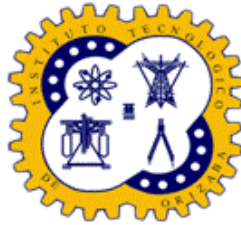




**TECNOLÓGICO  
NACIONAL DE MÉXICO**



**SEP**  
SECRETARÍA DE  
EDUCACIÓN PÚBLICA

**TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE ORIZABA  
DIVISIÓN DE ESTUDIOS DE POSGRADO E  
INVESTIGACIÓN**

---

**MAESTRÍA EN SISTEMAS COMPUTACIONALES**

---

**TESIS**

**DESARROLLO E IMPLEMENTACIÓN DE UN MÓDULO DE  
NUEVOS COMANDOS E INSTRUCCIONES PARA GOOGLE  
ASSISTANT**

**PRESENTADO POR:**

**I.S.C. NOÉ HERNÁNDEZ GARCÍA  
M13011151**

**DIRECTOR DEL PROYECTO DE TESIS:**

**M.R.T. IGNACIO LÓPEZ MARTÍNEZ**

**CODIRECTOR DEL PROYECTO DE TESIS:**

**M.C.E. BEATRIZ A. OLIVARES ZEPAHUA**



# *Autorización de Impresión*

# *Revisión de Trabajo Escrito*

## Agradecimientos

*A el cuerpo académico.*

*Agradezco al Consejo de Ciencia y Tecnología (CONACYT) y al Instituto Tecnológico de Orizaba, por su apoyo y patrocinio para la realización de este proyecto de tesis.*

*Agradezco a mi asesor de tesis el MRT Ignacio López Martínez, alias el profe Nacho como todos lo conocimos, por ser mi guía y mi consejero incondicionalmente durante el desarrollo de este proyecto.*

*A mis revisores de tesis, M.C.E. Beatriz A. Olivares Lepakua, M.C. S. Gustavo S. Peláez Camarena y M.C. Ma. Antonieta Abud Figueroa quienes con sus valiosos comentarios y recomendaciones me permitieron realizar mejoras a este trabajo de tesis. También quiero agradecer a el maestro Luis Ángel y al Dr. José Luis por apoyarnos siempre y escucharnos cuando nos quejábamos. De igual manera a la Coordinadora de la maestría en sistemas la maestra Celia por siempre tenernos paciencia en el papeleo.*

*Agradezco a todo los demás maestros de la maestría en sistemas por haber tenido esa plática motivacional sobre lo que estaba pasando con nuestras calificaciones y la cual nos impulsó a no cometer los mismos errores y seguir adelante.*

*A mi familia.*

*Agradezco a mi madre Elvia García Morales, a quien doy gracias por darme a luz, a pesar de que siempre fui el hijo más hiperactivo y el que le causó muchos dolores de cabeza, agradezco que siempre me haya llamado la atención en los momentos clave y que siempre me haya demostrado su amor incondicional y que estaba orgullosa de mí.*

*A mi padre Noé Hernández Trejo quien a pesar de ser estricto e inexpressivo siempre me decía las cosas sin rodeos al igual que consejos, que todo lo que hiciera iba a ser para mí bien que para nadie más y siempre me impulsó a realizar las cosas para nunca lamentarme de no haberlas hecho.*

*A mis hermanos Alexis Uriel Hernández García quien siempre me impulsó a continuar estudiando, a mi otro hermano Giovanni Hernández García el cual me ve como su ejemplo a seguir y eso me impulsaba a no darme por vencido.*

*Al amor de mi vida la Ing. Raquel Lizue Martínez Ramos con quien vivo grandes aventuras, quien siempre me impulsa a ser más de lo que puedo ser, por darme todo su apoyo y amor incondicional.*

*A mi tío Moisés Hernández Trejo quien me apoyó incondicionalmente durante mi licenciatura y que si ese apoyo no habría llegado a la maestría.*

*A mis abuelos Teresa y Rigoberto quienes siempre me han apoyado y dado palabra de aliento, así como sus grandes consejos y su cariño.*

*A mis amigos, no es necesario nombrarlos dado que ellos saben quiénes son, los cuales siempre me apoyaron y con los que compartí grandes experiencias y aventuras.*

*A todos ellos quienes con su confianza, cariño y apoyo sin escatimar esfuerzo alguno, me han convertido en una persona de prowecho, ayudándome al logro de una meta más: Mi maestría en Sistemas Computacionales. Por compartir tristezas, alegrías, éxitos y fracasos por todos los detalles que me han brindado durante mi vida como estudiante.*

*Gracias.*

*Vale más hacer y arrepentirse, que no hacer y arrepentirse*  
*-Nicolás Maquiavelo-*

# Índice General

Lista de tablas .....	IX
Lista de Figuras .....	X
Resumen .....	XII
Abstract.....	XIII
Introducción.....	XIV
Capítulo 1. Antecedentes.....	1
1.1. Marco teórico .....	1
1.1.1. Domótica .....	1
1.1.2. Lenguaje natural .....	1
1.1.2.1. Procesamiento de lenguaje natural .....	1
1.1.3. Agentes inteligentes.....	2
1.1.4. Google Assistant.....	2
1.1.4.1. SDK de Google Assistant .....	2
1.1.4.1.1. Biblioteca de Google Assistant.....	3
1.1.4.1.2. Servicio de Google Assistant.....	3
1.1.4.2. Acciones de Google Assistant .....	3
1.2. Situación tecnológica, económica y operativa de la empresa.....	4
1.3. Planteamiento del problema.....	4
1.4. Objetivo general y específicos.....	5
1.4.1. Objetivo general .....	5
1.4.2. Objetivos específicos .....	5
1.5. Justificación .....	5
Capítulo 2. Estado de la práctica .....	7
2.1 Trabajos relacionados .....	7

2.2	Análisis comparativo .....	14
2.3	Propuesta de solución .....	22
2.3.1	Justificación de la solución seleccionada .....	25
Capítulo 3. Aplicación de la metodología .....		26
3.1	Fase de Modelado .....	26
3.1.1	Modelado de la lógica de negocio .....	26
3.1.2	Arquitectura de desarrollo .....	27
3.2	Fase de Implementación .....	29
3.2.1	Implementación en Servicio Web.....	33
3.2.2	Implementación en dispositivos externos.....	37
3.3	Fase de Pruebas.....	45
3.4	Fase de Despliegue, Fase de Configuración y Administración de Proyecto .....	46
Capítulo 4. Resultados.....		48
4.1	Casos de estudio.....	48
4.1.1	Caso de estudio: Encendido y apagado de diversos dispositivos externos mediante un nuevo comando de voz de Google Assistant.....	48
4.1.2	Caso de Estudio: Uso de servicio web mediante un nuevo comando de voz de Google Assistant para la búsqueda de términos básicos de un diccionario. ....	55
Conclusiones.....		57
Recomendaciones .....		57
Productos académicos.....		59
Referencias .....		60



## **Lista de tablas**

Tabla 2.1 Análisis comparativo de los trabajos relacionados.....	14
Tabla 3.1 Solución para desarrollo de los comando de voz. ....	25

## Lista de Figuras

Figura 3.1 Diag. De Componente Google Assistant. ....	26
Figura 3.2 Diag. De Actividades Google Assistant.....	27
Figura 3.3 Arquitectura para el desarrollo de los nuevos comandos de voz. ....	28
Figura 3.4 Interfaz de Dialogflow (Intents).....	30
Figura 3.5 Interfaz de Dialogflow (Intents-parametros).....	31
Figura 3.6 JSON (Intents-parametros). ....	31
Figura 3.7 Interfaz de Dialogflow (Entities). ....	32
Figura 3.8 Fragmento del archivo index.js para servicio web.....	35
Figura 3.9 Debug del consumo del servicio 1. ....	35
Figura 3.10 Debug del consumo del servicio 2. ....	36
Figura 3.11 Debug del consumo del servicio 3. ....	36
Figura 3.12 Interfaz de Dialogflow (Fulfillment) del Servicio Web.....	37
Figura 3.13 Fragmento del archivo index.js para dispositivos externos. ....	40
Figura 3.14 Interfaz de Dialogflow (Fulfillment) para dispositivos externos. ....	40
Figura 3.15 Variables almacenadas en Firebase para dispositivos externos. ....	41
Figura 3.16 Placa NodeMCU ESP8266. ....	41
Figura 3.17 Esquema de conexión.....	42
Figura 3.18 Programacion del NodeMCU ESP8266.....	44
Figura 3.19 Simulador para el asistente virtual de Google.....	45
Figura 3.20 Fragmento de solicitud del simulador.....	46
Figura 3.21 Configuración general.....	47
Figura 4.1 Esquema de conexión realizado.....	49
Figura 4.2 Inicio del comando del comando de voz mediante simulador de Google.....	49
Figura 4.3 Encendido de la luz de la sala mediante simulador de Google.....	50
Figura 4.4 Apagado de la luz de la sala mediante simulador de Google.....	51
Figura 4.5 Encendido de la luz de la cocina mediante simulador de Google.....	51
Figura 4.6 Apagado de la luz de la sala mediante simulador de Google.....	52
Figura 4.7 Encendido de la alarma mediante simulador de Google.....	52
Figura 4.8 Apagado de la alarma mediante simulador de Google.....	53
Figura 4.9 Funcionamiento de los comando de voz en dispositivo Móvil parte 1.....	53

Figura 4.10 Funcionamiento de los comando de voz en dispositivo Móvil parte 2. ....	54
Figura 4.11 Prueba de funcionamiento de dispositivos.....	54
Figura 4.12 Términos básicos del servicio web. ....	55
Figura 4.13 Búsqueda continua de significados. ....	56
Figura 4.14 Búsqueda de significados mediante dispositivo móvil. ....	56

## Resumen

Los comandos de voz, desarrollados en *Google Assistant* son un tema de gran interés, los cuales permiten mejorar el desempeño en el estilo de vida empelando el uso de la domótica.

El objetivo del presente trabajo es el desarrollo e implementación de un conjunto de instrucciones que no están presentes actualmente para el asistente virtual *Google Assistant* mediante un análisis comparativo entre las diversas tecnologías para su implementación. De esta manera se podrá innovar en el desarrollo de nuevos conjuntos de comando de voz para el mejoramiento del estilo de vida.

Las tecnologías a ocupar son: como IDE (*Integrated Development Environment*, Entorno de Desarrollo Integrado) se utilizará la plataforma de Google Cloud, como agente inteligente Dialogflow el cual se encuentra en la plataforma seleccionada y esto en conjunto con el SDK (*Software Development Kit*, Kit de Desarrollo de Software) de *Google Assistant*.

## **Abstract**

The voice commands, developed in Google Assistant are a topic of great interest, which allow to improve the performance in the lifestyle by using the home automation.

The aim of this work is the development and implementation of a set of instructions that are not currently present for the Google Assistant virtual assistant through a comparative analysis between the various technologies for their implementation. In this way, it will be possible to innovate in the development of new voice command sets for the improvement of life style.

The technologies to be used are: as an IDE (Integrated Development Environment) the Google Cloud platform will be used, as an intelligent agent Dialogflow which is in the selected platform and this in conjunction with the SDK (Software Development Kit) of Google Assistant.

## Introducción

Los asistentes virtuales inteligentes son un conjunto de programas informáticos capaces de interactuar con los seres humanos en su propio lenguaje, esto quiere decir que son entidades autónomas que ofrecen colaboración a las personas para alcanzar objetivos a través de diferentes canales de interacción. En [1] los asistentes virtuales se pueden clasificar de la siguiente manera: aquellos que tienen un dispositivo físico y los que se encuentran embebidos en los sistemas móviles.

No hay un estudio que compare las diferentes tecnologías para el desarrollo e implementación de un conjunto de instrucciones que no están presentes actualmente para el asistente virtual Google, ello conlleva que las personas que busquen una técnica adecuada para el desarrollo de un conjunto instrucciones según sus necesidades.

Este trabajo resuelve la problemática sobre el desarrollo de un conjunto instrucciones, realizando un análisis de las tecnologías y comparando cada una de ellas, para el desarrollo de un conjunto de instrucciones, eligiendo una para su posterior implementación y validación. De esta manera se podrá elegir de mejor manera la tecnología que más se adecue a las necesidades de ciertos escenarios.

Este trabajo se organiza en cinco capítulos: en el capítulo uno abarca el marco teórico, planteamiento del problema, objetivo general y específicos y la justificación; el capítulo dos aborda el estado de la práctica y un análisis comparativo; el capítulo tres abarca la propuesta de solución específica, la descripción de la solución, el cronograma de actividades y un análisis comparativo de los IDEs, lenguajes de programación, agentes inteligentes y metodologías; el capítulo cuatro se muestran los resultados sobre la realización del proyecto de tesis mediante dos casos de estudio; por último el quinto capítulo se muestran algunas conclusiones y recomendaciones a futuro para el proyecto de tesis.

## **Capítulo 1. Antecedentes**

### **1.1.Marco teórico**

A continuación, se definen algunos términos relevantes para el trabajo de investigación.

#### **1.1.1. Domótica**

La integración tecnológica de los sistemas electrónicos en el hogar se conoce como domótica. En [1] se define este término como agrupaciones automatizadas de equipos, normalmente asociados por funciones, que disponen de la capacidad de comunicarse interactivamente entre ellos a través de una conexión multimedia que los integra.

#### **1.1.2. Lenguaje natural**

El lenguaje natural es el conjunto de signos y símbolos orales por medio de los cuales los seres humanos se comunican entre sí, a través de la formación de palabras que expresan ideas [2].

En los asistentes virtuales el uso del lenguaje natural es una pieza clave en el desarrollo de nuevos comandos e instrucciones para el Google Assistant, ya que es la forma en la que el usuario interactúa con este último; si bien la complejidad de identificar, más que entender dicho lenguaje está cubierta por el propio asistente virtual, aún existen adecuaciones a la forma de expresarse que no se interpretan de forma correcta.

##### **1.1.2.1. Procesamiento de lenguaje natural**

El Procesamiento del Lenguaje Natural (PLN) es una transformación del lenguaje natural a una representación formal, dicha representación se manipula y si es necesario, se llevan los resultados nuevamente al lenguaje natural.

El PLN incluye la recuperación y extracción de información, traducción automática, sistemas de búsquedas de respuestas, generación de resúmenes automáticos, minería de datos y análisis de sentimientos [3].

### **1.1.3. Agentes inteligentes**

Un agente es una entidad autónoma capaz de almacenar conocimiento sobre sí misma y sobre su entorno, de otra forma, es un programa que basándose en su propio conocimiento, realiza un conjunto de operaciones para satisfacer las necesidades de un usuario o de otro programa, bien por iniciativa propia o porque alguno de los anteriores se lo ordena [4]. El agente inteligente de *Google*, es capaz de analizar el lenguaje natural y obtener las palabras clave que sirven para ejecutar una acción o instrucción.

### **1.1.4. Google Assistant**

El asistente virtual personal de *Google* es una tecnología desarrollada con inteligencia artificial, su principal característica es la conversación bidireccional entre los usuarios y *Google*, donde las preguntas y respuestas se pueden complementar sin restringirse a un solo tema [5]. Brinda la posibilidad de comunicación bidireccional en forma de una conversación real. El usuario puede interactuar con *Google Assistant* (GA) con la ayuda de comandos de voz gracias a sus algoritmos de procesamiento de lenguaje natural, además, realizar búsquedas en Internet, programar citas, configurar alarmas y hasta administrar la capa de sesión de medios, también se encuentra como un servicio disponible en la nube.

Además *Google Assistant*, brinda posibilidades de incrementar las funciones o acciones que permitan alcanzar un mayor grado de interacción con los usuarios, para ello se dispone de un entorno de desarrollo proporcionado por el propio *Google*.

#### **1.1.4.1. SDK de Google Assistant**

El SDK de *Google Assistant* es una tecnología que permite agregar nuevos comandos de voz mediante la detección de palabras clave, usando control de voz, comprensión del lenguaje natural y recibiendo las ideas usando la inteligencia de *Google*. Captura una emisión (una solicitud de audio hablada, como ¿Qué hay en mi calendario?), la envía al asistente y recibe una respuesta de audio hablada, además del texto en bruto de la emisión [6].

El SDK ofrece dos opciones para integrar el asistente en proyectos: la Biblioteca de *Google Assistant* y el Servicio de *Google Assistant*.



#### 1.1.4.1.1. Biblioteca de Google Assistant

La biblioteca de *Google Assistant* es una solución de llave en mano para cualquier persona que quiera integrar rápidamente el asistente de Google en un proyecto. La biblioteca está escrita en Python y es compatible con hardware con las arquitecturas linux-armv7l y linux-x86\_64 (como las computadoras Raspberry Pi 3 B y Ubuntu) [6]. La biblioteca expone una API de alto nivel basada en eventos que es fácil de ampliar. La cual proporciona las siguientes características:

- Captura y reproducción de audio.
- Gestión del estado de conversación.
- Gestión de temporizadores y alarmas.

#### 1.1.4.1.2. Servicio de Google Assistant

El servicio de asistente de Google proporciona la flexibilidad y el soporte de plataforma amplia. Expone una API de bajo nivel que manipula directamente los bytes de audio de una solicitud y respuesta del Asistente. Los enlaces para esta API se pueden generar para lenguajes como Node.js, C#, C ++, Java para todas las plataformas que soportan gRPC.

El código de referencia se proporciona en Python para la captura de audio, la reproducción de audio y la administración del estado de conversación [6].

#### 1.1.4.2. Acciones de Google Assistant

Es una plataforma para desarrolladores que permite crear software para ampliar la funcionalidad de *Google Assistant*, en más de 500 millones de dispositivos, incluidos altavoces inteligentes, teléfonos, automóviles, televisores, auriculares y más. Los usuarios entablan conversación con el asistente de Google para hacer las cosas, como comprar comestibles o reservar un viaje [7].

Esto es posible utilizando como parámetros los siguientes términos.

- **Intención:** un objetivo o tarea que los usuarios desean realizar, como pedir un café o encontrar una pieza musical. En acciones en *Google Assistant*, esto se representa como un identificador único y las expresiones de usuario correspondientes que pueden desencadenar la intención.

- **Acción:** Interacción que se construye para el asistente que respalda una intención específica y tiene un cumplimiento correspondiente el cual procesa.
- **Cumplimiento:** un servicio, aplicación, fuente, conversación u otra lógica que maneja un intento y lleva a cabo la acción correspondiente.

## **1.2. Situación tecnológica, económica y operativa de la empresa**

El Instituto Tecnológico de Orizaba ofrece carreras a nivel licenciatura tales como Ingeniería Eléctrica, Ingeniería Electrónica, Ingeniería en Sistemas Computacionales, Ingeniería Química, Ingeniería Mecánica, Ingeniería en Gestión Empresarial, Ingeniería Industrial e Ingeniería Informática. Esto limita el acceso a las tecnologías de información, sin embargo, existe la posibilidad de solicitar el uso de algún laboratorio de forma temporal sin garantizar la petición.

A su vez, el instituto ofrece carreras a nivel maestría y doctorado, como son Maestría en Ingeniería Electrónica, Maestría en Ingeniería Industrial, Maestría en Sistemas Computacionales, Maestría en Ingeniería Administrativa, Maestría en Ciencias en Ingeniería Química y Doctorado en Ciencias de la Ingeniería. Particularmente, la Maestría en Sistemas Computacionales cuenta con un laboratorio para cada una de sus líneas de trabajo, las cuales son: 1) Ingeniería de Software, y 2) Tecnologías Web.

## **1.3. Planteamiento del problema**

El reconocimiento de voz avanzó en las últimas décadas y permitió la capacidad de hablar de forma natural y contextual con un sistema computacional interactuando en la ejecución de comandos o dictado.

En el caso de los asistentes virtuales presenta una tecnología probada, con mucho potencial y que permite hacer volar la imaginación sobre donde o que se podría hacer con ella, sin embargo en el futuro inmediato de esta tecnología aún no se encuentra automatizada en construcción de aplicaciones por parte del cliente final, es decir, si se requiere que un asistente virtual acepte un nuevo conjunto de instrucciones, que active, encienda o ejecute algún dispositivo que no has sido configurado dentro de su colección de datos a interpretar, se vuelve difícil para el usuario final hacer una programación o configuración para su uso.

Para esto, se revisarán los asistentes virtuales y robots conversacionales que utilicen el SDK *Google Assistant* desarrollados en los últimos cinco años, por medio de la búsqueda en las principales bibliotecas digitales como son *ACM Digital Library*, *Springer Link*, *Science Direct* y *IEEEExplore*, posteriormente se realizará el desarrollo e implementación de un conjunto de instrucciones que no están presentes actualmente para el asistente virtual *Google Assistant* mediante un análisis comparativo para determinar sus ventajas y desventajas tras lo cual se seleccionará una de las tecnologías utilizadas anteriormente analizadas.

#### **1.4. Objetivo general y específicos**

A continuación, se muestra el objetivo general y los objetivos específicos.

##### **1.4.1. Objetivo general**

Desarrollar e implementar un conjunto de instrucciones que no están presentes actualmente, para el asistente virtual *Google Assistant*.

##### **1.4.2. Objetivos específicos**

- Analizar el estado del arte de los asistentes virtuales y robots conversacionales que utilicen el SDK *Google Assistant*.
- Caracterizar las instrucciones y los dispositivos donde se llevará a cabo la implementación.
- Establecer las condiciones contextuales donde se realizarán los casos de estudio.
- Analizar la tecnología actual del modelo de desarrollo de Google.
- Desarrollo mediante el servicio de Google Assistant para los casos de estudio.
- Implementar el conjunto nuevo de instrucciones en los casos de estudio.

#### **1.5. Justificación**

Los asistentes virtuales son soluciones de software que brindan sistemas de interacción con los humanos a través de la voz. Su objetivo es responder a determinadas preguntas y ayudar en tareas y procesos que se lleven a cabo dentro de un entorno digital. Se pueden encontrar en altavoces, aplicaciones de mensajería, como parte del sistema operativo de un móvil, en relojes, en aplicaciones móviles, en automóviles, etc. Es ahora, uno de los segmentos con mayor crecimiento y proyección en materia de inteligencia artificial e Internet. Además, una de las tendencias de marketing.

Los usuarios que cuenten con un asistente virtual podrán utilizarlo para llevar a cabo actividades como las siguientes:

- Solicitar información sobre el tiempo y datos de *Wikipedia* o *IMDb*.
- Buscar las noticias del día.
- Ajustar alarmas y listas de pendientes o compras.
- Reproducir música de servicios de *streaming* y estaciones de radio. Además, leen audiolibros.
- Reproducir vídeos, programas y películas en televisiones.
- Comprar artículos.
- Complementar y/o reemplazar servicios de atención al cliente por humanos. Este, en particular, es una de las grandes ventajas a nivel empresarial.

Sim embargo los asistentes virtuales al momento de hacer uso de los dispositivos externos como lo son focos, ventiladores, o cualquier otro dispositivo domótico o no domótico, se encuentra una carencia, la carencia de estos es que los dispositivos solo pueden ejecutar una acción y esto provoca que la interacción con que lleva con el usuario sea monótona.

Debido a las necesidades presentadas es preciso realizar una programación utilizando el SDK *Google Assistant* para incursionar en la creación y desarrollo de un módulo que incluya un nuevo juego de comandos, instrucciones, acciones e inclusive aprendizajes que se puedan utilizar en diferentes dispositivos, principalmente el *Google Home Mini* y se convierta en parte del contexto habitual de uso.

Por ejemplo; para activar algún dispositivo como un ventilador, una serie de focos tradicionales, una cafetera, entre otros que no tiene diferentes funciones o que desencadenen otras funciones permitiendo una interacción más fluida con el usuario.

## Capítulo 2. Estado de la práctica

Antes de introducirse al tema por completo, se revisó un considerable número de artículos de los cuales se tomó la información más relevante y se revisaron meticulosamente las investigaciones relacionadas con el tema propuesto. El análisis de los artículos dio como resultado la importancia que actualmente tienen los asistentes personales y cómo sacarles un gran provecho, dejando ver que hoy en día se requieren mucho más para cubrir muchas necesidades por parte de los usuarios. Cada artículo consultado hizo su contribución en esta investigación.

### 2.1 Trabajos relacionados

En [8] se estableció que el uso de asistentes inteligentes y la automatización de hogares inteligentes se vuelven cada vez más fuerte. Los asistentes virtuales habilitados para voz ofrecen una amplia variedad de servicios orientados a la red y, en algunos casos, logran conectarse a entornos inteligentes, mejorándolos con interfaces de usuario nuevas y efectivas; sin embargo, dichos dispositivos revelan nuevas necesidades y debilidades, por ello, se desarrolló un nuevo asistente inteligente con la capacidad de observar un rostro y determinar las emociones de este, dicho asistente inteligente se ejecuta en un dispositivo *Raspberry Pi 3* por medio de una interfaz gráfica desarrollada en *HTML5* que contiene *JavaScript* necesario para comunicarse con una conexión *Websocket* y que implementa la API (*Application Programming Interface*, Interfaz de Programación de Aplicaciones) *RESTful* con los módulos de síntesis y reconocimiento de voz, esto permite una mejor interacción entre el usuario y el asistente inteligente.

Debido a que los comandos de voz se encuentran expuestos a diversas situaciones que alteran el entendimiento del lenguaje natural en los dispositivos, Michaely et al. [9] presentó un nuevo sistema de localización de palabra clave, este sistema que utiliza el ASR (*Automatic Speech Recognition*, Reconocimiento Automático de Voz) del lado del servidor y una frase desencadenante para mejorar la precisión general de KWS (*Knowledge Work System*, Sistemas de Manejo de Conocimiento), el cual, permite al usuario hablar sin interrupciones entre la frase de activación y el comando de voz, reduciendo así significativamente la tasa de

falsas aceptaciones (FA) (en un 89%) al mismo tiempo que aumenta de manera mínima la tasa de falsos rechazos (FR) en un 0.2% para mejorar la calidad de ASR.

Las NUI (*Natural User Interface*, Interfaz de Usuario Natural) deben ser utilizadas por los humanos de una manera muy lógica. Sin embargo, la ejecución para implementar las NUI basadas en el habla por parte de la industria ha tenido un gran impacto en la naturalidad de tales interfaces. En [10] se presentó una prueba de usabilidad de la NUI en los asistentes inteligentes más prestigiosos e internacionalmente utilizados (Alexa, Siri, Cortana y *Google Assistant*). También se realizó una comparación de los servicios que proporciona cada uno considerando: acceso a servicios de música, agenda, noticias, clima, listas de tareas y mapas o direcciones, entre otros. Los resultados mostraron que hay muchos servicios disponibles y mucho que hacer para mejorar la usabilidad de estos sistemas.

El asistente virtual otorga beneficios tecnológicos y de implementación, para todos en esta nueva era del siglo XXI. Allanoó el camino para una nueva tecnología en la que se hacen preguntas a la máquina e interactúa como lo hacen las personas con los humanos. Esta nueva tecnología atrajo a casi todo el mundo de muchas maneras, como teléfonos inteligentes, computadoras portátiles, computadoras, por mencionar algunos. Algunos de los importantes son Siri, el Asistente de Google, Cortana y Alexa. Sin embargo, el reconocimiento de voz, la comprensión contextual y la interacción humana son los problemas que aún no se han resuelto. Entonces, para resolver esos problemas Tulshan y Dhage [11] presentaron un análisis a partir de una encuesta llevada a cabo para esta investigación. De acuerdo con los resultados, estos asistentes cubrieron muchos servicios, pero aún se requieren algunas mejoras en el reconocimiento de voz, la comprensión contextual y la interacción con manos libres. Después de abordar estas mejoras, los asistentes virtuales aumentaron definitivamente su uso cumpliendo así el objetivo principal del análisis.

En [12] se presentó un estudio exploratorio de métodos mixtos que describe el uso de asistentes inteligentes por parte de usuarios de bajos y medianos ingresos. Dado que asistentes inteligentes están ganando popularidad a un ritmo creciente. Con una variedad de interfaces de usuario de voz disponibles, algunas como parte de las tecnologías digitales como los teléfonos inteligentes (*Google Assistant*, búsqueda habilitada por voz, *Siri* de *Apple*) y otras como artefactos de tecnología independientes (*Google Home*, *Alexa* de

Amazon). Existe una amplia gama de usuarios que están adoptando la tecnología para una variedad de propósitos por lo cual es necesario realizar dicho estudio. Mediante el uso de una metodología basada en entrevistas, después los datos son codificados y sometidos a análisis temático. Actualmente, se lleva a cabo el estudio diariamente y continúa con un análisis en profundidad de los datos recopilados.

En los últimos años, debido al progreso de las tecnologías de la información, el hogar inteligente ha sustituido cada vez más el patrón de vida en el hogar. Si bien el estilo de casa inteligente puede traer enormes beneficios a las personas, hoy en día las casas inteligentes ofrecen funciones para garantizar la seguridad de la vida doméstica, el entorno de vida saludable y para proporcionar una calidad confortable de la vida doméstica. Sin embargo, existen funciones que no se tienen contempladas, como lo es el ahorro de energía, el cual es distribuido a todos los dispositivos mediante un socket el cual no puede ser controlado. Por lo tanto, en [13] se desarrolló una función hecha a medida para los usuarios, el cual consiste en el encendido y apagado de un socket, utilizando el reconocimiento de voz de Google Home con la concepción del aprendizaje automático para probar el análisis de viabilidad sobre cómo satisfacer las necesidades de los usuarios mediante un patrón de hogar inteligente con el diseño del aprendizaje automático utilizando *LSA (Latent Semantic Analysis, Análisis semántico latente)* y *TF-IDF (Term Frequency–Inverse Document Frequency, Frecuencia de Término – Frecuencia Inversa de Documento)*. Para conectar el dispositivo móvil con el Raspberry Pi se utilizó *BluePy* que es una suite que proporciona *API (Application Programming Interface, Interfaz de Programación de Aplicaciones)* para conectar Bluetooth bajo consumo basados en el lenguaje *Python* el cual se conecta con el dispositivo Google Home, el Bluetooth abre una conexión utilizando *API.ia* hoy conocido como *Dialogflow* permitiendo conversaciones en lenguaje natural humano-computadora mediante su *SDK (Software Development Kit, Kit de Desarrollo de Software)*, este permitirá la creación de la nueva función. El experimento permite a los usuarios enviar comentarios al reconocimiento de voz de Google Home y luego transferir la señal de Bluetooth a Raspberry Pi para controlar los dispositivos.

Hwang et al. [14] propusieron una arquitectura para la generación automática de guías de interacción de usuario con asistente inteligente. Dado que el mayor obstáculo es que es muy

difícil para los usuarios conocer el alcance del servicio (característica o conocimiento) y, por lo tanto, es difícil entregar la solicitud correcta. Si el asistente inteligente sugiere las funciones adecuadas basadas en el contexto o las preferencias del usuario con el idioma o las voces naturales, podría hacer que los usuarios estén familiarizados con el asistente inteligente, por lo que podría hacer que los asistentes inteligentes sean muy utilizados. Para el desarrollo de esta arquitectura se utilizó una ontología denominada red de tareas, esta red de tareas se usa para encontrar la secuencia correcta de acciones para lograr la solicitud del usuario. Se utilizan los métodos NLG/TTS (*Natural Language Generation/Text-To-Speech*, Generación de Lenguaje Natural/Texto A Voz) para hacer la guía natural basada en los planes que el sistema desea recomendar.

El autocontrol de la diabetes por parte de los pacientes es crucial para reducir su progresión crónica y futuras complicaciones de salud graves. Debido a las complejidades físicas y emocionales de la diabetes, la adherencia a las actividades de autocuidado disminuye rápidamente debido al progreso desalentador. Si bien existe una tecnología que se adapta al autocontrol de los diabéticos, el estado actual de la técnica ayuda principalmente a los pacientes a través de aplicaciones móviles, sin embargo, para las personas mayores, estas aplicaciones móviles son marginalmente efectivas e incluso frustrantes de usar. En Cheng et al.[15] desarrollaron y evaluaron una aplicación con el asistente de Google Home que actúa como una estrategia de intervención innovadora para ayudar a los pacientes ancianos con el autocontrol de la diabetes tipo 2. Este marco de aplicación combinó la interfaz de voz de *Google Home* para alojar el agente de conversación utilizando la plataforma de *API.ia* hoy conocido como Dialogflow en conjunto con el SDK de Google y una interfaz web para la visualización de datos se creó un *WebHook*, el cual consiste en devolución de llamada *HTTP* (*Hypertext Transfer Protocol*, Protocolo de Transferencia de Hipertexto) utilizando *Node.js* como plataforma, este utiliza JavaScript con lenguaje de programación. Se realizó una comparación basada en las características de la aplicación con las aplicaciones móviles disponibles actualmente para determinar el cumplimiento relativo de los componentes oficiales. Como resultado la aplicación mejoró el estado actual de la técnica al aumentar la satisfacción y conveniencia del usuario.



Hoy en día, *Google Assistant* se está convirtiendo en un asistente personal cada vez más popular y la gente usa con frecuencia, sus funciones de las aplicaciones de sus dispositivos inteligentes. Debido a que la mayoría de las herramientas de reconocimiento de voz existentes proporcionan el resultado del procesamiento del habla en una salida de texto de forma libre o en una salida de texto de forma estructurada para utilizar la aplicación de TV basada en Android, es necesario detectar patrones que corresponden a comandos como subir/bajar canal, subir/bajar volumen, entre otros, sin embargo, el objetivo es mejorar la calidad de experiencia del usuario, por lo que no tiene que usar un control remoto, sino un comando de voz. Por lo cual en [16] se presentó una arquitectura de software que admite la integración de *Google Assistant* en una aplicación de TV diseñada para el sistema operativo Android, utilizando el SDK (*Software Development Kit*, Kit de Desarrollo de Software) de Google junto con el sistema operativo Android. Como resultado la integración de *Google Assistant* junto a la aplicación de Android, permite al usuario administrar la aplicación de TV o el receptor de TV digital mediante comandos de voz.

El uso de aparatos inteligentes es un campo que no se encuentra explorado en su totalidad y el cual proporciona una amplia gama de oportunidades de desarrollo, sin embargo, esto genera que también se consideren necesidades para los usuarios, los cuales buscan la comodidad y sencillez al utilizar los aparatos inteligentes, Mischie et al. [17] investigaron la implementación del asistente de voz de Google en una microcomputadora Raspberry Pi utilizando un micrófono y un altavoz para el idioma Rumano, el cual se prueba mediante el encendido de focos LEDs. La implementación hace uso del ASR (*Automatic Speech Recognition*, Reconocimiento Automático de Voz), SLU (*Spoken Language Understanding*, Comprensión del Lenguaje Hablado), DM (*Dialogue Manager*, Administrador de Diálogo), KDB (*Knowledge Data Base*, Base de Datos de Conocimiento), NLG (*Natural Language Generation*, Generador de Lenguaje Natural) y TTS (*Text To Speech*, Texto A Voz). La implementación funciona con cualquier idioma, sin embargo, solo uno a la vez.

Uno de los objetivos de la inteligencia artificial (IA) es la realización de un diálogo natural entre humanos y máquinas. Muchas compañías han usado la tecnología de sistemas de diálogo para establecer varios tipos de VPA (*Virtual Personal Assistants*, Asistentes Personales Virtuales) basados en sus aplicaciones y áreas, como Cortana de Microsoft, Siri

de Apple, Amazon Alexa y Asistente de Google. Sin embargo los VPA, no cuentan con muchas funcionalidades que se requieren para tener un mejor control sistemático en las compañías, en [18] se propuso una nueva estructura utilizando los sistemas de diálogo multimodal que procesan dos o más modos de entrada de usuario combinados, como el habla, la imagen, el video, el tacto, los gestos manuales, la mirada, el movimiento de la cabeza y el cuerpo para diseñar el modelo de la próxima generación de VPA. El nuevo modelo de VPA utilizó diferentes tecnologías, las cuales fueron el TTS, SLU, NLG, DM, ASR, dichas tecnologías cumplieron con las tareas de reconocimiento de gestos, reconocimiento de imágenes/video, reconocimiento de voz, manejador de diálogo, la base de conocimiento conversacional y la base de conocimiento general. Como resultado el modelo de VPA utilizó voz, gráficos, video, gestos y otros modos de comunicación en los canales de entrada y salida. Además, el modelo de VPA se utilizó para aumentar la interacción entre los usuarios y las computadoras, de igual manera, este modelo permite una larga conversación con los usuarios mediante el uso de la amplia base de conocimientos de diálogo.

*Google Home* permite la interacción con el usuario para llevar a cabo tareas que solo requieren el comando de voz por parte del usuario, sin embargo, se desconoce qué tan relevante es el dispositivo comparado con Alexa de Amazon, al igual qué tipo de componente lo conforman, debido a que los desarrolladores encuentran dificultades al interactuar con el dispositivo y se ven en la necesidad de utilizar las tecnologías que estos mismos utilizan, sin poder tener otra alternativa. Dempsey [19] presentó un análisis de los componentes que integran, de la misma forma qué impacto tiene el dispositivo *Google Home* en costo y beneficio en el mercado. Los componentes con los que cuenta el dispositivo *Google Home* son el procesador de medios, memoria (flash), SoC (*System on Chip*, Sistema en Chip) inalámbrico, amplificador de audio, memoria DRAM (*Dynamic Random Access Memory*, Memoria Dinámica de Acceso Aleatorio), conductor del altavoz, microcontrolador, controlador LED y micrófonos. Como resultado del análisis se demostró que el dispositivo *Google Home* aunque todavía necesita crecer, tienen la infraestructura más grande en la actualidad y los beneficios de la portabilidad.

Para las personas mayores, el aislamiento social es un factor importante en el deterioro de la calidad de vida. Tiene un profundo impacto en la salud general y se produce por la

disminución de las interacciones sociales. Hoy en día, existe una tecnología que puede recuperar datos contextuales del entorno del usuario e interactuar con él de alguna manera simple pero efectiva. Los asistentes personales inteligentes pueden interactuar con la persona mediante un lenguaje de voz natural. En [20], se evaluó la utilización de un modelo que sirve como guía para construir un sistema con IPA (*Intelligent Personal Assistants*, Asistentes Personales Inteligentes) que sirve para apoyar las interacciones con las personas mayores. El resultado obtenido de todas las IPA, demuestra que podrían ejecutar las interacciones hasta cierto punto, pero no completamente como lo requiere el modelo. Existe una brecha entre lo que pueden hacer las IPA y lo que efectivamente hacen.

Incluso con la disponibilidad de varias pruebas para proporcionar claridad al elegir la trayectoria profesional, la decisión sigue siendo difícil de tomar. La mayoría de las pruebas disponibles son monótonas, lo que resulta en un esfuerzo tedioso para realizarlas por completo, o simplemente son aburridas. Calvo et al. [21], sin embargo, presentó un enfoque nuevo y diferente de los sistemas de orientación profesional. Utilizado *Google Home* como una interfaz basada en el habla y *Telegram* como una interfaz basada en texto para generar una conversación entre los usuarios y un bot para orientación profesional en conjunto con el SDK del asistente de Google y *API.ai*, hoy conocido como *Dialogflow*. El sistema proporcionó una interfaz fácil y amigable con una experiencia de usuario interactiva al tiempo que recopila los datos necesarios para proporcionar orientación profesional.

Višekruna y Savić [22] realizaron la integración del asistente de Google en una aplicación de prueba de Android. La solución permite el control por voz de la reproducción de contenido en un televisor o receptor digital. El usuario puede iniciar o detener la reproducción, ir al segmento siguiente o anterior, o saltar un cierto número de segundos. Esto es posible gracias a la implementación de Sistema Android en conjunto con el SDK del asistente de Google utilizando la plataforma de Google Cloud. Se determinó que la velocidad de ejecución del comando no cambiaba en relación con la velocidad cuando los usuarios emitían los comandos, la única ventaja es que no había problemas para entender el idioma del asistente. Surge dado que el uso de la tecnología de comando de voz está en aumento hoy en día. Los productos electrónicos permiten una nueva funcionalidad de voz y esto provoca la necesidad de desarrollar nuevas soluciones por parte de los desarrolladores.

## 2.2 Análisis comparativo

En la tabla 2.1 se muestra un análisis comparativo de los trabajos descritos, identificando diferencias y similitudes, buscando información importante para el proyecto.

Tabla 2.1 Análisis comparativo de los trabajos relacionados.

Artículos	Problema	Contribución	Tecnologías	Resultados	Estado
<b>Iannizzotto et al. [8].</b>	Los asistentes virtuales habilitados para voz ofrecen una amplia variedad de servicios, sin embargo, dichos dispositivos revelan nuevas necesidades y debilidades.	Desarrollo de asistente inteligente el cual se ejecuta en un dispositivo Raspberry Pi 3.	<i>HTML5</i> , <i>JavaScript</i> , conexión <i>Websocket</i> y API <i>RESTful</i> , SDK de Google.	El asistente es efectivo y eficiente en recursos, así como interactivo y personalizable.	Finalizado.
<b>Michaely et al. [9].</b>	Los dispositivos detectan rápidamente una frase de activación, sin embargo son propensos a errores en el reconocimiento de voz, los cuales conceden causar una experiencia de usuario insatisfactoria.	Sistema de localización de palabra clave.	Sistema KWS, ARS.	Permite al usuario hablar sin interrupciones entre la frase de activación y el comando de voz, reduciendo así significativamente la tasa de FA (en un 89%) al mismo tiempo que aumenta de manera mínima la tasa de FR (en un 0.2%).	Finalizado.

Artículos	Problema	Contribución	Tecnologías	Resultados	Estado
<b>Lopez et al. [10].</b>	Las NUI deben ser utilizadas por los humanos de una manera muy lógica, sin embargo, la ejecución para implementar las NUI por parte de la industria ha tenido un gran impacto.	Comparación de interfaces de usuario naturales basadas en el habla.	Ninguna.	Los resultados muestran que aunque hay muchos servicios disponibles aún queda mucho que hacer para mejorar la usabilidad de estos sistemas.	Finalizado.
<b>Tulshan et al. [11].</b>	La interacción con los asistentes virtuales carece de comprensión en el reconocimiento de voz, la comprensión contextual y la interacción humana.	Análisis a partir de encuesta.	Ninguna.	De acuerdo con los resultados, estos asistentes cubrieron muchos servicios, pero aún se requieren algunas mejoras en el reconocimiento de voz, la comprensión contextual y la interacción con manos libres.	Finalizado.
<b>Bhalla [12].</b>	Existe una amplia gama de usuarios que están adoptando la tecnología para una variedad de propósitos, por lo que es necesario un estudio a fondo.	Estudio exploratorio de métodos mixtos que describe el uso de asistentes inteligentes por parte de usuarios de bajos y medianos ingresos.	No se menciona la tecnología.	Actualmente, se lleva a cabo el estudio diariamente y continuando con un análisis en profundidad de los datos recopilados.	En proceso.

Artículos	Problema	Contribución	Tecnologías	Resultados	Estado
<b>C. Peng y R. Chen [13].</b>	Hoy en día las casas inteligentes ofrecen eficiente servicio de funciones para garantizar la seguridad de la vida doméstica, el entorno de vida saludable, y para proporcionar una calidad confortable de la vida doméstica. Sin embargo, existen funciones que no se tienen contempladas como lo es el ahorro de energía, el cual es distribuido a todos los dispositivos mediante un socket, este mismo no puede ser controlado.	Nueva función para el control de un socket inteligente.	<i>Python, Dialogflow</i> y SDK de Google.	El experimento permite a los usuarios enviar comentarios al reconocimiento de voz de Google Home y luego transferir la señal de Bluetooth a Raspberry Pi para controlar los dispositivos.	Finalizado.
<b>Hwang et al. [14].</b>	Dado que el mayor obstáculo para los usuarios es conocer el alcance del servicio (característica o conocimiento), por lo tanto, es difícil entregar la solicitud correcta.	Arquitectura para la generación automática de guías de interacción de usuario con asistente inteligente.	Ontología y Métodos NLG/TTS.	Se generó automáticamente las guías de interacción de usuario con asistente inteligente.	Finalizado.

Artículos	Problema	Contribución	Tecnologías	Resultados	Estado
<b>Cheng et al. [15].</b>	Si bien existe una tecnología que se adapta al autocontrol de los diabéticos, el estado actual de la técnica ayuda principalmente a los pacientes a través de aplicaciones móviles, Sin embargo para las personas mayores, estas aplicaciones móviles son marginalmente efectivas e incluso frustrantes de usar.	Desarrollo y evaluación de una aplicación con el asistente de Google Home que actúa como una estrategia de intervención innovadora para ayudar a los pacientes ancianos con el autocontrol de la diabetes tipo 2.	<i>Dialogflow, SDK de Google, WebHook y Node.js (JavaScript).</i>	La aplicación mejoró el estado actual de la técnica al aumentar la satisfacción y conveniencia del usuario.	Finalizado.
<b>Lazić et al. [16].</b>	Para utilizar la aplicación de TV basada en Android, es necesario detectar patrones que correspondan a comandos como subir/bajar canal, subir/bajar volumen, entre otros, sin embargo, el objetivo es mejorar la calidad de experiencia del usuario, por lo que no tiene que usar un control	Integración del asistente de Google en la aplicación de TV para Android OS.	<i>SDK de Google y SDK de Android.</i>	Permite al usuario administrar la aplicación de TV o el receptor de TV digital mediante comandos de voz.	Finalizado

Artículos	Problema	Contribución	Tecnologías	Resultados	Estado
	remoto, sino un comando de voz.				
<b>Mischie et al. [17].</b>	El uso de aparatos inteligentes es un campo que no se encuentra explorado en su totalidad y el cual proporciona una amplia gama de oportunidades de desarrollo, sin embargo, esto genera que también se consideren necesidades para los usuarios, los cuales buscan la comodidad y sencillez al utilizar los aparatos inteligentes.	Implementación de <i>Google Assistant</i> en Raspberry Pi para el apagado y encendido de leds mediante comandos de voz.	ASR, SLU, DM, KDB, NLG, TTS y <i>SDK</i> de Google.	La implementación funciona con cualquier idioma, sin embargo, solo uno a la vez.	Finalizado.
<b>Kěpuska y Bohouta [18].</b>	Muchas compañías han usado la tecnología de sistemas de diálogo para establecer varios tipos de VPA. Sin embargo, los VPA no cuentan con muchas funcionalidades que se requieren para tener un mejor control	Estructura para nuevo modelo de VPA.	TTS, SLU, NLG, DM, ASR.	El modelo de VPA utilizó voz, gráficos, video, gestos y otros modos de comunicación en los canales de entrada y salida. Además, el modelo de VPA se usó para aumentar la interacción entre los usuarios y las	Finalizado.



Artículos	Problema	Contribución	Tecnologías	Resultados	Estado
	sistemático en las compañías.			computadoras, de igual manera, este modelo permite una larga conversación con los usuarios mediante el uso de la amplia base de conocimientos de diálogo.	
<b>Dempsey [19].</b>	Google Home permite la interacción con el usuario para poder llevar a cabo tareas que solo requieren el comando de voz por parte del usuario, debido a que los desarrolladores encuentran dificultades al interactuar con el dispositivo y se ven en la necesidad de utilizar las tecnologías que estos mismos utilizan, sin poder tener otra alternativa.	Análisis de los componentes del dispositivo Google Home e impacto que tiene este en el mercado.	Componentes del dispositivo Google Home.	Se demostró que el dispositivo Google Home aunque todavía necesita crecer, tiene la infraestructura más grande en la actualidad y los beneficios de la portabilidad.	Finalizado.
<b>Reai et al. [20].</b>	Para las personas mayores, el aislamiento social es un factor importante en el deterioro	Modelo que sirve como guía para construir un sistema con IPA, que sirve	<i>Google Assistant</i> , Amazon Alexa, Apple Siri y Microsoft Cortana.	Se demuestra que podrían ejecutar las interacciones hasta cierto punto, pero no completamente como lo	Finalizado.

Artículos	Problema	Contribución	Tecnologías	Resultados	Estado
	de la calidad de vida. Tiene un profundo impacto en la salud general y se produce por la disminución de las interacciones sociales.	para apoyar las interacciones con las personas mayores.		requiere el modelo. Existe una brecha entre lo que pueden hacer las IPA y lo que efectivamente hacen.	
<b>Calvo et al [21].</b>	Incluso con la disponibilidad de varias pruebas para proporcionar claridad al elegir la trayectoria profesional, la decisión sigue siendo difícil de tomar. La mayoría de las pruebas disponibles son monótonas, lo que resulta en un esfuerzo tedioso para realizarlas por completo, o simplemente son aburridas.	Sistema de orientación profesional mediante la implementación del asistente de Google.	<i>Dialogflow</i> y <i>SDK de Google</i> .	El sistema proporcionó una interfaz fácil y amigable con una experiencia de usuario interactiva al tiempo que recopila los datos necesarios para proporcionar orientación profesional.	Finalizado.
<b>Višekruna y Savić [22].</b>	Surge dado que el uso de la tecnología de comando de voz está en aumento hoy en día. Los fabricantes de productos electrónicos permiten una	Integración del asistente de Google en la aplicación de Android para el control de voz para la	<i>SDK</i> del asistente de Google, Sistema Android.	Se determinó que la velocidad de ejecución del comando no cambiaba en relación con la velocidad cuando los usuarios emitían los comandos, la	Finalizado.

Artículos	Problema	Contribución	Tecnologías	Resultados	Estado
	nueva funcionalidad de voz y esto provoca que la necesidad de desarrollar nuevas soluciones por parte de los desarrolladores también está aumentando.	reproducción de medios.		única ventaja es que no había problemas para entender el idioma del asistente.	

Se concluye, después de haber analizado los anteriores artículos, que la mayor parte de estos utilizan las mismas tecnologías para el desarrollo de nuevos comandos de voz e instrucciones para el asistente de Google. En los trabajos [1], [6], [8], [9], [11] y [14] se atienden problemas de reconocimiento de nuevos comandos utilizando diferentes tecnologías en conjunto con el SDK del asistente Google para obtener así nuevos comandos e instrucciones. Para ello se debe hacer una comparación evaluando ventajas y desventajas de cada uno de ellos, para posteriormente tomar el más adecuado según ciertos criterios, implementarlo para obtener nuevos comandos e instrucciones. En los trabajos restantes se muestran comparativas y soluciones que se podrían dar si se emplearán los asistentes inteligentes en las diferentes áreas que existen en el mundo, desde un uso básico hasta como asistentes de adultos mayores para que tengan una mejor calidad de vida. Es por eso, que en este proyecto se propone el desarrollo e implementación de un módulo de nuevos comandos e instrucciones para el asistente de Google que no están presentes hoy en día.

### **2.3 Propuesta de solución**

Con el objetivo de establecer el marco de trabajo acorde a la propuesta de solución de la investigación, se hizo un análisis bajo criterios de funcionalidad y pertinencia de las principales herramientas y metodologías existentes. Se investigaron plataformas, metodologías para el desarrollo de software y se identificó el lenguaje de programación más idóneo para el desarrollo de los nuevos comandos de voz.

*Google Cloud Platform* [23] proporciona herramientas y soporte adicionales para entornos de desarrollo para aplicaciones. *Google Cloud Platform* contiene herramientas y bibliotecas que permiten crear y gestionar recursos, incluyendo facilidad de *App Engine*, *Compute Engine*, almacenamiento en la nube, *BigQuery*, *Cloud SQL*, y *Cloud DNS*. Se ejecuta en *Windows*, *macOS* y *Linux*, y requiere *Python 2.7.x*.

*Google Cloud Platform* (GCP) consiste en un conjunto de activos físicos, como computadoras y unidades de disco duro, y recursos virtuales, como máquinas virtuales (VM), que se encuentran en los centros de datos de *Google* en todo el mundo. Cada ubicación del centro de datos está en una región global. Las regiones incluyen Estados Unidos central, Europa occidental y Asia oriental. Cada región es una colección de zonas, que están aisladas

unas de otras dentro de la región. Cada zona se identifica con un nombre que combina un identificador de letra con el nombre de la región.

Esta distribución de recursos brinda varios beneficios, incluida la redundancia en caso de fallas y una latencia reducida al ubicar los recursos más cerca de los clientes. Esta distribución también introduce algunas reglas sobre cómo los recursos pueden usarse juntos.

*Dialogflow* [24] permite crear interfaces de conversación sobre los productos y servicios al proporcionar un potente motor de comprensión del lenguaje natural (NLU) para procesar y comprender las entradas de lenguaje natural.

Las interfaces de computadora tradicionales requieren una entrada estructurada y predecible para funcionar correctamente, lo que hace que el uso de estas interfaces sea poco natural y, a veces, difícil.

*Dialogflow* permite lograr fácilmente una experiencia de usuario conversacional al manejar la comprensión del lenguaje natural (NLU). *Dialogflow* crea agentes que pueden comprender los vastos y variados matices del lenguaje humano y lo traduce a un significado estándar y estructurado que las aplicaciones y servicios pueden comprender.

JavaScript es un lenguaje de programación ampliamente utilizado para la programación de las interfaces de los sitios web, por otro lado Node.js permite la aplicación de JavaScript en más contextos y particularmente en servidores web. Al igual que el código fuente de JavaScript del lado del cliente es interpretado y ejecutado por un motor *JavaScript*, *Node.js* utiliza un intérprete rápido *V8 JavaScript*, escrito en C++ creado por *Google*.

*Node.js* es un contenedor del alto rendimiento, el mismo que proporciona APIs adicionales optimizadas para los casos de uso específicos, como la manipulación de datos binarios, que no son soportados por *JavaScript*. *Node.js* no sólo proporcionará acceso directo a la ejecución de las aplicaciones en su intérprete *V8 JavaScript*, sino que además hace que *JavaScript* sea más útil en otros contextos, como el de residir del lado del servidor.

*JavaScript* es un lenguaje orientado a eventos, y *Node.js* utiliza esta orientación para producir servidores altamente escalables, reduciendo la complejidad de la concurrencia para el desarrollo de aplicaciones [25].

*Firebase* CLI [26] permite desplegar una *Action* en *Cloud Functions*, que se tratarán a continuación en el siguiente apartado. Se debe instalar localmente con el gestor de paquetes de *Node.JS* (npm). Sólo será necesario para los proyectos con al menos un *intent* (del inglés *intento*) cuya respuesta requiera ejecutar un *fulfillment*.

Sincroniza fácilmente los datos de sus proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja. Usa un conjunto de herramientas multiplataforma: se integra fácilmente para plataformas web como en aplicaciones móviles.

Es compatible con grandes plataformas, como *IOS*, *Android*, aplicaciones web, *Unity* y *C++*.

Usa la infraestructura de *Google* y escala automáticamente para cualquier tipo de aplicación, desde las más pequeñas hasta las más potentes.

Crea proyectos sin necesidad de un servidor: Las herramientas se incluyen en los *SDK* para los dispositivos móviles y web, por lo que no es necesario la creación de un servidor para el proyecto.

El proceso unificado ágil (AUP) [27] adopta una filosofía “en serie para lo grande” e “iterativa para lo pequeño” a fin de construir sistemas basados en computadora. Al adoptar las actividades en fase clásicas del proceso unificado (UP) —concepción, elaboración, construcción y transición—, el AUP brinda un revestimiento en serie (por ejemplo, una secuencia lineal de actividades de ingeniería de software) que permite que el equipo visualice el flujo general del proceso de un proyecto de software. Sin embargo, dentro de cada actividad, el equipo repite con objeto de alcanzar la agilidad y entregar tan rápido como sea posible incrementos de software significativos a los usuarios finales. Cada iteración del AUP aborda las actividades siguientes:

- *Modelado*. Se crean representaciones de UML de los dominios del negocio y el problema. No obstante, para conservar la agilidad, estos modelos deben ser “sólo suficientemente buenos” para permitir que el equipo avance.
- *Implementación*. Los modelos se traducen a código fuente.
- *Pruebas*. Igual que con la XP, el equipo diseña y ejecuta una serie de pruebas para detectar errores y garantizar que el código fuente cumple sus requerimientos.

- *Despliegue.* El despliegue en este contexto se centra en la entrega de un incremento de software y en la obtención de retroalimentación de los usuarios finales.
- *Configuración y administración del proyecto.* En el contexto del AUP, la administración de la configuración incluye la administración del cambio y el riesgo, y el control de cualesquiera productos del trabajo persistentes que produzca el equipo.

### 2.3.1 Justificación de la solución seleccionada

A continuación, se muestra la tabla 3.1, que presenta la solución a adoptar en este trabajo.

Tabla 3.2 Solución para desarrollo de los comando de voz.

Lenguaje de programación	Plataforma de desarrollo (IDE)	SDK de Google Assistant	Plataformas externas	Metodología
Node.JS (JavaScript)	Google Cloud	✓	Dialogflow Firebase CLI	AUP

La tabla 3.4 muestra la solución para desarrollar el comando de voz, esto es debido a que la plataforma *Google Cloud* cuenta con las herramientas suficientes para realizarlo, se utilizará JavaScript como lenguaje de programación nativo permitiendo una rápida implementación con los comandos de voz desarrollados con el uso de la plataformas externas de *Google* y se utilizará la metodología AUP debido al tiempo de desarrollo y al hecho de no ignorar demasiado la documentación, ya que al regirse bajo un enfoque ágil permite cierta flexibilidad para la inclusión de los artefactos que la ingeniería de software recomienda para sustentar los requerimientos y características de la solución planteada, enfocándose en la entrega del producto de software a desarrollar de forma iterativa e incremental sin abandonar la documentación.

## Capítulo 3. Aplicación de la metodología

En el presente capítulo se describe el diseño de los nuevos comandos de voz aplicando la metodología AUP. Esta metodología describe de una manera simple y fácil la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. Posteriormente se describe la parte de mayor importancia para la construcción de los nuevos comandos de voz para el asistente virtual de *Google*.

### 3.1 Fase de Modelado

Su objetivo es entender la lógica de negocio de los nuevos comandos, el dominio del problema del proyecto e identificar una solución viable para la creación de los nuevos comandos de voz para el asistente virtual de *Google*.

#### 3.1.1 Modelado de la lógica de negocio

A continuación, se muestra el proceso del funcionamiento de los nuevos comandos de voz para el asistente virtual de *Google*, mediante diagramas *UML* (Componentes, despliegue y actividades).

En la figura 3.1 se observan los componentes que intervienen al crear nuevos comandos de voz para el asistente virtual de *Google* logrando una simbiosis correcta y buen funcionamiento.

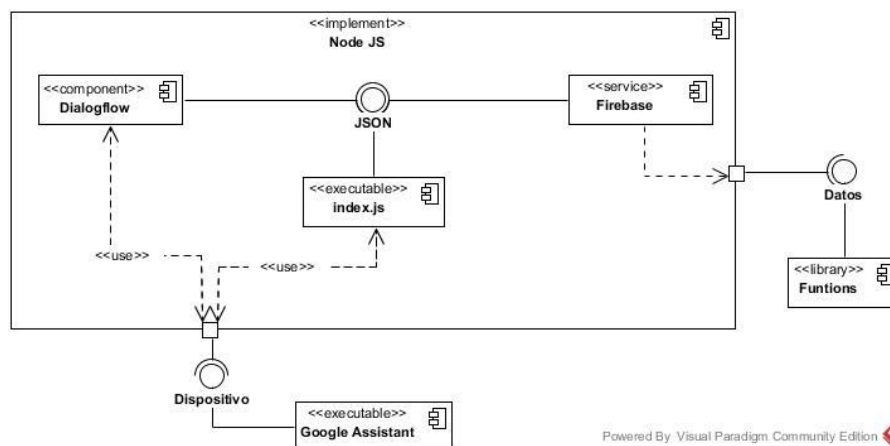


Figura 3.1 Diag. De Componente Google Assistant.



La plataforma de *Dialogflow*, *Firebase* y el archivo ejecutable *index.js* se comunican entre ellos utilizando como entrada y salida el formato de texto *JSON*; la plataforma de *Firebase* proporciona funciones que se encuentran en su biblioteca y permiten al archivo *index.js* una mayor comodidad al crear nuevas características para el comando, así como funciones extras.

La creación de estos comandos de voz se lleva a cabo bajo el entorno de desarrollo de *Node.js* el cual es altamente recomendado por *Google*.

Estos nuevos comandos se ejecutan en un dispositivo el cual debe contar con el asistente virtual de *Google* instalado para poder llevar a cabo la ejecución de los comandos

La figura 3.2 muestra los pasos en que se ejecuta un comando de voz para lograr un funcionamiento correcto.

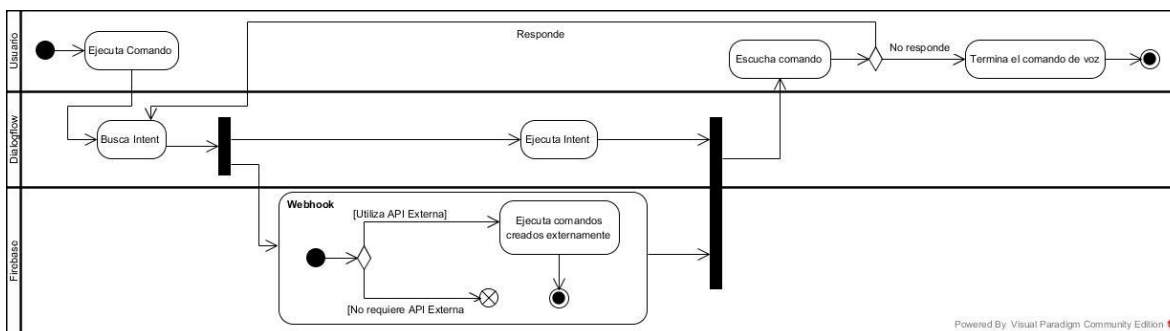


Figura 3.2 Diag. De Actividades Google Assistant

El usuario mediante el habla inicia el nuevo comando de voz, el asistente virtual de Google buscará en la plataforma de *Dialogflow* el *intent* que corresponda, dependiendo del *intent* que encuentre utilizará o no la plataforma de *Firebase*, en *Firebase* se hará uso de una API externa o no, se ejecutará el *intent* correspondiente y, una vez ejecutado el *intent*, el asistente virtual de *Google* enviará una respuesta adecuada, donde el usuario podrá proseguir a la respuesta del asistente virtual o quedarse callado. Si el usuario contesta se volverá a buscar otro *intent* que se adecue a lo dicho y ejecutará el flujo de nuevo, si el usuario no respondió terminara el comando de voz.

### 3.1.2 Arquitectura de desarrollo

La arquitectura que dará solución al desarrollo de los nuevos comandos de voz para el asistente virtual de *Google* se muestra en la figura 3; es importante destacar que dicha

arquitectura se construyó basándose en la misma que maneja *Google*, por lo que es necesario atender las actualizaciones que la empresa realice en las tecnologías usadas para el desarrollo del proyecto.

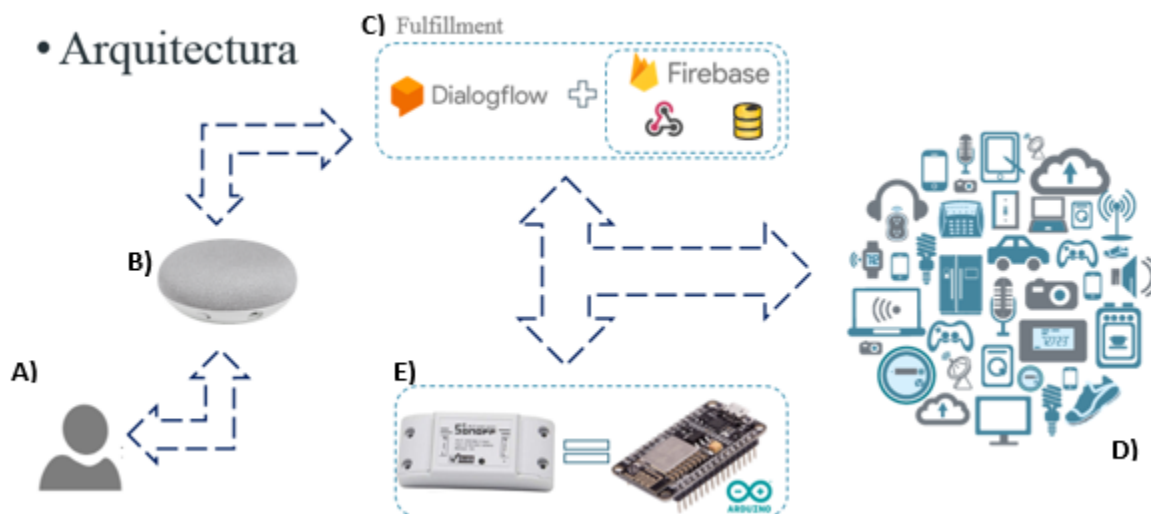


Figura 3.3 Arquitectura para el desarrollo de los nuevos comandos de voz.

**A) Usuario:** corresponde a quien utilizará el lenguaje natural para comunicarse con el dispositivo *Google Home Mini*, es importante resaltar que es necesario usar la frase “Ok Google” antes de alguna instrucción, para que el dispositivo reconozca cualquier cosa después de dicha oración como una orden.

La frase “OK Google”, ya se encuentra definida y es un punto clave para desencadenar la acción deseada; esta frase no se puede cambiar y es de uso exclusivo del *Google Assistant*.

**B) Google Home Mini:** el dispositivo obtendrá la orden expresada por el usuario, y se encontrará conectado a la plataforma de *Dialogflow*.

El *Google Home Mini* es el dispositivo físico por medio del cual se puede hacer uso del *Google Assistant*, dado que cuenta con hardware particular que permite una mejor captación de sonido en lugares más amplios, evitando el desfase de error por eco al momento de recibir la orden.

**C) Fullfillment:** es el servicio encargado de procesar el lenguaje natural de la instrucción que manda el usuario, su objetivo es encontrar en dicha instrucción, las palabras claves que

sirven para identificar el comando de voz que se requiere, su acción puede ser complementada con *WebHooks*, como se observó en [8] y [12].

El *Fullfillment* es donde se manejará la mayor parte de la lógica para lograr que el comando de voz pueda hacer la interacción y/o nuevas funcionalidades que se requieran para que este pueda lograr su objetivo.

La interacción con el dispositivo de *Google* se llevará a cabo mediante uno o varios *JSON* dependiendo de los comandos y/o acciones que se requieran. El *JSON* es un formato de texto sencillo el cual permite el intercambio de datos de manera más accesible logrando agilizar el proceso de comunicación.

En este *JSON* se mostrarán las acciones que ejecutará el comando de voz; además, el archivo *JSON* se apoyará del lenguaje de programación que admita gRPC para especificar funcionalidades extras y tener un buen funcionamiento. En esta sección se encuentra la mayor responsabilidad sobre la arquitectura.

**D) Sonoff y/o NodeMCU ESP8266:** es un dispositivo que actuará de intermediario para la conexión con los dispositivos externos; se planea controlar mediante comandos de voz.

Este dispositivo tiene la particularidad de permitir el entendimiento con el asistente virtual de *Google* y ofrecer un mejor control sobre los dispositivos externos como el *Raspberry Pi* [10].

Obtendrá todos los datos necesarios desde la plataforma *Firebase* y serán programados en el *IDE de Arduino* y se cargará dicha configuración al intermediario.

**E) Dispositivos externos:** estos realizarán una acción dependiendo del comando de voz que se haya ejecutado, por ejemplo, sea el caso de que este dispositivo esté representado por un Smartphone, una acción válida para tal, será encenderse o apagarse.

### 3.2 Fase de Implementación

Durante esta fase se codificará haciendo uso de los diagramas anteriores para tener una vista clara de la forma en que interactúan los componentes en conjunto con las plataformas.

La figura 3.4 muestra la interfaz que proporciona la plataforma *Dialogflow* de *Google*, ésta interfaz permite la implementación de *intent*, que son los que darán el flujo conversacional y respuesta a las preguntas de los usuarios dependiendo de las preguntas de estos.

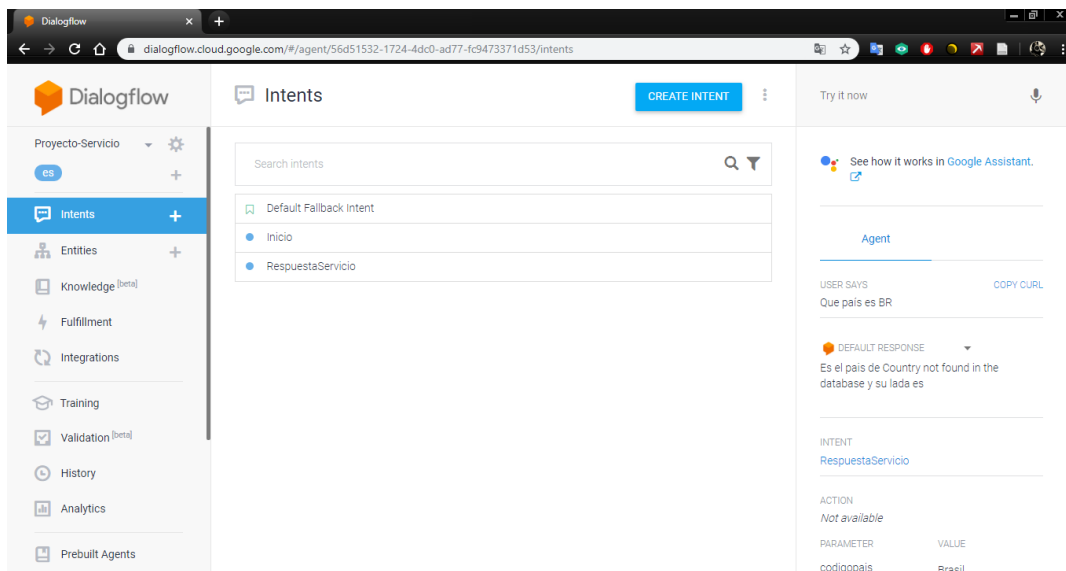


Figura 3.4 Interfaz de Dialogflow (Intents).

Dentro de cada *intent* se caracterizará el comando de voz de acuerdo a las necesidades que se requiera. Una de las funcionalidades que cuenta los *intents*, es que permitirán la obtención de valores predefinidos mediante el uso de variables que tiene la plataforma de *Dialogflow*, esto permitirá extraer la información necesaria y ayudará al entendimiento del agente a la hora de interactuar con el usuario como se muestra en la figura 3.5.

Los *intents* se pueden enlazar directamente a otro *intent* o código dependiendo de las necesidades del programador, también, es posible decir que lo que se quiere que el *intent* responda al momento de que es invocado, pero no es muy recomendado si es que se desea hacer que el comando de voz tenga más dinámica con el usuario o si se hará uso de diferentes tecnologías para que el comando de voz realice otras funcionalidades además de solo responder un texto plano y simple.

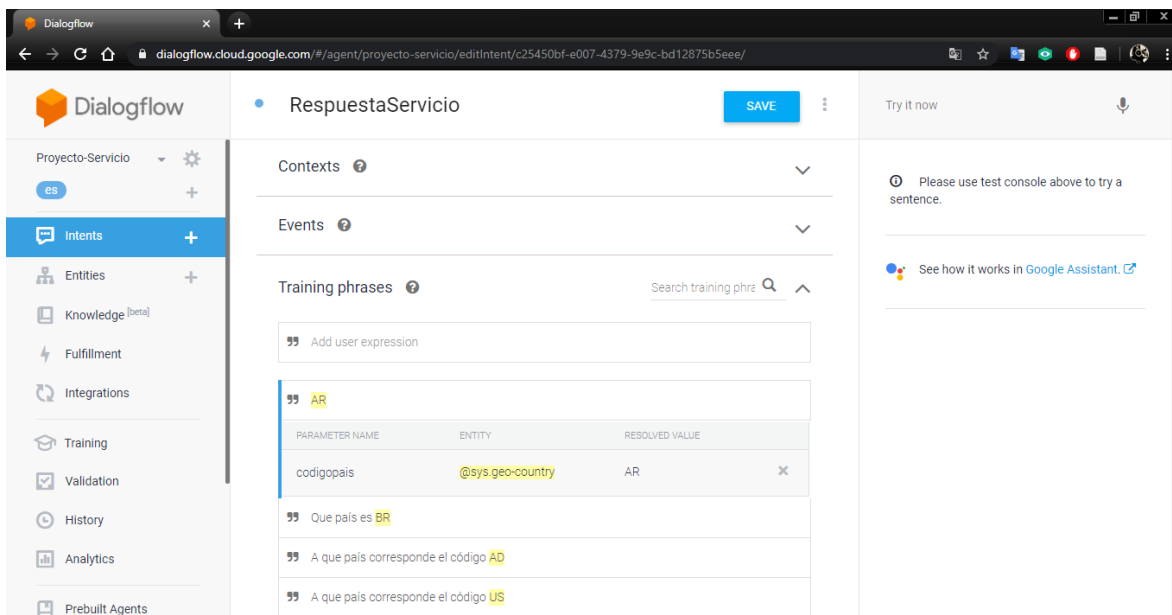


Figura 3.5 Interfaz de Dialogflow (Intents-parametros).

Los *intents* convertidos a *JSON* permiten una mejor visibilidad de cada una de sus funciones y logran introducir nuevas características como se muestran en la figura 3.6.

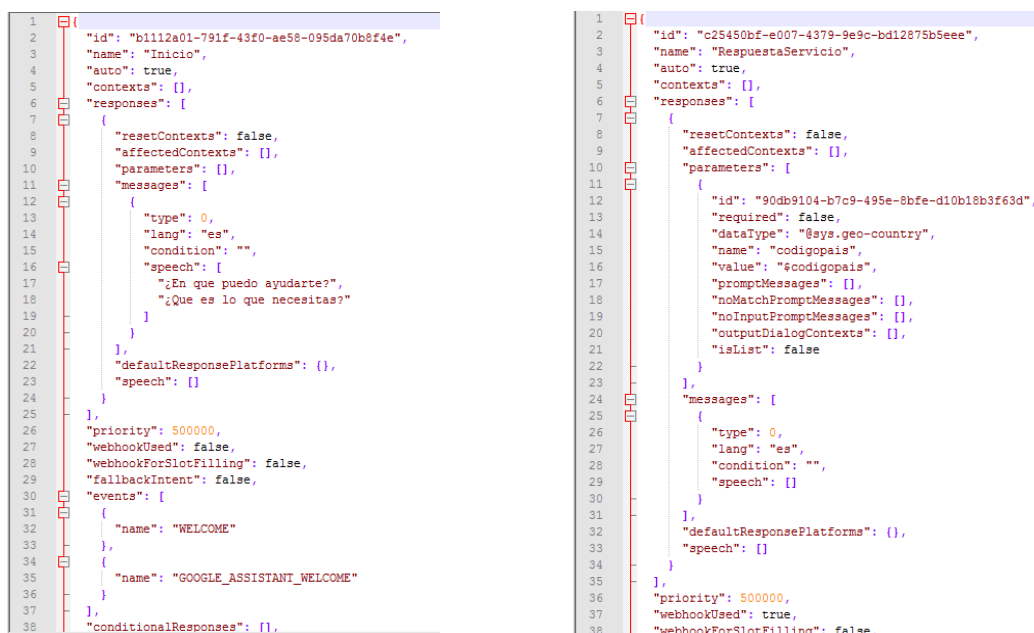


Figura 3.6 JSON (Intents-parametros).

Dentro de la plataforma de *Dialogflow* se podrán crear las propias entidades que van a servir de palabras reservadas para que el asistente virtual de Google comprenda mejor las interacciones que tiene con el usuario. También podrá identificar ciertas partes esenciales en

el desarrollador de frases dichas por el usuario para obtener el valor determinado y que el comando de voz ejecute la acción adecuada para los parámetros proporcionados como se muestra en la figura 3.7.

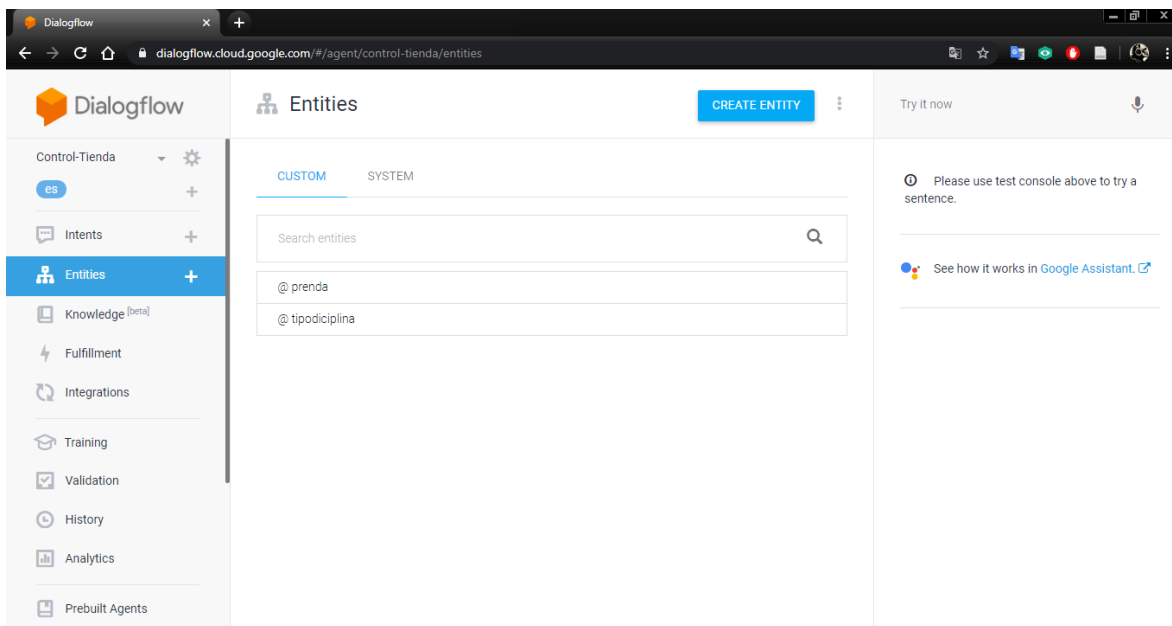


Figura 3.7 Interfaz de Dialogflow (Entities).

Al contar con la plataforma de *Dialogflow* para encontrar datos específicos, es posible extender las variables predefinidas haciendo uso de las funciones que nos proporciona la plataforma de *Firebase* las cuales se tienen que instalar en una computadora local debido a que *Google* propone esto para poder tener un mejor manejo sobre las funcionalidades de los *intents*.

Para ello se debe instalar el entorno de desarrollo de *Node.js* para poder ocupar las librerías de *Firebase* mediante los comandos que se expondrán más adelante; estos deben ser ejecutados a través del símbolo de sistemas; se debe ubicar el usuario en una carpeta vacía debido en que se instalarán las dependencias para hacer uso del comando de voz creado en *Google Assistant* y dar más funcionalidades al comando de voz.

A continuación, se presentarán los comandos para la instalación que (exceptuando al primero) deben realizarse mediante símbolo del sistema.

- Descarga el *Node.js* desde su página oficial, este contendrá el sistema de gestión de paquetes (NPM) para instalar las demás dependencias necesarias.

- Instalar el *Firebase* mediante el comando: `npm -g install firebase-tools`.
- Iniciar sesión en Firebase con el comando: `firebase login`.
- Entrar en la carpeta vacía.
- Ejecutar el comando: `firebase init functions`, para iniciar las funciones de *Firebase* y poder descargar el comando de voz.
- Seleccionar si desea crear una función o descargar una ya creada (Comando de voz).
- Seleccionar el tipo de lenguaje que utilizará para programar (*JavaScript* o *TypeScript*).
- Seleccionar la jerarquía de archivos a usar.
- Entrar a la carpeta creada llamada *functions*.
- Ejecutar el comando: `firebase use --project < ID del proyecto >` para hacer uso del comando de voz o proyecto.
- Instalar el paquete de *Google Assistant* usando el comando: `npm install actions-on-google`.
- Instalar los paquetes faltantes: `npm install`.
- Programar las funcionalidades de comando de voz.
- Desplegar y/o actualizar las funcionalidades del comando de voz una vez programadas usando el comando de algunos de los siguientes comandos de voz: `firebase deploy --project < ID del proyecto >` o `firebase deploy --only functions`

### 3.2.1 Implementación en Servicio Web

Una vez ya instalado todo lo necesario, se codificará para mostrar cada una de las partes de la realización del nuevo comando de voz para el asistente virtual de Google. La siguiente implementación consumirá un servicio web éste, proporciona el significado dependiendo de la palabra a buscar.

```
'use strict';
const http = require('http');
const functions = require('firebase-functions');
exports.ControlServicio = functions.https.onRequest((req, res) =>
{
```

```
    let valor = req.body.queryResult.parameters['palabra'];
console.log(valor);
    llamarServicio(valor).then((output) => {
      res.json({ 'fulfillmentText': output });
    }).catch(() => {
      res.json({ 'fulfillmentText': `Ocurrió un error` });
    });
  });
function llamarServicio (valor) {
  const options = {
    hostname: 'a044eaf5.ngrok.io',
    port: 80,
    path:
'/ProvDiccionarioRESTFUL/servi/miservicio/por_palabra/'+valor,
    method: 'GET',
    headers: {
      'Content-Type': 'application/json'
    }
  };
  console.log(options);
  return new Promise((resolve, reject) => {
    http.get(options, (res) => {
      let body = '';
      res.on('data', (d) => { body += d; });
      res.on('end', () => {
        let response = JSON.parse(body);
        console.log(response);
        let output = `${response['Definicion']}`;
        console.log(output);
        resolve(output);
      });
    });
  });
}
```



```

    });
    res.on('error', (error) => {
        console.log(`Error al llamar el servicio: ${error}`)
        reject();
    });
});
});
}

```

Figura 3.8 Fragmento del archivo index.js para servicio web.

En la figura 3.8 se muestra el archivo index.js como solución al consumo del servicio, el cual será mostrado como respuesta cuando el usuario pregunte por el significado de una palabra. Este archivo recupera el significado mediante la palabra proporcionada por el usuario, la palabra será almacenado en una variable, a través de una función se hace un llamado de tipo *GET* al servicio usando el protocolo HTTP, la variable servirá para poder completar el enlace y obtener los datos del país como se muestra en el texto de la figura 3.9.

```

ControlServicio
  ▶ caballo
ControlServicio
  - { hostname: 'a044eaf5.ngrok.io',
      port: 80,
      path: '/ProvDiccionarioRESTFUL/servi/miservicio/por_palabra/caballo',
      method: 'GET',
      headers: { 'Content-Type': 'application/json' } }
ControlServicio
  - { Palabra: 'caballo',
      Definición: 'Animal mamífero macho de tamaño mediano o grande. Tiene la cabeza alargada y las orejas pequeñas. Las uñas se llaman cascos. Tiene cola y pelo largo en el cuello. Se domestica con facilidad. La hembra del caballo es la yegua.' }

```

Figura 3.9 Debug del consumo del servicio 1.

En la figura 3.10 se muestra el *debug* del comando de voz usando el servicio, en la primera fila se muestra la palabra que el usuario le dijo al asistente virtual que buscara, mientras que en la segunda fila se exponen los parámetros que se utilizan para hacer uso del servicio (*hostname, port, path, method, headers*).

```
  ▶ caballo
  ⓘ ControlServicio
  ▶ { hostname: 'a044eaf5.ngrok.io',
    port: 80,
    path: '/ProvDiccionarioRESTFUL/servi/miservicio/por_palabra/caballo',
    method: 'GET',
    headers: { 'Content-Type': 'application/json' } }
```

*Figura 3.10 Debug del consumo del servicio 2.*

Debido a que el asistente virtual de Google se comunica utilizando el formato *JSON*, el tipo de datos que este debe recibir deben estar en el mismo formato. La figura 3.11 muestra los datos de la palabra en *JSON*, que son utilizados como respuesta y en la última fila se muestra la salida del asistente virtual, la cual será presentada como respuesta, completando el comando de voz.

```
  ⓘ ControlServicio
  ▶ { Palabra: 'caballo',
    Definicion: 'Animal mamífero macho de tamaño mediano o grande. Tiene la cabeza alargada y las orejas pequeñas. Las uñas se llaman cascos. Tiene cola y pelo largo en el cuello. Se domestica con facilidad. La hembra del caballo es la yegua.' }
```

```
  ⓘ ControlServicio
  ▶ Animal mamífero macho de tamaño mediano o grande. Tiene la cabeza alargada y las orejas pequeñas. Las uñas se llaman cascos. Tiene cola y pelo largo en el cuello. Se domestica con facilidad. La hembra del caballo es la yegua.
```

*Figura 3.11 Debug del consumo del servicio 3.*

Para que la ejecución del comando de voz se lleve de manera correcta, se debe enlazar el archivo `index.js` con la plataforma de *Dialogflow (Fulfillment)*, efectuando así las funcionalidades y/o actualizaciones realizadas para el comando de voz como se muestra en la figura 3.12.

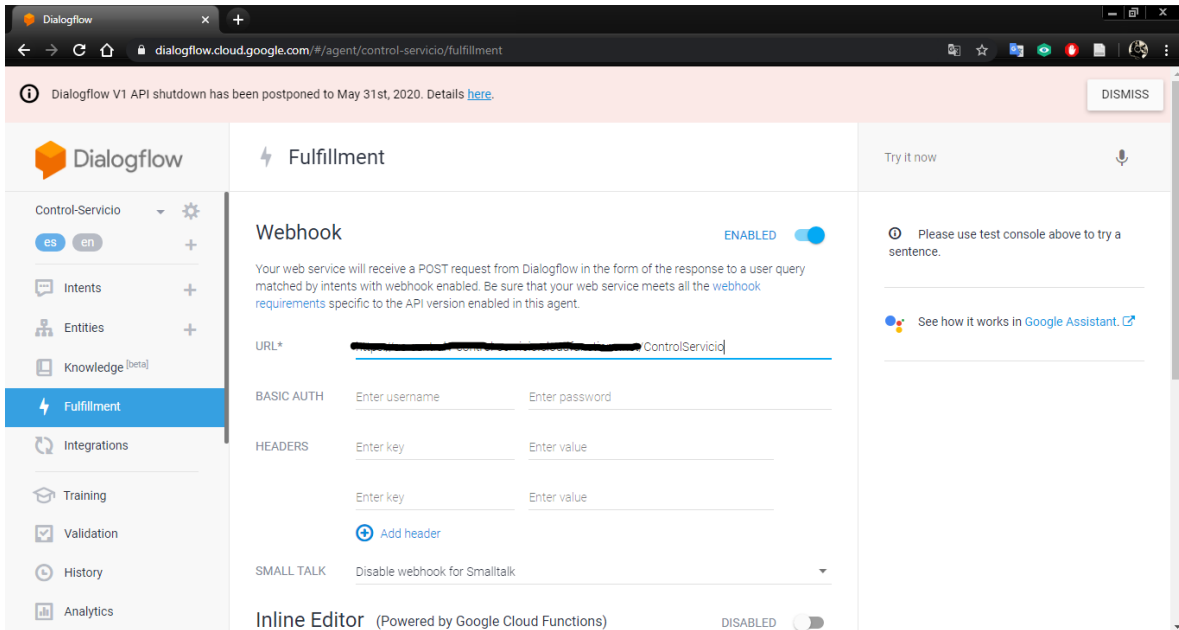


Figura 3.12 Interfaz de Dialogflow (Fulfillment) del Servicio Web.

### 3.2.2 Implementación en dispositivos externos

La siguiente implementación se hará mediante el uso de dispositivos externos, por lo que es necesario el uso del archivo `index.js`, a continuación, en la figura 3.13 se muestra cómo se codificó este.

```
'use strict';
const functions = require('firebase-functions');
const admin = require('firebase-admin');
admin.initializeApp(functions.config().firebase);
var database = admin.database();
let res, accionSala, accionCocina, accionAlarma, lugarSala,
lugarCocina, lugarAlarma;

exports.controlSeminario = functions.https.onRequest((request,
response) => {
  let accion = request.body.queryResult.parameters['accion'];
  let lugar = request.body.queryResult.parameters['lugar'];
  let alarma = request.body.queryResult.parameters['alarma'];
```

```
if (accion == "encender") {
  console.log("Entro a encender ");
  if (lugar == "sala" & alarma == "") {
    res = "La luz de la sala se encendió";
    accionSala = "encender";
    console.log("Entro a SALA " + res + accionSala);
    database.ref().update({
      accionsala: accionSala
    });
  }else if (lugar == "cocina" & alarma == "") {
    res = "La luz de la cocina se encendió";
    accionCocina = "encender";
    console.log("Entro a COCINA " + res + accionCocina);
    database.ref().update({
      accioncocina: accionCocina
    });
  }else if (lugar == "todo" & alarma == "") {
    res = "La luz de la casa se encendió";
    accionCocina = "encender";
    accionSala = "encender";
    console.log("Entro a COCINA " + res + accionCocina);
    database.ref().update({
      accioncocina: accionCocina,
      accionsala: accionSala
    });
  }else {
    res = "La alarma se encendió";
    accionAlarma = "encender";
    database.ref().update({
      accionalarma: accionAlarma
```

```

    });
  }
}else {
  console.log("Entro a apagar ");
  if (lugar == "sala" & alarma == "") {
    res = "La luz de la sala se apagó";
    accionSala = "apagar";
    console.log("Entro a SALA " + res + accionSala);
    database.ref().update({
      accionsala: accionSala
    });
  }else if (lugar == "cocina" & alarma == "") {
    res = "La luz de la cocina se apagó";
    accionCocina = "apagar";
    console.log("Entro a COCINA " + res + accionCocina);
    database.ref().update({
      accioncocina: accionCocina
    });
  }else if (lugar == "todo" & alarma == "") {
    res = "La luz de la casa se apagó";
    accionCocina = "apagar";
    accionSala = "apagar";
    console.log("Entro a COCINA " + res + accionCocina);
    database.ref().update({
      accioncocina: accionCocina,
      accionsala: accionSala
    });
  }else {
    res = "La alarma se apagó ";
    accionAlarma = "apagar";
    database.ref().update({

```

```

        accionalarma: accionAlarma
    });
}
}
response.json({
    'fulfillmentText': res + ' correctamente'
});
});

```

Figura 3.13 Fragmento del archivo index.js para dispositivos externos.

El archivo index.js se programó con un gran esfuerzo, debido a que hace uso de la plataforma de *Firebase* para el almacenamiento de variables, ya que los dispositivos externos cambiarán el estado de estas.

Para que la ejecución del comando de voz se lleve de manera correcta, se debe enlazar el archivo index.js con la plataforma de *Dialogflow (Fulfillment)* para que se efectúen las funcionalidades y/o actualizaciones que indica el comando de voz, tal y como se muestra en la figura 3.14.

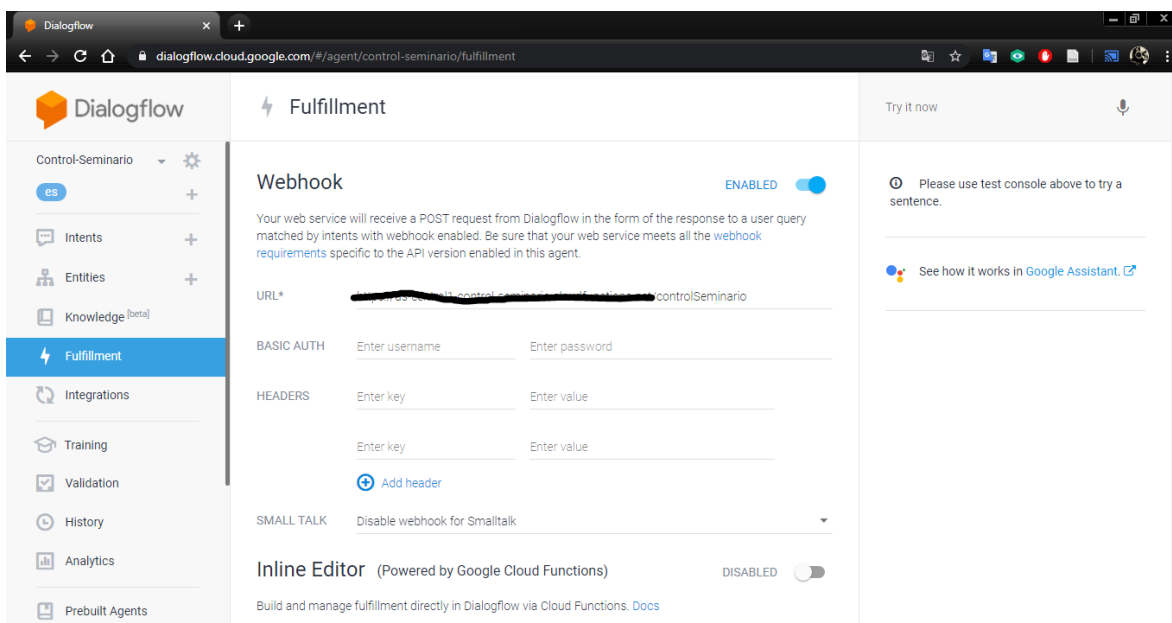


Figura 3.14 Interfaz de Dialogflow (Fulfillment) para dispositivos externos.

Las variables que se almacenan en la base de datos de *Firebase* en tiempo real se visualizan en la figura 3.15

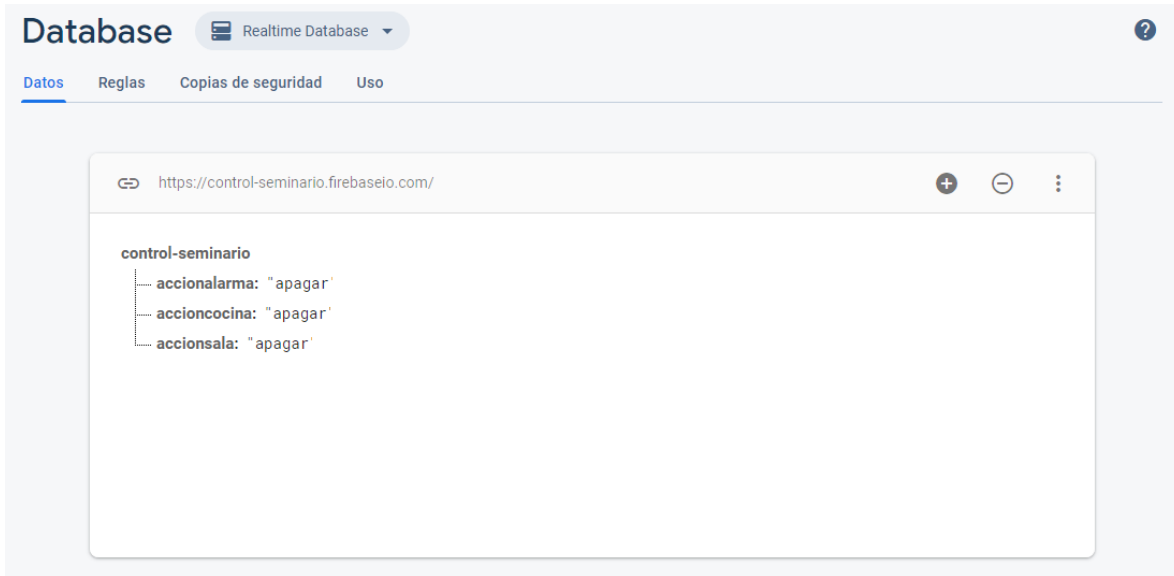


Figura 3.15 Variables almacenadas en *Firebase* para dispositivos externos.

Una vez que el comando de voz envía y actualiza las variables de estado en la plataforma de *Firebase*, es necesario instalar el IDE de Arduino, ya que este nos permitirá programar de manera directa la placa *NodeMCU ESP8266*, representada en la figura 3.16.

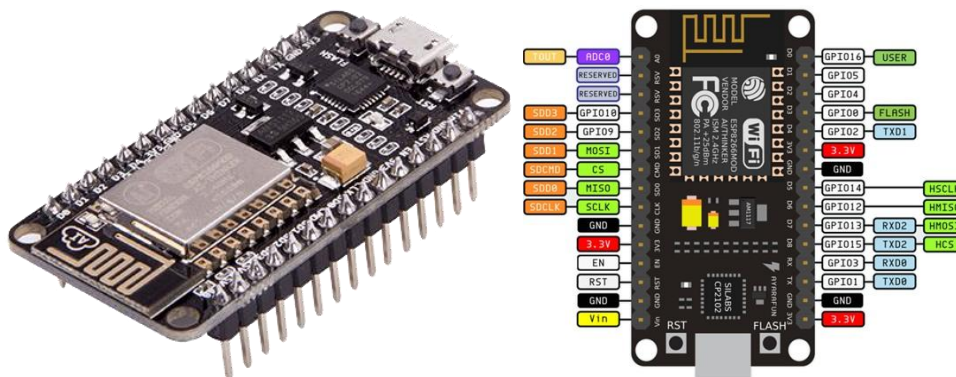


Figura 3.16 Placa *NodeMCU ESP8266*.

En la figura 3.17 se muestra el esquema de conexión del control de los dispositivos externos.

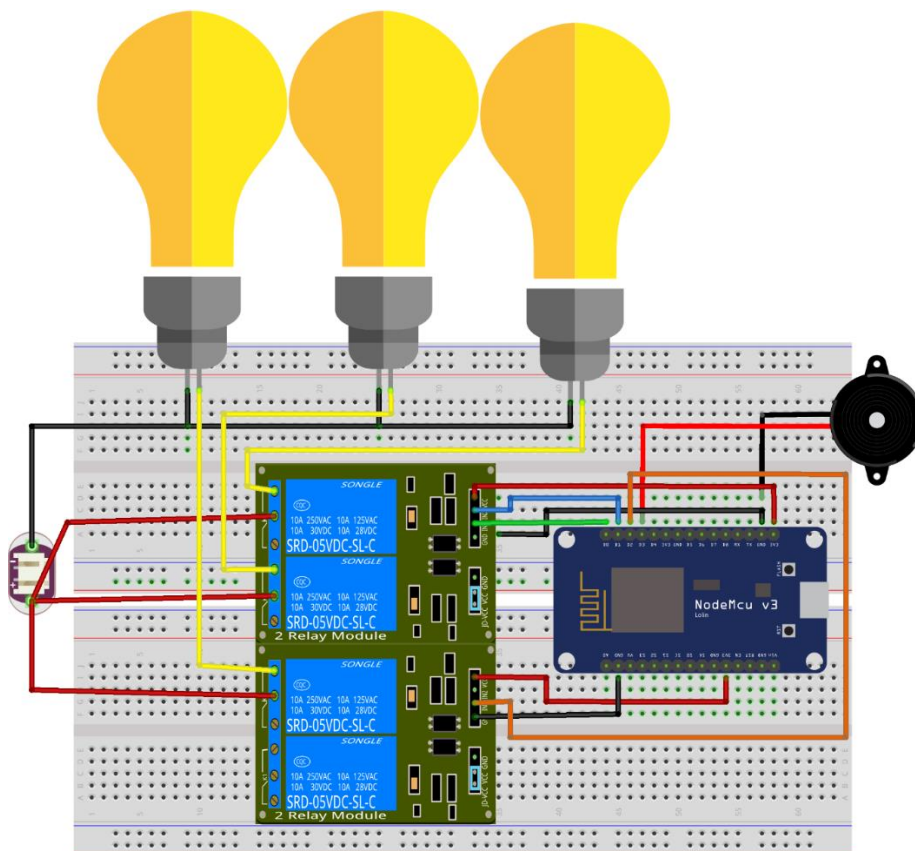


Figura 3.17 Esquema de conexión.

Una vez hecha la conexión de los circuitos y componentes, se programa la placa *NodeMCU ESP8266* como se muestra la figura 3.18

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#define FIREBASE_HOST "Nombre del dominio de Firebase"
#define FIREBASE_AUTH " Contraseña del dominio de Firebase "
#define WIFI_SSID "Nombre de la red"
#define WIFI_PASSWORD " Contraseña de la red "
String status;
String focoSala;
String focoCocina;
String focoAlarma;
void setup() {
```



```
Serial.begin(9600);

pinMode(2, OUTPUT);
pinMode(5, OUTPUT);
pinMode(4, OUTPUT);
pinMode(0, OUTPUT);

WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
Serial.print("INICIANDO CONEXION CON EL INTERNET");
while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
}
Serial.println();
Serial.print("CONECTANDO: ");
Serial.println(WiFi.localIP());

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
int n = 0;
void loop() {
  focoSala = Firebase.getString("accionsala"); //recupera el valor
del parametro accion
  focoCocina = Firebase.getString("accioncocina"); //recupera el
valor del parametro accion
  focoAlarma = Firebase.getString("accionalarma"); //recupera el
valor del parametro accion
Serial.println("+++++++");
Serial.println(focoSala);
Serial.println(focoCocina);
Serial.println(focoCocina);
```

```

Serial.println("+++++++");
  if(focoCocina == "apagar"){
    digitalWrite(4,LOW);
    Serial.println("FOCO COCINA APAGADO");
  }else{
    digitalWrite(4, HIGH);
    Serial.println("FOCO COCINA ENCENDIDO");
  }
  if(focoSala == "apagar"){
    digitalWrite(2,LOW);
    Serial.println("FOCO SALA APAGADO");
  }else{
    digitalWrite(2,HIGH);
    Serial.println("FOCO SALA ENCENDIDO");
  }
  if(focoAlarma == "apagar"){
    digitalWrite(5,LOW);
    digitalWrite(0,LOW);
    Serial.println("ALARMA APAGADA");
  }else{
    digitalWrite(5, HIGH);
    digitalWrite(0,HIGH);
    Serial.println("ALARMA ENCENDIDA");
  }
  return;
}

```

Figura 3.18 Programacion del NodeMCU ESP8266.

Dentro del archivo se deben usar las bibliotecas que necesita *Firebase* para tener comunicación con el *IDE* de *Arduino*. Se debe crear un conexión con una red que proporcione Internet, a su vez conectar en el proceso con la base de datos de *Firebase* la cual cuenta con un dominio así como una contraseña de protección para evitar filtraciones de información.

Una vez hechas las conexiones necesarias, se debe obtener las variables almacenadas de la figura 3.5, para que la lógica de encendido y apagado se lleve a cabo de forma adecuada.

La placa *NodeMCU ESP8266* estará recuperando los valores almacenados en la base de datos de *Firebase* en tiempo real, esto provocará que cada vez que haya un cambio provocado por la interacción entre el usuario y el asistente virtual de *Google* usando el comando de voz se efectúe al instante sin ninguna demora.

### 3.3 Fase de Pruebas

En este apartado se realizará una evaluación de los objetivos para asegurar la calidad del producto de tesis, encontrar defectos, validar que el sistema funciona como fue diseñado y verificar que los requisitos se cumplen.

Sin embargo, la parte medular de esta sección, es demostrar la funcionalidad de los nuevos comandos de voz para el asistente virtual de *Google*, por lo que se hizo uso de la plataforma de *Google*, ya que nos permite utilizar un simulador con dicho asistente como se muestra en la figura 3.19,

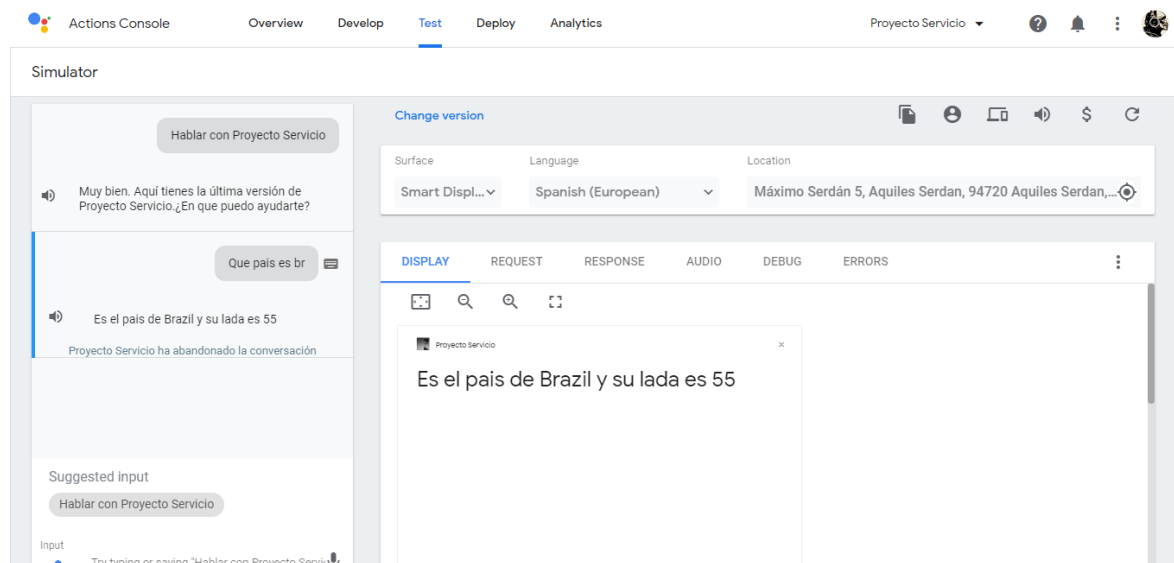


Figura 3.19 Simulador para el asistente virtual de Google.

El simulador permite observar errores y fallos a la hora de ejecutar comandos de voz, de igual manera nos muestra una vista previa del funcionamiento de estos, las solicitudes y respuestas.

Todas estas secciones se muestran con formato *JSON* para que se comprendan mejor. En la figura 3.20 se muestra un fragmento de la respuesta del simulador en dicho formato,

```

DISPLAY    REQUEST    RESPONSE    AUDIO    DEBUG    ERRORS
-----
{
  "responseId": "ce835cf1-0663-4b5d-8aea-c2311a65c4b3-415208ed",
  "queryResult": {
    "queryText": "Que pais es br",
    "parameters": {
      "codigopais": "BR"
    },
    },
  "allRequiredParamsPresent": true,
  "fulfillmentMessages": [
    {
      "text": {
        "text": [
          ""
        ]
      }
    }
  ]
}
    
```

Figura 3.20 Fragmento de solicitud del simulador.

También es posible probar estos comandos de voz en algún dispositivo que cuente con el asistente virtual de *Google* instalado.

### 3.4 Fase de Despliegue, Fase de Configuración y Administración de Proyecto

Durante esta fase se llevan una serie de pasos que son realizados a través de la plataforma de *Google* para el despliegue de los nuevos comandos de voz.

Antes de un despliegue *Google* se debe llenar un formulario con las características del proyecto como se muestra en la figura 3.21. En este formulario se debe llenar la siguiente información.

- ➔ Información del directorio: abarca desde una descripción simple hasta las políticas de privacidad que maneje el desarrollador del comando de voz.
- ➔ Ubicación: es en donde será desplegado el comando y las capacidades de este.
- ➔ Compatibilidad: Se decide si el comando de voz requiere una mayor especificación y/o dispositivos que contengan el asistente virtual de *Google* instalado de fábrica (Dispositivos electrónicos).

- ➔ Detalles de la empresa: Nombre, datos del desarrollador para poder contactarse con él si surge un problema con el comando de voz, entre otros.
- ➔ Verificación y publicación.

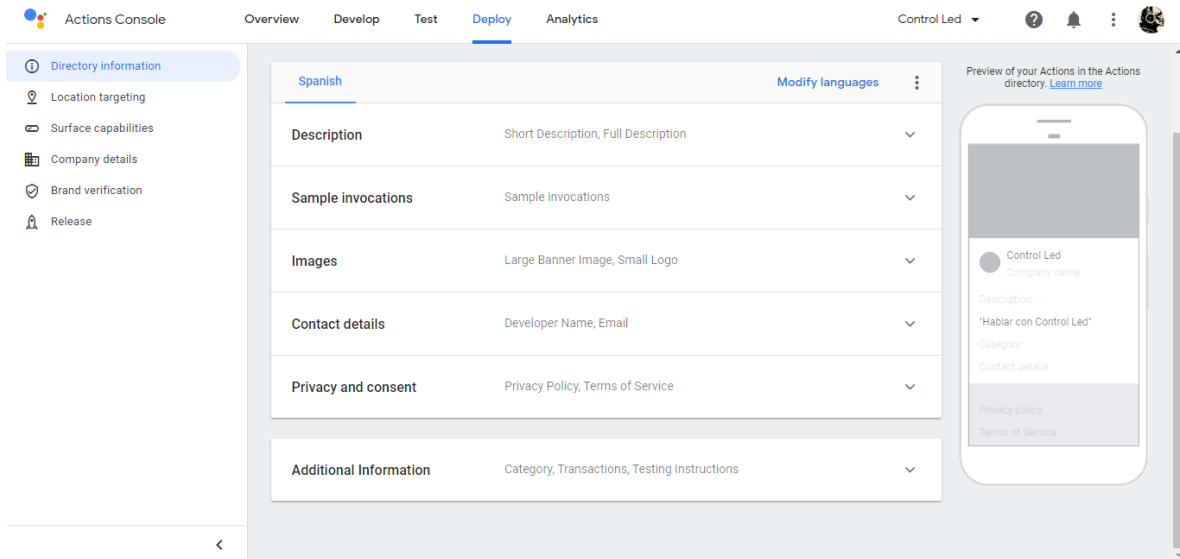


Figura 3.21 Configuración general.

## Capítulo 4. Resultados

En este capítulo se describen los resultados obtenidos con la realización del presente proyecto, para lo cual, se presentan dos casos de estudio relacionados con el ámbito de la creación de nuevos comandos de voz para el asistente virtual de *Google*.

### 4.1 Casos de estudio

El primer caso de estudio abarca el funcionamiento de diferentes dispositivos utilizando una única placa de desarrollo para su funcionamiento mediante un comando de voz de *Google Assistant*, dicho comando será capaz de procesar y ejecutar las órdenes de usuario final para el encendido y apagado de diversos dispositivos, así como la activación de una alarma en caso de peligro.

La finalidad del comando de voz es que el usuario tenga el control de los dispositivos y que estos a su vez puedan ejecutar otras instrucciones internas además de las de encendido y apagado.

En el segundo caso de estudio, el nuevo comando de voz de *Google* buscará la definición de cualquier palabra, esto será mediante un servicio web, todas las palabras y sus respectivas definiciones se encontrarán almacenadas en una base de datos de PostgreSQL.

#### **4.1.1 Caso de estudio: Encendido y apagado de diversos dispositivos externos mediante un nuevo comando de voz de Google Assistant.**

Mayormente, el uso de dispositivos externos se encuentra limitado por solo una acción, esto quiere decir que, cuando el usuario invoca un comando mediante “OK Google” seguido de la acción que desea, esta se llevará a cabo únicamente en un solo dispositivo, por lo que en este caso de estudio se logró superar esta limitante debido a que un comando de voz se ejecuta en más de un dispositivo externo.

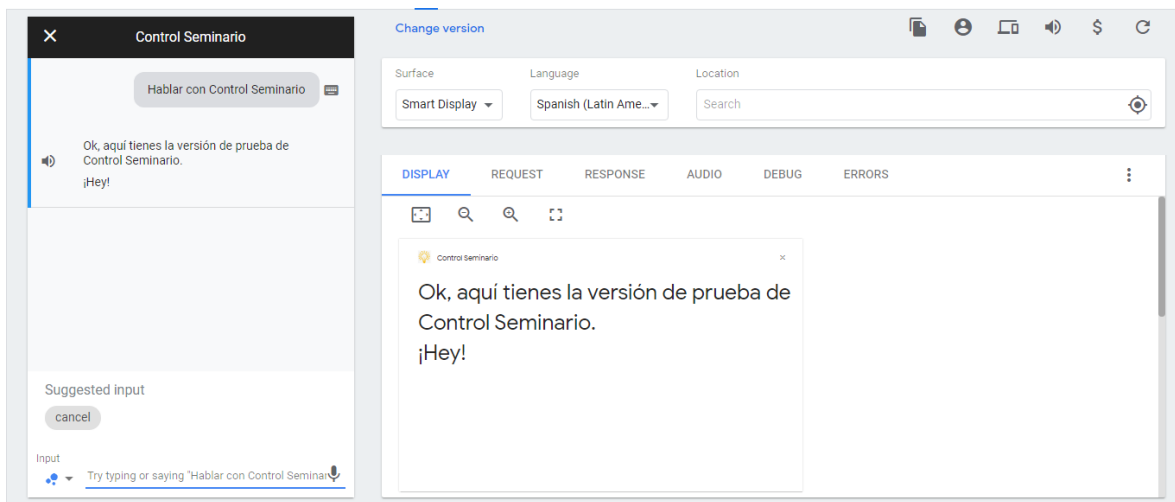


*Figura 22 Esquema de conexión realizado.*

La figura. 4.1 nos permite entender el funcionamiento y el control del *NodeMCU ESP8266* sobre cada uno de los dispositivos externos (Foco de la sala, Foco de la cocina, Foco de la alarma, Zumbador de la alarma).

Se optó usar el esquema de la figura 3.17 debido a que así se tiene un mejor control sobre cada dispositivo, además de que es posible contralar cada acción sobre esté y así se evitan errores al momento de ejecutar acciones.

Para iniciar el comando de voz es necesario que el usuario pronuncie la palabra clave “OK Google” para desencadenar la acción del comando, esto se demuestra en la figura 4.2.

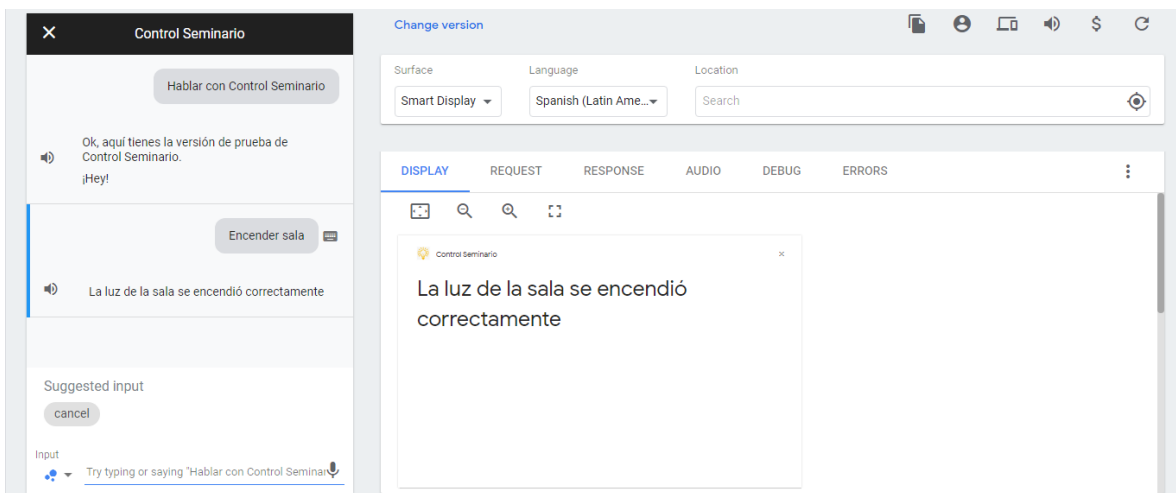


*Figura 23 Inicio del comando del comando de voz mediante simulador de Google.*

Una vez que el comando de voz es reconocido, el usuario tendrá total control de los dispositivos disponibles.

Se hace uso del simulador de Google para interactuar con el comando de voz, y cabe resaltar que cualquier dispositivo que cuente con el asistente virtual de Google podrá hacer uso del comando de voz.

Suponiendo que el usuario enciende la luz de la sala, el dispositivo con el asistente virtual de Google responderá de manera visual y/o mediante voz para permitir una interacción clara y concisa con el usuario.



*Figura 24 Encendido de la luz de la sala mediante simulador de Google.*

De la misma forma que el usuario se comunicó con el asistente virtual de Google, para encender la luz de la sala figura 4.3, también podrá apagarla como se muestra en la figura 4.4.



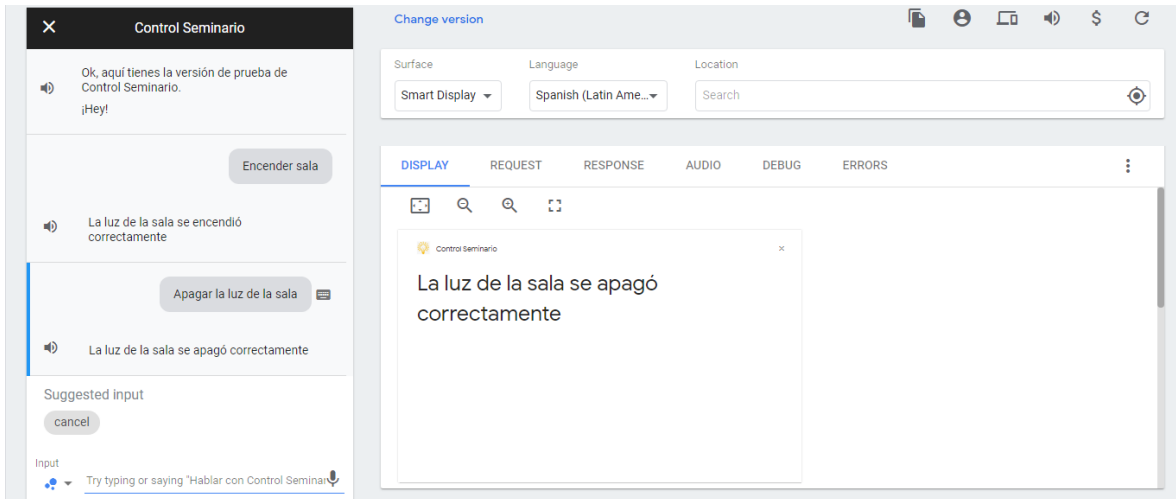


Figura 25 Apagado de la luz de la sala mediante simulador de Google.

El comando de voz no se dejará de ejecutar; esto permite al usuario poder encender otro dispositivo que se encuentre a su disposición, proporcionando el control de más de un dispositivo y otorgando una gran flexibilidad en el manejo de estos.

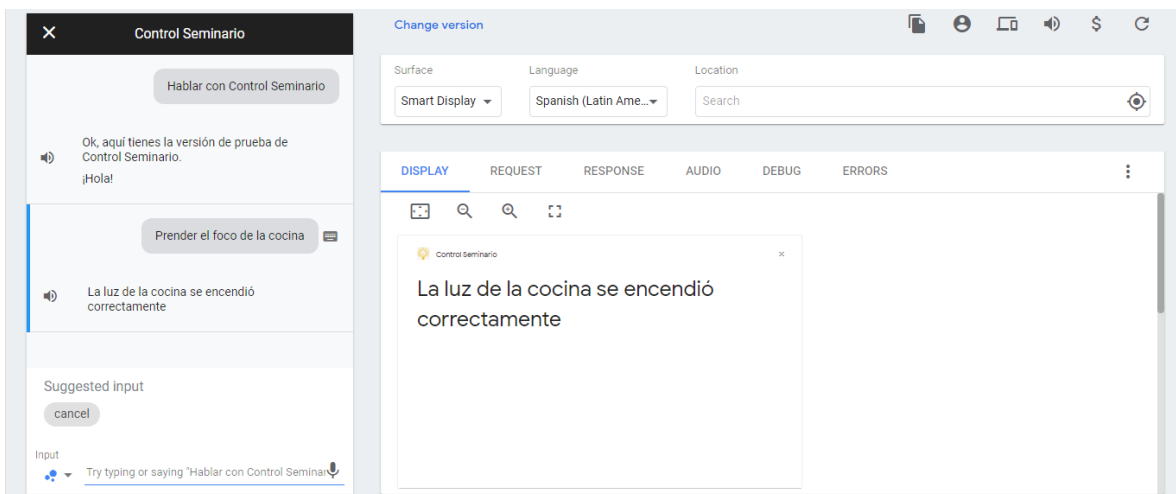
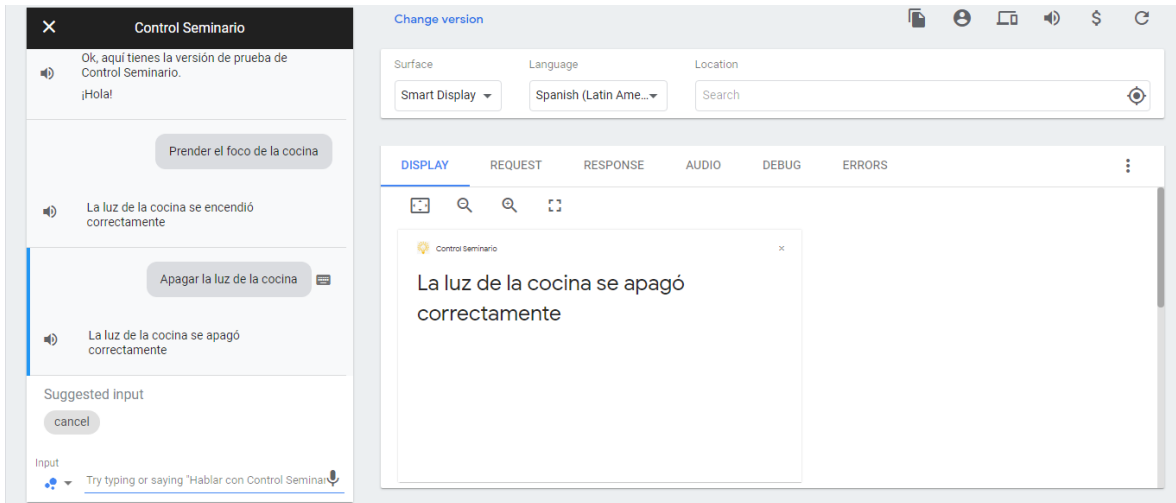
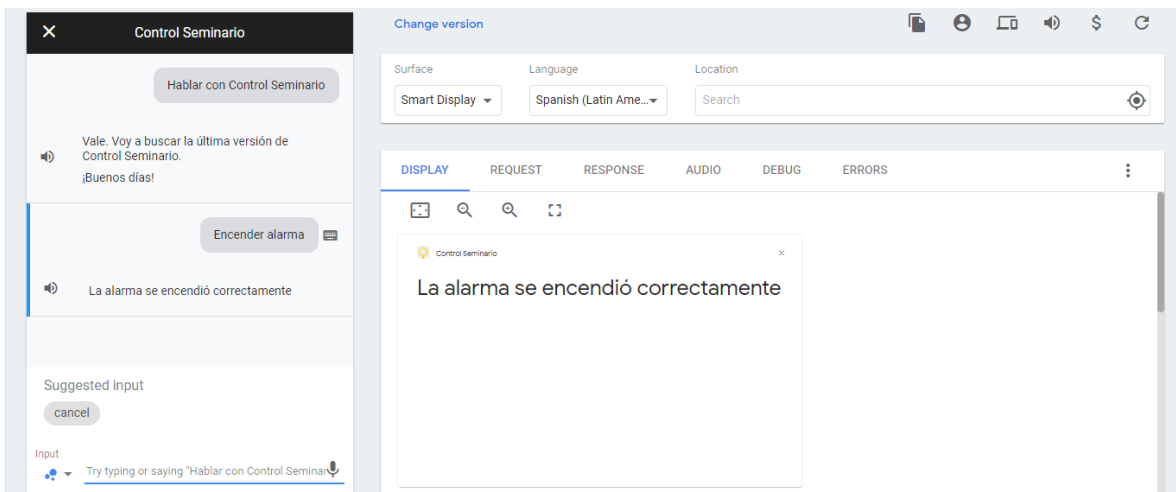


Figura 26 Encendido de la luz de la cocina mediante simulador de Google.



*Figura 27 Apagado de la luz de la sala mediante simulador de Google.*

El usuario tiene a su disposición un comando de voz para pánico, dicho comando hará un encendido total de todas las luces y de una alarma que sonará a toda potencia. Esta es una característica destacable debido a que muchas asistentes virtuales no cuentan con un encendido total de todos los dispositivos en un solo comando de voz como se muestra en la figura 4.7.



*Figura 28 Encendido de la alarma mediante simulador de Google.*

De la misma forma que el usuario puede encender un foco de un lugar específico del hogar, este puede apagar la alarma como se muestra en la figura 4.8.

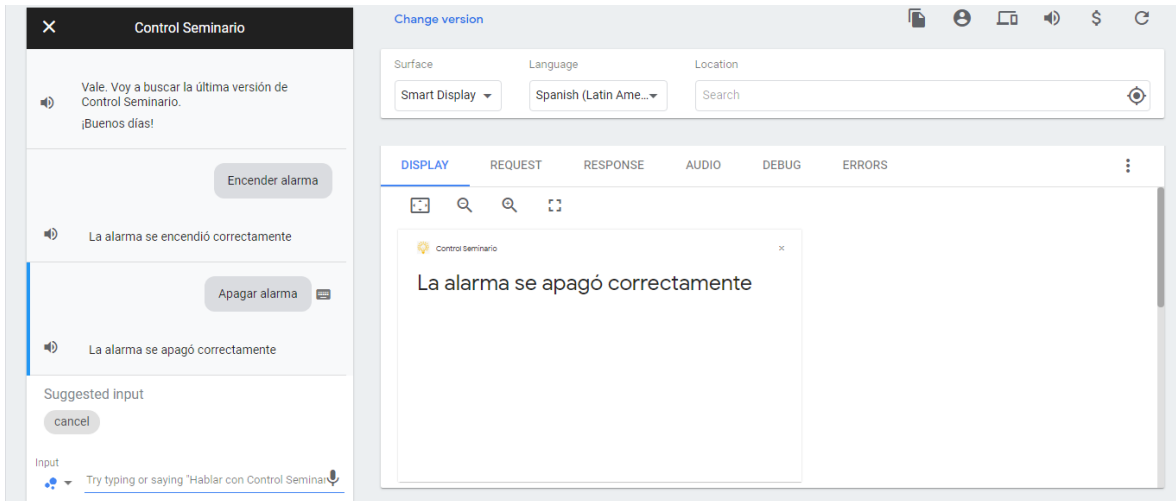


Figura 29 Apagado de la alarma mediante simulador de Google.

A continuación, se muestran una serie de imágenes que demuestran el funcionamiento de los comandos en cualquier dispositivo que cuente con el asistente virtual de Google figura 4.9.

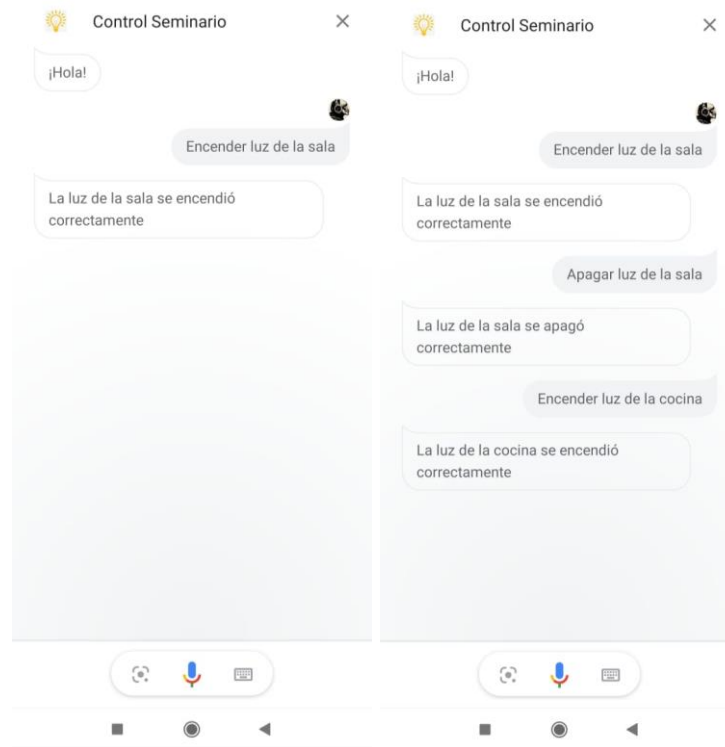


Figura 30 Funcionamiento de los comando de voz en dispositivo Móvil parte 1.

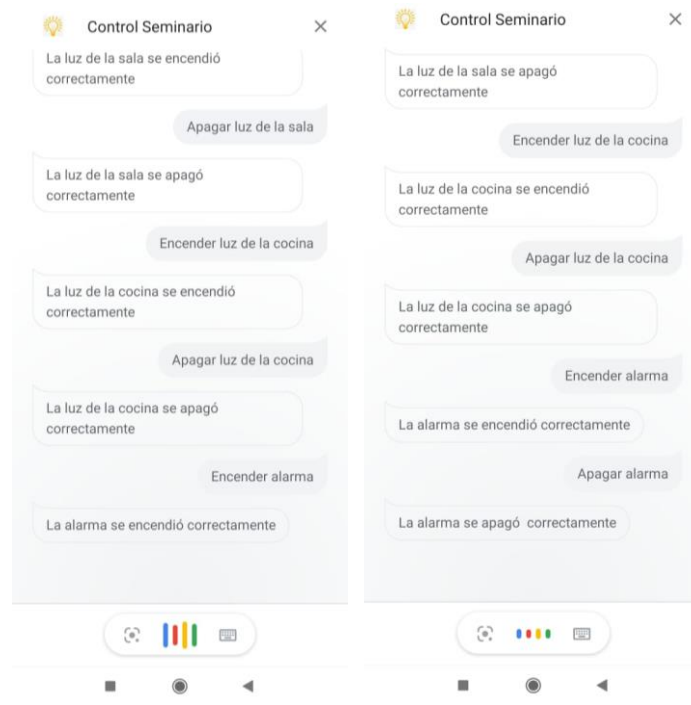


Figura 31 Funcionamiento de los comando de voz en dispositivo Móvil parte 2.

Si siguiendo con la demostración del uso de los nuevos comandos de voz, se presentarán las imágenes que demuestran el funcionamiento de los dispositivos externos y la forma en que estos interactúan.

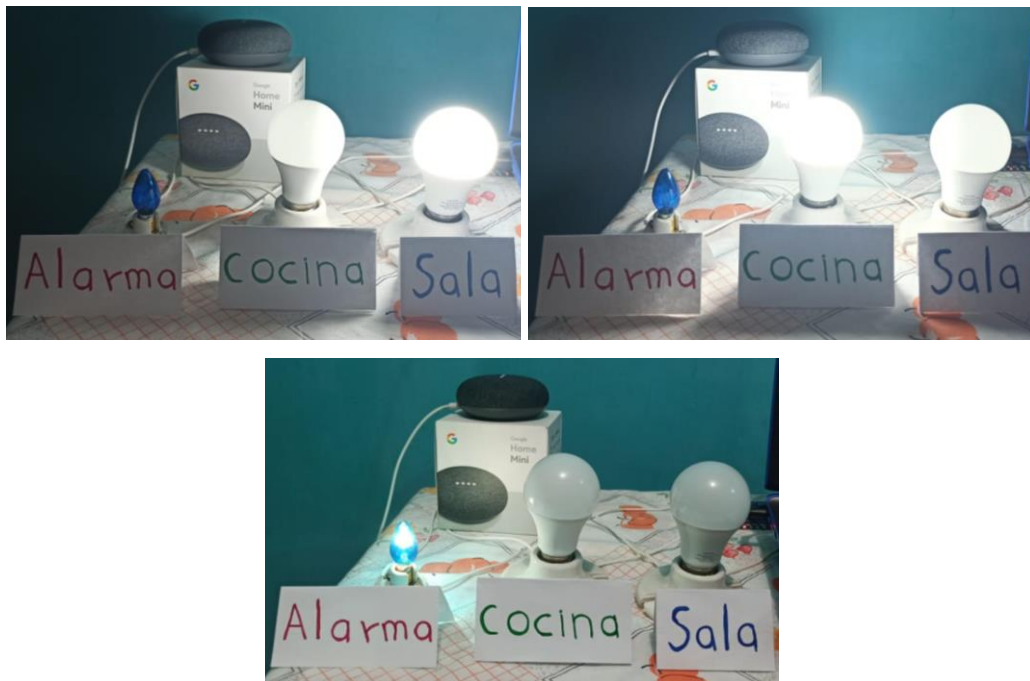


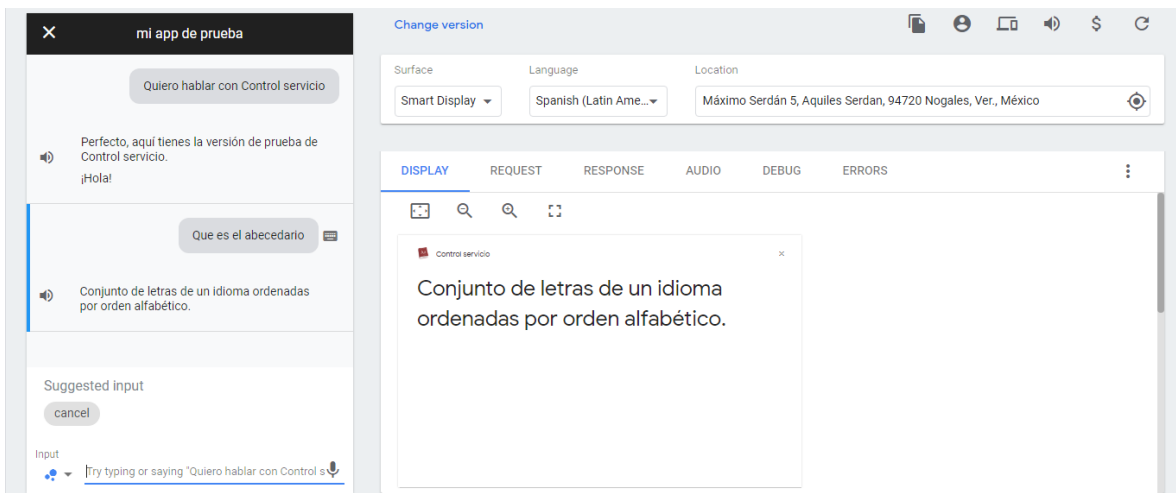
Figura 32 Prueba de funcionamiento de dispositivos.

#### 4.1.2 Caso de Estudio: Uso de servicio web mediante un nuevo comando de voz de Google Assistant para la búsqueda de términos básicos de un diccionario.

El uso de *Web Service*, o Servicio Web, es un método de comunicación entre dos aparatos electrónicos en una red. Es una colección de protocolos abiertos y estándares usados para intercambiar datos entre aplicaciones o sistemas.

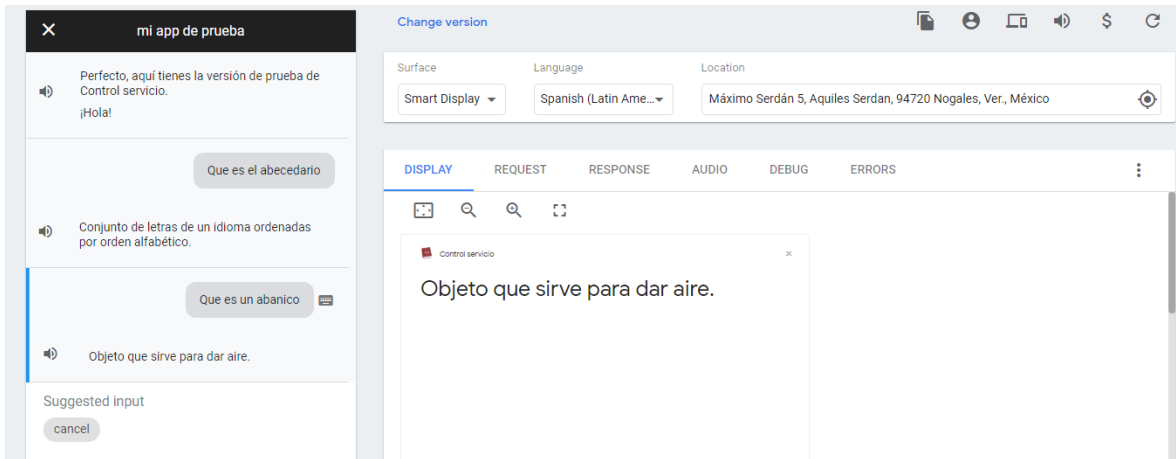
Las aplicaciones escritas en varios lenguajes de programación que funcionan en plataformas diferentes pueden utilizar dichos servicios para intercambiar información a través de una red, además, estos ofrecen información con un formato estándar que puede ser entendido fácilmente por una aplicación.

El asistente virtual de Google hará el uso de un servicio web mediante comando de voz para la búsqueda de términos básicos de un diccionario.



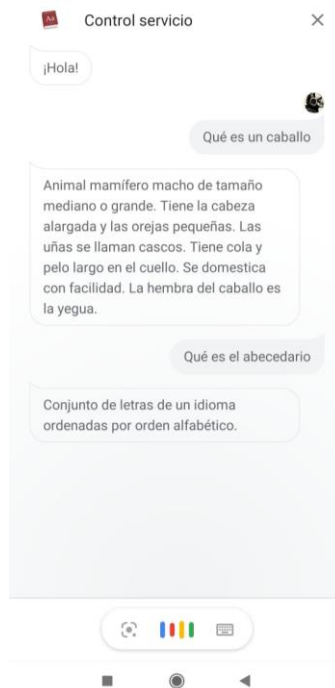
*Figura 33 Términos básicos del servicio web.*

El comando de voz permite al usuario seguir buscando términos de manera indefinida, para que no se tenga que dar la instrucción para activar el comando de voz una y otra vez.



*Figura 34 Búsqueda continua de significados.*

El comando de voz se puede utilizar en cualquier dispositivo que tenga instalado el asistente virtual de Google como se muestra en la figura 4.14.



*Figura 35 Búsqueda de significados mediante dispositivo móvil.*

Este comando de voz no solo se usaría para un sitio o servicio web en específico, ya que es posible que se use en más de un servicio, de manera que se ha abierto una puerta para que los desarrolladores implemente el asistente virtual de Google para la promoción y/o uso de sus servicios web y las funcionalidades que permitan.

## **Conclusiones**

El desarrollo e implementación de un conjunto de instrucciones para el asistente virtual Google Assistant, es un tópico que está creciendo a pasos agigantados actualmente y lo proveen muy pocas plataformas.

Actualmente se observa una escasa cantidad de información para el desarrollo e implementación de nuevas instrucciones

Si bien el desarrollo e implementación de un conjunto de instrucciones es un tema no muy abordado, ya que sólo se analizan algunas tecnologías de desarrollo e implementación para un conjunto de instrucciones.

Con este trabajo se beneficiarán los desarrolladores que aborden la necesidad de crear e implementar instrucciones, ya que podrán observar las ventajas y desventajas de cada tecnología que se puede utilizar.

Debido a que la estructura de la arquitectura, así como las tecnologías que usa Google Assistant y a su vez la forma en que se desarrolla los comandos de voz el asistente virtual de Google va evolucionando conforme pasa el tiempo y abre una gran posibilidad de crear nuevas funcionalidades que hasta el día de hoy no se puedan realizar.

## **Recomendaciones**

Google Assistant es un asistente virtual bastante útil. A pesar de ser muy conocido entre los usuarios, pocas personas saben realmente sacar provecho a todas sus funciones.

Este trabajo ayudará a estos desarrolladores poniendo en marcha el desarrollo e implementación de un conjunto de instrucciones que no están presentes actualmente para el asistente virtual Google Assistant. Como parte de las recomendaciones para el desarrollo de nuevos comandos de voz para Google Assistant, se encuentran:

- 1) Desarrollar comandos de voz que utilicen de manera exhaustiva el uso de servicios web: Mediante el asistente virtual de Google se pueda navegar y/o buscar los diferentes productos y servicios de una tienda digital sin la necesidad de buscar en el

navegador y que a su vez se puedan todas las funciones que ésta conlleva (compra, venta, búsqueda, y administración del sitio).

- 2) Enviar mensaje de emergencia al activar la alarma: Una vez enviada la señal de encendido de la alarma, el asistente será capaz de enviar un mensaje de alerta a las autoridades específicas para que actúen de acuerdo a la situación presentada.



## **Productos académicos**

En esta sección se brinda información acerca de los productos académicos que se realizaron con base en el proyecto de Tesis.

### **Artículos de congresos internacionales**



I.S.C Noé Hernández García, M.R.T. Ignacio López Martínez, M.C.E. Beatriz A. Olivares Zepahua, M.C. S. Gustavo S. Peláez Camarena, M.C. Ma. Antonieta Abud Figueroa.

*Propuesta de arquitectura para el desarrollo e implementación de un módulo de nuevos comandos e instrucciones para Google Assistant.*

## Referencias

- [1] S. Junestrand, X. Passaret, y D. Vázquez, *Domótica y hogar digital*. Editorial Paraninfo, 2004.
- [2] E. Méndez y J. A. Moreira González, “Lenguaje natural e indización automatizada”, *Ciencias de la Información*, 1999. [En línea]. Disponible en: <http://eprints.rclis.org/12685/>. [Consultado: 07-abr-2019].
- [3] M. B. Hernández y J. M. Gómez, “Aplicaciones de Procesamiento de Lenguaje Natural”, *Rev. Politécnica*, vol. 32, núm. 0, jul. 2013.
- [4] P. Lara Navarra y J. A. Martínez Usero, *Agentes inteligentes en la búsqueda y recuperación de información*. Planeta UOC, 2004.
- [5] Google, “Actions on Google Glossary | Actions on Google”, *Google Developers*. [En línea]. Disponible en: <https://developers.google.com/actions/glossary>. [Consultado: 15-mar-2019].
- [6] Google, “Google Assistant SDK”, *Google Developers*. [En línea]. Disponible en: <https://developers.google.com/assistant/sdk/overview>. [Consultado: 14-mar-2019].
- [7] Google, “Actions on Google”, *Google Developers*. [En línea]. Disponible en: <https://developers.google.com/actions/extending-the-assistant>. [Consultado: 15-mar-2019].
- [8] G. Iannizzotto, L. L. Bello, A. Nucita, y G. M. Grasso, “A Vision and Speech Enabled, Customizable, Virtual Assistant for Smart Environments”, en *2018 11th International Conference on Human System Interaction (HSI)*, 2018, pp. 50–56.
- [9] A. H. Michaely, X. Zhang, G. Simko, C. Parada, y P. Aleksic, “Keyword spotting for Google assistant using contextual speech recognition”, en *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 272–278.
- [10] G. López, L. Quesada, y L. A. Guerrero, “Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces”, en *Advances in Human Factors and Systems Interaction*, 2018, pp. 241–250.
- [11] A. S. Tulshan y S. N. Dhage, “Survey on Virtual Assistant: Google Assistant, Siri, Cortana, Alexa”, en *Advances in Signal Processing and Intelligent Recognition Systems*, 2019, pp. 190–201.

- [12] A. Bhalla, “An Exploratory Study Understanding the Appropriated Use of Voice-based Search and Assistants”, en *Proceedings of the 9th Indian Conference on Human Computer Interaction*, New York, NY, USA, 2018, pp. 90–94.
- [13] C. Peng y R. Chen, “Voice recognition by Google Home and Raspberry Pi for smart socket control”, en *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, 2018, pp. 324–329.
- [14] I. Hwang, J. Jung, J. Kim, Y. Shin, y J. Seol, “Architecture for Automatic Generation of User Interaction Guides with Intelligent Assistant”, en *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2017, pp. 352–355.
- [15] A. Cheng, V. Raghavaraju, J. Kanugo, Y. P. Handrianto, y Y. Shang, “Development and evaluation of a healthy coping voice interface application using the Google home for elderly patients with type 2 diabetes”, en *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2018, pp. 1–5.
- [16] A. Lazić, M. Z. Bjelica, D. Nad, y B. M. Todorović, “Google Assistant Integration in TV Application for Android OS”, en *2018 26th Telecommunications Forum (TELFOR)*, 2018, pp. 420–425.
- [17] S. Mischie, L. Mățiu-Iovan, y G. Gășpăresc, “Implementation of Google Assistant on Raspberry Pi”, en *2018 International Symposium on Electronics and Telecommunications (ISETC)*, 2018, pp. 1–4.
- [18] V. Kěpuska y G. Bohouta, “Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)”, en *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 99–103.
- [19] P. Dempsey, “The teardown: Google Home personal assistant”, *Eng. Technol.*, vol. 12, núm. 3, pp. 80–81, abr. 2017.
- [20] A. Reis *et al.*, “Using intelligent personal assistants to assist the elderlies An evaluation of Amazon Alexa, Google Assistant, Microsoft Cortana, and Apple Siri”, en *2018 2nd International Conference on Technology and Innovation in Sports, Health and Wellbeing (TISHW)*, 2018, pp. 1–5.

- [21] D. Calvo, L. Quesada, G. López, y L. A. Guerrero, “Multiplatform Career Guidance System Using IBM Watson, Google Home and Telegram”, en *Ubiquitous Computing and Ambient Intelligence*, 2017, pp. 689–700.
- [22] U. Višekruna y M. Savić, “Integration of Google Assistant in Android Application for Voice Control of Media Playback”, en *2018 26th Telecommunications Forum (TELFOR)*, 2018, pp. 1–4.
- [23] “Google Cloud”, *Google Cloud*. [En línea]. Disponible en: <https://cloud.google.com/docs/overview/developer-and-admin-tools>. [Consultado: 06-abr-2019].
- [24] “Dialogflow”. [En línea]. Disponible en: <https://dialogflow.com/docs/intro>. [Consultado: 06-abr-2019].
- [25] F. de Node.js, “NodeJS”, Node.js. [En línea]. Disponible en: <https://nodejs.org/es/about/>. [Consultado: 23-nov-2019].
- [26] Firebase, “Firebase”, Firebase. [En línea]. Disponible en: <https://firebase.google.com/docs>. [Consultado: 14-nov-2019].
- [27] R. S. Pressman, V. Campos Olguín, J. Enríquez Brito, B. J. Ferro Castro, y C. Villegas Quezada, *Ingeniería del software: un enfoque práctico*. México D. F.: McGraw-Hill, 2010.