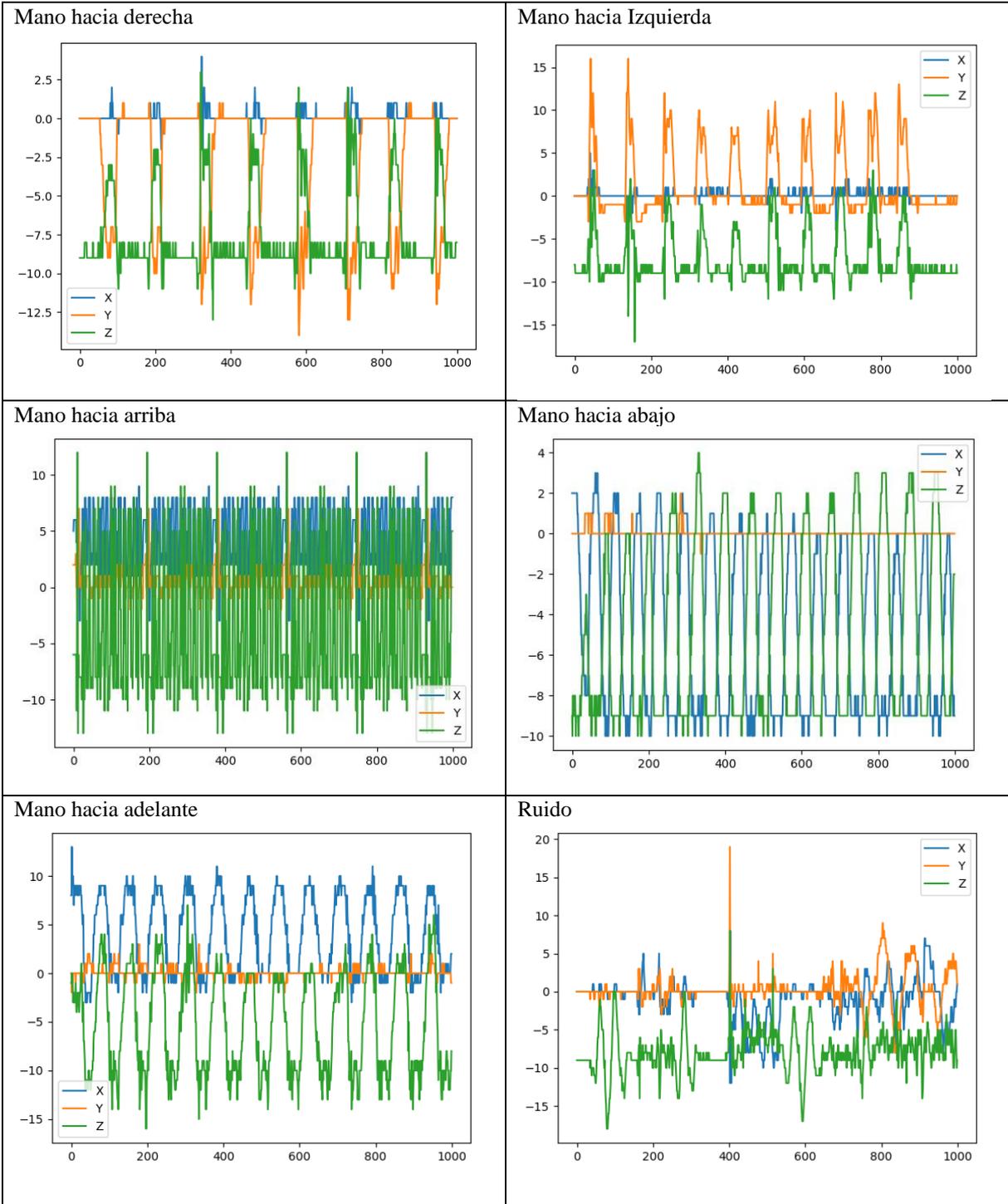


Figura 1414: Visualización de picos para cada gesto.

La segunda función es `extraer_ruido` que se muestra en la Figura 15. Esta se encarga de igual forma extraer la curtosis, simetría y varianza, pero de un dataset que contiene gestos en falso, también denominado ruido, para mejorar el reconocimiento de los gestos.

```
def extraer_ruido():
    total_caracteristicas = []
    tamano = len(ruido[:,2])
    samples_done = 0
    for i in range(40, tamano - 45, 25):
        samples_done = samples_done + 1
        frame = ruido[i - 35: i+45, :]
        frame_var = np.var(frame, axis = 0)
        frame_skew = stat.skew(frame, axis = 0)
        frame_kurt = stat.kurtosis(frame, axis = 0)

        caracteristicas = np.hstack((frame_var, frame_skew, frame_kurt))
        if np.size(total_caracteristicas) == 0:
            total_caracteristicas = caracteristicas
        else:
            total_caracteristicas = np.vstack((total_caracteristicas, caracteristicas))
    y = np.zeros(samples_done)
    return total_caracteristicas, y
```

Figura 1515: Algoritmo para la extracción de características del ruido.

La tercera función, `entrenar_modelo`, ilustrada en la Figura 16, requiere de un conjunto “x” para entrenamiento y un conjunto “y” para pruebas, el algoritmo de aprendizaje supervisado que utiliza es el Clasificador de bosque aleatorio. Esta función retorna el modelo que será utilizado para realizar el entrenamiento.

```
def entrenar_modelo(X, y):
    modelo = RandomForestClassifier(n_estimators=20)
    cross_val_scores = cross_val_score(modelo, X, y, cv = 6)
    print('Cross scores: ', cross_val_scores)
    modelo.fit(X, y)
    return modelo
```

Figura 1616: Algoritmo para la creación del modelo ML

El método principal, `train`, que se muestra en la Figura 17, se encarga de llamar a las funciones que se describieron anteriormente. Primero se llama a `extraer_caracteristicas` para guardar las características de cada uno de los dataset y posteriormente se guardan en variables para entrenamiento y para prueba. Una vez se pre procesa los datos, se llama a la función `entrenar_modelo` para realizar el entrenamiento, enviando los conjuntos de datos de “x” y “y”, por último, se realiza una validación de modelo utilizando la matriz de confusión.

```

caracteristicas_izquierda, y_izquierda = extraer_caracteristicas('Datasets/izquierda.csv', 'izquierda', 1)
caracteristicas_derecha, y_derecha = extraer_caracteristicas('Datasets/derecha.csv', 'derecha', 2)
caracteristicas_abajo, y_abajo = extraer_caracteristicas('Datasets/abajo.csv', 'abajo', 3)
caracteristicas_arriba, y_arriba = extraer_caracteristicas('Datasets/arriba.csv', 'arriba', 4)
caracteristicas_adelante, y_adelante = extraer_caracteristicas('Datasets/adelante.csv', 'adelante', 5)
caracteristicas_ruido, y_ruido = extraer_ruido()

#Combinación de los datasets
y = np.vstack(( y_izquierda.reshape(-1,1) , y_derecha.reshape(-1, 1) , y_arriba.reshape(-1, 1),
y_abajo.reshape(-1, 1), y_adelante.reshape(-1, 1), y_ruido.reshape(-1, 1)))
X = np.vstack((caracteristicas_izquierda, caracteristicas_derecha,
caracteristicas_arriba, caracteristicas_abajo, caracteristicas_adelante, caracteristicas_ruido))

print('Pre-procesamiento: obteniendo caracteristicas de entrenamiento...')
data = np.hstack((X, y))
np.random.shuffle(data)
y = data[:, -1]
X = data[:, :-1]

#Entrenamiento del modelo
print('Entrenando modelo...')
modelo_entrenado = entrenar_modelo(X,y)

```

Figura 1717: Método principal del entrenamiento

## Implementación de servidor HTTP en Node-RED

En una máquina virtual alojada en Oracle Cloud Free Tier, se alojó el servidor HTTP como se muestra en la Figura 18, la cual se encarga de recibir las peticiones GET que se generan desde Python (ver Recolección de datos y conexión hacia el servidor).

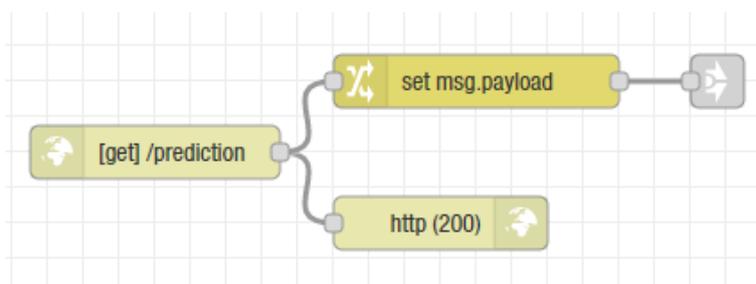


Figura 18: Servidor HTTP.

Para la realización de las acciones Smart Home, se agregó el nodo node-red-contrib-alexa-cakebaked, con el cual se puede vincular una cuenta de Amazon Alexa y controlar todos los dispositivos que pertenezcan a dicha cuenta. En la Figura 19 se muestran los nodos de inicialización de Alexa.

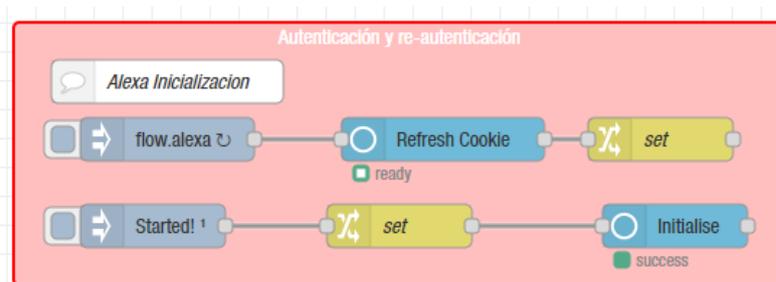


Figura 19: Autenticación y re-autenticación de Alexa.

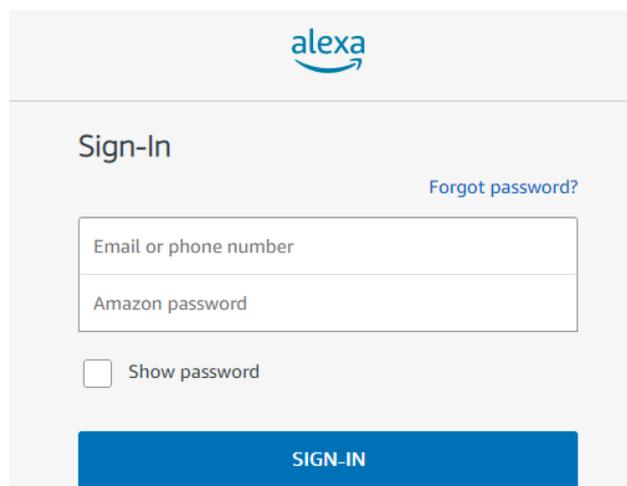
Para el inicio de sesión se debe configurar una nueva cuenta en el nodo Initialise, la configuración resultante se muestra en la Figura 20, en donde el método de autenticación se realiza por proxy, el único soportado a día de hoy, seguido se escribe la IP del servidor y el puerto 3456, que servirá para abrir la ventana de autenticación.



Auth Method	Proxy
This IP	144.22.34.220
Port	3456
File Path	Optional
Refresh	3 days

**Figura 20:** Configuración de nueva cuenta de Alexa.

Luego de pulsar el botón Deploy, se pedirá ingresar a la ip junto con el puerto seleccionado para el inicio de sesión tal como se muestra en la Figura 21. Una vez realizada esta acción se podrá hacer uso de la paleta de nodos de Alexa con todos los dispositivos vinculados a la cuenta en la que se inició sesión.



alexia

Sign-In [Forgot password?](#)

Email or phone number

Amazon password

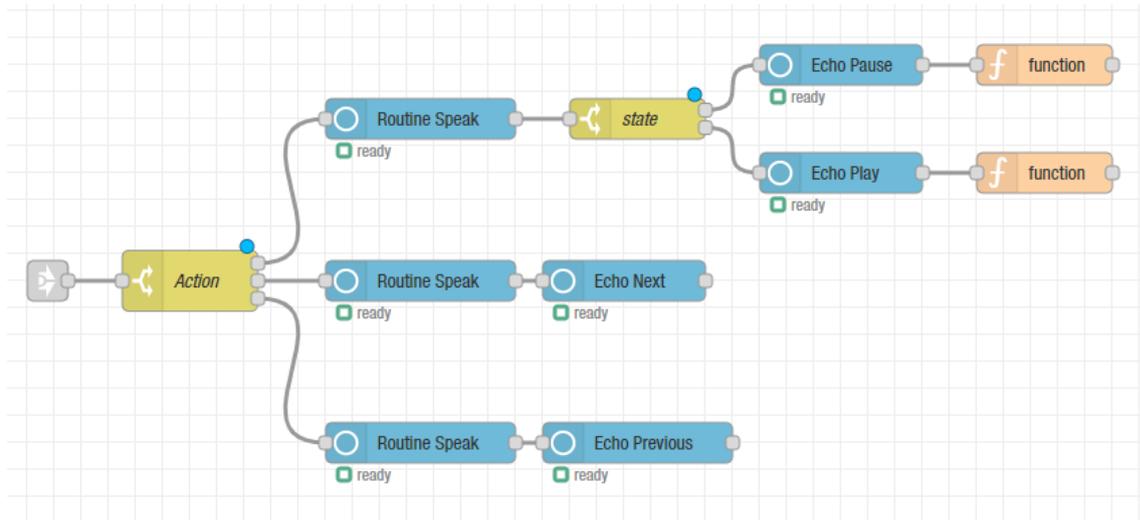
Show password

SIGN-IN

**Figura 21:** Ventana de inicio de sesión de Alexa.

En la Figura 22 se observa el flujo utilizado para controlar el dispositivo Amazon Echo Dot, en específico para la reproducción de música. El nodo Action se encarga de enviar la señal al nodo correspondiente según la acción recibida por el servidor HTTP. Los nodos Routine Speak son nodos de retroalimentación Text-to-Speech que nombran el gesto realizado. El nodo State cambia el estado entre play/pause dependiendo del estado actual. El nodo Echo Pause y Echo Play, pausan

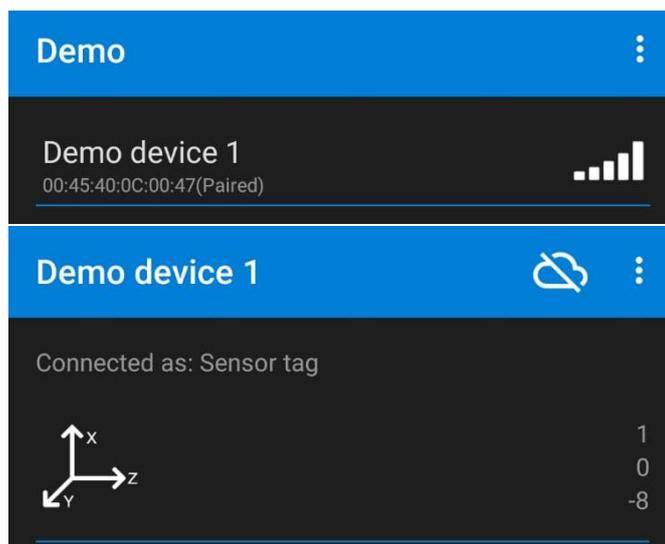
y reproducen música correspondientemente. El nodo Echo Next y Echo Previous, adelanta o retrocede una canción correspondientemente.



**Figura 22:** Flujo para el control del dispositivo Amazon Echo Dot.

## Recolección de datos y envío de orden al servidor

La recolección de datos se realiza en un smartphone haciendo uso de la aplicación oficial de Hexiwear a la cual se le realizaron ciertos cambios previamente. En la Figura 23 se observa cómo es detectado el reloj dentro de la aplicación y como son obtenidos los datos que posteriormente son analizados internamente con el algoritmo de predicción descrito anteriormente en este documento.



**Figura 23:** Recepción de datos en aplicación Hexiwear.

Como se muestra en la Figura 24, la conexión hacia el servidor se realiza a través de peticiones GET que se envían desde un método de la aplicación móvil de Hexiwer. Este método recibe como parámetro la orden resultante de la predicción obtenida tras el análisis de los datos del acelerómetro que se envían por bluetooth desde el wearable hexiwear hasta un smartphone.

```
private void server(String action){
    RequestQueue queue = Volley.newRequestQueue(this);
    String url ="http://129.151.107.75:1880/prediction?value="+action;

    StringRequest stringRequest = new StringRequest(Request.Method.GET, url,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
            }
        });
    queue.add(stringRequest);
}
```

Figura 24: Método para la recolección de datos.

## Desplegar

La sub fase de Despliegue IoT Methodology se explica a detallan en el apartado de ejecución del prototipo del presente trabajo de titulación.

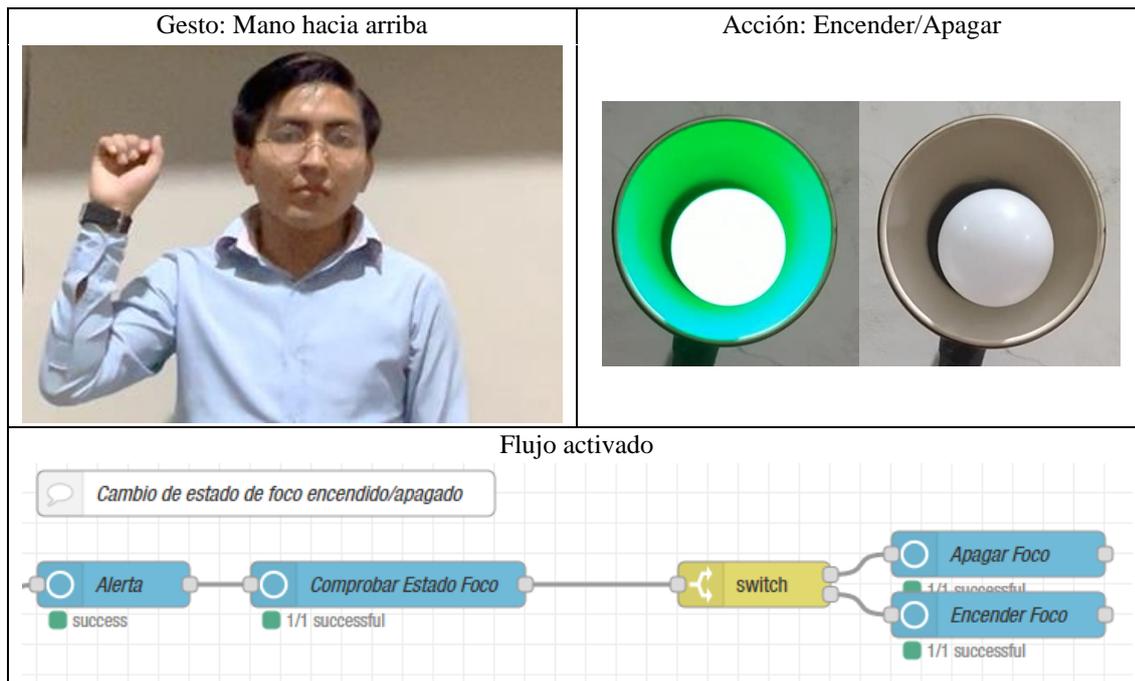
### 2.3.3. Tune

La fase Tune de IoT Methodology, junto a las sub-fases Identificar, Clasificar, Actuar y ajustar, se describen a detalle en el Capítulo III. Evaluación del Prototipo, del presente trabajo de titulación.

## 2.4.Ejecución del prototipo

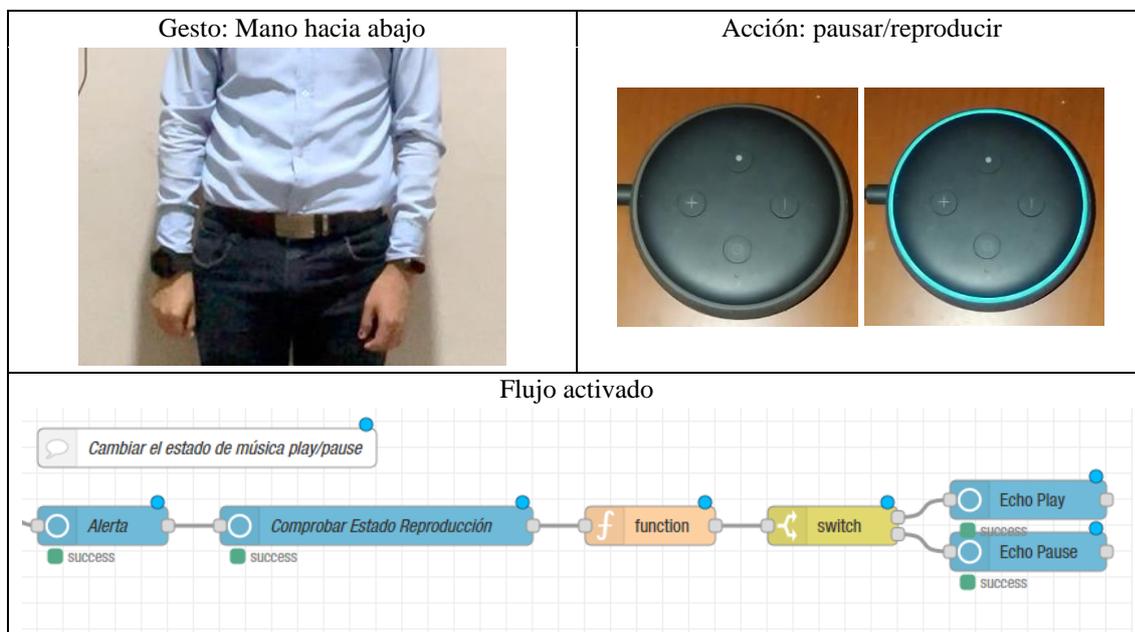
Para la ejecución del prototipo se probaron cada uno de los gestos una vez terminada su implementación, para lo cual se conectó el wearable Hexiwear a un smartphone a través de bluetooth y se configuró dentro de la aplicación móvil de Hexiwear para la recolección de datos y el envío de las órdenes al servidor.

El gesto “Mano hacia arriba” (Figura 25), Se utiliza como método de apertura para todos los gestos, pero a su vez sirve para cambiar el estado de un foco entre encendido y apagado. Cuando se encuentra en modo detección el dispositivo Echo Dot emitirá un mensaje TTS con la frase “Esperando gesto”, si se mantiene el mismo gesto por más de tres segundos, el mensaje TTS cambiará a “Gesto hacia arriba detectado” y en Node-Red se activará el flujo que enciende/apaga el foco.



**Figura 25:** Funcionamiento del gesto “Mano hacia arriba”.

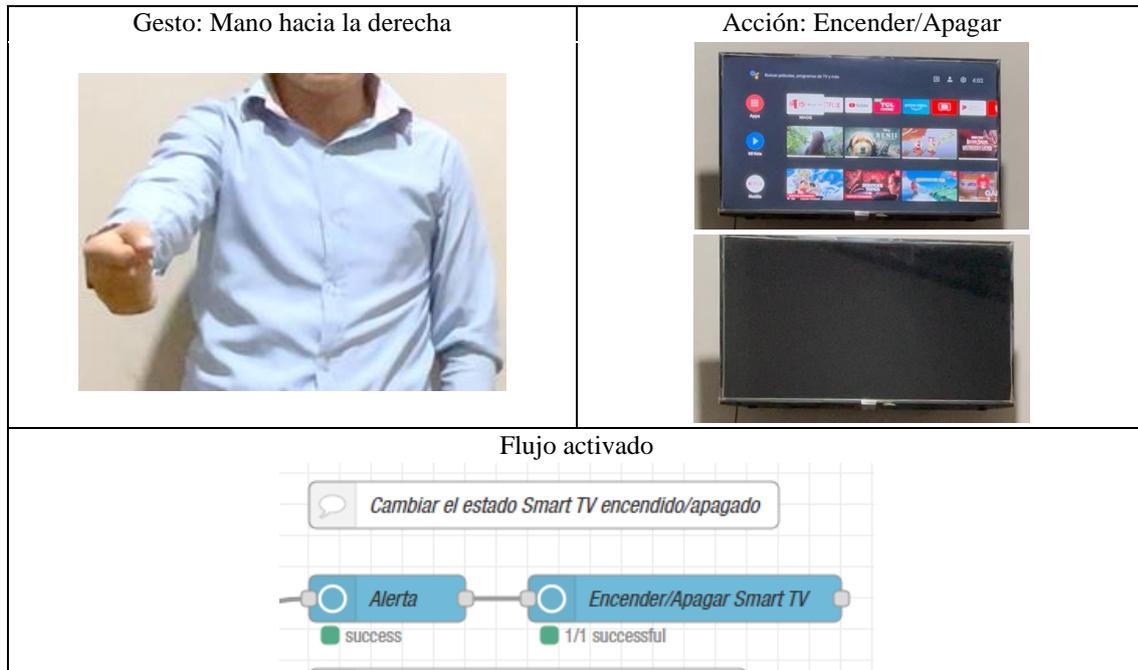
El gesto “Mano hacia abajo” (Figura 26), es detectado cuando la mano en la que se encuentra el dispositivo se desplaza hacia abajo posterior al gesto de activación, la acción se envía al servidor y activa el TTS para decir “Gesto hacia abajo detectado”, como paso siguiente activa el flujo de Node-Red que reproduce/la música en Echo Dot



**Figura 26:** Funcionamiento del gesto “Mano hacia abajo”.

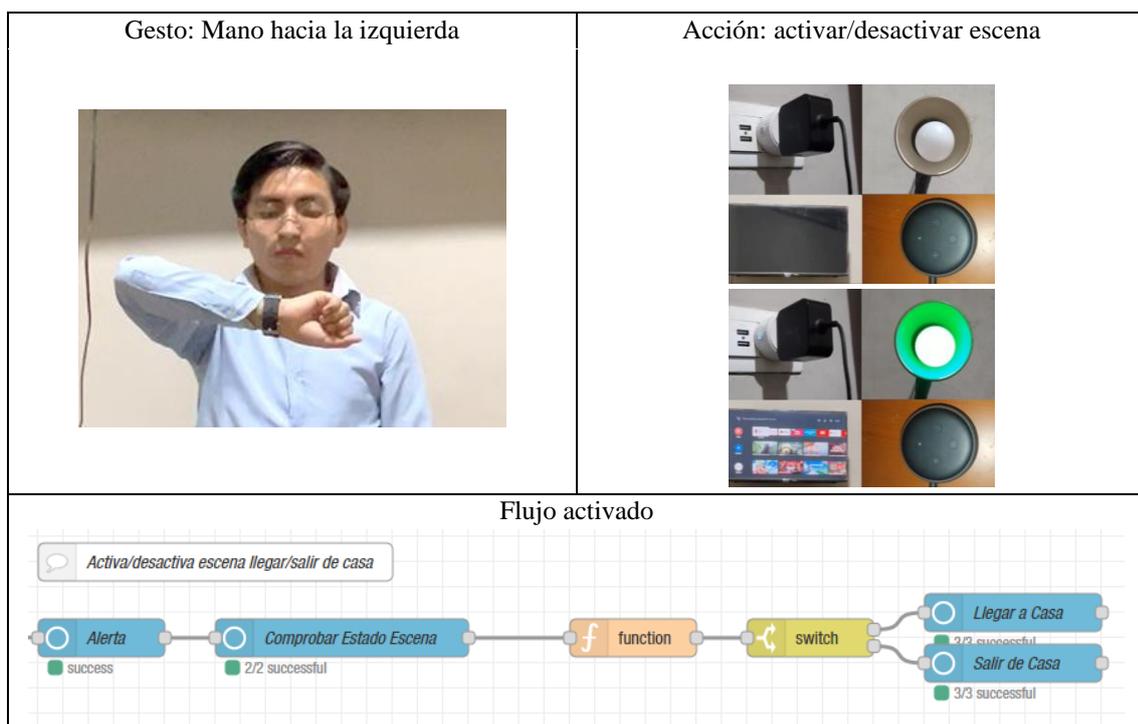
El gesto “Mano hacia la derecha” (Figura 27), es detectado cuando la mano en la que se encuentra el dispositivo se desplaza hacia el centro y se gira hacia la derecha posterior al gesto de activación,

la acción se envía al servidor y activa el TTS para decir “Gesto hacia la derecha detectado”, como paso siguiente activa el flujo de Node-Red que enciende/apaga la smart TV.



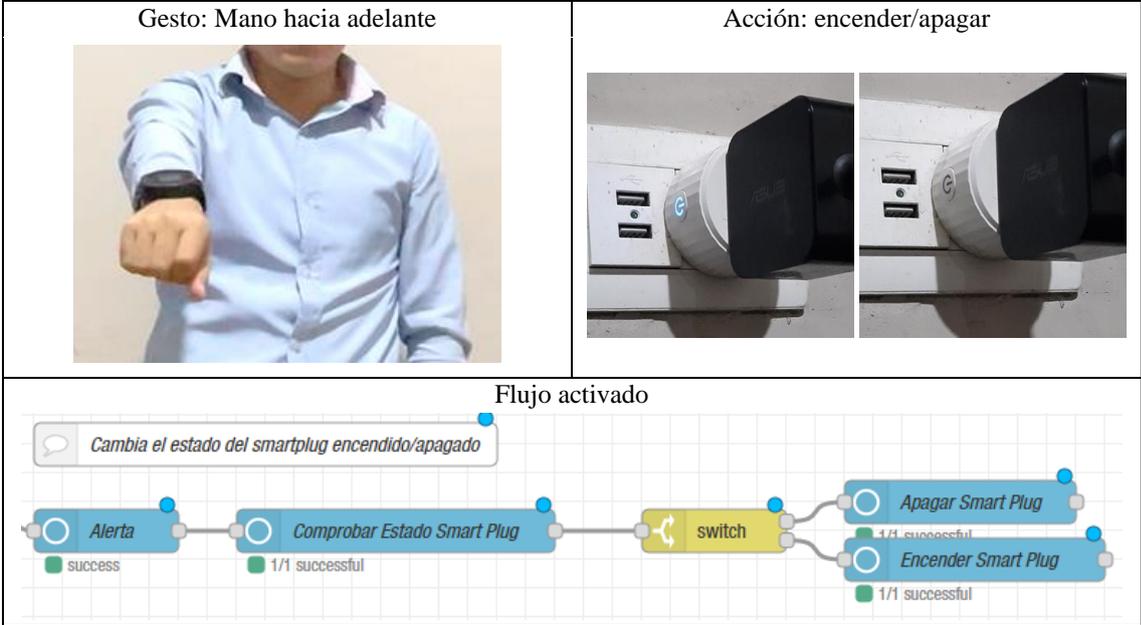
**Figura 27:** Funcionamiento del gesto “Mano hacia la derecha”.

El gesto “Mano hacia la izquierda” (Figura 28), es detectado cuando la mano en la que se encuentra el dispositivo se desplaza hacia el centro y se gira hacia la izquierda posterior al gesto de activación, la acción se envía al servidor y activa el TTS para decir “Gesto hacia la izquierda detectado”, como paso siguiente activa el flujo de Node-Red que activa o desactiva una escena.



**Figura 28:** Funcionamiento del gesto “Mano hacia la izquierda”.

El gesto “Mano hacia adelante” (Figura 29), es detectado cuando la mano en la que se encuentra el dispositivo se desplaza hacia al frente posterior al gesto de activación, la acción se envía al servidor y activa el TTS para decir “Gesto hacia adelante detectado”, como paso siguiente activa el flujo de Node-Red que enciende/apaga el smart plug.



**Figura 29:** Funcionamiento del gesto “Mano hacia adelante”.

### 3. CAPITULO III. EVALUACIÓN DEL PROTOTIPO

#### 3.1. Plan de evaluación del prototipo

Previo a la evaluación del prototipo, se ha desarrollado un plan de evaluación de prototipo, el cual está dividido en tres apartados principales, el primero de ellos es la planificación donde se describe el tipo de prueba a realizar, el objetivo de las mismas y la métrica que se usa. El segundo apartado son los Requerimientos, estos pueden ser de Hardware, Software y Físicos. El tercer apartado son los Diagramas o estructura de los escenarios, en donde se describe a través de esquemas el funcionamiento de cada escenario que se prevé utilizar para las pruebas.

##### 3.1.1. Pruebas

La planificación consta de los tipos de pruebas que se plantear realizar, estas pruebas fueron enfocadas principalmente en el rendimiento, latencia y funcionalidad. Para cada una de las pruebas se definió un objetivo y las métricas que se esperaron obtener tras la ejecución del prototipo.

#	Tipo de prueba	Objetivo	Métrica
1	Rendimiento de algoritmo KNN	Medir el rendimiento del algoritmo KNN a través de métricas para su uso en el reconocimiento de gestos.	F1-Score, precision, cross-validation, recall
2	Rendimiento de algoritmo Decision Tree	Medir el rendimiento del algoritmo Decision Tree a través de métricas para su uso en el reconocimiento de gestos.	F1-Score, precision, cross-validation, recall
3	Rendimiento de algoritmo Random Forest	Medir el rendimiento del algoritmo Random Forest a través de métricas para su uso en el reconocimiento de gestos.	F1-Score, precision, cross-validation, recall
4	Latencia en la acción de encender	Obtener el tiempo promedio de la ejecución de la acción de encender, a través de la medición del tiempo existente entre la realización de un gesto y el encendido del dispositivo, para su posterior evaluación y mejoramiento.	Tiempo
5	Latencia en la acción de apagar	Obtener el tiempo promedio de la ejecución de la acción de apagar, a través de la medición del tiempo existente entre la realización de un gesto y el apagado del dispositivo, para su posterior evaluación y mejoramiento.	Tiempo
6	Latencia en la acción de reproducir	Obtener el tiempo promedio de la ejecución de la acción de reproducir, a través de la medición del tiempo existente entre la realización de un gesto y la reproducción de música, para su posterior evaluación y mejoramiento.	Tiempo
7	Latencia en la acción de pausar	Obtener el tiempo promedio de la ejecución de la acción de pausar, a través de la medición del tiempo existente entre la realización de un gesto y el pausado de música, para su posterior evaluación y mejoramiento.	Tiempo
8	Funcionalidad del prototipo	Obtener el porcentaje de acciones ejecutadas correctamente y acciones fallidas a través de la ejecución del prototipo de manera repetida para el posterior análisis y mejoramiento del mismo.	Porcentaje de acciones ejecutadas correctamente, porcentaje de acciones fallidas

**Tabla 1616:** Pruebas para la evaluación del prototipo

En la Tabla 16 se muestran las métricas resultantes de cada una de las pruebas realizadas. A continuación, se detalla cada una de ellas:

**Matriz de confusión.** – La matriz de confusión no se considera una métrica como tal por eso no se muestra entre las métricas en la Tabla 16, pero es el punto de partida para la obtención de métricas como lo es precision y recall. Esta matriz consta de cuatro nomenclaturas principales, Verdaderos Negativo (TN) y Verdaderos Positivos (TP) que son las predicciones que se realizan correctamente para cada una de las clases, y, Falsos Negativos (FN) y Falsos Positivos (FP) que son las predicciones incorrectas para cada una de las clases. A continuación, se muestra la estructura de la matriz de confusión[37]:

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

**Figura 30:** Matriz de confusión. Fuente: [34]

**Precision.** – Esta métrica nos permite obtener una medida de calidad del modelo machine learning, esta medida es obtenida gracias a los Verdaderos Positivos (TP) y Falsos Positivos (FP) como se muestra en la siguiente formula[38]:

$$Precision = \frac{TP}{TP + FP}$$

**Recall.** – El valor recall o exhaustividad, muestra la capacidad que posee el modelo machine learning para realizar una predicción. Esta métrica se obtiene al combinar los Falsos Negativos (FN) y Verdaderos Positivos (TP) como se detalla a continuación[39]:

$$Recall = \frac{TP}{TP + FN}$$

**F1-Score.** – Es el valor obtenido tras la combinación del valor precisión y recall, con ello se asume la importancia de estos dos valores por igual en la demostración del rendimiento de un modelo machine learning. A continuación, se muestra su fórmula[40]:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

**Cross-Validation.** – La validación cruzada es una técnica utilizada para evaluar la eficacia de un modelo con datos que no se utilizaron durante el entrenamiento, para ello se separa parte de los datos para ser utilizados posteriormente como datos de verificación[41].

**Tiempo.** – El tiempo es la sumatoria de los tiempos utilizados por cada uno de los subprocesos utilizados en la ejecución del prototipo como se muestra en el Escenario 2. Para ello se han definido dos fórmulas que ayudan a obtener un tiempo promedio luego de la ejecución del prototipo 100 veces.

Para el Foco BLE:

$$\bar{t} = \frac{\sum_{i=1}^{100}(t_0 + t_1 + t_2 + t_3 + t_4)_i}{100}$$

Para el resto de dispositivos:

$$\bar{t} = \frac{\sum_{i=1}^{100}(t_0 + t_1 + t_2 + t_3)_i}{100}$$

**Porcentaje de acciones ejecutadas correctamente.** – Es el porcentaje obtenido de las acciones realizadas correctamente tras la ejecución de cada uno de los gestos 100 veces. Su fórmula es:

$$\%A_c = A_c \cdot 100 \cdot 0.01$$

**Porcentaje de acciones fallidas.** – Es el porcentaje obtenido de las acciones fallidas tras la ejecución de cada uno de los gestos 100 veces. Su fórmula es:

$$\%A_f = 100 - \%A_c$$

### 3.1.2. Requerimientos

La Tabla 17 describe los requerimientos de Hardware, Software y Físicos para la ejecución de cada uno de los escenarios utilizados para la evaluación del prototipo. Estos requisitos pueden variar de las Herramientas y/o Materiales definidos en el Capítulo II, debido a que algunos

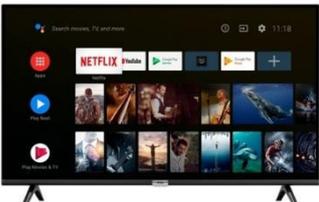
elementos propuestos en estos requisitos son solo con fines de prueba y pueden no estar incluidos en el prototipo final.

Hardware	Software	Físicos
<ul style="list-style-type: none"> <li>• Kit Hexiwear</li> <li>• Móvil Android Xiaomi Mi 8 Lite</li> <li>• Echo dot (3ra Gen.)</li> <li>• Govee WIFI Smart Plugs (B5081102)</li> <li>• Bombilla Sengled BLE (B11-N1EW4P)</li> <li>• Smart TV TCL Android (S6500)</li> <li>• Laptop Asus Vivo Book 15 (X512JA)</li> </ul>	<ul style="list-style-type: none"> <li>• Visual Code</li> <li>• Python</li> <li>• Oracle Cloud</li> <li>• CakeBaked</li> <li>• Amazon Alexa</li> <li>• Aplicación Android Hexiwear.</li> <li>• Android Studio</li> <li>• Mbed Studio</li> </ul>	<ul style="list-style-type: none"> <li>• Casa con conexión a internet</li> </ul>

*Tabla 1717: Requerimientos para la evaluación del prototipo*

En la Tabla 18, se listan los dispositivos que se adquirieron para la realización de cada una de las pruebas, para ello se describe el nombre junto al modelo exacto del dispositivo, una imagen de referencia y el enlace al sitio web de donde se puede adquirir los dispositivos para la replicación o mejoramiento del presente proyecto.

Dispositivo	Imagen	Enlace
Kit Hexiwear		<a href="#">Visitar web</a>
Móvil Android Xiaomi Mi 8 Lite		<a href="#">Visitar web</a>
Echo Dot (3ra Gen)		<a href="#">Visitar web</a>
Govee WIFI Smart Plugs (B5081102)		<a href="#">Visitar web</a>

Dispositivo	Imagen	Enlace
Bombilla Sengled BLE (B11-N1EW4P)		<a href="#">Visitar Web</a>
Smart TV TCL Android (S6500)		<a href="#">Visitar Web</a>
Laptop Asus Vivo Book 15 (X512JA)		<a href="#">Visitar Web</a>

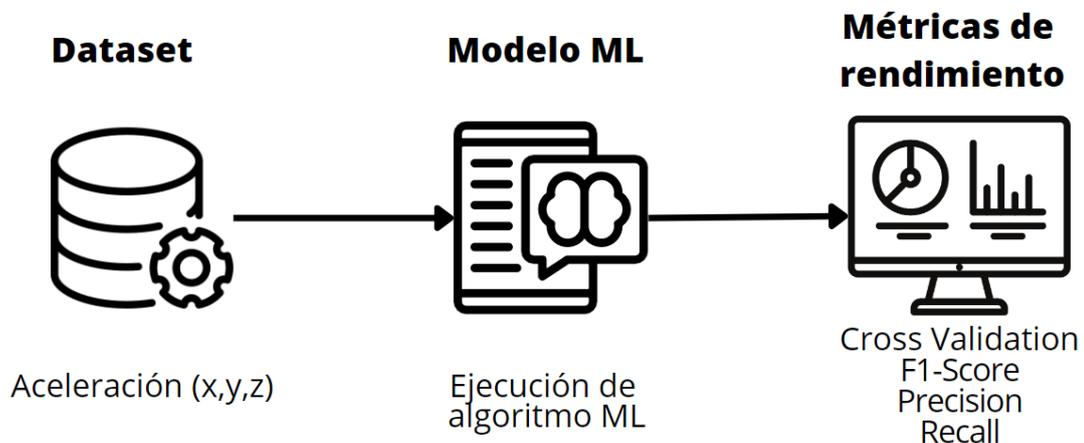
**Tabla 1818:** Modelo de dispositivos utilizados para las pruebas de prototipo

### 3.1.3. Diagrama o estructura de los escenarios

Para la realización de la evaluación del prototipo, se definieron dos escenarios que agrupan varias pruebas cada uno. En cada uno de los escenarios se describen gráfica y textualmente los elementos o requerimientos utilizados durante su ejecución, así como las métricas a obtener.

#### Escenario 1: Medición de rendimiento de algoritmos Machine Learning

El Escenario 1, es utilizado para medir el rendimiento de varios algoritmos machine learning, para así definir el modelo con mejores métricas de rendimiento y así utilizarlo dentro del prototipo final y para las demás pruebas.



**Figura 31:** Escenario 1: Medición de rendimiento de algoritmos Machine Learning

Este primer escenario es utilizado para la realización de las pruebas 1, 2 y 3 que están destinadas a la medición del rendimiento de los algoritmos machine learning y así definir el mejor modelo a través de las métricas F1-Score, Precision, Cross-Validation y Recall. Los tres algoritmos puestos a prueba son KNN, Decision Tree y Random Forest.

Número de Prueba	Algoritmo a Ejecutar	Métricas a obtener
Prueba 1	KNN	<ul style="list-style-type: none"> <li>• F1-Score</li> <li>• Precision</li> <li>• cross-validation</li> <li>• recall</li> </ul>
Prueba 2	Decision Tree	
Prueba 3	Random Forest	

Tabla 1919: Pruebas y métricas que se obtienen para el Escenario 1.

## Escenario 2: Realización de gestos y cambio de estado de dispositivos

El Escenario 2 se utiliza para la realización de la mayoría de las pruebas y para la obtención de métricas como el tiempo promedio de ejecución, el porcentaje de acciones realizadas correctamente y el porcentaje de acciones fallidas. Este escenario contiene la arquitectura IoT de 4 capas descrita en el Capítulo II, debido a que se mide los tiempos de la ejecución de cada uno de los subprocesos existentes en las diferentes capas de la arquitectura.

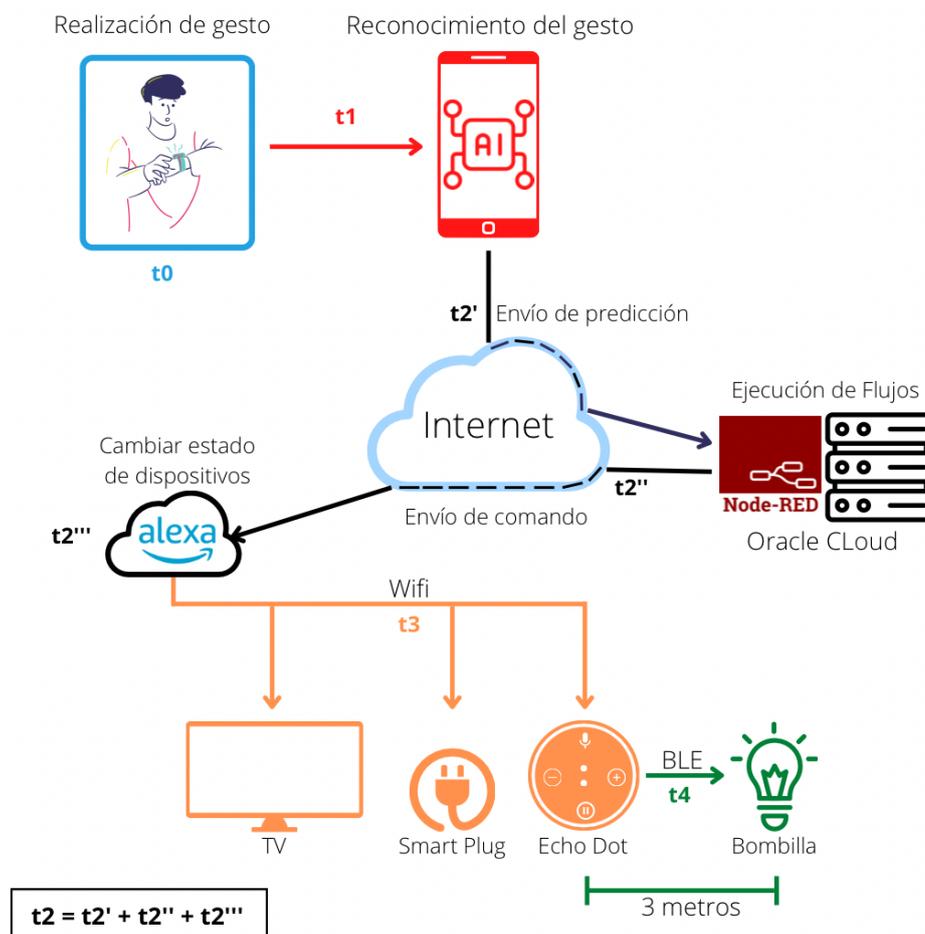


Figura 32: Escenario 2: Realización de gestos y cambio de estado de dispositivos.

En este escenario se ejecutan las pruebas 4, 5, 6,7 y la prueba 8 en simultaneo con las anteriores. Las métricas a obtener son los tiempos promedios de ejecución de cada acción y los porcentajes de acciones correctas y fallidas.

Número de Prueba	Acciones a ejecutar	Métricas a obtener
Prueba 4 y 8	a) Encender Foco	<ul style="list-style-type: none"> <li>Tiempo promedio Para foco: <math>\bar{t} = \frac{\sum_{i=1}^{100}(t_0+t_1+t_2+t_3+t_4)_i}{100}</math> Para demás dispositivos: <math>(\bar{t} = \frac{\sum_{i=1}^{100}(t_0+t_1+t_2+t_3)_i}{100})</math></li> <li>Porcentaje de acciones ejecutadas correctamente (%Ac=A<sub>c</sub>*100*0.01).</li> <li>Porcentaje de acciones fallidas (%A<sub>f</sub>=100-%A<sub>c</sub>).</li> </ul>
	b) Encender TV	
	c) Encender Smart Plug	
Prueba 5 y 8	a) Apagar Foco	
	b) Apagar TV	
	c) Apagar Smart Plug	
Prueba 6 y 8	a) Reproducir música en Echo DOT	
Prueba 7 y 8	a) Pausar música en Echo DOT	

Tabla 2020: Pruebas y métricas que se obtienen para el Escenario 2

## 3.2.Resultados de la evaluación

### 3.2.1. Rendimiento de algoritmos Machine Learning

Para medir el rendimiento de los algoritmos Machine Learning se hizo uso del Escenario 1, descrito en la Figura 32, la herramienta usada fue Python, en específico la librería Sklearn. Lo primero que se obtuvo fue la matriz de confusión, debido a que a partir de ella se obtienen métricas como el F1-Score, Precision, Recall y Cross-Validation.

El primer algoritmo que se probó fue KNN. Este algoritmo fue el que menor rendimiento presentó entre los tres que se probaron. Como primera observación, varios gestos obtuvieron falsos positivos y falsos negativos en la matriz de confusión, dando como resultado un Accuracy de 0.95. Por otro lado, para Cross-Validation se obtuvo un valor mínimo de 0.80 para el gesto hacia Adelante y un valor máximo de 0.90 para el gesto hacia la Izquierda, para Precisión se obtuvo un valor mínimo de 0.91 para el gesto hacia Arriba y un valor máximo de 0.97 para el gesto hacia Adelante, para F1-Score se obtuvo un valor mínimo de 0.91 para el gesto hacia Arriba y un valor máximo de 0.99 para el gesto hacia Adelante, por último para Recall se obtuvo un valor mínimo de 0.91 para el gesto hacia Adelante y un valor máximo de 1.00 para el gesto hacia Adelante. Los resultados completos del rendimiento KNN para cada uno de los gestos se detallan en la Tabla 21.

Algoritmo:		KNN					
		Matriz de confusión:					
Valores reales	Derechaizquierda	35	0	0	2	0	0
	Derecha	0	21	2	0	0	0
	Arriba	0	2	21	0	0	0
	Abajo	0	0	0	51	0	3
	Adelante	0	0	0	0	73	0
	Ruido	2	0	0	3	2	82
		Predicciones					
		Izquierda	Derecha	Arriba	Abajo	Adelante	Ruido
		Métricas por gesto:					
Gesto	Cross Validation	Precision	F1-Score	Recall	Accuracy		
Izquierda	0.90	0.95	0.95	0.95	0.95		
Derecha	0.89	0.91	0.91	0.91			
Arriba	0.86	0.91	0.91	0.91			
Abajo	0.88	0.91	0.93	0.94			
Adelante	0.80	0.97	0.99	1.00			

Tabla 2121: Rendimiento del algoritmo KNN

El segundo algoritmo que se probó fue Decision Tree. Este algoritmo obtuvo un rendimiento regular entre los tres que se probaron. Su matriz de confusión no mostró ningún falso positivo ni falso negativo, dando como resultado un Accuracy de 1.00. Por otro lado, para Cross-Validation se obtuvo un valor mínimo de 0.92 para el gesto hacia la Izquierda y un valor máximo de 1.00 para el gesto hacia abajo, el resto de métricas como son Precisión, F1-Score y Recall obtuvieron un valor de 1.00 para los cinco gestos. Los resultados completos del rendimiento de Decision Tree para cada uno de los gestos se detallan en la Tabla 22.

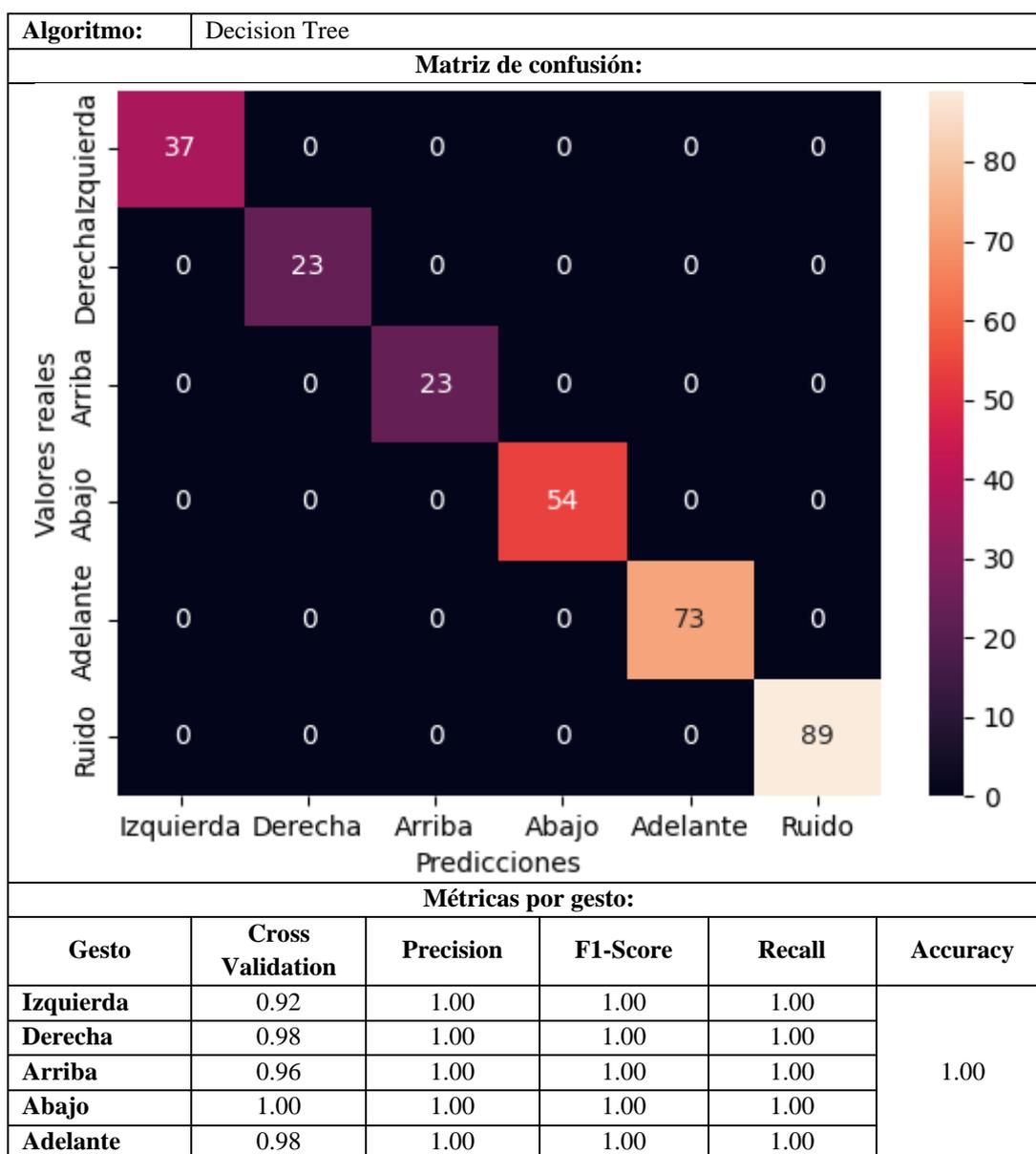


Tabla 2222: Rendimiento del algoritmo Decision Tree

El tercer algoritmo que se probó fue Random Forest. Este algoritmo obtuvo el mejor rendimiento entre los tres que se probaron, razón por la cual fue elegido para ser usado en el prototipo y en las pruebas de latencia y funcionalidad. Su matriz de confusión no mostró ningún falso positivo ni falso negativo, dando como resultado un Accuracy de 1.00. Por otro lado, para Cross-Validation se obtuvo un valor mínimo de 0.97 para el gesto hacia la Arriba y un valor máximo de 1.00 para el gesto hacia Abajo, el resto de métricas como son Precisión, F1-Score y Recall obtuvieron un valor de 1.00 para los cinco gestos. Los resultados completos del rendimiento de Random Forest para cada uno de los gestos se detallan en la Tabla 23.

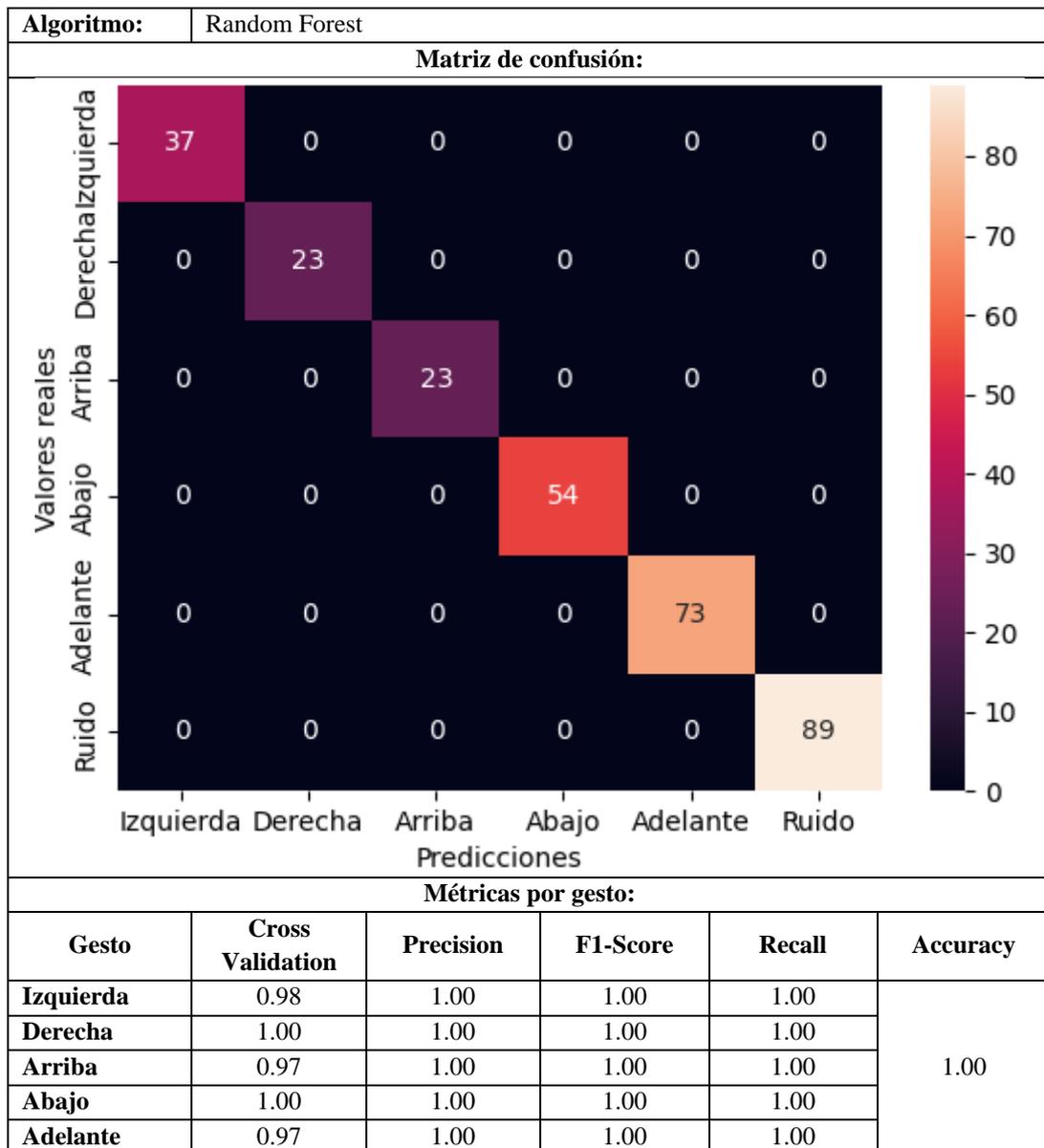


Tabla 2323: Rendimiento del algoritmo Random Forest

### 3.2.2 Latencia y funcionalidad de acciones

Para medir la latencia y funcionalidad de las acciones se utilizó el Escenario 2, el cual es descrito en la figura 32, para la realización de pruebas se modificó el software de manera que permita obtener las métricas necesarias entre diferentes acciones para determinar resultados como tiempo de ejecución y el porcentaje de acciones correctas y fallidas. Para determinar la latencia de los de la realización de las acciones, se tuvo en consideración el tiempo desde que el software del wearable se demora en armar el conjunto de datos ( $t_0$ ); el tiempo en el que el algoritmo de clasificación reconoce el gesto ( $t_1$ ); el conjunto de tiempos desde el envío de la predicción del gesto del Gateway al servidor en Oracle Cloud ( $t_2'$ ), el tiempo de ejecución de flujos en Node Red ( $t_2''$ ), el tiempo del envío del comando desde el servidor a la nube de Alexa ( $t_2'''$ ); el tiempo de cambio de estado de los dispositivos ( $t_3$  y  $t_4$ ).

Cada acción se repitió 100 veces, para la medición del tiempo no se tomaron en cuenta los intentos fallidos, ero estos si se anotaron para medir la funcionalidad del prototipo. Con respecto a los tiempos promedio de ejecución, la acción de Encender el Smart Plug es la que menos tiempo llevó realizarse con un tiempo de 3.13 segundos, mientras que Apagar Smart TV fue la que mayor tiempo de ejecución obtuvo con 4.18 segundos, obteniendo un tiempo promedio general de 3.71 segundos. Por otro lado, en cuanto a la funcionalidad, la acción de Encender Foco fue la que menos intentos fallidos contabilizó, mientras que Encender Smart Plug fue la acción con más intentos fallidos obteniendo un promedio general de 91.5% de efectividad. En la Tabla 22 se detalla cada uno de los gestos con su tiempo promedio de ejecución en segundos, los intentos correctos e incorrectos con el porcentaje que estos representan.

Acción	Tiempo promedio de ejecución (segundos)	Intentos Correctos		Intentos Fallidos	
		Intentos	Porcentaje	Intentos	Porcentaje
Encender Smart TV	4.16	100	89%	12	11%
Apagar Smart TV	4.18	100	93%	7	7%
Encender Foco	3.93	100	92%	9	8%
Apagar Foco	3.93	100	95%	5	5%
Encender Smart Plug	3.13	100	88%	14	12%
Apagar Smart Plug	3.32	100	90%	11	10%
Reproducir Echo Dot	3.55	100	93%	7	7%
Pausar Echo Dot	3.46	100	92%	9	8%
<b>Promedio</b>	3.71	100	91.5%	9.25	8.5%

**Tabla 2424:** Latencia y funcionalidad de acciones

#### 4. CONCLUSIONES

Se realizó una búsqueda bibliográfica que sirvió como punto de partida para el desarrollo del presente trabajo de titulación. Entre los principales hallazgos que se encontraron fueron los diferentes tipos y algoritmos de machine learning que se pueden usar para realizar el reconocimiento de gestos, es por ello que se decidió usar algoritmos de clasificación.

Se diseñó el prototipo utilizando la metodología IoT Methodology y utilizando una arquitectura de cuatro capas, aplicación, servicio, red y dispositivos. Con esto se obtuvo un prototipo usando el wearable Hexiwear y conectándose a través de Bluetooth Low Energy (BLE) a un smartphone, siendo capaz de detectar cinco gestos que son: mano hacia arriba como gesto de apertura para el reconocimiento de los demás comandos, además, se usa este mismo gesto para encender/apagar un foco, mano hacia abajo para reproducir/pausar el Echo Dot, hacia la derecha para

encender/apagar Smart TV, hacia la izquierda para activar/desactivar escena y hacia adelante para encender/apagar Smart Plug.

Se obtuvo un modelo machine learning haciendo una comparación de rendimiento entre los algoritmos KNN, Decision Tree y Random Forest, del cual este último obtuvo mejores métricas de rendimientos. Este modelo realiza predicciones analizando las características de varianza, simetría y curtosis de los conjuntos de datos de x, y, z que se obtienen del acelerómetro del dispositivo Hexiwear en un periodo de tiempo determinado.

Se elaboró el firmware del dispositivo Hexiwear, con la capacidad de recolectar los datos del sensor acelerómetro integrado en el wearable, estos datos se convierten de unidad de gravedad(g) a aceleración( $m/s^2$ ) para posteriormente ser enviado por Bluetooth Low Energy a un smartphone para su respectivo análisis y predicción con el modelo machine learning.

Se obtuvieron resultados satisfactorios en la evaluación del prototipo, para ello se crearon dos escenarios, el primer escenario para la elección del algoritmo machine learning en donde Random Forest resultó elegido por presentar mejores métricas, y el segundo escenario para medir el tiempo de ejecución y la funcionalidad de cada acción, en donde la acción de Encender el Smart Plug fue la que menos tiempo llevó realizarse con 3.13 segundos, mientras que Apagar Smart TV fue la que mayor tiempo de ejecución obtuvo con 4.18 segundos. Por otro lado, en cuanto a la funcionalidad, la acción de Encender Foco fue la que menos intentos fallidos contabilizó con un 8%, mientras que Encender Smart Plug fue la acción con más intentos fallidos con un 12%.

Se consiguió desarrollar de una aplicación wearable con reconocimiento de gestos para smart home obteniendo resultados satisfactorios, pues el reconocimiento de los gestos y la activación de dispositivos se realizó correctamente en un 88% de las veces en que se probaron los distintos gestos y activación de dispositivos.

## **5. RECOMENDACIONES**

Se recomienda realizar una investigación previa sobre las técnicas machine learning, porque según el caso puede ser más útil aprendizaje supervisado o el no supervisado, además cada uno de ellos posee más subcategorías.

Se recomienda aplicar una metodología ágil que ayude a planificar correctamente el proyecto en todas sus etapas, así como la elección de una arquitectura para aplicar en el diseño e implementación del prototipo.

Se recomienda probar el entrenamiento y medición de rendimiento del modelo utilizando diferentes algoritmos de machine learning, porque la data puede funcionar mejor con un tipo de algoritmo que con otros y así se puede elegir el mejor para el proyecto.

Se recomienda manejar todos los datos con unidades de medidas estándar por ejemplo el Sistema Internacional de Unidades o el Sistema Ingles de Medidas, evitando mezclar los dos u otros para no cometer errores al realizar cálculos con dichos datos.

Se recomienda realizar varias pruebas en un mismo escenario de pruebas para la optimización de recursos como el tiempo durante el proceso de evaluación del prototipo. También, cada prueba debe realizarse varias veces y al final obtener promedios de cada una de las métricas conseguidas.

Se recomienda entrenar el modelo con nuevos gestos o utilizando la combinación de los datos de varios sensores para obtener mejores resultados y superar el 88% de activación correcta de dispositivos que se consiguió con el modelo utilizado en el presente trabajo.

## 6. REFERENCIAS BIBLIOGRÁFICAS

- [1] I. Ramírez-Morales, B. Mazon-Olivo, and D. Hernández-Rojas, “Análisis de Datos Agropecuarios.”
- [2] A. Wright, “U.S. Smart Home Device Forecast Update, 2021–2025: CY 4Q21,” Nov. 2021. <https://www.idc.com/getdoc.jsp?containerId=US48305821> (accessed Nov. 27, 2021).
- [3] Statista, “Conexiones IoT en América Latina por modelo de negocio 2025 | Statista,” Jul. 02, 2021. <https://es.statista.com/estadisticas/1190778/conexiones-iot-america-latina-por-modelo-negocio/> (accessed Nov. 27, 2021).
- [4] el Telégrafo, “La internet de las cosas se usa más en seguridad.” <https://www.eltelegrafo.com.ec/noticias/tecnologia/1/internet-seguridad-dispositivos-electronicos-datos> (accessed Nov. 27, 2021).
- [5] A. I. Faisal, S. Majumder, R. Scott, T. Mondal, D. Cowan, and M. Jamal Deen, “A Simple, Low-Cost Multi-Sensor-Based Smart Wearable Knee Monitoring System,” *IEEE Sens J*, vol. 21, no. 6, pp. 8253–8266, Mar. 2021, doi: 10.1109/JSEN.2020.3044784.
- [6] D. Carrizo, C. Moller, D. Carrizo, and C. Moller, “Estructuras metodológicas de revisiones sistemáticas de literatura en Ingeniería de Software: un estudio de mapeo sistemático,” *Ingeniare. Revista chilena de ingeniería*, vol. 26, pp. 45–54, Nov. 2018, doi: 10.4067/S0718-33052018000500045.
- [7] J. Novillo-Vicuña, D. Hernández Rojas, B. Mazón Olivo, J. Molina Ríos, and O. Cárdenas Villavicencio, *Arduino y el Internet de las cosas*. Editorial Científica 3Ciencias, 2018. doi: 10.17993/ingytec.2018.45.
- [8] D. Ferraris, D. Bastos, C. Fernandez-Gago, and F. El-Moussa, “A trust model for popular smart home devices,” *Int J Inf Secur*, vol. 20, no. 4, pp. 571–587, Aug. 2021, doi: 10.1007/S10207-020-00519-2.
- [9] D. Marikyan, S. Papagiannidis, and E. Alamanos, “Smart Home Sweet Smart Home,” *International Journal of E-Business Research*, vol. 17, no. 2, pp. 1–23, Feb. 2021, doi: 10.4018/IJEER.2021040101.
- [10] P. C. Siswipraptini, R. N. Aziza, I. Sangadji, Indrianto, R. R. A. Siregar, and G. Sondakh, “IoT for smart home system,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 733–739, Aug. 2021, doi: 10.11591/IJEECS.V23.I2.PP733-739.
- [11] A. García-Pérez and J. A. Mesa, “Sustainable engineering and Internet of Things (IoT): Trends and perspectives,” *Proceedings of the LACCEI international Multi-conference for Engineering, Education and Technology*, 2020, doi: 10.18687/LACCEI2020.1.1.109.
- [12] C. Bodei, P. Degano, G. L. Ferrari, and L. Galletta, “Modelling and analysing IoT systems,” *J Parallel Distrib Comput*, vol. 157, pp. 233–242, Nov. 2021, doi: 10.1016/J.JPDC.2021.07.004.
- [13] M. Kurz, R. Gstoettner, and E. Sonnleitner, “Smart Rings vs. Smartwatches: Utilizing Motion Sensors for Gesture Recognition,” *Applied Sciences*, vol. 11, no. 5, p. 2015, Feb. 2021, doi: 10.3390/APP11052015.
- [14] M. Silverio-Fernández, S. Renukappa, and S. Suresh, “What is a smart device? - a conceptualisation within the paradigm of the internet of things,” *Visualization in Engineering*, vol. 6, no. 1, Dec. 2018, doi: 10.1186/S40327-018-0063-8.

- [15] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas, and A. Pan, "Rules engine and complex event processor in the context of internet of things for precision agriculture," *Comput Electron Agric*, vol. 154, pp. 347–360, Nov. 2018, doi: 10.1016/j.compag.2018.09.013.
- [16] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin, "Benefits and risks of smart home technologies," *Energy Policy*, vol. 103, pp. 72–83, Apr. 2017, doi: 10.1016/J.ENPOL.2016.12.047.
- [17] S. Chang and K. Nam, "Smart Home Adoption: The Impact of User Characteristics and Differences in Perception of Benefits," *Buildings 2021, Vol. 11, Page 393*, vol. 11, no. 9, p. 393, Sep. 2021, doi: 10.3390/BUILDINGS11090393.
- [18] "IoT Methodology – The Internet of Things project lifecycle guide for creative, technical and business people." <http://www.iotmethodology.com/> (accessed Dec. 17, 2021).
- [19] S. Sharma and S. Singh, "Vision-based hand gesture recognition using deep learning for the interpretation of sign language," *Expert Syst Appl*, vol. 182, Nov. 2021, doi: 10.1016/j.eswa.2021.115657.
- [20] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, "A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors," *IEEE Sens J*, vol. 16, no. 16, pp. 6425–6432, Aug. 2016, doi: 10.1109/JSEN.2016.2581023.
- [21] A. Qaroush, S. Yassin, A. Al-Nubani, and A. Alqam, "Smart, comfortable wearable system for recognizing Arabic Sign Language in real-time using IMUs and features-based fusion," *Expert Syst Appl*, vol. 184, p. 115448, Dec. 2021, doi: 10.1016/j.eswa.2021.115448.
- [22] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 131–153, Jan. 2019, doi: 10.1007/s13042-017-0705-5.
- [23] A. Adegboye and M. Kampouridis, "Machine learning classification and regression models for predicting directional changes trend reversal in FX markets," *Expert Syst Appl*, vol. 173, Jul. 2021, doi: 10.1016/j.eswa.2021.114645.
- [24] A. Roohi, K. Faust, U. Djuric, and P. Diamandis, "Unsupervised Machine Learning in Pathology: The Next Frontier," *Surgical Pathology Clinics*, vol. 13, no. 2. W.B. Saunders, pp. 349–358, Jun. 01, 2020. doi: 10.1016/j.path.2020.01.002.
- [25] B. Ulfenborg, A. Karlsson, M. Riveiro, C. X. Andersson, P. Sartipy, and J. Synnergren, "Multi-assignment clustering: Machine learning from a biological perspective," *J Biotechnol*, vol. 326, pp. 1–10, Jan. 2021, doi: 10.1016/j.jbiotec.2020.12.002.
- [26] S. Singh and A. H. Ganie, "Applications of picture fuzzy similarity measures in pattern recognition, clustering, and MADM," *Expert Syst Appl*, vol. 168, Apr. 2021, doi: 10.1016/j.eswa.2020.114264.
- [27] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," *Information Fusion*, vol. 59, pp. 44–58, Jul. 2020, doi: 10.1016/j.inffus.2020.01.005.
- [28] P. C. Santana-Mancilla, L. E. Anido-Rifón, J. Contreras-Castillo, and R. Buenrostro-Mariscal, "Heuristic Evaluation of an IoMT System for Remote Health Monitoring in Senior Care," *International Journal of Environmental Research and Public Health 2020, Vol. 17, Page 1586*, vol. 17, no. 5, p. 1586, Mar. 2020, doi: 10.3390/IJERPH17051586.

- [29] J. Li *et al.*, “Real-Time Hand Gesture Tracking for Human-Computer Interface Based on Multi-Sensor Data Fusion,” *IEEE Sens J*, vol. 21, no. 23, pp. 26642–26654, Dec. 2021, doi: 10.1109/JSEN.2021.3122236.
- [30] M. Kurz, R. Gstoettner, and E. Sonnleitner, “Smart Rings vs. Smartwatches: Utilizing Motion Sensors for Gesture Recognition,” *Applied Sciences*, vol. 11, no. 5, p. 2015, Feb. 2021, doi: 10.3390/APP11052015.
- [31] H. P. Gupta, H. S. Chudgar, S. Mukherjee, T. Dutta, and K. Sharma, “A Continuous Hand Gestures Recognition Technique for Human-Machine Interaction Using Accelerometer and Gyroscope Sensors,” *IEEE Sens J*, vol. 16, no. 16, pp. 6425–6432, Aug. 2016, doi: 10.1109/JSEN.2016.2581023.
- [32] W. Jiang *et al.*, “Wearable on-device deep learning system for hand gesture recognition based on FPGA accelerator,” *Mathematical Biosciences and Engineering*, vol. 18, no. 1, pp. 132–153, 2020, doi: 10.3934/MBE.2021007.
- [33] S. M. Baba and M. Tariq Banday, “Application development for wearable internet of things using hexiwear,” *2020 7th International Conference on Signal Processing and Integrated Networks, SPIN 2020*, pp. 542–548, Feb. 2020, doi: 10.1109/SPIN48934.2020.9071090.
- [34] D. Iyer *et al.*, “Generalized hand gesture recognition for wearable devices in iot: Application and implementation challenges,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9729, pp. 346–355. doi: 10.1007/978-3-319-41920-6\_26.
- [35] U. A. Siddiqui *et al.*, “Wearable-sensors-based platform for gesture recognition of autism spectrum disorder children using machine learning algorithms,” *Sensors*, vol. 21, no. 10, May 2021, doi: 10.3390/s21103319.
- [36] M. Kim, “A Quality Model for Evaluating IoT Applications,” *undefined*, vol. 8, no. 1, pp. 66–76, 2016, doi: 10.17706/IJCEE.2016.8.1.66-76.
- [37] “Precision, Recall, F1, Accuracy en clasificación - IArtificial.net.” <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/> (accessed Jul. 29, 2022).
- [38] “sklearn.metrics.precision\_score — scikit-learn 1.1.1 documentation.” [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision\\_score.html?highlight=precision#sklearn.metrics.precision\\_score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html?highlight=precision#sklearn.metrics.precision_score) (accessed Jul. 29, 2022).
- [39] “sklearn.metrics.recall\_score — scikit-learn 1.1.1 documentation.” [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall\\_score.html?highlight=recall#sklearn.metrics.recall\\_score](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html?highlight=recall#sklearn.metrics.recall_score) (accessed Jul. 29, 2022).
- [40] “sklearn.metrics.f1\_score — scikit-learn 1.1.1 documentation.” [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html) (accessed Jul. 29, 2022).
- [41] “Cross-Validation : definición e importancia en Machine Learning.” <https://datascientest.com/es/cross-validation-definicion-e-importancia> (accessed Jul. 29, 2022).

## 7. ANEXOS

### Anexo 1: Matriz de consistencia

Problema, objeto y campo	Objetivo	Marco Teórico	Hipótesis	Variables	Metodología
<p><b>Problema:</b> ¿Cómo desarrollar una aplicación wearable con reconocimiento de gestos e integración con smart home para personas con hogares inteligentes en la ciudad de Machala, durante el periodo 2021 – 2022?</p> <p><b>Problemas específicos:</b></p> <ul style="list-style-type: none"> <li>• ¿Cómo controlar un entorno Smart Home con una aplicación wearable?</li> <li>• ¿Cómo obtener los datos de movimiento de la mano a través de sensores?</li> <li>• ¿Cómo crear un firmware para el funcionamiento de un wearable?</li> <li>• ¿Cómo integrar la aplicación wearable para el control de Smart Home?</li> </ul> <p><b>Objeto de estudio:</b> Aplicación wearable con reconocimiento de gestos para entornos Smart Home</p> <p><b>Campo de Acción:</b> Desarrollo de una aplicación wearable con aprendizaje autónomo supervisado y predictivo para Smart Home.</p>	<p><b>Objetivo General:</b> - Desarrollar una aplicación wearable con capacidad de reconocimiento de gestos, mediante aprendizaje automático con técnicas machine learning, para el control de dispositivos integrados a una Smart Home.</p> <p><b>Objetivos Específicos:</b></p> <ul style="list-style-type: none"> <li>• Realizar una búsqueda bibliográfica para la elaboración del estado del arte y marco teórico.</li> <li>• Diseñar el prototipo utilizando una metodología para el desarrollo de aplicaciones wearable.</li> <li>• Implementar un modelo de aprendizaje automático para reconocimiento de gestos.</li> <li>• Elaborar el firmware(aplicación) con las funciones necesarias para el wearable.</li> <li>• Realizar las pruebas de funcionamiento del prototipo en un entorno Smart Home.</li> </ul>	<p><b>Antecedentes históricos a nivel internacional y nacional del objeto, campo:</b></p> <ul style="list-style-type: none"> <li>• Smart home en la actualidad</li> </ul> <p><b>Fundamentos Teóricos de objeto, campo y variables:</b></p> <ul style="list-style-type: none"> <li>• Smart Home</li> <li>• Tipos de Reconocimientos de Gestos</li> <li>• Machine Learning</li> <li>• Hexiwear</li> </ul>	<p>Hipótesis General: - El desarrollo de la aplicación wearable, con técnicas machine learning para el reconocimiento de gestos facilita a los usuarios el control de sus dispositivos conectados al entorno Smart Home.</p>	<p><b>Variable 1 / Independiente:</b> - Aplicación wearable con técnicas machine learning</p> <p><b>Dimensiones o categorías:</b></p> <ol style="list-style-type: none"> <li>1. Internet of Things</li> <li>2. Machine learning</li> <li>3. Plataforma de hardware</li> </ol> <p><b>Variable 2/ Dependiente:</b> - Control de dispositivos conectados al entorno Smart Home</p> <p><b>Dimensiones o categorías:</b></p> <ol style="list-style-type: none"> <li>1. Experiencia de usuario</li> <li>2. Funcionamiento</li> </ol>	<p><b>Enfoque:</b> El enfoque será cuantitativo, debido a que la investigación se basará en la medición de variables cuyos valores son de datos numéricos y se espera que los resultados también se apoyen en datos numéricos y estadísticos.</p> <p><b>Alcance:</b> El alcance de la investigación se enfocará principalmente en descriptivo, debido a que se estudiarán características del fenómeno, es decir los movimientos de la mano del ser humano, pasando a medir variables a través de sensores que posteriormente serán puestas bajo análisis.</p> <p><b>Diseño:</b> El diseño de la investigación será cuasi-experimental, porque el sujeto de estudio no se elige de manera aleatoria, si no se elige previo al inicio de la investigación, que en este caso el sujeto sería los gestos que una persona puede realizar con su mano.</p> <p><b>Unidades de análisis:</b> <b>Población:</b> La población utilizada en el presente estudio serán los</p>

				<p>datos arrojados por los sensores (giroscopio y acelerómetro) del dispositivo wearable.</p> <p><b>Muestra:</b> No se obtendrá una muestra para el presente trabajo.</p> <p><b>Técnicas e instrumentos de recolección de datos:</b> Técnicas - Observación y - Análisis de datos</p> <p>Instrumento - Guía de observación y análisis de control - Guía de análisis de datos</p> <p><b>Técnicas de procesamiento de datos:</b> Tras la revisión bibliográfica de trabajos similares se concluyó que la técnica de procesamiento de datos más adecuada para llevar a cabo la investigación es el Machine Learning con aprendizaje supervisado, específicamente algoritmos de clasificación, todo esto debido a la naturaleza de la investigación y los resultados a los que se pretende llegar.</p>
--	--	--	--	--

## Anexo 2: Guía de observación y análisis de control para las pruebas de prototipo

Fecha:	
Observador/es:	
Objetivo:	
Ubicación:	
Dispositivo/s:	
Algoritmo usado para el entrenamiento	

Responder de acuerdo a lo observado:

No.	Aspecto a evaluar	Si	No	Talvez	Observación
1	La acción del dispositivo reacciona al gesto pre asignado				
2	Los tiempos de acción de los dispositivos son rápidos				
3	Los dispositivos realizan acciones al realizar gestos no intensionales				
4	El número de aciertos es mayor al número de errores al enviar comandos a los dispositivos				
5	El wearable no genera incomodidad al realizar los gestos				

## Anexo 3: Guía de análisis de datos

Fecha:	
Responsable:	
Prueba #	

Técnica u algoritmo	M. Confusión	Precisión	F1 Score	Observación