



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

CONTROL DOMÓTICO CON ASISTENTE DE VOZ Y REALIDAD
AUMENTADA PARA EL INTERNET DE LAS COSAS.

CHIMBOLEMA MACAS CRISTIAN OMAR
INGENIERO DE SISTEMAS

MACHALA
2020



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

Control Domótico con asistente de voz y realidad aumentada para el
internet de las cosas.

CHIMBOLEMA MACAS CRISTIAN OMAR
INGENIERO DE SISTEMAS

MACHALA
2020



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

Control Domótico con asistente de voz y realidad aumentada para el internet de las cosas.

CHIMBOLEMA MACAS CRISTIAN OMAR
INGENIERO DE SISTEMAS

HERNANDEZ ROJAS DIXYS LEONARDO

MACHALA, 18 DE DICIEMBRE DE 2020

MACHALA
2020

tesis chimbolema

INFORME DE ORIGINALIDAD

2%

INDICE DE SIMILITUD

1%

FUENTES DE
INTERNET

0%

PUBLICACIONES

1%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

1

Submitted to Universidad Técnica de Machala

Trabajo del estudiante

1%

2

repositorio.utn.edu.ec

Fuente de Internet

<1%

3

Submitted to Universidad Ricardo Palma

Trabajo del estudiante

<1%

4

repositorio.utmachala.edu.ec

Fuente de Internet

<1%

5

Submitted to Consorcio CIXUG

Trabajo del estudiante

<1%

6

scielo.senescyt.gob.ec

Fuente de Internet

<1%

7

doku.pub

Fuente de Internet

<1%

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

El que suscribe, CHIMBOLEMA MACAS CRISTIAN OMAR, en calidad de autor del siguiente trabajo escrito titulado Control Domótico con asistente de voz y realidad aumentada para el internet de las cosas., otorga 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 tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara 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.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume 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.

Machala, 18 de diciembre de 2020



CHIMBOLEMA MACAS CRISTIAN OMAR
0705458198

DEDICATORIA

El presente trabajo de titulación se lo dedico es una evidencia del transcurso en la Carrera de Ingeniería de Sistemas, doy las gracias a mis padres que me han apoyado durante el transcurso de mi formación académica.

A aquellas personas que siempre me dieron motivación para no rendirme y confiaron en mí.

AGRADECIMIENTO

Doy las gracias a Dios, por brindarme la oportunidad de culminar con éxito mi carrera universitaria. A mis padres, quienes me han enseñado la importancia de perseverar y no rendirse para alcanzar nuestros sueños. Y a mi tutor del trabajo de titulación, el Ing. Dixys Leonardo Hernández Rojas por su constante guía y apoyo durante su culminación.

RESUMEN

El presente trabajo de titulación tuvo como objetivo la implementación de control domótico para el internet de las cosas (IoT) mediante el asistente de voz de Google home y realidad aumentada. Se utilizó el asistente de voz de Google home para la interacción de los elementos los cuales interactúan entre sí mediante la Cloud de los proveedores.

Los dispositivos que se usaron son una bombilla led con cambio de múltiples colores y un enchufe inteligente de las marcas Nexxt, los cuales forman parte del sistema IoT domótico junto con el uso de Home Assistant para el servicio de automatización ya que esta plataforma nos brinda facilidades de servicio e información, control de dispositivos, rastreo y monitoreo de estado del hogar entre otros que facilitan al usuario la configuración de reglas a sus dispositivos inteligentes.

Esta plataforma IoT está alojada en una Raspberry Pi3 B+, se debe a que nos presenta un dispositivo con un buen procesamiento de datos y conectividad enfocado únicamente en el control y procesamiento de los datos de los dispositivos inteligentes.

Además, se cuenta con la ayuda de la herramienta Unity para el desarrollo de la aplicación en realidad aumentada, la cual presenta un ambiente interactivo que permite el monitoreo y control de los dispositivos inteligentes, para el desarrollo proceso de detección se realizó mediante imágenes alojadas en el servidor de base de datos de Vuforia. Lo que se requiere es un ambiente interactivo que permita al usuario visualizar y acceder a la información en tiempo real dentro del mundo real.

El proceso por el que se realiza la conexión de datos se lleva a cabo mediante el uso del protocolo MQTT, debido a que es uno de los medios de comunicación más popular dentro de IoT por sus características tales como ligereza, no requieren dispositivos de gran potencia, menor consumo de energía, requiere un

mínimo de ancho de banda, sin olvidar que presenta una mejor facilidad de uso ya que su estructura conformada por un bróker que distribuye la información de clientes publicadores a suscritos.

Uno de los medios usados para conectar nuestro hardware con sus respectivos servicios es Node-Red, el cual se encuentra integrado en la plataforma HASSIO, permitiendo el uso de automatizaciones de los diferentes dispositivos de hardware que se encuentran vinculados a nuestra plataforma. Lo hace mediante un conjunto de nodos disponibles que puede incorporar funciones, plantillas o flujos a disposición de los usuarios, lo cual ayuda a generar un ambiente de trabajo fácil de incorporar.

Al finalizar este trabajo se obtiene un sistema domótico interconectado mediante el uso de la plataforma Home Assistant, la cual integra los dispositivos además del protocolo MQTT encargado de su comunicación, junto con node-red que se encarga mediante un flujo de nodo de receptar instrucciones e iniciar el servicio correspondiente. Y una aplicación móvil para Android con objetos en 3D que permitan monitorear y controlar los elementos que se encuentran dentro del sistema domótico, como también un control mediante el asistente de voz haciendo uso de google home.

Palabras clave: Asistente de voz, Domótica, Home Assistant, Realidad Aumentada, Unity.

ABSTRACT

The objective of this degree work was the implementation of domotic control for the Internet of Things (IoT) through the voice assistant of Google home and augmented reality. I used the Google home voice assistant to interact with the elements that interact with each other through the cloud of the providers.

The devices used are a multi-color LED bulb and a smart plug from the Nexxt brands. Which are part of the home automation IoT system along with the use of Home Assistant for automation service as this platform provides us with service and information facilities, device control, tracking and monitoring home status and others make it easy for the user to set up rules for their smart devices.

This IoT platform is hosted on a Raspberry PI3 B , because it presents us with a device with good data processing and connectivity focused solely on the control and processing of data from smart devices.

In addition, the Unity tool for the development of the Augmented Reality application is available, which features an interactive environment that allows monitoring and control of smart devices. For the development process of detection was carried out by means of images hosted on the database server of Vuforia. What is required is an interactive environment that allows the user to view and access information in real time within the real world.

The process by which the data connection is made is carried out by using the MQTT protocol, because it is one of the most popular media within IoT because of its characteristics such as lightness, they do not require devices of great influence. lower power consumption, requires a minimum of bandwidth, not forgetting that it has a better ease of use since its structure formed by a broker that distributes information from publishing clients to subservients.

One of the means used to connect our hardware with their respective services is Node-Red, which is integrated into the HASSIO platform, allowing the use of

automations of the different hardware devices that are linked to our platform. It does this through a set of available nodes that can incorporate functions, templates, or flows available to users, helping to create an easy-to-incorporate work environment.

At the end of this work, you get an interconnected domotic system using the Home Assistant platform, which integrates the devices in addition to the MQTT protocol responsible for their communication, along with node-network that is responsible for a node flow of receiving instructions and starting the corresponding service. And a mobile Android app with 3D objects that allow you to monitor and control the items inside the home system, as well as a voice assistant control using google home.

Keywords: Voice Assistant, Home Assistant, Augmented Reality, Unity.

CONTENIDO

DEDICATORIA	1
AGRADECIMIENTO	2
RESUMEN	3
ABSTRACT	5
INTRODUCCIÓN	12
1 CAPÍTULO I. MARCO TEÓRICO	14
1.1 Problema	14
1.2 Objetivo general	14
1.3 Internet de las Cosas (IoT)	14
1.3.1 Plataformas IoT	14
1.3.1.1 OpenHab	15
1.3.1.2 Domoticz	15
1.3.1.3 Home Assistant	15
1.4 Domótica	15
1.5 Realidad Aumentada	15
2 CAPÍTULO II. DISEÑO DE PROTOTIPO	17
2.1 Back-End	17
2.2 Front_End	17
2.3 Análisis de requerimientos	19
3 CAPÍTULO III. IMPLEMENTACIÓN DE PROTOTIPO	19
3.1 Home Assistant – HASSIO	19
3.1.1 Instalación de HASSIO en Raspberry pi3 B+.	19
3.1.2 Instalación de complementos de HASSIO para realizar configuraciones	22
3.2 Integración de dispositivos	28
3.3 Unity	29
3.3.1 Descarga de librerías de realidad aumentada Easyar para Unity	29
3.4 Ejecución de prototipo	33
4 CAPÍTULO IV. EXPERIMENTOS	43
4.1 Experimentos de detección	43
4.1.1 Cálculo de latencia	43
4.1.2 Objetivo QR	50
4.1.3 Objetivo Cámara (Object detection)	50
4.2 Control ON	50
4.2.1 Interfaz de Usuario – Botón	50
4.2.2 Detección de Gesto – Hand Detection	51
4.3 Control OFF	51
4.3.1 Interfaz de Usuario – Botón	51
4.3.2 Detección de Gesto – Hand Detection	51

4.4	Análisis de resultados	52
4.5	CONCLUSIONES	53
4.6	RECOMENDACIONES	54
5	<i>Bibliografía</i>	56

LISTA DE FIGURAS

Figura 1 Back-End de aplicación domótica	17
Figura 2 Prototipo de interfaz de usuario - conexión MQTT broker	18
Figura 3 Prototipo de interfaz de usuario - Detección de elementos mediante cámara	19
Figura 4 Registro de usuario HASSIO	21
Figura 5 Configuración de localización de HASSIO	22
Figura 6 Add-ons de Home Assistant	23
Figura 7 Búsqueda de File Editor en Add-ons	23
Figura 8 Menú de File Editor	24
Figura 9 Búsqueda de Terminal en Add-ons	24
Figura 10 Menú de Terminal	25
Figura 11 Ventana principal de Terminal de Home Assistant	25
Figura 12 Búsqueda de Samba Share en Add-ons	26
Figura 13 Configuración de Samba Share	27
Figura 14 Menú de Samba Share	28
Figura 15 Configuración de nueva Integración en Home Assistant	29
Figura 16 Ingreso de credenciales Tuya	29
Figura 17 Plugin EasyAR	30
Figura 18 Importación EasyAR a Unity	30
Figura 19 Registro de License Key de EasyAR	31
Figura 20 Agregación de Licencia de EasyAR en Unity	32
Figura 21 EasyAR Controlador	32
Figura 22 EasyAR ImageTarget	33
Figura 23 Implementación de ImageTarget	33
Figura 24 Panel de Control de dispositivos - HASSIO	34
Figura 25 Configuración Trigger State	35
Figura 26 Configuración de nodo MQTT OUT	36
Figura 27 Configuración de servidor	36
Figura 28 Configuración de servidor homeassistant	37
Figura 29 Disparador de estado actual de dispositivo	37
Figura 30 Configuración de nodo MQTT IN	38
Figura 31 Configuración de nodo Switch	38

Figura 32 Enrutamiento de nodo Switch	39
Figura 33 Configuración de nodo Call service	39
Figura 34 Propiedades de servicio ON para dispositivo switch	40
Figura 35 Estructura de control para recepción de comando y activación de servicio	40
Figura 36 Conexión con Broker MQTT – Aplicación en Android	41
Figura 37 Reconocimiento QR Enchufe	42
Figura 38 Reconocimiento QR Bombilla	42
Figura 39 Consola - Detección QR	43
Figura 40 Cálculo de latencia - Detección Cámara-Objeto	44
Figura 41 Consola - Control ON por Botón	45
Figura 42 Consola HASSIO - Estado ON	45
Figura 43 Consola - Tiempo de respuesta HASSIO	45
Figura 44 Consola - Control OFF por Botón	46
Figura 45 Consola HASSIO - Estado OFF	46
Figura 46 Consola - Tiempo de respuesta HASSIO	47
Figura 47 Consola - Control ON por Gesto	47
Figura 48 Consola HASSIO - Estado ON Gesto	48
Figura 49 Consola - Tiempo de respuesta HASSIO	48
Figura 50 Consola - Control OFF por Gesto	49
Figura 51 Consola HASSIO - Control OFF por Gesto	49
Figura 52 Consola - Tiempo de respuesta HASSIO	49
Figura 53 Ejecución de modelado - Gráfico de resultados	52
Figura 54 Ejecución de controles - Gráfico de resultados	53

LISTA DE TABLAS

Tabla 1 Cálculo de latencia - Detección QR	43
Tabla 2 Cálculo de latencia - Detección Cámara-Objeto	44
Tabla 3 Cálculo de latencia - Control ON por Botón	46
Tabla 4 Cálculo de latencia - Control OFF por Botón	47
Tabla 5 Cálculo de latencia - Control ON por Gesto	48
Tabla 6 Cálculo de latencia - Control OFF por Gesto	50
Tabla 7 Tiempo de ejecución de detección QR	50

Tabla 8 Tiempo de ejecución de detección de objetos	50
Tabla 9 Tiempo de ejecución de control ON mediante Interfaz de Usuario	51
Tabla 10 Tiempo de ejecución de control ON mediante Detección de Gesto	51
Tabla 11 Tiempo de ejecución de control OFF mediante Interfaz de Usuario	51
Tabla 12 Tiempo de ejecución de control OFF mediante Detección de Gesto	52
Tabla 13 Análisis de resultados de ejecución	52
Tabla 14 Análisis de resultados de controles de dispositivos	53

INTRODUCCIÓN

Con el paso del tiempo los dispositivos de uso cotidiano como sensores, cámaras, comunicadores, entre otros, cuya funcionalidad se basaba en circuito cerrado han mejorado su funcionamiento para brindar mayor soporte para los usuarios [1]. Debido a la necesidad de mejorar un mejor acceso y administración de la información obtenida de los mismos, se implementó el uso de redes que interconectan los componentes físicos mediante el uso de internet [2].

Aprovechando estudios [3], [4], [5] se puede entender que su funcionamiento se basa principalmente en la operación remota. Lo cual se realiza mediante la asignación de direcciones IP a los dispositivos inalámbricos, por el cual se puede recibir instrucciones y al estar conectado a internet tiene la facilidad de comunicarse con un servidor externo y enviar los datos que ha recogido del entorno.

En base a este principio, domótica es de gran importancia en el proceso de automatización para los hogares, su importancia radica en mejorar la accesibilidad a un conjunto de elementos dentro de la vivienda a corta o largas distancias además de aportar un sistema de vigilancia y seguridad [6].

Por otra parte, la realidad aumentada busca mostrar al usuario y añadir información virtual al mundo real mediante el uso de un dispositivo tecnológico [7]. Sus ventajas entran en varios ámbitos tales como: Videojuegos, Visión de realidad aumentada, compras y ventas, seguridad e investigación judicial, publicidad, transporte e industria, entre otros. Brindando al usuario experiencias interactivas y que pueden manipular de acuerdo a sus gustos, pudiendo comunicarse con elementos intangibles mezclandolos con elementos físicos en tiempo real [8].

Aprovechando lo anteriormente mencionado se ha planteado la incorporación de un sistema domótico mediante el uso de la realidad aumentada, esto dará al usuario una mayor interactividad permitiendo un ambiente que mezcla la realidad con lo artificial dentro de nuestro campo visual.

Este trabajo se ha estructurado en cuatro capítulos para un mejor desarrollo, los cuales son:

Capítulo 1: Se da detalles correspondientes al trabajo, es decir, se define el marco teórico, se describe el problema y el objetivo que se espera lograr.

Capítulo 2: Se plantea el diseño del prototipo, explicando los diferentes aspectos necesarios para su funcionalidad.

Capítulo 3: Se detalla el proceso de instalación necesario para que se pueda implementar nuestro prototipo.

Capítulo 4: Se plantea el análisis de los resultados obtenidos tras haber instalado los dispositivos e implementado el prototipo.

1 CAPÍTULO I. MARCO TEÓRICO

1.1 Problema

El avance tecnológico ha dado a la humanidad un mejor estilo de vida, por lo cual el uso de diferentes dispositivos es cotidiano. El manejo de estos se ha hecho de manera que permita facilidad de uso a sus usuarios, esto lo hacen mediante internet generando y recibiendo información la cual debe ser administrada, para estos casos cuando nos referimos a dispositivos para el hogar es requerible que sea automatizada para facilitar ciertas tareas tales como encendido de luminarias, uso de sensores, e incluso prevención y seguridad.

Uno de los tópicos que está revolucionando la tecnología interactiva es la realidad aumentada, la cual brinda una visión de la información digital compartiendo espacio con información del mundo real a través de un dispositivo.

Tomando a consideración lo anterior es necesario realizar un control domótico que permita a los usuarios interactuar con elementos digitales y del mundo real a través de un dispositivo, presentando una aplicación que aporte con el desarrollo dentro del área de internet de las cosas.

1.2 Objetivo general

Implementar control domótico para IoT mediante el uso de asistente de voz y realidad aumentada.

1.3 Internet de las Cosas (IoT)

En la actualidad [9], [10], [11] se han dado muchos cambios en aspectos técnicos y tecnológicos en el desarrollo de la tecnología, tal que la mayor parte de los dispositivos pueden no solo responder a una orden manual del usuario también pueden responder a señales que se pueden enviar desde nuestros ordenadores a su vez como desde diferentes dispositivos móviles.

Por lo cual ha surgido una diversidad de plataformas IoT [12], [13], [14] que se encargan de registrar la información de los dispositivos inalámbricos, cuya aplicación muchas veces suelen concentrarse en medición y automatización de procesos industriales y residenciales. Además de enrutar datos en tiempo real y de controlar los dispositivos, las plataformas IoT brindan mayor facilidad de interacción con el usuario presentando una interfaz gráfica para acceder a procesamiento de datos y accionar de manera remota uno o varios dispositivos [15].

1.3.1 Plataformas IoT

Las plataformas IoT se caracterizan por brindar al usuario un ambiente de trabajo organizado, lo cual hace que las tareas de automatización, monitoreo y control de los

dispositivos inalámbricos sea fácil de manejar [16]. Entre ellos varios ofrecen diferentes especificaciones, en el medio los más implementados son:

1.3.1.1 OpenHab

Es una plataforma de código abierto para la automatización del hogar, debido a que es independiente de proveedores y la tecnología la comunidad se enfoca en agregar los dispositivos populares. Su arquitectura se basa en proporcionar compatibilidad a diferentes dispositivos domésticos [17].

1.3.1.2 Domoticz

Esta plataforma tiene como mejor propiedad que es un sistema ligero que permite supervisar y controlar diversos dispositivos, si se lo compara con Home Assistant y OpenHab. Tienen funciones que permiten ampliar su funcionalidad que se realiza principalmente mediante su interfaz web [18].

1.3.1.3 Home Assistant

Es una plataforma que tiene un enfoque en que debe presentarse lo más fácil posible de usar, con un gran crecimiento en aumento y al igual que domoticz se puede ampliar sus funciones mediante el uso de complementos que se encuentran disponibles dentro de la su sección de add-ons que es actualizada con mucha frecuencia por la comunidad [19].

1.4 Domótica

Consiste en sistemas que permiten el control y/o automatización de una vivienda para controlar la mayoría de los servicios que se relacionan con la energía [20]. Lo que hace es comunicar o vincular los dispositivos disponibles en casa dándonos como ventaja una facilidad de acceso de los diferentes dispositivos electrónicos de una manera eficiente y rápida [6].

Con las innovaciones y avances [21], [22], [23] se ha presentado como una oportunidad en que los hogares inteligentes no solo presenten a las personas un ambiente confortable también en uno seguro e integral, en donde mediante la red interconectada de sensores y dispositivos de monitoreo se puede establecer un área segura para los miembros residenciales.

1.5 Realidad Aumentada

A diferencia de la realidad virtual [8], [24], la realidad aumentada se basa en el concepto de establecer una combinación del mundo real y el virtual. Este proceso [25], se hace insertando un objeto virtual en el mundo físico, mediante el uso de la interfaz del ambiente real (Lo cual puede ser apoyado en una imagen o un objeto en específico) con el apoyo de la tecnología en tiempo real.

En varios estudios [26], [27], [28], [29] la realidad aumentada ha brindado una gran contribución en campos como la educación, en el cual permite al usuario tener una visión completa de aspectos técnicos de su entorno que mediante el aprendizaje tradicional es difícil de comprender e incluso de recordar. Otros ejemplos de estos en los estudios presentados [30], [31], [32] en donde se implementa la realidad aumentada como una alternativa para educar tanto a niños como a jóvenes, empleando como técnica un entorno favorable e interactivo que permita al estudiante a conocer las propiedades de los elementos que conforman su entorno. Presentando tecnologías, métodos, técnicas y estrategias que combinan la cognición de la experiencia física, completada con la percepción e integración del mundo real estimulados por la comprensión de conceptos y contenidos abstractos de un entorno virtual.

2 CAPÍTULO II. DISEÑO DE PROTOTIPO

2.1 Back-End

Para el desarrollo de la aplicación se tomó como plataforma IoT domótica a Home assistant (HASSIO), esto se debe a sus múltiples funciones que se pueden incorporar y la facilidad en lo que respecta a compatibilidad e integración de dispositivos inteligentes. Para la comunicación entre nuestro servidor y los dispositivos se planteó el uso del protocolo MQTT, el cual consiste en un sistema subscriber-publisher mediante un broker el cual recibe y distribuye los mensajes.

Como se puede apreciar en **Figura 1**, el proceso consiste en la integración de los dispositivos inteligentes al servidor de Home Assistant, el cual contará con la integración de mosquito MQTT broker que ayudará con la comunicación entre el cliente (Unity – Android).

Finalmente se implementa Node-Red para ejecutar los servicios correspondientes a los comandos recibidos mediante el protocolo MQTT.

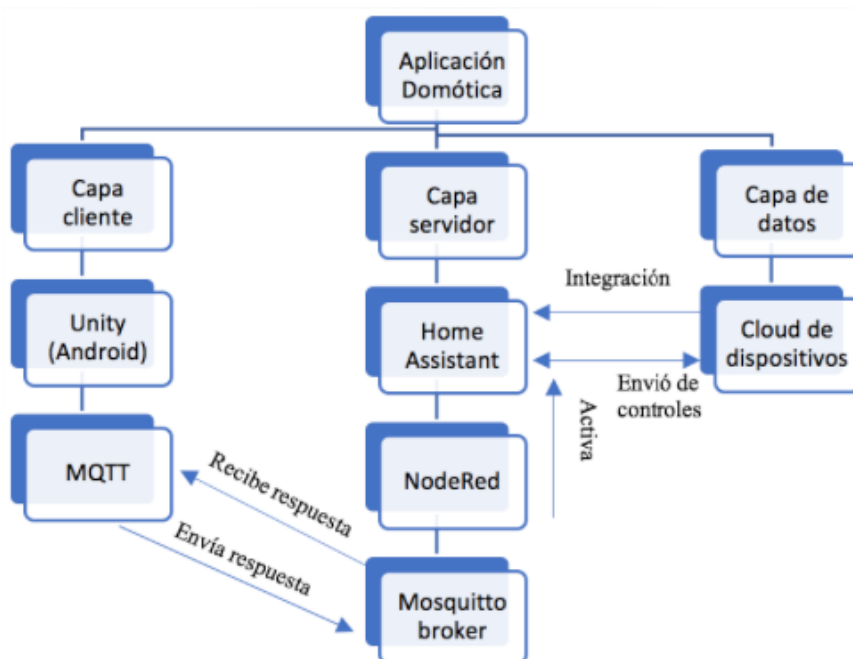


Figura 1 Back-End de aplicación domótica

Fuente: Elaboración Propia

2.2 Front_End

Se planteó una interfaz de usuario que permita el ingreso por pantalla del broker al que estará destinado la conexión MQTT representado en **Figura 2**. Lo cual permitirá que la aplicación pueda conectarse al servidor broker, mosquitto MQTT broker alojado en la raspberry pi 3.

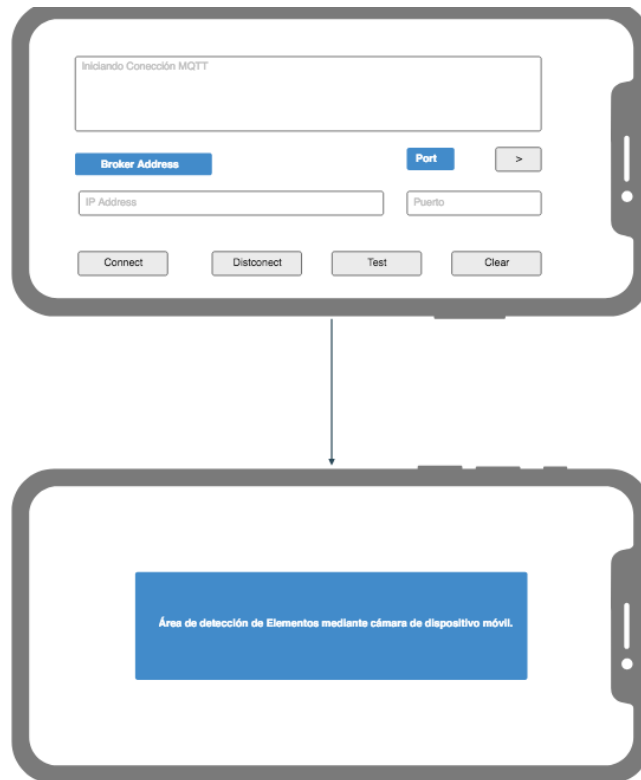


Figura 2 Prototipo de interfaz de usuario - conexión MQTT broker

Fuente: Elaboración Propia

Tras terminar el proceso de conexión al servidor MQTT se inicia la cámara del dispositivo, la cual procederá al proceso de detección como se puede apreciar en la **Figura 3**. Tras detectar un objetivo que identifique a un elemento, se mostrará en pantalla las opciones de usuario para dicho dispositivo, lo cual permitirá al usuario el acceso a sus funcionalidades disponibles.

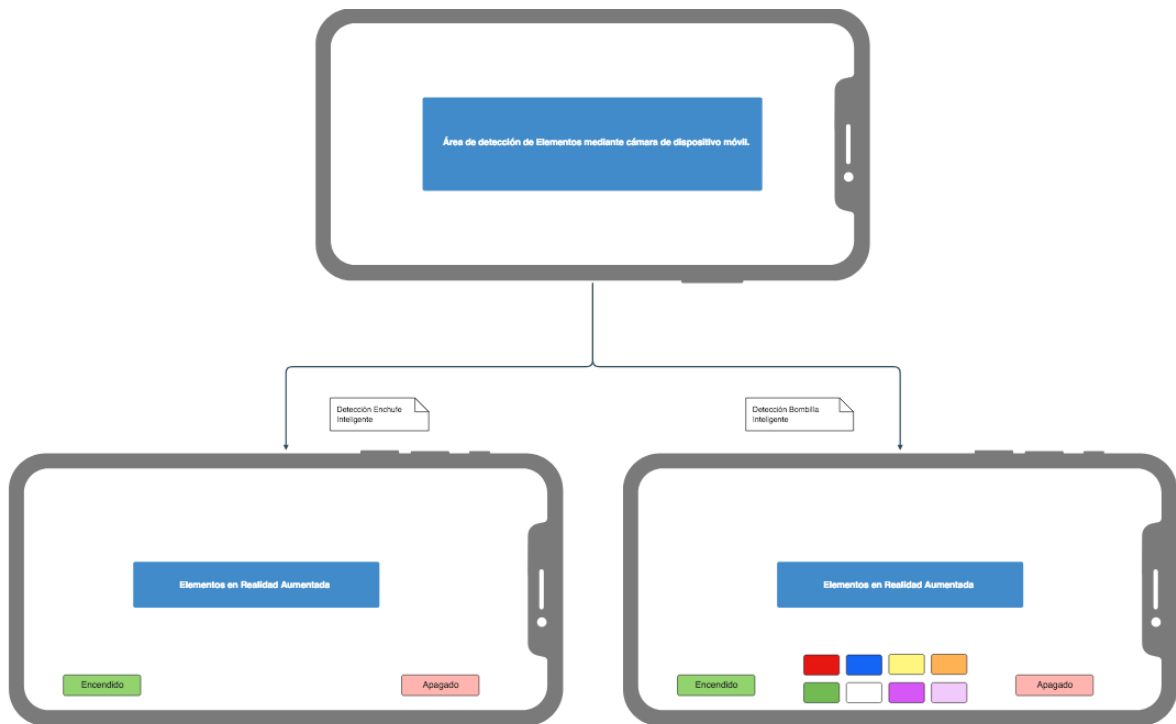


Figura 3 Prototipo de interfaz de usuario - Detección de elementos mediante cámara

Fuente: Elaboración Propia

2.3 Análisis de requerimientos

Se requiere un sistema domótico en realidad aumentada que permita el control de dispositivos inteligentes mediante el uso de una aplicación móvil. El control llevará a cabo acciones de encendido y apagado para los dispositivos.

El monitoreo se realiza en tiempo real mostrado por pantalla junto con el modelo en realidad aumentada, de esta manera actualizando también su estado actual.

3 CAPÍTULO III. IMPLEMENTACIÓN DE PROTOTIPO

3.1 Home Assistant – HASSIO

3.1.1 Instalación de HASSIO en Raspberry pi3 B+.

Como primer paso se debe dirigirse a la página principal de Home Assistant:

<https://www.home-assistant.io/>

- Nos dirigimos a getting started
- En la sección de Software Requeriments pulsamos en “your devices”
- A continuación, seleccionamos la versión necesaria para nuestro equipo

- Las opciones son:
 - Imágenes para dispositivos Raspberry
 - Imágenes virtuales
- Para lo cual seleccionamos Raspberry pi 3 b+ 32-bit

Mientras se descarga nuestra imagen del sistema Hassio descargamos en la página principal de hassio el software “balenaEtcher” que nos ayudará a flashear nuestra unidad SD.

INSTALACIÓN

- Coloque la tarjeta SD en su lector de tarjetas.
- Abra balenaEtcher, seleccione la imagen de Home Assistant y muévela a la tarjeta SD.
- Desmonte la tarjeta SD y extraigala de su lector de tarjetas.
- La configuración de red más confiable para su Raspberry Pi es conectarla con un cable Ethernet, sin embargo, si desea configurar Wi-Fi o una dirección IP estática (este paso requiere una memoria USB), puede probar esto:

En la raíz de la tarjeta SD después de instalar la imagen del sistema creamos el siguiente directorio: /CONFIG/network/my-network.

Siendo my-network el archivo donde guardaremos la configuración deseada la cual podemos encontrar ejemplos en el siguiente enlace: <https://github.com/home-assistant/operating-system/blob/dev/Documentation/network.md>.

El archivo my-network deberá guardarse como un archivo de texto sin una extensión.

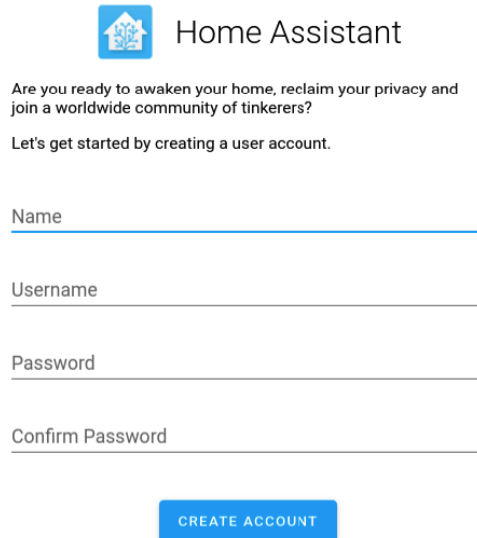
- Inserta la tarjeta SD en tu Raspberry Pi. Si va a utilizar un cable Ethernet, conéctelo también.
- Conecte su fuente de alimentación a la Raspberry Pi.
- La Raspberry Pi ahora se iniciará, se conectará a Internet y descargar la última versión de Home Assistant. Esto le llevará unos 20 minutos.
- Home Assistant estará disponible en <http://homeassistant.local:8123>. Si está ejecutando una versión anterior de Windows o tiene una configuración de red más estricta, es posible que deba acceder a Home Assistant en <http://homeassistant:8123>o <http://X.X.X.X:8123>(reemplácelo X.X.X.X con la

dirección IP de su Pi). Si usó una memoria USB para configurar la red, ahora puede quitarla.

PRIMER INICIO

- Ingrese un nombre, nombre de usuario, contraseña y haga clic en "crear cuenta".

Visto en la siguiente **Figura 4**.



The image shows the Home Assistant user registration interface. At the top, there is a blue square icon with a white house and a person, followed by the text "Home Assistant". Below this, a message reads: "Are you ready to awaken your home, reclaim your privacy and join a worldwide community of tinkerers? Let's get started by creating a user account." The form consists of four input fields: "Name", "Username", "Password", and "Confirm Password". Each field has a horizontal line for text entry. At the bottom of the form, there is a blue button with the text "CREATE ACCOUNT" in white capital letters.

Figura 4 Registro de usuario HASSIO

Fuente: Elaboración Propia

- A continuación, en la **Figura 5**, puede ingresar un nombre para su hogar y establecer su ubicación y sistema de unidad.

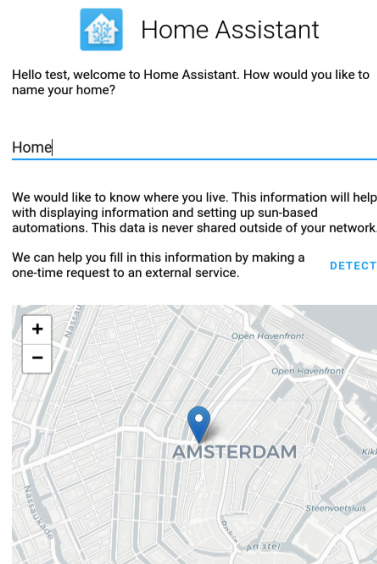


Figura 5 Configuración de localización de HASSIO

Fuente: Elaboración Propia

- Una vez que haya terminado, haga clic en Siguiente. En esta pantalla, Home Assistant mostrará cualquier dispositivo que haya descubierto en su red. Damos click en finish y el proceso termina.

3.1.2 Instalación de complementos de HASSIO para realizar configuraciones

Para realizar la instalación de complementos en hassio nos dirigimos a la sección de Supervisor y seleccionamos la pestaña de Add-on store, donde podremos seleccionar los complementos oficiales para nuestro Home Assistant, visto en la **Figura 6**.

Para mayor facilidad podemos utilizar el buscador para explorar las opciones disponibles.

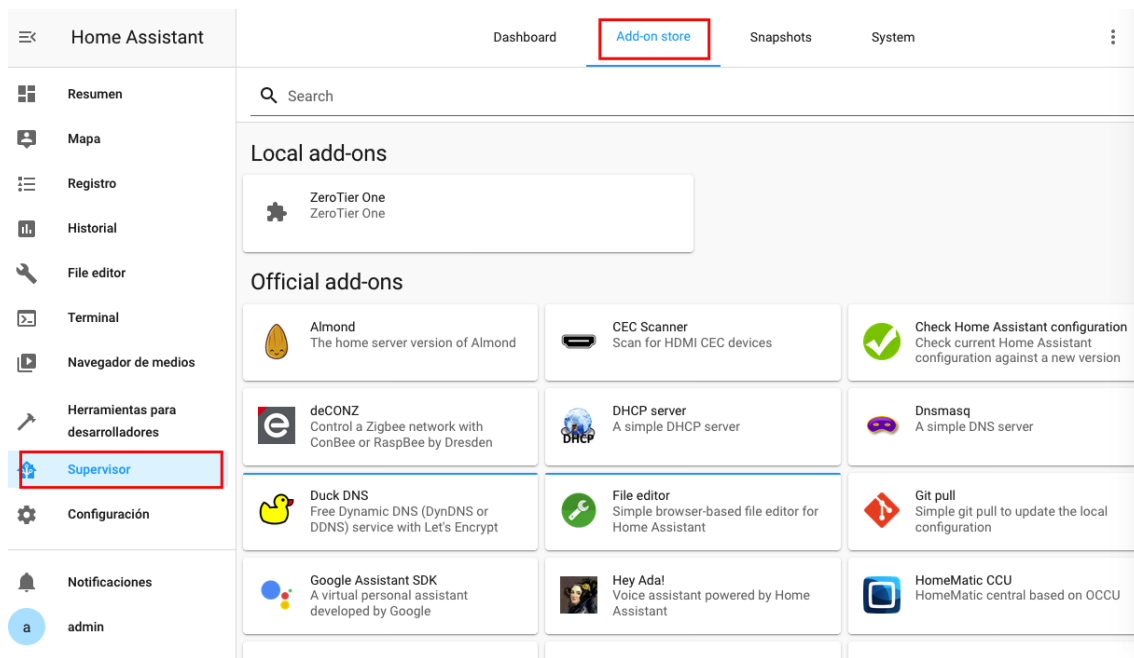


Figura 6 Add-ons de Home Assistant

Fuente: Elaboración Propia

El primero que se necesitará para realizar cambios en los archivos de configuración es File Editor, el cual en **Figura 7** podemos ubicar:

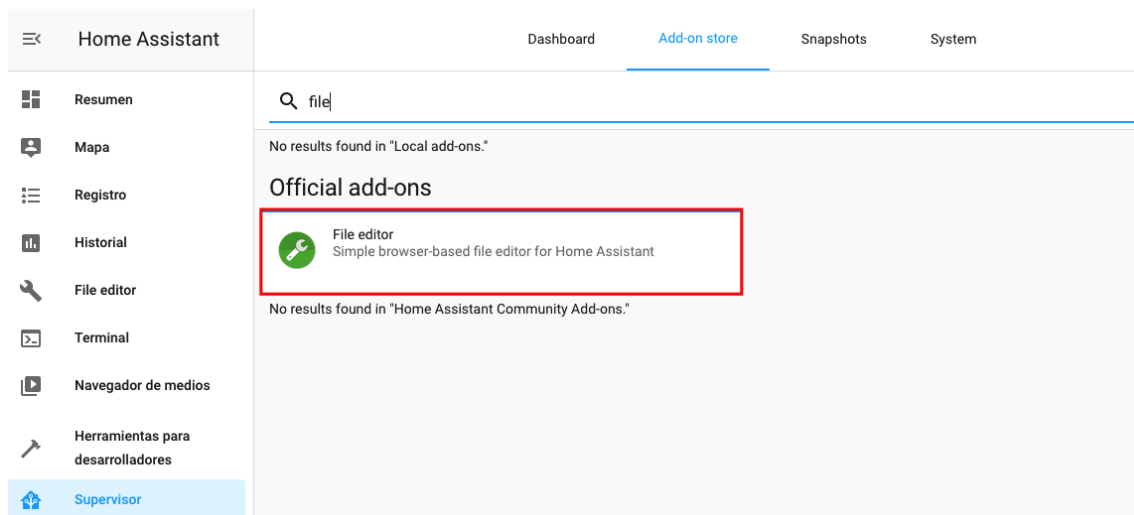


Figura 7 Búsqueda de File Editor en Add-ons

Fuente: Elaboración Propia

Como se aprecia en **Figura 8**, al seleccionarlo damos click en la sección de Install y tras terminar el proceso marcamos la sección de auto update para que el sistema actualiza automáticamente si se presenta una nueva versión y en show in sidebar para que se nos presente en la sección principal de Home Assistant.

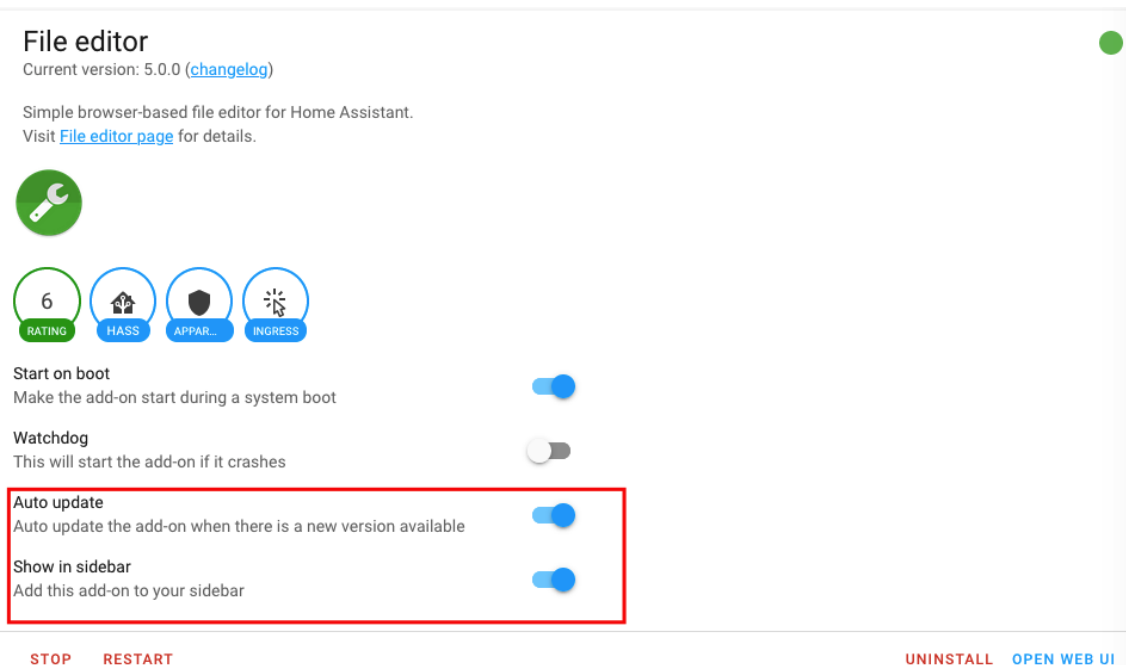


Figura 8 Menú de File Editor

Fuente: Elaboración Propia

En **Figura 9**, vemos otra opción necesaria para cambios en consola esta en la herramienta Terminal & SSH:

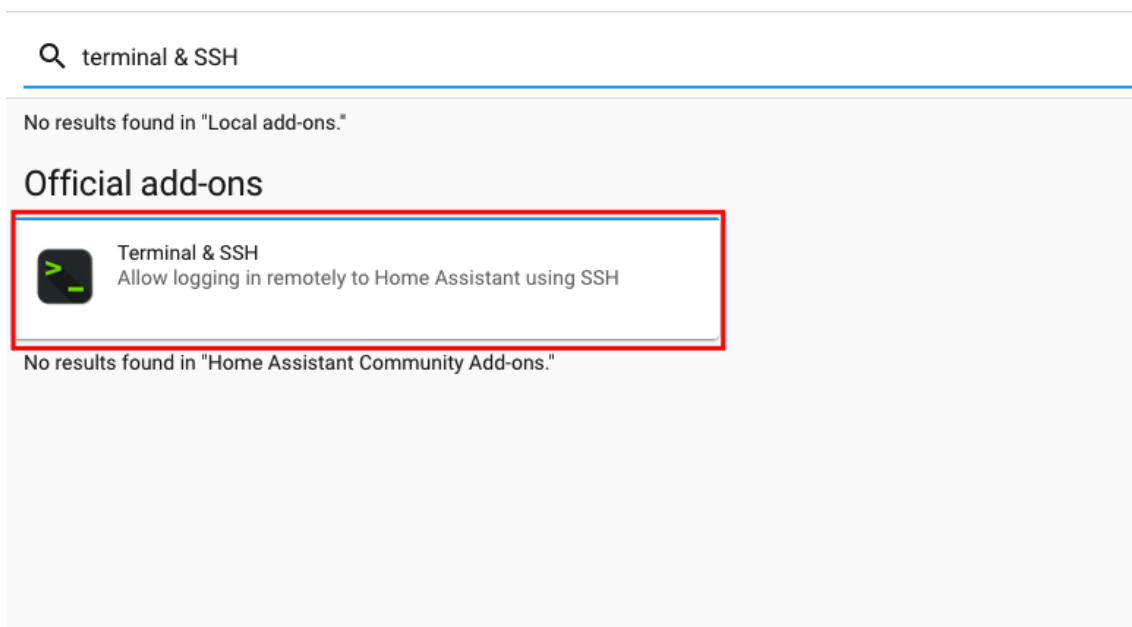


Figura 9 Búsqueda de Terminal en Add-ons

Fuente: Elaboración Propia

La cual realizamos el mismo proceso anterior **Figura 10**, para que se nos presente en la sección principal de Home Assistant e iniciamos el servicio para finalizar.

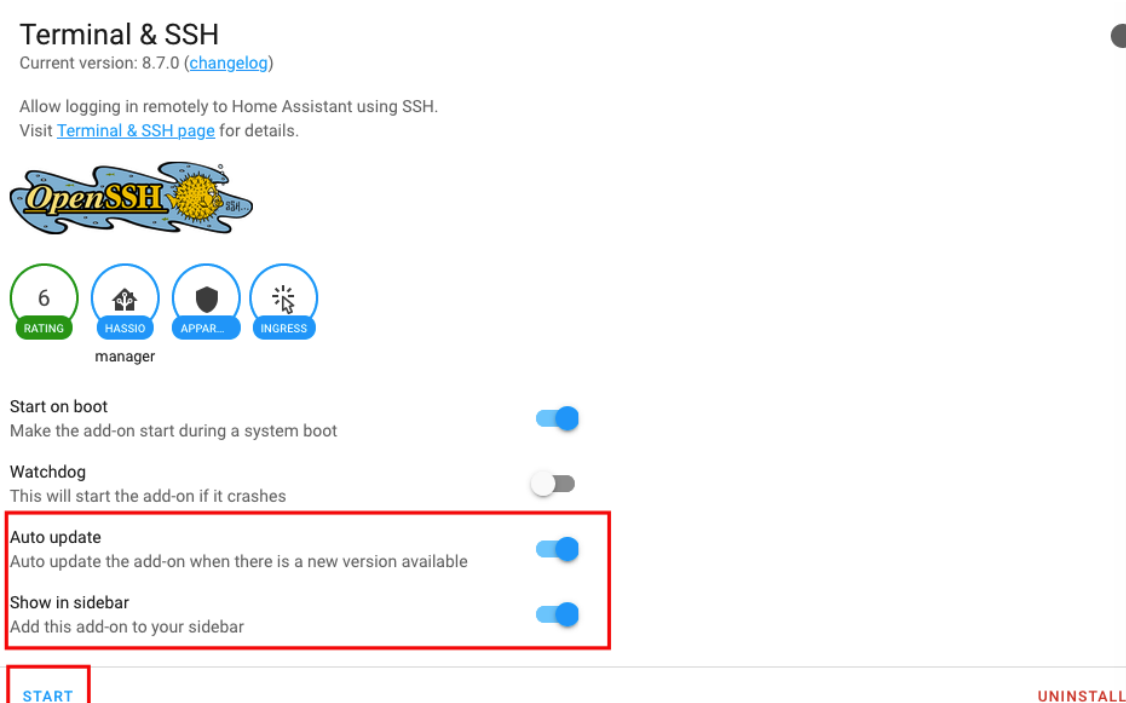


Figura 10 Menú de Terminal

Fuente: Elaboración Propia

Al hacerlo **Figura 11** podemos ingresar desde la sección principal de Home Assistant a la terminal del sistema.

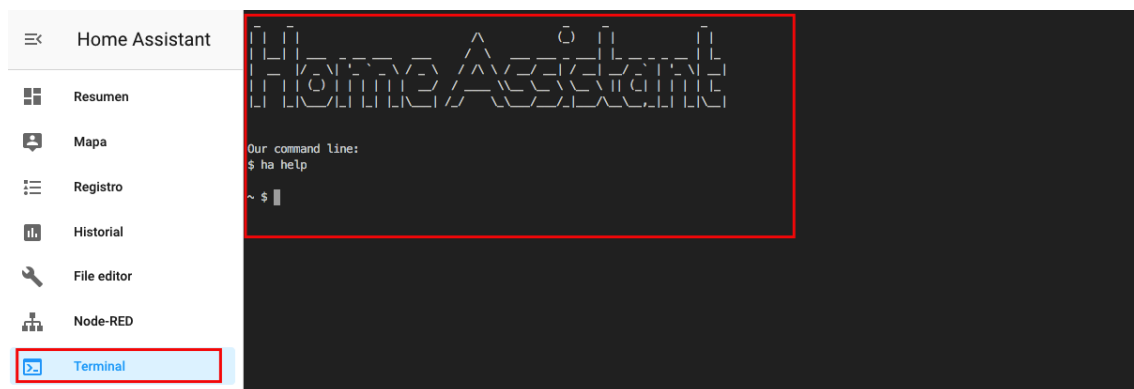


Figura 11 Ventana principal de Terminal de Home Assistant

Fuente: Elaboración Propia

En caso de que se presente algún problema con la configuración realizada a nuestro Hassio y no se nos permita el acceso mediante la configuración estándar que es “**IP asignada : puerto de acceso**”, entonces debemos mantener un canal de acceso seguro con nuestro ordenador que nos permita corregir el error.

Para ello se requiere la instalación de Samba share, Visto en **Figura 12**.

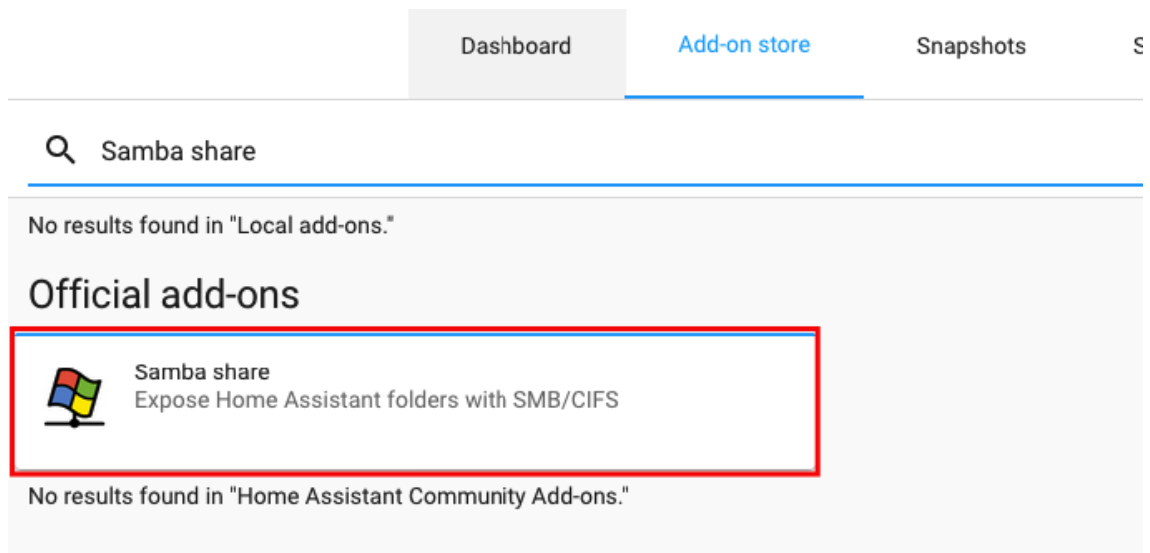


Figura 12 Búsqueda de Samba Share en Add-ons

Fuente: Elaboración Propia

En Figura 13 podemos ver la configuración necesaria para iniciar servicio de Samba Share. El cual nos permite ingresar al sistema de archivos de nuestra raspberry pi 3.

← Info Documentation **Configuration** Log

Samba share

Configuration

```
1 workgroup: WORKGROUP
2 username: homeassistant
3 password: null
4 interface: ''
5 allow_hosts:
6   - 10.0.0.0/8
7   - 172.16.0.0/12
8   - 192.168.0.0/16
9   - 'fe80::/10'
10 veto_files:
11   - .*
12   - .DS_Store
13   - Thumbs.db
14   - icon?
15   - .Trashes
16 compatibility_mode: false
17
```

RESET TO DEFAULTS **SAVE**

Figura 13 Configuración de Samba Share

Fuente: Elaboración Propia

Lo que debemos hacer visto en **Figura 14**, una vez instalado el complemento debemos ir a configuración y cambiar las opciones de usuario y contraseña para el acceso, una vez completado guardamos los cambios.

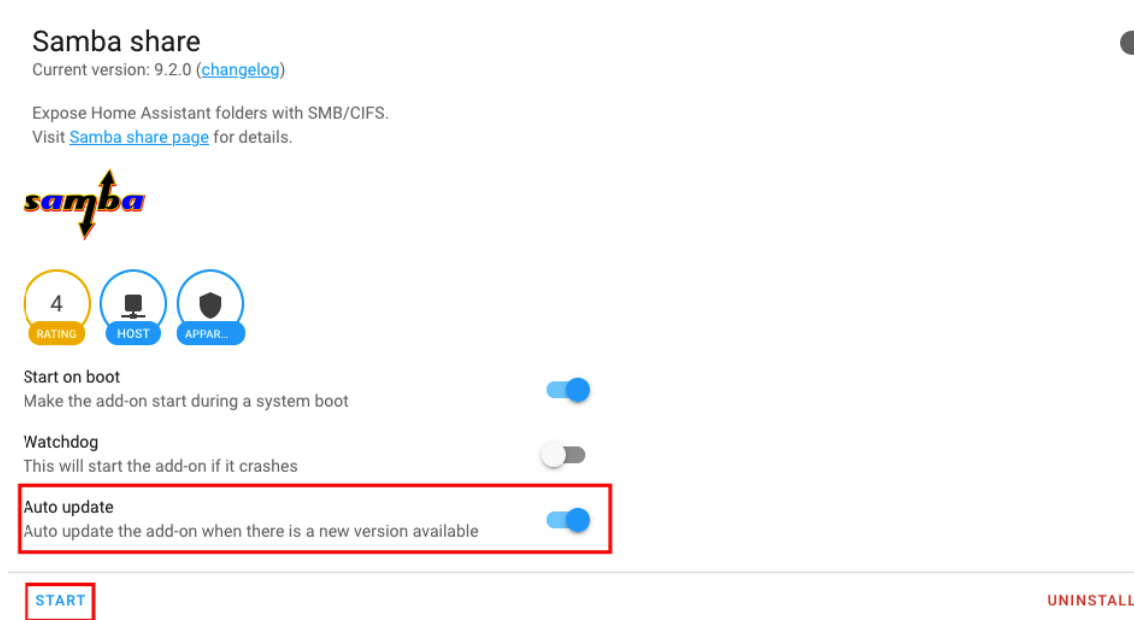


Figura 14 Menú de Samba Share

Fuente: Elaboración Propia

Al terminar regresamos a la pestaña de info e iniciamos el servicio, con lo cual ya podemos ingresar a nuestro sistema de archivos de Home Assistant desde nuestro ordenador.

3.2 Integración de dispositivos

Para la integración de dispositivos se realizó el método de tomar un servidor libre que en este caso pertenece a TUYA, el cual también cuenta dentro del catálogo soportado por Home Assistant. Existen otros métodos sin embargo se optó por este ya que tiene una mejor adaptabilidad con nuestro servidor domótico.

Para ello, tras instalar nuestros dispositivos inteligentes se los configura usando la app perteneciente a TUYA Smart las opciones pueden ser: Smart Life, Tuya Smart o Jinvo Smart.

En este caso se optó por usar Jinvo Smart, en la cual se agregaran los dispositivos estableciendo su instancia dentro del servidor de Tuya Smart.

Una vez realizado este paso procedemos dentro de las configuraciones de Home Assistant, en la sección de Integraciones y seleccionamos la opción para configurar una nueva integración **Figura 15**.

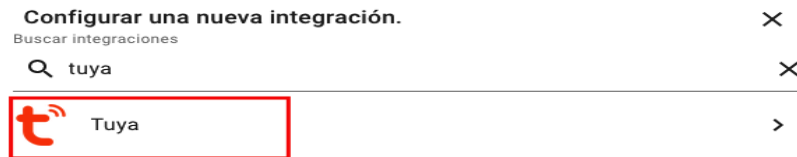


Figura 15 Configuración de nueva Integración en Home Assistant

Fuente: Elaboración Propia

Para configurar el acceso al servidor de Tuya **Figura 16**, debemos llenar los campos de: Usuario y contraseña el cual usamos para registrar nuestra cuenta. Finalmente seleccionamos la aplicación en la que se ha registrado la cuenta.

Figura 16 Ingreso de credenciales Tuya

Fuente: Elaboración Propia

3.3 Unity

3.3.1 Descarga de librerías de realidad aumentada Easyar para Unity

Se debe ingresar a la página principal de Easyar: <https://www.easyar.com/view/login.html>, donde se debe crear un login para su acceso. Seleccionamos la opción de "Download" y en la sección de Unity Plugin seleccionamos el enlace al archivo "EasyARSenseUnityPlugin.zip"

Tras terminar la descargar y descomprimir el archivo obtendremos lo siguiente **Figura 17**:

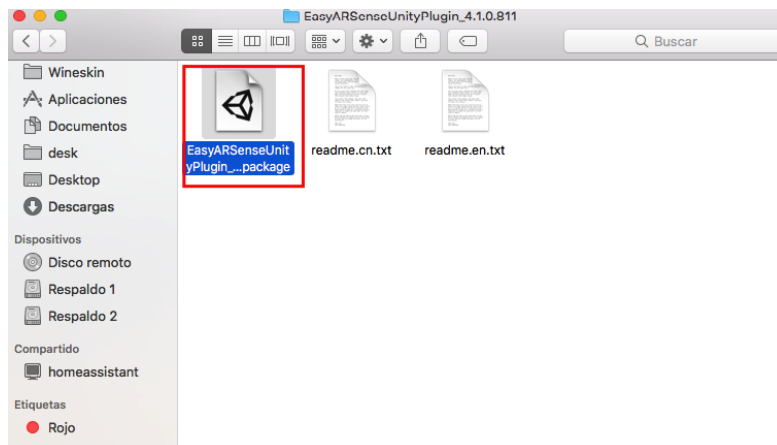


Figura 17 Plugin EasyAR

Fuente: Elaboración Propia

Lo cual ahora podemos usar para agregar las librerías necesarias e implementar Easyar en nuestro editor de Unity **Figura 18**. Para esto una vez abierto el editor damos doble click en el plugin y esto nos abrirá una ventana que facilita la importación de Easyar.

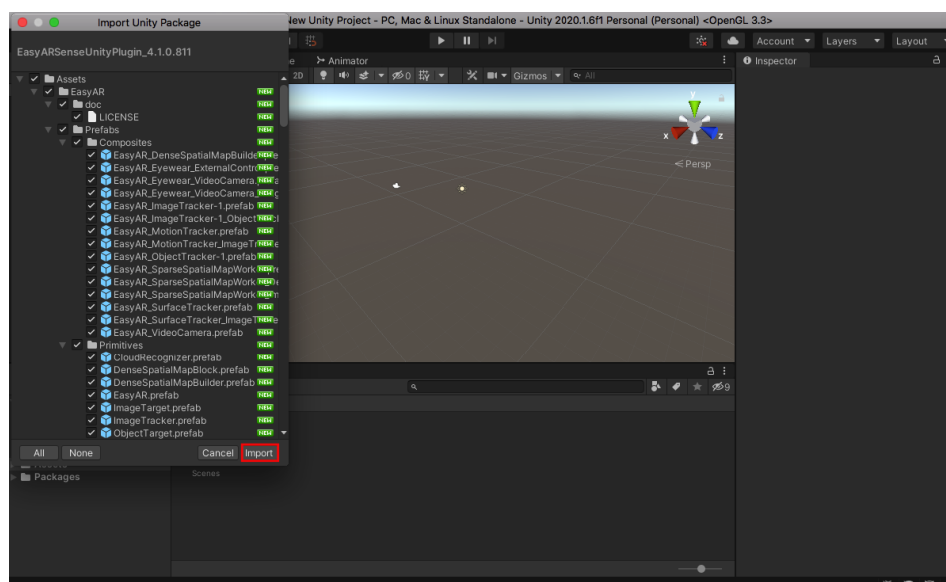


Figura 18 Importación EasyAR a Unity

Fuente: Elaboración Propia

Tras esto ahora se puede utilizar el complemento de EasyAr que nos permitirá establecer un objetivo para la aplicación de realidad aumentada. Y finalmente debemos ir a la página principal de EasyAr e ingresar a la sección de Develop Center.

Y creamos una Licence Key **Figura 19**, en la sección de nueva licencia seleccionamos la opción personal, en App details colocamos un nombre para nuestra licencia el cual puede ser a elección del usuario, en las secciones siguientes colocamos el nombre del paquete Android que usamos en nuestro editor Android para la aplicación en desarrollo.

Y finalmente antes de terminar el proceso en la sección de SpatialMap Database colocamos en la opción de no crear.

New Sense License Key

Type EasyAR Sense 4.0

[View Sense Feature Comparison](#)

☒ **EasyAR Sense 4.0 Personal**
Free, not for commercial use, with watermark, the number of SpatialMap download API request is limited to 100 times per day.

☐ EasyAR Sense 4.0 Professional
Monthly payment, no watermark, customized number of SpatialMap download API request.

☐ EasyAR Sense 4.0 Classic
One-time charge for lifetime use, no watermark, all features of Professional included.

Authorized Functions Selected by default, no modification supported

☒ Dense Spatial Map ☒ Motion Tracking ☒ 3D Object Tracking ☒ CRS Support

☒ Sparse Spatial Map ☒ Surface Tracking ☒ Planner Image Tracking ☒ Recording

Application Details

App Name
It can be modified.

Bundle ID
iOS
It can be modified. The Sense License Key of iOS platform needs to be used with the Bundle ID.

Package Name
Android
It can be modified. The Sense License Key of Android platform needs to be used with the Package Name.

SpatialMap Database: ☐ Create a new SpatialMap database ☒ Don't create temporarily [?](#)

Term(month): unlimited

Fee: \$0

[Confirm](#)

Figura 19 Registro de License Key de EasyAR

Fuente: Elaboración Propia

Al terminar ingresamos desde la sección principal de Develop Center a nuestra licencia y podemos copiar la cadena de caracteres que será necesaria en nuestro editor de unity para activar el plugin de Easyar **Figura 20**.

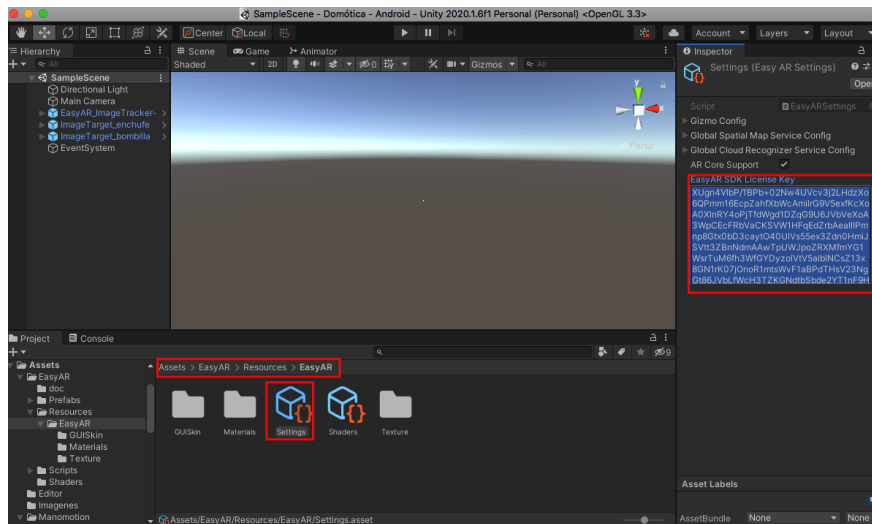


Figura 20 Agregación de Licencia de EasyAR en Unity

Fuente: Elaboración Propia

Ahora si deseamos agregar un target el primer paso es crear una carpeta nueva dentro de Assets llamada “StreamingAssets” lugar donde se alojan las imágenes referencia para nuestro objetivo. Hay que recordar que estas deberían contar con una buena definición y diferencias que permitan que Easyar pueda distinguirlo de manera correcta. Y para preparar el escenario arrastramos el siguiente prefab **Figura 21**.

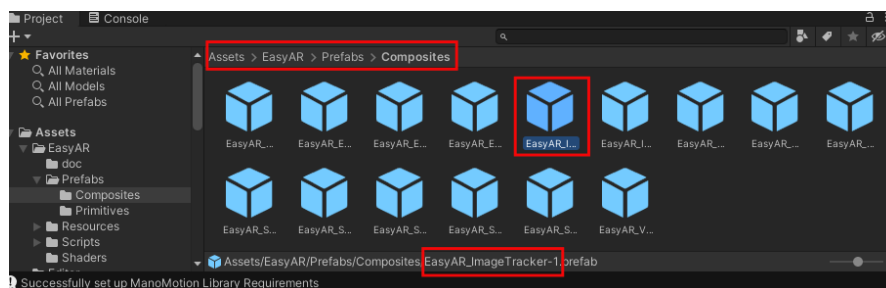


Figura 21 EasyAR Controlador

Fuente: Elaboración Propia

Y también arrastramos a la escena ImageTarget ubicado en **Figura 22**. Esto servirá para colocar el objetivo para la realidad aumentada el cual se puede implementar 1 o más de ser necesarios a excepción de EasyAR_ImageTracker el cual solo será necesario uno.

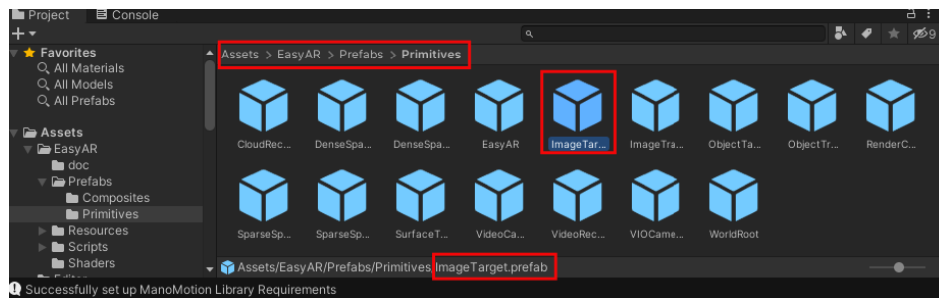


Figura 22 EasyAR ImageTarget

Fuente: Elaboración Propia

Ahora con nuestro objetivo en escena lo que queda es asignar una imagen **Figura 23**, para lo cual en la sección de Image File Source seleccionamos como Path Type la carpeta de StreamingAssets, en Path colocamos el nombre de uno de los archivos contenido incluido su extensión e inmediatamente la imagen se proyectará en el objetivo de realidad aumentada.

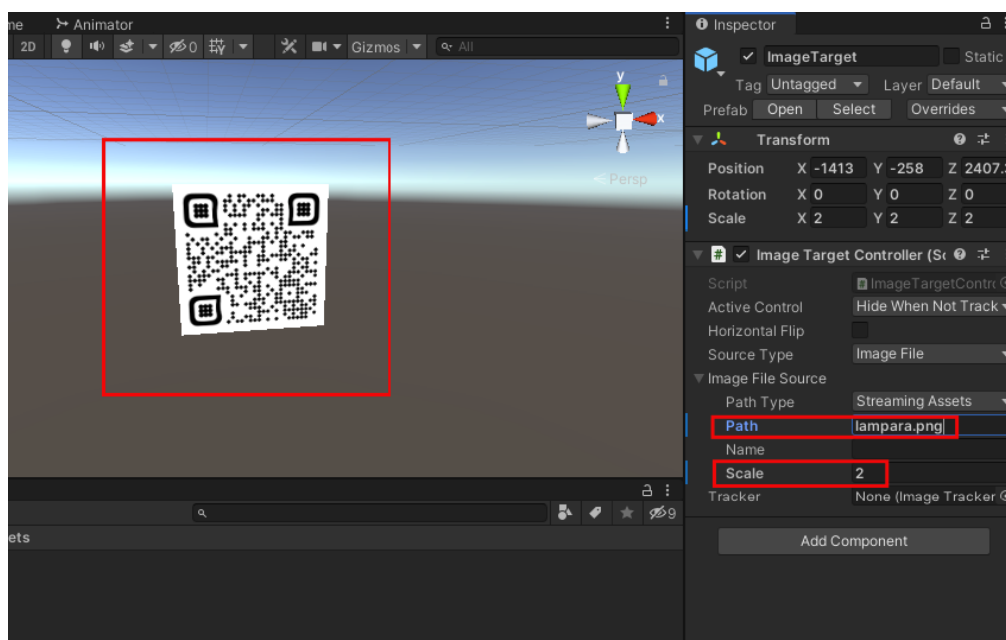


Figura 23 Implementación de ImageTarget

Fuente: Elaboración Propia

3.4 Ejecución de prototipo

Una vez que los dispositivos se encuentren integrados en la plataforma de Home Assistant se puede proceder en la sección de resumen y editar un panel para acceder a sus funcionalidades luciendo como se aprecia en la **Figura 24**.



Figura 24 Panel de Control de dispositivos - HASSIO

Fuente: Elaboración Propia

Tras cumplir con lo requerido, el funcionamiento de nuestro prototipo se encargará de realizar conexión con el servidor MQTT visto en **Figura 25**. Tras culminar este proceso presionamos el botón que se muestra a continuación para activar la cámara e iniciar el reconocimiento de objetivos.

Node-Red permite al usuario un sistema completo para el diseño estructural de controles específicos mediante un medio visual intuitivo para usuarios principiantes y profesionales.

Cuenta con estructuras de servidor, funciones, red, secuenciales, almacenamiento, plataforma local entre otros. Que facilitan y permiten una amplia gama de diseño de controles para nuestros dispositivos.

Para este caso se requería una estructura de control que permita el envío del estado actual de nuestros dispositivos mediante MQTT y a su vez que reciba nuevas instrucciones de origen MQTT para realizar cambios en el estado de los dispositivos, tales como encender o apagar.

Para el primero se implementó la estructura de control de home assistant, trigger:state que permite algunas funciones avanzadas en torno a los casos de uso de automatización comunes, es decir se rige mediante una premisa de condicionales en las que dado el caso nos enviara de regreso un estado de la entidad establecida.

Properties

Server: Home Assistant

Entity ID: switch.eb63c76ba554950261etuu Exact

State Type: String

Conditions

This Entity

Current State

is a_z on

+ add

Custom outputs

Default Msg

Send Always

is

Figura 25 Configuración Trigger State

Fuente: Elaboración Propia

En este caso como se puede apreciar se establece como servidor base Home Assistant del cual vendrá el estado del dispositivo que se determina en Entity ID (este se especifica en Home Assistant tras la integración del mismo), en el cual se especifica que si el dispositivo Switch al estar en estado encendido (on) nos dará como resultado un mensaje que lo especifica y al estar en default el resultado será "on".

No hay que olvidar llenar la casilla de (Output on Connect) ya que esto facilitará que se envíe el mensaje de estado al conectarse el servidor.

Para finalizar necesitamos que este mensaje sea enviado mediante MQTT para que el cliente pueda receptor el estado actual del dispositivo, usando "mqtt out" en la sección de network para este fin ya que se conecta a un intermediario MQTT y publica mensajes

Figura 26.

Properties

Server: server

Topic: enchufe/estado/

QoS: 0 Retain: ☐

Name: ESTADO

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

Figura 26 Configuración de nodo MQTT OUT

Fuente: Elaboración Propia

Especificamos el servidor MQTT y el topic para publicar el mensaje de estado del dispositivo actual. Hay que recordar que este servidor debe ser configurado antes de culminar de establecer parámetros **Figura 27**.

Connection | Security | Messages

Server: 192.168.1.9 Port: 1883

☐ Enable secure (SSL/TLS) connection

Client ID: Leave blank for auto generated

Figura 27 Configuración de servidor

Fuente: Elaboración Propia

Se ingresa en la sección de connection el dominio del servidor que en este caso es la dirección estática de nuestra raspberry pi 3 B, y terminamos con añadir en la sección de security el usuario y contraseña de acceso a nuestro broker MQTT, que en este caso es homeassistant como se especificó durante la integración de mosquitto bróker **Figura 28**.

Connection	Security	Messages
<div> <div> <div></div> <div>Username</div> </div> <div> <div>homeassistant</div> </div> </div>		
<div> <div> <div></div> <div>Password</div> </div> <div> <div>.....</div> </div> </div>		

Figura 28 Configuración de servidor homeassistant

Fuente: Elaboración Propia

Al finalizar tenemos lo siguiente el cual es un disparador que permite el envío del estado actual de nuestro dispositivo Switch, visto en **Figura 29**.

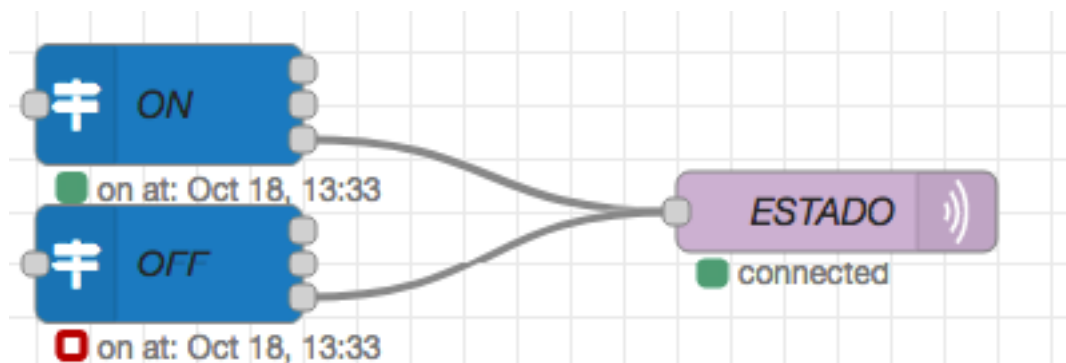


Figura 29 Disparador de estado actual de dispositivo

Fuente: Elaboración Propia

Finalmente, para el segundo caso en donde se requerirá que se recepte un mensaje enviado desde un cliente y se verifique el tipo de control que es y envíe un comando que active un servicio de Home Assistant.

Se requiere la estructura “mqtt in” **Figura 30**, ubicado en la sección de network el cual permite conectarse a un MQTT broker y suscribirse a mensajes de un topic específico.

Properties

Server: server

Topic: enchufe/estado/

QoS: 0

Output: auto-detect (string or buffer)

Name: ESTADO

Figura 30 Configuración de nodo MQTT IN

Fuente: Elaboración Propia

Ahora el servidor ya obtiene el mensaje del cliente con el correspondiente comando de dispositivo, para hacer enrutar mensajes en función de sus valores de propiedad o posición de secuencia es decir que se verifique el tipo de control y se envíe el correspondiente se implementó la función de “Switch”, como se puede observar en **Figura 31**.

Properties

Name: Name

Property: msg. payload

Rule 1: a_z on → 1

Rule 2: a_z off → 2

Figura 31 Configuración de nodo Switch

Fuente: Elaboración Propia

La cual al recibir el mensaje MQTT creará una ruta **Figura 32**, dependiendo si el comando recibido en este caso encendido (on) o apagado (off):



Figura 32 Enrutamiento de nodo Switch

Fuente: Elaboración Propia

Ahora solo debemos usar las rutas 1 y 2 para vincularlas con el correspondiente servicio que se requiere en nuestro Home Assistant, para lo cual nos ubicamos en la sección de home assistant y usamos la estructura “call service” **Figura 33**, para llamar al correspondiente servicio de nuestro dispositivo.

Name	on
Server	Home Assistant
Domain	switch
Service	turn_on
Entity Id	switch.eb63c76ba554950261etuu
Data	<div>▼ {} {"entity_id":"switch.eb63c76ba55495026 ...</div> <div><input type="checkbox"/> Use alternative template tags for the Data field</div>
Merge Context	lightOptions
Output Location	▼ None

Figura 33 Configuración de nodo Call service

Fuente: Elaboración Propia

Para este caso especificaremos el servicio para Switch on, donde se especifica el servidor que es Home Assistant, el dominio el cual es el tipo de servicio y dispositivo que es Switch, en service el tipo de servicio requerido que es “turn_on”, en entity_id la

entidad objetivo del cambio de estado y finalmente en data se especifica mediante el formato JSON el acceso de medios y de ser requerido las propiedades dependiendo estas del dominio, dispositivo y servicio especificado **Figura 34**. (Este comando es el que afecta directamente a nuestro dispositivo vinculado al servidor Home Assistant).

Service: switch.turn_on

Turn a switch on.

Property	Desc	Example
entity_id	Name(s) of entities to turn on	switch.living_room

Figura 34 Propiedades de servicio ON para dispositivo switch

Fuente: Elaboración Propia

En **Figura 35**, como se especifica en el ejemplo en la parte inferior no hay ninguna propiedad adicional para nuestra línea de comando JSON por lo cual solo se incluye la propiedad entity_id. (En caso de no tener conocimientos del manejo de comandos en JSON se pueden generar plantillas en la sección de Herramientas para desarrolladores y sección de servicios).

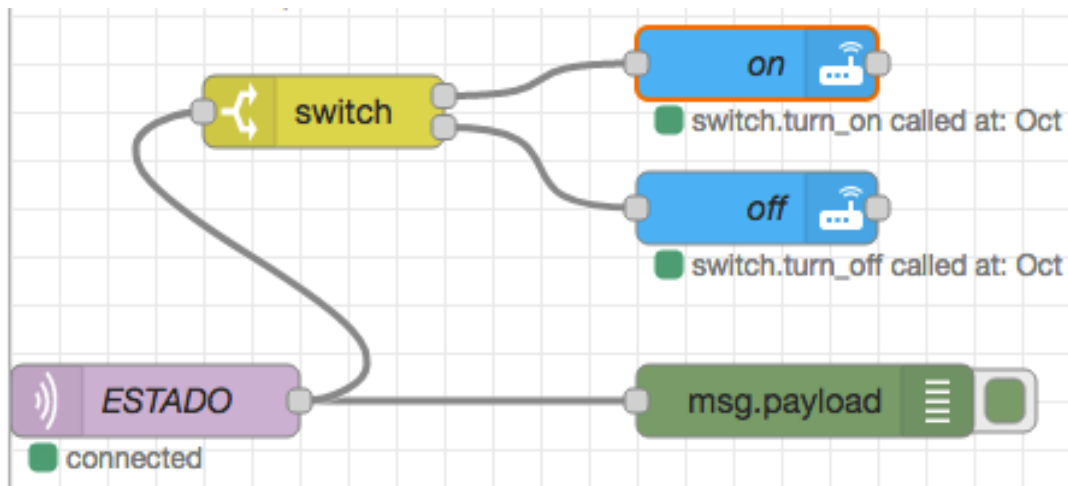


Figura 35 Estructura de control para recepción de comando y activación de servicio

Fuente: Elaboración Propia

Se repite el mismo proceso para el comando de apagado “off” y se enruta el servicio a la correspondiente salida especificada previamente establecida en “Switch”. Dándonos como resultado una estructura de control que recibe los mensajes enviados a nuestro servidor los lee para realizar la correspondiente validación y finalmente activan el correspondiente servicio de dispositivo. (la estructura en verde es denominada “debug”

ubicada en la sección de common esta muestra las propiedades de mensaje seleccionadas en la pestaña de la barra lateral de depuración y, opcionalmente, en el registro en tiempo de ejecución. De forma predeterminada, muestra msg.payload, pero se puede configurar para mostrar cualquier propiedad, el mensaje completo o el resultado de una expresión JSONata.)

Tras cumplir con lo requerido, el funcionamiento de nuestro prototipo se encargará de realizar conexión con el servidor MQTT visto en **Figura 36**. Tras culminar este proceso presionamos el botón que se muestra a continuación para activar la cámara e iniciar el reconocimiento de objetivos.

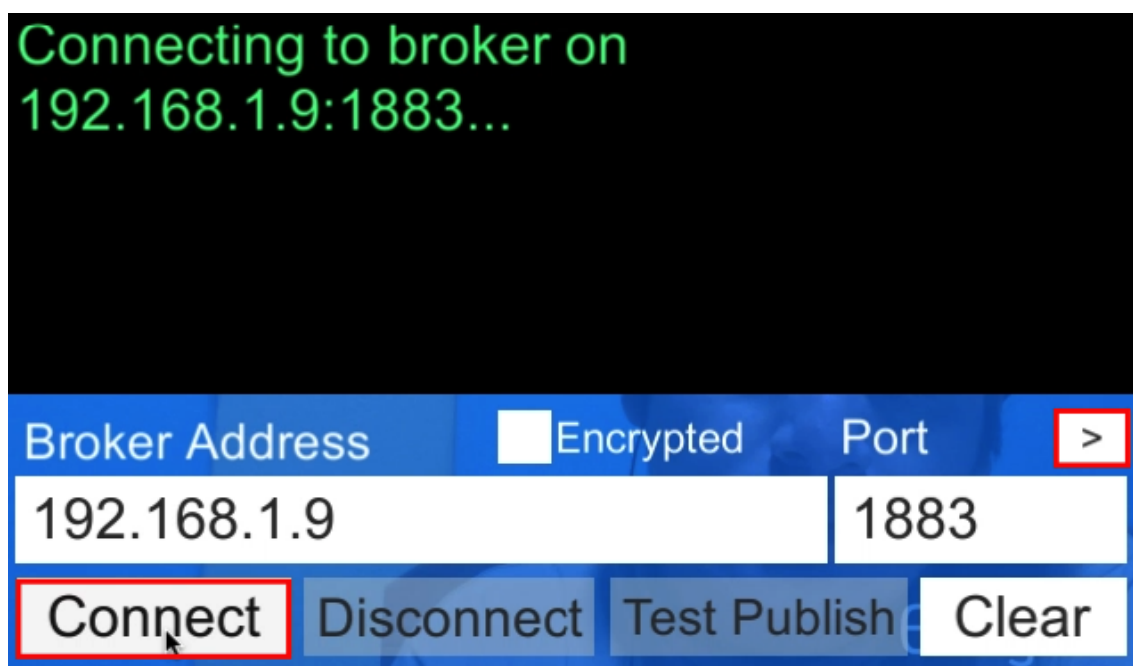


Figura 36 Conexión con Broker MQTT – Aplicación en Android

Fuente: Elaboración Propia

Una vez que la cámara reconoce **Figura 37**, **Figura 38**, mostrará por pantalla los elementos en realidad aumentada y sus opciones de dispositivo. En donde los elementos mostraran su estado mediante cambios sutiles en su diseño.



Figura 37 Reconocimiento QR Enchufe

Fuente: Elaboración Propia

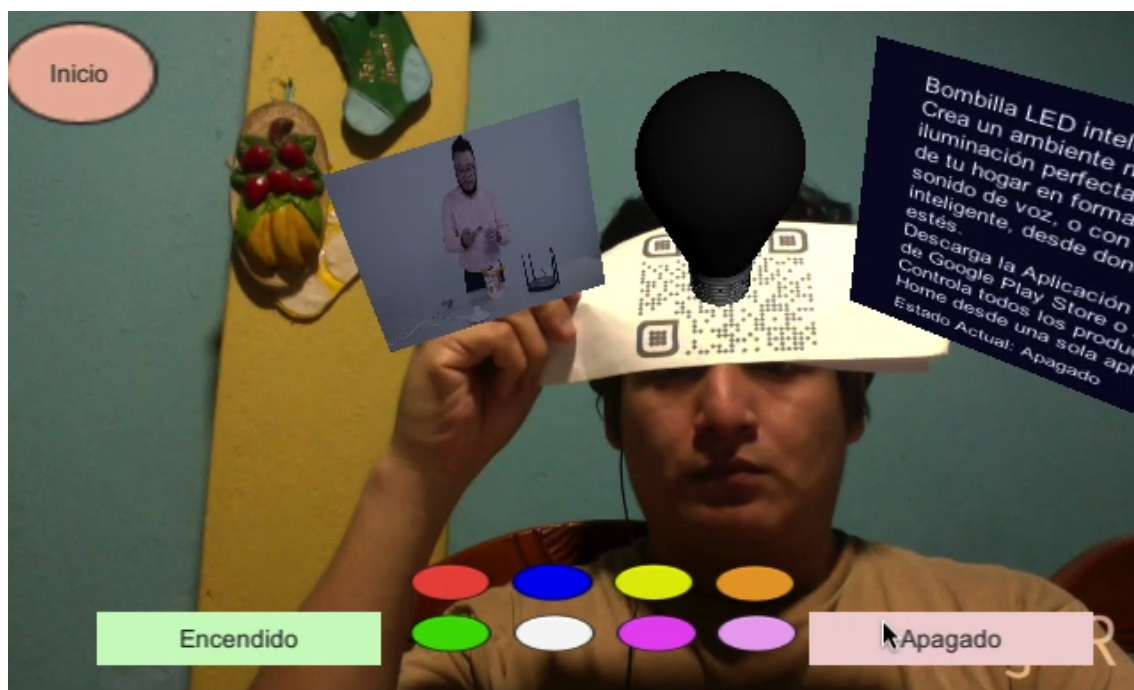


Figura 38 Reconocimiento QR Bombilla

Fuente: Elaboración Propia

4 CAPÍTULO IV. EXPERIMENTOS

4.1 Experimentos de detección

4.1.1 Cálculo de latencia

Para obtener un valor que ayude a medir la capacidad del tiempo de respuesta de la aplicación se obtuvo de la siguiente manera, **Figura 39** se muestra el tiempo transcurrido al iniciar la aplicación y al cargar el modelo 3D.

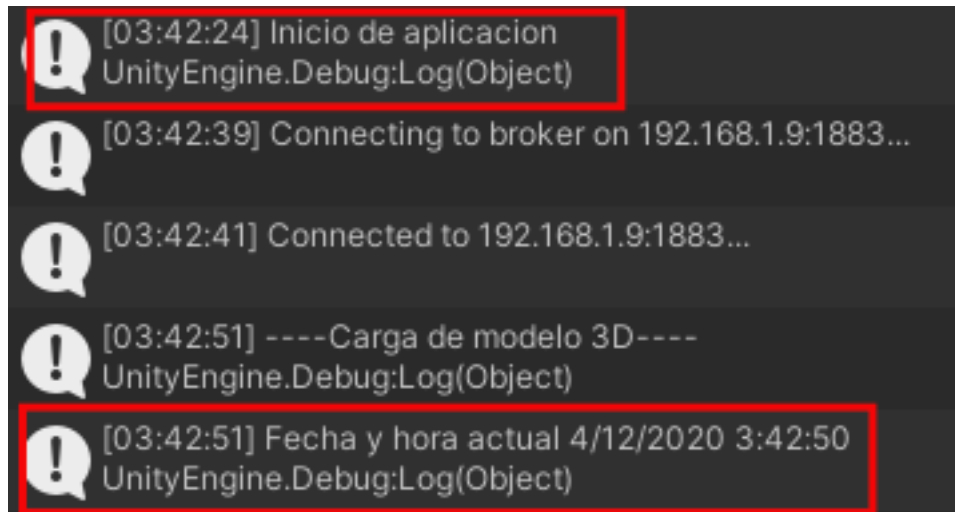


Figura 39 Consola - Detección QR

Fuente: Elaboración Propia

Entonces al hacer un Cálculo diferencial entre tiempo inicial y tiempo final se obtiene como se indica en la **Tabla 1**, el resultado de 27 segundos de tiempo necesario para cargar el modelo 3D mediante el método de detección QR.

Tabla 1 Cálculo de latencia - Detección QR

Inicio de aplicación	03:42:24
Carga de modelo	03:42:51
Tiempo de carga de modelo 3D	00:00:27

Fuente: Elaboración Propia

Repetimos este proceso durante la ejecución de detección de objeto, **Figura 40** en donde se obtiene el tiempo inicial y final tras la ejecución de la aplicación.

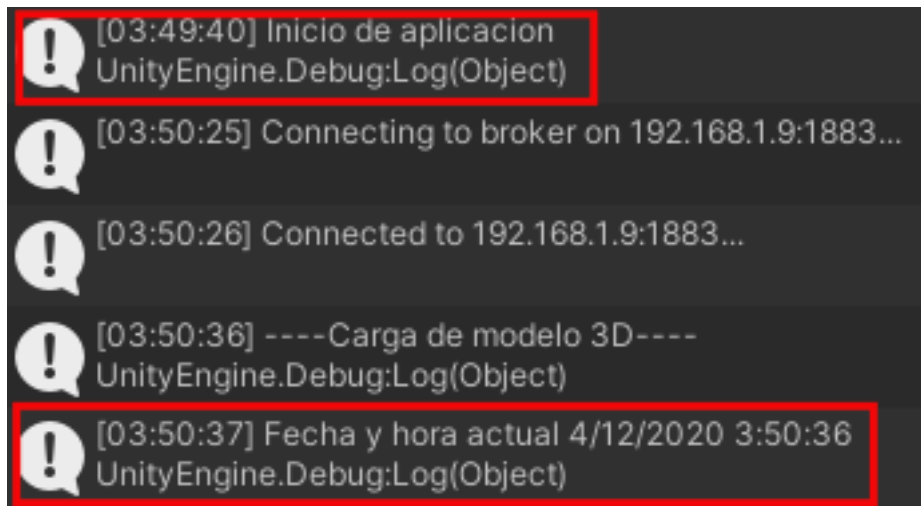


Figura 40 Cálculo de latencia - Detección Cámara-Objeto

Fuente: Elaboración Propia

Entonces al hacer un Cálculo diferencial entre tiempo inicial y tiempo final se obtiene como se indica en la **Tabla 2**, el resultado de 56 segundos de tiempo necesario para cargar el modelo 3D mediante el método de detección objeto.

Tabla 2 Cálculo de latencia - Detección Cámara-Objeto

Inicio de aplicación	03:49:40
Carga de modelo	03:50:36
Tiempo de carga de modelo 3D	00:00:56

Fuente: Elaboración Propia

Al culminar se realizó el estudio de latencia de los controles de dispositivo como se aprecia en **Figura 41**, el comando ON mediante la opción de UI.

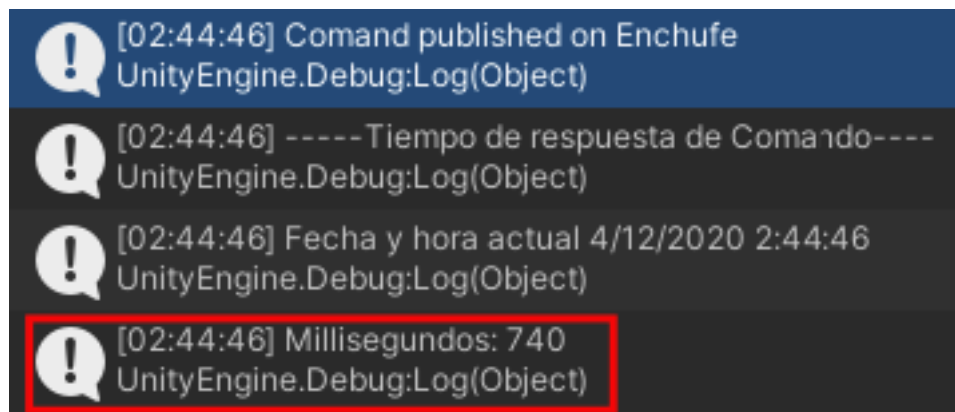


Figura 41 Consola - Control ON por Botón

Fuente: Elaboración Propia

Para su análisis se obtuvo la respuesta de Node-Red en **Figura 42**, y mediante Unity **Figura 43** se midió el tiempo en milisegundos en completarse la tarea de encender el dispositivo.

```
4/12/2020 2:44:47 node: 1faba771.1b1051
enchufe/estado/out/ : msg.payload : string[2]
"on"
```

Figura 42 Consola HASSIO - Estado ON

Fuente: Elaboración Propia

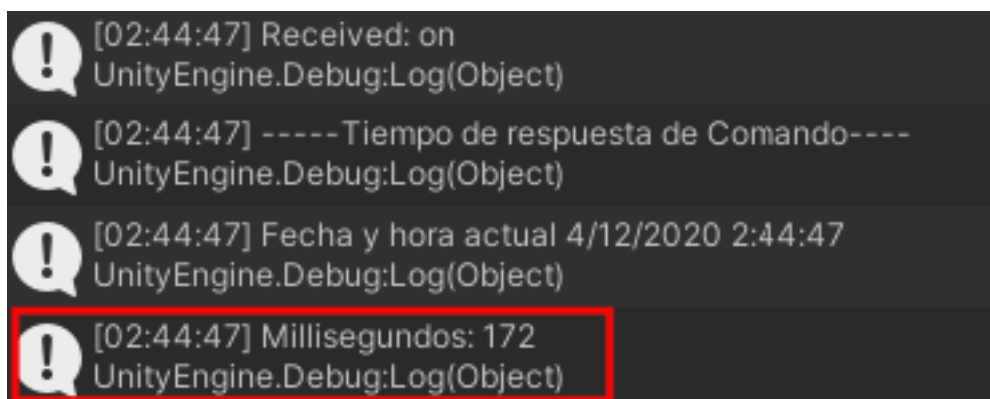


Figura 43 Consola - Tiempo de respuesta HASSIO

Fuente: Elaboración Propia

Dándonos el tiempo inicial y final presentados en la **Tabla 3**, permitiendo obtener el tiempo total de ejecución del comando ON mediante UI.

Tabla 3 Cálculo de latencia - Control ON por Botón

Envío de comando (ms)	740
Recepción de comando (Node-Red ms)	172
Tiempo total de ejecución de comando(ms)	912

Fuente: Elaboración Propia

Para continuar se realizó el estudio de latencia de los controles de dispositivo como se aprecia en **Figura 44**, el comando OFF mediante la opción de UI.

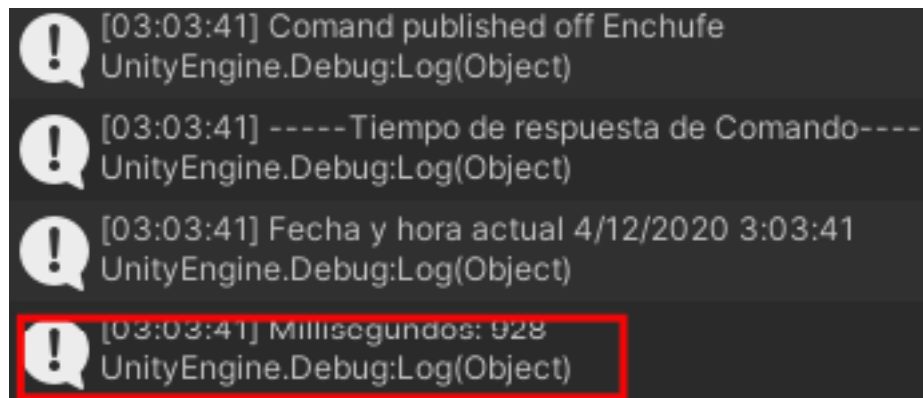


Figura 44 Consola - Control OFF por Botón

Fuente: Elaboración Propia

Para su análisis se obtuvo la respuesta de Node-Red en **Figura 45**, y mediante Unity **Figura 46** se midió el tiempo en milisegundos en completarse la tarea de encender el dispositivo.

```
4/12/2020 3:03:42 node: 1faba771.1b1051
enchufe/estado/out/ : msg.payload : string[3]
"off"
```

Figura 45 Consola HASSIO - Estado OFF

Fuente: Elaboración Propia

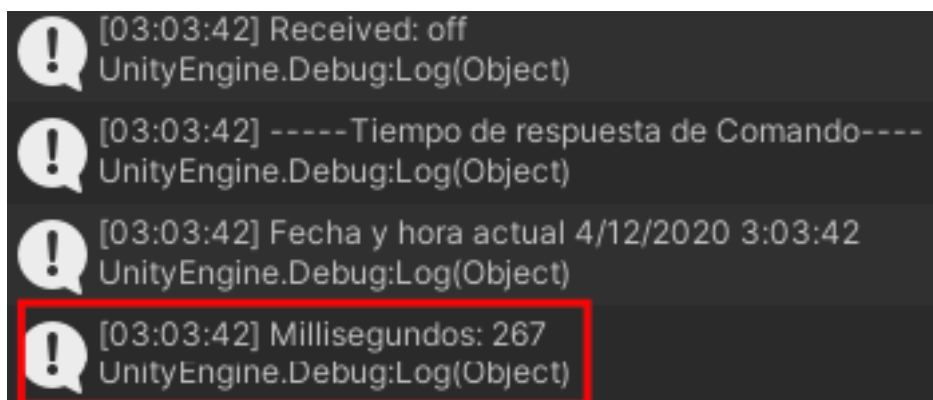


Figura 46 Consola - Tiempo de respuesta HASSIO

Fuente: Elaboración Propia

Dándonos el tiempo inicial y final presentados en la **Tabla 4**, permitiendo obtener el tiempo total de ejecución del comando OFF mediante UI.

Tabla 4 Cálculo de latencia - Control OFF por Botón

Envío de comando (ms)	928
Recepción de comando (Node-Red ms)	267
Tiempo total de ejecución de comando(ms)	1195

Fuente: Elaboración Propia

Para continuar se realizó el estudio de latencia de los controles de dispositivo como se aprecia en **Figura 47**, el comando ON mediante la opción de reconocimiento de gesto.

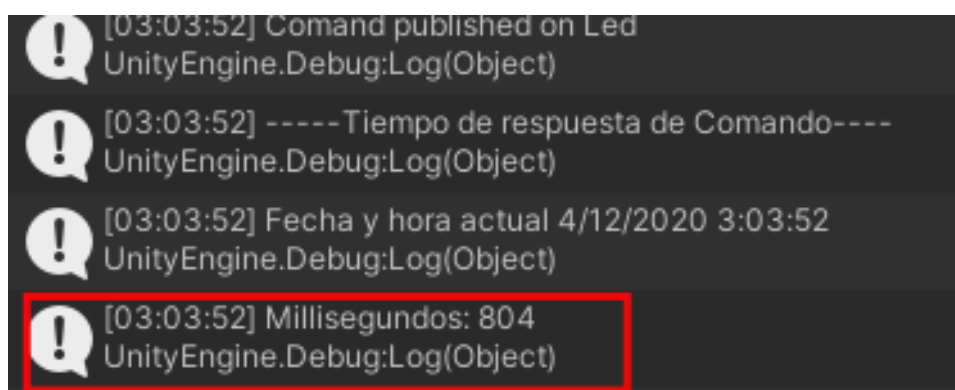


Figura 47 Consola - Control ON por Gesto

Fuente: Elaboración Propia

Para su análisis se obtuvo la respuesta de Node-Red en **Figura 48**, y mediante Unity **Figura 49** se midió el tiempo en milisegundos en completarse la tarea de encender el dispositivo.

```
4/12/2020 3:03:53 node: 4c248a91.3d43'
led/estado/out/ : msg.payload : string[2]
"on"
```

Figura 48 Consola HASSIO - Estado ON Gesto

Fuente: Elaboración Propia

```
[03:03:53] Received: on
UnityEngine.Debug:Log(Object)
[03:03:53] -----Tiempo de respuesta de Comando-----
UnityEngine.Debug:Log(Object)
[03:03:53] Fecha y hora actual 4/12/2020 3:03:53
UnityEngine.Debug:Log(Object)
[03:03:53] Millisegundos: 244
UnityEngine.Debug:Log(Object)
```

Figura 49 Consola - Tiempo de respuesta HASSIO

Fuente: Elaboración Propia

Dándonos el tiempo inicial y final presentados en la **Tabla 5**, permitiendo obtener el tiempo total de ejecución del comando ON mediante detección de gesto.

Tabla 5 Cálculo de latencia - Control ON por Gesto

Envío de comando (ms)	804
Recepción de comando (Node-Red ms)	224
Tiempo total de ejecución de comando(ms)	1028

Fuente: Elaboración Propia

finalmente se realizó el estudio de latencia de los controles de dispositivo como se aprecia en **Figura 50**, el comando OFF mediante la opción de detección de gesto.

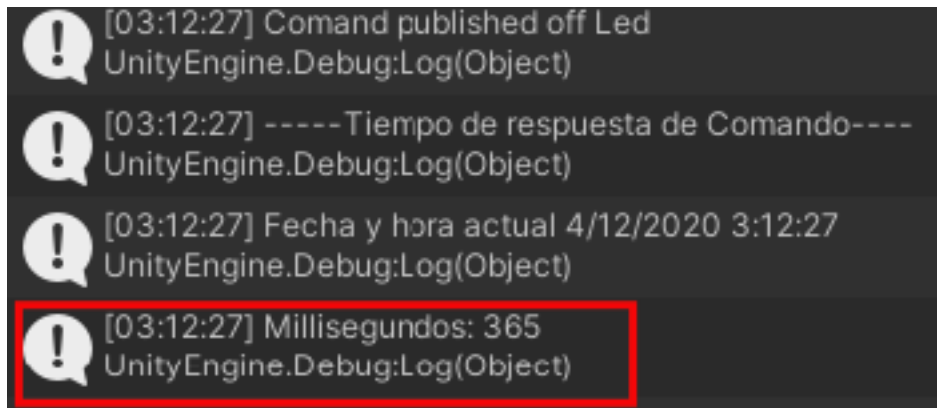


Figura 50 Consola - Control OFF por Gesto

Fuente: Elaboración Propia

Para su análisis se obtuvo la respuesta de Node-Red en **Figura 51**, y mediante Unity **Figura 52** se midió el tiempo en milisegundos en completarse la tarea de encender el dispositivo.

```

4/12/2020 3:12:27 node: 4c248a91.3d431c
led/estado/out/ : msg.payload : string[3]
"off"

```

Figura 51 Consola HASSIO - Control OFF por Gesto

Fuente: Elaboración Propia

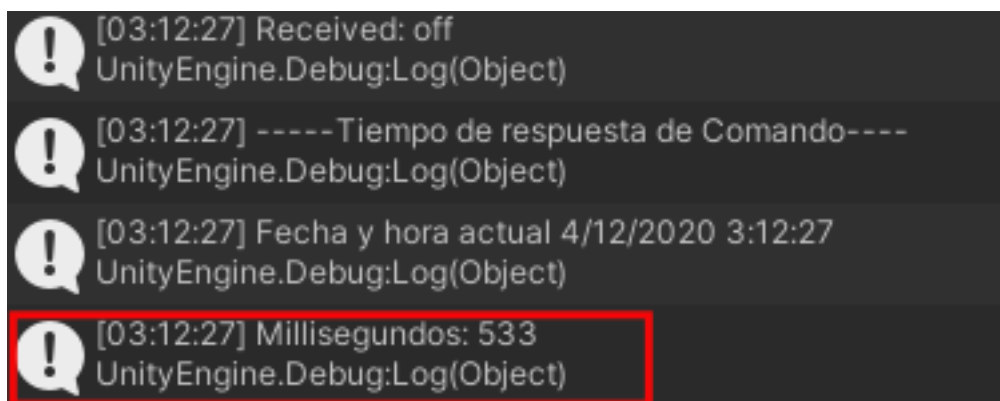


Figura 52 Consola - Tiempo de respuesta HASSIO

Fuente: Elaboración Propia

Dándonos el tiempo inicial y final presentados en la **Tabla 6**, permitiendo obtener el tiempo total de ejecución del comando OFF mediante detección de gesto.

Tabla 6 Cálculo de latencia - Control OFF por Gesto

Envío de comando (ms)	365
Recepción de comando (Node-Red ms)	533
Tiempo total de ejecución de comando(ms)	898

Fuente: Elaboración Propia

4.1.2 Objetivo QR

Para el experimento de detección QR se implementó un código QR el cual, al ser detectado por la cámara del dispositivo móvil, mostrará el correspondiente panel de control en realidad aumentada.

En este proceso se realizó un estudio de latencia mediante la toma de 100 ejecuciones en tiempo real, obteniéndose en **Tabla 7** el tiempo de ejecución de detección QR promedio.

Tabla 7 Tiempo de ejecución de detección QR

BOMBILLA INTELIGENTE	ENCHUFE INTELIGENTE
00:00:59	00:00:54

Fuente: Elaboración Propia

4.1.3 Objetivo Cámara (Object detection)

Para la detección de cámara se estableció como objetivo un elemento en este caso el dispositivo correspondiente, el cual al ser detectado mediante la cámara de nuestro dispositivo móvil se podrá ver el modelo en realidad aumentada del mismo.

Obteniendo como resultante en **Tabla 8** el tiempo de ejecución de detección de objetos promedio, en este proceso se realizó un estudio de latencia mediante la toma de 100 ejecuciones en tiempo real.

Tabla 8 Tiempo de ejecución de detección de objetos

BOMBILLA INTELIGENTE	ENCHUFE INTELIGENTE
00:00:52	00:00:46

Fuente: Elaboración Propia

4.2 Control ON

4.2.1 Interfaz de Usuario – Botón

Tras la detección de un elemento se puede visualizar en la pantalla de nuestro dispositivo móvil los correspondientes controles de dispositivo y al presionar el botón de encendido envía el comando de ON, lo que se estudia a continuación en **Tabla 9** es la latencia ocurrida tras el envío de un control a nuestro servidor domótico.

Tabla 9 Tiempo de ejecución de control ON mediante Interfaz de Usuario

BOMBILLA INTELIGENTE	ENCHUFE INTELIGENTE
1035,47 ms	967,76 ms

Fuente: Elaboración Propia

4.2.2 Detección de Gesto – Hand Detection

Tras la detección de un elemento se puede visualizar en la pantalla de nuestro dispositivo móvil los correspondientes controles de dispositivo, dándonos la opción de detectar gestos mediante el uso de Hand Detection en este caso al mostrar frente a cámara la palma de mano envía el comando de ON, lo que se estudia a continuación en **Tabla 10** es la latencia ocurrida tras el envío de un control a nuestro servidor domótico.

Tabla 10 Tiempo de ejecución de control ON mediante Detección de Gesto

BOMBILLA INTELIGENTE	ENCHUFE INTELIGENTE
1035,72 ms	1023,23 ms

Fuente: Elaboración Propia

4.3 Control OFF

4.3.1 Interfaz de Usuario – Botón

Tras la detección de un elemento se puede visualizar en la pantalla de nuestro dispositivo móvil los correspondientes controles de dispositivo y al presionar el botón de apagado envía el comando de OFF, lo que se estudia a continuación en **Tabla 11** es la latencia ocurrida tras el envío de un control a nuestro servidor domótico.

Tabla 11 Tiempo de ejecución de control OFF mediante Interfaz de Usuario

BOMBILLA INTELIGENTE	ENCHUFE INTELIGENTE
1025,45 ms	1051,82 ms

Fuente: Elaboración Propia

4.3.2 Detección de Gesto – Hand Detection

Tras la detección de un elemento se puede visualizar en la pantalla de nuestro dispositivo móvil los correspondientes controles de dispositivo, dándonos la opción de detectar gestos mediante el uso de Hand Detection en este caso al mostrar frente a cámara el puño envía el comando de OFF, lo que se estudia a continuación en **Tabla 12** es la latencia ocurrida tras el envío de un control a nuestro servidor domótico.

Tabla 12 Tiempo de ejecución de control OFF mediante Detección de Gesto

BOMBILLA INTELIGENTE	ENCHUFE INTELIGENTE
1025,27 ms	998,76 ms

Fuente: Elaboración Propia

4.4 Análisis de resultados

Mediante los datos obtenidos de la muestra de 100 ejecuciones de nuestra aplicación en realidad aumentada, se presentó un promedio que se refleja en la **Tabla 13** y **Tabla 14**. Donde se reflejan un tiempo de respuesta para el proceso de detección entre 46 a 59 segundos, esto se da debido a el enfoque de la cámara del dispositivo en el objetivo de reconocimiento.

Tabla 13 Análisis de resultados de ejecución

DETECCIÓN	BOMBILLA INTELIGENTE	ENCHUFE INTELIGENTE
Detección QR	00:00:59	00:00:54
Detección de Objetos	00:00:52	00:00:46

Fuente: Elaboración Propia

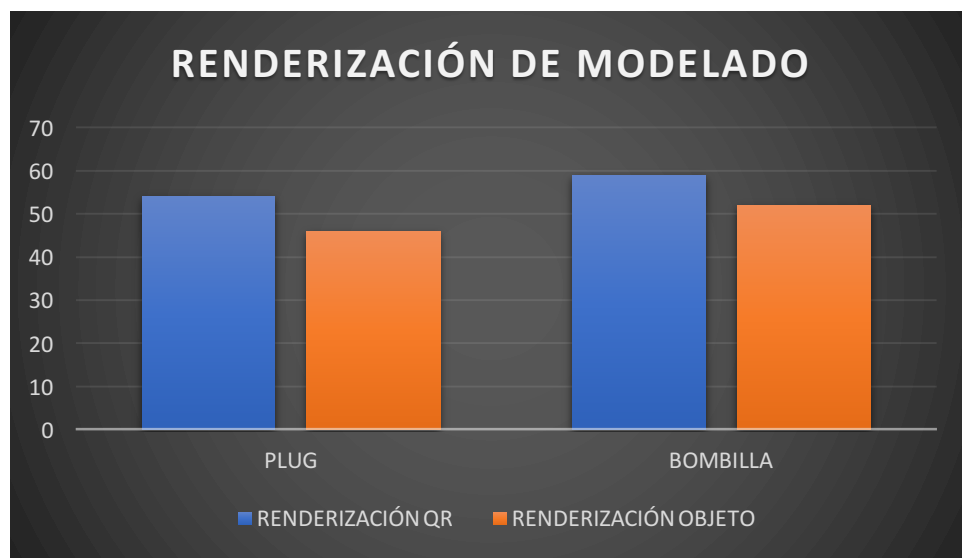


Figura 53 Ejecución de modelado - Gráfico de resultados

Fuente: Elaboración Propia

Tabla 14 Análisis de resultados de controles de dispositivos

CONTROL	BOMBILLA INTELIGENTE	ENCHUFE INTELIGENTE
Control de encendido mediante UI	1035,47 ms	967,76 ms
Control de apagado mediante UI	1025,45 ms	1051,82 ms
Control de encendido mediante detección de gestos	1035,72 ms	1023,23 ms
Control de apagado mediante detección de gestos	1025,27 ms	998,76 ms

Fuente: Elaboración Propia

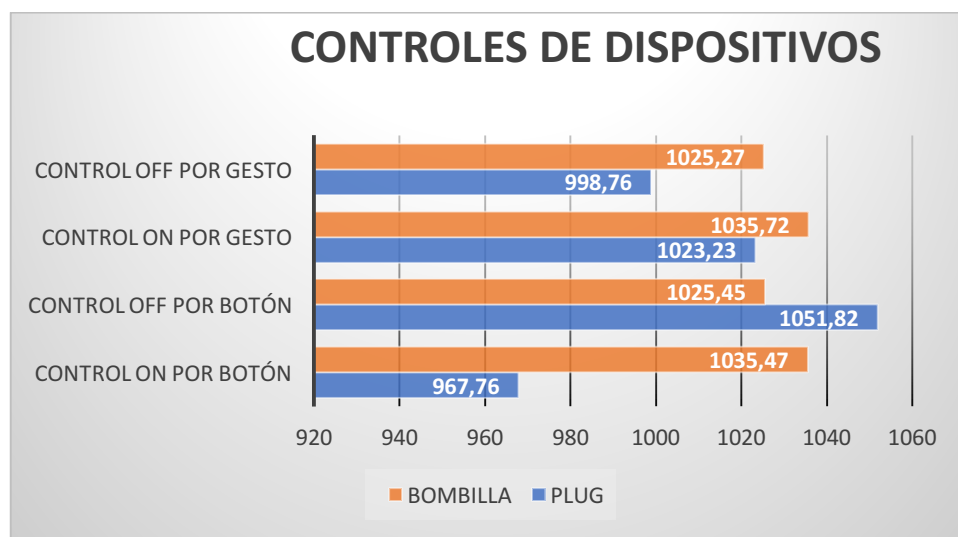


Figura 54 Ejecución de controles - Gráfico de resultados

Fuente: Elaboración Propia

En **Figura 53** y **Figura 54** se puede observar un gráfico de los resultados obtenidos a partir del estudio realizado a partir de 100 muestras, en donde se obtuvo un acceso de datos para el reconocimiento que se establece en segundos debido a aspectos técnicos del dispositivo y en cuanto a los controles, estos se reflejan en milisegundos debido a que son instrucciones que se ejecutan de manera inmediata, dándonos una aplicación con un buen tiempo de respuestas acorde a las necesidades que se podrían esperar.

4.5 CONCLUSIONES

- Se pudo comprender mediante la realización de este trabajo que se debe contar con una correcta instalación de los dispositivos ya que esto puede afectar en los tiempos de reacción de los comandos dados por la plataforma IoT.

- Dentro de las plataformas IoT la más importante para estos trabajos es Home Assistant debido a que presenta la mayor compatibilidad y soporte dado por la comunidad, para procesos de instalación, integración e inclusión de funcionalidades.
- Para el desarrollo de un sistema en Unity se debe considerar aspectos tales como las librerías que se deben implementar y los elementos que se manejan, ya que este es un entorno de desarrollo enfocado a programación multiplataforma brinda al usuario múltiples elementos útiles para el diseño estilizado de objetos.
- El uso del protocolo MQTT es factible para domótica debido a su flexibilidad, su alto rendimiento, su poco consumo energético y de recursos que representa un medio óptimo para el manejo y transmisión de datos entre los dispositivos y la plataforma IoT.
- La realidad aumentada fomenta un ambiente que permite al usuario el aprendizaje y la manipulación inmediata de las tecnologías, de manera que no se requiere un conocimiento técnico especializado para hacerlo.

4.6 RECOMENDACIONES

- Se recomienda que se verifique los diferentes dispositivos que son compatibles con la plataforma IoT domótica antes de realizar su adquisición, esto se debe a que no todos los productos disponibles en el mercado pueden ser vinculados.
- Se debe verificar los requisitos de software y hardware necesarios para la instalación de nuestra plataforma IoT domótica, ya que de esto dependerá el tiempo de respuesta para nuestros procesos de automatización.
- Es importante revisar las condiciones en que se encuentra nuestro hardware, en especial si contamos con un dispositivo base como la Raspberry ya que de presentarse inconveniente de alimentación o de almacenamiento nuestro sistema domótico no funcionaría.

- Se debe establecer medidas en caso de presentarse problemas para el acceso remoto a la plataforma IoT, ya que esto dificultará el proceso de configuración y actualización de la misma en caso de requerir agregar otro dispositivo e implementar una nueva automatización.

5 Bibliografía

- [1] C. R. Egas, D. Viracocha y J. Rivera, «Implementación de una red inalámbrica de sensores para la gestión de luminarias utilizando IPv6,» *Enfoque UTE*, vol. 10, nº 4, pp. 45-56, 2019.
- [2] J. Novillo-Vicuña, D. Hernández Rojas, B. Mazón Olivo, J. Molina Ríos y O. Cárdenas Villavicencio, «Arduino y el internet de las cosas,» vol. 45, 2018.
- [3] A. M. Campoverde, D. L. Hernández y B. E. Mazón, «Cloud computing con herramientas open-source para Internet de las cosas,» *Maskana*, vol. 6, pp. 173-182, 2015.
- [4] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas y A. Pan, «Rules engine and complex event processor in the context of internet of things for precision agriculture,» *Computers and Electronics in Agriculture*, vol. 154, pp. 347-360, 2018.
- [5] J. Berrú-Ayala, D. Hernandez-Rojas, P. Morocho-Díaz, J. Novillo-Vicuña, B. Mazon-Olivo y A. Pan, «SCADA System Based on IoT for Intelligent Control of Banana Crop Irrigation,» *Communications in Computer and Information Science Applied Technologies*, pp. 243-256, 2019.
- [6] M. I. Romero Castro y K. E. Macías Carreño, «Estudio de factibilidad de un sistema domótico para el control de iluminación al acceso de la edificación de la carrera de ingeniería en computación y redes,» *UNESUM*, p. 102, 2020.
- [7] S. Pérez-Lisboa, C. G. Ríos-Binimelis y J. Castillo Allaria, «Realidad Aumentada y simuladores: astronomía para niños y niñas de cinco años,» *Alteridad*, vol. 15, nº 1, pp. 25-35, 2020.
- [8] J. Cabero, J. Barroso y C. Llorente, «La realidad aumentada en la enseñanza universitaria,» *REDU. Revista de Docencia Universitaria*, vol. 17, nº 1, p. 105, 2019.
- [9] V. Alvear-Puertas, P. Rosero-Montalvo, D. Peluffo-Ordóñez y J. Pijal-Rojas, «Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura,» *Enfoque UTE*, vol. 8, nº 1, p. 244–256, 2017.

- [10] D. L. Hernández-Rojas, T. M. Fernández-Caramés, P. Fraga-Lamas y C. J. Escudero, «Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing through BLE Beacons in IoT Telemetry Applications,» *Sensors*, vol. 18, nº 2, p. 57, 2017.
- [11] D. Hernandez-Rojas, B. Mazon-Olivo, J. Novillo-Vicuña, C. Escudero-Cascon, A. Pan-Bermudez y G. Belduma-Vacacela, «IoT android gateway for monitoring and control a WSN,» *Communications in Computer and Information Science Technology Trends*, pp. 18-32, 2017.
- [12] M. Quinones-Cuenca, V. Gonzalez-Jaramillo, R. Torres y M. Jumbo, «Sistema De Monitoreo de Variables Medioambientales Usando Una Red de Sensores Inalámbricos y Plataformas De Internet De Las Cosas.,» *Enfoque UTE*, vol. 8, nº 1, pp. 329-343, 2017.
- [13] R. Rodríguez-Gómez, «Internet de las cosas: Futuro y desafío para la epidemiología y la salud pública,» *Univ. Salud*, vol. 21, nº 3, pp. 253-260, 2019.
- [14] J. Ramos Guardarrama, O. Hernández Areu y R. C. Silverio Freire, «Sistema de supervisión para el monitoreo de redes eléctricas inteligentes,» *Ingeniería Energética*, vol. 40, nº 3, pp. 264-272, 2019.
- [15] D. L. Hernández-Rojas, P. Fraga-Lamas, C. J. Escudero y T. M. Fernández-Caramés, «A Plug-and-Play Human-Centered Virtual TEDS Architecture for the Web of Things,» *Sensors*, vol. 18, nº 7, p. 2052, 2018.
- [16] A. Calvopiña, F. Tapia y L. Tello-Oquendo, «Uso del asistente virtual Alexa como herramienta de interacción para el monitoreo de clima en hogares inteligentes por medio de Raspberry Pi y DarkSky API,» *RISTI - Revista Ibérica de Sistemas e Tecnologías de Informação*, nº 36, pp. 102-115, 2020.
- [17] OpenHab, «Empowering the smart home,» [En línea]. Available: <https://www.openhab.org/>. [Último acceso: 28 noviembre 2020].
- [18] Domoticz, «Control at your finger tips,» [En línea]. Available: <https://www.domoticz.com/>. [Último acceso: 28 noviembre 2020].
- [19] Home Assistant, «Home Assistant Community,» [En línea]. Available: <https://www.home-assistant.io/>. [Último acceso: 28 noviembre 2020].

- [20] E. Escobar Gallardo y A. Villazón, «Sistema de monitoreo energético y control domótico basado en tecnología “Internet de las cosas”,» *Investigación & Desarrollo*, vol. 18, nº 1, pp. 103-116, 2018.
- [21] R. Robin Murphy, «Smart houses and domotics,» *Science*, vol. 3, nº 24, p. 38, 2018.
- [22] J. Fahrenkamp-Uppenbrink , «Cost of the “Internet of Things”,» *Science*, vol. 364, nº 6438, pp. 347-349, 2019.
- [23] R. Putra y H. Suryamen, «Sistem Monitoring Tanah Longsor Berbasis Internet of Things dan Geographic Information System,» *JITCE (Journal of Information Technology and Computer Engineering)*, vol. 3, nº 2, pp. 70-77, 2019.
- [24] G. Lara, A. Santana, A. Lira Y A. Peña, «El Desarrollo del Hardware para la Realidad Virtual,» *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, nº 31, pp. 106-117, 2019.
- [25] N. C. Huacho Suntaxi y J. D. Sañaicela Cueva, «Diseño e implementación de una aplicación Web adaptiva (servidor de aplicaciones) que controle un sistema domótico cumpliendo el estándar ISO/IEC 25010 para reducir el consumo de energía eléctrica en el hogar,» *Universidad Politécnica Salesiana Ecuador*, nº 1, pp. 44-61, 2020.
- [26] E. T Guerra Correa, «Estudio del motor de videojuego unity con sdk vuforia para el desarrollo de aplicaciones móviles de realidad aumentada, aplicación de tarjetas ilustradas en pares para niños,» *Repositorio digital Universidad Técnica del Norte*, pp. 27-62, 2019.
- [27] M. Muñoz-Sajama, D. Aracena-Pizarro, R. Cornejo-Mejías y M. Navarrete-Álvarez, «Una aplicación de Realidad Aumentada para recorrer el sitio patrimonial ‘Aldea de San Lorenzo’,» *Ingeniare. Revista chilena de ingeniería*, vol. 26, p. 65–76, 2018.
- [28] C. Salvador Ruiz, «Realidad aumentada y aprendizaje en la química orgánica,» *Apertura*, vol. 12, nº 1, pp. 106-117, 2020.
- [29] F. G. B. Molina, G. T. Toledo, F. A. Acevedo y E. M. Mendoza, «Augmented reality application centered on the child as a resource in a virtual learning environment,» *Apertura*, vol. 12, nº 1, pp. 88-105, 2020.

- [30] G. Nolasco de Almeida Mello y J. Cabero Almenara, «Realidad Aumentada en la enseñanza de hormigón reforzado: percepción de los alumnos,» *Alteridad*, vol. 15, nº 1, pp. 12-24, 2020.
- [31] J. Barroso-Osuna, J. Cabero-Almenara y J. Gutiérrez-Castillo, «La producción de objetos de aprendizaje en realidad aumentada por estudiantes universitarios. Grado de aceptación de esta tecnología y motivación para su uso,» *RMIE. Revista mexicana de investigación educativa*, vol. 23, nº 79, pp. 1261-1283, 2018.
- [32] V. Marín y B. E. Sampedro-Requena, «La Realidad Aumentada en Educación Primaria desde la visión de los estudiantes,» *Alteridad*, vol. 15, nº 1, pp. 61-73, 2019.