

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**  
**SEDE VIÑA DEL MAR – JOSÉ MIGUEL CARRERA**

**SISTEMA DE CONTROL DOMÓTICO POR VOZ, BASADO EN PLACA DE DESARROLLO RASPBERRY**  
**PI**

Trabajo de Titulación para optar al Título  
Profesional de Ingeniero de Ejecución en  
CONTROL E INSTRUMENTACIÓN INDUSTRIAL

Alumno:

Javier Ignacio Fuentes Valenzuela

Profesor Guía:

Ing. Sergio Riquelme Bravo

**2018**

## RESUMEN

**KEYWORDS:** IOT, DOMÓTICA, PARLANTE INTELIGENTE, RASPBERRY PI, NODEMCU

Una encuesta realizada por Ericsson CONSUMERLAB en Chile, reveló que el 65% de los chilenos tiene interés en tener un hogar conectado (también llamado Smarthome). Esto es debido a diversos motivos, tales como tener mayor control de la seguridad de la casa, entretenimiento, control de temperatura, monitoreo de consumo eléctrico y otros. El auge que ha experimentado esta idea en el último tiempo, ha llevado a que empresas tales como Apple, Amazon o Google, hayan creado dispositivos llamados “parlantes inteligentes” que además de ser altavoces, cuentan con un micrófono para poder interactuar con ellos por medio de comandos de voz, son capaces de acceder a información tales como clima, tráfico, noticias y otros, además de poder comandar otros periféricos.

Surgen tres principales problemas frente a la implementación de un hogar conectado comandado por un parlante inteligente: escasa oferta, hermetismo de los sistemas y elevados precios.

Como una alternativa a estos equipos se plantea la posibilidad de implementar un sistema de control domótico por voz, el cual estará funcionando sobre la placa de desarrollo Raspberry Pi. Los comandos de voz serán interpretados por el asistente de Google (Google Assistant), el cuál ha sido liberado para la creación de diversas aplicaciones. A su vez, este asistente estará comunicado con una placa NodeMCU, la cual se encargará de la activación o desactivación de los periféricos a controlar. Es importante destacar que por el momento, el lenguaje de interacción con el dispositivo, es por medio del idioma inglés, dado que las herramientas para desarrolladores en español, aún se encuentra en proceso de desarrollo.

El proceso realizado es posible subdividirlo en tres etapas: configuración de Google Assistant en la Raspberry Pi, comunicación Google Assistant - NodeMCU y control de periféricos.

Es preciso resaltar que para extender las funcionalidades que vienen por defecto en el asistente, es necesario crear acciones adicionales, por medio del programa de desarrolladores “Actions on Google”, el cual a su vez posee las herramientas necesarias para la generación de respuestas habladas por parte del asistente.

El resultado esperado es la implementación del parlante inteligente junto con el hardware necesario para la activación y/o desactivación de periféricos.

## ÍNDICE

### RESUMEN

### SIGLAS Y SIMBOLOGÍAS

INTRODUCCIÓN .....	1
--------------------	---

### **CAPÍTULO 1: ANTECEDENTES, PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN..... 2**

<b>1. ANTECEDENTES GENERALES.....</b>	<b>3</b>
1.1. DOMÓTICA .....	3
1.2. INTERNET DE LAS COSAS.....	4
1.3. PLANTEAMIENTO DEL PROBLEMA .....	5
1.4. DEFINICIÓN DEL PROBLEMA .....	5
1.4.1. Importancia de resolverlo .....	5
1.4.2. Involucrados .....	6
1.5. PROPOSICIÓN DE SOLUCIÓN DEL PROBLEMA .....	7
1.6. REQUERIMIENTOS.....	9
1.7. ALTERNATIVAS DE SOLUCIÓN .....	11
1.7.1. Alternativa N°1: “Hacer nada” .....	11
1.7.2. Alternativa N°2: “Sistema de control domótico por voz basado en placa de desarrollo Raspberry Pi ” .....	11
1.7.3. Alternativa N°3: “Adquisición y configuración de sistemas preexistentes” .....	12
1.8. EVALUACIÓN DE LAS ALTERNATIVAS DE SOLUCIÓN.....	13
1.8.1. Alternativa seleccionada .....	13
1.9. OBJETIVOS DEL PROYECTO.....	14
1.9.1. Objetivo general .....	14
1.9.2. Objetivos específicos.....	14

### **CAPÍTULO 2: DESARROLLO TÉCNICO DE LA SOLUCIÓN ..... 15**

<b>2. DESARROLLO TÉCNICO DE LA SOLUCIÓN.....</b>	<b>16</b>
2.1. ETAPA 1: INSTALACIÓN DE SO EN RASPBERRY PI .....	17
2.2. ETAPA 2: CONFIGURACIÓN DE DISPOSITIVOS DE ENTRADA Y SALIDA DE AUDIO.....	20
2.2.1. Modificación del fichero config.txt .....	20
2.2.2. Reconocimiento de dispositivos de grabación y reproducción de sonido .....	22
2.2.3. Modificación del fichero asound.conf.....	24

2.2.4	Forzar las salidas de audio .....	25
2.3	ETAPA 3: INSTALACIÓN DE GOOGLE ASSISTANT EN LA RASPBERRY PI.....	27
2.4	ETAPA 4: CREACIÓN DE PROGRAMA DE CONTROL POR VOZ .....	34
2.4.1	Creación de Fulfillment .....	37
2.4.2	Instalación de Firebase en el sistema operativo .....	38
2.5	ETAPA 5: PROGRAMACIÓN DEL NODEMCU.....	39
<b>CAPÍTULO 3: EVALUACIÓN DE COSTOS .....</b>		<b>43</b>
<b>3.</b>	<b>EVALUACIÓN DE COSTOS DEL PROYECTO .....</b>	<b>44</b>
3.1	CARTA GANTT.....	44
3.2	COSTOS DE HARDWARE .....	45
3.3	COSTOS DE SOFTWARE .....	46
3.4	COSTOS DE PERSONAL .....	47
3.5	COMPARACIÓN CON EQUIPOS CONVENCIONALES .....	47
3.6	CONEXIÓN DE INTERNET .....	49
3.7	CONSUMOS DE ENERGÍA CONSIDERADOS.....	50
3.8	COMPARACIÓN ECONÓMICA.....	50
3.9	BENEFICIOS LOGRADOS.....	51
3.9.1	Internet de las cosas.....	51
3.9.2	Opciones de ingreso de información por voz o texto .....	52
3.9.3	Control a distancia.....	52
3.10	MEJORAS AL SISTEMA DESARROLLADO .....	53
3.10.1	Incorporación de nuevos idiomas en Google Assistant .....	53
3.10.2	Incorporación de nuevos idiomas en Dialogflow .....	53
3.10.3	Mejoras en entradas y salidas de audio.....	53
3.10.4	Reconocimiento de usuarios .....	53
3.10.5	Canales de salida .....	54
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>		<b>55</b>
<b>BIBLIOGRAFÍA .....</b>		<b>56</b>
<b>ANEXO A: CÓDIGO DE DIALOGFLOW WEBHOOK .....</b>		<b>57</b>
<b>ANEXO B: CÓDIGO DE NODEMCU .....</b>		<b>59</b>
<b>ANEXO C: DIAGRAMA GENERAL DE LA SOLUCIÓN.....</b>		<b>62</b>

## ÍNDICE DE FIGURAS

Figura 1-1	Diagrama de funcionamiento del controlador X10	3
Figura 1-2	Posibles dispositivos con internet de las cosas	5
Figura 1-3	Razones de interés de un servicio integrado	6
Figura 1-4	Interés general en servicios integrados	7
Figura 1-5	Amazon Echo primera generación	8
Figura 1-6	Programa para desarrolladores “Actions on Google”	9
Figura 1-7	Raspberry Pi	11
Figura 1-8	Home Monitoring kit de Samsung SmartThings	12
Figura 2-1	Diagrama general de desarrollo	16
Figura 2-2	Proceso de grabación con Etcher.io	18
Figura 2-3	Escritorio del SO.	19
Figura 2-4	Apariencia de la terminal	19
Figura 2-5	Archivo /boot/config.txt	21
Figura 2-6	Lineas de código finales de /boot/config.txt	22
Figura 2-7	Resultado de comando aplay -l	23
Figura 2-8	Resultado de comando arecord -l	24
Figura 2-9	Configuración de /etc/asound.conf	25
Figura 2-10	Menú de raspi-config	26
Figura 2-11	Opciones de audio	26
Figura 2-12	Forzado de salida	27
Figura 2-13	Acceso a Google Cloud	28
Figura 2-14	Creación de nuevo proyecto	28
Figura 2-15	Indicación de nuevo proyecto	29
Figura 2-16	Nombre del nuevo proyecto	29
Figura 2-17	Acceso a biblioteca de APIs	30
Figura 2-18	Búsqueda de credenciales	31
Figura 2-19	Nombre de nuevo cliente	31
Figura 2-20	Configuración de pantalla de autorización de OAuth	32
Figura 2-21	Código en formato .txt	32
Figura 2-22	Comando nano	33
Figura 2-23	Código en consola	33

Figura 2-24	Asistente funcionando	34
Figura 2-25	Menú de navegación	35
Figura 2-26	Frases de entrenamiento	36
Figura 2-27	Entities	36
Figura 2-28	Parámetros	37
Figura 2-29	URL del Webhook	39
Figura 2-30	Descarga de librerías	40
Figura 2-31	Incorporación de librerías	41
Figura 2-32	Actualización de placas	41
Figura 2-33	Placas basadas en ESP8266	42
Figura 3-1	Despliegue de información en Google Assistant	52

### ÍNDICE DE TABLAS

8Tabla 1-1	Requerimientos del proyecto	10
Tabla 1-2	Puntuación de alternativas	13
Tabla 1-3	Resultados obtenidos por cada alternativa	13
Tabla 2-1	Características de la Raspberry Pi	17
Tabla 2-2	Características del NodeMCU	39
Tabla 3-1	Diagrama Gantt del proyecto	44
Tabla 3-2	Costos de hardware del centro de control domótico	45
Tabla 3-3	Costos de software	46
Tabla 3-4	Costos de personal	47
Tabla 3-5	Precios de parlantes inteligentes	48
Tabla 3-6	Sistemas de control ofrecidos por el mercado	48
Tabla 3-7	Configuración comercial económica	49
Tabla 3-8	Conexión a internet	49
Tabla 3-9	Consumos de energía	50
Tabla 3-10	Comparación Económica	51

## SIGLAS Y SIMBOLOGÍAS

### a) Siglas

IOT	: Internet Of Things (Internet de las cosas)
SCE	: Sistema de cableado estructurado
GUI	: Graphical User Interface (Interfaz gráfica de usuario)
VUI	: Voice User Interface (Interfaz mediante voz del usuario)
IFTTT	: If This Then This (Si esto, entonces esto)
SO	: Sistema Operativo
USB	: Universal Serial Bus (Bus universal en serie)
HDMI	: High-Definition Multimedia Interface (Interfaz multimedia de alta definición)
SSH	: Secure Shell (Consola de seguridad)
API	: Application Programming Interface (Interfaz de programación de aplicaciones)
UI	: User Interface (Interfaz de usuario)
IA	: Inteligencia artificial
IDE	: Integrated Development Environment (Entorno de desarrollo integrado)
URL	: Uniform Resource Locator (Localizador uniforme de recursos)

### b) Simbologías

GHz	: Gigahertz
mm	: Milímetros
GB	: Gigabytes
V	: Volt
W	: Watt
kW	: Kilowatt

## INTRODUCCIÓN

El internet de las cosas se refiere a la interconexión digital de objetos cotidianos mediante internet. Esta conexión se da tanto a nivel industrial como nivel de consumo hogareño, esto debido a la facilidad para monitorear permanentemente dispositivos en una misma red. Es en el ambiente hogareño, donde se han masificado los parlantes inteligentes (dispositivos comandados mediante instrucciones de voz), como una forma de mantener el control de los (cada vez más) dispositivos conectados a la red del hogar.

El presente proyecto, busca mostrar una alternativa económica y abierta para la creación de un parlante inteligente capaz de poder entregar información actualizada de tráfico, temperatura, cálculos matemáticos, gramática, condiciones climáticas o funcionamiento de dispositivos junto con la capacidad de activar y/o desactivarlos en el momento que se considere oportuno.

El proyecto, persigue como finalidad la creación de un sistema de control domótico de bajo costo y alta personalización, haciendo uso de herramientas ya existentes para estos propósitos.

Otro de los propósitos de este proyecto consiste en la familiarización con tecnologías emergentes tales como Firebase (utilizado para la creación de bases de datos), Dialogflow (creación de interfaces conversacionales), programación de NodeMCU (microcontrolador con la capacidad de conexión a internet), programación de Raspberry Pi (computador de bajo costo, altamente personalizable), así como también perfeccionar habilidades de programación.

Por otro lado es importante mencionar que las tecnologías mencionadas se encuentran en permanente cambio, debido a lo que los códigos expuestos en el informe pueden ser no compatibles con las versiones más nuevas de software u otros.

**CAPÍTULO 1: ANTECEDENTES, PLANTEAMIENTO DEL PROBLEMA Y SOLUCIÓN**

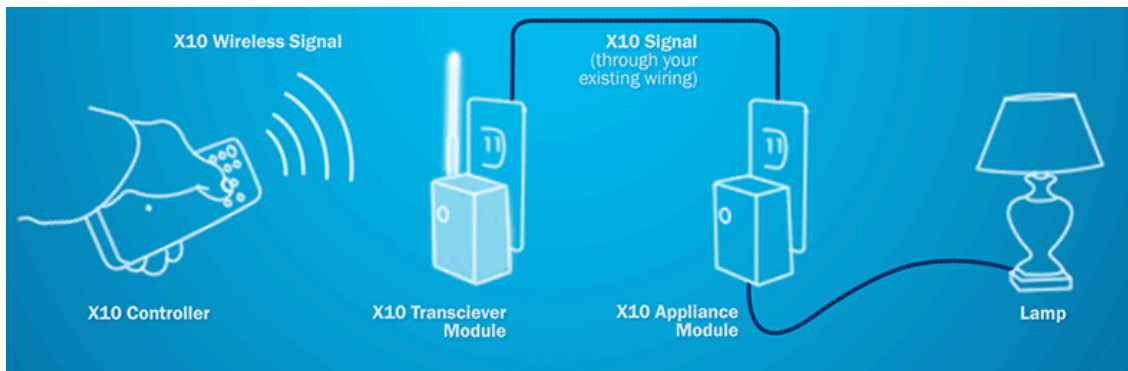
## 1. ANTECEDENTES GENERALES

En esta sección se explicará el contexto general en el cual está inmerso el proyecto.

### 1.1. DOMÓTICA

La domótica es “el conjunto de sistemas que automatizan las diferentes instalaciones de una vivienda”. Si bien es cierto el término ha experimentado un auge durante el último tiempo, los inicios de la domótica se remontan a la década del 60 en Estados Unidos, y solamente consideraba la opción de controlar la temperatura, aunque para ese entonces, tampoco era una práctica muy frecuente.

Más tarde, en Escocia se desarrolló un protocolo para control de dispositivos eléctricos que utiliza la red eléctrica, llamado protocolo X10. Una de las limitaciones de estos dispositivos es el área en el cuál es posible controlar los dispositivos, esto debido al ancho de banda de las señales. La figura 1-1 muestra un esquema básico de comunicación por medio de ese protocolo.



Fuente: <https://www.x10.com/x10-basics.html>

Figura 1-1. Diagrama de funcionamiento del controlador X10

Fue con el desarrollo de los computadores y su masificación como herramienta de trabajo, cuando la domótica experimentó un cambio sustancial, esto debido a que fue necesaria la instalación de Sistemas de cableado estructurados (SCE) para comunicarlos, los cuales permitieron el transporte interno de datos y voz.

Actualmente con la masificación de las redes inalámbricas y evolución de protocolos de comunicaciones, los sistemas cuentan con más herramientas para poder comunicarse entre sí. Uno de los conceptos que se emplea en la actualidad para referirse a la interconexión de dispositivos es el denominado “internet de las cosas”.

## 1.2. INTERNET DE LAS COSAS

Se conoce como internet de las cosas a la interconexión de objetos cotidianos a internet, es decir, dispositivos que no cuentan en la actualidad con la capacidad de conexión a internet en forma inherente como por ejemplo: máquinas expendedoras, vehículos, electrodomésticos y otros. Este concepto no es algo propio del campo de la domótica, ya que incluso empresas como WINDOWS, CISCO, IBM y en general los grandes conglomerados tecnológicos se encuentran desarrollando distintos proyectos relacionados con esa área.

Entre las ventajas que ofrece el internet de las cosas se encuentran:

- Automatización de procesos: Dado que los distintos elementos de un sistema podrá ser controlado por el usuario, y tendrán comunicación entre ellos.
- Análisis de datos más optimizados: El mantener la información a tiempo real de un dispositivo permite realizar un análisis mucho más rápido y preciso que careciendo de esta tecnología.
- Optimización de recursos: El tener conocimientos sobre la ubicación, fechas de caducidad u otras características de distintos elementos, permite generar menos desperdicios y optimizar la generación de recursos en función de lo que sea estrictamente necesario.

La figura 1-2 muestra un ejemplo de posibles dispositivos con internet de las cosas.



Fuente: <https://www.androidcentral.com>

Figura 1-2. Posibles dispositivos con internet de las cosas

### 1.3 PLANTEAMIENTO DEL PROBLEMA

La empresa consultora y de investigación Gartner, en un estudio realizado el año 2017 estimó que para el año 2020, existirán 20 mil millones de dispositivos conectados al “internet de las cosas”, por lo tanto la presencia de este tipo de dispositivos se hará cada vez más cercana a la realidad hogareña, debido a esto, surge la necesidad de controlar estos equipos de una manera económica, que permita cierta flexibilidad en cuanto al número de dispositivos, y que no sólo pueda realizarse por medio de una interfaz gráfica.

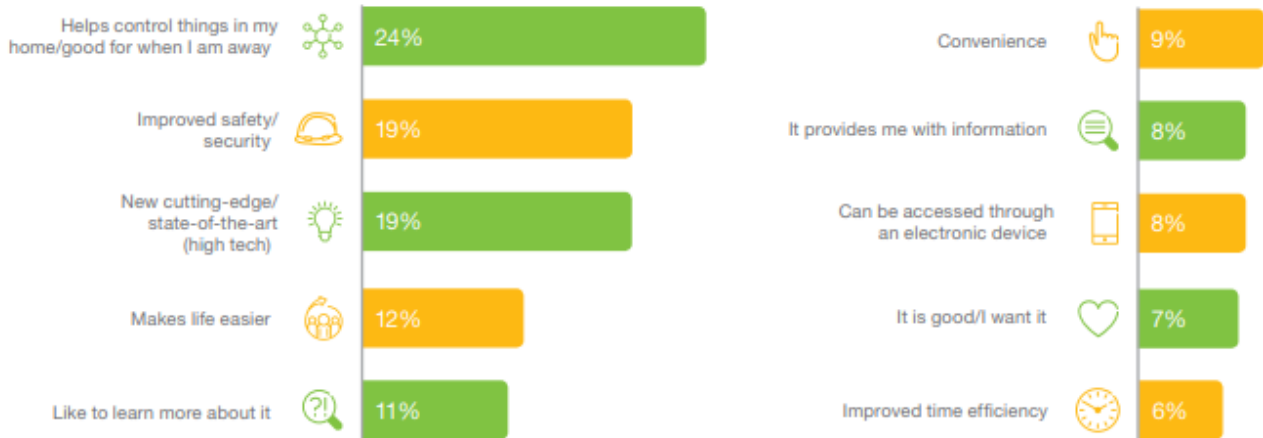
### 1.4 DEFINICIÓN DEL PROBLEMA

Necesidad de realizar control domótico mediante interfaz de voz y en forma económica.

#### 1.4.1 Importancia de resolverlo

En primera instancia, resolver este problema permitirá al usuario tener mayor control sobre cada uno de los dispositivos que tiene conectado. Por otra parte, también existen otras

motivaciones de los usuarios para poder tener un centro de control domótico, tal cual muestra la figura 1-3.



Fuente: Ericsson ConsumerLab, Connected Homes, 2015

Figura 1-3. Razones de interés de un servicio integrado

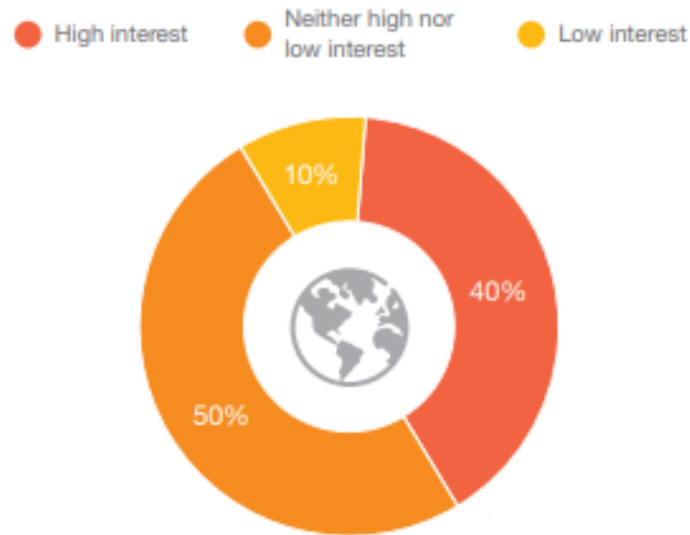
Las 10 razones mencionadas por los usuarios son las siguientes:

- 1) Ayudar a controlar la casa cuando uno está lejos.
- 2) Mejorar la seguridad.
- 3) Estar en la vanguardia tecnológica.
- 4) Facilitar la vida.
- 5) Aprender cosas nuevas.
- 6) Conveniencia.
- 7) Provee con información.
- 8) Puede ser accesible mediante dispositivos electrónicos.
- 9) Es bueno, me gusta.
- 10) Ayuda a optimizar el tiempo.

#### 1.4.2 Involucrados

Una encuesta realizada por la Ericsson CONSUMERLAB (división de investigación de Ericsson), realizada en junio del año 2008 a un total de 1000 personas entre los 18 y los 69 años de edad, indicó que en Chile el 50% de los encuestados tiene un interés por los servicios integrados, esto quiere decir, un sistema “todo en uno” para el hogar, que consiste en una serie

de dispositivos y servicios conectados tales como monitoreo de energía doméstica, calefacción, iluminación y otros, cuyo propósito es la simplificación de labores hogareñas. El resumen de estas estadísticas se muestra en la figura 1-4.



Fuente: Ericsson ConsumerLab, Connected Homes, 2015

Figura 1-4. Interés general en servicios integrados en hogares de Chile

## 1.5 PROPOSICIÓN DE SOLUCIÓN DEL PROBLEMA

Uno de los dispositivos que se acercan a la capacidad de poder controlar otros dispositivos son los denominados “parlantes inteligentes”.

El 23 de Junio del año 2015, Amazon puso a disposición del mercado un pequeño parlante inalámbrico, controlado por voz de nombre “Amazon Echo”. Este dispositivo cuenta con un asistente personal inteligente llamado “Alexa”, el cual es capaz de interactuar con el usuario por medio de comandos de voz. La figura 1-5 muestra el aspecto del parlante.



Fuente: <https://www.amazon.com>

Figura 1-5. Amazon Echo primera generación

Algunas de las funciones que ofreció este parlante inteligente fueron:

- Escuchar música: Dado que cuentan con sincronización con aplicaciones como Spotify, Deezer o Youtube.
- Información meteorológica: Dependiendo del parlante inteligente, es el servicio al cuál se conecta para obtener esta información.
- Preguntas generales: Debido a que estos parlantes cuentan con una conexión permanente a internet, es posible hacer consultas tales como fechas, cálculos matemáticos, gramática y otros.
- Alarmas y cronómetros: Es posible indicar al parlante que recuerde una determinada cita, o funcione como alarma.

Después del éxito inicial de este producto y su asistente, otras empresas empezaron a desarrollar sus propios parlantes y asistentes, entre las empresas que se encuentran desarrollando este tipo de productos se encuentran: Amazon, Google, Alibaba, Microsoft, Samsung y otros, los que han ido ofreciendo cada vez mayor cantidad de servicios.

La consultora CANALYS estimó que para el año 2018, los parlantes inteligentes tendrían un crecimiento hasta alcanzar los 56,3 millones de unidades vendidas, mucho más que dispositivos utilizados para la realidad aumentada o la realidad virtual, los que también se encuentran en su apogeo comercial.

Google por su parte, además de producir los equipos Google Home Max y Google Home Mini, han liberado para los desarrolladores diferentes herramientas para poder desarrollar proyectos independientes basados en interfaces de voz de usuario (también llamadas VUI). Lo anterior sin incluir otras herramientas como “Actions on Google”, las que permitirían

eventualmente diseñar funciones basadas en las necesidades de los desarrolladores. En la figura 1-6 se muestra el ícono de esta suite de Google.



Fuente: <https://developers.google.com/actions/>

Figura 1-6. Programa para desarrolladores “Actions on Google”

Actualmente en Chile, el mercado de los parlantes inteligentes se encuentra todavía en fase inicial, con precios que oscilan entre \$39.900 y los \$329.990, esto es sin considerar los costos involucrados en la adquisición de periféricos que se pueden llegar a controlar con este tipo de dispositivos. La oferta existente aún se limita a marcas como Apple o Google, los cuales a su vez tienen sistemas cerrados que no permiten mayor interacción con dispositivos que no estén reconocidos previamente por estos parlantes.

Considerando los factores anteriores, es que se buscaría un dispositivo que permita comunicación mediante voz, pero que a la vez, posea un costo conveniente y facilidad de comunicación con otros periféricos.

## 1.6 REQUERIMIENTOS

En la tabla 1-1 se indican las principales características que deben tener el proyecto junto con una pequeña descripción y los criterios de aceptación de cada uno de los requerimientos.

Tabla 1-1. Requerimientos del proyecto.

N°	Requerimiento	Descripción	Criterio de aceptación
1	Control por medio de voz.	El centro de control domótico, debe permitir la interacción no solamente por medio de GUI, sino que también por medio de VUI.	Reconocimiento de lenguaje, ya sea inglés o español.
2	Permitir el control de periféricos.	El sistema de control domótico debe permitir el manejo de cierto número de periféricos iniciales considerando encendido y apagado de estos.	4 periféricos mínimo.
3	Comunicación con periféricos de forma inalámbrica.	La comunicación del centro de control domótico con los otros dispositivos debe realizarse de forma inalámbrica.	Uso de WIFI para las comunicaciones.
4	Respuestas del equipo por medio de voz.	El equipo debe tener tanto la capacidad de recibir instrucciones de voz, así como entregar información de esta forma.	Preferentemente del mismo lenguaje que el control.
5	Comandar dispositivos a distancia.	En caso que el dispositivo no pueda ser comandado por voz, debe tener la facultad de recibir instrucciones en forma alternativa.	Via WIFI, mediante asistente virtual, IFTTT, u otra tecnología que permita comunicación.

Fuente: Elaboración propia

## 1.7 ALTERNATIVAS DE SOLUCIÓN

Para satisfacer los requerimientos necesarios, se realizan tres propuestas de solución.

### 1.7.1 Alternativa N°1: “Hacer nada”

El “Hacer nada” implica no realizar cambios en la forma en la cual se controlan los distintos dispositivos, por lo que no existe mayor control en la vivienda, ni capacidad de monitoreo.

### 1.7.2 Alternativa N°2: “Sistema de control domótico por voz basado en placa de desarrollo Raspberry Pi ”

La Raspberry Pi es un computador de placa reducida surgido en agosto del 2011. Tiene la particularidad de soportar sistemas operativos derivados de Linux, además de algunas versiones de Windows. Cuenta con pines generales de entrada/salida, además de capacidad de conexión a internet vía Ethernet o WIFI, lo que permite a la placa conectarse a distintos periféricos. Tiene una comunidad de desarrolladores a lo largo de todo el mundo, ya que es ampliamente utilizada para todo tipo de proyectos. El aspecto de la Raspberry Pi es el que se muestra en la figura 1-7.



Fuente: <https://www.islabit.com>

Figura 1-7. Raspberry Pi modelo 3B

Otra de las particularidades de la Raspberry Pi son los distintos complementos con los que se puede dotar de otras capacidades a la placa. Google ha liberado al público ciertas herramientas de voz y visión que pueden ser adquiridas.

### 1.7.3 Alternativa N°3: “Adquisición y configuración de sistemas preexistentes”

Esta alternativa se refiere a adquirir sistemas de control domótico preexistentes, tales como los que son provistos por Ozom, Smarththings de Samsung u otros. Esta alternativa, considera adquisición por un lado del centro de control por voz, y por otro lado de los periféricos a actuar. La figura 1-8 muestra los principales sensores que son incluidos en el Home Monitoring Kit de Samsung.



Fuente: <https://www.samsung.com/us/smart-home/smarthings/>

Figura 1-8. Home Monitoring kit de Samsung SmartThings

## 1.8 EVALUACIÓN DE LAS ALTERNATIVAS DE SOLUCIÓN

Para evaluar las alternativas de solución se utilizarán dos tablas: en primer lugar, la tabla 1-2 que establecerá la puntuación correspondiente a cada situación, y en segundo lugar la tabla 1-3, que contiene las formulaciones a ser medidas.

Tabla 1-2. Puntuación de alternativas.

Muy deficiente	Deficiente	Aceptable	Bueno	Óptimo
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>

Fuente: Elaboración propia.

Tabla 1-3. Tabla comparativa.

Comparación	Alternativa 1	Alternativa 2	Alternativa 3
<b>Innovación</b>	1	4	2
<b>Beneficio</b>	1	4	3
<b>Factibilidad</b>	5	5	4
<b>Puntuación total</b>	7	13	9

Fuente: Elaboración propia basada en comparación de alternativas.

### 1.8.1 Alternativa seleccionada

Tal cual se aprecia en la tabla 1-3, si bien es cierto todas las alternativas son factibles, la número 3 se ve limitada debido a los altos costos de implementación del sistema, por otro lado, la alternativa número dos, presenta mayores facilidades de personalización de sus características.

## 1.9 OBJETIVOS DEL PROYECTO

Dentro de los objetivos a alcanzar por el proyecto se encuentran los siguientes:

### 1.9.1 Objetivo general

**Crear e implementar un sistema de control domótico, comandado por voz.**

### 1.9.2 Objetivos específicos

1. Determinar los equipos necesarios para la implementación del centro de control domótico, así como programas necesarios para su configuración.
2. Configurar la placa Raspberry Pi, de tal forma que pueda ejecutar el asistente de Google.
3. Programar la placa NodeMCU para el control de los periféricos.
4. Comunicar vía internet las dos placas (Raspberry Pi y NodeMCU).
5. Evaluar costos de implementación del sistema.
6. Realizar comparación económica entre sistemas existentes y el centro de control domótico basado en Raspberry Pi.

## **CAPÍTULO 2: DESARROLLO TÉCNICO DE LA SOLUCIÓN**

## 2. DESARROLLO TÉCNICO DE LA SOLUCIÓN

En el presente capítulo se detallará el procedimiento para la implementación de la solución propuesta en el primer capítulo.

La implementación del “Parlante inteligente” es posible desarrollarla en 5 etapas:

Etapa 1: Instalación de S.O en Raspberry Pi.

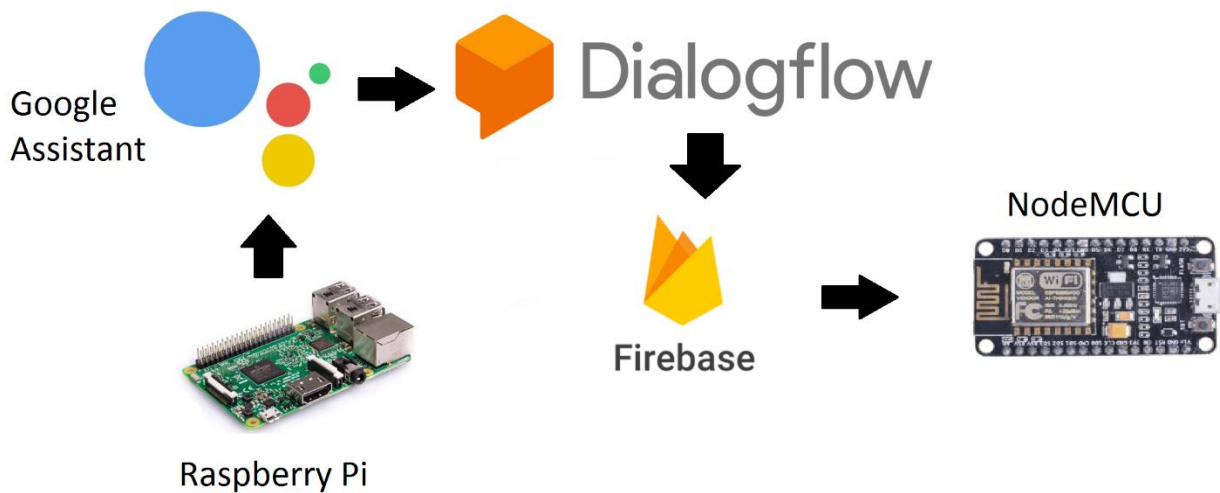
Etapa 2: Configuración de dispositivos de entrada y salida de audio.

Etapa 3: Instalación de Google Assistant en la Raspberry Pi

Etapa 4: Generación de programa de control por voz.

Etapa 5: Programación de placa NodeMCU.

El diagrama presentado en la figura 2-1, muestra el desarrollo a seguir. En primer lugar la Raspberry Pi se comunicará con la aplicación de Google Assistant cargada, la cual estará en contacto con Dialogflow (programa para crear interfaces conversacionales), que permitirá guardar los parámetros deseados en una base de datos de Firebase, que indicará en tiempo real las condiciones que se desean en las salidas (ya sea, encendido o apagado).



Fuente: Elaboración propia

Figura 2-1. Diagrama general de desarrollo

## 2.1 ETAPA 1: INSTALACIÓN DE SO EN RASPBERRY PI

La Raspberry Pi, tal como fue mencionado en el capítulo anterior, es una pequeña placa de desarrollo fabricada en Gran Bretaña, cuyo propósito es el facilitar la enseñanza de la programación en establecimientos educacionales. Las principales características de la placa están expuestas en la tabla 2-1.

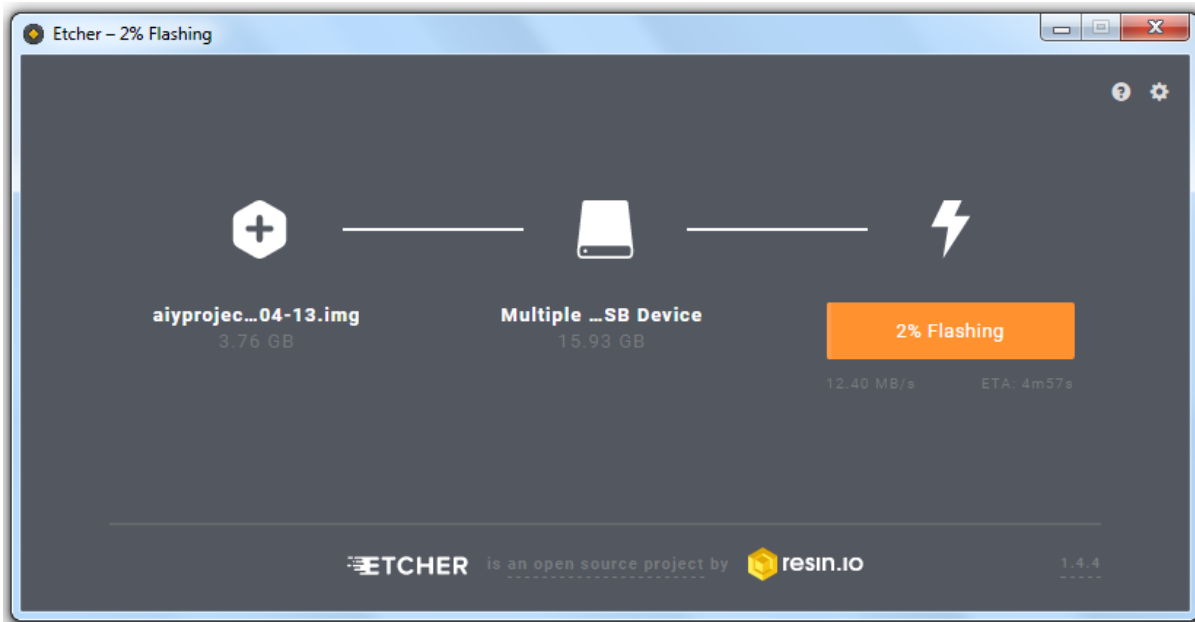
Tabla 2-1. Características de la Raspberry Pi

Características Generales	
Modelo	Raspberry Pi 3B
SoC	Broadcom BC2837
CPU	1.2 GHz 64-bit quad-core ARMv8
Puertos USB	4
Salida de Audio	HDMI, jack 3.5 [mm]
Salida de Video	HDMI
Almacenamiento integrado	MicroSD
Sistema operativo soportado	Raspbian, Arch Linux, otros

Fuente: Elaboración propia

Para propósitos de este proyecto, el sistema operativo que será usado será una variante de Raspbian, la cual incluye herramientas para la implementación del asistente de Google. La descarga del sistema operativo puede realizarse en formato .zip, desde la siguiente página: <https://github.com/google/aiyprojects-raspbian>

Para la grabación de la imagen en el dispositivo, la opción seleccionada es Etcher.io, el cual es un programa libre que permite grabar imágenes .zip o ISO en dispositivos extraíbles tales como unidades USB o microSD. En este caso, se utilizará una tarjeta microSD de una capacidad de 16 [GB]. La imagen 2-2, muestra la apariencia del programa en ejecución.



*Fuente: Elaboración propia*

Figura 2-2- Proceso de grabación con Etcher.io

Es importante destacar que las imágenes existentes de los sistemas operativos se encuentran en permanente actualización. Para el centro de control domótico, la imagen usada corresponde a la fecha 13 de Abril del año 2018, esto debido a la estabilidad que ofrece esta versión respecto a las más recientes.

Para la configuración inicial del centro de control domótico, es necesario realizar ciertas conexiones básicas de la Raspberry Pi, tales como:

- Pantalla por medio de HDMI.
- Teclado por medio de puerto USB.
- Mouse por medio de puerto USB.
- Micrófono USB.
- Parlante con Jack de 3.5 [mm] conectada a la placa.
- Fuente de alimentación a la placa Raspberry Pi.

Al insertar la tarjeta MicroSD, energizar la Raspberry Pi y encender la pantalla, es posible observar el escritorio del nuevo SO tal como se observa en la figura 2-3. En caso contrario es recomendable realizar nuevamente el proceso de grabado de la imagen del SO.

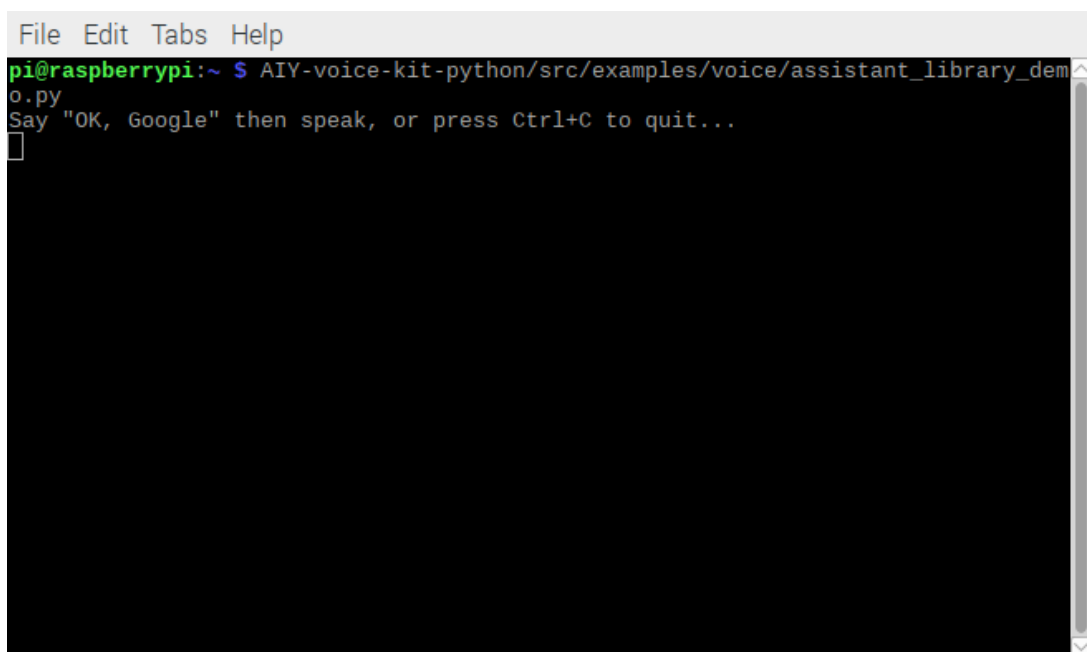


Fuente: Elaboración propia

Figura 2-3. Escritorio del SO

Otro aspecto importante a considerar es que los sistemas operativos de Linux permiten su control por medio de la consola (también llamada terminal de comandos).

La terminal o consola de Linux, es un intérprete de comandos u órdenes, que sirven de interfaz entre el usuario y el sistema operativo. En la figura 2-4 se observa la apariencia de una consola en Linux.



Fuente: Elaboración propia

Figura 2-4. Apariencia de la terminal

El aprender a utilizar el terminal para configurar la Raspberry Pi, es con el propósito de poder realizar modificaciones a distancia por medio de SSH (Secure Shell). SSH es un protocolo de administración remota que permite el modificar equipos remotos a través de internet, solamente sabiendo su dirección IP. Es posible usar este protocolo por medio de complementos de navegadores tales como Secure Shell Extension. De esta manera, la Raspberry Pi es configurable sin necesidad de tener una pantalla conectada todo el tiempo.

## **2.2 ETAPA 2: CONFIGURACIÓN DE DISPOSITIVOS DE ENTRADA Y SALIDA DE AUDIO**

En el mercado, existen equipos desarrollados por Google para la implementación de prototipos funcionales, pero tienen la desventaja de ser escasos en el mercado. La primera dificultad que se encuentra en el desarrollo del proyecto, es que las imágenes del sistema operativo, funcionan en base a estos equipos, por lo que es necesario realizar ciertas modificaciones en el código para que considere otros periféricos para comunicarse.

Las modificaciones a realizar son cuatro:

- Modificación del fichero config.txt.
- Reconocimiento de dispositivos de grabación y reproducción de sonido.
- Modificación del fichero asound.conf.
- Forzar las salidas de audio.

### **2.2.1 Modificación del fichero config.txt**

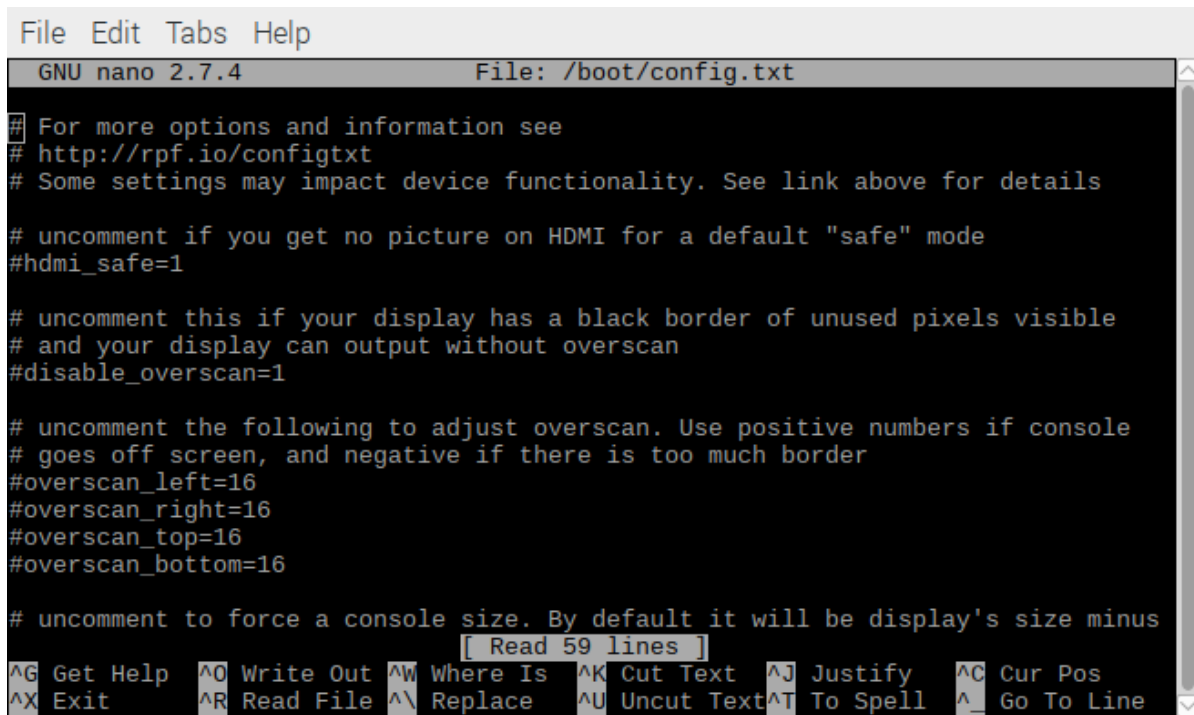
En los S.O derivados de Linux, no se habla de carpeta o archivo, sino de fichero. La ventaja que presenta un sistema de fichero por sobre los sistemas Windows, es la facultad de poder realizar modificaciones, que logran personalizar los equipos, además de tener un mayor control debido a las restricciones para administradores (también llamados superusuarios).

En el caso en particular de config.txt, este es un fichero encargado de actuar como BIOS de la Raspberry Pi (dado que el sistema intenta optimizar la mayor cantidad de memoria posible). Este fichero se encuentra a su vez en otro fichero llamado boot, en el cuál se encuentran todo lo necesario para el arranque del dispositivo.

Para realizar la modificación de config.txt, es necesario acceder por medio del comando nano y con privilegios de administrador (superusuario). Nano es un editor de textos para la terminal, para utilizarlo es necesario escribir en la terminal el siguiente comando:

```
sudo nano /boot/config.txt
```

Al ejecutar el comando, es posible observar en la terminal una imagen igual a la presentada en la figura 2-5.



```
File Edit Tabs Help
GNU nano 2.7.4 File: /boot/config.txt
# For more options and information see
# http://rpf.io/configtxt
# Some settings may impact device functionality. See link above for details
# uncomment if you get no picture on HDMI for a default "safe" mode
#hdmi_safe=1
# uncomment this if your display has a black border of unused pixels visible
# and your display can output without overscan
#disable_overscan=1
# uncomment the following to adjust overscan. Use positive numbers if console
# goes off screen, and negative if there is too much border
#overscan_left=16
#overscan_right=16
#overscan_top=16
#overscan_bottom=16
# uncomment to force a console size. By default it will be display's size minus
[ Read 59 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Fuente: Elaboración Propia

Figura 2-5. Archivo /boot/config.txt

En la imagen, aparecen ciertas líneas de código que poseen un “#” al inicio, ese símbolo indica al compilador que esa línea no debe ser considerada para propósitos de compilación (es decir, están comentadas).

Al bajar a las últimas líneas de código es posible observar una línea comentada que dice “dtparam=audio=on”, es necesario “descomentar” ésta, y comentar las 3 líneas de código siguientes. Esto es para permitir al dispositivo utilizar el módulo de sonido por defecto de la Raspberry Pi (snd\_bcm2835). En la figura 2-6 se muestran las líneas de código resultantes.

```

File Edit Tabs Help
GNU nano 2.7.4 File: /boot/config.txt
#dtoverlay=lirc-rpi
# Additional overlays and parameters are documented /boot/overlays/README
# Enable audio (loads snd_bcm2835)
dtparam=audio=on
# dtoverlay=dwc2
#start_x=1
#gpu_mem=128

```

Fuente: Elaboración propia

Figura 2-6. Líneas de código finales de /boot/config.txt

Al terminar de modificar el archivo, se guarda la nueva información con la combinación Control+O, y se procede al reconocimiento de los dispositivos de grabación y reproducción de sonidos.

### 2.2.2 Reconocimiento de dispositivos de grabación y reproducción de sonido

Para poder indicarle a la Raspberry Pi cuales dispositivos debe considerar por defecto para salida y entrada de audio, es necesario reconocer ciertos parámetros tanto del micrófono, como de los parlantes conectados. Para realizar esto se utilizan dos comandos: aplay y arecord.

El comando aplay permite reconocer los dispositivos conectados a la placa que son reproductores de sonido. En el caso particular del comando introducido, se agrega, después del nombre del comando un guión con una letra L minúscula, esto para indicarle al comando que la información presentada sea en forma de una lista (el propósito de esto es netamente orden). El resultado se muestra de la forma presentada en la figura 2-7.

```

File Edit Tabs Help
pi@raspberrypi:~$ aplay -l
**** List of PLAYBACK Hardware Devices ****
card 0: ALSA [bcm2835 ALSA], device 0: bcm2835 ALSA [bcm2835 ALSA]
  Subdevices: 8/8
    Subdevice #0: subdevice #0
    Subdevice #1: subdevice #1
    Subdevice #2: subdevice #2
    Subdevice #3: subdevice #3
    Subdevice #4: subdevice #4
    Subdevice #5: subdevice #5
    Subdevice #6: subdevice #6
    Subdevice #7: subdevice #7
card 0: ALSA [bcm2835 ALSA], device 1: bcm2835 ALSA [bcm2835 IEC958/HDMI]
  Subdevices: 1/1
    Subdevice #0: subdevice #0
pi@raspberrypi:~$

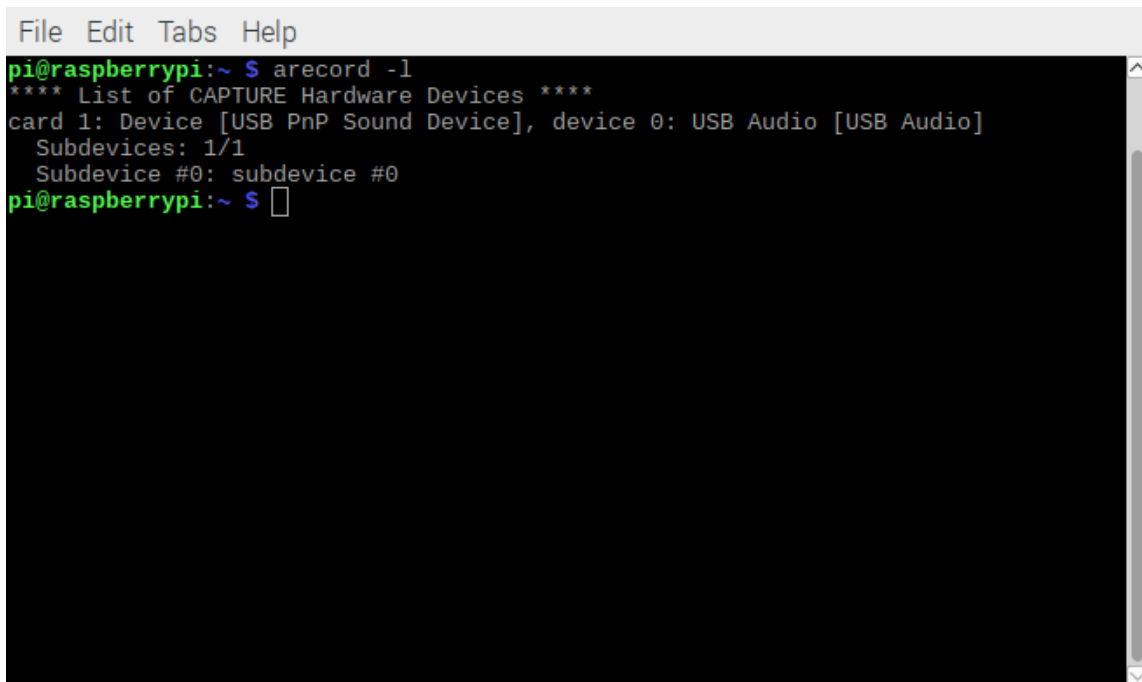
```

Fuente: Elaboración propia

Figura 2-7. Resultado de comando aplay -l

Es importante diferenciar entre los dos dispositivos mostrados en la lista. El dispositivo que se necesita es el bcm2835 ALSA, dado que la segunda opción tiene relación con salida por medio de HDMI, lo cual implicaría que el sistema debe tener necesariamente conectada una pantalla para poder escuchar la respuesta del parlante (ya que el cable HDMI transmite tanto sonido como imagen). La información que se necesita en esta etapa es el número de tarjeta (card) y número de dispositivo (device).

Para obtener el número de tarjeta y dispositivo del micrófono, el proceso es similar a los dispositivos de reproducción, la diferencia radica en el comando utilizado (arecord para este caso). El resultado esperado se ve reflejado en la figura 2-8.



```

File Edit Tabs Help
pi@raspberrypi:~ $ arecord -l
**** List of CAPTURE Hardware Devices ****
card 1: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
  Subdevices: 1/1
  Subdevice #0: subdevice #0
pi@raspberrypi:~ $ █

```

*Fuente: Elaboración propia*

Figura 2-8. Resultado de comando arecord -l

Al igual que en el caso anterior, la información necesaria es el número de tarjeta y dispositivo, aunque a diferencia del caso anterior, solamente existe un dispositivo de grabación.

### 2.2.3 Modificación del fichero asound.conf

El fichero asound.conf, es el encargado de especificar las salidas y entradas de audio que la Raspberry Pi considerará cada vez que inicie sesión. Está incorporado dentro del fichero etc. Este tiene por finalidad el almacenar scripts de arranques, o de configuración general.

El procedimiento a seguir es similar al anterior, se necesita acceder al fichero por medio de privilegios de administrador. Para realizar eso se utiliza nuevamente el comando sudo, con diferencia de la nueva dirección.

```
sudo nano /etc/asound.conf.
```

Es necesario borrar la información pre-existente en la dirección, y reemplazarla por las líneas de código expuestas en la figura 2-9.

```
File Edit Tabs Help
GNU nano 2.7.4 File: /etc/asound.conf

pcm.!default{
  pcm.!default{
    type asym
    capture.pcm "mic"
    playback.pcm "speaker"
  }
}

pcm.mic{
  type plug
  slave{
    pcm "hw: 1,0"
  }
}

pcm.speaker{
  type plug
  slave{
    pcm "hw: 0,0"
  }
}

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^ Go To Line
```

Fuente: Elaboración propia

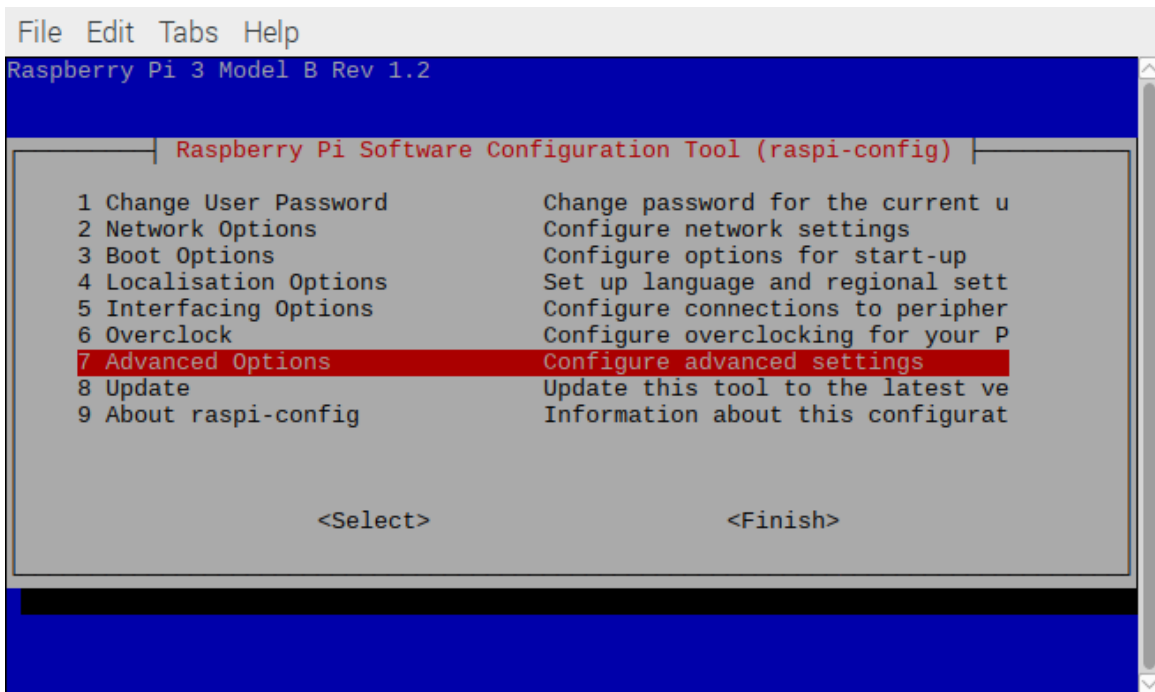
Figura 2-9. Configuración de /etc/asound.conf

Tal como se observa dentro de pcm.mic y pcm.speaker existen las líneas con los valores “hw: X,Y” donde X corresponde al número de la tarjeta, e Y al número de dispositivo. Se reemplazan los valores de X e Y por los obtenidos en el paso anterior, donde pcm.mic corresponde al micrófono y pcm.speaker al altavoz, de forma que reconozca esos dispositivos por defecto.

#### 2.2.4 Forzar las salidas de audio

La Raspberry Pi tiene dos posibles salidas de audio: La salida HDMI y la salida de audio de 3.5 [mm]. Si bien es cierto, en el paso previo se estableció como salida de audio esta última, es preferible forzar la salida de audio por medio del comando raspi-config.

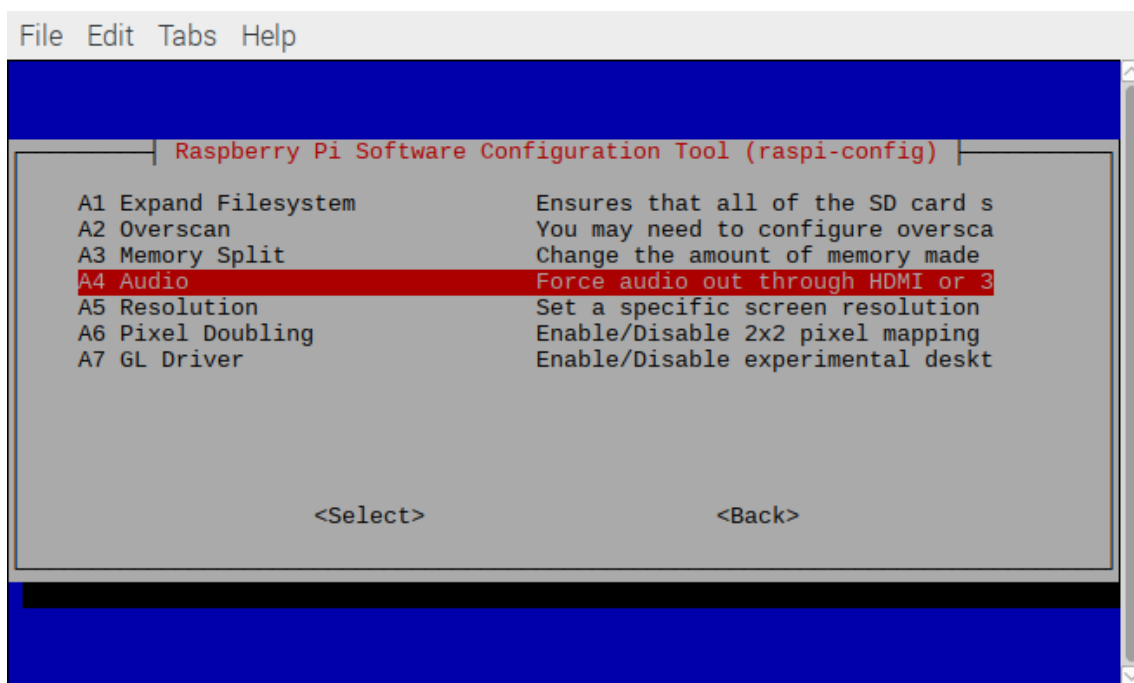
El comando raspi-config es una herramienta de configuración de la Raspberry Pi. Las posibles acciones a realizar se muestran en la figura 2-10.



Fuente: Elaboración propia

Figura 2-10. Menú de raspi-config

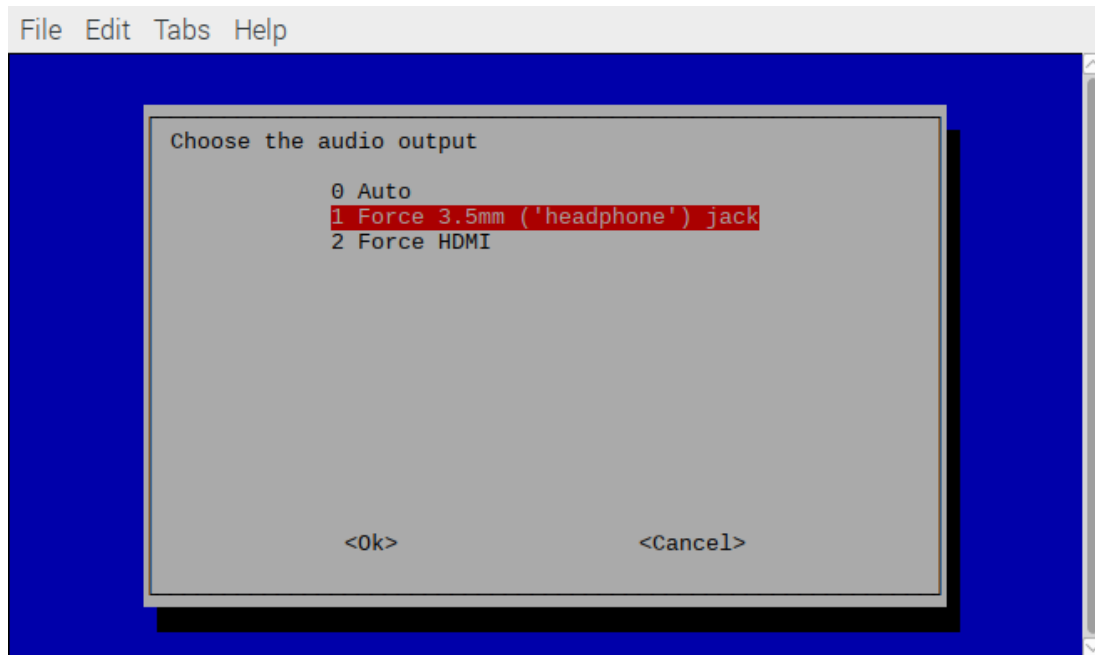
El poder forzar la salida de audio se realiza en la alternativa de “opciones avanzadas”. Las alternativas disponibles dentro de esa opción se muestran en la figura 2-11.



Fuente: Elaboración propia

Figura 2-11. Opciones de audio

Se selecciona la opción A4 de forzado de sonido y en el nuevo menú, se selecciona la opción de forzar el Jack de 3.5 [mm], tal como se observa en la figura 2-12.



*Fuente: Elaboración propia*

Figura 2-12. Forzado de salida.

Con el forzado de la salida de audio, se termina la configuración del hardware. El siguiente paso, es la configuración de los servicios a utilizar como asistente. En este caso, se utilizarán los servicios entregados por Google.

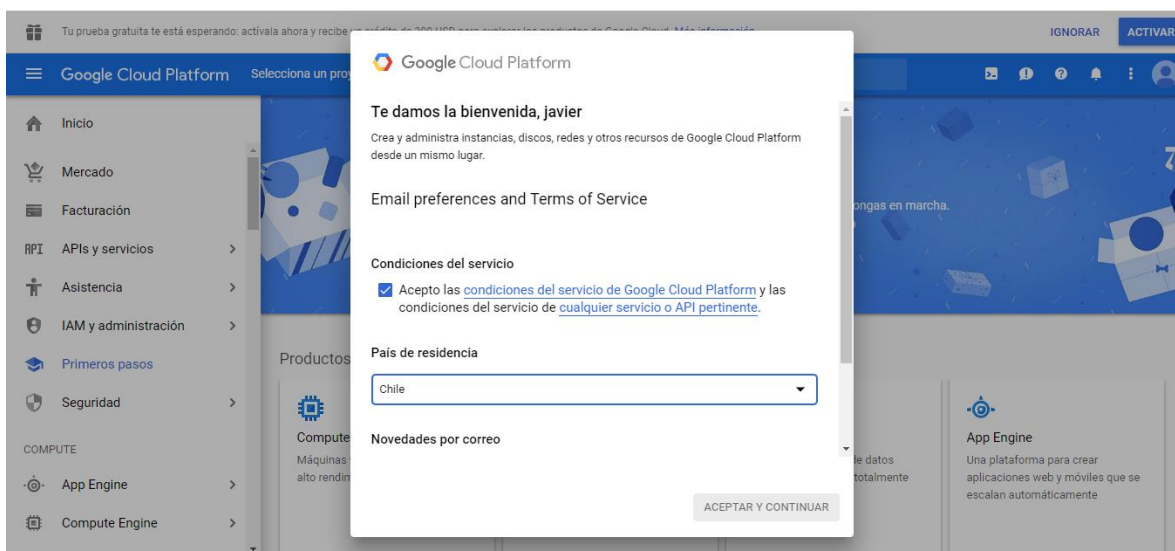
### **2.3 ETAPA 3: INSTALACIÓN DE GOOGLE ASSISTANT EN LA RASPBERRY PI**

Google Assistant es el asistente inteligente de Google. Surgió en el año 2016 de forma exclusiva para dispositivos de esa empresa, tales como su parlante Google Home o teléfonos de la gama Pixel. Es en el año 2017 cuando Google libera las herramientas para desarrolladores de su aplicación, de forma que éstos sean capaces de implementar el asistente en sus proyectos. La selección de este asistente está basado en el hecho de que otros tales como Siri, Alexa, o Mycroft, aún no cuentan con las correspondientes herramientas de desarrolladores, o es necesaria la creación de cuentas adicionales para su control. Sin embargo, las herramientas de Google Assistant se encuentran disponibles a toda persona que tenga una cuenta en Google.

Otra de las plataformas de la misma empresa que deben ser usadas durante este proceso, es Google Cloud, la cual unifica todas las aplicaciones de desarrollo web de Google.

En primera instancia, es necesario tener una cuenta de Google creada para acceder a los servicios a utilizar durante el desarrollo de este proyecto. Se omite el proceso de creación de cuenta de correo, de forma de simplificar el proceso.

El segundo paso consiste en acceder con la cuenta de Google ya creada a la siguiente dirección web: <https://console.cloud.google.com>. La apariencia del entorno se observa en la figura 2-13.



*Fuente: Elaboración propia*

Figura 2-13. Acceso a Google Cloud

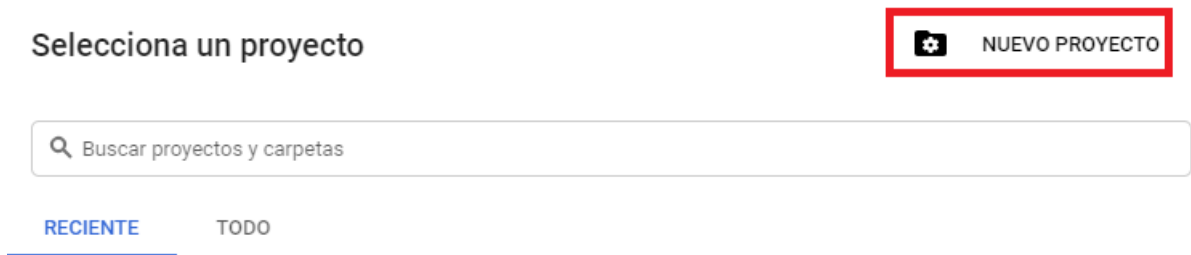
En este paso es necesario aceptar las condiciones de servicio de Google, indicar el país de residencia, e indicar si el desarrollador desea recibir actualizaciones sobre la plataforma. Luego, es necesario crear un nuevo proyecto. Para realizar esto, se accede a la pestaña “Selecciona un proyecto” en la parte superior izquierda tal como muestra la figura 2-14.



*Fuente: Elaboración propia*

Figura 2-14. Creación de nuevo proyecto

Al oprimir esa opción, se despliega una nueva ventana de creación de proyectos como se observa en la figura 2-15. Acá se selecciona la opción “Nuevo Proyecto”.



Fuente: Elaboración propia

Figura 2-15. Indicación de nuevo proyecto

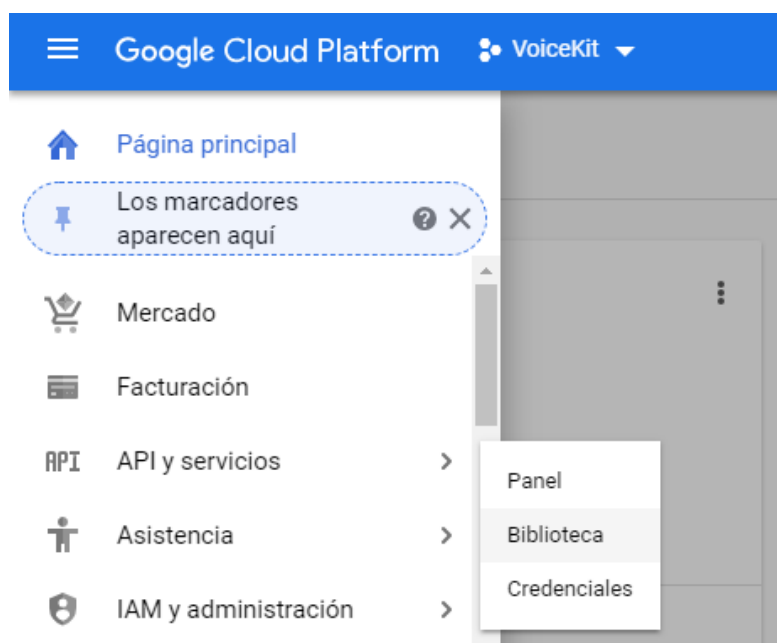
En la siguiente ventana se ingresa el nombre del proyecto. Es importante destacar, que existe un límite para los proyectos a desarrollar dentro de este entorno, esto debido a motivos de seguridad de Google (no generar picos de solicitudes de información). Se escribe el nuevo nombre de la aplicación en la casilla llamada “Nombre del proyecto” (ver figura 2-16). En este caso, no es necesario ingresar el nombre de una ubicación. Luego de ingresar el nombre del proyecto se vuelve a la pantalla principal.

Fuente: Elaboración propia

Figura 2-16. Nombre del nuevo proyecto

Al volver a la página de entrada, se observan una serie de estadísticas de uso de la aplicación, reportes de errores y otros. El proyecto está creado, pero no se encuentra asociado a ninguna API. Una API es un método de acceder a una funcionalidad de una determinada aplicación. En este caso, la Raspberry Pi, querrá utilizar la API de Google Assistant, por lo que es necesario darle permisos a la placa de forma que pueda conectarse a ella.

Para poder acceder a la API de Google Assistant, en el menú de navegación, se accede a la opción “API y servicios” y dentro de ésta, se selecciona la opción Biblioteca. La imagen 2-17 muestra el proceso general.



*Fuente: Elaboración propia*

Figura 2-17. Acceso a biblioteca de APIs

En la casilla de Buscar “APIs y Servicios” se busca “Google Assistant” y se selecciona la opción “Crear Credencial”. Se desplegará una ventana tal como la figura 2-18.

## Credenciales

## Añadir credenciales al proyecto

## 1 Averigua qué tipo de credenciales necesitas

Te ayudaremos a configurar las credenciales adecuadas.  
Puedes saltarte este paso y crear una [clave de API](#), un [ID de cliente](#) o una [cuenta de servicio](#).

## ¿Qué API estás utilizando?

Las API utilizan diversas plataformas de autorización, y se pueden restringir algunas credenciales para que solo llamen a ciertas API.

Google Assistant API

## ¿Desde dónde llamarás a la API?

Para restringir las credenciales, se puede tener en cuenta el contexto de la llamada. No obstante, no es seguro usar algunas credenciales en contextos determinados.

Otra UI (por ejemplo, Windows, herramienta de CLI)

## ¿A qué tipo de datos accederás?

En función del tipo de datos que solicites, se precisan unas credenciales determinadas para autorizar el acceso.

- Datos de usuario  
Accede a datos pertenecientes a un usuario de Google (con su permiso)
- Datos de aplicación  
Accede a datos pertenecientes a tu propia aplicación

¿Qué credenciales necesito?

Fuente: Elaboración propia

Figura 2-18. Búsqueda de credenciales

Para determinar la credencial necesaria para el proyecto, se indica que la API a utilizar es la Google Assistant API, que se llamará desde “Otra UI” y busca acceder a los datos del usuario. En base a esa selección se crea una nueva ID de OAuth con un nombre a elección. La figura 2-19 muestra el procedimiento.

VoiceKit

Credenciales

Añadir credenciales al proyecto

Averigua qué tipo de credenciales necesitas  
Llamar a Google Assistant API desde una plataforma basada en UI

2 Crear un ID de cliente de OAuth 2.0

Nombre

Crear ID de cliente de OAuth

3 Configurar la pantalla de autorización de OAuth 2.0

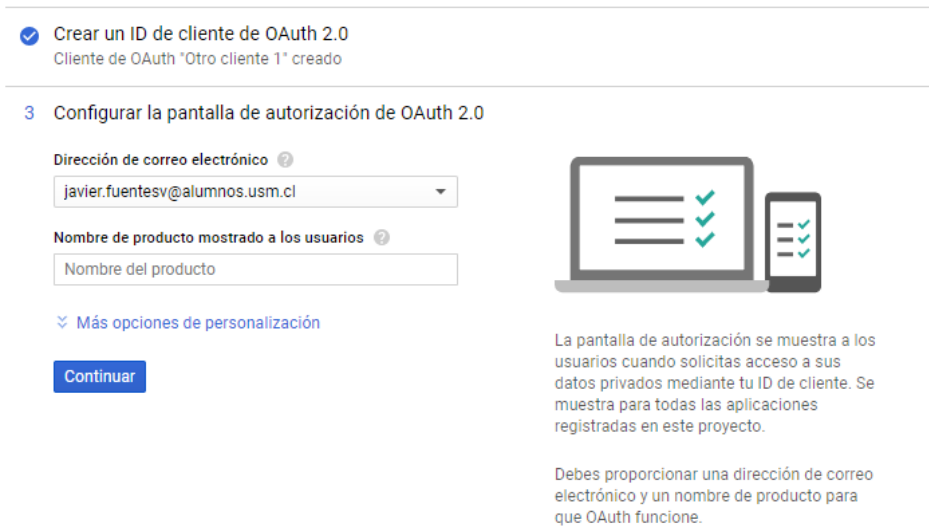
4 Descargar credenciales

Cancelar

Fuente: Elaboración propia

Figura 2-19. Nombre de nuevo cliente

Después de haber creado la nueva ID de cliente, se configura la pantalla de autorización de OAuth 2.0, indicando un correo electrónico y un nombre de dispositivo (opcional el último). La figura 2-20 muestra la ventana a completar.



✓ Crear un ID de cliente de OAuth 2.0  
Cliente de OAuth "Otro cliente 1" creado

3 Configurar la pantalla de autorización de OAuth 2.0

Dirección de correo electrónico ?  
javier.fuentesv@alumnos.usm.cl

Nombre de producto mostrado a los usuarios ?  
Nombre del producto

⌵ Más opciones de personalización

Continuar

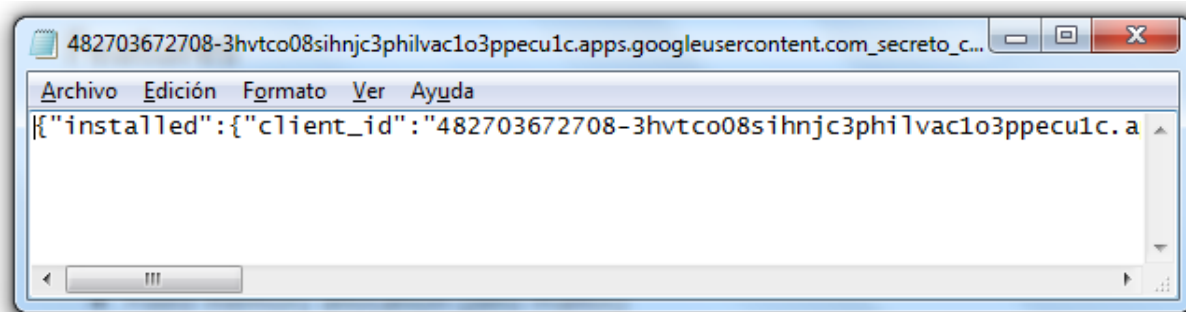
La pantalla de autorización se muestra a los usuarios cuando solicitas acceso a sus datos privados mediante tu ID de cliente. Se muestra para todas las aplicaciones registradas en este proyecto.

Debes proporcionar una dirección de correo electrónico y un nombre de producto para que OAuth funcione.

Fuente: Elaboración propia

Figura 2-20. Configuración de pantalla de autorización de OAuth

Luego se despliega la opción de descargar las credenciales. Las cuales deben ser abiertas en algún editor de texto (pese a que inicialmente no lo reconozca como tal el SO). La figura 2-21 muestra la forma en la cual se ve el nuevo tipo de credencial.



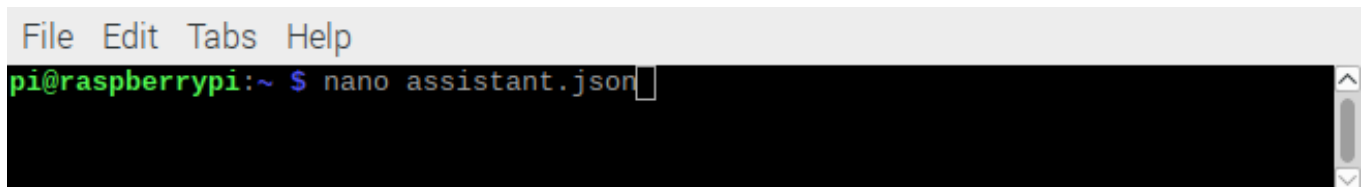
```
{\"installed\": {\"client_id\": \"482703672708-3hvtco08sihnjc3philvac1o3ppecu1c. a
```

Fuente: Elaboración propia

Figura 2-21. Código en formato .txt

El texto que se encuentra en el interior se copia y luego se ingresa a la consola de Linux (esto es posible también realizado mediante un servicio de SSH).

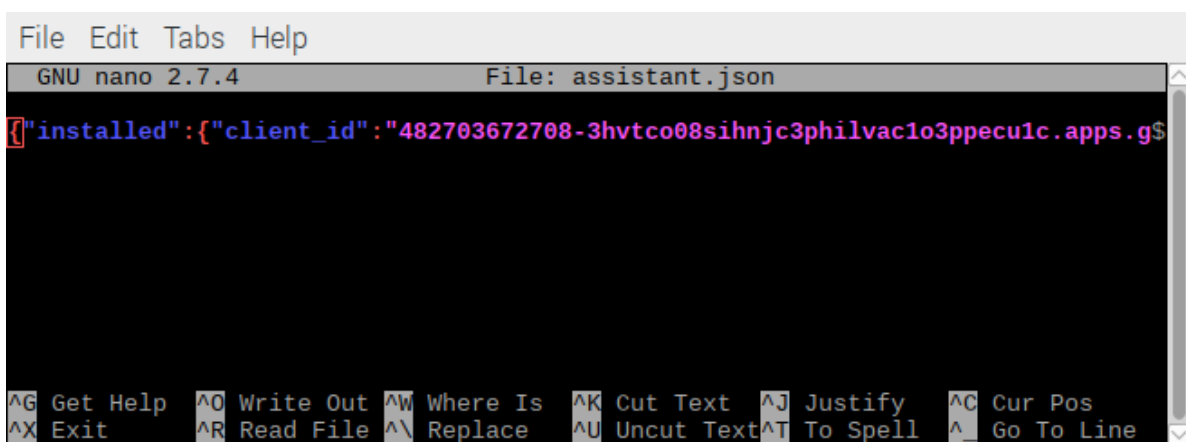
Se crea un nuevo archivo de texto mediante el comando nano. El nombre de este archivo es assistant.json. La figura 2-22 muestra la forma en que se ingresa el comando en la terminal. La figura 2-23 muestra el aspecto que queda en el archivo de texto.



```
File Edit Tabs Help
pi@raspberrypi:~ $ nano assistant.json
```

Fuente: Elaboración propia

Figura 2-22. Comando nano

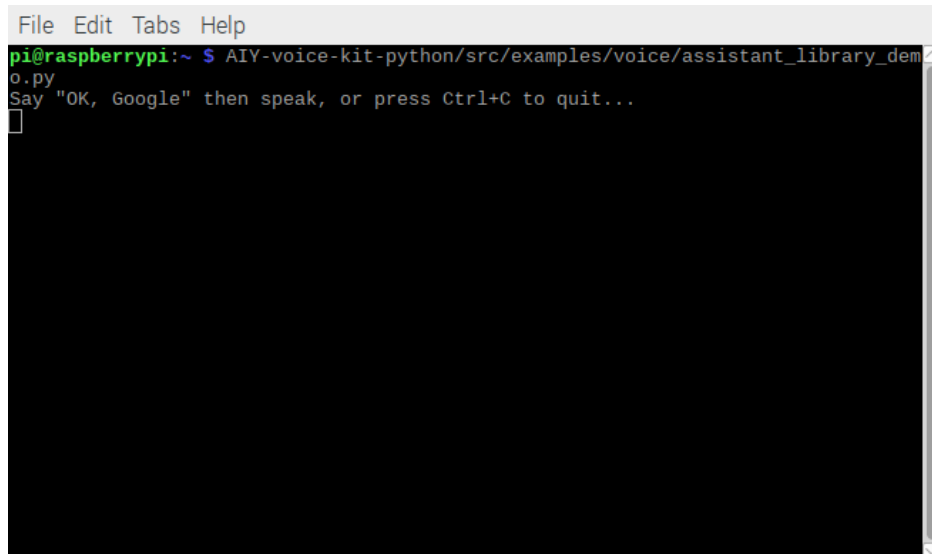


```
File Edit Tabs Help
GNU nano 2.7.4 File: assistant.json
{"installed":{"client_id":"482703672708-3hvtco08sihnjc3philvac1o3ppecu1c.apps.g$
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Fuente: Elaboración propia

Figura 2-23. Código en consola

Al querer ejecutar por primera vez el programa pedirá ciertas autorizaciones de Google junto con entregar una dirección de correo electrónico para descargar el código de autorización de la aplicación. Ingresando el código ya es posible ejecutar el asistente por medio del comando mostrado en la figura 2-24.



```
File Edit Tabs Help
pi@raspberrypi:~ $ AIY-voice-kit-python/src/examples/voice/assistant_library_demo.py
Say "OK, Google" then speak, or press Ctrl+C to quit...

```

*Fuente: Elaboración propia*

Figura 2-24. Asistente funcionando

Con este último paso se termina la configuración del asistente de Google en la Raspberry Pi.

## 2.4 **ETAPA 4: CREACIÓN DE PROGRAMA DE CONTROL POR VOZ**

Dialogflow es una herramienta de Google que permite la creación y desarrollo de interfaces conversacionales, gracias a IA (Inteligencia Artificial). Se utiliza este programa debido a que es posible hacer uso de estas herramientas utilizando la misma cuenta para Google Cloud. Entre las características que se destacan de Dialogflow se encuentran las siguientes:

- Posibilidad de comunicación texto-voz y voz-texto: Debido a lo cual puede entregar información ya sea en forma sonora, visual o ambas.
- Cognitiva: Posee por defecto algoritmos de procesamiento de lenguaje natural, por lo que es posible entrenarlo sin necesidad de desarrollar un programa extenso.
- Contextual: Reconoce el significado de las palabras en función del contexto en el cual se encuentran.

Es importante destacar que este trabajo no busca explicar nociones de IA, dado que el software funciona en forma cerrada basado en los parámetros definidos por el usuario. La dirección de la página de Dialogflow es: <https://dialogflow.com/>. El método de ingreso es por el correo electrónico de Google.

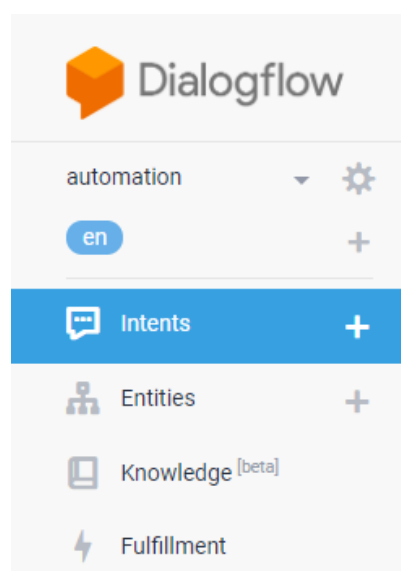
Existen tres conceptos fundamentales para la comprensión de las interfaces conversacionales: Intents, Entities y Fulfillment.

- Intents: Es una meta, u objetivo que se quiere lograr tal como enviar un mensaje, preguntar sobre el clima, o encender un dispositivo. Un proyecto puede ser compuesto de distintos Intents.
- Fulfillment: Es un servicio, aplicación u otro proceso lógico encargado de manejar un Intent y realizar una determinada acción.
- Entities: Representa un objeto real que puede ser usado en las conversaciones. Al referirnos a objeto real, no necesariamente significa algo físico tal como un dispositivo, ya que también un número, o una fecha es un entity.

Dado que con el programa se busca el control de cuatro periféricos conectados a una placa NodeMCU, en este caso se usarán 2 Entities: device (dispositivo) y status (estado). El programa Dialogflow viene por defecto con una serie de Entities de amplio uso. En caso de no encontrarse el buscado, es posible crear los que sean necesarios.

Lo que se busca crear es un programa que sea capaz de discernir dentro de una oración el concepto de encendido/apagado, además del nombre (o número del dispositivo a controlar), y guardar esa información en una base de datos provista por Firebase (otro servicio de Google), la que a su vez estará comunicada directamente con la placa de desarrollo NodeMCU.

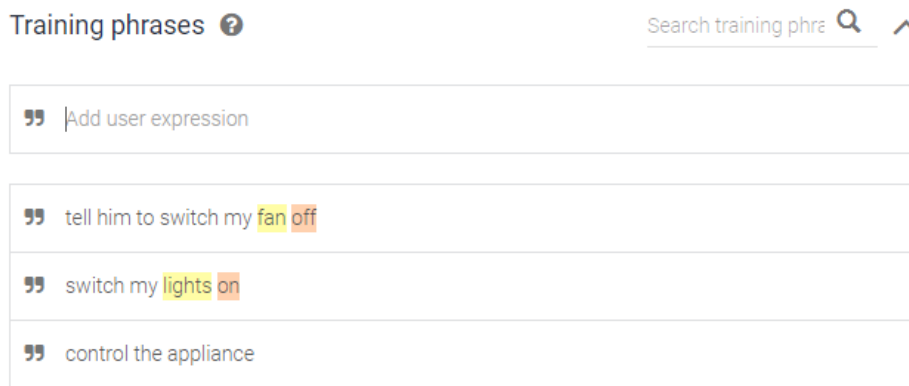
Al ingresar a la página de Dialogflow, en la izquierda se ve un menú como el de la figura 2-25.



*Fuente: Elaboración propia*

Figura 2-25. Menú de navegación

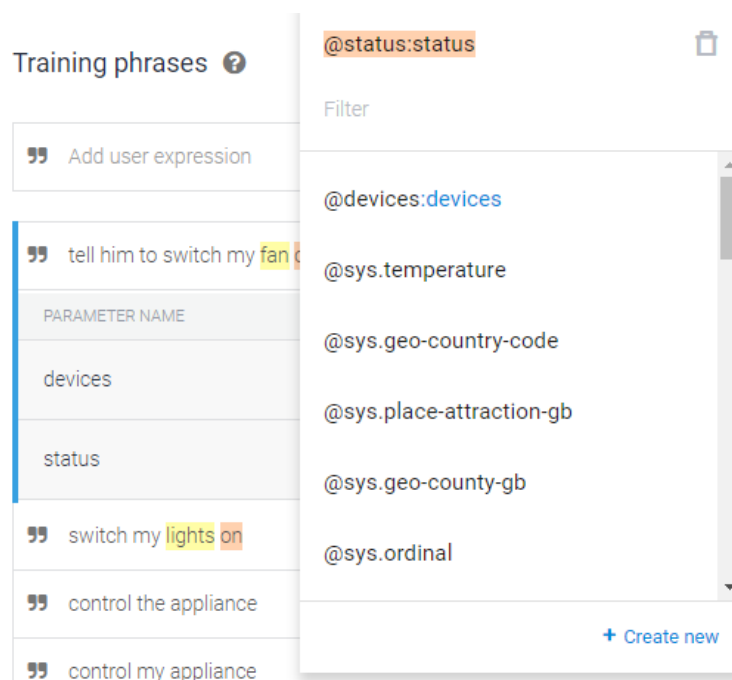
Al acceder al menú Intents, es posible indicar frases de entrenamiento al software, de manera que sea capaz de identificar los Entities en esta oración. La figura 2-26 muestra la forma en la cual se agregan las frases de entrenamiento en Intents.



Fuente: Elaboración propia

Figura 2-26. Frases de entrenamiento

Existen dos textos que se encuentran seleccionados con distintos colores. Para poder reconocer los parámetros útiles de la oración, se seleccionan las palabras claves y se identifica el parámetro, tal como se muestra en la figura 2-27.



Fuente: Elaboración propia

Figura 2-27. Entities

Los parámetros rescatados aparecen bajo las frases de entrenamiento en la subsección “Action and parameters”, de la forma mostrada en la figura 2-28.

Action and parameters ^

control.appliance

REQUIRED <small>?</small>	PARAMETER NAME <small>?</small>	ENTITY <small>?</small>	VALUE	IS LIST <small>?</small>	PROMPTS <small>?</small>
<input checked="" type="checkbox"/>	devices	@devices	Sdevices	<input type="checkbox"/>	Which appliance...
<input checked="" type="checkbox"/>	status	@status	Sstatus	<input type="checkbox"/>	Do you want me ...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	–

[+ New parameter](#)

Fuente: Elaboración propia

Figura 2-28. Parámetros

#### 2.4.1 Creación de Fulfillment

Tal como fue mencionado, el Fulfillment corresponde al proceso lógico utilizado para obtener una salida en base a las entradas de audio dadas a Dialogflow. Es posible implementar un Fulfillment de dos formas, ya sea usando el editor online, o utilizar un Webhook.

- **Inline Editor:** Es un editor que viene por defecto dentro de la página de Dialogflow. Solamente permite la programación en NodeJS y tiene funciones limitadas. Una de las desventajas que presenta para propósitos del proyecto es su incapacidad de comunicación con bases de datos.
- **Webhook:** Permite la programación de aplicaciones en lenguajes distintos a NodeJS, además de programas más extensos.

Debido a lo mencionado anteriormente, se desarrollará un Webhook que permita la interacción con la base de datos de Firebase (Realtime Database). Para lograr eso es necesario instalar Firebase, esto es posible realizarlo en la Raspberry Pi, o en otro equipo con sistema operativo Linux.

#### 2.4.2 Instalación de Firebase en el sistema operativo

Firebase es una plataforma de desarrollo móvil de Google. Permite crear aplicaciones web que trabajen en la nube, de tal forma que se pueden acceder a datos en tiempo real. Ofrece servicios tales como configuración remota de equipos, testeo, mensajería en la nube y otros. En particular el uso que se le da a Firebase consiste en utilizar las bases de datos ofrecidas (Realtime Database). Para tener una cuenta en Firebase, al igual que con Actions on Google y Dialogflow, solamente es necesario tener una cuenta de Google para acceder. Al igual que los otros servicios antes mencionados, es posible crear cuentas sin costo económico, solamente con la restricción del número de proyectos a realizar.

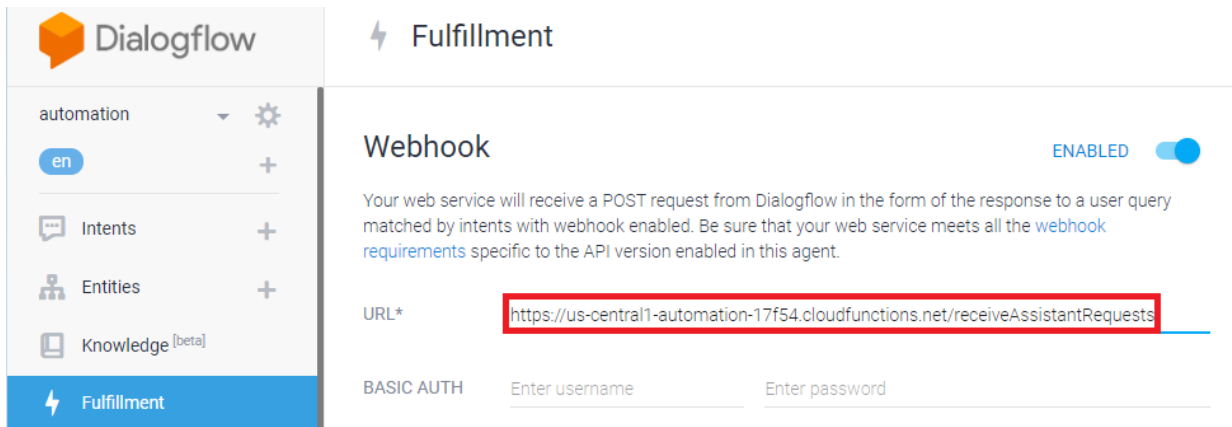
Para instalar Firebase es necesaria la instalación previa de dos programas llamados npm y nodejs, los cuales pueden ser instalados mediante el comando de Linux `sudo apt-get install`. Para la instalación de Firebase propiamente tal se utiliza el comando:

```
npm install -g firebase-tools
```

Esto instalará Firebase para todos los usuarios (no solamente aquellos que tienen permisos de administrador). El registro correspondiente se hace mediante el comando: `firebase login`. Al ejecutar este comando, se accede a internet para registrarse con la cuenta de Google correspondiente. Luego, se crea una carpeta nueva en donde se utiliza la función `firebase init functions`, la cual permite la creación de archivos que serán adjuntados en la dirección del Webhook.

El proceso anterior crea un archivo llamado `index.js`, el cual contiene las instrucciones que serán ejecutadas por el Webhook. El contenido del archivo se adjunta en el ANEXO-A.

Al terminar de editar el archivo `index.js` se ejecuta el comando `firebase deploy`, el cual entregará la dirección web que es insertada en la sección Fulfillment, tal como se muestra en la figura 2-29.



Fuente: Elaboración propia

Figura 2-29. URL del Webhook

## 2.5 ETAPA 5: PROGRAMACIÓN DEL NODEMCU

NodeMCU es tanto firmware OpenSource como una placa de desarrollo basada en el microcontrolador ESP8266. La particularidad de este dispositivo es que posee la capacidad de conectarse a internet, por lo que es ampliamente utilizado en aplicaciones relacionadas con el internet de las cosas. Algunas de sus características principales son presentadas en la tabla 2-2.

Tabla 2-2. Características del NodeMCU

Características Generales	
Pines de Entrada Salida	17 (multiplexados con otras funciones)
Convertor Análogo/Digital	1 (Resolución de 1024)
Alimentación	3.3[V]
Salida	3.3[V]
Tipo de conexión	WIFI

Fuente: Elaboración propia

El lenguaje de programación utilizado para la programación de la placa de desarrollo es un lenguaje llamado LUA, el cual no es de amplio uso, debido a eso, existen ciertas librerías para la programación del microcontrolador en el ARDUINO IDE (Entorno de desarrollo integrado).

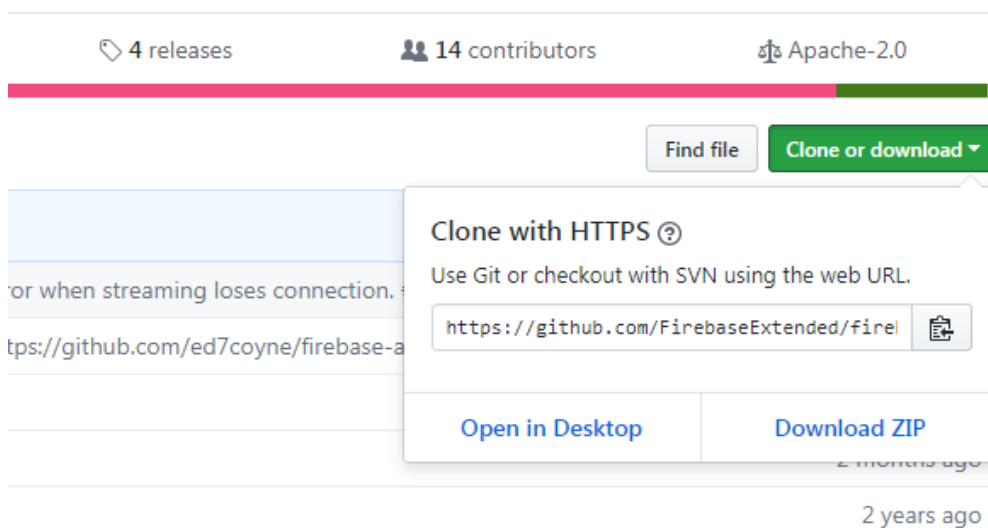
Otro de los detalles a tener presente, es que tal como fue mencionado previamente, se busca la modificación en tiempo real de los valores (los que estarán almacenados en la Realtime

Database de Firebase), por lo que se necesitan las bibliotecas que permitan la interacción del NodeMCU con Firebase.

Se necesitan específicamente tres librerías:

- Arduino JSON Master: Es una librería en C++ que fue creada para aplicaciones de IOT para Arduino y otras placas. Es descargable desde la siguiente dirección: <https://github.com/bblanchon/ArduinoJson>
- Firebase-Arduino-Master: Es la librería que permite la comunicación entre Firebase y Arduino. No funciona de no estar instalada la primera librería mencionada. Se encuentra para su descarga en la siguiente página: <https://github.com/FirebaseExtended/firebase-arduino>
- ESP8266: Permite el control de los dispositivos basados en el microcontrolador ESP8266 (como el caso del NodeMCU).

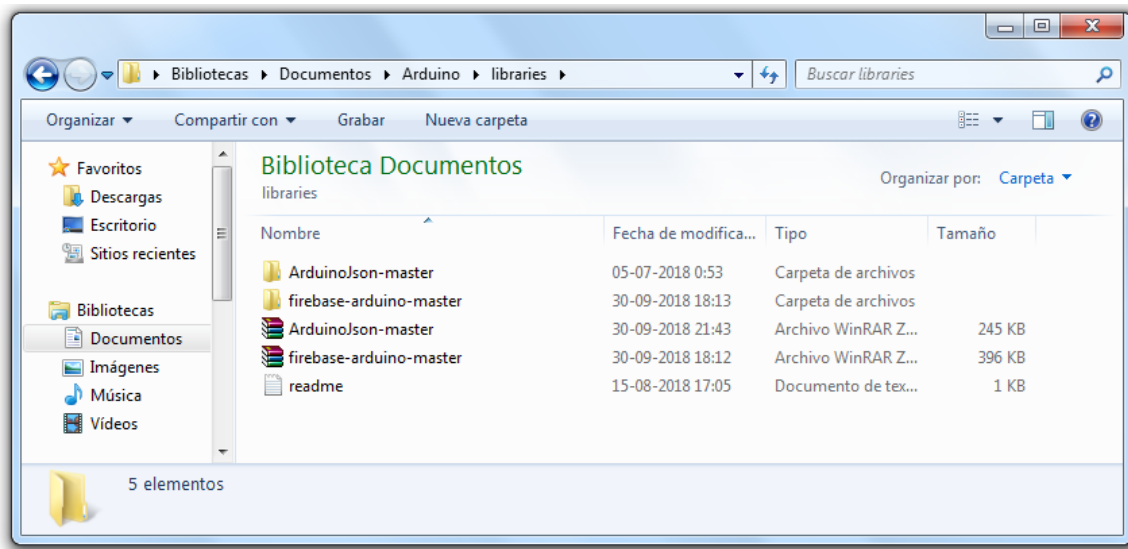
Para instalar librerías en el Arduino IDE existen diversos métodos. El utilizado con las primeras dos librerías consiste en ir a las correspondientes direcciones en Github y descargar la librería comprimida en formato .zip (dado que al incorporarla en algún programa las descomprime solas). La figura 2-30 muestra como descargar las librerías.



Fuente: Elaboración propia

Figura 2-30. Descarga de librerías

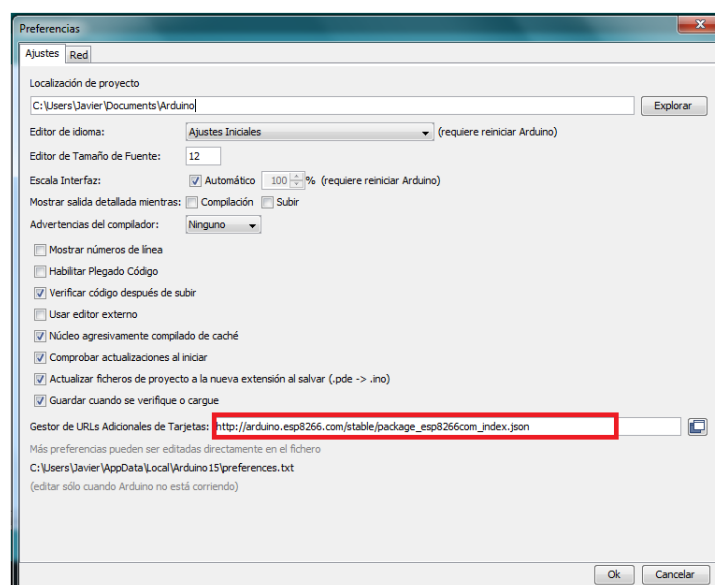
Luego de descargadas las librerías simplemente se copian en la carpeta Arduino/Libraries, como se muestran en la figura 2-31.



Fuente: Elaboración propia

Figura 2-31. Incorporación de librerías

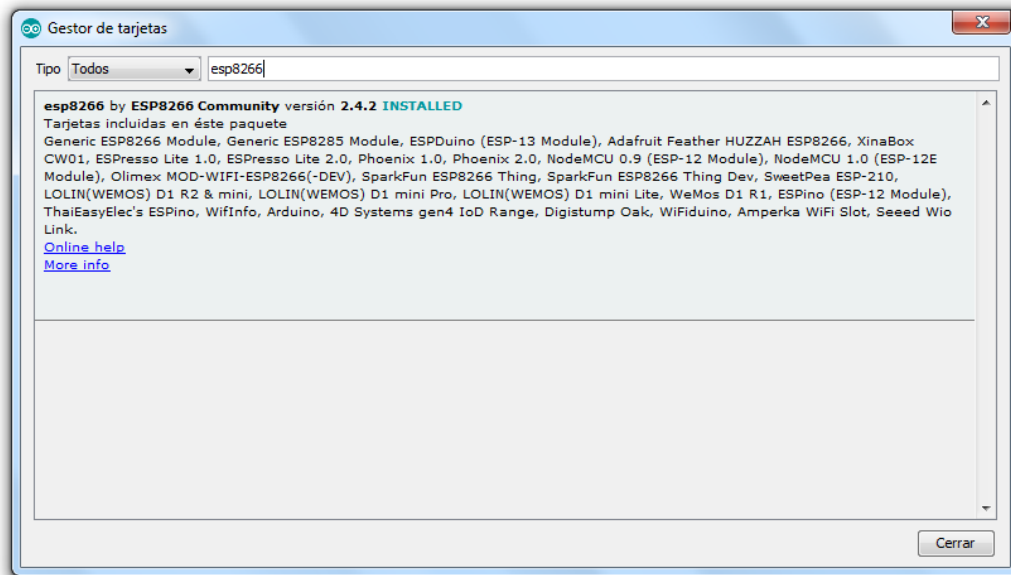
Para utilizar la biblioteca ESP8266 es diferente, debido a que no existe NodeMCU como tarjeta seleccionable, debido a eso es que se necesita ingresar información adicional. Para esto se accede al menú de preferencias dentro del ARDUINO IDE y se agrega una URL adicional para poder acceder a las nuevas placas, tal como se observa en la figura 2-32. La URL a agregar es: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json).



Fuente: Elaboración propia

Figura 2-32. Actualización de placas

La dirección introducida solamente da información respecto a donde hallar las tarjetas adicionales, pero aún no se ha instalado la tarjeta necesaria. Para hacer eso, es necesario acceder al gestor de tarjetas y buscar “ESP8266”, de la forma mostrada en la figura 2-33.



Fuente: Elaboración propia

Figura 2-33. Placas basadas en ESP8266

Al instalar la tarjeta, ya es posible utilizar el NodeMCU con el IDE de ARDUINO. Existen diversas formas de verificar el correcto funcionamiento para la tarjeta. En la sección ANEXO-B se adjuntará el código de funcionamiento de la placa NodeMCU.

El Diagrama General de la solución del problema se indica en el ANEXO C – DIAGRAMA DE LA SOLUCIÓN.

### **CAPÍTULO 3: EVALUACIÓN DE COSTOS**

### 3. EVALUACIÓN DE COSTOS DEL PROYECTO

El proyecto a realizar cuenta con las siguientes limitaciones:

- Cantidad de proyectos en paralelo: Los proyectos disponibles a realizar en Google son un máximo de 10. Un número mayor de proyectos implica la adquisición de nuevos tipos de cuenta para Firebase, Actions on Google y Google Cloud, los que a diferencia de las consideradas en el desarrollo del proyecto no son gratis.
- El valor de ciertos componentes son susceptible a cambio, debido a que los precios están en euros. Se realiza conversión a pesos chilenos (CLP).
- Se consideran como dispositivos a controlar luces led convencionales (4 en este caso). Es posible conectar otros dispositivos, pero para propósitos de estimación de costos, se consideran luces dado que son el dispositivo más genérico a controlar.

#### 3.1 CARTA GANTT

La particularidad de este proyecto es que la implementación del equipo es realizada en muy poco tiempo, sin embargo, la mayor parte del desarrollo del proyecto es en cuánto a labores de programación del equipo. Debido a eso, la Carta Gantt presentada en la figura 3-1 muestra los trabajos realizados a cabo en investigación y desarrollo.

Tabla 3-1. Diagrama Gantt del proyecto

ACTIVIDADES		Mes 01				Mes 02					Mes 03		
		1	2	3	4	1	2	3	4	5	1	2	3
1	Adquisición de equipamiento	█											
2	Instalación de Software	█											
3	Configuración y modificación de archivos		█	█	█								
4	Instalación de Google Assistant			█	█								
5	Instalación de Firebase y programación			█	█	█	█	█	█	█			
6	Programación de Dialogflow				█	█	█	█	█	█	█	█	█
7	Programación de Node MCU				█	█	█	█	█	█	█	█	█
8	Etapas de potencia										█	█	█
9	Pruebas de funcionamiento		█	█	█	█	█	█	█	█	█	█	█

Fuente: Elaboración propia

### 3.2 COSTOS DE HARDWARE

La tabla 3-2, indica los componentes necesarios para montar la integridad del sistema. Los elementos tales como mouse, teclado y pantalla no son considerados dentro del costo, debido a que solamente son utilizados para realizar la configuración inicial del dispositivo.

Tabla 3-2. Costos de hardware del centro de control doméstico

N°	Componente	Precio (CLP)	Referencia
1	Tarjeta MicroSD	8.990	<a href="https://nuevo.jumbo.cl/micro-sd-sandisk-16-gb/p">https://nuevo.jumbo.cl/micro-sd-sandisk-16-gb/p</a>
2	Raspberry Pi Modelo 3B	25.932	<a href="https://www.ebay.es/itm/Raspberry-Pi-3-Model-B/264047691320?hash=item3d7a780638:g:X5wAAOSw3LVb8Zm7">https://www.ebay.es/itm/Raspberry-Pi-3-Model-B/264047691320?hash=item3d7a780638:g:X5wAAOSw3LVb8Zm7</a>
3	Carcasa de protección de Raspberry Pi	3.990	<a href="https://www.pcfactory.cl/producto/27580-carcasa-para-raspberry-pi-b-pi-2-pi3-transparente-base-">https://www.pcfactory.cl/producto/27580-carcasa-para-raspberry-pi-b-pi-2-pi3-transparente-base-</a>
4	Cable Jack-Jack	1.790	<a href="https://www.pcfactory.cl/producto=23269-Cable Audio Estéreo 3 5mm a 3 5mm - 2mts">https://www.pcfactory.cl/producto=23269-Cable Audio Estéreo 3 5mm a 3 5mm - 2mts</a>
5	Parlante Bluetooth	4.990	<a href="https://simple.ripley.cl/parlante-lg-ph1-2000363125073p">https://simple.ripley.cl/parlante-lg-ph1-2000363125073p</a>
6	Cargador de Raspberry Pi	8.771	<a href="https://www.ebay.es/itm/Raspberry-Pi-Official-Universal-Power-Supply-Unit/253149174824?hash=item3af0dddc28:g:pYgAAOSwt5tZ1N64:rk:1:pf:0">https://www.ebay.es/itm/Raspberry-Pi-Official-Universal-Power-Supply-Unit/253149174824?hash=item3af0dddc28:g:pYgAAOSwt5tZ1N64:rk:1:pf:0</a>
7	Micrófono USB	1.100	<a href="https://www.ebay.es/itm/F15D-USB-Plug-Mini-Studio-Microphone-Audio-MIC-For-Computer-PC-Laptop-Desktop/173614421730?hash=item286c39d6e2:g:GusAAOSwjZlb138U:rk:1:pf:0">https://www.ebay.es/itm/F15D-USB-Plug-Mini-Studio-Microphone-Audio-MIC-For-Computer-PC-Laptop-Desktop/173614421730?hash=item286c39d6e2:g:GusAAOSwjZlb138U:rk:1:pf:0</a>
8	NodeMCU	2.213	<a href="https://www.ebay.es/itm/NodeMcu-Lua-WIFI-Internet-Things-Development-Board-Based-ESP8266-CP2102-Modulo/112459553278?hash=item1a2f1c99fe:g:kGAAAOSwyupa8plb:rk:16:pf:0">https://www.ebay.es/itm/NodeMcu-Lua-WIFI-Internet-Things-Development-Board-Based-ESP8266-CP2102-Modulo/112459553278?hash=item1a2f1c99fe:g:kGAAAOSwyupa8plb:rk:16:pf:0</a>

9	Placa de Relés	2.183	<a href="https://www.ebay.es/itm/5V-ModULO-RELe-4-CANALES-PARA-PIC-ARM-AVR-DSP-AVR-MSP430-L5X9/173493370685? trkparms=aid%3D555017%26algo%3DPL.CASSINI%26ao%3D1%26asc%3D20180221161347%26meid%3D71ef356887a948c88fea86d453db45c5%26pid%3D100505%26rk%3D1%26rkt%3D1%26%26itm%3D173493370685&amp; trksid=p2045573.c100505.m3226">https://www.ebay.es/itm/5V-ModULO-RELe-4-CANALES-PARA-PIC-ARM-AVR-DSP-AVR-MSP430-L5X9/173493370685? trkparms=aid%3D555017%26algo%3DPL.CASSINI%26ao%3D1%26asc%3D20180221161347%26meid%3D71ef356887a948c88fea86d453db45c5%26pid%3D100505%26rk%3D1%26rkt%3D1%26%26itm%3D173493370685&amp; trksid=p2045573.c100505.m3226</a>
10	Luces LED (x4)	23.160	<a href="https://www.sodimac.cl/sodimac-cl/product/3156079/Pack-de-ampolletas-LED-E-27-40-W-Calida-2-unidades">https://www.sodimac.cl/sodimac-cl/product/3156079/Pack-de-ampolletas-LED-E-27-40-W-Calida-2-unidades</a>
11	Transformador AC/DC 5[V]	3.900	<a href="https://www.mcielectronics.cl/shop/product/transformador-ac-dc-5v-1amp-jack-dc-5-5x2-1mm-19600">https://www.mcielectronics.cl/shop/product/transformador-ac-dc-5v-1amp-jack-dc-5-5x2-1mm-19600</a>
12	Conector para alimentación DC	380	<a href="https://www.mcielectronics.cl/shop/product/conector-para-alimentacion-dc-compatible-con-protoboard-10367">https://www.mcielectronics.cl/shop/product/conector-para-alimentacion-dc-compatible-con-protoboard-10367</a>
12	<u>SPARKFUN LOGIC LEVEL CONVERTER - BI-DIRECTIONAL</u>	1.990	<a href="https://www.sparkfun.com/products/12009">https://www.sparkfun.com/products/12009</a>
Total		\$89.579	

Fuente: Elaboración propia

### 3.3 COSTOS DE SOFTWARE

En la tabla 3-3 se mencionan los distintos servicios y programas necesarios para la programación del centro de control domótico, junto con sus correspondientes costos.

Tabla 3-3. Costos de software

Software	Precio	Referencia
Raspbian AIY Proyects	Libre	<a href="https://github.com/google/aiyprojects-raspbian">https://github.com/google/aiyprojects-raspbian</a>
Etcher.io	Libre	<a href="https://www.balena.io/etcher/">https://www.balena.io/etcher/</a>
Google Cloud	Libre	<a href="https://console.cloud.google.com">https://console.cloud.google.com</a>
Actions on Google	Libre	<a href="https://developers.google.com/actions/">https://developers.google.com/actions/</a>

Firestore	Libre	<a href="https://firebase.google.com/?hl=es-419">https://firebase.google.com/?hl=es-419</a>
Arduino IDE	Libre	<a href="https://www.arduino.cc/en/Main/Software">https://www.arduino.cc/en/Main/Software</a>
<b>Total</b>		<b>\$0</b>

Fuente: Elaboración propia

### 3.4 COSTOS DE PERSONAL

Debido a que el centro de control domótico está centrado en una Raspberry Pi con sus periféricos correspondientes, los tiempos de implementación del sistema propiamente tal no supera la hora de trabajo. La diferencia con otros proyectos, radica en que la mayor parte del proyecto corresponde a labores de programación y desarrollo. Los costos considerados para personal se incluyen en la tabla 3-4.

Tabla 3-4. Costos de personal

N° de Horas	Personal	Costo
1	Programador	1 UF

Fuente: Elaboración propia

### 3.5 COMPARACIÓN CON EQUIPOS CONVENCIONALES

Es necesario especificar que el equipo diseñado, reúne la función de dos sistemas. Por un lado el sistema correspondiente al parlante inteligente, y por otro lado, el sistema encargado del control de los dispositivos. El segundo sistema debe ser reconocido por el asistente incluido en el parlante inteligente (esta compatibilidad está determinada solamente por convenios entre las empresas). La tabla 3-5 indica el precio en el mercado chileno de los parlantes inteligentes. Se escogen específicamente parlantes que posean como asistente a Google Assistant o Alexa, debido a que son los que poseen mayor compatibilidad con los equipos disponibles en el mercado chileno.

Tabla 3-5. Precios de parlantes inteligentes.

N°	Nombre	Precio (CLP)	Referencia
1	Google Home Mini Carbon	\$37.990	<a href="https://www.sodimac.cl/sodimac-cl/product/3538230/Google-Home-Mini-Carbon/3538230">https://www.sodimac.cl/sodimac-cl/product/3538230/Google-Home-Mini-Carbon/3538230</a>
2	Google Home	\$95.990	<a href="https://www.sodimac.cl/sodimac-cl/product/3450457/Google-Home-Blanco/3450457">https://www.sodimac.cl/sodimac-cl/product/3450457/Google-Home-Blanco/3450457</a>
3	Alexa Echo Dot Segunda Generación	\$45.590	<a href="https://www.pcfactory.cl/producto/30019-asistente-de-voz-wifi-alexa-echo-dot-blanco-2da-generacion-">https://www.pcfactory.cl/producto/30019-asistente-de-voz-wifi-alexa-echo-dot-blanco-2da-generacion-</a>
4	Amazon Echo Segunda Generación	\$94.990	<a href="https://simple.ripley.cl/amazon-echo-2aa-generacion-altavoz-inteligente-con-alexa-tela-de-color-gris-oscuro-mpm00001929448?queryID=e500267386a9ad0c05b90507f407edc0&amp;objectID=MPM00001929448">https://simple.ripley.cl/amazon-echo-2aa-generacion-altavoz-inteligente-con-alexa-tela-de-color-gris-oscuro-mpm00001929448?queryID=e500267386a9ad0c05b90507f407edc0&amp;objectID=MPM00001929448</a>

Fuente: Elaboración propia.

Para el segundo sistema, se seleccionan equipos compatibles con los asistentes mencionados previamente. La tabla 3-6 indica las principales alternativas con sus costos asociados.

Tabla 3-6. Sistemas de control ofrecidos por el mercado.

N°	Nombre	Precio (CLP)	Referencia
1	WIFI Smart-Plug Vanzavanzu (x2) + 4 luces led	\$108.558	<a href="https://articulo.mercadolibre.cl/MLC-465802864-wifi-smart-plug-vanzavanzu-16a-toma-de-corriente-inteli-JM?matt_tool=73634802&amp;matt_word&amp;gclid=Cj0KCQiAxNnfBRDwARIsAJIH29DgnP5ztRFYBjTLOZtVaSVsMW1INvkkZL6EcBcAJUt6_CfM7yARtAaArd6EALw_wcB&amp;quantity=1">https://articulo.mercadolibre.cl/MLC-465802864-wifi-smart-plug-vanzavanzu-16a-toma-de-corriente-inteli-JM?matt_tool=73634802&amp;matt_word&amp;gclid=Cj0KCQiAxNnfBRDwARIsAJIH29DgnP5ztRFYBjTLOZtVaSVsMW1INvkkZL6EcBcAJUt6_CfM7yARtAaArd6EALw_wcB&amp;quantity=1</a>  <a href="https://www.sodimac.cl/sodimac-cl/product/3156079/Pack-de-ampolletas-LED-E-27-40-W-Calida-2-unidades">https://www.sodimac.cl/sodimac-cl/product/3156079/Pack-de-ampolletas-LED-E-27-40-W-Calida-2-unidades</a>

2	Philips Hue Bridge + Philips Hue Luz	\$150.980	<a href="https://www.paris.cl/store/producto/kit-3-ampolletas-led-started-new-hue-philips-213176-ppp-?utm_source=knasta.cl&amp;utm_medium=knasta&amp;utm_campaign=knasta&amp;utm_domain=knasta.cl">https://www.paris.cl/store/producto/kit-3-ampolletas-led-started-new-hue-philips-213176-ppp-?utm_source=knasta.cl&amp;utm_medium=knasta&amp;utm_campaign=knasta&amp;utm_domain=knasta.cl</a>  <a href="https://www.paris.cl/store/producto/ampolleta-led-new-hue-philips-213190-ppp-">https://www.paris.cl/store/producto/ampolleta-led-new-hue-philips-213190-ppp-</a>
---	--------------------------------------	-----------	--

Fuente: Elaboración propia

En el sistema compatible 2, el Equipo considera 3 ampolletas de Philips, por lo que es necesario agregar una extra, de manera que considere el control de 4 dispositivos. Al igual que con los parlantes inteligentes, se consideran solamente las opciones disponibles en el mercado chileno. Basados en las tablas anteriores, la combinación más económica que realiza la misma función que el proyecto se presenta en la tabla 3-7.

Tabla 3-7. Configuración comercial económica.

N°	Nombre	Precio (CLP)
1	Google Home Mini Carbon	\$37.990
2	WIFI Smart-Plug Vanzavanzu (x2) + 4 luces led	\$108.558
<b>Total</b>		<b>\$146.548</b>

Fuente: Elaboración propia.

### 3.6 CONEXIÓN DE INTERNET

Todo altavoz inteligente requiere de conexión permanente a internet para realizar sus funciones (consulta de información y envío de información). La conexión a internet debe tener un mínimo de velocidad de 5 MBPS. La tabla 3-8 indica los valores considerados para los propósitos de conexión.

Tabla 3-8. Conexión a internet

Nombre	Precio (CLP)
Internet mega 30 VTR (30 MB/s)	\$20.990

Fuente: Elaboración propia.

### 3.7 CONSUMOS DE ENERGÍA CONSIDERADOS

La Raspberry Pi, dado que cumple funciones de parlante inteligente, está conectada a internet de forma permanente al igual que la placa NodeMCU, que se encuentra en las mismas condiciones, conectadas las 24 horas del día. Los componentes considerados dentro de la potencia son los gastos implicados por la conexión del NodeMCU y la Raspberry Pi, mientras que por el equipo comercial se considera solamente el parlante Google Assistant, dado que el consumo de los dispositivos comandados es variable. En la tabla 3-9 se indican las potencias consideradas para los dos casos.

Tabla 3-9. Consumos de energía

Dispositivo	Consumo	Costo mensual (CLP)
Raspberry Pi + NodeMCU	0.002[kW]	\$161
Google Home Mini	0.009[kW]	\$726,2784

Fuente: Elaboración propia

Es importante destacar que, para el cálculo del costo mensual, se considera un mes de 30 días con 24 horas de funcionamiento del dispositivo. La tarifa considerada es BT1 con un precio de 112,08 pesos el kilowatt.

### 3.8 COMPARACIÓN ECONÓMICA

Finalmente, la comparación económica se realiza entre el centro de control doméstico desarrollado en el proyecto, y los precios de sistemas equivalentes en el mercado. La comparación se observa en la tabla 3-10.

Tabla 3-10. Comparación económica

	Raspberry Pi	Comercial
Costos Hardware	\$89.479	\$146.548
Costos Software	\$0	\$0
Costos de Personal	\$27.518	\$0
Internet	\$20.990	\$20.990
Electricidad	\$161	\$726
<b>Total (CLP)</b>	<b>\$138.788</b>	<b>\$168.264</b>

Fuente: Elaboración propia

El valor de la UF considerado para los costos es de \$27.518 correspondiente al 16 de noviembre del año 2018. La diferencia en los costos de personal radica en que los productos comerciales, no incluyen servicios de instalación o configuraciones mayores, dado que las configuraciones son realizadas por el usuario.

### 3.9 BENEFICIOS LOGRADOS

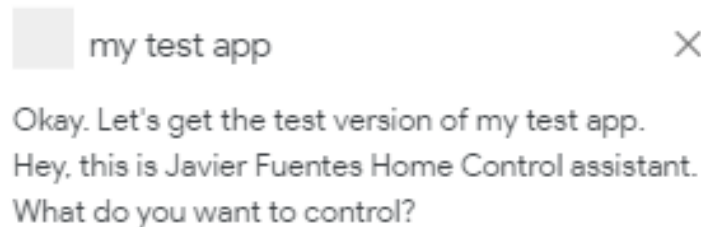
Los beneficios que se obtienen con la realización del proyecto abarcan las siguientes aristas:

#### 3.9.1 Internet de las cosas

Es inminente la interconexión de los objetos cotidianos con internet u otra red. Eso presupone ventajas tales como constante monitoreo de los dispositivos conectados a la red, ahorro energéticos o simplemente comodidad. Otro de los factores a considerar es que el proyecto al ser un sistema abierto, es modificable para lograr control sobre mayor cantidad de dispositivos.

### 3.9.2 Opciones de ingreso de información por voz o texto

Dialogflow, además de permitir generar una interfaz de voz para el control de dispositivos, simultáneamente genera una interfaz visual que permite el ingreso de órdenes mediante texto, la cual se despliega en la aplicación de Google Assistant, que puede estar instalada de forma optativa en cualquier dispositivo con S.O Android, esto permitiría el control para personas con problemas de movilidad, visión u otros. La figura 3-1 muestra la interfaz generada.



Fuente: Elaboración propia

Figura 3-1. Despliegue de información en Google Assistant

### 3.9.3 Control a distancia

Tal como fue explicado en el punto anterior, el hecho que la aplicación creada en Dialogflow, permite acceder a ella desde cualquier dispositivo Android (que tenga asociada la misma cuenta de Google registrada en Dialogflow), permite controlar los dispositivos conectados al parlante no solamente dando órdenes directamente a este, sino que también mediante la aplicación del asistente de Google.

### **3.10 MEJORAS AL SISTEMA DESARROLLADO**

Una de las ventajas que los asistentes se encuentren permanentemente conectados a internet, es que las actualizaciones que se realizan al asistente en sus servidores, son realizadas en el usuario final en forma instantánea. Entre las mejoras en el sistema en los próximos meses se cuentan:

#### **3.10.1 Incorporación de nuevos idiomas en Google Assistant**

El 30 de agosto del año 2018, Google incorporó al asistente una tecnología llamada LangID, la cual permite al asistente realizar un reconocimiento de dos idiomas en forma simultánea en el asistente de Google. El reconocimiento dual reconoce por el momento el español, inglés, alemán, francés, italiano y japonés, aunque Google agregará soporte para otros idiomas dentro de los próximos meses.

#### **3.10.2 Incorporación de nuevos idiomas en Dialogflow**

El programa utilizado para la creación de las interfaces conversacionales (Dialogflow) permite un trabajo fluido con el idioma inglés, no así con los demás. Actualmente los modos de desarrollo en lenguaje español se están implementando junto con otros idiomas, lo que permitiría tener el programa de control de dispositivos en lenguaje español.

#### **3.10.3 Mejoras en entradas y salidas de audio.**

El propósito de los micrófonos y parlantes de pequeño tamaño, es la facilidad de portabilidad del centro de control domótico. En ciertas ocasiones se pueden necesitar una salida de audio más potente, por lo cual es recomendable cambiar el parlante conectado a la Raspberry Pi. Lo mismo es aplicable para el micrófono.

#### **3.10.4 Reconocimiento de usuarios.**

El asistente de Google posee un sistema llamado Voice Match, el cual permite que el parlante inteligente sea capaz de reconocer a un número máximo de 6 usuarios distintos, de manera que se pueda generar contenido personalizado para esos 6 usuarios. La limitación que

existe respecto a este sistema (además del número de usuarios) es que Voice Match necesita de una aplicación externa (Google Home) para poder ser utilizada.

#### 3.10.5 Canales de salida

Para el proyecto propuesto se consideran 4 canales de salida, los cuales son ampliables a los canales que sean necesarios. En el proyecto actual, solamente se consideran el encendido y/o apagado de estas cuatro salidas, pero es posible incorporar salidas análogas.

## **CONCLUSIONES Y RECOMENDACIONES**

Los modernos avances en domótica, inteligencia artificial e internet de las cosas han surgido tanto en base a la necesidad de tener un mayor control sobre nuestros recursos, así como en función de aumentar la comodidad en nuestros hogares.

Los parlantes inteligentes son dispositivos que permiten por medio de los avances mencionados anteriormente, ejercer un control tanto sobre dispositivos actuales, como sobre otros que tengan compatibilidad con el asistente empleado. Actualmente no se encuentran tan masificados dentro del mercado chileno, debido a lo anterior, alternativas basadas en placas de desarrollo tales como la Raspberry Pi, suponen una ventaja debido a que permiten mayor personalización que los sistemas disponibles en el mercado, así como menores costos de adquisición.

La capacidad de conexión a internet de los distintos dispositivos que existen (internet de las cosas), permiten mejorar tanto la eficiencia energética como tener información a tiempo real sobre diferentes variables de interés.

Por otro lado, el avance en la tecnología relacionada con interfaces conversacionales, permite acceder a la información dejando de lado los mecanismos tradicionales de ingreso y salida de datos (teclados y pantallas).

Con respecto al cumplimiento de los objetivos, estos son logrados, aunque de forma estrecha, esto, debido a los precios cada vez más económicos de dispositivos de índole comercial y la competencia económica entre los países que desarrollan este tipo de tecnología tales como China y Estados Unidos.

Debido a las razones previamente mencionadas, se hace indispensable tener conocimiento sobre estas tecnologías, ya que no solamente es utilizada en el ambiente hogareño, sino que constituyen el motor de la denominada Industria 4.0.

**BIBLIOGRAFÍA**

DIALOGFLOW. Dialogflow [en línea]. 2018 [consulta 5 de Septiembre de 2018]. Disponible en: <<https://console.dialogflow.com/api-client/#/login>>

ERICSSON. Ericsson [en línea].2015. [consulta 20 de Agosto de 2018].Disponible en: <<https://www.ericsson.com/assets/local/news/2015/6/ericsson-consumerlab-connected-homes.pdf>>

EUPHROSINE, Johan. Github [en línea]. 2017 [consulta 20 de Octubre de 2018]. Disponible en: <<https://media.readthedocs.org/pdf/firebase-arduino/latest/firebase-arduino.pdf>>

GARTNER. Gartner [en línea] 2017. [consulta 22 de Agosto de 2018]. Disponible en: <[https://www.gartner.com/imagesrv/books/iot/iotEbook\\_digital.pdf](https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf)>

GOOGLE. Actions on Google [en línea]. 2018 [consulta 15 de Junio de 2018]. Disponible en: <<https://developers.google.com/actions/>>

GOOGLE. Firebase [en línea]. 2018 [consulta 3 de Octubre de 2018]. Disponible en: <<https://firebase.google.com/?hl=es-419>>

GOOGLE. Google Cloud [en línea]. 2018 [consulta 10 de Junio de 2018]. Disponible en: <<https://console.cloud.google.com>>

RASHMIN, Ravindu. Medium [en línea]. 2018 [consulta 28 de Octubre de 2018]. Disponible en: <<https://medium.com/coinmonks/arduino-to-android-real-time-communication-for-iot-with-firebase-60df579f962>>

**ANEXO A: CÓDIGO DE DIALOGFLOW WEBHOOK**

```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 index.js
const functions = require('firebase-functions');
const admin = require('firebase-admin');
admin.initializeApp();
const {dialogflow} = require('actions-on-google');
exports.receiveAssistantRequests = functions.https.onRequest((request, response) => {
  const app = new dialogflow({request: request, response: response});
  function handlerRequest(app) {
    const device = app.getArgument('devices');
    const status = app.getArgument('status');
    return admin.database().ref(`/automation/${device}/value`).set(status)
      .then(snapshot => {
        app.ask(`Ok, switching ${device} ${status}. Do you want to control anything else?`);
        response.send(200);
        return null;
      });
  }
  app.handleRequest(handlerRequest);
});
$
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición M-U Deshacer
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar txt ^I Ortografía ^ Ir a línea M-E Rehacer
```

**ANEXO B: CÓDIGO DE NODEMCU**

```

#include <FirebaseArduino.h> // Agrega la Librería de Firebase
#include <ESP8266WiFi.h> // Agrega la Librería del ESP8266

// Set these to run example.
#define WIFI_SSID "evalme_living" // Nombre de la red
#define WIFI_PASSWORD "16726509k" // Password de la red
#define FIREBASE_DB_URL "automation-17f54.firebaseio.com" // Dirección URL de Firebase
#define FIREBASE_DB_SECRET_KEY "I9UzF2c1HgY129pT47N7jNhX2ZjmAscTQ6n3pc2I" // Clave de acceso a la DB

/* El proyecto de Firebase después de ser creado, crea automáticamente llaves de acceso. La FIREBASE_DB_URL
se encuentra en la dirección del proyecto creado en la pestaña DESARROLLA -> DATABASE. Por otro lado, la
FIREBASE_DB_SECRET_KEY se encuentra en: Project Overview -> Configuración del proyecto -> Cuentas de
servicio -> Secretos de la base de datos y se encuentra encriptada.
*/

void setup() { // Función de configuración
  Serial.begin(115200); // Comunicación serial a 115200

  pinMode(D1, OUTPUT); // D1 se selecciona como salida
  pinMode(D2, OUTPUT); // D2 se selecciona como salida
  pinMode(D3, OUTPUT); // D3 se selecciona como salida
  pinMode(D4, OUTPUT); // D4 se selecciona como salida

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD); // Se inicia WIFI
  while (WiFi.status() != WL_CONNECTED) { // Retardo de 500 ms
    delay(500);
  }

  Firebase.begin(FIREBASE_DB_URL, FIREBASE_DB_SECRET_KEY); // Inicia comunicaciones con las DB
  Firebase.stream("/automation-17f54"); // Transmite cambios a los nodos del path
}

void loop() { // Función principal
  if (Firebase.failed()) { // Si Firebase falla
    Serial.println("streaming error"); // Muestra error de conexión
    Serial.println(Firebase.error());
  }

  /* Un dato a considerar, es que si la dirección termina en un "/", es necesario eliminarlo, ya que era necesario en
versiones previas del equipo*/

  if (Firebase.available()) { // Si Firebase está disponible...
    Serial.println("Firebase disponible"); // Indica disponibilidad de Firebase
    String valor_1= Firebase.getString("/iotday/fan/value"); // Extracción de valor dispositivo_1
    String valor_2= Firebase.getString("/iotday/light/value"); // Extracción de valor dispositivo_2
    String valor_3= Firebase.getString("/iotday/heater/value"); // Extracción de valor dispositivo_3
    String valor_4= Firebase.getString("/iotday/lamp/value"); // Extracción de valor dispositivo_3
  }

  /* Es posible agregar todos los periféricos que se estimen convenientes, la ruta inicial es
* "iotday/nombre_del_dispositivo/value". Para propósitos del proyecto propuesto, se consideran 4 periféricos de
* salidas. Los cuales también son archivos de dato String, que deben ser considerados en el programa
* correspondiente en dialogflow:
*/
}

```

```

Serial.println(valor_1);
Serial.println(valor_2);
Serial.println(valor_3);
Serial.println(valor_4);

if(valor_1.equals("on")){
  digitalWrite(D1,HIGH);
}
if(valor_1.equals("off")){
  digitalWrite(D1,LOW);
}

if(valor_2.equals("on")){
  digitalWrite(D2,HIGH);
}
if(valor_2.equals("off")){
  digitalWrite(D2,LOW);
}

if(valor_3.equals("on")){
  digitalWrite(D3,HIGH);
}
if(valor_3.equals("off")){
  digitalWrite(D3,LOW);
}

if(valor_4.equals("on")){
  digitalWrite(D4,HIGH);
}
if(valor_4.equals("off")){
  digitalWrite(D4,LOW);
}
}
}

// Verificar lectura de dispositivo_1
// Verificar lectura de dispositivo_2
// Verificar lectura de dispositivo_3
// Verificar lectura de dispositivo_4

// Si el valor de value de fan es on...
// Se enciende FAN

// En caso contrario...
// Se apaga FAN

// Si el valor de value de light es on...
// Se enciende LIGHT

// En caso contrario...
// Se apaga LIGHT

// Si el valor de value de dispositivo_3 es on...
// Se enciende dispositivo_3

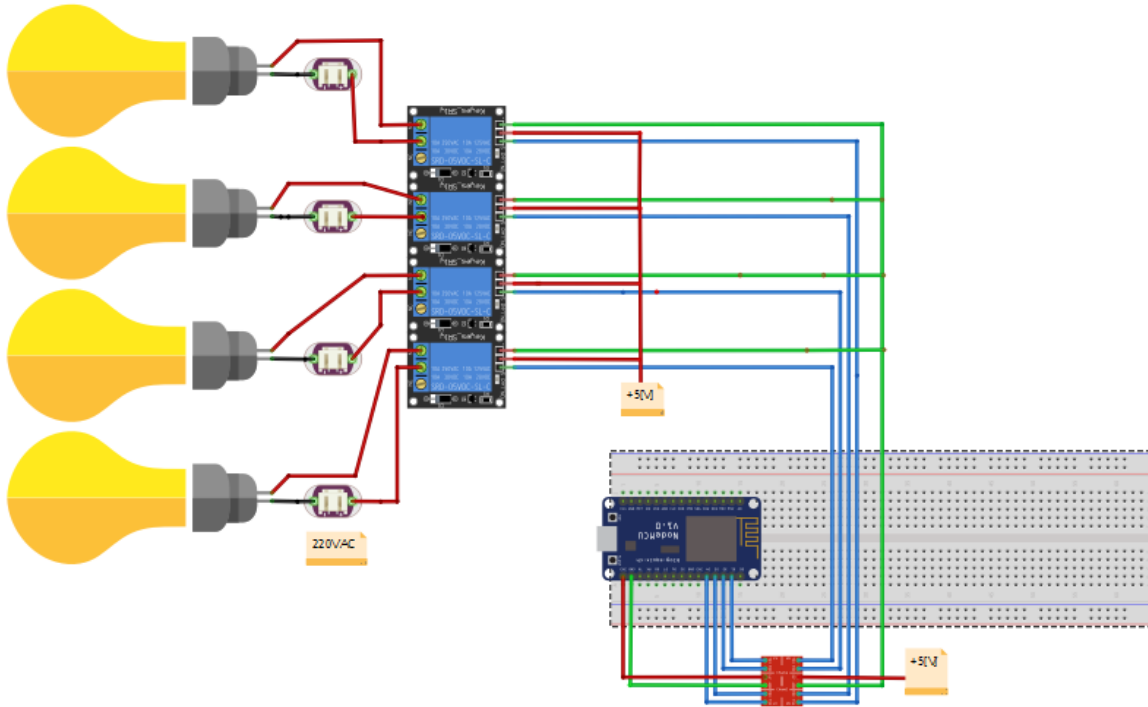
// En caso contrario...
// Se apaga dispositivo_3

// Si el valor de value de dispositivo_4 es on...
// Se enciende dispositivo_4

// En caso contrario...
// Se apaga dispositivo_4

```

**ANEXO C: DIAGRAMA GENERAL DE LA SOLUCIÓN**



El diagrama muestra las salidas del NodeMCU conectados a un Logic Level Converter de Sparkfun (Convertor lógico de nivel). Los relés de la placa por defecto son de 5[V], debido a lo anterior, es mediante el Logic Level Converter, que los voltajes de 3,3[V] pasan a ser 5[V], los cuales permiten la conmutación de los relés.