



**CONSTRUCCIÓN DE UN PROTOTIPO  
DE ROBOT SOCIAL BASADO EN UN  
MINIORDENADOR RASPBERRY PI 3  
Y ARDUINO**

**Autores:**  
González Miguel  
Infante Melanie

Urb. Yuma II, calle N° 3. Municipio San Diego  
Teléfono: (0241) 8714240 (máster) – Fax: (0241) 8712394



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELECTRÓNICA

**CONSTRUCCIÓN DE UN PROTOTIPO DE ROBOT SOCIAL BASADO EN  
UN MINIORDENADOR RASPBERRY PI 3 Y ARDUINO**

Trabajo de grado presentado como requisito para optar al título de

**INGENIERO ELECTRÓNICO**

**Autores:** González Miguel

C.I: V-21.277.418

Infante Melanie

C.I: V-24.496.171

**Tutor:** Ing. Dinorah Giménez

CI.: V-4.071.295

San Diego, junio de 2017



Universidad José Antonio Páez  
Facultad de Ingeniería

FI-TG-2017-1CR-062

Valencia, 13 de Enero de 2017.

Ciudadanos:

**Melanie Infante**

**C.I. 24.496.171**

**Miguel González**

**C.I. 21.277.418**

Presente.-

Cumplo con informarle que la Comisión de Trabajo de Grado y Pasantías de la Facultad de Ingeniería en su reunión N° 1-2017 de fecha 13/01/2017 aprobó el proyecto de trabajo de grado titulado **“CONSTRUCCIÓN DE UN PROTOTIPO DE ROBOT SOCIAL BASADO EN UN MINI ORDENADOR RASBERRY PI 3 Y ARDUINO”** Presentado por ustedes como requisito para optar al título de Ingeniero electrónico.

Se ratifica la designación de la Ing. Dinorah Gimenez C.I. 4.071.295 y la Ing. Alicia Pizzella, C.I. 4.598.880 como Tutotes Académicos que lo asesorarán en el desarrollo de este proyecto.

Atentamente,

Prof. Marlene Zambrano  
Decana (Encargada) de la Facultad de Ingeniería  
(CU502 de fecha 11/10/2016)



c. c. Coordinación de Pasantías y Trabajo de Grado (2).  
Archivo.

MEZ/jp



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELECTRÓNICA

**APROBACIÓN DEL TUTOR**

Quien suscribe, Ingeniero DINORAH GIMÉNEZ portadora de la cédula de identidad N° V-4.071.295, en mi carácter de tutor del trabajo de grado presentado por los ciudadanos, GONZÁLEZ MENENDEZ MIGUEL ANGEL portador de la cédula de identidad N° V-21.277.418 e INFANTE ROMERO MELANIE DENISSE portador de la cédula de identidad N° V-24.496171, titulado **CONSTRUCCIÓN DE UN PROTOTIPO DE ROBOT SOCIAL BASADO EN UN MINI ORDENADOR RASPBERRY PI 3 Y ARDUINO**, presentado como requisito parcial para optar al título de INGENIERO ELECTRONICO, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los ocho días del mes de Junio del año dos mil diecisiete.

Ing. Dinorah Giménez  
C.I.: V-4.071.295





REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELECTRÓNICA

## AGRADECIMIENTOS

Agradezco a Dios por permitirme vivir esta experiencia de la cual me llevo mucho aprendizaje, disciplina y amistades; por darme la fuerza y la perseverancia de seguir adelante día a día y por poner en mi camino a personas maravillosas que de una u otra forma contribuyeron a mi formación a lo largo de la carrera.

Quiero dar gracias a mi mamá Editza Romero que, aunque no sabía nada de lo que le hablaba, siempre estaba ahí sentada a mi lado con una palabra de aliento, con un tecito para calmarme, con un abrazo para consolarme o con un plato de comida, eres la mejor mamá del mundo.

A mi Papá, William por siempre estar dispuesto a ayudarme en lo poco o en lo mucho, siempre utiliza lo que tiene a su alcance o se las inventa para resolverme los problemas y cumplir con lo que necesito en el momento.

A mi hermana, que a pesar de que discutamos siempre me presta su apoyo en cualquier situación que la necesito.

A mi super compañero de tesis quiero agradecerle por el inmenso apoyo que significa para mí, no solo en la realización de este trabajo de grado, sino a lo largo de toda mi carrera universitaria, realmente fue una bendición.

Se le agradecen muchas cosas a Maritza que se vino días antes para ayudarnos con la maqueta del proyecto, nos tuvo paciencia, escucho nuestras preocupaciones, nos apoyó y nos prestó ayuda en todo.

También agradezco a mis compañeros de estudio por preocuparse y por ponerse a la orden siempre que los necesite.

Melanie Infante



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELECTRÓNICA

### **AGRADECIMIENTOS**

A Dios, mi madre, mi padre, mis hermanos, mi novia, mi suegra y suegro,  
amigos y compañeros de estudio.

Miguel González



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELECTRÓNICA

**DEDICATORIA**

Quiero dedicarle este trabajo de grado:

A Dios, por permitirme vivir esta experiencia y darme la inteligencia, la sabiduría y las herramientas para lograr esta meta.

A mi mama Editza, por darme la vida, apoyarme en todos mis proyectos y ser mi ejemplo a seguir. Te amo mami.

A mi Papá, William por siempre estar dispuesto a ayudarme en todo lo que en el momento necesito.

A mi hermana Melissa, porque quiero ser ejemplo de un “si se puede”, y que ella puede lograr todo lo que se proponga en la vida.

A mi querido Abuelo Armando, por ser el máximo ejemplo de vida que se pueda tener, ejemplo de perseverancia, entrega, positivismo, animo, AMOR. Eres una de mis mayores inspiraciones.

A mi compañero de Tesis, amigo y novio Miguel, por ser a lo largo de este tiempo mi maestro y mi guía con sus ideas alocadas y soñadoras.

Melanie Infante



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELECTRÓNICA

### **DEDICATORIA**

Quiero dedicar este trabajo de grado A Dios, mi madre, mi padre, mis hermanos, mi novia, mi suegra y suegro, amigos, compañeros de estudio y a todas las mentes soñadoras.

Miguel González

## ÍNDICE GENERAL

<b>CONTENIDO</b>	<b>Pp.</b>
<b>ÍNDICE DE FIGURAS</b> .....	xi
<b>ÍNDICE DE GRÁFICOS</b> .....	xiii
<b>RESUMEN</b> .....	xiv
<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO</b>	
<b>I. EL PROBLEMA</b>	
1.1. Planteamiento del Problema.....	3
1.2. Formulación del Problema.....	5
1.3. Objetivos de la investigación.....	5
1.3.1. Objetivo General.....	5
1.3.2. Objetivos Específicos.....	6
1.4. Justificación del Problema.....	6
1.5. Alcance.....	6
<b>II. MARCO TEÓRICO</b>	
2.1 Antecedentes.....	8
2.2 Bases Teóricas.....	9
2.2.1 Robot Social.....	9
2.2.2 Dispositivos de Control.....	10
2.2.2.1 Raspberry Pi.....	10
2.2.2.2 Arduino.....	10
2.2.3 Domótica.....	11
2.2.4 Lenguaje de Programación.....	11
2.2.4.1 Lenguaje C++.....	11
2.2.4.2 Python.....	12
2.2.4.3 JavaScript.....	12
2.2.4.4 Nodejs.....	12
2.3 Definición de términos básicos.....	13
<b>III. MARCO METODOLÓGICO</b>	
3.1 Tipo de Investigación.....	14
3.2 Diseño de la Investigación.....	14

<b>CONTENIDO</b>	<b>Pp.</b>
3.3 Nivel de la Investigación.....	15
3.4 Población y Muestra.....	15
3.5 Técnicas e Instrumentos de investigación.....	16
3.5.1 Técnicas.....	16
3.5.1.1 Observación Directa.....	16
3.5.1.2 Encuesta.....	17
3.5.2 Instrumento.....	17
3.5.2.1 Cuestionario.....	17
3.6 Fases Metodológicas.....	17
<b>IV. RESULTADOS</b>	
4.1. Fase I. Examinar las aplicaciones de robótica y domótica existentes en Venezuela.....	20
4.2. Fase II. Identificar potenciales funcionalidades que podría tener el Robot Social a desarrollar.....	22
4.3. Fase III. Seleccionar e instalar los softwares adecuados para la elaboración del Robot Social.....	26
4.3.1. Instalación de los softwares secundarios del Raspberry Pi.....	28
4.3.1.1. SSH en el Raspberry Pi.....	28
4.3.1.2. VNC en el Raspberry Pi.....	29
4.3.2. Alexa.....	30
4.3.3. Node-red.....	34
4.3.4. Flask y Flask-ask.....	37
4.3.5. Ngrok.....	38
4.3.6. IDE Arduino.....	38
4.4. Fase IV. Desarrollar el prototipo del Robot Social.....	39
4.4.1. Programación de Skill para el envío de una señal eléctrica por petición del hablada del usuario.....	40
4.4.2. Programación para la Skill de interacción “Presentation”.....	51
4.4.3. Programación para la Skill de seguridad.....	58
4.4.4. Programación de Skill para el manejo de servomotores por comando de voz.....	62
<b>CONCLUSIONES</b> .....	67
<b>RECOMENDACIONES</b> .....	68
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	69

<b>CONTENIDO</b>	<b>Pp.</b>
<b>REFERENCIAS ELECTRÓNICAS</b> .....	69
<b>APENDICE</b>	
<b>A Encuesta</b> .....	73

## ÍNDICE DE FIGURAS

<b>FIGURA</b>	<b>CONTENIDO</b>	<b>Pp.</b>
1	Ejecución del cliente PuTTY.....	28
2	Ventana terminal del Raspberry Pi para PuTTY. ....	29
3	Desarrollador de Amazon. ....	31
4	Device Detail. ....	32
5	Ventana Terminal del Raspberry Pi para Alexa I.....	33
6	Ventana Terminal del Raspberry Pi para Alexa II.....	33
7	Ventana Terminal del Raspberry Pi para Alexa III.....	34
8	Ventana Terminal del Raspberry Pi para Alexa IV.....	34
9	Editor de flujos de Node-red.....	35
10	Consola Node-red.....	36
11	Ngrok ejecutado en el Raspberry Pi.....	38
12	Programación en el Entorno Arduino de Skill de envío de señales eléctricas.....	41
13	Programación en Python de Skill de envío de señales eléctricas.....	42
14	Ejecución en Python para la Skill de envío de señales eléctricas.....	43
15	Ejecución de Ngrok para la Skill de envío de señales eléctricas.....	43
16	Menú de Amazon Developer.....	43
17	Listado de Skills del usuario.....	44
18	Skill information para la Skill de envío de señales eléctricas.....	44
19	Intent Schema para la Skill de envío de señales eléctricas.....	45
20	Utterances para la Skill de envío de señales eléctricas.....	46

<b>FIGURA</b>	<b>CONTENIDO</b>	<b>Pp.</b>
21	Ventana “Configuration” para la Skill de envío de señales eléctricas.....	47
22	Ventana “SSL Certificate” para la Skill de envío de señales eléctricas.....	47
23	Ventana “Test” para la Skill de envío de señales eléctricas.....	48
24	Service Simulator para la Skill de envío de señales eléctricas.....	48
25	Service Simulator para la Skill de envío de señales eléctricas II.....	48
26	Respuesta en la ventana terminal de Ngrok para la Skill de envío de señales eléctricas.....	49
27	Diagrama de Comunicación para la Skill de envío de señales eléctricas.....	50
28	Diagrama de Conexión para la Skill de envío de señales eléctricas.....	50
29	Diagrama Esquemático para la Skill de envío de señales eléctricas.....	51
30	Nodo https para la Skill Presentation.....	52
31	Nodo Switch para la Skill Presentation I.....	53
32	Nodo Switch para la Skill Presentation II.....	53
33	Nodo Template para la Skill Presentation I.....	54
34	Nodo Template para la Skill Presentation II.....	54
35	Flujo en Node-red para la Skill Presentation.....	55
36	Skill Information para la Skill Presentation.....	56
37	Intent Schema para la Skill Presentation.....	56
38	Ventana “Utterances” para la Skill Presentation.....	56
39	Ventana “Configuration” para la Skill Presentation.....	57
40	Diagrama de Comunicación para la Skill Presentation.....	57
41	Código Arduino para el sensor de movimiento.....	58
42	Diagrama de flujo en Node-red para el sensor de movimiento.....	59
43	Configuración del nodo “Twilio” para el sensor de movimiento.....	59
44	Mensajes de texto indicando cambios de estado del sensor.....	60
45	Correo indicando cambios de estado del sensor.....	60
46	Diagrama de Comunicación para el sensor de movimiento.....	61

<b>FIGURA</b>	<b>CONTENIDO</b>	<b>Pp.</b>
47	Diagrama de Conexión para el sensor de movimiento.....	61
48	Diagrama Esquemático para el sensor de movimiento.....	62
49	Código IDE Arduino de los Servomotores.....	63
50	Código Python de los Servomotores.....	64
51	Intent Schema de los Servomotores.....	65
52	Utterances de los Servomotores.....	65
53	Diagrama de Conexión de los Servomotores.....	66
54	Diagrama Esquemático de los Servomotores.....	66

### ÍNDICE DE GRÁFICOS

<b>GRÁFICO</b>	<b>CONTENIDO</b>	<b>Pp.</b>
1	Resultado Ítem 1.....	23
2	Resultado Ítem 2.....	23
3	Resultado Ítem 3.....	24
4	Resultado Ítem 4.....	24
5	Resultado Ítem 5.....	24
6	Resultado Ítem 6.....	25
7	Resultado Ítem 7.....	25
8	Resultado Ítem 8.....	25
9	Resultado Ítem 9.....	26



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERIA ELECTRONICA

## CONSTRUCCIÓN DE UN PROTOTIPO DE ROBOT SOCIAL BASADO EN UN MINIORDENADOR RASPBERRY PI 3 Y ARDUINO

**Autores:** González Miguel, Infante Melanie

**Tutor:** Ing. Dinorah Giménez

**Fecha:** Enero, 2017

### RESUMEN

El trabajo de investigación tiene como objetivo general la construcción de un prototipo de robot social basado en un miniordenador Raspberry pi 3 y Arduino. Dicho prototipo interactuara con el ser humano dando demostración de inteligencia social, de esta manera el robot entrara en el marco de clasificación de robot social. Esta interacción se realizará mediante reconocimiento de voz, siendo esta la ventana necesaria para la interacción máquina-hombre. Dicho prototipo tendrá el fin de facilitar las labores cotidianas y repetitivas del usuario, siendo capaz de tomar decisiones sencillas como encender una bombilla mediante comandos de voz hasta hacer un recordatorio de las actividades planificadas en la agenda del usuario.

**Descriptor:** Herramientas de programación, Robótica, Microcontroladores

## INTRODUCCIÓN

El ser humano lleva muchos siglos con el sueño de la creación de máquinas autónomas y obedientes, que sean capaces de llevar a cabo los trabajos más difíciles y exigentes. Para esto fueron diseñados los robots, palabra proveniente de “robotnik” o “robota”, que se podría traducir como “trabajo tedioso” en checo y fue empleada por Karel Capek en su obra de teatro “Opilek” pero siendo Isaac Asimov quien popularizó dicho termino en sus escritos de ciencia ficción. También existe una variedad inmensa de tipos diferentes de robots, por lo que se han ido clasificando con el pasar del tiempo, habiendo robots industriales, robots del tipo médico, pero existe una nueva clasificación que aún está en pleno desarrollo siendo estos los robots sociales. La construcción de los llamados robots sociales es un campo emergente que abarca muchas disciplinas, como la robótica y la inteligencia artificial, esta progresivamente atrayendo la atención de todo el mundo.

Los robots sociales fueron diseñados con la tarea única de hacer la vida cotidiana de las personas mucho más sencillas, en este tipo de robots, la comunicación entre el hombre y la maquina es el aspecto básico y fundamental, ya sea utilizando programas avanzados de reconocimiento de voz o facial que pueden distinguir la situación sentimental del individuo y conforme a esto tomar decisiones para ayudar a la persona que se comunica con el robot, hasta pantallas táctiles donde la persona debe seleccionar la tarea que desea que el robot le facilite. Esta clase de robot tiene algunas funciones, como el entretenimiento, la educación, la información (navegador web, servicios web, entre otros), la organización (la agenda, contactos, tareas, notas...) compañía, comunicación, vigilancia, control domótico entre otras.

En el primer capítulo, el problema, engloba el planteamiento del problema y de los objetivos, la justificación del trabajo realizado y el alcance del mismo. En el segundo capítulo, marco teórico, se pretende contextualizar la investigación, a través

de la exposición de los factores que delimitan el tema en cuestión. En el tercer capítulo, marco metodológico, se explica desde el punto de vista teórico la metodología de trabajo empleada para el desarrollo de la investigación en cuanto a actividades, productos esperados por cada etapa y técnicas empleadas. En el cuarto capítulo, resultados, se describen las actividades llevadas a cabo para alcanzar los objetivos y cumplir con ellos.

# CAPÍTULO I

## EL PROBLEMA

### 1.1 Planteamiento del Problema

Para lograr entender de manera más precisa el tema que se desarrolla en el presente trabajo, se debe conocer la definición de la palabra robot, el cual es una máquina programable capaz de realizar determinadas operaciones de manera autónoma y sustituir a los seres humanos en dichas tareas, en especial aquellas en las cuales el individuo exponga su salud, bienestar y seguridad, puede estar dotado de sensores que le permiten adaptarse a la situación que le rodea.

Por siglos, el ser humano se ha encontrado en la constante elaboración de máquinas que imitan las partes del cuerpo humano. Los antiguos egipcios unieron brazos elaborados mecánicamente a las estatuas de sus dioses, de igual manera los griegos elaboraron sistemas hidráulicos para las estatuas de sus dioses, los cuales se utilizaban más que nada con el fin de fascinar a aquellos que adoraban a dichos dioses. El inicio de la robótica actual se puede fijar en el año 1801, cuando el Francés Joseph Jacquard, trabajador de la industria textil del siglo XVIII inventa una máquina textil programable mediante el uso de tarjetas perforadas y en 1805 cuando el suizo Henri Maillardert construye una muñeca mecánica capaz de realizar dibujos.

En el siglo XIX se acuña la palabra “robot” por primera vez, específicamente en el año 1920, cuando el dramaturgo checo Karel Capek, la escribe en una obra llamada “los Robots universales de Rossums”, pero no es sino hasta el año 1939 cuando el Ruso Isaac Asimov comienza con su contribución a la robótica, es a este último a quien se le atribuye el origen del término “Robótica” y con esto surgen las denominadas “Tres Leyes de la Robótica”:

1.- Un robot no puede actuar contra un ser humano o, mediante la inacción, que un ser humano sufra daños.

2.- Un robot debe obedecer las órdenes dadas por los seres humanos, salvo que esta orden entre en conflicto con la primera ley.

3.- Un robot debe proteger su propia existencia, a no ser que este en conflicto con las dos primeras leyes.

Ya con una industria bastante avanzada, en el año de 1908, Henry Ford implementa la automatización de su propia línea de producción de vehículos, con este hecho, comienza a fortalecerse la idea de crear máquinas autónomas, pero no es hasta el año 1932 cuando aparece el robot “Lilliput” un robot creado por una empresa con sede en Japón, el cual tenía la facultad de caminar, obviamente, dicho robot no fue elaborado con un fin industrial, sino más bien estaba hecho para recrear a los humanos que le rodeasen.

En la actualidad, el rápido desarrollo de las tecnologías ha ayudado a los avances en la robótica, y gracias a esto aparece un nuevo término o una nueva aplicación, que son los robots domésticos o robots sociales, los cuales se han nutrido de tal forma con estos avances que han logrado mantener una comunicación casi directa con los humanos, gracias a que pueden recibir órdenes a través de comandos de voz y también al hecho de que poco a poco han adquirido apariencias muchísimo más amigables para el ser humano, hasta el punto que son capaces de simular las expresiones de los mismos y de dar respuestas más elaboradas.

En América del Sur, el tema de la robótica social se encuentra apenas comenzando, ya que en estas regiones hay gran variedad de robots de índole industrial, dejando a un lado a los robots sociales, para la fecha apenas se han hecho populares los robots de tipo aspiradora, mientras que, en países como Estados Unidos, Corea, Japón, es un tema fundamental de desarrollo.

Ahora bien, países como Chile y México ya han mostrado interés en dar pasos en estos temas de investigación, puesto que en el año 2015 Chile organiza una feria de robótica llamada “Robotics Day” la cual fue catalogada como la mayor feria de

robótica en Latinoamérica, en dicha feria estuvieron presentes robots de índole educativo, asistencial, socorrista, entre otros.

En México ya hay empresas encargadas de llevar kits de aprendizaje a las escuelas, para que niños desde el nivel del kínder hasta el nivel universitario mantengan contacto con este tipo de tecnología. La empresa “Róbo-Ed” es una de las más fuertes del mercado mexicano y para ellos la robótica Educativa se define como “el conjunto de actividades pedagógicas que apoyan y fortalecen en áreas específicas del conocimiento y desarrollan habilidades y competencias en el alumno, a través de proceso de concepción, creación, ensamble y puesta en funcionamiento de robots.” (Róbo-Ed S.A de C.V, 2008).

En Venezuela, el tema de la robótica doméstica se encuentra estancado, solo algunas universidades han intentado promover este tema, siendo la Universidad Católica Andrés Bello la que más resalta en el área, dando inicio desde el año 2004 a un programa llamado “La robótica va a la escuela”. A pesar de que, en nuestro país, la robótica se ve como un tema bastante inalcanzable, en este proyecto se desea desarrollar un robot, el cual cumpla con algunos requisitos mínimos para quedar dentro de la categoría de “Robot Social”.

## **1.2 Formulación del Problema**

Para dar cumplimiento a las expectativas planteadas en el párrafo anterior los investigadores se preguntan:

¿Cómo llevar a cabo la construcción de un prototipo de Robot Social?

## **1.3 Objetivos de la Investigación**

### **1.3.1 Objetivo General**

Construir un prototipo de Robot Social basado en un miniordenador Raspberry Pi 3 y Arduino.

### **1.3.2 Objetivos Específicos**

- Examinar aplicaciones de robótica y domótica existentes en Venezuela.
- Identificar potenciales funcionalidades que podría tener el Robot Social a desarrollar.
- Determinar los softwares adecuados para la elaboración del Robot Social.
- Desarrollar el prototipo del Robot Social.

### **1.4 Justificación del Problema**

En la actualidad, se considera necesario explorar e investigar los extensos campos de aplicación de la Robótica, debido a que esta es un área que va en pro del desarrollo en el campo de la medicina, de la industrial, en lo doméstico, entre otros. En este estudio se implementan conocimientos, obtenidos en la carrera de ingeniería electrónica, unido con tecnologías de vanguardia, para así dar a conocer a la comunidad Ujapista y a Venezuela alguna de las aplicaciones de esta área como lo es la Robótica, la cual es capaz de unirse con otras ciencias como la informática y la mecatrónica para dar como resultado muchos avances tecnológicos que pueden traducirse en desarrollo de las comunidades, poblaciones y hasta países.

Por otro lado, también se busca suministrar información sobre esta área a los estudiantes de electrónica, ya que se considera importante el hecho de que no se queden solamente con la información que se les da en el aula de clase, sino que estén actualizados con las nuevas tecnologías y sus aplicaciones, así como también que estén en conocimiento que la electrónica es bastante versátil, flexible y que está en constante avance.

### **1.5 Alcance**

En este trabajo de investigación se construirá un prototipo de un Robot Social con el miniordenador Raspberry Pi, con la finalidad de lograr emular algunas características de robots sociales ya elaborados por grandes empresas como Google,

Microsoft y Amazon; dicha simulación se realizará mediante un prototipo construido en casa.

## CAPÍTULO II

### MARCO TEÓRICO

Para llevar a cabo esta investigación, se ha realizado una búsqueda y revisión de trabajos escritos anteriormente, consultando a una serie de trabajos relacionados con el tema que servirán como base y orientación.

#### **2.1 Antecedentes de la Investigación**

**Rey J., Blanco A., (2016)** en el trabajo realizado titulado **“Diseño de un sistema de control domótico con una aplicación web para el control inteligente y automatización del hogar con un miniordenador Raspberry Pi”** para optar por el título de Ingeniero Electrónico de la Universidad José Antonio Páez (Venezuela) presentan la evaluación de automatizar un hogar utilizando alternativas de bajo costo, en dicho trabajo presentan la selección de los dispositivos utilizados y su funcionamiento, siendo la iluminación, los electrodomésticos, la seguridad, el riego, el clima y el almacenamiento de agua los puntos trabajados. Utilizaron características de programación web como el HTML5 y el lenguaje C, posteriormente se explica el diseño de las tarjetas electrónicas utilizadas y la interfaz de control por parte del usuario.

Así mismo, **Saunders J., Syrdal D.S., Koay K., Burke N., Dautenhahn K. (2015)** en el trabajo de investigación titulado **“Teach Me – Show Me – End-User personalisation of a Smart home and companion robot”** realizado en la University of Hertfordshire (Reino Unido) donde se estudian los problemas de atención y costos asociados con el aumento de la población de ancianos, la cual se está convirtiendo en una preocupación importante para muchos países, los investigadores detallaron en el artículo el enfoque de diseño de un sistema de enseñanza y aprendizaje de un robot acoplado, como una unidad integrada, a los componentes de un hogar inteligente. Los

resultados de esta evaluación indicaron que este enfoque de personalización es bastante útil y fácil de usar, además sintieron que eran capaces de usarlo en una situación real, sin embargo, también hubo alguna diferencia individual destacada dentro de la muestra de participantes.

Por último, **Valencia P., González L. (2014)** en el trabajo titulado “**Diseño e implementación de un prototipo de robot asistente para personas con discapacidad motriz y adultos mayores, basado en inteligencia artificial**” para optar por el título de Ingeniero Electrónico e Ingeniero en Sistemas respectivamente, en la Universidad Politécnica Salesiana (Ecuador), presentan la implementación de un asistente personal con el fin de hacer avances en el área de la salud. Este trabajo estuvo orientado a la ayuda de personas que presentan alguna discapacidad. Se utilizó un ordenador de tipo Laptop y algoritmos basados en lenguaje C y C++, debido a su poco consumo de los recursos del laptop y su sencillez, se apoyaron en librerías como OpenCV, las cuales realizan reconocimiento facial de distintos individuos, y las librerías Web Speech API de la compañía Google para complementar el reconocimiento efectivo del individuo en cuestión. Además, utilizan la placa de software libre Arduino para los dispositivos de control.

## **2.2 Bases Teóricas**

A continuación, se describirán las bases teóricas de los distintos temas necesarios, para una buena comprensión y realización exitosa de la investigación.

### **2.2.1 Robot Social**

“Un robot social es aquel que interactúa y se comunica con las personas de forma sencilla y agradable, siguiendo comportamientos, patrones y normas sociales, para eso se necesita que disponga de habilidades que se ubican dentro del dominio de la llamada inteligencia social la cual no es más que la habilidad que se utiliza para relacionarnos con afectividad con las personas que nos rodean” (Sergio Moriello, 2008). Los llamados robots sociales es una nueva generación de robots, que están

dando mucho de qué hablar en estos tiempos ya que son robots capaces de interactuar de forma directa, amigable y sencilla con su entorno, para la realización de diversas tareas como la seguridad, la automatización de hogares, la comunicación a través de voz, imágenes o videos, la educación, entretenimiento, entre otras muchas funciones.

## **2.2.2 Dispositivos de control**

Según Calderon S. (2014), se define un dispositivo de control como “un aparato eléctrico o electrónico que sirve para transmitir ordenes de control a los aparatos que lo soporten”. Son dispositivos encargados de administrar, ordenar, dirigir o regular el comportamiento de otro sistema, con el fin de reducir las probabilidades de fallo y obtener los resultados deseados. En esta investigación se empleará el miniordenador Raspberry Pi y el microcontrolador Arduino.

### **2.2.2.1 Raspberry Pi**

La Fundación Raspberry Pi (2012) mediante su página oficial define el Raspberry Pi como “un miniordenador de placa reducida de bajo costo. Este puede ser utilizado en proyectos de electrónica, y para muchas de las cosas que hace el PC de escritorio, como hojas de cálculo, procesamiento de textos, navegación por internet, y jugar. También reproduce vídeo de alta definición”. Por lo que podemos hacer notar que es un ordenador de tamaño bastante reducido, siendo esta una de sus mayores ventajas, debido a la comodidad que oferta a la hora de realizar un trabajo bastante compacto y presentarlo de forma adecuada, gracias a sus grandes características de procesamiento se pueden procesar datos y almacenarlos de manera bastante efectiva.

### **2.2.2.2 Arduino**

Según David Cuartielles (2011) co-creador del Arduino, este se define como “el hardware cuyas fuentes quedan abiertas para su uso por otras personas y entidades”. Arduino es una plataforma de hardware y software de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo

que está basado en el lenguaje de programación Processing. Es decir, una plataforma de código abierto para prototipos electrónicos. Al ser open source, tanto su diseño como su distribución, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin necesidad de licencia.

### **2.2.3 Domótica**

Definido por la Real Academia Española (2016), la domótica es un “conjunto de sistemas que automatizan diferentes instalaciones de una vivienda”. En si la domótica es un conglomerado de sistemas capaces de automatizar un hogar o un espacio de trabajo para obtener múltiples beneficios como lo son el ahorro de energía, seguridad, comunicación integral dentro y fuera del hogar.

### **2.2.4 Lenguaje de Programación**

Arias M. (2008), define los lenguajes de programación como “aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucción, operadores y reglas de sintaxis, que pone a disposición del programador para que este pueda comunicarse con los dispositivos de hardware y software existentes”. Siendo el lenguaje de programación la herramienta utilizada en este proyecto para crear todas las interacciones entre la máquina y el hombre.

#### **2.2.4.1 Lenguaje C++**

Este es definido por Linda I. Olivares Flores (2008) de la siguiente manera “C++ es un lenguaje de programación, creado como extensión del lenguaje C. En la actualidad, C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones, ya sea en Windows o GNU Linux”. Siendo el lenguaje C++ un lenguaje bastante sencillo y bastante genérico, acorde para la realización de este proyecto.

#### **2.2.4.2 Python**

Raúl González Duque (2009), define Python como “un lenguaje de programación creado por Guido Van Rossum a principios de los años 90. Trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados)”. Es una herramienta de programación multiparadigma, ya que es capaz de soportar la programación orientada a objetos y la programación funcional.

#### **2.2.4.3 JavaScript**

Definida por la Oracle Corporation (2012), como “un lenguaje de programación interpretado, dialecto estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y muy dinámico”. Es un lenguaje utilizado principalmente en su forma del lado del cliente, implementado como parte de un navegador web, que permite mejoras en la interfaz de usuario y páginas web dinámicas.

#### **2.2.4.4 Nodejs**

Definida por la Nodejs Foundation (2017), como “un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones Entradas/Salidas sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo”. Es un entorno en tiempo de ejecución multiplataforma, de código abierto para la capa del servidor, pero no limitándose a ello.

### 2.3 Definición de términos básicos.

**Prototipo:** Primer ejemplar que se fabrica de una figura, un invento u otra cosa, y que sirve de modelo para fabricar otras iguales, o molde original con el que se fabrica.

**Miniordenador:** computador de tamaño pequeño, el cual posee una capacidad de memoria mediana, puede funcionar de manera automática o conectado a una red.

**Microcontrolador:** es un circuito integrado programable, compuesto por bloques funcionales que cumplen funciones específicas las cuales están guardadas en su memoria.

**Librería o biblioteca:** en informática, una librería (o biblioteca), es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación que ofrece una interfaz bien definida para la funcionalidad a la que se invoca.

**V8:** es el entorno de ejecución para JavaScript creado para Google Chrome. Escrito en C++ y compila el código fuente JavaScript en código de máquina en lugar de interpretarlo en tiempo real.

**Skill:** se le llama así a las capacidades o habilidades que se le pueden añadir o configurar a un Robot.

## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

#### **3.1 Tipo de Investigación.**

Sobre el tipo de investigación, Canales (1996) señala que "hay diferentes tipos de investigación, los cuales se clasifican según distintos criterios..." La presente investigación cumple con la estructura de un proyecto factible, el cual según define Arias (2006), como "una propuesta de acción para resolver un problema práctico o satisfacer una necesidad. Es indispensable que dicha propuesta se acompañe de una investigación, que demuestre su factibilidad o posibilidad de realización".

En el caso del presente trabajo, la situación planteada anteriormente, se responderá con la construcción de un prototipo de Robot Social basado en un miniordenador Raspberry Pi y Arduino, para cumplir funciones básicas como un asistente a la persona que habita el hogar, todo esto con la finalidad de desarrollar e impulsar en mayor medida el uso de la tecnología robótica en Venezuela. Antes de emprender este tipo de estudio, se debe tener conciencia de que existen muchos procesos que necesitan ser comprendidos y manejados.

#### **3.2 Diseño de la Investigación**

Según Arias (1999), define el diseño de la investigación como "la estrategia que adopta el investigador para responder al problema planteado" (p.30). El siguiente trabajo está clasificado como una investigación de campo siendo este definido por la UPEL (2005) como "el análisis sistemático de problemas en la realidad con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y efectos o producir su ocurrencia, haciendo uso de métodos característicos de cualquier paradigma o enfoques de investigaciones conocidas o en desarrollo".

### **3.3 Nivel de la investigación**

Según Arias (1997), el nivel de investigación puede definirse como “el grado de profundidad con que se aborda un objeto o fenómeno” (p.47). El tipo de investigación a realizar determina los niveles que es preciso desarrollar” (p.101).

Este trabajo se ha considerado de tipo descriptivo el cual es definido por Sabino (1986) como “una investigación que trabaja sobre realidades de hechos, y su característica fundamental es la de presentar una interpretación correcta. Para la investigación descriptiva, su preocupación primordial radica en descubrir algunas características fundamentales de conjuntos homogéneos de fenómenos, utilizando criterios sistemáticos que permitan poner de manifiesto su estructura o comportamiento. De esta forma se pueden obtener las notas que caracterizan a la realidad estudiada”. (Pág. 51). Este nivel de investigación consiste, fundamentalmente, en caracterizar un fenómeno o situación concreta indicando sus rasgos más peculiares o diferenciadores, es decir, en si el objetivo de este nivel de investigación es el de conocer las situaciones y actitudes frente a un tema en particular, no quedándose solo en la recolección de datos sino también ayuda a predecir e identificar la relación que existe entre dos o más variables.

### **3.4 Población y Muestra**

La población es un conjunto de individuos de la misma clase, limitada por el estudio. Según Tamayo y Tamayo, (1997) “La población se define como la totalidad del fenómeno a estudiar donde las unidades de población poseen una característica común la cual se estudia y da origen a los datos de la investigación” (P.114). La muestra por otro lado es a la que se le aplicará el instrumento de recolección de datos en nuestro caso para identificar o precisar la información que se requiere. Según Tamayo y Tamayo (1997) afirma que la muestra “es el grupo de individuos que se te toma de la población para estudiar el fenómeno estadístico” (p.38).

Tomando en cuenta las definiciones anteriores en este trabajo se tomará como población a los alumnos de Ingeniería Electrónica del 8vo, 9no y 10mo semestre y como muestra tomaremos el 50% de dicha población.

### **3.5 Técnicas e Instrumentos de investigación**

Carlos Sabino (1996) expone que “un instrumento de recolección de datos es en principio cualquier recurso de que pueda valerse el investigador para acercarse a los fenómenos y extraer de ellos información. De este modo el instrumento sintetiza en sí toda la labor previa de la investigación resume los aportes del marco teórico al seleccionar datos que corresponden a los indicadores y, por lo tanto, a las variables o conceptos utilizados”. Por otro lado, Fernando Castro Márquez indica que “las técnicas están referidas a la manera como se van a obtener los datos y los instrumentos son los medios materiales, a través de los cuales se hace posible la obtención y archivo de la información requerida para la investigación”. Con respecto a estas definiciones, se procede a precisar las diferentes técnicas e instrumentos utilizados en esta investigación.

#### **3.5.1 Técnicas**

##### **3.5.1.1 Observación Directa**

Gloria Otalvaro (2014) establece que “el método de observación directa es uno de los más utilizados, por su eficacia. Su aplicación resulta mucho más eficaz cuando se consideran estudios de micro-movimientos, de tiempos y métodos. El análisis del cargo se efectúa observando al ocupante del cargo, de manera directa y dinámica, en pleno ejercicio de sus funciones, mientras el analista de cargos anota los datos clave de su observación en la hoja de análisis de cargos”. Este método se ejecutará observando las necesidades de la población de forma directa, es decir, estando inmersos en el lugar y en el momento que ocurren los acontecimientos; para así definir las principales características que va a tener el prototipo en cuestión.

### **3.5.1.2 Encuesta**

La encuesta la define García Ferrado (1993) como “una investigación realizada sobre una muestra de sujetos representativa de un colectivo más amplio, utilizando procedimientos estandarizados de interrogación con intención de obtener mediciones cuantitativas de una gran variedad de características objetivas y subjetivas de la población”. Mediante la encuesta se obtienen datos de interés sociológico interrogando a los miembros de un colectivo o de una población.

La encuesta es una técnica de recolección de datos mediante la aplicación de un cuestionario a una muestra de individuos. A través de las encuestas se pueden conocer las opiniones, las actitudes y los comportamientos de los ciudadanos. En este caso se utilizará esta técnica para analizar la aceptación que se tiene acerca de la tecnología utilizada en este trabajo y qué tan familiarizados están acerca del tema. Por otro lado, se les propondrán algunas características potenciales que pueda tener el prototipo en cuestión para analizar sus impresiones.

### **3.5.2 Instrumento**

#### **3.5.2.1 Cuestionario**

Según Manuel Galán Amador (2009) “El cuestionario es un conjunto de preguntas diseñadas para generar los datos necesarios para alcanzar los objetivos propuestos del proyecto de investigación. El cuestionario permite estandarizar e integrar el proceso de recopilación de datos”. En este caso el cuestionario es un conjunto de preguntas cerradas que serán utilizadas como instrumento para respaldar la encuesta, con el fin de dejar asentado las impresiones de los encuestados y hacer más fácil y confiable la recolección de datos.

### **3.6 Fases Metodológicas**

Para el desarrollo de las fases metodológicas se apoyará en lo descrito en los objetivos específicos, describiendo de manera breve como se lograrán desarrollar y lo que se pretende obtener con cada objetivo.

### **Fase I: Examinación de las aplicaciones de robótica y domótica existentes en Venezuela.**

En esta fase, se utilizara la investigación documental o revisión bibliográfica, siendo definida por Arias (2004) “la investigación documental es un proceso basado en la búsqueda, recuperación, análisis, crítica e interpretación de datos secundarios, es decir, los obtenidos y registrados por otros investigadores en fuentes documentales, impresas, audiovisuales o electrónicas”; se realiza dicha actividad con el fin conocer el nivel de desarrollo del país respecto a este tema, para luego realizar una toma de decisiones y situar el proyecto en donde haya más déficit y así lograr un desarrollo innovador.

### **Fase II: Identificación de las potenciales funcionalidades que podría tener el Robot Social a desarrollar.**

En la siguiente fase, se procederá a estudiar el caso con la técnica de observación directa, la cual permitirá visualizar las necesidades más comunes en este tipo de proyecto. Para la presente investigación se plantea utilizar como técnica de recolección de datos la encuesta. Para Trespacios, Vázquez y Bello (2005), “las encuestas son instrumentos de investigación descriptiva que precisan identificar a priori las preguntas a realizar, las personas seleccionadas en una muestra representativa de la población, especificar las respuestas y determinar el método empleado para recoger la información que se vaya obteniendo”.

### **Fase III: Determinación de los softwares adecuados para la elaboración del Robot Social.**

En la presente fase, se tomarán en cuenta los resultados obtenidos de la encuesta realizada en la fase II, también el aspecto económico y el tiempo otorgado para la realización de este trabajo de grado, con el fin de realizar la investigación pertinente que facilite la selección e instalación adecuada de los diferentes softwares, tomando en

cuenta la interacción entre ellos, permitiendo desarrollar satisfactoriamente las Skills del Robot Social.

**Fase IV: Desarrollo del prototipo de Robot Social.**

En esta fase del proyecto, se procederá a realizar la elaboración de las distintas skills del Robot Social seleccionadas en la fase II, utilizando los softwares seleccionados en la fase III.

## **CAPÍTULO IV**

### **RESULTADOS**

Las siguientes fases describen el desarrollo de los objetivos específicos establecidos al inicio de este trabajo de grado.

#### **4.1 Fase I. Examinación de las aplicaciones de robótica y domótica existentes en Venezuela.**

En la siguiente fase se examinaron los diferentes proyectos y aplicaciones existentes en Venezuela, pertenecientes al área de la robótica y la domótica, esto con el fin de conocer el grado de desarrollo de las mismas en el país y de establecer los aportes que ofrece esta investigación.

En Venezuela se ha demostrado un creciente interés por los avances tecnológicos y los aparatos electrónicos de última generación que se encuentran en el mercado, siendo uno de los países latinoamericanos ubicados en los primeros lugares en cuanto al consumismo electrónico se refiere, superando incluso a México y Brasil. Cabe destacar que adquirir dicha tecnología de punta, en la gran mayoría de los casos, representa un gasto económico el cual muy pocos pueden darse el lujo de costear; el venezolano es muy selectivo a la hora de adquirir cualquier aparato de última generación esperando que su inversión sea merecedora del esfuerzo realizado para adquirir dicho artefacto.

Como consecuencia de esto, en Venezuela, la cantidad de emprendedores independientes o empresas relacionadas con este tipo de estudio son bastante escasas destacando la participación de la Asociación Venezolana de Robótica y Domótica (AVEROD), la cual es una asociación sin fines de lucro, donde investigadores, empresarios y aficionados pueden compartir sus conocimientos, tendencias y novedades sobre el campo de la robótica y domótica. Según los estatutos en el artículo

2, esta asociación tiene como fines:

- Dar a conocer al público en general el mundo de la robótica, la domótica y promover la investigación y desarrollo en esta rama de la ciencia.
- Ofrecer un espacio público, a todo aquel que esté interesado en la robótica y la domótica.
- Creación de documentos, talleres, cursos en robótica y áreas afines.
- Generar una publicación periódica anual.
- Promover en el sector productivo las virtudes y beneficios de la robótica.
- Canalizar las necesidades del sector productivo en robótica y domótica, así como áreas afines, hacia los institutos, centros, laboratorios y grupos afiliados a la asociación.
- Ofrecer y pedir apoyos al resto de la sociedad, ya sean grupos privados, estatales o empresariales.

Por otro lado, la competencia nacional “USBBots”, involucra a distintas áreas de conocimiento como: computación, informática, electrónica, mecánica, control, automatización y mecatrónica. Esta competencia se hace con la participación de investigadores, estudiantes y aficionados.

Durante la competencia se desarrollan exhibiciones de microbots y robots de diferentes grupos a nivel nacional, adicionalmente, este evento es la sede de la reunión anual de la Asociación Venezolana de Robótica y Domótica (AVEROD). USBBots se basa en competencias similares a las que se realizan en todo el mundo y pretende incentivar a estudiantes y aficionados a la construcción de robots para tareas específicas y compartir experiencias con grupos de diferentes universidades nacionales; la última de estas competencias fue desarrollada del 7 al 9 de noviembre del 2016 en la ciudad de Caracas. Estas competencias tienen como objetivos:

- Fomentar el desarrollo de la capacidad de abstracción y acercamiento al pensamiento lógico formal.
- Incentivar el trabajo en equipo.

- Generar espacios de discusión con respecto a las diversas soluciones posibles que presentan los desafíos en robótica.
- Estimular la vocación en los alumnos por la investigación y el desarrollo tecnológico.
- Lograr un punto de encuentro entre estudiantes, aficionados y profesionales al mundo de la robótica.
- Propiciar el intercambio de experiencias de grupos de robótica promoviendo la participación en proyectos conjuntos para resolver problemas específicos del ámbito nacional.

En Abril de 2016, en la Universidad José Antonio Páez, se presenta el trabajo de grado titulado “Diseño de un Sistema de Control Domótico con una Aplicación Web para el Control Inteligente y Automatización del Hogar con un miniordenador Raspberry Pi”, desarrollado por Rey José y Blanco Alfredo, el cual por medio de desarrollo web, busca dotar de un control y una automatización del hogar utilizando distintos elementos electrónicos, siendo el Raspberry Pi, el elemento principal del trabajo de grado, logrando así un desarrollo eficiente y de bajo costo al momento de su implementación.

#### **4.2 Fase II. Identificación de las potenciales funcionalidades que podría tener el Robot Social a desarrollar.**

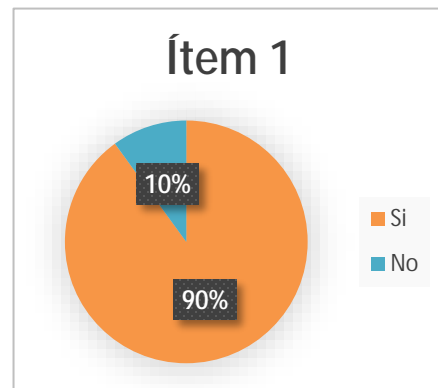
Citando a Christian Balkenius (2003) Profesor de la Universidad de Lund, ubicada en Suecia, el cual es experto en el área de la inteligencia artificial y redes neuronales artificiales, expresa que “un robot social debería tener apenas la inteligencia necesaria para limpiar, encerar, lavar, planchar, transportar objetos alrededor de la casa, sacar la basura, utilizar artefactos eléctricos del hogar, reconocer a las personas, sostener conversaciones simples y atender visitas”.

Con el fin de analizar las opiniones, aceptación y nivel de familiarización que tiene la muestra definida en el Capítulo III acerca del tema en cuestión, se llevó a cabo

una encuesta de preguntas cerradas. Sabiendo dicha información, y considerando otros aspectos como el económico y el tiempo definido para la realización de esta investigación, se seleccionarán las características y funcionalidades que va a tener el prototipo de Robot Social a desarrollar.

**1. ¿Había oído hablar sobre el Robot Social?**

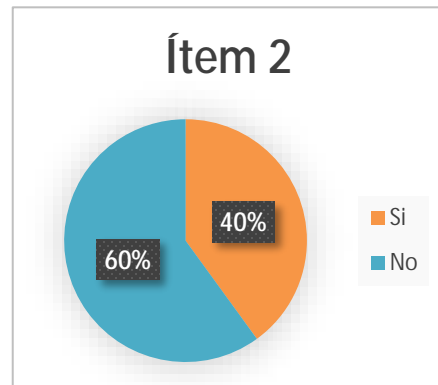
El 90% de la muestra ha oído hablar de lo que es un Robot social. (Ver Gráfico 1).



**Gráfico 1. Resultado ítem 1.**

**2. ¿Cree usted que para que un dispositivo se considere robot social debe tener estrictamente apariencia física?**

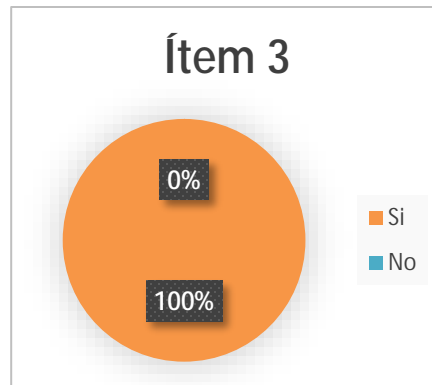
El 60% de la muestra esta consiente que para que un dispositivo se considere Robot Social no necesariamente debe contar con una apariencia física. (Ver Gráfico 2).



**Gráfico 2. Resultado ítem 2.**

**3. ¿Está de acuerdo con el desarrollo de este tipo de tecnología?**

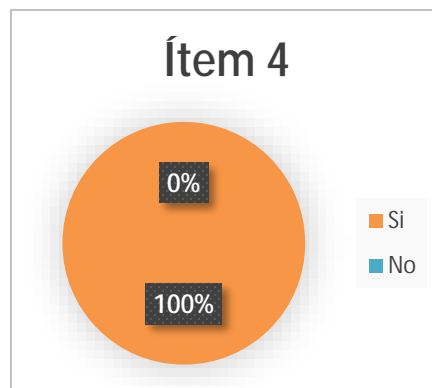
El 100% de la muestra considera que se debe desarrollar la tecnología expuesta en este trabajo de grado. (Ver Gráfico 3).



**Gráfico 3. Resultado ítem 3.**

**4. ¿Le gustaría que el Robot Social interactúe o controle los dispositivos de su hogar?**

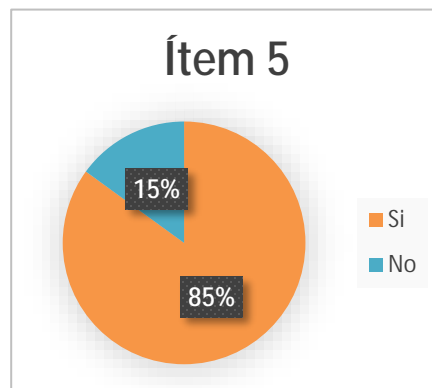
Al 100% de los encuestados les gustaría que el Robot Social interactúe o controle los dispositivos de sus hogares. (Ver Gráfico 4).



**Gráfico 4. Resultado ítem 4.**

**5. ¿Le gustaría tener un dispositivo que lo ayude a organizar sus actividades diarias?**

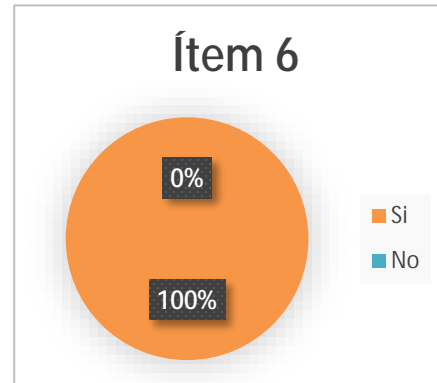
Al 85% de los encuestados les gustaría tener un dispositivo que les ayude a organizar sus actividades diarias como llevar un cronograma de actividades, tener cargada una lista de mercado. (Ver Gráfico 5).



**Gráfico 5. Resultado ítem 5.**

**6. ¿Cree usted pertinente tener un dispositivo que lo apoye en la seguridad de su hogar?**

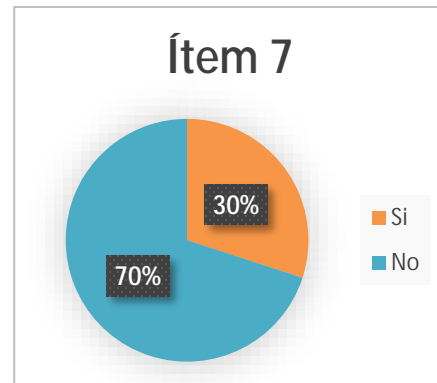
El 100% de los encuestados están interesados en tener un dispositivo de seguridad para sus hogares, hasta el punto de catalogar esta característica de primera necesidad. (Ver Gráfico 6).



**Gráfico 6. Resultado ítem 6.**

**7. ¿Implementaría en su hogar una inteligencia artificial que reciba a sus visitas?**

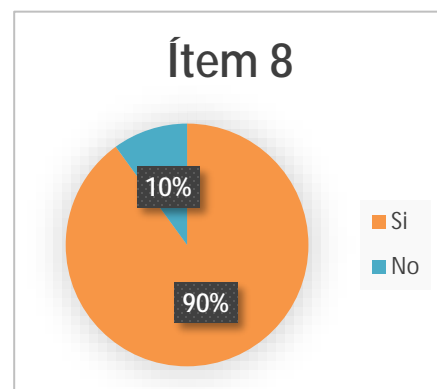
El 70% de los encuestados no está de acuerdo con implementar esta característica, ya que en su mayoría alegan que es mejor atender a las visitas de forma personal. (Ver Gráfico 7).



**Gráfico 7. Resultado ítem 7.**

**8. ¿Le gustaría un dispositivo inteligente con el que usted pueda mantener una conversación simple?**

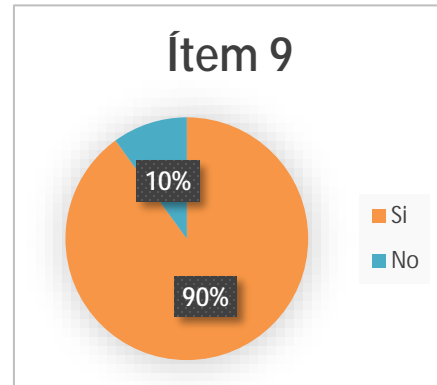
El 90% de la muestra le gustaría interactuar mediante una conversación simple con un dispositivo inteligente. (Ver Gráfico 8).



**Gráfico 8. Resultado ítem 8.**

**9. ¿Quisiera usted tener un dispositivo que le ayude con las labores domésticas?**

El 90% de la muestra quisiera tener un dispositivo que ayude con las labores domésticas. (Ver Gráfico 9).



**Gráfico 9. Resultado ítem 9.**

Al analizar los datos, se pudo obtener un panorama claro de las funcionalidades que debe tener el prototipo a desarrollar. Unido a esto, se tomó en consideración el ámbito económico con el que se contaba y el tiempo otorgado para la realización de este trabajo de grado, llegando a la conclusión de que las características o funcionalidades que finalmente va a tener el prototipo de Robot Social son:

- Controlar artefactos eléctricos.
- Sostener conversaciones simples.
- Seguridad.

**4.3 Fase III. Determinación de los softwares adecuados para la elaboración del Robot Social.**

Conociendo los resultados obtenidos en la fase II de esta investigación y sabiendo que se dispone de un miniordenador Raspberry Pi y un microcontrolador Arduino para la elaboración del prototipo de Robot Social, se necesita seleccionar entornos de programación compatibles con los mismos, así como un software principal que nos permita enlazar todos estos entornos.

Se estudiaron las diferentes opciones ya disponibles en la Internet y el mercado hasta dar con Alexa, la cual es una asistente de voz desarrollada por la mundialmente

ya conocida empresa Amazon. Esta asistente virtual funciona en el Amazon Echo y Amazon Dot, dos productos desarrollados por la misma empresa, y posee una amplia libertad para realizarle modificaciones y adiciones a necesidad del usuario, convirtiéndolo así en el software principal más adecuado para la elaboración del prototipo.

El miniordenador Raspberry Pi es considerado como la unidad de control maestro del sistema y para dicha aplicación se le instalaron los programas esenciales, los cuales se muestran a continuación:

- Sistema Operativo Raspbian Jessie.
- Paquetes y librerías necesarios para la conexión y control del microcontrolador Arduino.
- Asistente de Voz Alexa.

Para la instalación del sistema operativo se utilizó NOOBS versión 2.4.0, es un sencillo instalador que contiene el sistema Raspbian Jessie, el cual fue seleccionado de entre tantas opciones debido a que no se necesita una conexión a internet para poder prescindir del mismo y posee más de 36.000 paquetes de software pre compilados en un formato amigable y de fácil instalación.

El Raspberry Pi no posee memoria interna, por lo tanto, para instalar dicho sistema operativo se necesitó el uso de una memoria microSD clase 10. El fabricante del producto recomienda utilizar una de más de 8 GB, a pesar de esto en el prototipo desarrollado en este trabajo de grado se utilizó una de 8GB y no se presentaron problemas con el espacio de almacenamiento disponible. Este tampoco posee monitor, teclado y mouse, por lo que se tuvo que usar un cable HDMI para conectar al miniordenador a un monitor externo, un teclado USB y un mouse USB fueron agregados al conjunto de hardware; que posteriormente fueron retirados por los altos consumos de energía.

Al término de la instalación de los programas esenciales para el correcto funcionamiento del Raspberry Pi, se procede a instalar programas secundarios

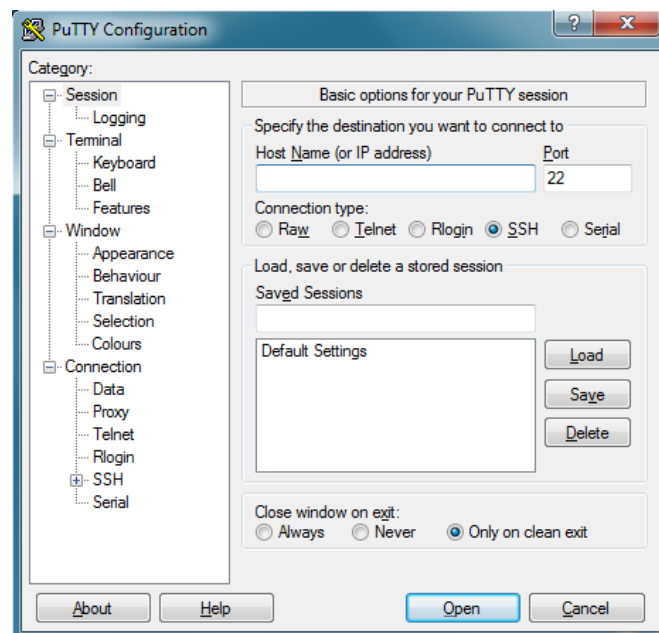
considerados necesarios para el desarrollo del prototipo de Robot Social. Con ayuda de la configuración de dos programas secundarios como lo son el PuTTY y el VNC Viewer logra desconectar los agentes externos o periféricos del Raspberry Pi (monitor, teclado y mouse) para dar mayor libertad al desarrollo del prototipo, a partir de este punto se procede a explicar brevemente su instalación y funcionamiento.

### 4.3.1 Instalación de los softwares secundarios del Raspberry Pi.

#### 4.3.1.1 SSH en el Raspberry Pi.

Es el nombre de un protocolo y del programa que lo implementa, sirve para acceder a máquinas remotas a través de una red. Este programa viene por defecto instalado en el Raspbian Jessie, por lo que configurarlo no es problema, solo se deben seguir los siguientes pasos:

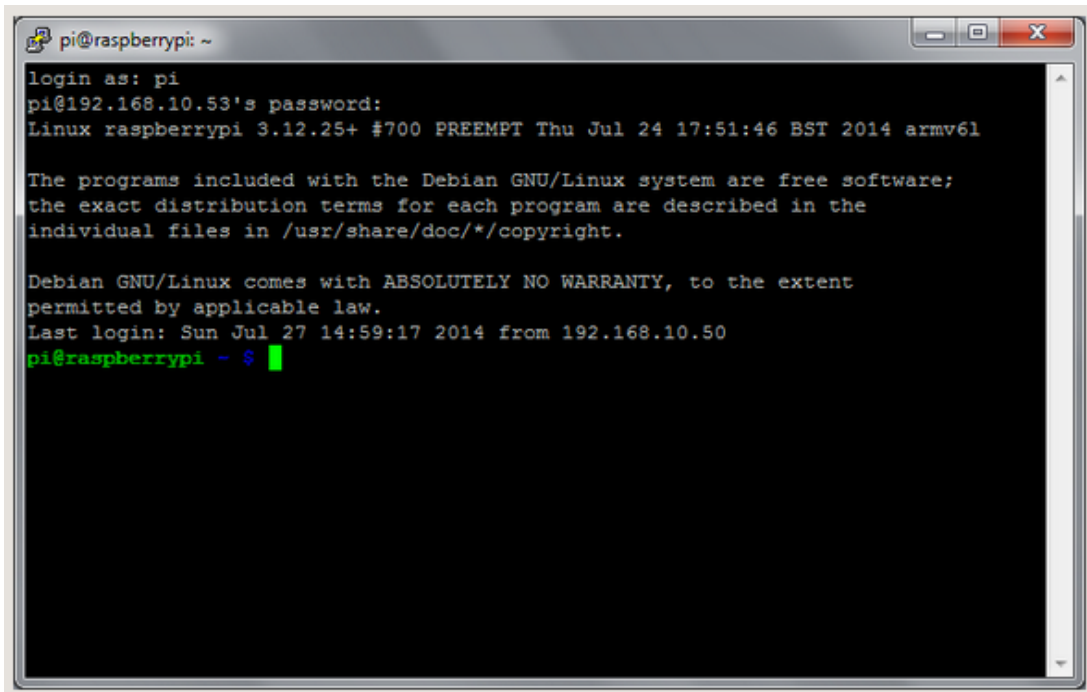
1. Como el computador disponible utiliza Windows, se necesita descargar un cliente de SSH, en este caso se optó por PuTTY, el cual no necesita instalación, ya que es un ejecutable y tiene un aspecto como se ve en la figura 1.



**Figura 1. Ejecución del cliente PuTTY.**

Fuente: Infante y González (2017).

2. Una vez abierto el ejecutable, en la sección de “Host Name” se colocó la dirección IP del Raspberry Pi.
3. A continuación, aparece una ventana común de inicio de sesión, en el cual se introdujo el usuario (pi) y la contraseña (raspberry) del Raspberry Pi.
4. Luego de esto, se obtuvo acceso a una ventana terminal del Raspberry como se muestra en la figura 2.



```
pi@raspberrypi: ~
login as: pi
pi@192.168.10.53's password:
Linux raspberrypi 3.12.25+ #700 PREEMPT Thu Jul 24 17:51:46 BST 2014 armv61

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Jul 27 14:59:17 2014 from 192.168.10.50
pi@raspberrypi ~ $
```

**Figura 2. Ventana terminal del Raspberry Pi para PuTTY.**

Fuente: Infante y González (2017).

#### **4.3.1.2 VNC en el Raspberry Pi.**

Para instalar el VNC en el Raspberry, se utilizó un pseudo-programa llamado TightVNC Server, el cual cumple la función de hacer al Raspberry Pi un servidor. Para este proceso se siguen los siguientes pasos:

1. Se abrió una ventana terminal del Raspberry Pi en la que se escribió:

*sudo apt-get install tightvncserver*

2. Se inició una instalación de manera automática; seguidamente se inició el servidor VNC al escribir en la misma ventana terminal:

*tightvncserver*

3. El sistema exige configurar una clave de acceso al Raspberry Pi, la cual se necesitó para acceder al Raspberry Pi desde un computador o teléfono inteligente.

En otro orden de ideas, se destaca que cada vez que el Raspberry Pi es reiniciado o apagado, el servidor VNC se cae, de manera que hay que reconectar el monitor, mouse y teclado externos para poder iniciar de nuevo el servidor; con ayuda de un Script se resolvió ese inconveniente y el VNC se inicia ahora de forma automática al momento de que el sistema operativo se inicia.

A continuación, se pretende acceder al Raspberry Pi vía VNC por medio de la nube, es decir, en el procedimiento anterior se configuró al Raspberry Pi como servidor, ahora se requirió de la utilización de otra parte del programa VNC, llamada “VNC Viewer”, para tener acceso al servidor y manipularlo a través del computador. Con estas configuraciones ya se pudo desconectar el monitor, el mouse y el teclado del Raspberry Pi, gracias a la configuración de ambos programas, todo el trabajo se pudo desarrollar a través del computador. Una vez finalizada la instalación de los considerados softwares secundarios, se procede a la instalación del software principal, Alexa.

#### **4.3.2 Alexa.**

Antes de la instalación del asistente de voz Alexa se tuvieron que agregar componentes de hardware adicional al Raspberry Pi para su correcto funcionamiento, como lo son un par de cornetas externas con salida de audio de 3.5mm y un mini micrófono USB 2.0. Además, se debieron incluir librerías que no posee el Raspbian Jessie, que son parte vital para el funcionamiento de Alexa y para los entornos de programación utilizados en la creación de las skills de la misma. Estas librerías son:

**Maven (MVN)**, es una herramienta de software para la gestión y construcción de proyectos java. Es similar a la funcionalidad de Apache Ant y PEAR de PHP, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML.

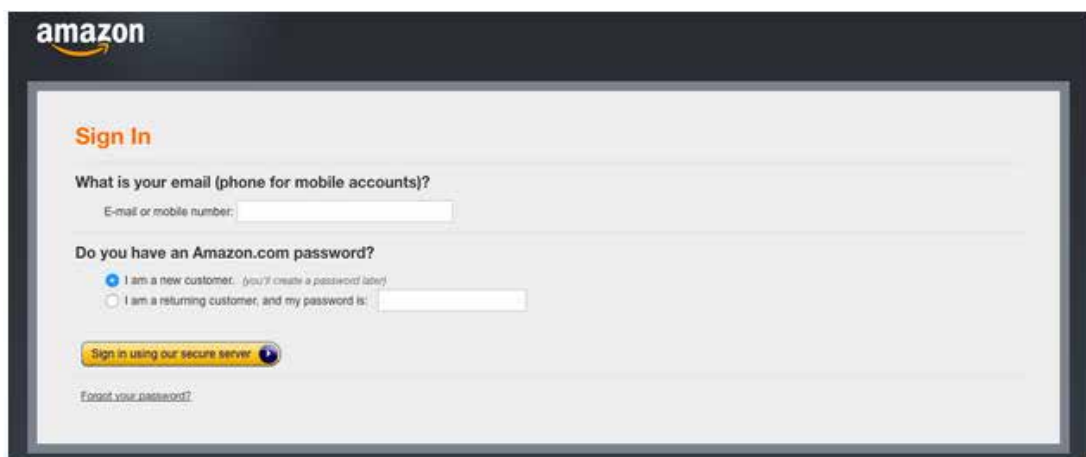
**Node.js**, el cual es un intérprete Javascript del lado del servidor, que cambia la noción de cómo debería trabajar un servidor. Su meta es permitir a un programador construir aplicaciones altamente escalables y escribir códigos que maneje decenas de miles de conexiones simultaneas en una sola máquina física.

**Npm (Node Package Manager)**, Gestor de paquetes de javascript de Node.js por excelencia.

**Nvm (Node Version Manager)**, Script que permite manejar múltiples versiones del Node.js al mismo tiempo.

Ahora se procede con el registro de Alexa en el Desarrollador de Amazon de este asistente a través de los siguientes pasos:

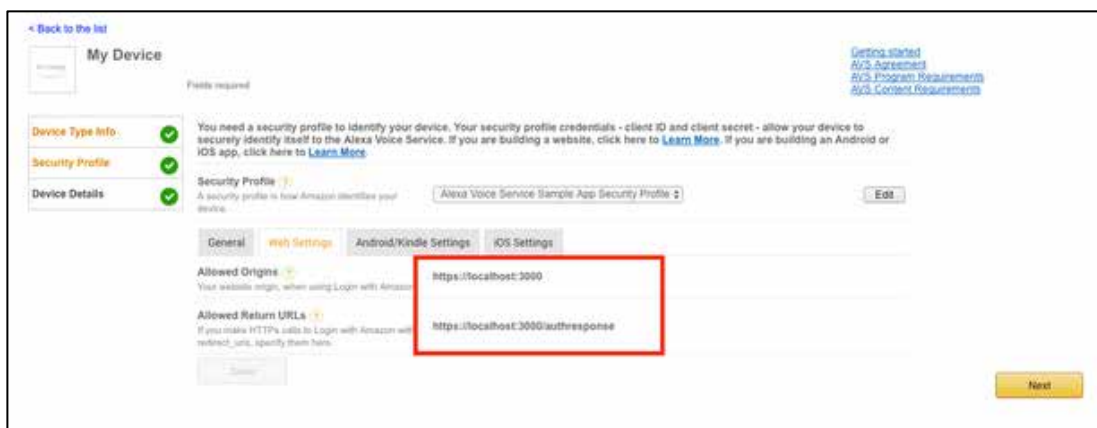
1. Se creó un usuario en el Desarrollador de Amazon accediendo al portal designado de Amazon para este fin (Ver Figura 3), se registró la cuenta de correo electrónico (o número telefónico) a la que se asoció Alexa.



**Figura 3. Desarrollador de Amazon.**

Fuente: <https://developer.amazon.com> (2017).

2. Se llenó el perfil del dispositivo.
3. Se registró el producto.
4. Luego se creó un Dispositivo Alexa y un perfil de seguridad. Durante este proceso se tomó nota de los parámetros “ProductID”, “ClientID” y “ClientSecret”, ya que estos parámetros serán necesarios más adelante.
5. En Device Detail, se le asignó la dirección IP a Alexa de la siguiente forma: en “Allowed Origins” se colocó `https://localhost:3000` y en “Allowed Returns URLs”: se colocó `https://localhost:3000/authresponse`. Con esto se le indica que Alexa en donde se va a ejecutar. (Ver figura 4)



**Figura 4. Device Detail.**

Fuente: <https://developer.amazon.com> (2017).

6. Para finalizar con la instalación y el registro del desarrollador de Amazon se le colocó un dominio de internet, recomendado por el mismo desarrollador (`www.example.com`).

Luego de este procedimiento se instaló la aplicación demo de Alexa en el Raspberry Pi, teniendo en consideración que el instalador debe ser modificado con las credenciales obtenidas en el paso 4 del registro en el desarrollador de Amazon (“ProductID”, “ClientID” y “ClientSecret”). (Ver figura 5).

```
pi@raspberrypi: ~/Desktop/alexa-avs-sample-app
File Edit Tabs Help
GNU nano 2.2.6 File: automated_install.sh Modified
#!/bin/bash
#-----
# Paste from developer.amazon.com below
#-----
# This is the name given to your device or mobile app in the Amazon developer portal.
To look this up, navigate to https://developer.amazon.com/edw/home.html. It may be lab
eled Device Type ID.
ProductID=my_device
# Retrieve your client ID from the web settings tab within the developer console: http
s://developer.amazon.com/edw/home.html
ClientID=amzn1.application-
# Retrieve your client secret from the web settings tab within the developer console:
https://developer.amazon.com/edw/home.html
ClientSecret=88b915
#-----
# No need to change anything below this...
#-----
AG Get Help   AG WriteOut  AR Read File  AY Prev Page  AR Cut Text   AG Cur Pos
AX Exit       AJ Justify   AR Where Is  AY Next Page  AC UnCut Text AT To Spell
```

**Figura 5. Ventana Terminal del Raspberry Pi para Alexa I.**

Fuente: Infante y González (2017).

Al término de la instalación se ejecutaron tres scripts importantes, el servicio web, la demo de Alexa y los motores de activación por palabra.

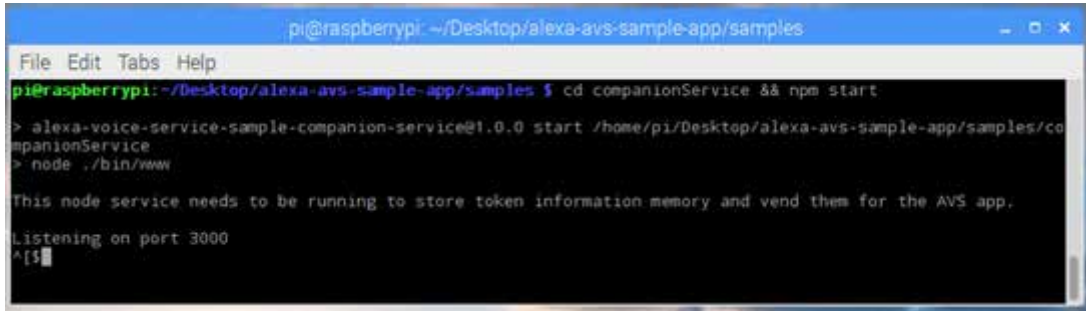
1. Para ejecutar el servicio web por autorización ver figura 6.

```
pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples
File Edit Tabs Help
pi@raspberrypi:~/Desktop/alexa-avs-sample-app/samples $ cd companionService && npm start
```

**Figura 6. Ventana Terminal del Raspberry Pi para Alexa II.**

Fuente: Infante y González (2017).

2. Para ejecutar la demo de Alexa ver figura 7.



```
pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples
File Edit Tabs Help
pi@raspberrypi:~/Desktop/alexa-avs-sample-app/samples $ cd companionService && npm start
> alexa-voice-service-sample-companion-service@1.0.0 start /home/pi/Desktop/alexa-avs-sample-app/samples/companionService
> node ./bin/www

This node service needs to be running to store token information memory and vend them for the AVS app.
Listening on port 3000
^C$
```

**Figura 7. Ventana Terminal del Raspberry Pi para Alexa III.**

Fuente: Infante y González (2017).

3. Para ejecutar los motores de activación por palabra ver figura 8.



```
pi@raspberrypi: ~/Desktop/alexa-avs-sample-app/samples
File Edit Tabs Help
pi@raspberrypi:~/Desktop/alexa-avs-sample-app/samples $ cd javaclient && mvn exec:exec
```

**Figura 8. Ventana Terminal del Raspberry Pi para Alexa IV.**

Fuente: Infante y González (2017).

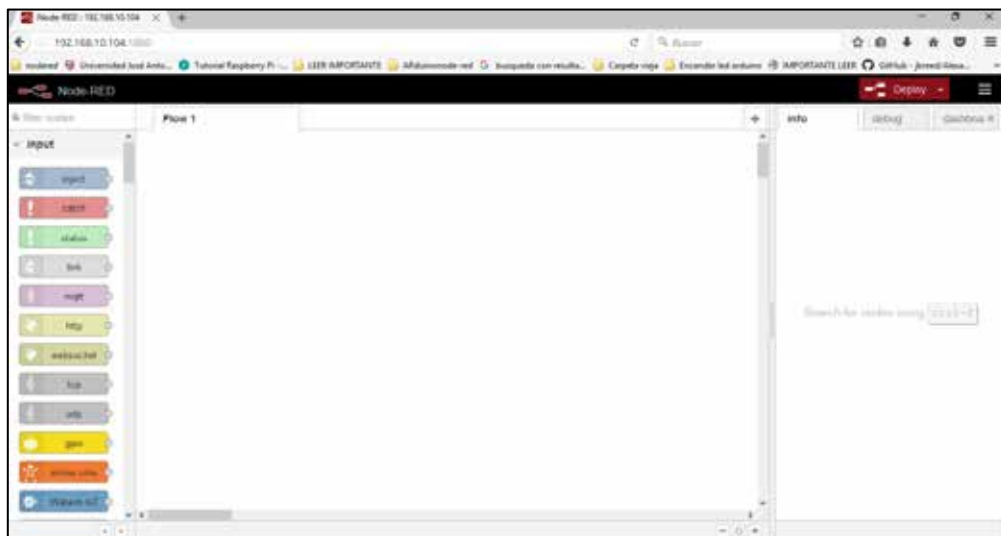
Al dejar funcional al asistente Alexa, se prosiguió con la instalación y configuración del software utilizados para la creación de los algoritmos que conforman las skills de dicho asistente.

### 4.3.3 Node-red.

Es un motor de flujos programado en NodeJS con enfoque IoT (Internet of Things o Internet de las Cosas) desarrollado por la empresa IBM Bluemix, con el fin de comercializar sus dispositivos IoT. Esta herramienta permite definir gráficamente flujos de servicios, a través de protocolos estándares como REST, MQTT, WebSocket,

AMQP entre otros, además de ofrecer integración con Apis de terceros, tales como Twitter, Facebook, Yahoo!. Por estas y otras razones grandes empresas están apostando por este tipo de motor de desarrollo.

El editor de flujos de Node-Red consiste en una sencilla interfaz HTML, accesible desde cualquier navegador, en la que arrastrando y conectando nodos entre sí, es posible definir un flujo que ofrezca un servicio. El editor de flujos de Node-Red es tal como se muestra en la figura 9, en donde se observan, en el lado izquierdo, las diferentes librerías con sus respectivos nodos, en la parte central el espacio donde se desarrolla la skill y a la derecha se muestra la información de cada nodo que estemos utilizando.



**Figura 9. Editor de flujos de Node-Red**

Fuente: Infante y González (2017).

El Raspberry Pi posee Node-Red preinstalado; pero desafortunadamente no es la última versión estable de dicho editor, sin embargo, se decide utilizar este motor de flujo debido a su gran versatilidad, además de que posee librerías de última tecnología que pueden ser agregadas al editor de desarrollo (librerías compatibles con Arduino, Raspberry Pi, sensores, entre otros).

Para utilizarlo y aprovechar todas sus ventajas, se actualizo la versión del Node-red de la siguiente forma:

1. En el terminal de las Raspberry Pi se escribe el comando:

```
update-nodejs-and-nodered
```

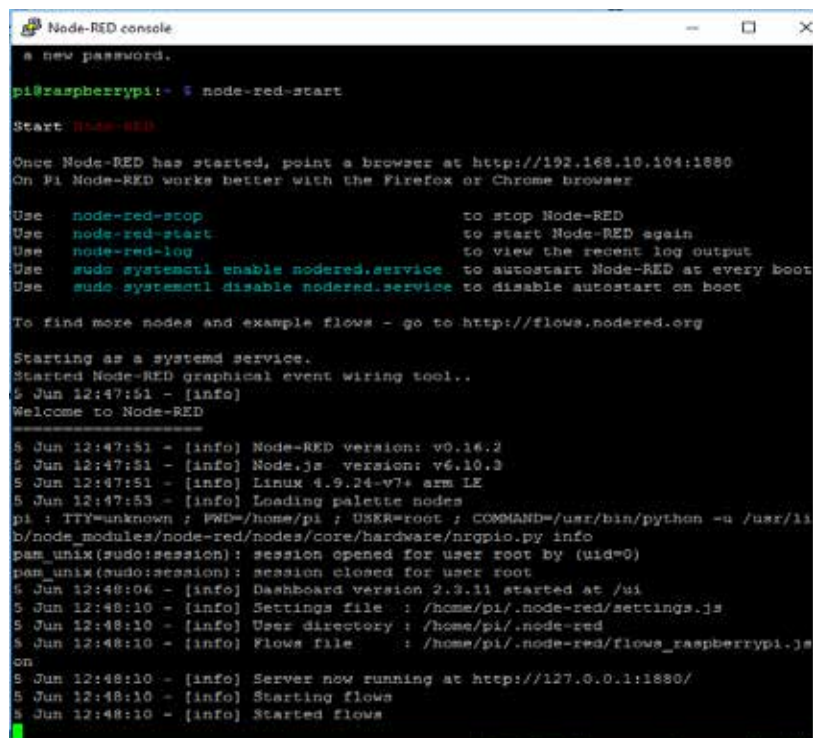
2. Durante esta actualización, la librería de npm se verá afectada, de manera que cuando se termina dicha actualización, hay que recuperar los datos perdidos de la librería npm, haciendo uso del siguiente comando en la misma ventana terminal:

```
cd ~/.node-red
```

```
npm rebuild
```

3. Con el paso anterior, Node-red ya se encuentra completamente funcional, ahora, para poder iniciarlo, en una nueva ventana terminal (ver figura 10), se escribe el comando:

```
node-red-start
```



```
Node-RED console
a new password.
pi@raspberrypi:~$ node-red-start
Start Node-RED

Once Node-RED has started, point a browser at http://192.168.10.104:1880
On Pi Node-RED works better with the Firefox or Chrome browser

Use node-red-stop to stop Node-RED
Use node-red-start to start Node-RED again
Use node-red-log to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot

To find more nodes and example flows - go to http://flows.nodered.org

Starting as a systemd service.
Started Node-RED graphical event wiring tool..
5 Jun 12:47:51 - [info]
Welcome to Node-RED

5 Jun 12:47:51 - [info] Node-RED version: v0.16.2
5 Jun 12:47:51 - [info] Node.js version: v6.10.3
5 Jun 12:47:51 - [info] Linux 4.9.24-v7+ arm LE
5 Jun 12:47:53 - [info] Loading palette nodes
pi : TTY=unknown ; PWD=/home/pi ; USER=root ; COMMAND=/usr/bin/python -u /usr/lib/node_modules/node-red/nodes/core/hardware/nrgpio.py info
pam_unix(sudo:session): session opened for user root by (uid=0)
pam_unix(sudo:session): session closed for user root
5 Jun 12:48:06 - [info] Dashboard version 2.3.11 started at /ui
5 Jun 12:48:10 - [info] Settings file : /home/pi/.node-red/settings.js
5 Jun 12:48:10 - [info] User directory : /home/pi/.node-red
5 Jun 12:48:10 - [info] Flows file : /home/pi/.node-red/flows_raspberrypi.js
on
5 Jun 12:48:10 - [info] Server now running at http://127.0.0.1:1880/
5 Jun 12:48:10 - [info] Starting flows
5 Jun 12:48:10 - [info] Started flows
```

**Figura 10. Consola Node-red.**

Fuente: Infante y González (2017).

Para poder acceder a la interfaz gráfica de edición de flujo, se abre el navegador (en este trabajo de grado se decide utilizar Firefox, ya que es el recomendado por la empresa desarrolladora de Node-red), en la barra del URL del navegador Firefox, se coloca la URL donde dice Node-red que se inicializo, en este caso se coloca *https://192.168.10.104:1880* y se puede acceder al editor de flujo.

Node-red posee nodos instalados por defecto, sin embargo, en esta investigación se necesitan librerías extras, por lo que se procede a su instalación, la cual puede ser realizada de dos maneras distintas, por ventana terminal o por el entorno de desarrollo de flujo. En este caso se utilizó la segunda opción.

Para instalar las librerías por el entorno de desarrollo de flujo se llevaron a cabo los siguientes pasos:

1. Se abrió el entorno de desarrollo ingresando la URL:

*https://192.168.10.104:1880*

2. En la esquina superior derecha se encuentra un menú desplegable, donde se selecciona la opción de “Manage Palette”, ésta habilitara un menú con dos opciones, la primera “Nodes”, donde se muestra todos los nodos instalados en el Node-red, la otra ventana es “Install”, en la cual se coloca el nombre del nodo que se busca y se mostrará una serie de nodos disponibles.
3. Se selecciona el nodo de interés para instalar, se da click en install y se espera a que finalice la instalación.

#### **4.3.4 Flask y Flask-ask.**

Flask es un framework web, en Python y Flask-ask es una extensión del mismo, que facilita la elaboración de skill de Alexa en el entorno de programación Python, ambos necesarios en la realización de algunas de las skills de Alexa. Para la instalación de este software, bastó con saber que versión de Python se utilizará (en este caso Python 2.7.9) y escribir en la ventana terminal los siguientes comandos:

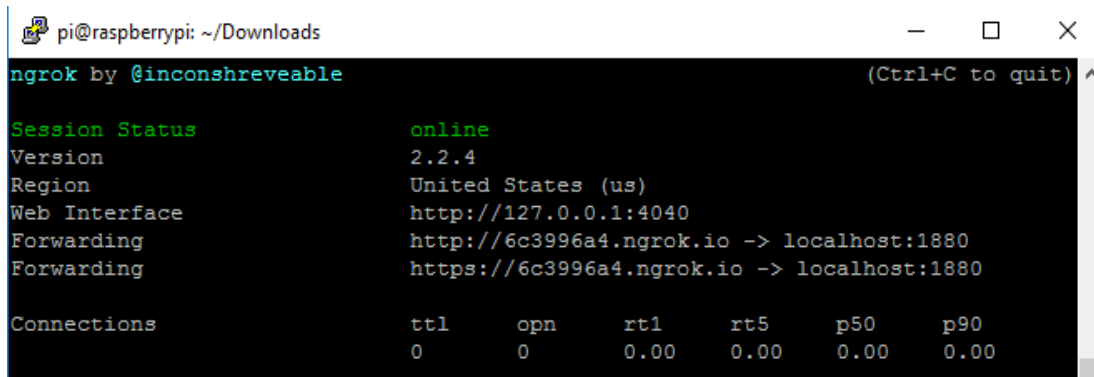
*Sudo apt-get install python-flask*

*Sudo pip install flask flask-ask*

### 4.3.5 Ngrok.

Ngrok es una herramienta que permite que se acceda a un servidor local a través de un URL generada dinámicamente. Ngrok realiza esto sin hacer ninguna configuración extra en el router o firewall, simplemente bastó con bajar la aplicación en página oficial de Ngrok (la versión Linux ARM que es la apropiada para Raspberry Pi). Para ejecutarlo se escribió en una ventana terminal del Raspberry Pi el siguiente comando (ver figura 11):

```
./ngrok http 1880
```



```
pi@raspberrypi: ~/Downloads
ngrok by @inconshreveable (Ctrl+C to quit)
Session Status      online
Version             2.2.4
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://6c3996a4.ngrok.io -> localhost:1880
Forwarding           https://6c3996a4.ngrok.io -> localhost:1880
Connections
  ttl    opn    rt1    rt5    p50    p90
  0      0      0.00  0.00  0.00  0.00
```

**Figura 11. Ngrok ejecutado en el Raspberry Pi.**

Fuente: Infante y González (2017).

### 4.3.6 IDE Arduino.

Es el entorno de desarrollo para Arduino; en esta sección se pretende hacer del Raspberry Pi una plataforma sobre la cual se llevan a cabo los diferentes proyectos de Arduino realizados en este trabajo de grado. En pocas palabras el Raspberry Pi, viene siendo el cerebro del prototipo y el Arduino el ejecutor de algunas de las skills de Alexa. Para lograr esto necesitamos que el desarrollador IDE sea instalado en el Raspberry Pi y así desde el miniordenador manipular la programación del Arduino. Para la instalación del entorno de desarrollo se debió:

1. Escribir en la ventana terminal el comando:

```
sudo apt-get install Arduino
```

2. Se verifica que se haya reconocido la tarjeta Arduino mediante el código:

```
dmesg | grep ttyACM
```

3. Por último, se realizó la configuración para que se pueda usar el ttyACM0, el cual es el puerto de comunicación entre el Arduino y el Raspberry Pi, mediante el código:

```
sudo chmod 666 /dev/ttyACM0
```

#### **4.4 Fase IV. Desarrollo del prototipo de Robot Social.**

En esta fase se desarrollan las diferentes habilidades del prototipo de robot social a través de la creación de skills para Alexa, así como otras habilidades que se desarrollan en los mismos entornos de programación, pero sin depender de la misma. Antes de empezar a explicar cada una de estas habilidades se debe destacar que en este trabajo de grado se utiliza una herramienta propia de Alexa para la creación de nuevas skills, esta herramienta es el Alexa Skill Kit (constructor de habilidades de Alexa). Existen tres grupos de skills, las cuales se definen a continuación:

- 1. Custom Skills:** el usuario define la petición del skill (conocido como “Intents”) y las palabras que usará para invocar dicha petición (conocidas como “Utterances”) estas habilidades pueden utilizarse para buscar información en un servicio web o integrar con un servicio web.
- 2. Smart Home Skill API:** son skills que permiten al usuario controlar sus dispositivos inteligentes enlazados con la nube, por ejemplo, luces y termostatos. También funcionan mediante “Utterances”, pero a diferencia de la clasificación de habilidad anterior esta no se maneja mediante peticiones, sino que asigna direcciones a cada dispositivo inteligente.
- 3. Flash Briefing Skill API:** proveen intercambio rápido de información y otro tipo de contenido como entrevistas y chistes.

En este trabajo se desarrollaron varias Skills de tipo “Custom Skills”, las cuales se explicarán de forma detallada a continuación:

#### **4.4.1 Programación de Skill para el envío de señales eléctricas por petición hablada del usuario**

La siguiente skill tiene el propósito de activar o desactivar las salidas del Arduino por medio de una petición hablada del usuario, otorgando la posibilidad de manipular justo la salida que el usuario solicita; a modo de demostración, se utilizaron leds que demuestran que la salida del Arduino otorga una señal de manera exitosa. Estos leds pueden ser sustituidos por bombillas, por motores DC o por cualquier dispositivo que se necesite de su activación o desactivación en un momento determinado, claro está que el Arduino entrega por sus salidas 5V en este caso, así que habría que hacer algunos arreglos en la circuitería para adaptar estos elementos.

Antes de explicar el desarrollo de esta skill es importante destacar que el Arduino y el Raspberry se encuentran conectados por comunicación serial, se hace esta salvedad debido a que es un punto muy importante a considerar para comenzar la programación de las skills. Tomando en cuenta lo anterior comienza a programar en el entorno Arduino como se muestra y explica en la figura 12.

Una vez terminada la programación en el entorno Arduino, se procede a la programación del script en Python, utilizando la librería de Flask. Como se muestra en la figura 13.

Cuando ya se tienen estos dos códigos se procede a ejecutar el script en Python, en una ventana terminal del Raspberry Pi, tal y como se muestra en la figura 14. Como se puede observar en la figura anterior, se inicializa en el puerto 5000, por lo tanto, en otra ventana terminal, inicializamos el Ngrok, en el puerto 5000 como se muestra en la figura 15.

Al término de este proceso, se va al desarrollador de Amazon, dentro de este se encuentran varias pestañas, se selecciona la pestaña de Alexa que se redirige a un menú como el mostrado en la figura 16. Se selecciona la opción de Alexa Skills Kit, el cual es el servicio de desarrollo de Skills proporcionado por Amazon que ya se ha mencionado en reiteradas oportunidades.



```
Pruebaled Arduino 1.6.12
Archivo Editar Programa Herramientas Ayuda
Pruebaled
int led = 13;
int led1 = 12;
int led2 = 8;
int led3 = 7;

void setup () {
  pinMode(led, OUTPUT); //LED 13 como salida
  pinMode(led1, OUTPUT); //LED 12 como salida
  pinMode(led2, OUTPUT); //LED 8 como salida
  pinMode(led3, OUTPUT); //LED 7 como salida

  Serial.begin(9600); //Inicializo el puerto serial a 9600 baudios
}

void loop () {
  if (Serial.available()) { //Si está disponible
    char c = Serial.read(); //Guardamos la lectura en una variable char
    if (c == 'Q') { //Si es una 'Q', enciendo el LED
      digitalWrite(led, HIGH);
      Serial.println("Led encendido");
    } else if (c == 'A') { //Si es una 'A', apago el LED
      digitalWrite(led, LOW);
      Serial.println("Led apagado");
    } else if (c == 'W') { //Si es una 'W', enciendo el LED1
      digitalWrite(led1, HIGH);
      Serial.println("Led1 encendido");
    } else if (c == 'S') { //Si es una 'S', apago el LED1
      digitalWrite(led1, LOW);
      Serial.println("Led1 apagado");
    } else if (c == 'E') { //Si es una 'E', enciendo el LED2
      digitalWrite(led2, HIGH);
      Serial.println("Led2 encendido");
    } else if (c == 'D') { //Si es una 'D', apago el LED2
      digitalWrite(led2, LOW);
      Serial.println("Led2 apagado");
    } else if (c == 'R') { //Si es una 'R', enciendo el LED3
      digitalWrite(led3, HIGH);
      Serial.println("Led3 encendido");
    } else if (c == 'F') { //Si es una 'F', apago el LED3
      digitalWrite(led3, LOW);
      Serial.println("Led3 apagado");
    } else if (c == 'Z') { //Si es una 'Z', enciendo todos los Leds
      digitalWrite(led, HIGH);
      digitalWrite(led1, HIGH);
      digitalWrite(led2, HIGH);
      digitalWrite(led3, HIGH);
      Serial.println("Todos los Led encendido");
    } else if (c == 'X') { //Si es una 'X', apago el LED
      digitalWrite(led, LOW);
      digitalWrite(led1, LOW);
      digitalWrite(led2, LOW);
      digitalWrite(led3, LOW);
      Serial.println("Todos los leds apagados");
    }
  }
}
```

**Figura 12. Programación en el Entorno Arduino de Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).

```

from flask import Flask
from flask_ask import Ask, statement
import requests
import json
import serial

ser = serial.Serial("/dev/ttyACM0", 9600)

app = Flask(__name__)
ask = Ask(app, '/')

@ask.launch

@ask.intent("LightGOD")
def on():
    ser.write(b'Z')
    return statement("All lights turned on")

@ask.intent("LightMaster")
def on():
    ser.write(b'Q')
    return statement("Master room light turned on")

@ask.intent("LightLiving")
def on():
    ser.write(b'W')
    return statement("Living light turned on")

@ask.intent("LightBed")
def on():
    ser.write(b'E')
    return statement("bedroom light turned on")

@ask.intent("LightGarage")
def on():
    ser.write(b'R')
    return statement("Garage light turned on")

@ask.intent("LightMasterOff")
def on():
    ser.write(b'A')
    return statement("master light turned off")

@ask.intent("LightLivingOff")
def on():
    ser.write(b'S')
    return statement("living light turned off")

@ask.intent("LightBedOff")
def on():
    ser.write(b'D')
    return statement("Bed light turned off")

@ask.intent("LightHolloff")
def on():
    ser.write(b'F')
    return statement("Garage light turned off")

```

**Figura 13. Programación en Python de Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).

```
pi@raspberrypi: ~/led
pi@raspberrypi:~ $ cd /home/pi/led
pi@raspberrypi:~/led $ python led3.py
* Running on http://127.0.0.1:5000/
* Restarting with reloader
```

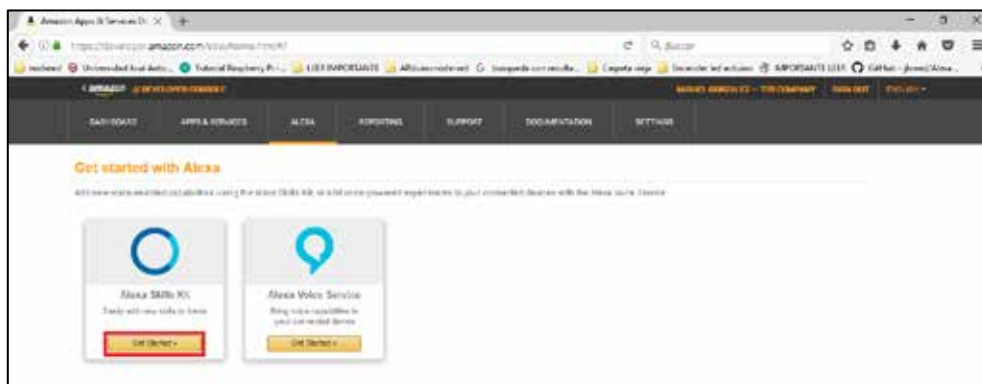
**Figura 14. Ejecución en Python para la Skill de envío de señales eléctricas.**  
Fuente: Infante y González (2017).

```
pi@raspberrypi: ~/Downloads
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Version             2.2.4
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://2f5c69bd.ngrok.io -> localhost:5000
Forwarding           https://2f5c69bd.ngrok.io -> localhost:5000

Connections
t1l   opn   rt1   rt5   p50   p90
0     0     0.00 0.00 0.00 0.00
```

**Figura 15. Ejecución de Ngrok para la Skill de envío de señales eléctricas.**  
Fuente: Infante y González (2017).



**Figura 16. Menú de Amazon Developer.**  
Fuente: Infante y González (2017).

Una vez dentro, se ve un menú en el que se muestran todas las Skills desarrolladas por el usuario (Ver figura 17). Para añadir una nueva Skill, se da click en el botón ubicado a la derecha que dice “Add a New Skill”.

Se redirigirá a una página, llamada Skill “Information” como la mostrada en la figura 18, donde se colocó el nombre de la skill, la palabra de invocación de la skill y otros datos relevantes como el idioma en el que se desenvolverá la skill, el ID de la skill y el tipo de Skill que se va a desarrollar.

**Building Alexa Skills with the Alexa Skills Kit** Add a New Skill

To learn more about building Alexa skills, see [Getting Started with the Alexa Skills Kit](#). To start building an Alexa skill for free using AWS Lambda, see [Creating an AWS Lambda Function for a Custom Skill](#). We encourage you to join the [Alexa Developer Forum](#) to collaborate with Alexa team members and fellow Alexa developers.

Name	Language	Type	Modified	Status	Actions
light room <a href="#">View Skill ID</a>	English (U.S.)	Custom	6/4/17	Development	<a href="#">Metrics</a> <a href="#">Edit</a> <a href="#">Delete</a>
Hello World <a href="#">View Skill ID</a>	English (U.S.)	Custom	6/4/17	Development	<a href="#">Metrics</a> <a href="#">Edit</a> <a href="#">Delete</a>
Hello all <a href="#">View Skill ID</a>	English (U.S.)	Custom	3/31/17	Development	<a href="#">Metrics</a> <a href="#">Edit</a> <a href="#">Delete</a>
Memory Game <a href="#">View Skill ID</a>	English (U.S.)	Custom	5/24/17	Development	<a href="#">Metrics</a> <a href="#">Edit</a> <a href="#">Delete</a>

**Figura 17. Listado de Skills del usuario.**

Fuente: Infante y González (2017).

English (U.S.) + Add a New Language

- Skill Information**
- Interaction Model**
- Configuration**
- SSL Certificate**
- Test**
- Publishing Information**
- Privacy & Compliance**
- Skills Beta Testing**

**Skill Type**  
Define a custom interaction model or use one of the preferred skill APIs. [Learn More](#)

**Language**  
Language of your skill: English (U.S.)

**Application ID**  
The ID for this skill: amzn1.ask.sll.a048023c-6ae0-421c-b68c-23a04fa785ac

**Name**  
Name of the skill that is displayed to customers in the Alexa app. Must be between 1-30 characters.

**Invocation Name**  
The name customers use to activate the skill. For example, "Alexa ask This Player."

**Global Fields**  
These fields apply to all languages supported by the skill.

**Audio Player**  
Does this skill use the audio player interface?  Yes  No [Learn More](#)

**For successful Alexa Skills Certification, please review and follow our [Invocation Name Guidelines](#) as well as our [Certification Requirements](#).**

**Figura 18. Skill Information para la Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).

Luego de llenar los datos correspondientes a la Skill se irá a la ventana del “Interaction Model”, donde se llenarán los campos de interacción, y se verá el cuadro

de “Intent Schema”, el cual debe ser programado en formato JSON, como se muestra a continuación en la figura 19.

```
Intent Schema
The schema of user intents in JSON format. For more information, see Intent Schema.
Also see built-in slots and built-in intents.

1 {
2   "intents": [
3     {
4       "intent": "LightGOD"
5     },
6     {
7       "intent": "LightMaster"
8     },
9     {
10      "intent": "LightLiving"
11    },
12    {
13      "intent": "LightBed"
14    },
15    {
16      "intent": "LightGarage"
17    },
18    {
19      "intent": "LightMasterOff"
20    },
21    {
22      "intent": "LightLivingOff"
23    },
24    {
25      "intent": "LightBedOff"
26    },
27    {
28      "intent": "LightGarageOff"
29    },
30    {
31      "intent": "LightGODOff"
32    }
33  ]
34 }
```

**Figura 19. Intent Schema para la Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).

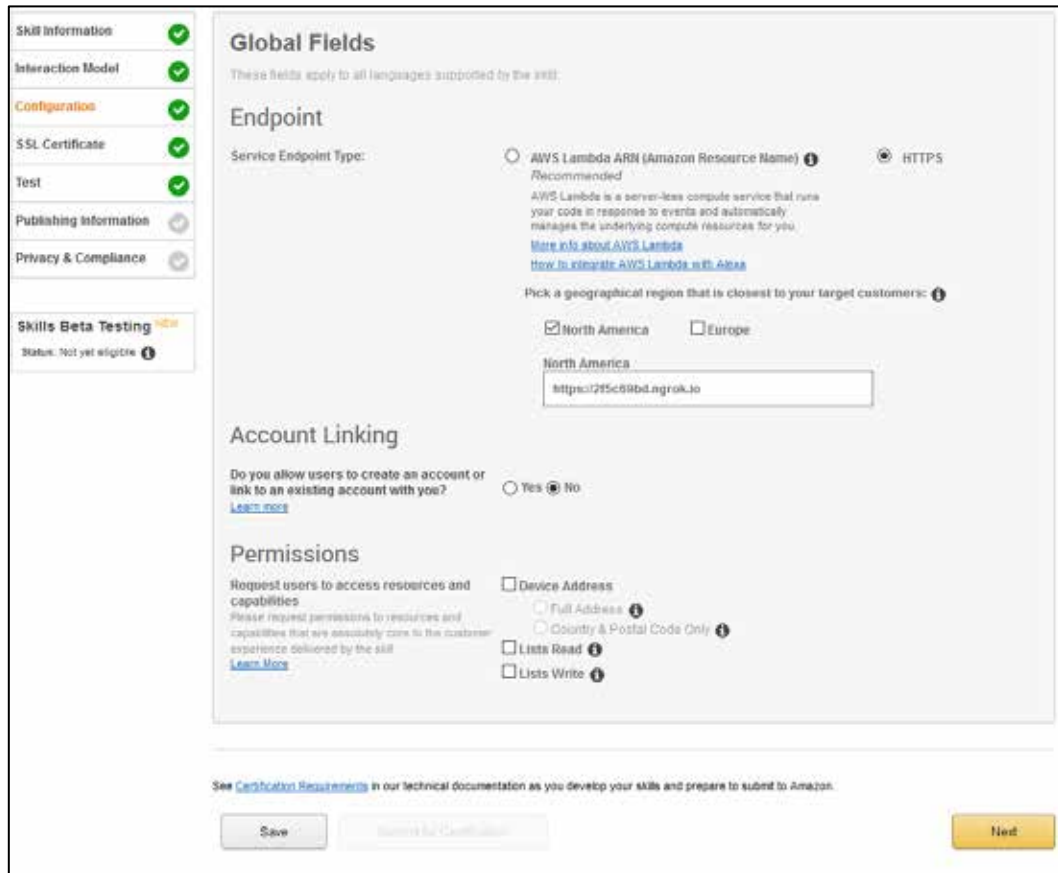
Por último, se verá el cuadro de “Utterances”, el cual se llenará con las posibles combinaciones de palabras que se utilizaran para activar la skill, como se muestra en la figura 20. Se da click en el botón “Next”, luego de un proceso de compilado se envía a la página de “Configuration”, en el apartado “Endpoint”, se selecciona HTTPS, se coloca el check mark en “North America” y enseguida se habilitará un espacio para colocar el HTTPS, el cual es la dirección proporcionada por el Ngrok (Ver figura 21).

En la sección de “SSL Certificate” se selecciona la opción que se muestra en la figura 22.

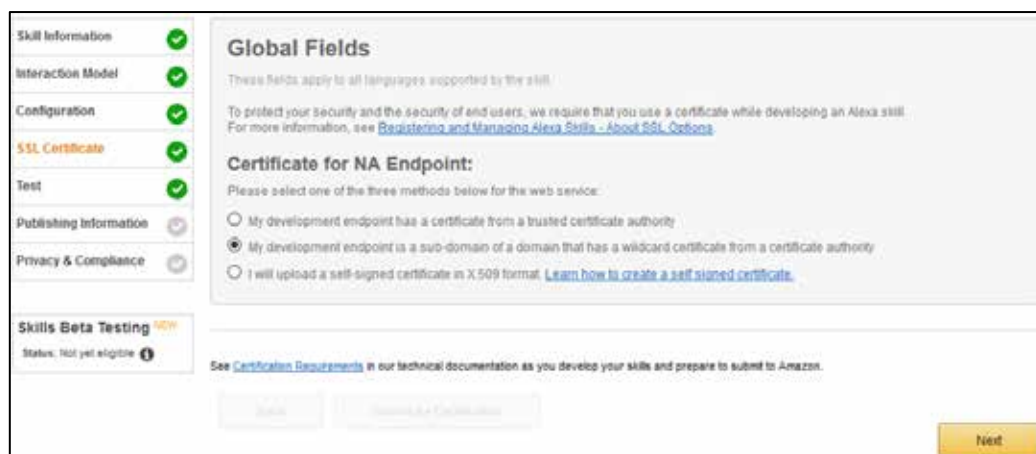
Sample Utterances	
These are what people say to interact with your skill. Type or paste in all the ways that people can invoke the intents. <a href="#">Learn more</a>	
Up to 3 of these will be used as Example Phrases, which are hints to users.	
1	LightGOD turn on all lights
2	LightGOD on all lights
3	LightGOD turn on all light
4	LightGOD on all light
5	LightMaster turn on master lights
6	LightMaster on master lights
7	LightMaster turn on master light
8	LightMaster on master light
9	LightLiving turn on living lights
10	LightLiving on living lights
11	LightLiving turn on living light
12	LightLiving on living light
13	LightBed turn on bed lights
14	LightBed on bed lights
15	LightBed turn on bed light
16	LightBed on bed light
17	LightGarage turn on garage lights
18	LightGarage on garage lights
19	LightGarage turn on garage light
20	LightGarage on garage light
21	LightMasterOff turn off master lights
22	LightMasterOff off master lights
23	LightMasterOff turn off master light
24	LightMasterOff off master light
25	LightLivingOff turn off living lights
26	LightLivingOff off living lights
27	LightLivingOff turn off living light
28	LightLivingOff off living light
29	LightBedOff turn off bed lights
30	LightBedOff off bed lights
31	LightBedOff turn off bed light
32	LightBedOff off bed light
33	LightGarageOff turn off garage lights
34	LightGarageOff off garage lights
35	LightGarageOff turn off garage light
36	LightGarageOff off garage light
37	LightGODOff turn off all lights
38	LightGODOff off all lights
39	LightGODOff turn off all light
40	LightGODOff off all light

**Figura 20. Utterances para la Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).

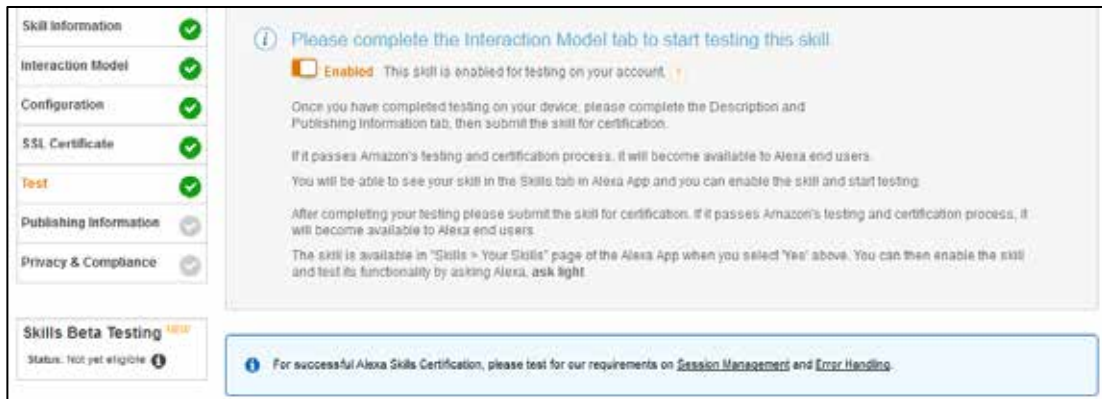


**Figura 21. Ventana “Configuration” para la Skill de envío de señales eléctricas.**  
Fuente: Infante y González (2017).



**Figura 22. Ventana “SSL Certificate” para la Skill de envío de señales eléctricas.**  
Fuente: Infante y González (2017).

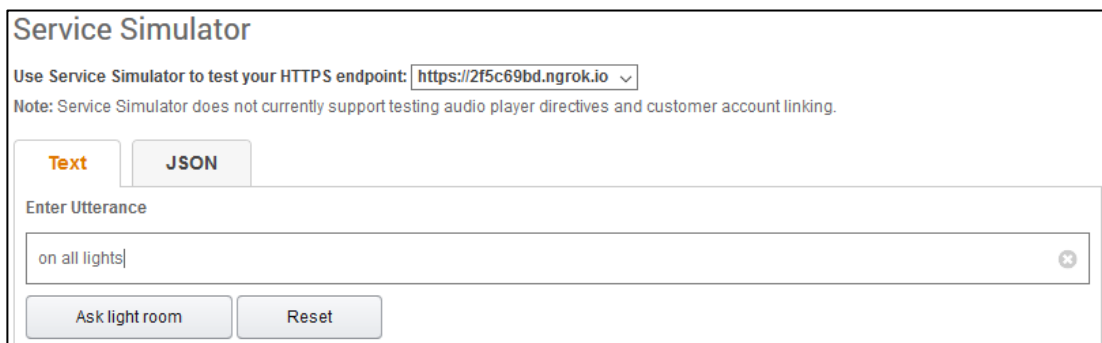
En la siguiente sección, primero se debe habilitar el botón de pruebas, como se muestra en la Figura 23.



**Figura 23. Ventana “Test” para la Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).

Después, en esa misma sección “Test”, se encontrará el servicio de simulación que la empresa de Amazon facilita; donde se tendrá una sección para colocar las Utterance que se desea probar, una vez colocada, se da click en “Ask {Nombre De la Skill}” y verá su comportamiento, en el cuadro a la izquierda, se ve como se hace la solicitud al servicio Alexa, en el cuadro a la derecha se ve la respuesta que otorga dicho servicio, como se muestra en la figura 24 y 25.



**Figura 24. Service Simulator para la Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).

```

Service Request
1 {
2   "session": {
3     "sessionId": "SessionId.e0e45188-de77-49bd",
4     "application": {
5       "applicationId": "amzn1.ask.skill.a0e8f2
6     },
7     "attributes": {},
8     "user": {
9       "userId": "amzn1.ask.account.AH3ZHNX6JYL
10    },
11    "new": true
12  },
13  "request": {
14    "type": "IntentRequest",
15    "requestId": "EdwRequestId.86528a69-6e83-4
16    "locale": "en-US",
17    "timestamp": "2017-06-05T19:58:40Z",
18    "intent": {
19      "name": "LightGOD",
20      "slots": {}
21    }
22  },
23  "version": "1.0"
24 }

Service Response
1 {
2   "version": "1.0",
3   "response": {
4     "outputSpeech": {
5       "type": "PlainText",
6       "text": "All lights turned on"
7     },
8     "shouldEndSession": true
9   },
10  "sessionAttributes": {}
11 }

```

**Figura 25. Service Simulator para la Skill de envío de señales eléctricas II.**

Fuente: Infante y González (2017).

Una vez hecha la prueba, se puede comprobar que realmente existe una comunicación efectiva entre Alexa y el servidor HTTPS proporcionado por Ngrok, debido a que se ve un intercambio de información en la ventana de ejecución del Ngrok, como se muestra en la figura 26.

```

pi@raspberrypi: ~/Downloads
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Version             2.2.4
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://2f5c69bd.ngrok.io -> localhost:5000
                    https://2f5c69bd.ngrok.io -> localhost:5000

Connections
-----
t1    opn    rt1    rt5    p50    p90
1     0      0.00  0.00  0.64  0.64

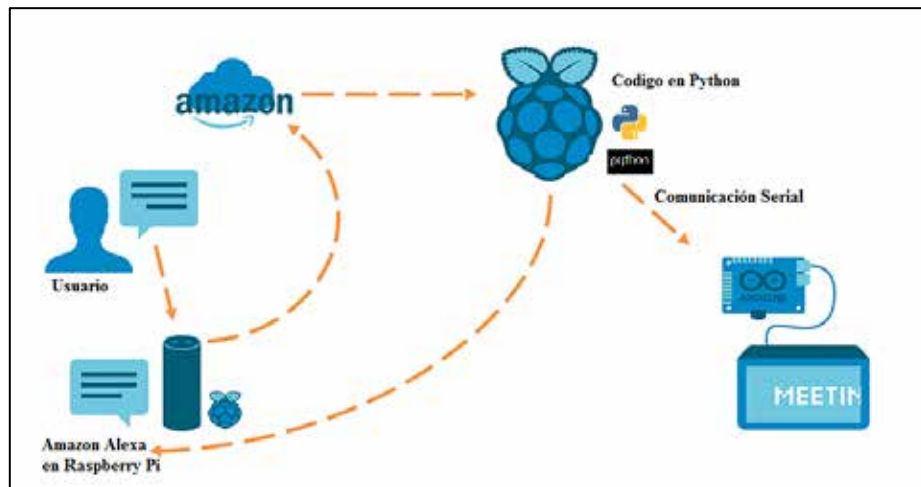
HTTP Requests
-----
POST /                200 OK

```

**Figura 26. Respuesta en la ventana terminal de Ngrok para la Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).

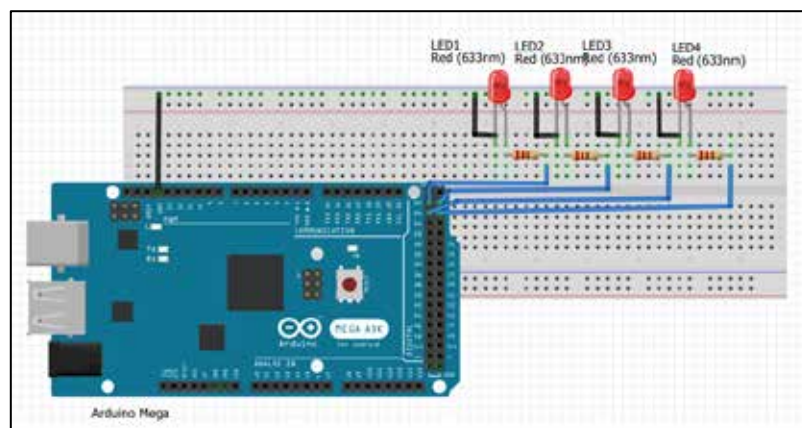
Para un mayor entendimiento de la comunicación que existe entre los programas utilizados para la Skill desarrollada se realizó un diagrama de comunicación presentado en la figura 27.



**Figura 27. Diagrama de Comunicación para la Skill de envío de señales eléctricas.**

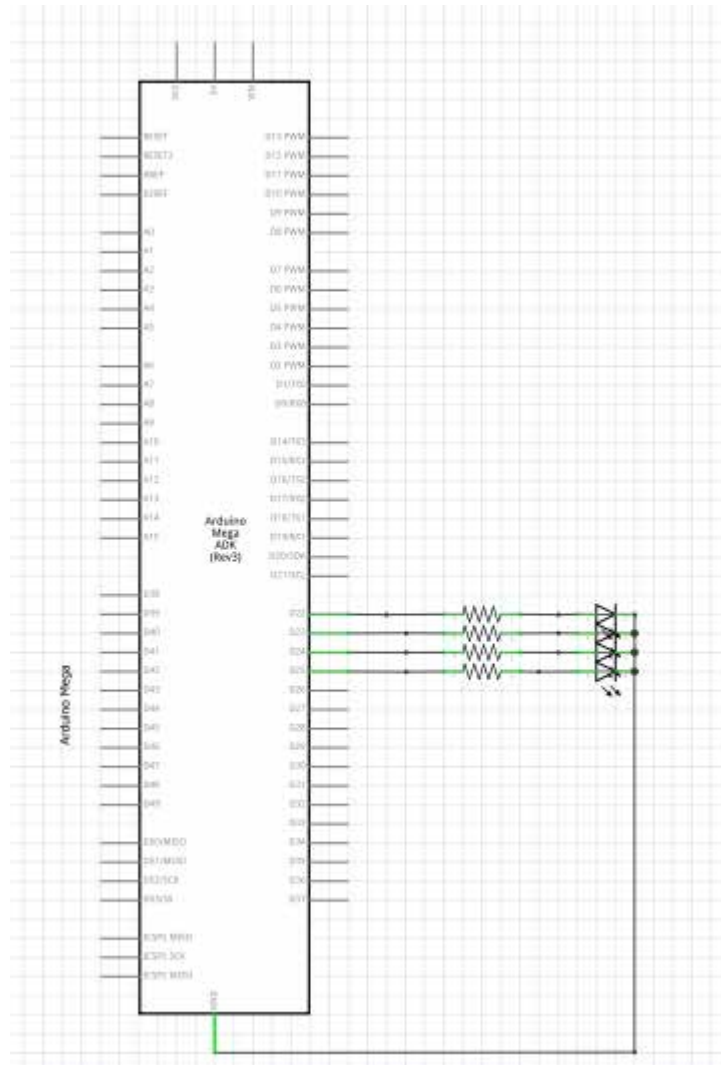
Fuente: Infante y González (2017).

En las figuras 28 y 29 se anexa el diagrama de conexión y el diagrama esquemático, del montaje en físico para la demostración del funcionamiento de la Skill recientemente desarrollada.



**Figura 28. Diagrama de Conexión para la Skill de envío de señales eléctricas.**

Fuente: Infante y González (2017).



**Figura 29. Diagrama Esquemático para la Skill de envío de señales eléctricas.**  
 Fuente: Infante y González (2017).

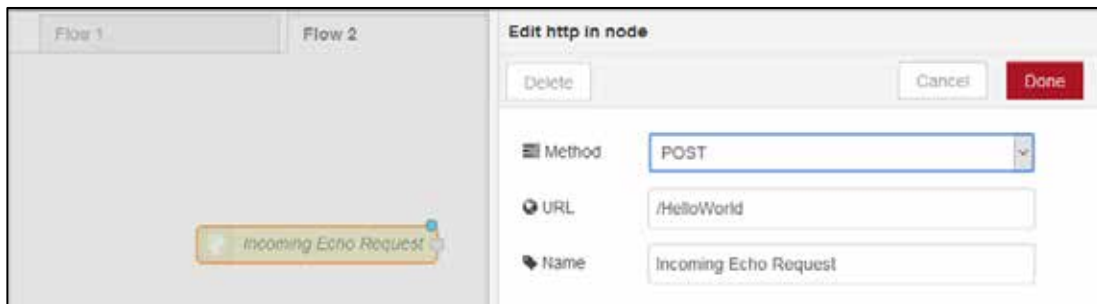
#### 4.4.2 Programación para la Skill de interacción Presentation

La siguiente skill, tiene la finalidad de tener una pequeña interacción hablada con el asistente Alexa, demostrando que se le puede configurar para que dé una respuesta hablada que se quiera a través de una petición o un mandato también hablado de parte del usuario.

Para la programación de esta skill, se utilizó Node-red, al igual que la programación anterior, este desarrollo tiene dos fases: la fase del desarrollo en Node-

red y la fase del desarrollo en el entorno de Amazon, para comenzar el desarrollo de la skill, se abre una nueva ventana terminal donde se inicializa Node-red y se accede al mismo por medio de la URL proporcionada.

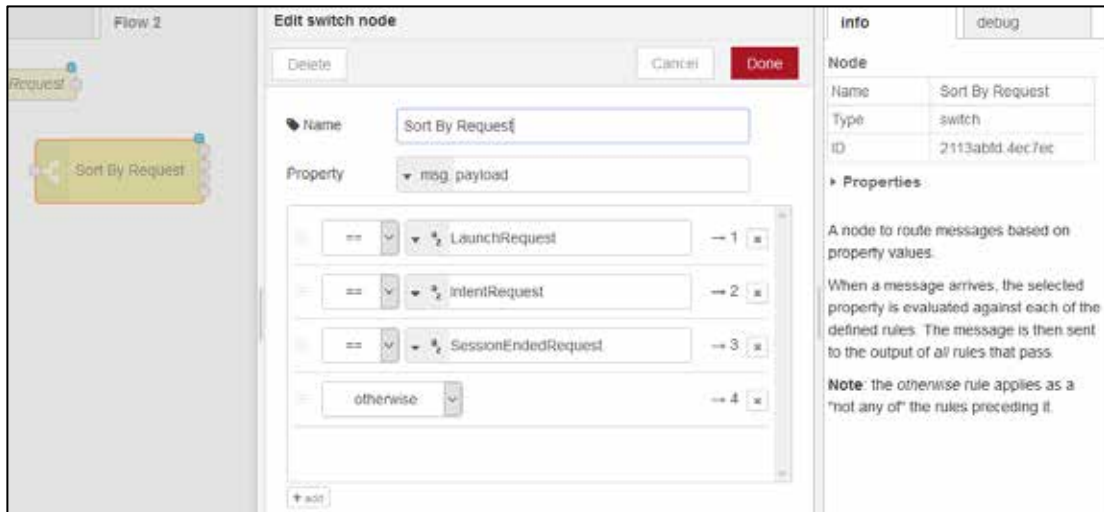
Para comenzar, se despliega el menú de los nodos de tipo “input” donde se selecciona un nodo del tipo http, para configurarlo se da doble click en el mismo, se le coloca un método de trabajo en el espacio de “Method”, el terminal del URL que estará asociado /Presentation y el nombre de etiqueta de dicho nodo, como se muestra en la figura 30 para finalizar se da click en el botón “Done”



**Figura 30. Nodo https para la Skill Presentation.**

Fuente: Infante y González (2017).

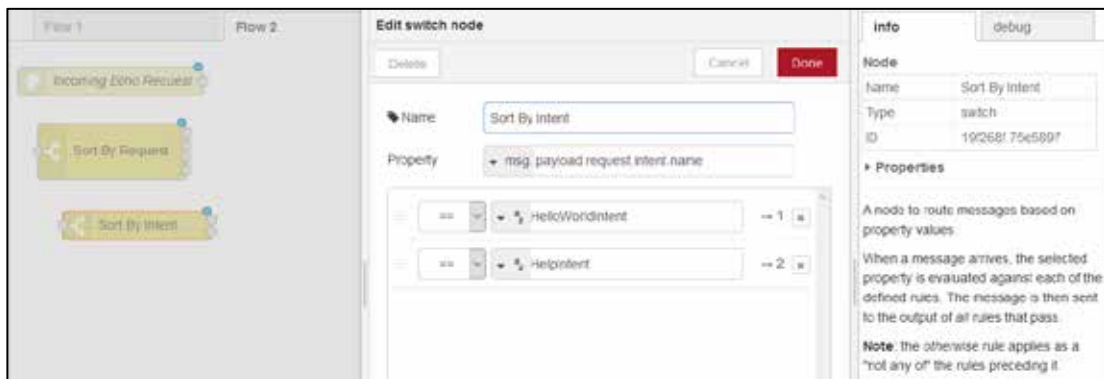
En la sección “Function” se selecciona un nodo del tipo “Switch” el cual se encarga de evaluar una propiedad seleccionada contra cada una de las propiedades definidas en cuanto el mensaje llegue, se configura dando doble click y se rellenan sus campos como se muestra en la figura 31, para cada propiedad definida, se crea en la salida del nodo un punto de salida.



**Figura 31. Nodo Switch para la Skill Presentation.**

Fuente: Infante y González (2017).

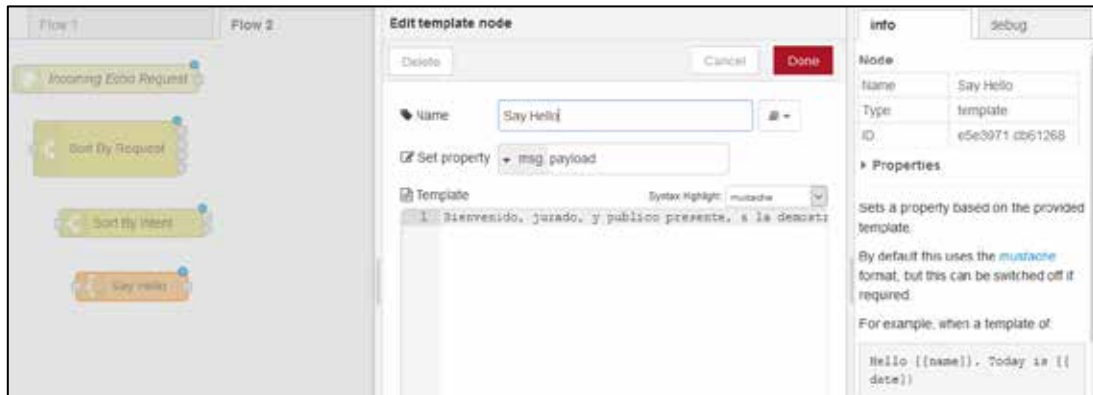
Para el siguiente paso, en la misma sección “Function” se selecciona otro nodo del tipo “Switch” ahora siendo su configuración interna distinta del anterior nodo switch, como se muestra en la figura 32.



**Figura 32. Nodo Switch.**

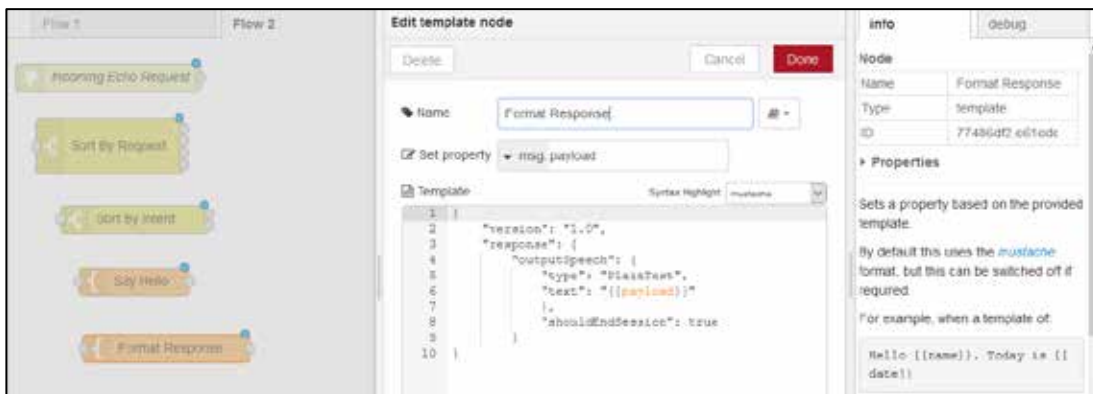
Fuente: Infante y González (2017).

Para continuar, se utiliza un nodo llamado “Template” que es el encargado de recibir el Intent del nodo Switch del paso anterior, en este nodo se carga el mensaje que será otorgado por Alexa al usuario, como se muestra en la figura 33.



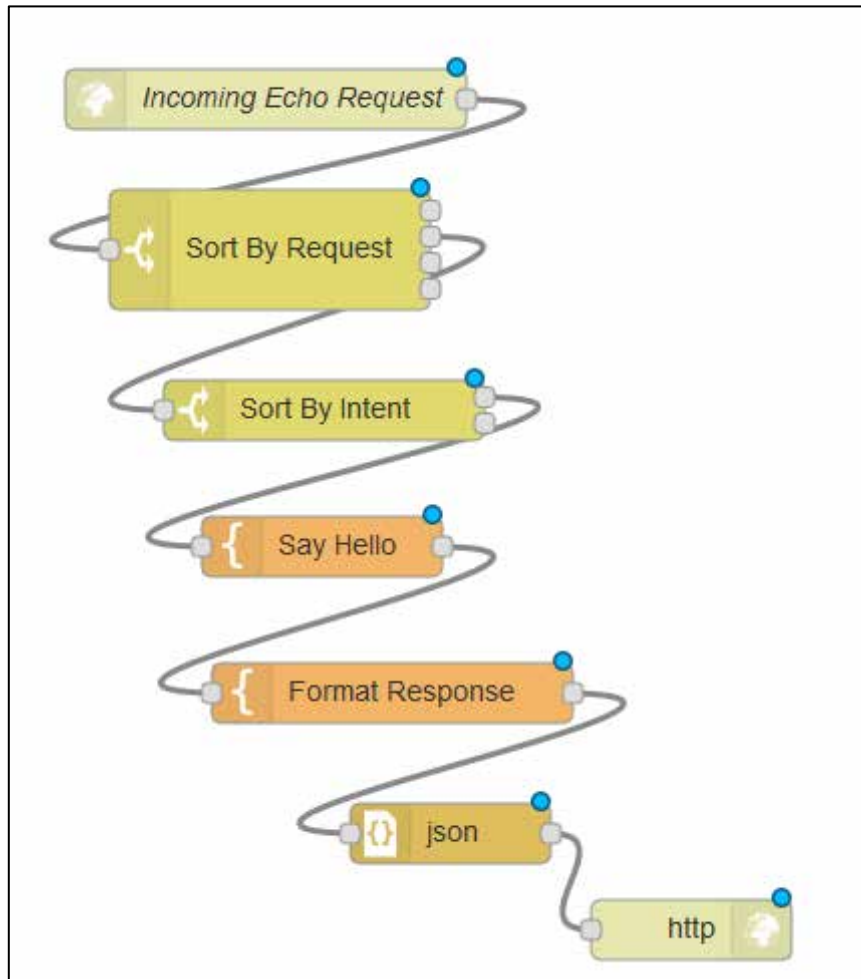
**Figura 33. Nodo Template para la Skill Presentation I.**  
Fuente: Infante y González (2017).

A continuación, se recurre nuevamente a un nodo “Template”, donde se va a colocar el formato de respuesta, se puede observar en la figura 34.



**Figura 34. Nodo Template para la Skill Presentation II.**  
Fuente: Infante y González (2017).

Ahora, utilizando un nodo “JSON” se convierte el mensaje de una cadena de caracteres JSON a un objeto en JavaScript, para finalizar, se utiliza un nodo función “http response” que no es más que el cierre del nodo de apertura “http”, cabe acotar que los nodos deben enlazarse entre sí y armar una ramificación la cual será la que dará cuerpo a toda la skill, como se ve en la figura 35.



**Figura 35. Flujo en Node-red para la Skill Presentation.**

Fuente: Infante y González (2017).

Terminada la configuración de la skill por el lado de node-red, se procede a la configuración en el desarrollador de Amazon, bastante similar a la configuración mostrada en la skill anterior, con las diferencias en el nombre de la skill, la palabra de invocación, el “Intent Schema”, los “Utterance” y el URL de la sección “Configuration”, como se muestra en las figuras 36-39.

**Skill Type**  
Define a custom interaction model or use one of the predefined skill APIs. [Learn more](#) Custom

**Language**  
Language of your skill English (U.S.)

**Application Id**  
The ID for this skill amzn1.ask.skill.806f3715-47b5-46c7-b0d2-58c7f3f18738

**Name**  
Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters. presentation

**Invocation Name**  
The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...". presentation

**Figura 36. Skill Information para la Skill Presentation.**

Fuente: <https://developer.amazon.com> (2017).

**Intent Schema**  
The schema of user intents in JSON format. For more information, see [Intent Schema](#). Also see [built-in slots](#) and [built-in intents](#).

```

1 {
2   "intents": [
3     {
4       "intent": "HelloWorldIntent"
5     },
6     {
7       "intent": "AMAZON.HelpIntent"
8     }
9   ]
10 }
```

**Figura 37. Intent Schema para la Skill Presentation.**

Fuente: <https://developer.amazon.com> (2017).

**Sample Utterances**  
These are what people say to interact with your skill. Type or paste in all the ways that people can invoke the intents. [Learn more](#)

Up to 3 of these will be used as Example Phrases, which are hints to users.

```

1 HelloWorldIntent say hello world
2 HelloWorldIntent hello
3 HelloWorldIntent say hi
4 HelloWorldIntent say hi world
5 HelloWorldIntent hi
6 HelloWorldIntent how are you
```

**Figura 38. Ventana "Utterances" para la Skill Presentation.**

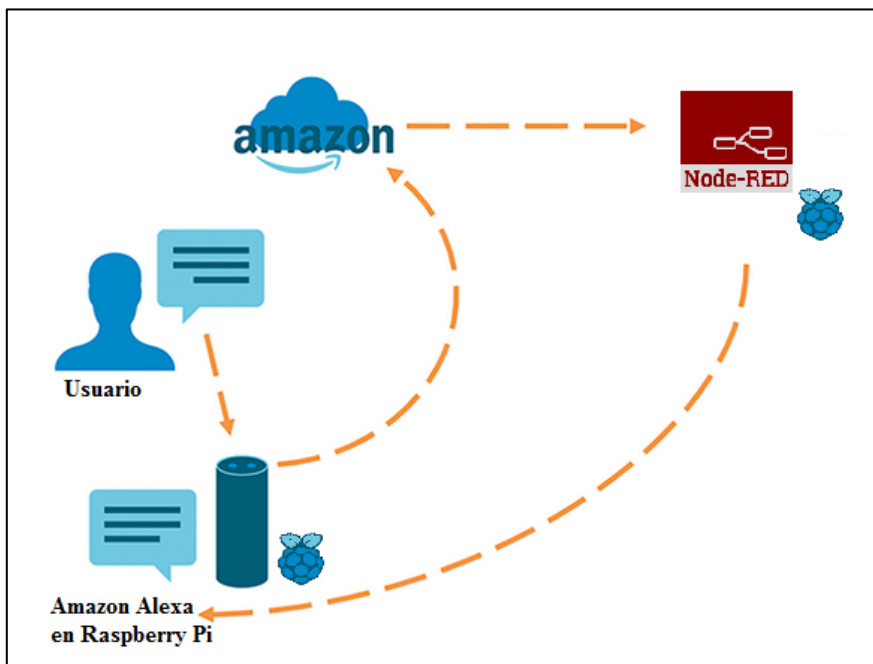
Fuente: <https://developer.amazon.com> (2017).



**Figura 39. Ventana “Configuration” para la Skill Presentation.**

Fuente: <https://developer.amazon.com> (2017).

Para finalizar se muestra el diagrama de comunicación que existe entre los programas que intervienen para la realización de esta Skill (ver figura 40).




**Figura 40. Diagrama de Comunicación para la Skill Presentation.**

Fuente: Infante y González (2017).

#### 4.4.3 Programación para la Skill de Seguridad.

Con la realización de esta Skill, se pretende demostrar que se pueden añadir sensores al prototipo de Robot Social, que complemente su correcto funcionamiento y funcionalidad. La Skill a desarrollar se plantea con la intención de que dicho prototipo pueda avisar al usuario de que un sensor se activó, por medio de un mensaje de texto, un correo o por otra vía de comunicación, y así el usuario pueda conocer el estado de dicho sensor, hablándolo de otra forma, se desea crear una Skill con un sensor infrarrojo de movimiento utilizado como dispositivo de seguridad instalado en el hogar u otro entorno que se desee monitorear. Para lograr esta meta, se comenzó inicializando el IDE Arduino en el que se escribió el código correspondiente a dicho sensor, como se muestra en la figura 41.



```
sensorpir

int ledPin = 53;           // Seleccionar el Pin para el LED
int inputPin = 52;        // Seleccionar el modo como entrada (Sensor PIR como entrada)
int pirState = LOW;       // se inicia asumiendo "Motion no detected!"
int val = 0;              // variable para leer el estado del pin

void setup() {
  pinMode(ledPin, OUTPUT); // declara LED como salida
  pinMode(inputPin, INPUT); // declara sensor como entrada

  Serial.begin(9600);
}

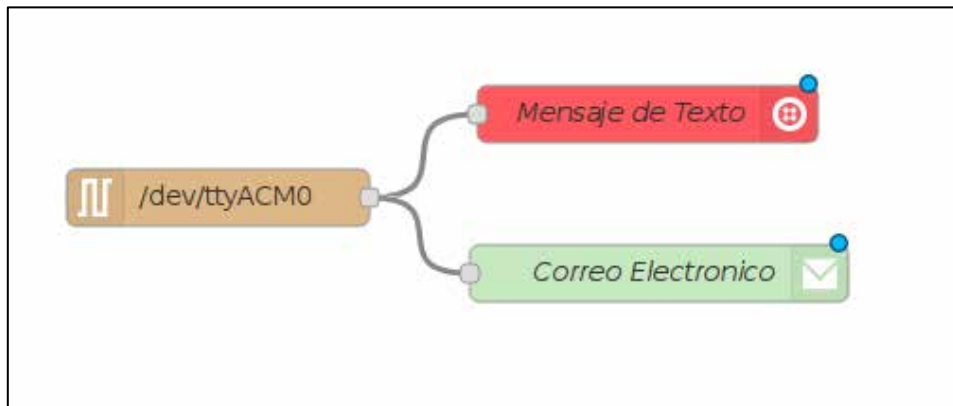
void loop(){
  val = digitalRead(inputPin); // leer el valor de entrada
  if (val == HIGH) {          // chequea si el valor de entrada es HIGH
    digitalWrite(ledPin, HIGH); // enciende el LED
    if (pirState == LOW) {
      // lo encendimos
      Serial.println("Motion detected!");
      //Solo se imprime el cambio de la salida, no el estado
      pirState = HIGH;
    }
  } else {
    digitalWrite(ledPin, LOW); // apaga el LED
    if (pirState == HIGH){
      // lo apagamos
      Serial.println("Motion no detected!");
      //Solo se imprime el cambio de la salida, no el estado
      pirState = LOW;
    }
  }
}

Guardado.
```

Figura 41. Código Arduino para el sensor de movimiento.

Fuente: Infante y González (2017).

Luego de escrito el código, se inicializó Node-red para hacer el flujo encargado de controlar los avisos de estado del sensor, en este caso, mediante mensajes de texto y correos enviados al usuario (ver figura 42). Para dicho flujo se tuvo que instalar el nodo “Twilio” y configura como se muestra en la figura 43.



**Figura 42. Diagrama de flujo en Node-red para el sensor de movimiento.**

Fuente: Infante y González (2017).

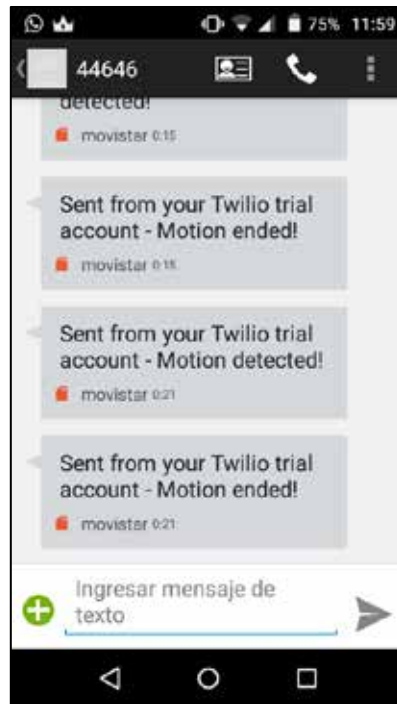
La imagen muestra la interfaz de configuración para el nodo 'twilio out' en Node-RED. El título de la ventana es 'Edit twilio out node'. En la parte superior hay tres botones: 'Delete', 'Cancel' y 'Done'. La configuración incluye:

- Credentials:** Un menú desplegable con la opción 'Use local credentials' seleccionada.
- Twilio:** Un campo de texto con un valor oculto por un recuadro rojo y un ícono de edición.
- Output:** Un menú desplegable con 'SMS' seleccionado.
- To:** Un campo de texto con un valor oculto por un recuadro rojo.
- Name:** Un campo de texto con el valor 'Name'.

**Figura 43 Configuración del nodo “Twilio”.**

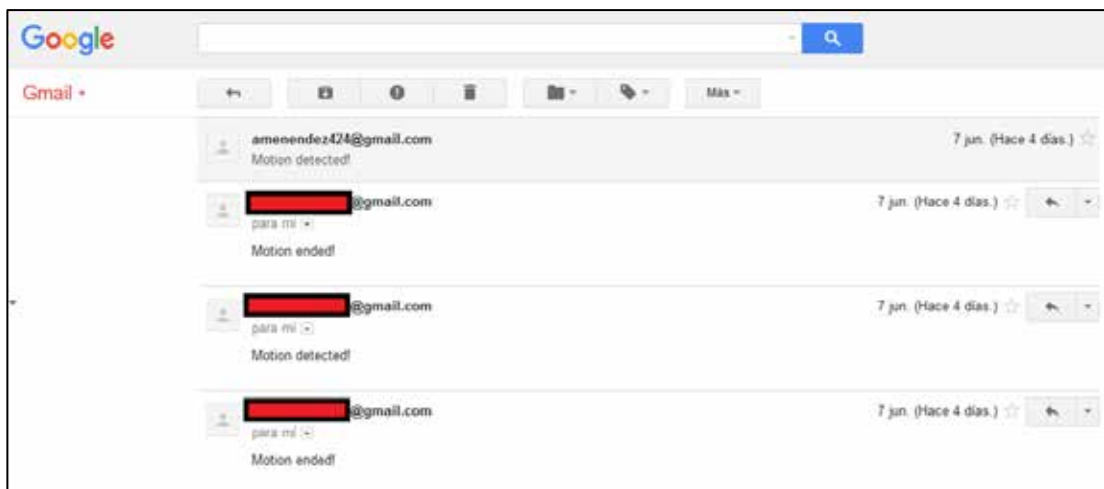
Fuente: Infante y González (2017).

En la figura 44 y 45, se puede apreciar el aviso que se le da al usuario, a través de mensajes de texto y correos, cuando el sensor de movimiento cambia de estado.



**Figura 44. Mensajes de texto indicando cambios de estado del sensor.**

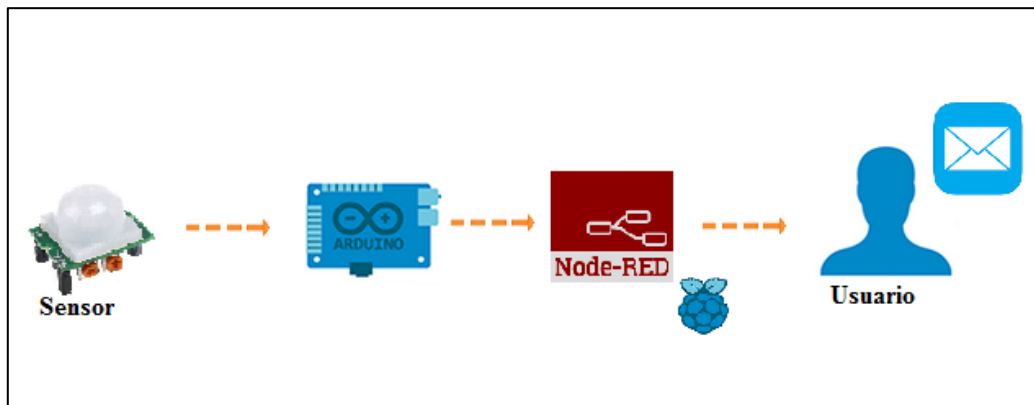
Fuente: Infante y González (2017).



**Figura 45. Correo indicando cambios de estado del sensor.**

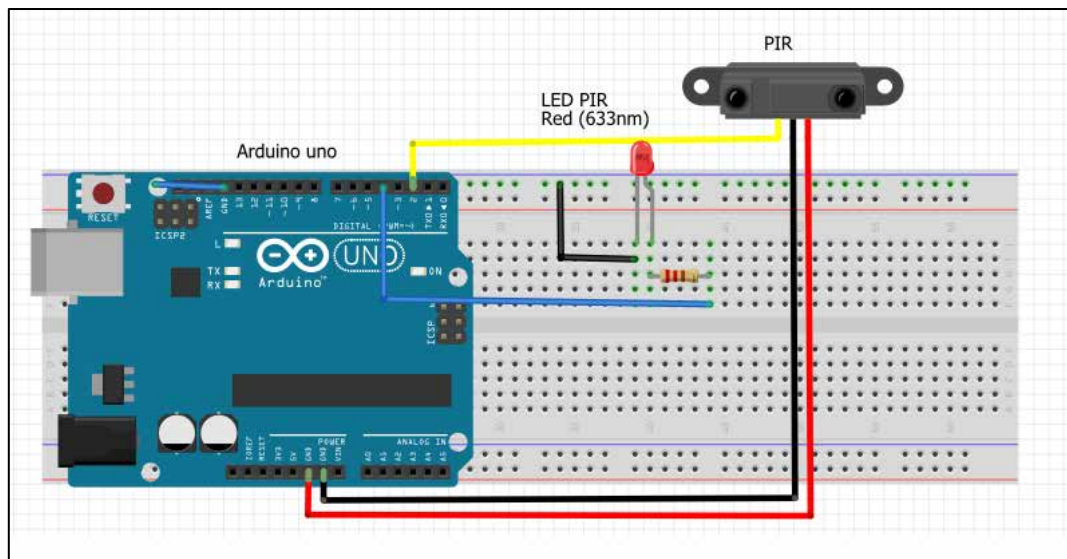
Fuente: Infante y González (2017).

Para explicar de forma gráfica la interacción de todos los elementos y programas que intervienen a lo largo del funcionamiento de esta Skill se hace uso de la figura 46.

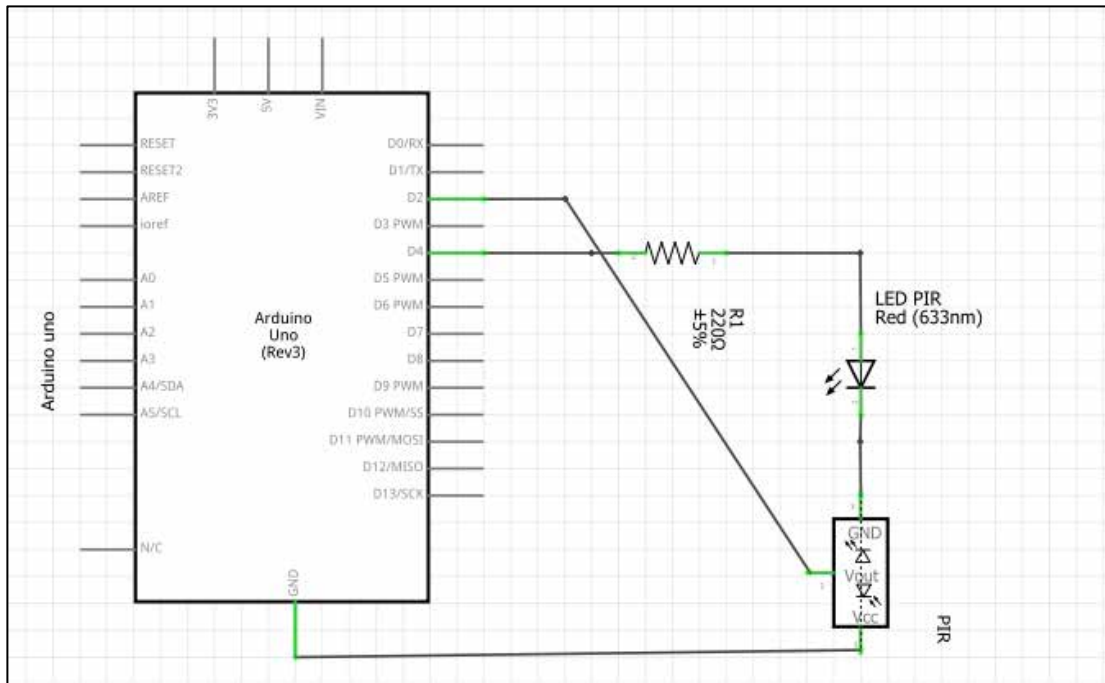


**Figura 46. Diagrama de Comunicación para el sensor de movimiento.**  
Fuente: Infante y González (2017).

En las figuras 47 y 48 se observa el diagrama de conexión y el diagrama esquemático respectivamente, para el funcionamiento de la Skill en cuestión.



**Figura 47. Diagrama de Conexión para el sensor de movimiento.**  
Fuente: Infante y González (2017).



**Figura 48. Diagrama Esquemático para el sensor de movimiento.**

Fuente: Infante y González (2017).

#### 4.4.4 Programación de Skill para el manejo de servomotores por comando de voz.

Con esta Skill, se pretende demostrar la gran versatilidad que posee el prototipo de Robot Social, ya que al ser capaz de controlar el encendido y el apagado de servomotores, se le puede dotar de movilidad. Dicho esto, se procede a la realización del código en IDE Arduino para el control de los servomotores (Ver figura 49); luego al igual que la skill del control de señales eléctricas ya explicada, esta skill basa su desarrollo en Flask-ask (ver figura 50). Finalizada la programación en Python, sigue la programación en el desarrollador de Amazon, siendo la configuración de esta parte de la skill, similar a las predecesoras mostradas en esta investigación, con la diferencia que se tienen que cambiar varios elementos como el “Intent Schema” (ver figura 51), también las “Utterances” (ver figura 52) y en la sección “Configuration” se coloca el URL obtenido de forma dinámica por el software Ngrok.

```

#include<Servo.h>
Servo servo;
Servo servo2;
int enviado; //Aqui enviamos el numero completo
int num; //Numero del servo
int posicion; //Posicion del servo
int led4 = 25;
int led5 = 27;
void setup()
{
  servo.attach(9);
  servo2.attach(11);
  servo.write(97);
  servo2.write(97);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  if(Serial.available() >= 1)
  {
    enviado = Serial.parseInt();
    num = enviado%10;
    enviado = enviado/10;
    posicion = enviado;
    if(num == 1)
    {
      digitalWrite(led4, HIGH);
      servo.write(posicion);
      delay(500);
      servo.write(97);
      digitalWrite(led4, LOW);
    }
    else if(num == 2)
    {
      digitalWrite(led5, HIGH);
      servo2.write(posicion);
      delay(1000);
      servo2.write(97);
      digitalWrite(led5, LOW);
    }
    else if(num == 3)
    {
      digitalWrite(led5, HIGH);
      servo2.write(posicion);
      delay(1000);
      servo2.write(97);
      digitalWrite(led5, LOW);
      digitalWrite(led4, HIGH);
      servo.write(posicion);
      delay(500);
      servo.write(97);
      digitalWrite(led4, LOW);
    }
  }
}

```

**Figura 49. Código IDE Arduino de los Servomotores.**

Fuente: Infante y González (2017).

```

from flask import Flask
from flask_ask import Ask, statement
import requests
import json
import serial

ser = serial.Serial("/dev/ttyACM0", 9600)
app = Flask(__name__)
ask = Ask(app, '/')
@ask.launch

@ask.intent("MotorRight")
def on():
    ser.write(b'1201')
    return statement("Motor Right moved")

@ask.intent("MotorLeft")
def on():
    ser.write(b'1202')
    return statement("Motor Left moved")

@ask.intent("MotorCenter")
def on():
    ser.write(b'1203')
    return statement("Motor Center moved")

@ask.intent("MotorBackRight")
def on():
    ser.write(b'801')
    return statement("Motor Back Right moved")

@ask.intent("MotorBackLeft")
def on():
    ser.write(b'802')
    return statement("Motor Back Left moved")

@ask.intent("MotorBackCenter")
def on():
    ser.write(b'803')
    return statement("Motor Back Center moved")

if __name__ == "__main__":
    app.run(debug=True)

```

**Figura 50. Código Python de los Servomotores.**

Fuente: Infante y González (2017).

```
Intent Schema  
The schema of user intents in JSON format. For more information, see Intent Schema.  
Also see built-in slots and built-in intents.  
  
34     "intent": "MotorRight"  
35   },  
36   {  
37     "intent": "MotorLeft"  
38   },  
39   {  
40     "intent": "MotorCenter"  
41   },  
42   {  
43     "intent": "MotorBackRight"  
44   },  
45   {  
46     "intent": "MotorBackLeft"  
47   },  
48   {  
49     "intent": "MotorBackCenter"  
50   }  
51 ]  
52 }
```

**Figura 51. Intent Schema de los Servomotores.**

Fuente: Infante y González (2017).

```
Sample Utterances  
These are what people say to interact with your skill. Type or paste in all the ways that people can invoke the intents. Learn more  
  
Up to 3 of these will be used as Example Phrases, which are hints to users.  
  
41 MotorRight on right  
42 MotorRight on rights  
43 MotorLeft on left  
44 MotorLeft on lefts  
45 MotorCenter on center  
46 MotorCenter on centers  
47 MotorRight on  
48 MotorLeft on  
49 MotorCenter on  
50 MotorBackRight off right  
51 MotorBackRight off rights  
52 MotorBackLeft off left  
53 MotorBackLeft off lefts  
54 MotorBackCenter off center  
55 MotorBackCenter off centers  
56 MotorBackRight off  
57 MotorBackLeft off  
58 MotorBackCenter off
```

**Figura 52. Utterances de los Servomotores.**

Fuente: Infante y González (2017).



## CONCLUSIONES

Se realizó una investigación donde se describieron los distintos proyectos en el área de la domótica y robótica existentes en Venezuela, gracias a esto se pudo saber cuan desarrollado estaba dicho país en el área. Por medio de este estudio se dieron a conocer los organismos encargados de realizar actividades relacionadas con estos temas en cuestión, sin embargo, por diversos motivos dichas actividades son ignoradas por una gran parte de la población venezolana, ya que no se promueven ni se incentiva a la comunidad en pro de la participación.

Se identificaron funcionalidades potenciales que se pueden agregar al prototipo de Robot Social, por medio de una encuesta realizada a los estudiantes de Ingeniería Electrónica de la Universidad José Antonio Páez, cursantes de los últimos semestres de la misma, ya que son considerados lo suficientemente experimentados en el área de la electrónica, para conocer su opinión y lograr saber cuánto manejaban ellos del tema a tratar, dando como resultado que las principales funcionalidades a explorar serían las de seguridad, interacción con el usuario y control de artefactos eléctricos.

Tras una ardua investigación se seleccionaron los software que se consideraron adecuados ya que existe una gran variedad de software que pueden ser utilizados para la complementación del software principal, además, después de haber elegido los software se procedió a la instalación y explicación del funcionamiento de los mismos, con el fin de cumplir las funcionalidades escogidas a desarrollar en este trabajo de grado, conjuntamente, se seleccionaron los componentes de hardware necesarios para el desarrollo tecnológico del prototipo de Robot Social.

Por último, se desarrollaron las distintas funcionalidades, propuestas desde la fase II de la investigación de estas, hasta la elaboración del prototipo de Robot Social, estas funcionalidades fueron desarrolladas a modo de demostración para el momento de la exhibición del prototipo.

## RECOMENDACIONES

- Se sugiere a la Universidad José Antonio Páez, la afiliación a la Asociación Venezolana de Robótica y Domótica, para incentivar a los estudiantes de dicha universidad a desarrollar sus habilidades por estas ramas de la ingeniería.
- Realizar el estudio pertinente para incluir al prototipo una biblioteca informática en el idioma español, para que este pueda hablar en dicho idioma.
- Mejorar el sistema de comunicación entre los módulos de Raspberry Pi y Arduino, habilitando una comunicación vía WiFi.
- Conectar los módulos de Raspberry Pi y Arduino a una fuente de alimentación portátil, de manera que el cableado externo no limite la posibilidad de movilidad ni el aspecto físico que se pueda dar a dicho prototipo
- Cambiar los servomotores de funcionamiento continuo a servomotores de funcionamiento estándar para lograr mayor precisión en el movimiento en el brazo del prototipo.
- Para mejorar su funcionamiento del brazo, dotar al prototipo de visión, la cual le otorgará una mayor movilidad y precisión en el control de los motores, a la hora de trasladarse a un punto en el espacio en particular.
- Se recomienda adicionar más Skills que le den valor agregado al prototipo de Robot Social, con su meta de automatizar y hacer más fácil las tareas cotidianas de los usuarios, adaptándose a cada necesidad.
- Agregar otras maneras de interacción entre el usuario y el prototipo, mediante el reconocimiento facial, por ejemplo.
- Dotar de un aspecto físico más amigable.
- Agregar un sistema de seguridad, para que únicamente los usuarios autorizados controlen al prototipo de Robot Social.

## REFERENCIAS BIBLIOGRAFICAS

Arias, F. (2.006). **El proyecto de investigación: Introducción a la metodología científica**, Caracas: Editorial Episteme.

Hurtado, J. (2.000). **Metodología de Investigación Holística (3ª. Ed.)**. Caracas: Editorial Sypal.

Rey J., Blanco A., (2.016). **Diseño de un sistema de control domótico con una aplicación web para el control inteligente y automatización del hogar con un mini ordenador Raspberry Pi**, San Diego: Venezuela.

Trespacios, J., Vázquez, R y Bello L (2.005). **Investigación de Mercados**, España: Editorial International Thomson Editores.

## REFERENCIAS ELECTRÓNICAS

<http://bot-boss.com/tutoriales-raspberry-pi/>, última visita: 15 de noviembre del 2.016 a las 9:00 PM.

<http://dspace.ups.edu.ec/bitstream/123456789/6779/1/UPS-CT003503.pdf>, última visita 26 de octubre de 2.016 a las 3:25 PM.

<http://www.eleconomista.es/apps/noticias/7177704/11/15/David-Cuartielles-cofundador-de-Arduino-Espana-es-uno-de-los-paises-con-una->

**cultura-maker-mas-sana-del-mundo.html**, última visita 27 de octubre de 2.016 a las 8:50 AM.

**https://hackaday.io/projects/tag/raspberry%20pi**, última visita: 2 de noviembre del 2.016 a las 8:14 PM.

Asociación venezolana de robótica y domótica (2.011).  
**http://www.ing.ula.ve/averod/?page\_id=14**, última visita: 23 de octubre del 2016 a las 3:26 PM.

**https://www.researchgate.net/publication/271483280\_Development\_of\_the\_Sociability\_of\_Non-Anthropomorphic\_Robot\_Home\_Companions**, última visita: 26 de octubre del 2.016 a las 1:48 PM.

**https://prezi.com/y7pw7azr4v/dispositivos-electricos-y-electronicos/**,  
Última visita 26 de octubre de 2.016 a las 9:10 PM.

**http://www.roboticaacademica.com/**, última visita: 8 de octubre de 2.016 a las 11:00 AM.

**https://www.raspberrypi.org/help/**, última visita 27 de octubre de 2.016 a las 7:10 AM.

**http://www.tendencias21.net/Robots-sociales-la-nueva-generacion\_a2833.html**, última visita 26 de octubre de 2.016 a las 7:42 PM.

**<https://nathan.chantrell.net/20160328/amazon-echo-alexa-skills-kit-integration-with-node-red/>**, última visita el 16 de abril de 2017 a las 1:40 am

**<https://www.hardill.me.uk/wordpress/2016/10/29/alexa-skills-with-node-red/>**, última visita el 20 de abril de 2017 a las 2: 50 pm

**<https://github.com/jbreed/AlexaDomoticz>**, última visita el 25 de abril de 2017 a las 5:50pm am

**<https://nodejs.org/es/docs/>**, última visita el 15 de mayo de 2017 a las 8:02 pm

**<https://docs.npmjs.com/>**, última visita el 18 de mayo de 2017 a las 3:15 pm

**<http://diymakers.es/conexion-remota-ssh-y-vnc/>**, última visita el 19 de mayo de 2017 a las 10:10 am

**<https://maven.apache.org/>**, última visita el 19 de mayo de 2017 a las 1:20 pm

**<https://www.arduino.cc/en/Reference/Servo>**, última visita el 25 de mayo de 2017 a las 6:40 pm

**<https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/using-a-pir>**, última visita el 25 de mayo de 2017 a las 7:40 pm

**<https://nodered.org/docs/>**, última visita el 1 de junio de 2017 a las 11:00 pm

**<https://developer.amazon.com/blogs/post/Tx14R0IYYGH3SKT/Flask-Ask-A-New-Python-Framework-for-Rapid-Alexa-Skills-Kit-Development>**, última visita el 1 de junio de 2017 a las 11:40 pm

**<https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/overviews/steps-to-build-a-custom-skill>**, última visita el 3 de junio de 2017 a las 8:00 pm

**<https://developer.amazon.com/alexa/smart-home>, última visita el 3 de junio de 2017  
a las 11:00 pm**

**<https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit/getting-started-guide>, última visita el 4 de junio a las 1:20 am**

**APENDICE A**  
**ENCUESTA**

	ÍTEM	SI	NO
1	¿Había oído hablar sobre el Robot Social?		
2	¿Cree usted que para que un dispositivo se considere robot social debe tener estrictamente apariencia física?		
3	¿Está de acuerdo con el desarrollo de este tipo de tecnología?		
4	¿Le gustaría que el Robot Social interactúe o controle los dispositivos de su hogar?		
5	¿Le gustaría tener un dispositivo que lo ayude a organizar sus actividades diarias?		
6	¿Cree usted pertinente tener un dispositivo que lo apoye en la seguridad de su hogar?		
7	¿Implementaría en su hogar una inteligencia artificial que reciba a sus visitas?		
8	¿Le gustaría un dispositivo inteligente con el que usted pueda mantener una conversación simple?		
9	¿Quisiera usted tener un dispositivo que le ayude con las labores domésticas?		