



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**TRABAJO DE TITULACIÓN, PREVIO LA OBTENCIÓN DEL
TÍTULO DE INGENIERA DE SISTEMAS E INFORMÁTICA**

**TEMA: “APLICACIÓN MÓVIL PARA EL CONTROL REMOTO DE
DISPOSITIVOS ELECTRÓNICOS DE UNA VIVIENDA BASADO EN UN
CHATBOT”**

**AUTORAS: DÍAZ TORRES, ISABEL DAYANARA
VÁSQUEZ FUELTALA, ELIANA LISSETTE**

DIRECTOR: Ing. RAURA RUÍZ, JORGE GEOVANNY, MS.c.

SANGOLQUÍ

2020



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "*APLICACIÓN MÓVIL PARA EL CONTROL REMOTO DE DISPOSITIVOS ELECTRÓNICOS DE UNA VIVIENDA BASADO EN UN CHATBOT*" fue realizado por las señoritas *Díaz Torres, Isabel Dayanara y Vásquez Fuetala, Eliana Lisette* el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 12 de Febrero del 2020

Ing. Raura Ruiz, Jorge Geovanny
C.C: 0501773063
DIRECTOR



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

AUTORÍA DE RESPONSABILIDAD

Nosotras, *Díaz Torres, Isabel Dayanara y Vásquez Fuelta, Eliana Lissette*, declaramos que el contenido, ideas y criterios del trabajo de titulación: *Aplicación móvil para el control remoto de dispositivos electrónicos de una vivienda basado en un chatbot* es de nuestra autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolquí, 12 de Febrero del 2020

Díaz Torres Isabel Dayanara
C.C: 1717642753

Vásquez Fuelta Eliana Lissette
C.C: 1721827366



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

AUTORIZACIÓN

*Nosotras, Díaz Torres, Isabel Dayanara y Vásquez Fweltala, Eliana Lissette, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Aplicación móvil para el control remoto de dispositivos electrónicos de una vivienda basado en un chatbot en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.***

Sangolquí, 12 de Febrero del 2020

Díaz Torres Isabel Dayanara
C.C: 1717642753

Vásquez Fweltala Eliana Lissette
C.C: 1721827366

DEDICATORIA

A mis padres Pedro e Isabel, quienes son el motor más importante en vida. De los cuales aprendí los valores que me han formado como persona, y quienes con su amor incondicional me han inculcado la fortaleza para continuar en la lucha de mis sueños y alcanzar las metas propuestas en esta etapa de mi vida. A mis hermanas María y Johanna, quienes con su entusiasmo y cariño me apoyaron para seguir adelante.

Isabel Díaz

La presente tesis la dedico primeramente a Dios, por darme la fuerza para lograr alcanzar mis objetivos; a mis padres que, con su amor, trabajo y sacrificio, me apoyaron siempre, gracias a ustedes he logrado llegar hasta aquí. Ha sido un orgullo y un privilegio ser su hija, a mi hermano, por estar siempre presente apoyándome hasta el final y, a todas las personas que han aportado su granito de arena para que este proyecto se realice con éxito en especial a aquellos que siempre estuvieron a mi lado apoyándome y creyeron en mí.

Eliana Vásquez

AGRADECIMIENTO

Agradezco a mi gran universidad UFA-ESPE, por abrirme las puertas para poder estudiar y culminar mi carrera, a mis profesores, quienes con su conocimiento me han guiado en mi etapa estudiantil, y de forma especial a mi tutor Ing. Geovanny Raura, por su orientación y apoyo para la realización de este proyecto. A mis compañeros que estuvieron día a día compartiendo gratas experiencias a lo largo de mi trayectoria, las cuales siempre recordare con mucho amor.

Isabel Díaz

Agradezco a Dios por guiarme a lo largo de mi vida, por ser el apoyo y fortaleza en los momentos difíciles, gracias a mis padres por confiar y creer en mí; además, de haberme formado con valores y principios que fueron aplicados en mi vida universitaria, agradezco a los docentes de la Universidad de las Fuerzas Armadas “ESPE” por haber compartido sus conocimientos a lo largo de mi carrera, de manera especial al Ing. Geovanny Raura, tutor de la presente tesis quién nos ha guiado con paciencia, y dedicación, gracias a mis amigos quienes me acompañaron en toda esta etapa tanto dentro y fuera de las aulas.

Eliana Vásquez

ÍNDICE DE CONTENIDOS

CERTIFICADO DEL DIRECTOR	i
AUTORÍA DE RESPONSABILIDAD	ii
AUTORIZACIÓN	ii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE TABLAS	xiii
ÍNDICE DE FIGURAS	xv
RESUMEN	xvii
ABSTRACT	xix
CAPÍTULO I	
ASPECTOS GENERALES	1
1.1 Antecedentes.....	1
1.2 Contexto del problema	3
1.2 Pregunta de investigación.....	4
1.3 Objetivos	4

1.3.1 Objetivo General	4
1.3.2 Objetivos Específicos	4
1.4 Justificación.....	5
1.5 Estado del arte	6
1.5.1 Planteamiento del mapeo de literatura	7
1.5.2 Obtención del grupo de control y palabras clave para la investigación	7
1.5.3 Creación y afinación de la cadena de búsqueda	8
1.5.4 Selección de estudios primarios	9
1.5.5 Conclusiones al estado del arte.....	12
1.6 Hipótesis de trabajo	13
1.7 Metodologías	13
1.7.1 Investigación-Acción.....	14
1.7.2 Metodología basada en prototipos.....	15
1.7.3 Evaluación de tecnologías para el desarrollo de chatbots	16
1.7.4 Validación de la solución propuesta.....	17
CAPÍTULO II	
MARCO TEÓRICO	18
2.1 Accidentes domésticos	18
2.1.1 Tipos de accidentes domésticos.....	18
2.1.2 Choque eléctrico.....	18
2.2 Nuevas propuestas tecnológicas	19

2.3 Internet de las Cosas (IoT)	20
2.3.1 Arquitectura IOT	20
2.3.2 Protocolos y Estándares.....	21
2.3.3 Aplicaciones IOT.....	22
2.3.4 Monitoreo y control.....	23
2.3.5 Domótica	23
2.3.6 Vivienda Inteligente	23
2.4 Inteligencia Artificial (IA).....	24
2.5.2 Principios fundamentales de la Inteligencia Artificial	25
2.5.3 Lenguaje natural	26
2.5.4 Interacción hombre-máquina (HCI)	26
2.5.5 Inteligencia artificial, asistentes virtuales y lenguaje natural.....	27
2.6 Chatbot	28
2.6.1 Historia de los chatbots	29
2.6.2 Características de los chatbots.....	31
2.6.3 Criterios para la evaluación de chatbots.....	32
2.7 Arquitectura y diseño de un chatbot.....	32
2.7.1 Conversión de voz a texto	33
2.7.2 Modelo de proceso de reconocimiento automático de voz (ASR).....	34
2.7.3 Procesamiento de lenguaje natural (PLN).....	36
2.8 Métodos y técnicas para extraer información.....	39
2.8.1 Reconocimiento de acto de diálogo.....	39

2.8.2 Identificación de intenciones	39
2.8.3 Extracción de la información	40
2.9 Administrador de diálogo	41
2.9.1 Estrategias de interacción con chatbots	41
2.10 Generación de respuestas de un chatbot (GR).....	42
2.10.1 Estrategia de generación de respuesta	43
2.10.2 Creación de base de conocimiento	43
2.11 Texto a voz	45
2.12 Plataformas para diseñar chatbots	45
2.13 Subpreguntas de investigación	47

CAPÍTULO III

EVALUACIÓN DE TECNOLOGÍAS PARA EL DESARROLLO DE CHATBOTS.....	49
3.1 Desarrollo de la semántica.....	49
3.1.1 API natural language	50
3.1.2 Características de la API Natural Language.....	51
3.1.3 Funcionalidad de API Natural Language	51
3.1.4 Análisis de entidades en API Natural Language	52
3.1.5 Análisis Sintáctico en API Natural Language	53
3.2 Dialogflow.....	55
3.2.1 Arquitectura de DialogFlow	56
3.3 IFTTT.....	57

3.3.1 Arquitectura de IFTTT	58
3.4 Amazon Alexa	59
3.4.1 Arquitectura Amazon Alexa	59
3.4.2 Alexa Skills Kit (ASK).....	60
3.4.3 Alexa Voice Services (AVS).....	61
3.5 Wit.ai.....	61
3.5.1 Arquitectura de Wit.ai	62
3.5.2 PNL incorporado en Facebook Messenger.....	62
3.6 Cuadros comparativos de las plataformas	63
3.7 Pruebas y resultados con las API's.....	67
3.7.1 Pruebas de precisión	67
3.7.2 Resultados de pruebas de precisión	71
3.7.3 Pruebas funcionales	73
3.7.4 Resultados de pruebas funcionales	75
CAPÍTULO IV	
IMPLEMENTACIÓN	77
4.1 Metodología basada en prototipo	77
4.2 Primera Fase: Identificar requerimientos básicos.....	78
4.2.1 Introducción.....	78
4.2.2 Descripción general	81
4.2.3 Requisitos específicos	83

4.3 Segunda Fase: Diseño del prototipo inicial	87
4.3.1 Diagrama de arquitectura	87
4.3.2 Diagramas de secuencia	88
4.3.3 Sketch board inicial del prototipo	91
4.4 Tercera Fase: Implementación del prototipo inicial	91
4.4.1 Selección de herramientas	92
4.4.2 Implementación de aplicación web	93
4.4.3 Implementación Cliente Wit.ai.....	94
4.4.4 Implementación de audio	97
4.4.5 Implementación de respuestas	98
4.4.6 Conexión con openHAB.....	100
4.4.7 Configuración de dispositivo.....	102
4.5 Cuarta Fase: Pruebas y mejora del aplicativo.....	104
4.5.1 Etapa I: Pruebas de requerimientos	104
4.5.2 Etapa II: Pruebas de Funcionalidad.....	105
4.5.1 Etapa III: Pruebas de Interfaz Gráfica	106
CAPÍTULO V	
VÁLIDACIÓN DE HIPÓTESIS.....	107
5.1 Estudio realizado	107
5.2 Características de los usuarios.....	108
5.3 Encuesta.....	109

5.3.1 Prueba de funcionalidad	109
5.3.2 Prueba de grado de aceptación	111
5.3.3 Usabilidad del sistema	112
5.4 Resultados obtenidos	113
CAPÍTULO VI	
CONCLUSIONES Y RECOMENDACIONES	118
6.1 Conclusiones	118
6.2 Recomendaciones	120
REFERENCIAS BIBLIOGRÁFICAS	121

ÍNDICE DE TABLAS

<i>Tabla 1 Estudios de control</i>	7
<i>Tabla 2 Estudios primarios</i>	9
<i>Tabla 3 Subpreguntas de investigación</i>	47
<i>Tabla 4 Características de la API de Natural Language</i>	51
<i>Tabla 5 Categorías gramaticales de API Natural Language</i>	53
<i>Tabla 6 Cuadro Comparativo. Dificultad de cada API</i>	64
<i>Tabla 7 Cuadro comparativo. Características generales de cada API</i>	65
<i>Tabla 8 Pruebas de coincidencia exacta</i>	67
<i>Tabla 9 Pruebas de Tasa de error por palabra</i>	69
<i>Tabla 10 Resultados obtenidos en prueba de Oraciones exactas reconocidas.</i>	72
<i>Tabla 11 Resultados obtenidos en prueba de Tasa de error por palabra</i>	72
<i>Tabla 12 Pruebas de funcionamiento</i>	73
<i>Tabla 13 Resultados obtenidos en pruebas de funcionalidad</i>	75
<i>Tabla 14 Documento ERS: Definiciones</i>	80
<i>Tabla 15 Documento ERS: Acrónimos</i>	80
<i>Tabla 16 Documento ERS: Abreviaturas</i>	80
<i>Tabla 17 Requerimiento RF01</i>	83
<i>Tabla 18 Requerimiento RF02</i>	84
<i>Tabla 19 Requerimiento RF03</i>	84
<i>Tabla 20 Requisitos no funcionales</i>	85
<i>Tabla 21 Equipos requeridos para el desarrollo del chatbot</i>	86

<i>Tabla 22 Herramientas utilizadas para el desarrollo del chatbot.....</i>	<i>86</i>
<i>Tabla 23 Estructura de ruta.....</i>	<i>93</i>
<i>Tabla 24 Constructor Wit.....</i>	<i>95</i>
<i>Tabla 25 Método Message</i>	<i>95</i>
<i>Tabla 26 Formato de respuestas Wit.ai</i>	<i>96</i>
<i>Tabla 27 Códigos de error.....</i>	<i>98</i>
<i>Tabla 28 Implementación de respuestas</i>	<i>99</i>
<i>Tabla 29 Parámetros definidos para la comunicación en tiempo real.....</i>	<i>100</i>
<i>Tabla 30 Solicitud al servidor openHAB</i>	<i>102</i>
<i>Tabla 31 Pruebas de requerimientos</i>	<i>104</i>
<i>Tabla 32 Pruebas de funcionalidad</i>	<i>105</i>
<i>Tabla 33 Pruebas de Interfaz gráfica</i>	<i>106</i>
<i>Tabla 34 Características demográficas de los participantes.....</i>	<i>109</i>
<i>Tabla 35 Cuestionario.....</i>	<i>113</i>
<i>Tabla 36 Resultados obtenidos en ejecución de pruebas.....</i>	<i>114</i>

ÍNDICE DE FIGURAS

<i>Figura 1. Grupo de consumo de energía eléctrica Fuente: (ARCONEL, 2018)</i>	4
<i>Figura 2. Ciclo de la metodología investigación – acción</i>	14
<i>Figura 3. Elementos de IoT</i>	21
<i>Figura 4. Protocolos y Estándares comúnmente de IoT</i>	22
<i>Figura 5. Línea de tiempo chatbots.</i>	30
<i>Figura 6. Arquitectura básica de un chatbot</i>	33
<i>Figura 7. Arquitectura de un chatbot basado en voz.</i>	35
<i>Figura 8. Arquitectura de un sistema PNL</i>	38
<i>Figura 9. Identificación de entidades e intenciones</i>	40
<i>Figura 10. Corpus de datos de DialogFlow</i>	50
<i>Figura 11. Funcionalidad de la API Natural Language</i>	52
<i>Figura 12. Árbol de dependencia</i>	55
<i>Figura 13. Generación de Intent</i>	57
<i>Figura 14. Arquitectura Amazon Alexa</i>	59
<i>Figura 15. Servidores de habilidades para skills nativas y terciarias.</i>	60
<i>Figura 16. Representación de entidades e intenciones</i>	62
<i>Figura 17. Pruebas de tasa de error por palabra con Alexa.</i>	70
<i>Figura 18. Resultados de prueba de precisión</i>	73
<i>Figura 19. Pruebas funcionales con Alexa</i>	75
<i>Figura 20. Diagrama de caso de uso.</i>	82
<i>Figura 21. Diagrama de arquitectura</i>	88

<i>Figura 22. Diagrama de Secuencia – Aplicación Web/Móvil</i>	<i>90</i>
<i>Figura 23. Interfaz inicial del prototipo</i>	<i>91</i>
<i>Figura 24. Interfaz de aplicación web.</i>	<i>94</i>
<i>Figura 25. Implementación audio.....</i>	<i>97</i>
<i>Figura 26. Respuestas del bot.....</i>	<i>99</i>
<i>Figura 27. Configuración del enlace del dispositivo.....</i>	<i>103</i>
<i>Figura 28. Configuración del dispositivo</i>	<i>103</i>
<i>Figura 29. Tareas realizadas por Grupo 1 (Voz y texto).....</i>	<i>114</i>
<i>Figura 30. Tareas realizadas por Grupo 2 (Voz y texto).....</i>	<i>115</i>
<i>Figura 31. Porcentaje obtenido con respecto a la aceptación y satisfacción de uso</i>	<i>115</i>
<i>Figura 32. Porcentaje obtenido en análisis de usabilidad</i>	<i>116</i>
<i>Figura 33. Resultados obtenidos en cuanto a la prevención de accidentes domésticos.....</i>	<i>117</i>

RESUMEN

Varias son las ocasiones en que se producen accidentes dentro del hogar, tanto por descuidos del ser humano o por uso inadecuado de artefactos eléctricos. El objetivo de esta investigación, es implementar un aplicativo para el control de artefactos eléctricos basado en un chatbot mediante lenguaje natural para disminuir el riesgo de accidentes domésticos. Para esto, el estudio evaluó y comparó cuatro herramientas tecnológicas para el desarrollo de chatbots, tales como Wit.ai, Dialogflow, Amazon Alexa e IFTTT. A fin de obtener la herramienta que brinde las mejores prestaciones, habiendo eligió a Wit.ai, por ser éste un software libre y por la simplicidad del manejo e integración a la API HTTP. Al realizar la comparación de las API's, los resultados obtenidos señalan que el principal problema es el reconocimiento de palabras homófonas y modismos. El prototipo desarrollado simuló el control de artefactos eléctricos mediante el uso de una Raspberry-Pi y un Smart-Led. Para la validación del estudio se aplicó una encuesta, la cual consta de pruebas de funcionalidad, grado de aceptación y usabilidad, cuyos resultados fueron: el grado de aceptación del aplicativo fue alto, la escala de usabilidad del aplicativo es aceptable y puede seguir mejorando. Finalmente, la utilización e implementación de herramientas de código abierto, tales como Wit.ai y OpenHAB, pueden generar una óptima solución al momento de controlar artefactos eléctricos sin depender de proveedores, ni generar altos costos.

PALABRAS CLAVE:

- **CHATBOT**
- **DOMÓTICA**

- **ACCIDENTES DOMÉSTICOS**
- **INTERNET DE LAS COSAS**

ABSTRACT

There are several occasions when accidents occur inside the home, both due to carelessness of the human being or the use of electrical appliances. The objective of this research is to implement an application for the control of electrical appliances based on a chatbot through natural language to reduce the risk of domestic accidents. For this, the study evaluated and compared four technological tools for the development of chatbots, stories such as Wit.ai, Dialogflow, Amazon Alexa and IFTTT. In order to obtain the tool that provides the best features, having chosen Wit.ai, as this is a free software and for the simplicity of handling and integration to the HTTP API. When comparing the APIs, the results indicated that the main problem is the recognition of homophonic words and idioms. The developed prototype simulated the control of electrical devices by using a Raspberry-Pi and a Smart-Led. For the validation of the study, a survey was applied, the constant quality of functionality tests, degree of acceptance and usability, whose results were: the degree of acceptance of the application was high, the scale of usability of the application is acceptable and can continue to improve. Finally, the use and implementation of open source tools, stories like Wit.ai and OpenHAB, can generate an optimal solution when controlling electrical appliances without dependence on suppliers, nor generate high costs.

KEYWORDS:

- **CHATBOT**
- **DOMOTICS**
- **DOMESTIC ACCIDENTS**
- **IOT**

CAPÍTULO I

ASPECTOS GENERALES

1.1 Antecedentes

La Organización Mundial de la Salud definió el accidente en el hogar como aquel que ocurre dentro y fuera de la vivienda propiamente dicha. Pueden ser debido en parte a la creciente mecanización de la vida doméstica y situaciones peligrosas en el hogar derivados del mal uso del agua, fuego, gas y electricidad, poniendo en riesgo la vida de las personas y perjudicando las viviendas parcial o totalmente (Chaparro, 1999).

Hace unos años atrás no era posible controlar los dispositivos a través de una red, la única forma de encender o apagar las máquinas o controlarlas era a través de interruptores e interfaces similares. Con el crecimiento tecnológico surgió el concepto de dispositivos inteligentes y casas habilitadas para el uso de internet de las cosas (IoT), donde a cada dispositivo se le asigna una dirección IP y, con la ayuda del internet, estos pueden ser controlados desde cualquier parte del mundo (Khanna, y otros, 2015).

El área de la ingeniería conocida como domótica o ciencia de la innovación, está creciendo de forma exponencial, y las principales empresas que se encargan de diseñar electrodomésticos están dotándolos de inteligencia para aumentar las funciones que tiene cada uno de ellos, supliendo las necesidades que se presentan en el hogar (Herrera, 2005). En la actualidad la industria de internet de las cosas (IoT) traerá consigo nuevas oportunidades y retos al implicar la interconexión masiva de dispositivos con el fin de proporcionar servicios de valor agregado en varios dominios

tales como la energía, transporte, salud, entre otros, logrando aumentar la productividad y mejorando la experiencia con el cliente (El Kaed, Ponnouradjane, & Shah, 2018).

Pero ahora, ¿Qué pasa con el siguiente nivel de interacción humana y máquinas? Prevemos que la interacción en lenguaje natural es la que tendrá lugar en el futuro. Los desarrollos ya están sucediendo en este campo (Khanna, y otros, 2015). El habla humano-computacional cada vez está ganando impulso como una técnica de interacción. En los últimos años existe un incremento en motores de búsquedas basados en voz como Siri, Google, Chrome y Cortana.

Se pueden aplicar técnicas de procesamiento de lenguaje natural, para analizar el habla, encontrando respuestas inteligentes mediante el diseño de un motor que brinde respuestas adecuadas de tipo humano, estos programas comúnmente se los denominados Chatbot (Abdul-Kader & Woods, 2015). Los Chatbot permiten dar respuestas adecuadas a las palabras clave o frases extraídas del vocabulario habitual para mantener una conversación continua, también tienen la capacidad de examinar e influir en el comportamiento del usuario al hacer y responder preguntas (Abdul-Kader & Woods, 2015).

Los chatbots están reemplazando lentamente las aplicaciones en dispositivos portátiles debido a su facilidad de uso e inteligencia, la cual se debe a las técnicas de procesamiento del lenguaje natural. Un chatbot debe tener la capacidad de comprender el contexto de una conversación, aprender de las conversaciones y mejorar a lo largo del tiempo (Baby, Khan, & Swathi, 2017).

El presente trabajo propone realizar el análisis de varias APIs, las cuales permiten el desarrollo de un chatbot, dentro de las cuales se escogerá la que tenga mejores prestaciones en cuanto al análisis de texto y reconocimiento de voz. El chatbot que se desarrollará se basará en el

procesamiento de lenguaje natural y se usará para el control de los dispositivos de forma remota mediante una aplicación, con el fin de disminuir posibles accidentes domésticos.

1.2 Contexto del problema

El consumismo actual de la tecnología y el auge en la vida moderna de los aparatos electrónicos facilitan el trabajo a sus usuarios ya sea fuera o dentro del hogar, pero a su vez hace que las redes eléctricas que fueron diseñadas para un determinado periodo y con una potencia baja colapsen, lo que acarrea que los equipos electrónicos sufran daños y en algunos casos la pérdida total de estos. Los accidentes domésticos que se pueden ocasionar, causa preocupación dentro de la población. Esto debido a varios factores como la precaria situación de instalaciones, el uso inadecuado de materiales y equipos electrónicos, los cuales pueden producir un impacto ambiental negativo, sobrecargas, cortocircuitos y en muchos casos pérdidas materiales al provocarse incendios dentro de la vivienda, o heridas graves por electrocuciones resultando afectada la persona (Flores Garate, 2018; Verduga & Zambrano, 2017).

Durante el año 2018 en el Ecuador el número de clientes por grupo de consumo de energía eléctrica, como se muestra en la Figura 1, tuvo su mayor representación en el sector residencial con el 88%, seguido por el comercial con un 9% (ARCONEL, 2018). Se evidencia que el mayor consumidor de energía eléctrica en el país es el sector residencial, esto corresponde a los electrodomésticos que usualmente se usa y a los que permanecen conectados sin utilizarlos, creando un gasto innecesario en la planilla eléctrica. Por ejemplo, cuando olvidamos apagar la luz de una habitación, incluso al salir de casa estamos intranquilos al no tener certeza del estado en que quedaron las cosas y nadie tiene acceso al lugar (Del Campo, Quiñonez, García, & Magnífico, 2009; Álvarez, 2015).



Figura 1. Grupo de consumo de energía eléctrica
Fuente: (ARCONEL, 2018)

1.2Pregunta de investigación

¿Es posible disminuir el riesgo de accidentes domésticos producidos por artefactos eléctricos mediante el uso de un chatbot?

1.3Objetivos

1.3.1Objetivo General

Desarrollar una aplicación para el control de artefactos eléctricos basada en un chatbot con interacción mediante lenguaje natural para disminuir el riesgo de accidentes domésticos.

1.3.2Objetivos Específicos

- ✓ Realizar un estudio de literatura que permita analizar diferentes técnicas de desarrollo e implementación de chatbots utilizando lenguaje natural en el área de IoT (Internet de las Cosas), con el fin de reducir el riesgo de accidentes domésticos.
- ✓ Definir una semántica específica para la comunicación entre la persona y el dispositivo móvil mediante el uso del procesamiento de lenguaje natural.

- ✓ Hacer un estudio comparativo de las APIs de reconocimiento de voz para determinar la que ofrece mejores prestaciones para el desarrollo del chatbot.
- ✓ Desarrollar un aplicativo para interacción con el chatbot y los servicios de interconexión, aplicando la metodología basada en prototipos.
- ✓ Validar la solución propuesta mediante la definición de pruebas funcionales y de rendimiento.

1.4Justificación

Los accidentes son la enfermedad de la negligencia y el descuido, también llamada como plaga del siglo XX aparece desde que el hombre comenzó a ponerse en contacto con la naturaleza en su lucha por la supervivencia y sus víctimas apuntan cada día en proporciones considerables (Nereida Pacios & Salazar, 1999).

Según la Organización Mundial de Salud (OMS) los accidentes domésticos representan la tercera parte de los accidentes que se reportan como productores de lesiones y muerte. Está define al accidente como “un acontecimiento fortuito, generalmente desgraciado o dañino, independiente de la voluntad humana, provocado por una fuerza exterior que actúa rápidamente y que se manifiesta por la aparición de lesiones orgánicas o trastornos mentales” (Nereida Pacios & Salazar, 1998).

Uno de los tipos de accidentes domésticos más frecuentes es el shock eléctrico, el cual puede ser el resultado de una sobrecarga en los circuitos, cables en mal estado, material aislado deficientemente, interruptores defectuosos y mal empleo de los aparatos eléctricos, los cuales pueden provocar quemaduras a los usuarios hasta incendios en la vivienda. En las casas antiguas

se deben revisar las instalaciones ya que los aparatos eléctricos modernos demandan mayor carga de corriente, sobrecargando los circuitos y en consecuencia haciendo que se sobrecalienten, debido al uso excesivo de aparatos electrónicos como computadoras, televisores, refrigeradoras, etc. (Minda Almagor, 2013).

En este trabajo, se plantea desarrollar una aplicación que permita el control de los dispositivos eléctricos mediante un chatbot, el cual pueda ayudar a disminuir los accidentes domésticos. En los siguientes apartados, se muestra el estado del arte, el marco teórico referencial, el análisis y diseño de un prototipo de solución y finalmente se realiza una validación con el usuario.

1.5 Estado del arte

Con el fin de identificar y analizar el estado del arte actual, para determinar la viabilidad de la solución propuesta, se realizó inicialmente un mapeo sistemático de literatura o también conocido como estudio de antecedentes (Wohlin, y otros, 2013). El cual hace referencia a una amplia revisión de estudios primarios en un área temática específica que tiene como objetivo el identificar qué evidencia está actualmente disponible sobre el tema (Kitchenham & Charters, 2007).

El proceso de estudio de antecedentes realizado, contempla las siguientes fases: (1) planteamiento del estudio sistemático de literatura, (2) obtención del grupo de control y palabras clave para la investigación, (3) creación y afinación de la cadena de búsqueda, (4) selección de estudios primarios y, (5) elaboración del estado del arte. A continuación, se describirá cada actividad de este proceso.

1.5.1 Planteamiento del mapeo de literatura

Como fase inicial del proceso de estudio, se realizó la descripción del problema en el cual está enfocada la investigación. A fin de definir el marco de búsqueda de estudios científicos, seguido del planteamiento de las preguntas de investigación y finalmente el establecimiento de los criterios de inclusión y exclusión.

1.5.2 Obtención del grupo de control y palabras clave para la investigación

Según (Levac & Colquhoun, 2010), esta fase implica identificar los estudios relevantes y desarrollar un plan de decisión para determinar dónde buscar, qué términos usar y qué fuentes se deben consultar. Las fuentes incluyen bases de datos electrónicas, listas de referencia, búsquedas manuales de revistas clave, organizaciones y conferencias. Tras el análisis de varios artículos científicos propuestos por los investigadores, se obtuvo el grupo de control (GC) del estudio de artículos relacionados al problema central, los cuales son: accidentes domésticos, aparatos eléctricos y tecnologías que dan solución a la problemática. Este grupo está conformado por 8 artículos científicos los cuales son relevantes en cuanto a los temas anteriormente mencionados como se detalla en la Tabla 1.

Tabla 1
Estudios de control

Código	Título	Cita	Palabras clave
EC1	IoT based monitoring and control system for home automation	(Pavithra & Balakrishnan, 2015)	fireplace accidents, appliances domestically, appliances, controlling home appliances
EC2	Combining cell phones and WSNs for preventing accidents in smart-homes with disabled people	(Freitas, y otros, 2015)	Monitoring, possible accidents, assisted living
EC3	A smart monitoring and control system for the household electric power usage	(Teng-Fa & Cheng-Chien, 2013)	Monitoring, control system, household electric, home appliance, household appliance, domestic appliances

CONTINÚA 

EC4	An IoT-Based Smart Solution for Preventing Domestic CO and LPG Gas Accidents	(Gomes, y otros, 2018)	gas monitoring systems, domestic accidents, accidents
EC5	User Indoor Localisation System Enhances Activity Recognition: A Proof of Concept	(Fiorini, Bonaccorsi, Betti, & Cavallo, 2016)	risk of domestic accidents
EC6	Detecting falls at home: User-centered design of a pervasive technology	(Chaumon, Cuvillier, Body, & Cros, 2016)	domestic accidents, falls at home
EC7	Toward pervasive computing system to enhance safety of ageing people in smart kitchen	(Abdulrazak, Yared, Tessier, & Mabilleanu, 2015)	domestic accidents, fire accidents in residence, risk
EC8	A User Recognition Method Using Accelerometer for Electric Appliances	(Terada, Watanabe, & Tsukamoto, 2013)	Appliances, acceleration sensor, prevent accidents, breaker falls

La selección de los artículos científicos pertenecientes al grupo de control (GC), permitieron determinar las palabras clave que se encuentren alineadas al objetivo de la investigación, las cuales fueron: *domestic appliances, appliances domestically, home appliances, household electric, household appliance, appliance, fireplace accidents, possible accidents, domestic accidents, fire accidents in residence, accidents, risk situations, prevent accidents, breaker falls.*

1.5.3 Creación y afinación de la cadena de búsqueda

Una vez que las palabras clave han sido identificadas, se procede a crear y probar diversas cadenas de búsqueda, en este caso, se lo realizó en la base digital IEEE. La primera cadena de búsqueda creada fue la siguiente:

{("appliance") OR ("domestic appliances") OR ("appliances domestically") OR ("home appliances") OR ("household electric") OR ("household appliance")} AND {"fireplace accidents") OR ("possible accidents") OR ("domestic accidents") OR ("fire accidents in residence") OR ("accidents") OR ("risk situations") OR ("prevent accidents") OR ("breaker falls")}

La cadena de búsqueda propuesta arrojó un resultado de 100 artículos, para seguir con el afinamiento de la cadena, dentro de los criterios de exclusión, a fin de detallar con mayor

precisión el tipo de artículo científico que se consideraría como válido para la realización del estudio de antecedentes, se determinó aplicar dos filtros de búsqueda adicionales:

Año: Los artículos candidatos deberían haber sido publicados a partir del año 2014. Se decidió aplicar este filtro con el objetivo de que las propuestas a la problemática investigada, se encuentren lo más apegadas a la realidad actual de los avances tecnológicos y al nacimiento de los asistentes virtuales (Alexa) y tecnologías referentes a la interacción humano computadora como es el caso de Api.ai.

Tipo de documento: Se consideraron únicamente documentos de tipo: Conference Paper y Article, debido a que este tipo de documentos son reflejo de una investigación que genera un impacto elevado en el mundo científico.

1.5.4 Selección de estudios primarios

Esta cadena de búsqueda presentó un total de 42 artículos científicos candidatos de los cuales, a criterio de los investigadores, fueron seleccionados 5 como estudios primarios y son detallados en Tabla 2.

Tabla 2
Estudios primarios

Código	Título	Cita
EP1	IoT based monitoring and control system for home automation	(Pavithra & Balakrishnan, 2015)
EP2	Web-based power board using raspberry Pi	(Calugay, Santos, Santos, Tolentino, & Zhuo, 2016)
EP3	Arc discharge detection caused by short-circuit in AC power supply cord	(Abe, Fukagawa, Mizuno, & Yoshida, 2016)
EP4	Automatic Gas Controller	(Dhianeswar, Sumathi, & Joshitha, 2018)
EP5	Intelligent security system for residential and industrial automation	(Yerragolla, Pallela, & Gera, 2016)

EP1 (Pavithra & Balakrishnan, 2015) IoT based monitoring and control system for home automation

La investigación propone una implementación eficiente basada en IoT (Internet of Things), cuyo objetivo principal es monitorear y controlar electrodomésticos a través de la Web mediante un teléfono inteligente. Utilizando dispositivos portátiles como interfaz de usuario. La comunicación de los dispositivos se la realiza bajo protocolos de bajo consumo los cuales son Zigbee, Wi-Fi, entre otros. Se implementó la Raspberry Pi como sistema de servidor. En cuanto a características adicionales del artículo, el sistema tiene una protección de incendios, expulsando el humo en el caso de generarse. Cuya finalidad es alertar y enviar una imagen al teléfono inteligente del residente.

El inconveniente de la aplicación web propuesta es que al utilizar protocolos de comunicación open source estos pueden ser vulnerados y afectar al sistema en sí, lo cual enviaría falsas alertas al sistema domótico y a los usuarios que lo utilizan. La diferencia de la solución propuesta dentro de este trabajo, es la implementación de un chatbot, mediante una interacción basada en el lenguaje natural, permitiendo una mejor interacción con el usuario.

EP2 (Calugay, Santos, Santos, Tolentino, & Zhuo, 2016) Web-based power board using raspberry Pi

Este artículo trata sobre un dispositivo diseñado para monitorear y controlar la placa de alimentación, la cual puede evitar incendios causados por la salida de electricidad que al extenderse rápidamente a las estructuras de los edificios vecinos ocasionaría pérdidas sustanciales. Con este dispositivo se puede acceder a la placa de alimentación de forma remota a

través del servidor web. También cuenta con una Raspberry Pi la cual, recibe señales de la placa de relé conectada a los receptáculos que reciben electricidad. Fue seguro su uso debido a los contactores, que juegan un gran factor en la regulación del flujo de electricidad, evitando la sobrecarga de dispositivos que provocan incendios. Por lo tanto, la placa de alimentación basada en web que utiliza Raspberry Pi es de gran utilidad para las personas que tienden a olvidarse de desconectar sus dispositivos.

EP3 (Abe, Fukagawa, Mizuno, & Yoshida, 2016) Arc discharge detection caused by short-circuit in AC power supply cord

Este artículo propone un método de detección de descarga de arco basado en características extraídas de una serie de experimentos de laboratorio. La descarga de arco no intencional causada por un cortocircuito en el suministro de energía con aislamiento eléctrico y/o mecánico dañado puede provocar accidentes por incendio. Es difícil detectar este tipo de cortocircuito, porque lo que se requiere un sistema de prevención de riesgos en áreas residenciales y plantas.

EP4 (Dhianeswar, Sumathi, & Joshitha, 2018) Automatic Gas Controller

Este artículo está principalmente enfocado en prevenir situaciones drásticas como incendios y desperdicios de gas en cocinas, enfocado a personas de edad avanzada. Principalmente personas con problemas de pérdida de memoria. Esta investigación se basa en el corte automático de gas, al no estar presente la persona. Para ello se implementó un circuito y un sensor el cual detecta la presencia humana mediante la emisión de la temperatura corporal. Mientras la persona se encuentra presente, el sensor recibe la entrada y la activación de la válvula está en condición

desactivada. Cuando la persona abandona el lugar, la entrada del sensor es cero y, por lo tanto, el microcontrolador ATMEGA 16 comienza a funcionar.

EP5 (Yerragolla, Pallela, & Gera, 2016) Intelligent security system for residential and industrial automation

El presente artículo aborda la problemática en cuanto a los accidentes domésticos. La solución que plantea es un sistema de seguridad inteligente, tanto para aplicaciones domésticas como industriales mediante el control de forma remota conjuntamente con aplicaciones inalámbricas.

La interfaz de usuario general, supervisa los módulos de detección de entorno, diagnóstico, control de dispositivos y módulos robóticos, para lo cual se incorporaron múltiples sensores. También tiene por objetivo supervisar módulos robóticos con la ayuda de una interfaz RF inalámbrica y aplicaciones de telefonía móvil basadas en GSM para la supervisión remota. La solución propuesta se probó como un sistema externo de seguridad en ausencia de movilidad humana en el hogar, utilizándolo como un sistema de detección de intrusos. Al darse el caso de intrusos se alerta a través de mensajes y correo electrónico capturando las imágenes por la cámara incorporada.

1.5.5 Conclusiones al estado del arte

Dentro del análisis del estado del arte se puede determinar que en los estudios primarios se encontró artículos relacionados con el problema central, al hablar de accidentes domésticos y aparatos eléctricos. Cada artículo arrojó una solución viable, en cuanto a la disminución de posibles accidentes generados dentro del hogar, tales como quemaduras, y corto circuitos que pueden provocar incendios. Las soluciones propuestas se basaban en sensores, micro

controladores para controlar fugas de gas, dispositivos para controlar la electricidad y métodos para la detección de descargas eléctricas, algunas propuestas controlaban los aparatos de forma remota. Otra opción es controlar los artefactos eléctricos mediante una página web y la implementación de IoT; sin embargo, no se ha trabajado con la interacción entre el usuario y un chatbot mediante un lenguaje natural.

El aporte que se realizará con el presente proyecto, será el enfocarse en la problemática, para tratar de disminuir los accidentes domésticos mediante el uso de una aplicación que interactúe con el usuario de forma natural basado en un chatbot, cuyo fin será la manipulación de los dispositivos eléctricos para determinar su estado, tanto activo como inactivo.

1.6 Hipótesis de trabajo

Se planteó la siguiente hipótesis de trabajo: La utilización de un aplicativo para el control de artefactos eléctricos basado en un chatbot e interacción con lenguaje natural, disminuirá el riesgo de accidentes domésticos.

Variable Dependiente: Accidentes domésticos

Variable Independiente: Aplicativo basado en un chatbot e interacción con lenguaje natural.

1.7 Metodologías

Para el desarrollo y validación de la solución propuesta, se ha planteado una investigación de carácter cualitativa y cuantitativa. A continuación, se describen las metodologías utilizadas.

1.7.1 Investigación-Acción

Para el desarrollo del proyecto se aplicará la metodología denominada “Investigación-Acción”, propuesta por (Susman & Evered, 1978; Baskerville & Pries-Heje, 1999), el método tiene 5 fases que se comportan de forma cíclica. Las etapas son: diagnosticar, planificar, actuar, observar y reflexionar, como se muestra en la Figura 2. La ejecución del proceso comprende cinco fases que permiten generar nuevos conocimientos por cada iteración que se realice, de esta manera, cada vez se consigue un mejor acercamiento a la solución. Los ciclos son ejecutados iterativamente hasta tener la solución final del problema.

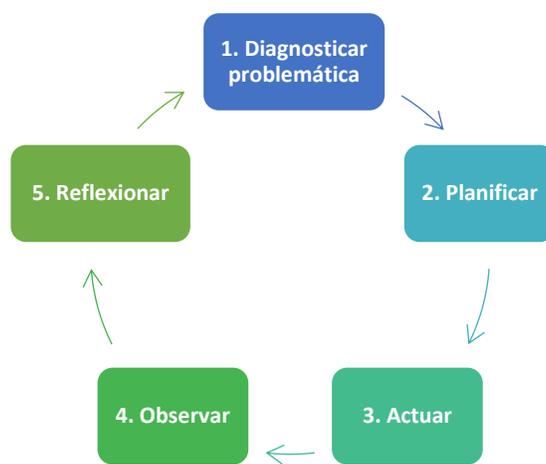


Figura 2. Ciclo de la metodología investigación – acción
Fuente: (Baskerville & Pries-Heje, 1999).

Dentro de las etapas se definirá lo que se realizó en cada una de ellas, adaptando la metodología a las necesidades del proyecto:

- **Diagnosticar:** Dentro de esta etapa se identifica la problemática y la situación actual de la misma.
- **Planificar:** Busca una solución para la problemática y determinar la viabilidad de la solución mediante un análisis del estado del arte.

- **Actuar:** Se desarrolla la aplicación, la cual se basará en software abierto.
- **Observar (Evaluar):** Probar la funcionalidad del aplicativo mediante pruebas de funcionalidad y rendimiento.
- **Reflexionar:** Se Concluye la efectividad del aplicativo mediante los resultados obtenidos en las pruebas realizadas.

Esta metodología de calidad, comparte características básicas de una buena investigación. Pring menciona cuatro características significativas de esta metodología (Sara Rodríguez, y otros):

- **Cíclica, recursiva:** Las etapas constituidas por esta metodología se repiten en una secuencia similar.
- **Participativa:** Todos los implicados se involucran al menos como participantes activos.
- **Cualitativa:** Utiliza más lenguaje que números.
- **Reflexiva:** El análisis crítico tanto en el proceso como en el resultado son partes relevantes de cada ciclo.

El diagnóstico y la planificación fueron abordados en el primer capítulo del presente trabajo, en tanto que las etapas: Actuar, Observar y Reflexionar, serán desarrollados en dentro del cuarto, quinto y sexto capítulo respectivamente.

1.7.2 Metodología basada en prototipos

Esta metodología se utilizó en conjunto con la investigación acción, en la fase actuar para el desarrollo de la solución propuesta, la cual consta de las siguientes fases:

- **Identificar requerimientos básicos:** Se implementó la identificación de requisitos bajo el estándar IEEE 830. En el cual se definió el alcance y funcionalidades del prototipo de software.
- **Diseño del prototipo inicial:** Se realizó un sketch board sobre el prototipo inicial.
- **Implementación del prototipo:** Una vez esquematizado el prototipo inicial, se desarrolló bajo las herramientas y uso de lenguajes de programación adecuados.
- **Pruebas y mejoras del prototipo:** Se realizaron las pruebas para identificar la funcionalidad y cumplimiento de los requisitos.

Esta metodología es óptima en cuanto al desarrollo rápido de aplicaciones bajo un reducido número de desarrolladores. Exige disponer de herramientas y lenguajes de programación adecuados dando mayor énfasis a la interfaz de usuario. El prototipo inicial de software se basaba en texto. posterior a ello, se incluyó mensajes de voz incorporando el modelo de reconocimiento automático de voz de la tecnología seleccionada.

1.7.3 Evaluación de tecnologías para el desarrollo de chatbots

Para la comparación de las tecnologías se tomaron como referencia varios estudios. Con el fin de incorporar los criterios necesarios, tales como características en general, costo, aspecto técnico y usabilidad de cada herramienta. Donde se procedió a evaluar y calificar a las plataformas escogidas. Por otro lado, se analizó la precisión en cuanto al reconocimiento de voz y funcionalidad del sistema mediante diferentes pruebas realizadas, adaptado el estudio realizado por SAP Conversational AI a las necesidades del presente trabajo.

1.7.4 Validación de la solución propuesta

Para la validación de la solución propuesta se llevó a cabo una encuesta. La cual es de tipo cualitativo y cuantitativo. Dentro de la encuesta se aplicaron las siguientes pruebas: la primera parte consta de pruebas de funcionalidad, donde se midió el tiempo en ejecutar una orden, ya sea en texto como en voz. La segunda prueba obtiene el grado de aceptación de la solución propuesta. Mientras que la última prueba consta de un test de usabilidad, adaptando la escala de usabilidad del sistema (SUS) el cual consta de varias preguntas y respuesta. Dentro del quinto capítulo se abordará con mayor profundidad la validación de la solución.

CAPÍTULO II

MARCO TEÓRICO

En este capítulo se abordarán temáticas con respecto al problema planteado tales como los accidentes domésticos. De igual forma se tratará temáticas con respecto a la solución propuesta, que abarcará temas como Internet de las Cosas (IoT), Inteligencia Artificial y, sobre todo, como los asistentes virtuales o chatbots pueden ayudar al ser humano al momento de automatizar tareas cotidianas. A demás se hablará de plataformas que pueden ayudar con el desarrollo de chatbots, las cuales serán utilizadas para su posterior comparación.

2.1 Accidentes domésticos

Accidente se define como cualquier acontecimiento que es provocado por una acción violenta e imprevista causada por un agente exterior y da lugar a una lesión física. Los accidentes domésticos son aquellos que suceden dentro del hogar o en sus alrededores por ejemplo: escaleras, patio, jardín, garaje, vestíbulos, etc. (Minda Almagor, 2013). Existen varios tipos de accidentes domésticos los cuales se mencionarán a continuación:

2.1.1 Tipos de accidentes domésticos

Los accidentes más habituales en el hogar son: mordeduras de animales, heridas, ahogamiento, choque eléctrico, traumatismos, intoxicaciones y sofocación. De los accidentes antes mencionados, se tomará en cuenta el choque eléctrico ya que es el que se relaciona con la problemática.

2.1.2 Choque eléctrico

Al tratar sobre el choque eléctrico, se anticipa a decir que la sobrecarga en los circuitos eléctricos o el mal uso de los dispositivos eléctricos pueden provocar incendios en los hogares.

Además de causar quemaduras en la persona, la electricidad manejada incorrectamente puede ocasionar conmociones e incluso la muerte, como consecuencia de fuertes descargas de la instalación eléctrica, ya que esta cuenta con altos niveles de tensión eléctrica que alimentan los aparatos electrodomésticos desde 125 o 220 voltios, los cuales son peligrosos tanto para niños y adultos. A continuación, se mencionarán algunas propuestas tecnológicas para evitar los accidentes domésticos. (Minda Almagor, 2013) .

2.2 Nuevas propuestas tecnológicas

Como consecuencia las oportunidades de reducir los accidentes domésticos son muy grandes: probablemente se podría reducir a la mitad con medidas sistemáticas y rigurosas, algunas de ellas son: desconectar manualmente los electrodomésticos, la sustitución de equipos por otros más modernos y eficientes, el buen uso de los mismos mantendría seguro a los usuarios de choques eléctricos, librándose de las graves consecuencias que generan estos incidentes dentro del hogar.

Sin duda estos métodos constituyen una alternativa práctica, mejorando la eficiencia del sistema eléctrico. Sin embargo el uso de equipos automáticos de desconexión de carga, la implementación de sistemas inteligentes tales como las smart homes, smart meters y smart appliances, entre otras, pueden ser una opción para disminuir el riesgo en cuanto accidentes domésticos (Orejuela, Arias, & Aguila, 2015). La siguiente temática ayudara a comprender de mejor manera como el crecimiento de la tecnología, equipos e internet pueden ayudar al ser humano en su vida cotidiana. Para entender un poco más se hablará del Internet de las Cosas (IoT).

2.3 Internet de las Cosas (IoT)

El gran avance en cuanto a tecnología ha permitido un mayor acceso a internet. Donde la mayoría de personas se conectan a este medio con fines informativos, entretenimiento, e incluso laborales y económicos. Cada vez existe mayor cantidad y variedad de dispositivos conectados a internet. Tanto como las personas y las cosas cotidianas del entorno, pueden conectarse a la red para aprovechar sus máximos beneficios. Internet de las cosas (IoT siglas en ingles), será la evolución de la siguiente etapa al hablar de la conectividad masiva de objetos.

Actualmente, algunos aparatos eléctricos inteligentes utilizan protocolos de comunicación IoT, para poder conectarse en red. Estos dispositivos son controlados desde cualquier parte del mundo o de manera local, utilizando comando de texto, voz o gestos para su monitoreo (Hamdan, Shanableh, Zaki, Al-Ali, & T, 2019). El uso e implementación de Internet de las Cosas (IoT) permitirá la integración e interconexión de dispositivos inteligentes, los cuales compartirán datos, recursos e información útil, por medio de una red, proporcionando servicios para mejorar la calidad de vida del ser humano en el aspecto social, económico, cultural, ambiental, etc. (Lee & Lee, 2015; Stergiou, Psannis, Kim, & Gupta, 2018).

2.3.1 Arquitectura IOT

La implementación de IoT contiene dispositivos heterogéneos con sensores interconectados a través de una red como lo representa la Figura 3. Los dispositivos en IoT son identificables de forma única. Las puertas de enlace se implementan para conectar dispositivos IoT al mundo exterior para el suministro remoto de datos y servicios a usuarios de IoT (Khan & Salah, 2018).

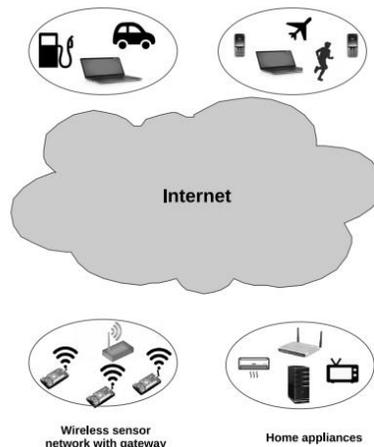


Figura 3. Elementos de IoT
Fuente: (Khan & Salah, 2018)

2.3.2 Protocolos y Estándares

La Figura 4 constituye una arquitectura en capas con los protocolos comúnmente utilizados en IoT para aplicaciones y mensajería, enrutamiento / reenvío, dispositivos físicos y aquellos para administración de claves y autenticación (Khan & Salah, 2018). El Internet de las cosas permitió llevar objetos físicos al mundo cibernético, lo cual fue posible a diferentes tecnologías como NFC, RFID y código de barras 2D, con el fin de identificar a los objetos a través de internet. Está integrada tecnología de sensores, radiofrecuencia, siendo la red presente en los recursos omnipresentes de hardware. Implica tecnologías sofisticadas de redes informáticas y comunicación externas, tales como la comunicación, transmisión de forma remota, análisis de inteligencia información y de control (Madakam, Ramaswamy, & Tripathi, 2015).

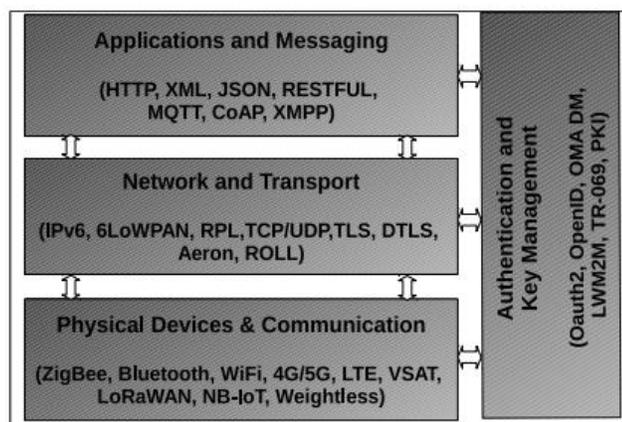


Figura 4. Protocolos y Estándares comúnmente de IoT

Fuente: (Khan & Salah, 2018)

2.3.3 Aplicaciones IOT

El Internet de las Cosas puede aplicarse a varios ámbitos, todo dependiendo de la iniciativa, creatividad e ingenio de los desarrolladores en búsqueda de nuevos productos y servicios. En la actualidad la factibilidad de implementar IoT en distintas áreas puede ser mayor o menor, constituyendo formas en las que eventualmente se puede usar esta plataforma. Si bien es cierto, nuevas oportunidades y mercados permitirán a las empresas atraer cambios positivos o mejorar los servicios existentes. Basado en las tendencias tecnológicas se identificó tres categorías de IoT para aplicaciones (Alcaraz, 2014; Lee & Lee, 2015; Stergiou, Psannis, Kim, & Gupta, 2018):

- Monitoreo y control.
- Big data y Análisis de negocios.
- Intercambio de información y colaboración.

De las cuales se centrará en el monitoreo y control, ya que es lo que la aplicación realizará al momento de interactuar con los dispositivos electrónicos.

2.3.4 Monitoreo y control

Los sistemas de monitoreo y control recopilan datos sobre el rendimiento del equipo, el uso de energía y las condiciones ambientales. Permiten a los gerentes y controladores automatizados rastrear constantemente el rendimiento en tiempo real en cualquier lugar y en cualquier momento. Un ejemplo de estos sistemas es en la domótica en las llamadas casas inteligentes y como estas pueden ayudar al ser humano en actividades cotidianas (Lee & Lee, 2015).

2.3.5 Domótica

Se sabe que la domótica está a la vanguardia de la innovación con respecto a los sistemas de monitoreo y control de IoT, ya que permite a los usuarios ajustar las luces, controlar el clima, administrar el sistema de seguridad, recibir notificaciones automáticas de eventos e incluso bloquear y desbloquear puertas a través de una computadora, tableta o teléfono inteligente. Todo esto se lo puede realizar con la ayuda de diversas tecnologías inalámbricas, las cuales permiten controlar el entorno doméstico desde cualquier lugar y así reducir los accidentes que ocurren regularmente, tomando medidas al respecto de forma más rápida o automatizando tareas. La tecnología IoT se utiliza para presentar ideas innovadoras y un gran crecimiento para viviendas inteligentes y así mejorar el estilo de vida, lo cual mencionaremos a continuación (Lee & Lee, 2015; Pavithra & Balakrishnan, 2015).

2.3.6 Vivienda Inteligente

La vivienda inteligente comprende tres factores los cuales son la evolución de la tecnología, los cambios sociales presentes y las oportunidades de negocios al momento de generar nuevos productos y servicios que beneficien a los usuarios. El aumento de seguridad en las viviendas y lugares más confortables eran objetivos de las primeras aplicaciones, ahora el ahorro energético,

la mejora de la salud e higiene son los nuevos objetivos primordiales. La vivienda puede tener un sin número de sistemas y aplicaciones complementando las instalaciones básicas permitiendo al usuario interactuar con estos controlando y ajustando a sus necesidades dicho entorno. (Junestrand, Passaret, & Vázquez, 2004; Alcaraz, 2014).

Durante los últimos años, gracias a los avances de métodos computacionales, se ha desarrollado sistemas de control conocidos como Smart Home, permitiendo monitorear, implementar seguridad, entretenimiento entre otras funciones, lo cuales utilizan inteligencia artificial. Mientras mayor sea el acceso a la tecnología, se necesitarán sistemas que contribuyan a la captación de datos. El papel de inteligencia artificial permitirá procesar datos, ya sea en casa, en espacios públicos, en infraestructuras, etc. Con el único objetivo de optimizar y personalizando productos y servicios de acuerdo a cada usuario, para mejorar la calidad de vida del ser humano. El siguiente tema hablará con mayor profundidad sobre la inteligencia artificial (Solans, 2009; Li, Li, Mak, & Tang, 2016).

2.4 Inteligencia Artificial (IA)

La Inteligencia Artificial nació de un estudio filosófico, donde su principal cuestión es si un objeto puede imitar el pensamiento humano. El principal objetivo de la IA es el poder imitar el pensamiento y la capacidad de aprender como lo hace el ser humano, desarrollar seres artificiales capaces de cumplir tareas inteligentes con base a ciertos principios. “En Ciencias Computacionales la Inteligencia Artificial se define como máquinas que manifiestan cierta forma de pensamiento”. Dentro de la Inteligencia Artificial (IA) el término heurística es ampliamente usado, el cual describe la relación con el sentido común en la resolución de problemas, un

sistema dotado de esto puede definir un problema y definir la mejor estrategia para formular una solución (Serna, Acevedo, & Serna, 2017).

2.5.2 Principios fundamentales de la Inteligencia Artificial

La IA tiene varios principios fundamentales para el trabajo en todas las disciplinas en las que se lo requiera implementar, los cuales se describirán a continuación (Serna, Acevedo, & Serna, 2017):

- **Autoaprendizaje:** La capacidad de auto aprender es el principio más importante, cuya finalidad es determinar soluciones rápidas, eficientes y analizar los datos al momento de asignar una tarea. Las aplicaciones de redes neuronales pueden crear caminos, los cuales ayudaran en la toma de decisiones, en base a lo que haya aprendido.
- **Interacción con el usuario:** Capacidad de interactuar con el usuario a través de una interfaz sencilla, entendiendo comandos de voz o lectura de ilustraciones. La IA debe ser capaz de dar respuesta clara y sólida al llevar a cabo una tarea asignada sin la mayor dificultad o problema posible.
- **Reacción en tiempo real:** Debe ser eficaz para trabajar en tiempo real respondiendo de forma inmediata a cada tarea asignada.
- **Autonomía:** Debe trabajar por cuenta propia sin necesidad de que el usuario intervenga.

Cualquier sistema con Inteligencia Artificial (IA) debe ser capaz de interpretar el lenguaje de varias formas, tanto escrito, oral, señas, o la interpretación de imágenes, para traducirla a lenguaje de máquina. También es fundamental que este sea adecuado para comunicarse con el usuario,

usando el mismo lenguaje natural que es el tema que se hablara a continuación. (Serna, Acevedo, & Serna, 2017).

2.5.3 Lenguaje natural

El lenguaje natural es un conjunto de signos, símbolos orales y escritos, es decir, las palabras cotidianas que se utilizan para expresar ideas y que permiten la comunicación con las demás personas. El lenguaje natural es la lengua o idioma que permite a los seres humanos comunicarse entre sí, este puede ser español, inglés, francés, entre otros. El lenguaje natural va evolucionado y perfeccionándose con el tiempo por lo que es utilizado para analizar situaciones complicadas y razonar cuidadosamente cada una de ellas (Vásquez, Quispe, & Huayna, 2009).

Una característica importante del lenguaje natural es que la comunicación entre las personas surja de modo espontáneo. En cambio, los lenguajes propios que utilizan protocolos matemáticos inflexibles, los manejan las computadoras para poder comunicarse entre sí. Se han realizado algunas investigaciones que garantizan que el lenguaje natural es el resultado del aprendizaje. Se puede concluir que las máquinas también pueden aprender, pero para esto la máquina tiene que interactuar con el hombre siendo el siguiente tema a tratar (Ramos & Velez, 2016).

2.5.4 Interacción hombre-máquina (HCI)

“La interacción hombre computadora (HCI) es el estudio de la manera en que la tecnología informática influye en el trabajo y las actividades humanas”. Durante las últimas décadas los profesionales e investigadores de la materia han logrado mejorar el diseño de interfaces gráficas de usuario, pero con el auge y rápido crecimiento de la tecnología, ha dado la apertura al desarrollo de interfaces de usuario basadas en el lenguaje natural. Lo cual implica nuevas oportunidades al momento de interactuar con servicios digitales dentro del campo de HCI. El

avance tecnológico ha determinado que el reconocimiento de voz es una de las técnicas más naturales y buscadas en la interacción de computadoras y dispositivos en red. Ampliamente aceptado como el futuro de la interacción con computadoras y aplicaciones móviles. (Abdul-Kader & Woods, 2015; Følstad & Brandtzæg, 2017).

2.5.5 Inteligencia artificial, asistentes virtuales y lenguaje natural

La Inteligencia Artificial, se la puede implementar en cada ámbito de trabajo del ser humano, las máquinas inteligentes son capaces de realizar un amplio número de tareas, donde los campos más destacados son la simulación del cerebro humano, el procesamiento de lenguaje natural y redes neuronales. Los avances en Inteligencia Artificial han proporcionado que la interacción en lenguaje natural pueda ser una opción viable para conectar agentes de máquinas y usuarios. Mediante el procesamiento del lenguaje natural la Inteligencia Artificial será capaz de interpretar y leer el lenguaje de varias formas, para traducirla a lenguaje de máquina (Dale, 2016; Serna, Acevedo, & Serna, 2017; Følstad & Brandtzæg, 2017; Khan & Salah, 2018).

Es fundamental que este sea adecuado de comunicarse con el usuario, usando el mismo lenguaje natural según su idioma, siendo hábil en la interpretación de las diversas tareas o problemas que le sean transmitidos. Dando una respuesta idónea en caso de no entender lo que el usuario intenta comunicar. Dentro de un futuro no muy lejano, las interfaces basadas en lenguaje natural marcaran un precedente al interactuar de forma natural con los usuarios (Dale, 2016; Serna, Acevedo, & Serna, 2017; Følstad & Brandtzæg, 2017; Khan & Salah, 2018). Dentro del procesamiento de lenguaje natural, durante los pasados años se ha proporcionado la ayuda de asistentes virtuales inteligentes, algunos de ellos conocidos como asistentes digitales, interfaces de conversación o comúnmente conocidos como chatbots, el cual tiene por objetivo proporcionar

respuestas al usuario mediante el uso de un lenguaje natural. La siguiente temática, abordará lo que es un chatbot (Khanna, Das, Pandey, Hussain, & Jain, 2016; Dale, 2016; Følstad & Brandtzæg, 2017).

2.6 Chatbot

¿Pueden las máquinas pensar?, Alan Turing fue el primero en cuestionarse y conceptualizar el término chatbot. “Desde Turing, la tecnología de chatbot ha mejorado con los avances en el procesamiento del lenguaje natural y el aprendizaje automático” (Cahn, 2017). En algunos años los chatbots estarán presentes dentro de las actividades cotidianas de la persona, solventando cualquier inquietud respecto a una temática en específico. Estos agentes conversacionales podrán estar integrados con sistemas de diálogo basados en voz, tomando como precursor Eco de Amazon. El uso del lenguaje natural como interfaz y los avances en la Inteligencia Artificial serán primordiales para interactuar con estos agentes (Følstad & Brandtzæg, 2017).

Un chatbot, o agente conversacional, es un sistema de diálogo humano computadora en línea, es decir una interfaz de usuario conversacional, que interactúa con los usuarios usando un lenguaje natural. Puede conocer un determinado dominio o tema, el cual responde a las siguientes funciones como: Comprensión del lenguaje natural, extracción del significado y la intención del enunciado, análisis del tono y el sentimiento, respuesta en un forma de lenguaje natural (Toledo, 2018; Cahn, 2017; Greyling, 2019). Las funciones básicas que se espera de un chatbot son:

- **Agente dialógico:** Debe comprender al usuario, ya sea la entrada textual u oral, las cuales serán analizadas mediante herramientas de procesamiento de lenguaje natural, generando respuestas apropiadas.

- **Agente racional:** Este debe tener acceso a una base externa de conocimiento y sentido común, proporcionando una función de competencia, almacenando información específica del contexto.
- **Agente encarnado:** Implementación de lenguaje con el fin de crear y dar funcionalidad de presencia al chatbots con el propósito de generar mayor confianza a los usuarios.

El propósito de un chatbot es interactuar y estimular una conversación con un determinado usuario sin ningún compromiso. Con los avances en tecnología y la mejora constante de las técnicas de extracción de datos y aprendizaje automático, estos han logrado a ser prácticos y de gran potencial comercial (Da Rocha Loureiro, 2017). A continuación, se mencionará algunas características principales de los chatbots.

2.6.1 Historia de los chatbots

La siguiente Figura 5 muestra una línea de tiempo en cuanto al desarrollo de los chatbots. Turing fue el primero en cuestionarse si las máquinas pueden pensar, realizando la prueba de Turing, con el fin de determinar si el humano y la máquina son indistinguibles. Joseph Weizenbaum creó el primer chatbot, ELIZA, haciéndose pasar por un psiquiatra, fue una imitación muy cercana al ser humano. Se basaba en el reconocimiento de palabras claves y preguntando sobre ellas. (Dones Téllez, 2019; Cahn, 2017).

Kenneth Colby creó PARRY, un bot esquizofrénico. A.L.I.C.E fue la creación de Richard Wallace, este bot generaba respuestas más rápidas mediante la coincidencia de patrones. Alice fue tres veces ganadora del premio Loebner. Clippy, fue el primer bot de Microsoft, ofreciendo asistencia a los usuarios de Microsoft Office. Dentro de esos años también apareció Albert One, el cual imitaba el comportamiento humano, ganando el premio Loebner. Echo y Alexa de

Amazon, Siri de Apple y Cortana de Microsoft son los modernos chatbots (Dones Téllez, 2019; Cahn, 2017).

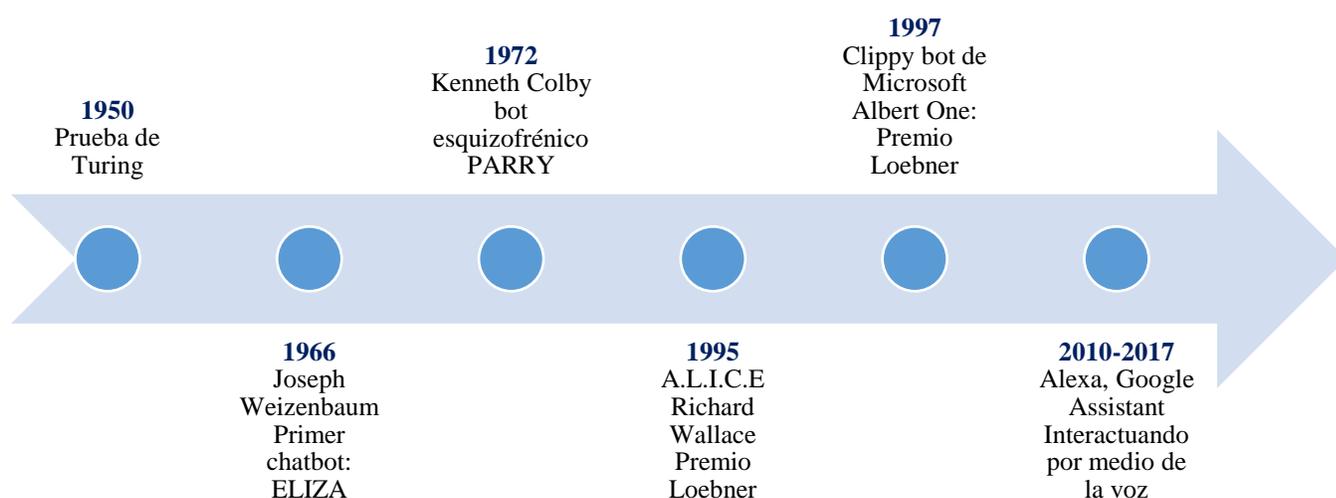


Figura 5. Línea de tiempo chatbots.

Hoy en día, los chatbots ofrecen una alternativa para realizar las tareas, su uso cada vez más está incrementando entre los usuarios, ofreciendo grandes ventajas a las compañías a la hora de comercializar sus servicios. Actualmente los chatbots pueden usarse a través de diversos canales, webs, smartphones, redes sociales, etc., los cuales se basan en técnicas de procesamiento de lenguaje natural para tener mayor precisión dentro del significado y la intención de la oración, y machine learning, para entrenar y ser capaz de reaccionar. (Dones Téllez, 2019; Cahn, 2017).

2.6.2 Características de los chatbots

Las principales características que diferencian a los agentes conversacionales son las que se describen a continuación (Cobos Torres, 2013):

- **Autonomía:** Capaz de interactuar basándose en su experiencia, siendo altamente capacitado para adaptarse.
- **Sociabilidad:** Comunicarse con otros agentes.
- **Racionalidad:** Debe realizar lo correcto a partir de los datos de entrada que percibe del entorno.
- **Reactividad:** Actúa como resultado de cambios en su entorno.
- **Pro-actividad:** Programados para comprender de mejor manera el contexto de la conversación.

Para que un asistente automatizado o chatbot funcione, cualquier conversación debe tener los siguientes elementos (Greyling, 2019):

- **Contexto:** El cual debe ser establecido para comprender la intención, a donde se dirige la conversación.
- **Continuidad / Vertical y Horizontalmente:** Llamamos continuidad vertical al iniciar una conversación en la mañana y continuarla en la tarde. La continuidad es horizontal en todas las modalidades, es decir iniciar la conversación por medio de voz en la mañana y en la tarde por medio de texto.
- **Diálogo dirigido:** Como humanos dirigimos el diálogo de la conversación. Es importante que el chatbot entienda el contexto dentro de la conversación y a donde dirigirla.

2.6.3 Criterios para la evaluación de chatbots

El autor (Cahn, 2017) expone diferentes criterios para evaluar el rendimiento de un chatbot, dentro del presente trabajo se tomó como referencia la recuperación de datos y la experiencia del usuario para la validación de la solución propuesta, los cuales se describirán a continuación:

- **Recuperación de datos:** Los chatbots tienen funciones específicas. El papel de los evaluadores es hacer las preguntas y solicitudes al chatbot, con la finalidad de evaluar la efectividad, precisión y exactitud, asignando un puntaje en relación con las respuestas correctas del chatbot.
- **Experiencia del usuario:** El evaluador debe encuestar a los usuarios, clasificando a los chatbots en función de la satisfacción del usuario y la usabilidad.

2.7 Arquitectura y diseño de un chatbot

Según (Zuñiga & Humberto, 2018) la arquitectura básica de un chatbot se divide en tres componentes desplegados en la siguiente Figura 6, las cuales se describirán a continuación:

- **Interfaz de usuario:** Entradas de información hacia el chatbot, puede ser en forma de texto u oral.
- **Motor de inferencia:** El cual analiza información, establece los objetivos y obtiene la respuesta de acuerdo a la base de conocimiento. El motor de inferencia usa dos elementos (hechos o evidencias) y el conocimiento (reglas almacenadas en la base de conocimiento) obteniendo nuevas conclusiones o hechos.
- **Base de conocimiento:** Su contenido es el conocimiento del experto humano, en función de plantillas, patrones y reglas.

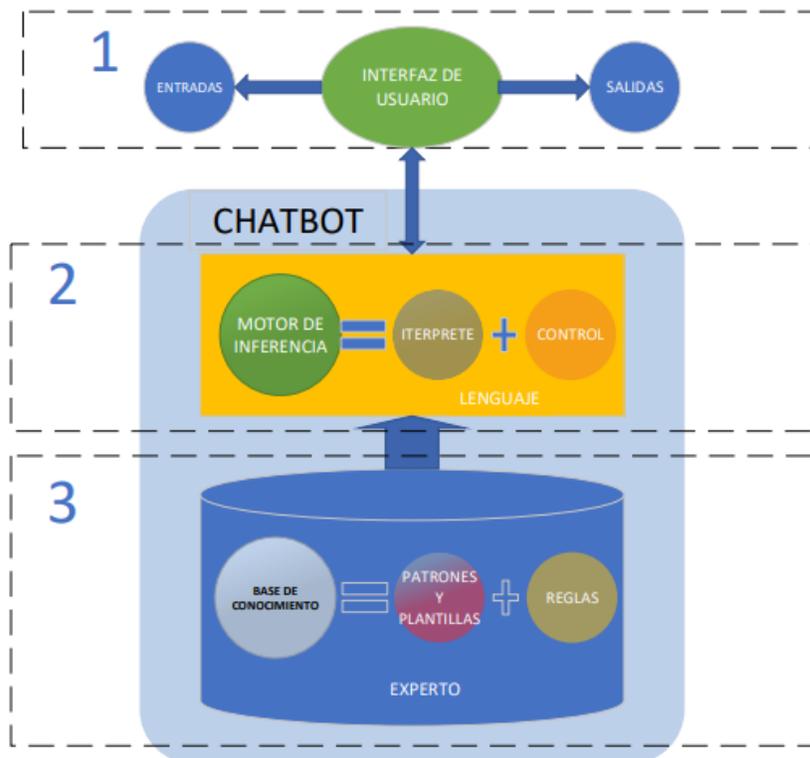


Figura 6. Arquitectura básica de un chatbot

Fuente: (Zuñiga & Humberto, 2018)

2.7.1 Conversión de voz a texto

El habla es uno de los modos de comunicación humana más natural, sofisticada y poderosa, del ser humano. Este medio de comunicación ocurre en diferentes niveles: semántico, lingüístico, articulatorio y acústico. Los grandes avances en la tecnología han determinado que el reconocimiento de voz es una de las técnicas más naturales y buscadas mayormente en la interacción de computadoras y dispositivos en red. Al incorporar el procesamiento del habla, es más fácil poder comunicarse con dispositivos, como resultado la interacción con chatbots se ha vuelto muy común en la actualidad. (Abdul-Kader & Woods, 2015; Salamanca, Eugenia, & Macq, 2019; Cahn, 2017).

La conversión de texto a voz comienza con el proceso de reconocimiento automático de voz con sus siglas en inglés ASR, cuyo objetivo es convertir la voz a texto y extraer información. El texto resultante es tratado, extrayendo el significado de las palabras de un extenso vocabulario al cual se lo denomina reconocimiento continuo de voz de vocabulario grande y sus siglas en inglés son LVCSR. Las mejoras en LVCSR se miden a través de las siguientes dimensiones (Salamanca, Eugenia, & Macq, 2019; Cahn, 2017):

- **Tamaño del vocabulario:** Manejo de extensas palabras en diferentes idiomas.
- **Independencia del hablante:** Es relevante solo si el chatbot utiliza la identidad del orador para generar respuestas específicas.
- **Coarticulación:** Procesamiento continuo de palabras.
- **Manejo del ruido:** Filtración del ruido.
- **Micrófono:** Procesamiento del habla a diferentes distancias del micrófono.

2.7.2 Modelo de proceso de reconocimiento automático de voz (ASR)

Un sistema ASR permite procesar e interpretar el habla humana, convirtiendo la señal del habla en palabras. La siguiente Figura 7 muestra la arquitectura de un chatbot que incorpora reconocimiento de voz. Para ello el sonido es grabado en un micrófono, donde la señal es discretizada, con una frecuencia de muestreo (16kHz). Dando paso a la segmentación de voz/no voz, donde el sistema ASR distingue unidades básicas de voz denominados fonemas. Puede generarse errores dentro del modelo al existir ruido de fondo ya que este puede clasificarse erróneamente como voz (Cahn, 2017; Salamanca, Eugenia, & Macq, 2019).

Posterior a ello se procede a la extracción de características acústicas. Las observaciones acústicas se extraen en marcos temporales de longitud uniforme en 25 milisegundos. Cada cuadro

es decodificado, asignando las palabras correspondientes más probables, a los vectores de características acústicas. Esto se lo realiza con la ayuda del modelo oculto de Markov (siglas en inglés HMM) (Cahn, 2017; Salamanca, Eugenia, & Macq, 2019).

Dentro del reconocimiento automático de voz, el modelo de Markov evalúa $P(X | W)$, dada una palabra, la probabilidad de percibir un sonido. Los sonidos que no sean relevantes entorno a la grabación son eliminados. Finalmente se procede a realizar un post procesamiento, basado en la regla de Bayes, donde da una lista de las probables secuencias de palabras, el ASR escoge el más probable, dejando a las otras hipótesis almacenadas, para ser usadas por algoritmos de aprendizaje y refuerzo, pretendiendo aprender y corregir errores en un futuro. En resumidas palabras el proceso ASR identifica y procesa el habla para reconocer las palabras o frases del audio (Cahn, 2017; Salamanca, Eugenia, & Macq, 2019).

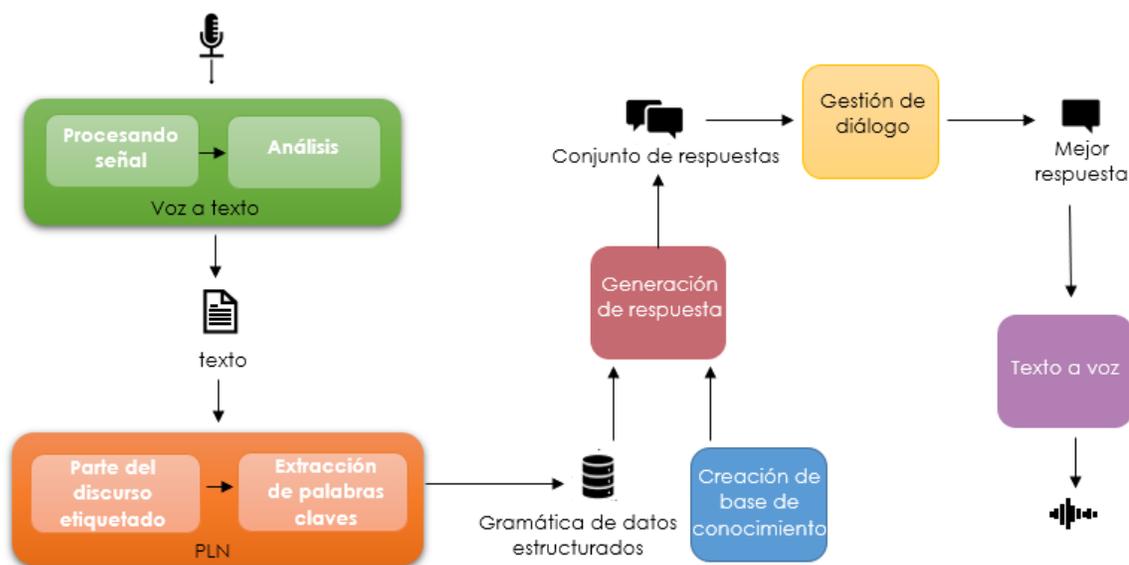


Figura 7. Arquitectura de un chatbot basado en voz.

Fuente: (Salamanca, Eugenia, & Macq, 2019)

Al trabajar con un sistema de reconocimiento automático de voz se presentan varios retos los cuales son (Cahn, 2017; Sandoval, 2019):

- Las personas hablan comúnmente con errores gramaticales y semánticos.
- El habla no contiene pausas entre cada palabra, estas aparecen después de cada oración.
- Existen varios dialectos por región.
- En el lenguaje existen palabras denominadas homófonas las cuales se pronuncian igual, pero se escriben y tienen diferente significado.

Cabe mencionar también que, si el hablante intenta comunicar una palabra, el resultado será diferente debido a varios factores como lo son el ruido ambiental, estado emocional, cansancio, distancia del micrófono, entre otras (Cahn, 2017).

2.7.3 Procesamiento de lenguaje natural (PLN)

El uso de lenguajes naturales facilita el desarrollo de programas y modelos que ayuden a la interacción hombre-máquina. El procesamiento de lenguaje natural (PNL) también conocido como computación lingüística, permite que la máquina interprete el lenguaje humano. Su principal objetivo es extraer información y significado de los idiomas escritos y hablados, tomando el texto no estructurado del reconocimiento automático de voz, para crear una estructura de datos gramaticales. Las cuales son procesadas por la unidad de gestión de diálogo o administrador del diálogo para así producir una representación estructurada del texto, tanto de manera oral o textual (Salamanca, Eugenia, & Macq, 2019; Saygin, Cicekli, & Akman, 2000; Cahn, 2017).

La arquitectura del sistema de procesamiento de lenguaje natural está sustentada por una definición de lenguaje natural, sus niveles son: fonológico, morfológico, sintáctico, semántico y pragmático. Los cuales se detallarán a continuación (Vásquez, Quispe, & Huayna, 2009):

- **Nivel Fonológico:** Es el cómo las palabras se relacionan con los sonidos.
- **Nivel Morfológico:** Es el cómo las palabras son construidas a partir de morfemas (unidad mínima de lenguaje que posee significado léxico o gramatical).
- **Nivel Sintáctico:** Es el cómo las palabras pueden unirse para formar oraciones, determinando el papel estructural que cada palabra juega en la oración.
- **Nivel Semántico:** Es el significado de las palabras y de cómo estos significados se unen para dar sentido a la oración, independiente del contexto.
- **Nivel Pragmático:** Es el cómo las oraciones son usadas en distintas situaciones y de cómo el uso afecta al significado de las oraciones. Determinando el contexto del que se habla al afectar inmediatamente oraciones anteriores.

La arquitectura de un sistema del procesamiento de lenguaje natural se representada en la siguiente Figura 8, donde las oraciones son analizadas en sentido morfológico y sintáctico, para determinar si las palabras contienen morfemas y si está bien estructurada la oración (Vásquez, Quispe, & Huayna, 2009). Posterior a ello para ser analizado de forma sintáctica y lexicográfica, el denominado scanner es el encargado de identificar componentes léxicos definidos anteriormente, para pasar por el denominado parser, el cual verifica si la estructura gramatical se encuentra en orden.



Figura 8. Arquitectura de un sistema PNL

Fuente: (Zuñiga & Humberto, 2018)

Las oraciones pasan hacer analizadas de forma semántica y se asigna el significado de estas expresiones lógicas. Las oraciones analizadas a nivel pragmático son inspeccionadas todas juntas, para posterior a ello obtener la expresión final (Vásquez, Quispe, & Huayna, 2009).

Comprensión del lenguaje natural (NLU): Rama del procesamiento de lenguaje natural (PNL), cuya finalidad es capacitar a un sistema para comprender el significado e intención de una determinada frase. Detecta dominios de diálogo mediante el análisis de expresiones por parte del usuario. Estas expresiones son clasificadas de acuerdo a la intención del usuario y asociadas a una plantilla semántica o base de conocimiento específica del dominio (Yang, y otros, 2017).

Los agentes conversacionales se encuentran estrictamente ligados con tecnologías de procesamiento de lenguaje natural, basándose en algoritmos de extracción de información, toma de decisión y aprendizaje. Dentro del NLU los métodos de etiquetado secuencial como modelos ocultos de Markov (HMM) y campo aleatorio condicional (CRF) se utilizan en las tareas de etiquetado de ranuras o entidades. A continuación se mencionara algunos métodos que ayudan a extraer información semántica y el significado del lenguaje tanto escrito como oral (Robalino, 2017; Cahn, 2017; Zuñiga & Humberto, 2018).

2.8 Métodos y técnicas para extraer información

2.8.1 Reconocimiento de acto de diálogo

Determinar la acción o función del texto, si es sugerencia, pregunta, orden, etc., para extraer el significado del lenguaje natural. Los datos de entrenamiento se etiquetan con la función de la oración, construyendo un modelo estadístico de aprendizaje automático. El modelo utiliza una serie de características para clasificar las oraciones (Cahn, 2017).

2.8.2 Identificación de intenciones

Dentro de la comprensión del lenguaje hablado en lugar de preguntar si la función del enunciado que ha generado el usuario es una pregunta o respuesta, se pregunta si la función contiene una acción como por ejemplo el cancelar un vuelo o realizar una reserva. La identificación de intenciones ha sido utilizada de manera más destacada por los bots de call center (Cahn, 2017).

Varios algoritmos de aprendizaje automático utilizados para la clasificación de acto de diálogo (DA), se utilizan para la identificación de intenciones. Las conversaciones basadas en intenciones son algoritmos de procesamiento de lenguaje natural en los que se utilizan intenciones y entidades para llevar a cabo la conversación, al reconocer los nombres y verbos se realiza una referencia cruzada con su diccionario para posterior a ellos el bot pueda validar la acción (Pérez Vallejo, 2019).

La Figura 9, muestra el módulo de clasificación de entidades e intenciones donde se identifica a partir de un mensaje cuales son las posibles intenciones, pasando al módulo de reconocimiento de entidades, donde se extrae bits estructurados del mensaje. El generador de respuesta procesa la solicitud del usuario. Luego selecciona la respuesta por medio de un selector el cual califica a

todos los candidatos de respuesta y emite su resultado conforme a lo que es mejor para el usuario (Rahman, Al Mamun, & Islam, 2017).

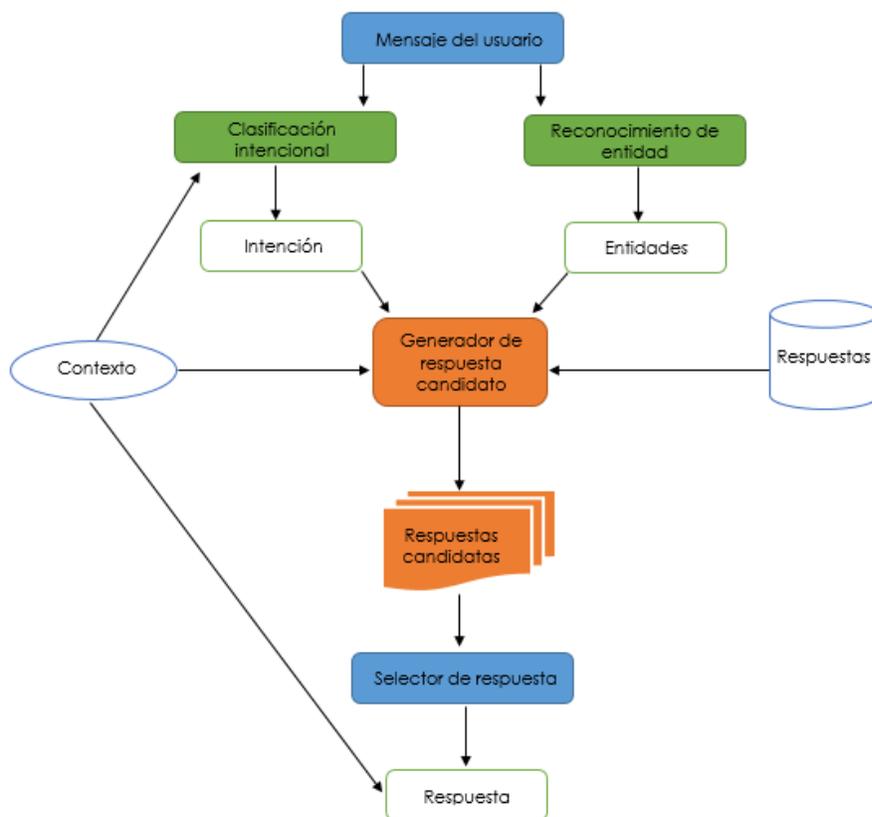


Figura 9. Identificación de entidades e intenciones

Fuente: (Rahman, Al Mamun, & Islam, 2017)

2.8.3 Extracción de la información

Existen muchas técnicas para llevar a cabo la extracción de información. Para ello la oración es dividida en tokens, representando sus componentes: palabras, signos de puntuación, números, etc. Dichos tokens se analizan mediante una serie de técnicas las cuales son: Bolsa de palabra, análisis semántico latente, expresiones regulares, etiquetado de parte del discurso (POS), reconocimiento de entidad nombre relación, etiquetado de rol semántico y creación de estructura

de datos gramaticales. Estas técnicas tienen objetivos similares: La entrada (voz a texto) se divide en palabras separadas, las cuales pasan a ser etiquetas como partes del discurso de acuerdo a su posición y palabras adyacentes en la oración. A continuación se mencionará las que se usan con más frecuencia (Salamanca, Eugenia, & Macq, 2019; Cahn, 2017).

- **Bolsa de palabras:** Las palabras son comparadas con la base de datos de conocimientos del bot para encontrar palabras claves similares. No requiere conocimiento de sintaxis, pero, no es lo suficientemente preciso para resolver problemas más complejos.
- **Expresiones regulares:** Las oraciones o enunciados se tratan como expresiones regulares, comparándose con la base de datos de conocimientos del bot.
- **Etiquetado de parte del discurso (POS):** Etiqueta cada palabra, por ejemplo, sustantivo, verbo, adjetivo, etc. Se crea modelos estocásticos que se entrenan en oraciones etiquetadas con el POS correcto, almacenando la información relevante. También se puede utilizar en la generación de respuestas deseadas indicando el tipo de objeto POS.

2.9 Administrador de diálogo

El administrador de diálogo se utiliza para controlar el proceso de diálogo, incluida la recepción y comprensión de la entrada del usuario, la búsqueda de la respuesta correcta en la base de conocimiento y la retroalimentación. Para lograrlo se debe elegir una serie de estrategias de comunicación o trucos de lenguaje para que parezca humano y el bot pueda entregar el mensaje (Abdul-Kader & Woods, 2015; Cahn, 2017).

2.9.1 Estrategias de interacción con chatbots

El primer paso en el diseño del sistema de gestión del diálogo es elegir una estrategia de interacción que determine quién lidera la conversación: El usuario, el bot o ambos. Al iniciar el

diálogo el usuario y el bot devuelve la respuesta. El problema de este enfoque es que el usuario puede proporcionar entradas mal definidas, a las cuales el bot no sabe cómo responder. Por otro lado, al iniciar el diálogo el bot y el usuario responde a las consultas. Esto reduce significativamente la cantidad de respuestas indefinidas del usuario y el campo de conversación, limitando el comportamiento del usuario. La estrategia de iniciativa mixta permite al usuario y al sistema liderar el diálogo en diferentes contextos (Abdul-Kader & Woods, 2015; Cahn, 2017).

El segundo paso es elegir una estrategia de manejo y confirmación de errores. Se pueden tomar dos enfoques principales: Confirmación explícita e implícita. Confirmación explícita el bot repite al usuario la consulta que cree haber escuchado, este es un enfoque que lleva mucho tiempo, pero asegura una buena comprensión de la consulta. Mientras que en la confirmación implícita, el bot incluye la consulta que cree haber escuchado como parte de la siguiente pregunta (Abdul-Kader & Woods, 2015; Cahn, 2017).

2.10 Generación de respuestas de un chatbot (GR)

La generación de respuesta es el módulo central de la arquitectura de un chatbot. La entrada que recibe el generador de respuesta es una representación estructurada del texto hablado, la cual transmite información sobre la historia del diálogo y el contexto. En base a la información obtenida, el objetivo de este componente es entregar un conjunto de respuestas, o contestaciones generadas por el administrador de diálogo con el fin de tomar una decisión al seleccionar una respuesta idónea (Abdul-Kader & Woods, 2015; Cahn, 2017). El sistema debe tener en cuenta:

- Base de datos de conocimiento / corpus de datos
- Corpus de historial de diálogo, que solo existe en modelos más complejos
- Fuente de datos externa, que proporciona inteligencia al bot.

A partir de este tema, se centrará en la base de conocimiento, ya que esto es lo que se utilizó para la implementación del presente trabajo.

2.10.1 Estrategia de generación de respuesta

Muchas de las estrategias de generación de respuestas se generan junto con puntajes de confianza que indican la probabilidad de que la respuesta del bot sea adecuada. Pero para las respuestas con baja probabilidad, el bot puede desarrollar algunos trucos de lenguaje que mejoran los resultados de la conversación, como los siguientes (Abdul-Kader & Woods, 2015; Cahn, 2017):

- Cambiar de tema.
- Realizar preguntas abiertas.
- Contar una broma para continuar con la conversación.
- Obtener más información: Si el chatbot no tiene una respuesta adecuada, simplemente puede pedirle al usuario que proporcione más información.

2.10.2 Creación de base de conocimiento

Los chatbots son tan inteligentes como el conocimiento al que tienen acceso. La base de conocimiento es como el cerebro del chatbot, el cual almacena toda la información para responder a las preguntas del usuario. Para construir la base de conocimiento o corpus de datos de un bot, es fundamental la información ingresada durante su entrenamiento, tanto la cantidad como la calidad de estos. Para así lograr una interacción similar a las de los humanos. El avance de los corpus de datos utilizados por los bots es un complemento al desarrollo de algoritmos para

mejorar la precisión de los bots. Esta sección se centra en los métodos de recopilación de datos para chatbots (Abdul-Kader & Woods, 2015; Cahn, 2017).

- **Corpus anotados por humanos**

La mayoría de los chatbots existentes poseen módulos de administración de diálogo, el que se encarga de controlar el proceso de conversación y las bases de conocimiento de chatbot, para responder a la entrada del usuario. La implementación típica de las bases de conocimiento de chatbot contiene un conjunto de plantillas que coinciden con la entrada del usuario y generan respuestas. Sin embargo, las plantillas utilizadas actualmente en chatbots están codificadas a mano. Por lo tanto, la construcción de la base de conocimiento de chatbot requiere mucho tiempo y es difícil de adaptar a nuevos dominios. Además, dado que el inglés es el idioma predominante en las investigaciones relacionadas al PLN, el uso de corpus ya anotados probablemente esté exclusivamente en el idioma inglés (Abdul-Kader & Woods, 2015; Cahn, 2017).

Reconociendo el valor de los corpus de diálogo ya existentes, Shawar et. Alabama y otros han desarrollado algoritmos para convertir diálogos anotados por humanos en lenguaje de marcado de inteligencia artificial (AIML) y otros formatos compatibles con chatbots. Sus programas tienen cuatro pasos básicos: (i) leer los diálogos de los corpus existentes, (ii) eliminar todo el texto superpuesto y rellenos no verbales, (iii) procesar el texto, y (iv) convertir estos diálogos en archivos AIML u otros formatos compatibles con chatbots para que estos diálogos puedan usarse como base de las bases de conocimiento (Abdul-Kader & Woods, 2015; Cahn, 2017).

- **Foros de discusión**

Un foro de discusión en línea es una comunidad web que permite a las personas discutir temas comunes, intercambiar ideas y compartir información en un determinado dominio. Crear hilos y

publicar respuestas son los principales comportamientos de los usuarios en las discusiones del foro. Grandes repositorios de hilos archivados y registros de respuestas en foros de discusión en línea, contienen una gran cantidad de conocimiento humano sobre muchos temas. Además de los estilos de respuesta de los autores son diversos (Abdul-Kader & Woods, 2015; Cahn, 2017).

- **Conversaciones por correo electrónico**

La conversación por correo electrónico para extraer pares de preguntas y respuestas. Este enfoque se beneficia de la supremacía de las conversaciones por correo electrónico, pero tiene una serie de inconvenientes que incluyen: monólogos largos utilizados, los correos electrónicos son asíncronos a diferencia del chat, diferencias estilísticas en el chat frente a la comunicación por correo electrónico y falta de claridad en cuanto a los pares específicos de preguntas y respuestas dentro de correos electrónicos más largos (Abdul-Kader & Woods, 2015; Cahn, 2017).

2.11 Texto a voz

Texto a voz o en inglés Text-to-Speech (TTS) es el paso final del proceso y convierte la respuesta generada a voz, la cual se devuelve al usuario. El primer paso es convertir el texto en fonemas con tono y duración. El segundo paso es la concatenación de los segmentos de voz grabada para formar la voz (Abdul-Kader & Woods, 2015; Cahn, 2017).

2.12 Plataformas para diseñar chatbots

Existen varias herramientas que permiten la creación de chatbots, dependiendo del tipo de uso que se dé al aplicativo (Zuñiga & Humberto, 2018). Estas herramientas son: Microsoft Bot Framework, Amazon LEX, IBM Watson Conversation, Wit.ai, Dialogflow, etc., pero para la

comparación y desarrollo de la tesis, las herramientas que se contemplaron, se describirán a continuación:

WIT AI: Solución de código abierto disponible para desarrolladores, su servicio permite el desarrollo de la lógica detrás de un chatbot utilizando PNL por la cual procesa los datos convirtiendo oraciones en datos estructurados. Es una herramienta simple de manejar, pero requiere una plataforma de mensajería. Se basa en el uso de entidades, intenciones, contextos, acciones e incorpora el procesamiento de lenguaje natural. Cada vez que se envíe un texto de entrada al Wit.AI, este responderá con un porcentaje de confianza, dando como respuesta la información solicitada y la intencionalidad del usuario. Se puede integrar a través de su API HTTP o mediante el uso de sus clientes oficiales (Da Rocha Loureiro, 2017; Salamanca, Eugenia, & Macq, 2019).

AMAZON ALEXA: Asistente virtual inteligente desarrollado por Amazon, usado conjuntamente con un altavoz y hardware de micrófono denominado Amazon “Echo”, el cual es un dispositivo controlado por voz. Tiene varias funciones según las skills que se incorporen a Alexa. Por ejemplo, reproducir música, controlar dispositivos inteligentes siendo el centro de automatización del hogar, activar alarmas, proporcionar información con respecto a noticias, entre otras funciones. Alexa también es compatible con dispositivos desarrollados por Amazon como Fire TV, Fire Phone, Kindle Fire. El funcionamiento de Alexa es parecido al de otros asistentes como Google Assistant, Siri y Cortana (López, Quesada, & Guerrero, 2017; Carvajal, Quesada, López, & Brenes, 2017).

DIALOGFLOW: Es una plataforma para desarrollo de tecnologías que permite crear interfaces de conversación para sitios web, apps, plataformas de mensajería y dispositivos de IoT. Dialogflow trabaja sobre Api.ai el elemento para trabajar técnicas NLP y dentro de esta se crea, configura y entrena el bot. La facilidad que ofrece Dialogflow es que se puede conectar a numerosas plataformas entre las que están: Facebook Messenger, Slack, Google Assistant, Twitter etc. (Torres, 2018; Beltrán, Barreto, Salazar, & Villareal, 2019).

IFTTT: Utiliza la sintaxis "IF This Then That" para especificar la programación de ejecución de una determinada acción (That) y la ocurrencia de un evento específico (This). IFTTT es una plataforma pública de servicios que ha ganado mucho impulso y adopción en la industria. IFTTT proporciona un paradigma lógico para controlar los servicios. Los usuarios deben iniciar sesión en la plataforma IFTTT y configuran sus dispositivos para que sean activados por los disparadores, por ejemplo, Youtube, openHAB, Spotify. Proporciona sus aplicaciones móviles como canales para funciones de teléfonos inteligentes y aplicaciones instaladas (Lucci & Paternò, 2015; López, Quesada, & Guerrero, 2017).

2.13 Subpreguntas de investigación

Tabla 3

Subpreguntas de investigación

Objetivo específico	Pregunta de investigación
i. Realizar un estudio de literatura que permita analizar diferentes técnicas de desarrollo e implementación de chatbots utilizando un lenguaje natural en el área de IoT (Internet de las Cosas), con el fin de evitar accidentes producidos.	a. ¿Cuáles han sido los aportes generados por otras investigaciones en el desarrollo e implementación de chatbots utilizando un lenguaje natural en el área de IoT? b. ¿Existen investigaciones que estén relacionadas con la problemática planteada?
ii. Definir una semántica específica para la comunicación entre persona y el dispositivo	b. ¿Qué aspectos fundamentales se considerarán para definir la semántica basada en un lenguaje natural?

móvil mediante el uso del procesamiento de lenguaje natural

- iii.** Hacer un estudio comparativo de las APIs de reconocimiento de voz para determinar la que ofrece mejores prestaciones para el desarrollo del chatbot.
 - iv.** Desarrollar un aplicativo móvil para interacción con el chatbot y los servicios de interconexión con el hardware, aplicando la metodología ágil basada en prototipos.
 - v.** Configurar la plataforma de hardware y software para validar la solución propuesta mediante la definición de pruebas funcionales y de rendimiento.
- c. ¿Cuál sería el proceso para la definición de la semántica?
 - d. ¿Qué funcionalidades se deben considerar para determinar que API es la más efectiva para el reconocimiento de lenguaje natural?
 - e. ¿Cuál es el aporte más significativo de cada API?
 - f. ¿Qué arquitectura de software será requerida para conseguir la unión eficiente de la aplicación y la API seleccionada?
 - g. ¿Por qué la metodología basada en prototipos es adecuada para el desarrollo del aplicativo?
 - h. ¿Qué hardware permitirá el control de los dispositivos electrónicos?
 - i. ¿Se reducirán el uso inadecuado de los aparatos eléctricos, mediante la aplicación propuesta?

CAPÍTULO III

EVALUACIÓN DE TECNOLOGÍAS PARA EL DESARROLLO DE CHATBOTS

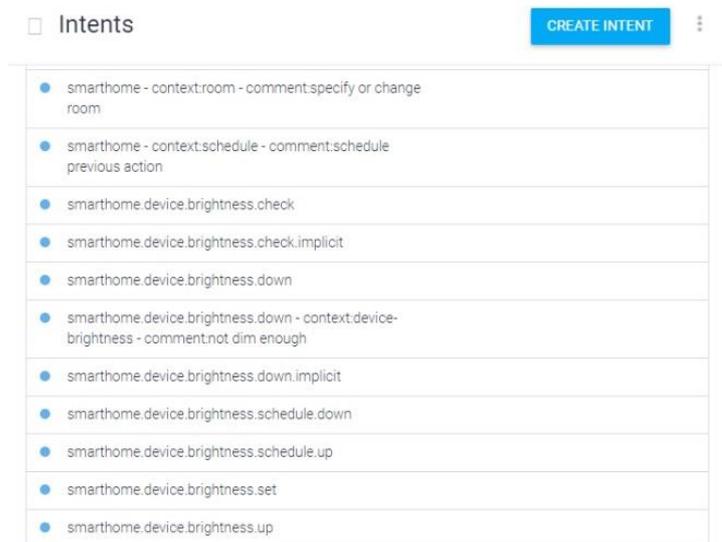
Dentro de este capítulo se hablará sobre la elección de la herramienta tecnológica más adecuada en cuanto a la creación de un chatbot. Para ello se ha comparado varias APIs que permiten el desarrollo de un bot. Los cuales en su mayoría usan procesamiento de lenguaje natural y reconocimiento de voz, permitiendo una comunicación de forma textual u oral con el usuario. Para el previo desarrollo del aplicativo, se implementó una semántica en específico, la cual trataremos a continuación.

3.1 Desarrollo de la semántica

Corpus es una colección de texto de lenguaje natural, elegido para caracterizar un estado o variedad de un idioma. El corpus tiene que mostrar a pequeña escala cómo funciona una lengua natural, pero para ello es necesario que esté diseñado correctamente y asegure que el resultado sea efectivamente un modelo de la realidad lingüística. El corpus informatizado es una recopilación de textos seleccionados según criterios lingüísticos, codificados de modo estándar y homogéneo, con la finalidad de ser tratados mediante procesos informáticos y destinados a reflejar el comportamiento de una o más lenguas (Torruella & Llisterri, 1999).

Para el desarrollo de la semántica, se ha utilizado el corpus o base de conocimiento ya establecido por la herramienta DialogFlow, el cual está relacionado con la temática de smart home. Este corpus cuenta con una gran variedad de: Intents (intenciones), Entities (entidades), ayudando a la alimentación de base de conocimiento. La Figura 10 muestra el corpus de datos en el idioma inglés, de donde se procedió a la correspondiente traducción. Esta fue una base fundamental para poder expandir el corpus de datos, en el cual se agregó sinónimos de las palabras, variedad de dispositivos, lugares o habitaciones dentro de una casa, tiempo, etc.

Una vez creada la base de conocimiento, la cual fue alimentada de manera manual, replicamos esta base en cada plataforma de desarrollo de chatbots. Las plataformas que utilizamos fueron: Dialogflow, IFTTT, Amazon Alexa y Wit.ai, con la única diferencia al momento de generar la respuesta, ya que cada chatbot responde de una manera particular, dependiendo de la programación o si la plataforma permite generar la respuesta.



The screenshot shows the 'Intents' section of the DialogFlow console. At the top right, there is a blue button labeled 'CREATE INTENT' and a vertical ellipsis menu icon. Below this is a list of ten intents, each with a blue circular icon to its left. The intents are as follows:

Intent Name
smarthome - context:room - comment:specify or change room
smarthome - context:schedule - comment:schedule previous action
smarthome.device.brightness.check
smarthome.device.brightness.check.implicit
smarthome.device.brightness.down
smarthome.device.brightness.down - context:device-brightness - comment:not dim enough
smarthome.device.brightness.down.implicit
smarthome.device.brightness.schedule.down
smarthome.device.brightness.schedule.up
smarthome.device.brightness.set
smarthome.device.brightness.up

Figura 10. Corpus de datos de DialogFlow
Fuente: (Google, 2019)

A continuación, se detallará el desarrollo del modelo basado en la herramienta API Natural Language de la Plataforma de Google Cloud.

3.1.1 API natural language

La herramienta proporcionada por la Plataforma de Google Cloud, permite analizar el texto, extrayendo información con respecto a personas, lugares y eventos, implementa el uso del aprendizaje automático el cual revela la estructura y el significado del texto. De igual manera permite analizar el texto en varios idiomas. Uno de los mayores beneficios de esta herramienta es que ofrece la misma tecnología de aprendizaje automático profundo el cual

potencia la capacidad de búsqueda de Google de responder a preguntas específicas de los usuarios y la comprensión de lenguajes detrás del asistente de Google (Google, 2019).

3.1.2 Características de la API Natural Language

Los modelos entrenados de la API de Natural Language, ofrecen al desarrollador una serie de características para la comprensión del lenguaje natural, las cuales son: Análisis de opiniones, análisis de entidades, análisis de opiniones sobre entidades, clasificación de contenido y análisis sintáctico los cuales se describirán a continuación en la Tabla 6 (Google, 2019).

Tabla 4
Características de la API de Natural Language.

MÉTODOS	DESCRIPCIÓN
<i>Análisis de opiniones</i>	Se centra en opiniones emocionales del texto dado, determinando como positiva, negativa o neutra la actitud de la persona.
<i>Análisis de entidades</i>	Determina las entidades más conocidas dentro del texto, las cuales pueden llegar a ser los nombres propios, puntos de referencia, sustantivos, entre otros y, etiqueta las entidades según su tipo como personas, fecha, ubicación, etc.
<i>Análisis de opiniones de entidades</i>	Este método es una combinación de los dos métodos anteriormente mencionados donde busca entidades conocidas dentro del texto dado y determina la actitud positiva, negativa, neutra de la persona con respecto a la entidad.
<i>Análisis sintáctico</i>	Extrae la información mediante la división del texto (oraciones y tokens), identifica las categorías gramaticales, crea árboles de análisis de dependencia para cada oración.
<i>Clasificación de contenido</i>	Realiza el análisis del contenido del texto y muestra la categoría del mismo. Puede llegar a clasificar documentos en más de 700 categorías predefinidas.

Fuente: (Google, 2019)

3.1.3 Funcionalidad de API Natural Language

La Figura 11, muestra la funcionalidad de la API Natural Language, la cual permite introducir un texto o frase, para generar el análisis de las entidades que se han identificado dentro del texto. En el ejemplo se detectó dos entidades (luces y cocina), en las que se ha

etiquetado como: otro y locación respectivamente. Mientras mayor sea el texto, mayor es el número de entidades descubiertas por la herramienta.

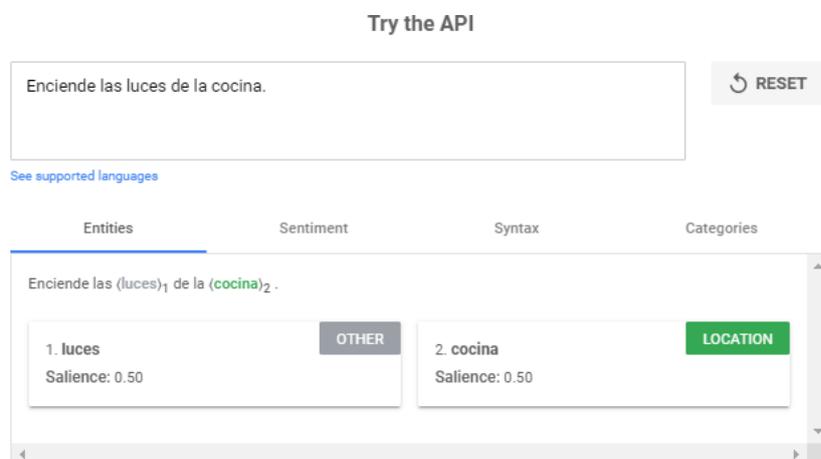


Figura 11. Funcionalidad de la API Natural Language
Fuente: (Google, 2019)

3.1.4 Análisis de entidades en API Natural Language

Dentro de este análisis como se lo ha mencionado con anterioridad, se proporciona información sobre las entidades del texto dado. Las entidades pueden ser clasificados en dos categorías; sustantivos propios (persona, lugar en específico, etc.) o sustantivos comunes (nominales). Un sustantivo por lo general se lo califica como entidad. En la anterior Figura 12 se muestra las entidades detectadas integradas al texto original.

- **Campos de respuestas del análisis de entidades**

En la Figura 12 se puede ver el conjunto de entidades que han sido detectadas, también los parámetros asociados a las mismas, como el tipo de entidad y la relevancia que tiene la entidad dentro del texto. Las puntuaciones “salience” indica la importancia de las entidades para todo el texto o documento, lo cual es ideal al momento de priorizar la relevancia de las entidades. Las puntuaciones cercanas a 1.0 llegan a ser las más importantes.

3.1.5 Análisis Sintáctico en API Natural Language

Este consiste en las siguientes operaciones: La extracción de oraciones donde se divide el contenido del texto en una serie de oraciones, la asignación de tokens (cada token corresponde a una sola palabra), para ser posteriormente procesados agregando información sintáctica a cada uno de ellos. En los siguientes párrafos se hablará sobre cómo se realiza la asignación de tokens, categorías gramaticales y árboles de dependencia, que proporciona la herramienta API Natural Language.

- **Asignación de tokens:** Transforma el texto en una serie de tokens los cuales corresponden a los diferentes elementos textuales del contenido. Al asignar estos tokens, los mismos son extraídos para determinar la categoría gramatical asociada. Son evaluados para colarse dentro de un árbol de dependencia y determinar el significado y la relación entre sí. La información sintáctica y morfológica asociada a los tokens es útil para comprender la estructura sintáctica de las oraciones dentro de la herramienta.
- **Categorías gramaticales:** Dentro de una solicitud sintáctica, las categorías gramaticales y la información morfológica se muestran en el campo de la respuesta. Este campo contiene un conjunto de subcampos con información sobre las categorías gramaticales part-of-speech tagging (POS) o etiquetado gramatical, además de información morfológica más explícita. Estos subcampos se enumeran a continuación en la Tabla 5.

Tabla 5

Categorías gramaticales de API Natural Language

SUBCAMPO	CARACTERÍSTICA
Tag	Denota la categoría gramatical que usa una etiqueta general para las POS (NOUN [SUSTANTIVO], VERB [VERBO], etc.), y proporciona información sintáctica superficial de nivel alto
Number	Denota un número gramatical. Por ejemplo: SINGULAR, PLURAL y DUAL (denota precisamente dos cantidades).
Person	Identifica la persona gramatical de una palabra. Este campo puede contener los siguientes valores: FIRST (primera) persona denota al emisor.

CONTINÚA 

	<p>SECOND (segunda) persona denota al receptor, es decir, la persona a quien se le habla.</p> <p>THIRD (tercera) persona denota a quien no es el emisor o receptor.</p> <p>REFLEXIVE_PERSON (persona reflexiva) denota al sujeto y el objeto hacen referencia a la misma entidad</p>
Gender	<p>Denota el género gramatical de un sustantivo. Este campo puede contener los siguientes valores:</p> <p>FEMININE (femenino)</p> <p>MASCULINE (masculino)</p> <p>NEUTER (neutro)</p>
Case	<p>Denota un caso gramatical y su función en una frase u oración. Este campo puede contener los siguientes valores:</p> <p>ACCUSATIVE (acusativo) indica el objeto directo de un verbo transitivo.</p> <p>ADVERBIAL indica una forma adverbial de un adjetivo.</p> <p>DATIVE (dativo) sirve para expresar la persona o cosa que recibe el beneficio o perjuicio de la acción verbal.</p> <p>GENITIVE (genitivo) indica posesión.</p> <p>INSTRUMENTAL (instrumental) indica si un sustantivo es el instrumento mediante el cual se completa una acción.</p> <p>LOCATIVE (locativo) indica el uso de una palabra para referirse a una ubicación.</p> <p>NOMINATIVE (nominativo) está asociado con el sujeto de un verbo.</p> <p>OBLIQUE (oblicuo) indica que una palabra se usa como objeto para un verbo o una preposición.</p> <p>PARTITIVE (partitivo) indica la "parcialidad" o falta de identidad específica de una palabra.</p> <p>PREPOSITIONAL (preposicional) indica el objeto de una preposición.</p> <p>REFLEXIVE_CASE (caso reflexivo) indica la identidad de un objeto de un verbo con respecto a su sujeto.</p> <p>VOCATIVE (vocativo) indica un sustantivo utilizado para dirigirse a alguien o algo, generalmente cuando se le habla.</p>
Tense	<p>Denota el tiempo gramatical, el cual indica la referencia del verbo con respecto a una posición en el tiempo. Este campo puede contener los siguientes valores:</p> <p>CONDITIONAL_TENSE (tiempo condicional)</p> <p>FUTURE (futuro)</p> <p>PAST (pasado)</p> <p>PRESENT (presente)</p> <p>IMPERFECT (imperfecto)</p> <p>PLUPERFECT (pluscuamperfecto)</p>
Aspect	<p>Denota el aspecto gramatical, se enfoca en las características del flujo de tiempo donde ocurre. Este campo puede contener los siguientes valores:</p> <p>PERFECTIVE (perfectivo) denota un evento que se "completó"</p> <p>IMPERFECTIVE (imperfectivo) denota un evento que está incompleto</p> <p>PROGRESSIVE (progresivo) denota un evento que es continuo.</p>
Mood	<p>Denota el modo gramatical de un verbo, el cual representa una actitud con respecto a una acción subyacente. Este campo puede contener los siguientes valores:</p> <p>CONDITIONAL_MOOD (modo condicional)</p> <p>IMPERATIVE (imperativo)</p> <p>INDICATIVE (indicativo)</p> <p>INTERROGATIVE (interrogativo)</p> <p>JUSSIVE (yusivo) indica una orden o solicitud en primera o tercera persona.</p> <p>SUBJUNCTIVE (subjuntivo)</p>
Voice	<p>Denota la voz gramatical de un verbo, la relación entre una acción y un sujeto u objeto. Este campo puede contener los siguientes valores:</p> <p>Voz ACTIVE (activa)</p> <p>Voz CAUSATIVE (causativa) indica una acción donde el sujeto realiza el efecto de la misma</p> <p>Voz PASSIVE (pasiva)</p>

Reciprocity	Denota la reciprocidad, de una palabra e indica que el pronombre hace referencia a una frase nominal en otra ubicación de la oración. Este campo puede contener los siguientes valores: RECIPROCAL (recíproco) indica que el pronombre es recíproco. NON_RECIPROCAL (NO recíproco) indica que el pronombre no es recíproco
Proper	Denota si un sustantivo es parte de un nombre propio. PROPER (propio) denota que el token es parte de un nombre propio. NOT_PROPER (no propio) denota que el token no es parte de un nombre propio.
Form	Denota formas morfológicas adicionales que no se ajustan perfectamente en el conjunto anterior de formas comunes. Este campo puede contener los siguientes valores: GERUND (gerundio), ORDEN, AUXILIAR, entre otros.

Fuente: (Google, 2019)

- **Árboles de dependencia:** Para cada oración dentro del texto proporcionado por la API de Natural Language para el análisis sintáctico, la API construye un árbol de dependencia que describe la estructura sintáctica de la siguiente oración “Enciende las luces de la cocina.”. A continuación, se muestra un diagrama del árbol de dependencia, la cual asigna los tokens de cada palabra, su respectivo etiquetado gramatical y el árbol de dependencia que se generó, obsérvese la Figura 12.

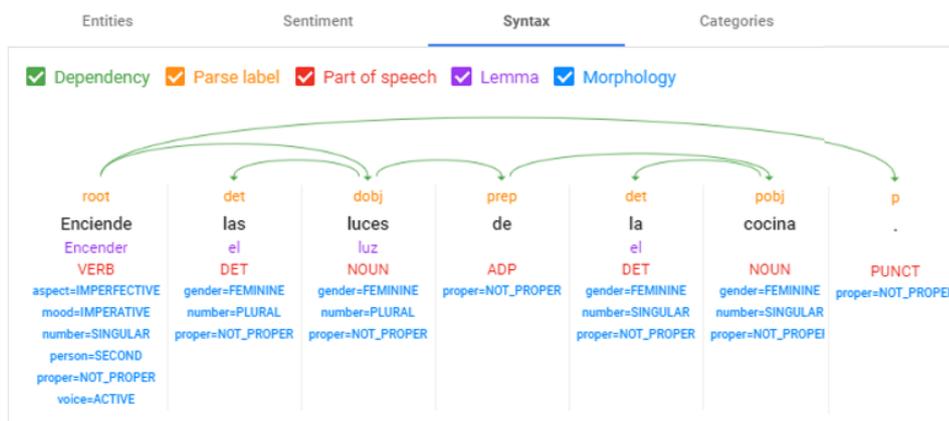


Figura 12. Árbol de dependencia

Fuente: (Google, 2019)

3.2 Dialogflow

Dialogflow permite crear interfaces de conversación sobre productos y servicios al proporcionar un potente motor de comprensión del lenguaje natural (NLU) para procesar y comprender las entradas de lenguaje natural (Google, 2019).

3.2.1 Arquitectura de DialogFlow

Agente virtual: Es un módulo de comprensión del lenguaje natural que comprende los matices del lenguaje humano, llegando a manejar conversaciones con los usuarios. Dialogflow traduce el texto o el audio del usuario final durante una conversación a datos estructurados que sus aplicaciones y servicios pueden entender. Estos agentes son entrenados para manejar los escenarios de conversación esperados, y su entrenamiento no necesita ser demasiado explícito (Google, 2019).

Componentes del Agente

Intents: Un intent clasifica la intención del usuario para un turno de conversación. Para cada agente, se define varios intents, donde estos combinados pueden manejar una conversación completa. Cuando un usuario envía un mensaje de texto o de voz, Dialogflow hace coincidir estas frases con el intent más adecuado de su agente. Un intent básico contiene lo siguiente (Google, 2019):

- **Frases de entrenamiento:** Corresponden a ejemplos de frases que los usuarios podrían decir. Dialogflow usa estas frases de entrenamiento y las expande a muchas más frases similares para construir un modelo de lenguaje que coincida con precisión con la entrada del usuario.
- **Acción:** El campo de acción es un campo de conveniencia sencillo que ayuda a ejecutar la lógica en el servicio.
- **Parámetros:** Son datos estructurados que se pueden usar fácilmente para realizar alguna lógica o generar respuestas.
- **Respuestas:** Texto o respuestas visuales que son devueltas al usuario, también pueden solicitar más información o finalizar la conversación. A continuación, la Figura 13 muestra un ejemplo de la generación de intent, en el cual se exponen varias frases de

entrenamiento que van a ser ingresadas al agente. Este intenta hacer coincidir la declaración con un intent particular, luego, el agente muestra la respuesta dentro de ese intent.

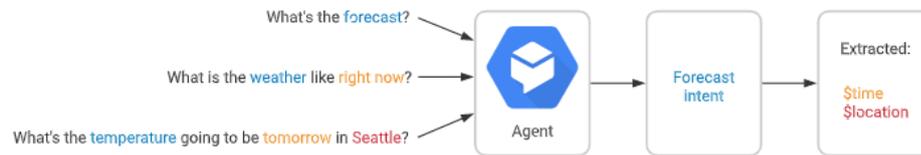


Figura 13. Generación de Intent

Fuente: (Google, 2019)

Entities: Son mecanismos para identificar y extraer datos útiles de las entradas de lenguaje natural. Dialogflow proporciona entidades del sistema predefinidas que pueden coincidir con muchos tipos comunes de datos. Por ejemplo, fechas, horas, colores, direcciones de correo electrónico, etc. También puede crear sus propias entidades de desarrollador según las necesidades del usuario (Google, 2019).

Contextos: Son similares al contexto del lenguaje natural. Para que Dialogflow maneje una expresión de usuario, debe proporcionarse un contexto para que coincida correctamente con un intent. Usando contextos se puede controlar el flujo de una conversación (Google, 2019).

3.3IFTTT

Hay dos perspectivas diferentes para el modelo de aplicación de IFTTT. La perspectiva del desarrollador (canal) y perspectiva del usuario (receta). Para proveer los canales, IFTTT administra canales básicos para servicios populares, mientras que los propietarios de canales de terceros son responsables de sus API web (autenticación / autorización / acceso). El alcance del desarrollo es abstraer y envolver operaciones relacionadas con API como una colección de disparadores / acciones en cada canal.

3.3.1 Arquitectura de IFTTT

Applet o Recetas: Un Applet conecta dos o más aplicaciones o dispositivos entre sí. Permite al usuario realizar algo que esas aplicaciones o dispositivos no podrían hacer por sí mismos. Los usuarios de IFTTT pueden activar Applets creados por servicios u otros usuarios, o pueden crear los suyos propios (IFTTT, 2019).

Componentes del Applet

- **Trigger o Activador:** Los activadores son eventos en un servicio, es decir, son los datos que, al ser modificados, logran que un Applet se ejecute en IFTTT.
- **Campos disparadores:** Los campos de activación permiten a los usuarios ingresar filtros o modificadores para un determinado activador.
- **Filtro:** Dictan qué acciones deben ejecutarse por medio de un código simple, en función de los datos de activación que ingresan.
- **Consultas:** Es una forma de solicitar datos adicionales si el activador no lo proporciona. IFTTT inicia una ejecución de consulta como resultado de un activador actualizado.
- **Action o Acción(es):** Las acciones es el envío de datos o la creación de recursos en su servicio. Se puede seleccionar una o más acciones para ejecutar cuando se utiliza un activador. Cada activador contiene ingredientes, es decir, datos individuales.
- **Campos de acción:** Un usuario agrega los ingredientes en los campos de acción. Los datos pueden rellenarse previamente y ocultarse en un campo de acción o el usuario puede completarlos. En el caso de los campos de acción, es importante ocultar cualquier campo que no necesite ser completado por el usuario.
- **Servicios:** Los servicios son los componentes básicos de IFTTT. Algunos servicios de ejemplo son Facebook, Twitter, Nest, Fitbit, Amazon Alexa y Gmail.

3.4 Amazon Alexa

Alexa se comercializa como un asistente virtual el cual fue desarrollado por Amazon, se puede interactuar con este asistente mediante la aplicación móvil o mediante los dispositivos físicos denominados Echo. Los cuales son altavoces inteligentes para el registro y transmisión de comandos de voz al servicio de Alexa. Integra múltiples funcionalidades como transmisión de música, información con respecto al clima, tráfico, noticias en tiempo real, etc (Haack, Severance, Wallace, & Wohlwend, 2017).

3.4.1 Arquitectura Amazon Alexa

Según el estudio “Security Analysis of the Amazon Echo” (Haack, Severance, Wallace, & Wohlwend, 2017), al momento de acceder al servicio mediante los dispositivos Echo, existen un número de componentes que interactúan para producir respuestas a los comandos generados por voz. En la Figura 14 se describe la arquitectura básica que maneja Amazon Alexa.

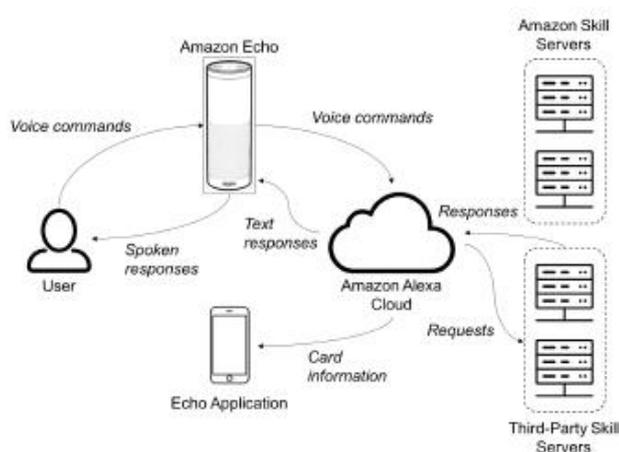


Figura 14. Arquitectura Amazon Alexa

Fuente: (Haack, Severance, Wallace, & Wohlwend, 2017)

- *Dispositivo Echo:* Dispositivos inteligentes, provee una interfaz basada en comandos de voz para interactuar con el usuario, transmite el audio al servicio en la nube de

Alexa, para recibir una respuesta basada en texto y proceder a reproducir el audio que será la respuesta por medio de voz.

- *Alexa Cloud*: servicio el cual proporciona el manejo de reconocimiento de voz, relaciona los comandos de voz a los intents, según la intención el servicio delega el comando a un servidor de habilidades, estas habilidades pueden ser nativas (clima, hora, noticias, etc.) o habilidades integradas por terceros.
- *Servidores de skills o habilidades nativas*: Estas funciones son nativas al ser incorporadas y desarrollados por Amazon.
- *Servidores de skills o habilidades de terceros*: Servicio donde se alojan habilidades incorporadas por desarrolladores externos. Estas habilidades deben habitar en la nube, para que los usuarios puedan hacer uso de las mismas, pudiendo alojarse en dos lugares como lo muestra la Figura 15: Amazon Web Services, AWS Lambda, o un punto de acceso HTTP mediante un certificado de seguridad (Guzmán Villagómez, 2018).

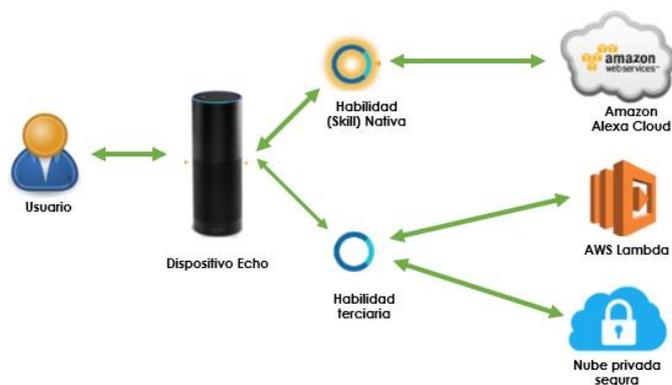


Figura 15. Servidores de habilidades para skills nativas y terciarias.

Fuente: (Guzmán Villagómez, 2018).

3.4.2 Alexa Skills Kit (ASK)

Alexa Skills Kit, es una colección de APIs, herramientas, documentación y ejemplos. Provee su plataforma para el uso de desarrolladores donde estos puedan generar habilidades

(skills) produciendo funcionalidades personalizadas en diferentes idiomas. Se puede encontrar documentación, muestras de código y mayor información para el desarrollo de una skill (Amazon, Amazon alexa, 2019). Por medio de Alexa Developer Console el desarrollador podrá crear una skill, donde se podrá generar un modelo de interacción, en el cual se debe tomar en cuenta ciertos aspectos para su desarrollo:

- *Nombre de la invocación:* Es el nombre con el cual el usuario invocará a la skill.
- *Intents:* Serán las acciones que el usuario lograra realizar.
- *Utterances:* Son las frases o sentencias definidas para cada intent.
- *Slots:* También conocidos como ranuras, las cuales actúan como variables dentro de los enunciados o utterances, pueden tener valores predefinidos.

3.4.3 Alexa Voice Services (AVS)

Es un servicio alojado en la nube, escalable, los productos desarrollados pueden ser activados mediante comandos de voz, estos manejan el reconocimiento automático de voz (ASR) y procesamiento de lenguaje natural (PNL), facilitando del uso de la interfaz y la construcción del producto. Permite integrar nuevos productos o funciones a Alexa proporcionando las herramientas necesarias (hardware y software), por medio de AVS, se puede añadir interfaces inteligentes, las cuales son controladas por medio de la voz (Guzmán Villagómez, 2018). Alexa Developer Console provee un simulador, el cual admite un input tanto en texto como en voz. Soporta SSML (Speech Synthesis Markup Language) lenguaje de marcado de síntesis de voz, el cual es un estándar para generación de voz a partir del texto (Amazon, Alexa skill kit docs, 2019).

3.5 Wit.ai

Wit.ai es una plataforma de software como servicio (SaaS), la cual fue adquirida por Facebook. Permite a los desarrolladores crear chatbots para aplicaciones, mediante la

interacción de voz y texto. Ofrece un kit de desarrollo de software (SDK) de forma gratuita, disponible para todo desarrollador. Wit.ai permite el desarrollo de apps móviles, tecnología en el hogar, robótica o también tecnología ponible. (Wit.ai, 2019).

3.5.1 Arquitectura de Wit.ai

El modelo de aprendizaje automático del procesamiento de lenguaje natural se basa en ejemplos, es decir que se alimenta la base de conocimiento mediante el ingreso de frases por parte del usuario. Una vez ingresadas las reglas, se las puede identificar mediante intenciones o entidades dependiendo del contexto que se maneje. Wit.ai tiene un gran mecanismo para entender las entidades del lenguaje, permitiendo asignar un rol a las entidades, para ayudar en el procesamiento ya sea de texto o de voz en el lado del servidor. En la Figura 16 se representa una intención “reserva de restaurante” y las entidades que la acompañan “número de personas”, “lugar”, “día”.

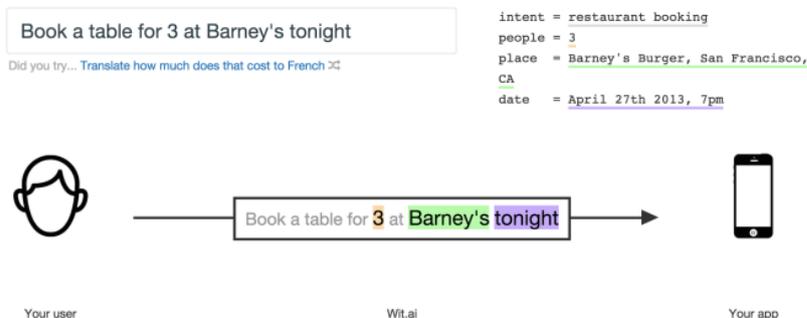


Figura 16. Representación de entidades e intenciones

Fuente: (Wit.ai, 2019)

3.5.2 PNL incorporado en Facebook Messenger

El procesamiento de lenguaje natural (PNL) permite extraer y comprender la información de tal manera en la que se pueda extraer entidades de mayor relevancia desde los mensajes que las personas envían. También se puede activar varios tipos de entidades preexistentes de Wit. Los cuales analizarán elementos de conversación como fechas, horas, saludos, etc..., o

bien se puede partir de un modelo propio, con la finalidad de identificar intenciones y dar respuestas automatizadas (Facebook, 2019).

El modelo desarrollado en la plataforma de Wit, se puede integrar mediante el token de servidor de acceso el cual está disponible en la plataforma. Como ejemplo de funcionalidad de la API una vez incorporado el procesamiento de lenguaje natural (PNL) a una página creada en la aplicación de Facebook. Se detecta automáticamente el significado y la intención de cada mensaje de texto antes de ser enviado al bot. El mensaje será enviado conjuntamente con las entidades e intenciones detectadas en el cuerpo. Esto quiere decir que la plataforma de Messenger devuelve un análisis de Wit de forma automática al webhook con cada mensaje (Facebook, 2019).

Webhook: Un webhook es una solicitud HTTP POST la cual interviene al momento de generarse un evento, los cuales son utilizados en tiempo real. En la plataforma Messenger se envía un evento webhook siempre y cuando se haya producido una acción al momento de interactuar con el bot. El webhook viene a ser un extremo HTTPS único el cual se expone y acepta solicitudes POST, para que el bot pueda procesar y responder a los eventos webhooks entrantes (Facebook, 2019).

3.6 Cuadros comparativos de las plataformas

Para la realización de los siguientes cuadros comparativos se ha tomado como referencias los siguientes estudios: “BiblioBot” (Beltrán, Barreto, Salazar, & Villareal, 2019), “Dialogflow vs Lex vs Watson vs Wit vs Azure Bot, Which Chatbot Service Platform To Use?” (Mamgain, 2019), y la investigación de Data Monsters, en la cual se estudiaron las plataformas más conocidas para la construcción de chatbots.

Se incorporó varios criterios para evaluar y calificar a las cuatro plataformas escogidas para la comparación. La Tabla 6, define los siguientes criterios con respecto a la usabilidad de

cada plataforma en el desarrollo de un chatbot. Para lo cual se ponderó con una calificación dependiendo de la dificultad de uso de cada herramienta, alto, medio y bajo son las ponderaciones que se tomaran como referencia.

Tabla 6

Cuadro Comparativo. Dificultad de cada API

	DIALOGFLOW	WIT.AI	AMAZON ALEXA	IFTTT
Dificultad de uso	Bajo	Medio	Bajo	Bajo
Dificultad en la configuración	Bajo	Bajo	Medio	Bajo
Confianza y seguridad	Alto	Alto	Alto	Alto
Navegabilidad y capacitación	Alto	Alto	Alto	Alto

En la Tabla 7 se definen aspectos generales como características, costo, integración con otras aplicaciones entre otros aspectos técnicos que aporten con la evaluación de las plataformas seleccionadas.

Tabla 7
Cuadro comparativo. Características generales de cada API

	DIALOGFLOW	WIT.AI	AMAZON ALEXA	IFTTT
Proveedor	Google	Facebook	Amazon	IFTTT
Características	<p>Permite coincidir consultas con las intenciones más adecuadas, de acuerdo a la información contenida en las intenciones.</p> <p>Modelo de aprendizaje automático del agente.</p> <p>Transforma el texto de la consulta en datos procesables, devolviendo un objeto de respuesta JSON.</p>	<p>Permite el uso de entidades, intentos, contexto y acciones.</p> <p>Procesamiento de lenguaje natural.</p> <p>No permite generación de respuesta</p>	<p>Permite el uso de intents, ranuras, contexto, desarrollo de modelo.</p> <p>Permite generación de respuesta, emulador de dialogo y pruebas para el procesamiento de lenguaje natural.</p>	<p>Permite la conexión entre dos servicios, no se implementa lenguaje de programación natural, basándose en reglas.</p>
Módulo de entrenamiento	Si	Si	Si	No
Importar/Exportar modelos	Si	Si	Si	No
Uso de Intents	Si	Si	Si	No
Entidades pre definidas	Si	Si	Si	No
Intents pre definidos	Si	Si	Si	No
Guardar proceso	Si	Si	Si	Si
Reconocimiento de voz	Si	Si	Si	Si
Integración con otras aplicaciones	Google Assistant, Facebook Messenger, Skype, Twitter,	Facebook Messenger, integración a una app	Integración de skills nativas y skills de terceros.	Integración con una gran gama de aplicaciones tales

CONTINÚA →

	Viber, Line, Telegram, Twilio, Cisco Spark	desarrollada o a un sitio web.		como Alexa, Facebook Messenger, Google, Spotify, Openhab, twitter, etc.
Integración con otros lenguajes de programación	SDKs: Android, iOS, Cordova, HTML, JavaScript, Node.js, Unity, Xamarin, C++, Python, Ruby, PHP, Java, Botkit.	Cientes: Node.js, Python, Ruby y otras plataformas HTTP API.	Node.js, Java, Python, C#, Go, HTTP API.	Ruby, Node.js
Idiomas que soporta	20	50	27	6
Límite de llamadas de la API	Ilimitado solo para el plan estándar. Para la versión Enterprise tiene un costo de \$0.002 la solicitud	Ilimitado	Ilimitado, costo adicional si la skill se aloja en los servicios que ofrece Amazon.	Ilimitado, ofrece planes para compañías y desarrolladores el cual tiene un costo.
Licencia	Acuerdo de licencia de Google Cloud Platform	Wit es gratis, incluso para uso comercial. Las aplicaciones Wit privadas y públicas son gratuitas y se rigen por los términos de la compañía.	Propietario/Software gratis	Gratuito, excepto para desarrolladores y compañías
Tipo de motor	Procesamiento de lenguaje natural (NLP)	Comprensión del lenguaje natural (NLU) y (NLP)	Comprensión del lenguaje natural y procesamiento de lenguaje natural	No
Generador de respuesta	Si	No	Si	Si
Administrador del diálogo	Si	No	Si	No

3.7 Pruebas y resultados con las API's

Para la realización de las pruebas con las APIs seleccionadas (Dialogflow, IFTTT, Amazon Alexa, WIT.AI), se ha tomado como referencia el siguiente estudio “You Shall not speak: Benchmarking famous speech recognition APIs for Chatbots” realizado por SAP Conversational AI, donde se analizó la precisión para el reconocimiento de voz y funcionalidad. Con el objetivo de medir y analizar el rendimiento de cada plataforma verificando si responde y entiende la orden.

3.7.1 Pruebas de precisión

Para medir la precisión, se adoptó del estudio mencionado las dos siguientes métricas las cuales son: Oraciones exactas reconocidas y tasa de error por palabra.

- *Oraciones exactas reconocidas*

Esta métrica pretende medir el porcentaje de oraciones exactas reconocidas por medio de la voz. Para esto se ha introducido las siguientes frases definidas en la siguiente Tabla 8, las cuales pretenden determinar el reconocimiento de palabras homófonas, modismos quiteños, números, signos de puntuación y ortografía más no la generación de respuestas de cada plataforma.

Tabla 8

Pruebas de coincidencia exacta.

Nº	Oración esperada
1	“Mientras un rayo cae sobre el patio, rallo queso para la cena”
2	“Después de hacer las paces, ya podemos ir a entrenar y probar unos pases.”
3	“Podrías encender las luces a las cinco de la mañana.”
4	“El ecuatoriano no tiene resaca. Tiene chuchaqui después de una chupa”.
5	“Tienes que ver a mi abuelo, llegando a los cien años y con el cabello alrededor de su sien”.

- **Pruebas con Dialogflow**

Para esta prueba Dialogflow no reconoció las pausas relacionados con las puntuaciones, en la oración número 2 reconoció las siguientes palabras homófonas “paces” y “pases” entendiendo el contexto de la oración por lo que le fue fácil diferenciar estas palabras. Mientras que en las oraciones 1 y 5 tuvo problemas. Al mencionar números en las pruebas 3 y 5, no se representó con palabras si no con la numeración respectiva, como por ejemplo “Podrías encender las luces a las 5 de la mañana.” También al referirse a modismos quiteños, la palabra “chuchaqui” fue reconocida adecuadamente mientras que con la palabra “chupa”, se tuvo que realizar la prueba una segunda vez para su reconocimiento.

- **Pruebas con IFTTT**

No se realizó esta prueba porque esta plataforma no cuenta con un modelo de ASR (reconocimiento automático de voz), ya que solo vincula servicios de Google Assistant y Facebook Messenger, y al realizar las pruebas con esta herramienta se estaría repitiendo las mismas pruebas realizadas con Dialogflow.

- **Pruebas con Amazon Alexa**

De las frases esperadas solo se obtuvo un bajo porcentaje de reconocimiento exacto de la voz. El problema radica en el reconocimiento de palabras homónimas, las cuales se encuentran en las frases 1, 2, donde las palabras “rallo” y “paces” no fueron identificadas de forma correcta, y hubo un error en la palabra “sien” la cual fue reemplazada con “sushi”. Dentro de los modismos ecuatorianos, se realizó varios intentos al enviar el audio con la cuarta frase, para que reconozca bien las palabras, donde la herramienta tuvo errores al retornar la palabra “chupe”, obteniendo como resultado palabras como: chica, lechuga, chu.

- **Pruebas con Wit.ai**

Al realizar las pruebas con esta herramienta, de igual forma el problema radica en el reconocimiento de palabras homónimas y reconocimiento de pausas en una oración, las cuales no fueron identificadas de forma correcta, pero no hubo error o distorsión de palabras. En los modismos ecuatorianos la palabra “chuchaqui” también se tuvo errores ya que fue reemplazada con otras palabras. Lo particular de esta herramienta es que permite validar los audios enviados y corregir las palabras si no son acorde al audio enviado.

- **Tasa de error por palabra**

La segunda métrica permite calcular la tasa de error por palabra. Para poder evaluar este error se ha definido algunos parámetros en la Tabla 9, donde se sustituyó, eliminó, adicionó y se modificó la estructura gramatical de una frase.

Tabla 9

Pruebas de Tasa de error por palabra

MÉTRICA	FRASE
Rendimiento esperado	Podrías apagar las luces
Sustitución	Puedes pagar las luces
Supresión	Puedes las luces
Adición	Podrías apagar las luces por favor
Contrario	Podrías luces apagar

- **Pruebas con DialogFlow**

Al realizar la primera prueba de rendimiento esperado y de adición se obtuvieron resultados óptimos, pero para la segunda prueba al sustituir la palabra “apagar” el bot no entendió la orden. Al realizar la tercera prueba de supresión en la que se eliminó la acción, el bot entendió la entidad “luces” por lo que dio una respuesta, pero esta no fue adecuada. Finalmente, al realizar la última

prueba de contrarios el bot entendió la orden a pesar de tener una estructura gramatical incorrecta por lo que se debe corregir este error.

- **Pruebas con IFTTT**

Para realizar las pruebas con IFTTT se integró Google Assistant y Facebook Messenger, ya que IFTTT no cuenta con un modelo de reconocimiento de voz automático. Al realizar las pruebas de rendimiento esperado y adición se obtuvieron los resultados deseados, pero para las pruebas de sustitución, supresión y de contrarios, IFTTT no entendió las ordenes, ya que no cuenta con un procesamiento del lenguaje natural.

- **Pruebas con Amazon Alexa**

La siguiente Figura 17 muestra las pruebas realizadas. El rendimiento esperado al decir el enunciado de “Podrías apagar las luces” es óptimo, pero al momento de sustituir la palabra “apagar” por “pagar”, la herramienta lo interpreta como si se debería apagar un dispositivo, lo cual es totalmente errado. Al pasar por las pruebas de adición, supresión de palabras y contrarios no hubo mayor problema.

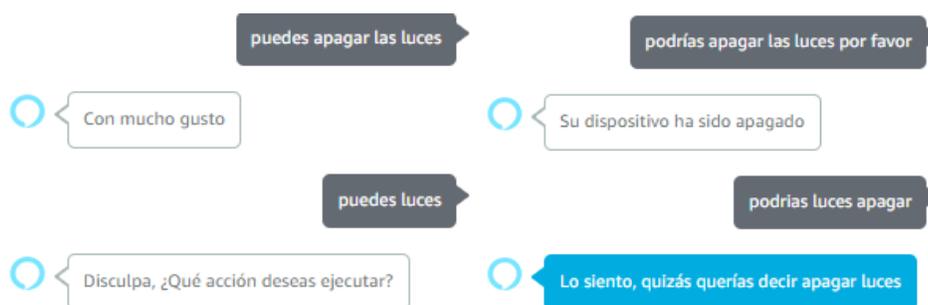


Figura 17. Pruebas de tasa de error por palabra con Alexa.

- **Pruebas con Wit.ai**

Los audios fueron enviados a la plataforma Wit.ai. Se retorna un JSON, con la identificación de las entidades generadas dentro del texto una vez que este ha sido procesado. En la prueba de sustitución, se detectó la palabra “pagar”, pero no se identificó acción alguna. Mientras que, en la prueba de contrarios, es decir el cambio de posición del verbo y el predicado, las entidades fueron identificadas, pero gramaticalmente la oración no tiene un orden adecuado, por lo que es necesario corregir este error dentro del modelo.

3.7.2 Resultados de pruebas de precisión

Para el resultado final de las pruebas de precisión, se ha distribuido el porcentaje entre las dos métricas establecidas. Por lo que se otorgó el 50 por ciento a la métrica de oraciones exactas reconocidas y el otro 50 por ciento a la métrica de tasa de error por palabra.

- ***Oraciones exactas reconocidas***

En general las herramientas tuvieron problemas al reconocer puntuaciones y palabras homónimas. Alexa y Wit.ai tuvieron problemas al reconocer los modismos ecuatorianos después de varios intentos, mientras que Dialogflow pudo reconocerlos sin mayor problema. Alexa y Wit.ai reconocieron los números con palabras mientras que Dialogflow reconoció la numeración correspondiente. Cabe recalcar que para estas pruebas no se utilizó la herramienta IFTTT ya que vincula Google Assistant, el cual actúa con el mismo reconocimiento automático de voz (ASR) de Dialogflow. La siguiente Tabla 10 muestra los porcentajes que se obtuvo de cada herramienta, dependiendo de la identificación exacta de las oraciones.

Tabla 10*Resultados obtenidos en prueba de Oraciones exactas reconocidas.*

API AI	Oraciones exactas reconocidas					Porcentaje
	OER1	OER2	OER3	OER4	OER5	
WIT.AI	5	5	10	5	5	30%
AMAZON	5	5	10	0	5	25%
DIALOGFLOW	5	10	10	10	5	40%

- *Tasa de error por palabra*

En la Tabla 13, se muestran los resultados obtenidos después de realizar las pruebas. En la prueba de sustitución, fue la que más se obtuvo errores por parte de las herramientas escogidas, en el caso de Dialogflow la palabra “pagar” fue reconocida pero no dio una respuesta satisfactoria al usuario, mientras que en Amazon e IFTTT no reconoció bien la palabra. La única herramienta que paso esta prueba fue WIT.ai, la cual reconoció y dio una respuesta en particular ya que no se ejecutaba alguna acción definida en el modelo.

También se obtuvo un puntaje bajo en la prueba de contrarios, es decir cambiar el verbo y predicado de una oración, si bien es cierto algunas herramientas reconocieron la acción y el dispositivo, pero en la gramática, la estructura de estas oraciones no es adecuada. Finalmente, para la prueba de supresión, la mayoría identificó el dispositivo, pero no genero una respuesta adecuada al usuario, para solventar estos errores se puede validar los modelos y generar una respuesta adecuada. Al tomar las pruebas de tasa de error por palabra descritas en la Tabla 11, el estudio arrojó los siguientes resultados.

Tabla 11*Resultados obtenidos en prueba de Tasa de error por palabra*

API AI	Tasa de error por palabra					Porcentaje
	TEP1	TEP2	TEP3	TEP4	TEP5	
WIT.AI	10	10	10	10	5	45%
AMAZON	10	0	10	10	10	40%
IFFFT	10	0	0	10	0	20%

CONTINÚA 

DIALOGFLOW	10	5	5	10	10	40%
-------------------	----	---	---	----	----	-----

En la Figura 18 se muestra el porcentaje total de esta prueba en las diferentes APIs escogidas.

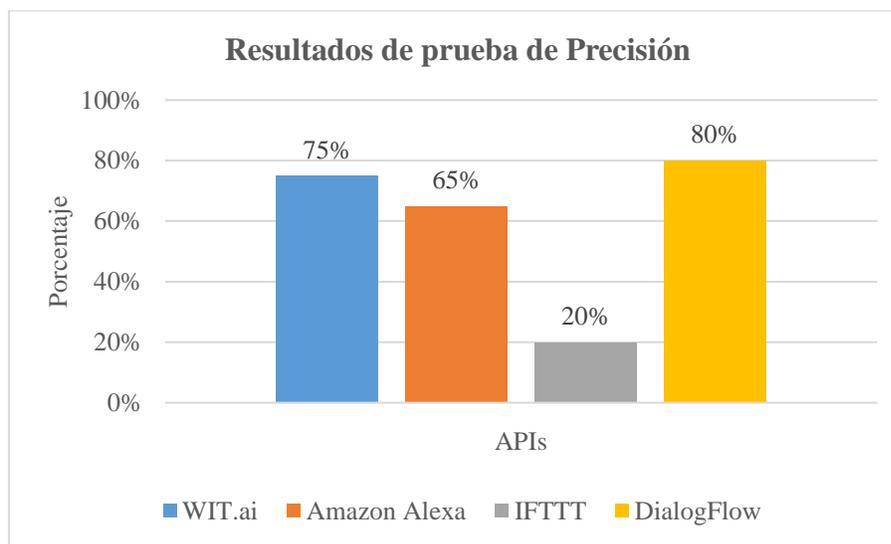


Figura 18. Resultados de prueba de precisión

3.7.3 Pruebas funcionales

Para la ejecución de estas pruebas se han asignado varias tareas que el bot debe realizar con la finalidad de verificar si entiende las órdenes que el usuario asigna. La acción requerida es “encender un dispositivo”, para lo cual se han tomado varias frases en las cuales se encuentran involucrados sinónimos de algunas palabras. Estas instrucciones se describen en la siguiente Tabla 12, las cuales llevan a la misma acción, es decir encender un determinado dispositivo en las diferentes APIs. Con el objetivo de verificar las respuestas que se obtienen de las mismas.

Tabla 12

Pruebas de funcionamiento

API	Entrada
API.AI	Enciende luces
IFTTT	Prender luces

CONTINÚA →

Alexa	Podrías prender las luces
WIT.AI	Podrías encender las luces

Pruebas con DialogFlow

Para realizar las pruebas se utilizaron las frases predeterminadas las cuales arrojaron los resultados deseados, ya que el bot respondió adecuadamente a cada una de ellas.

- **Pruebas con IFTTT**

Al integrar los servicios Google Assistant y Facebook Messenger, cada uno conto con una función. Google Assistant se encargó del reconocimiento automático de voz mientras que por su parte Facebook Messenger brindo la respuesta requerida a la acción ejecutada. Al realizar las pruebas con las instrucciones predeterminadas el bot entendió cada una de estas órdenes y respondió adecuadamente a estas, pero el limitante de esta plataforma es que solo permite una única respuesta al usuario.

- **Pruebas con Alexa**

Las respuestas que otorga Alexa para la misma acción fueron las que se habían determinado en la generación del modelo como lo muestra la siguiente Figura 19. Pueden existir errores al momento de procesar la voz en texto, las razones que podrían provocar estos errores son el hablar rápido o desactivar el micrófono antes de finalizar la oración. La siguiente Figura 19 muestra las pruebas realizadas.

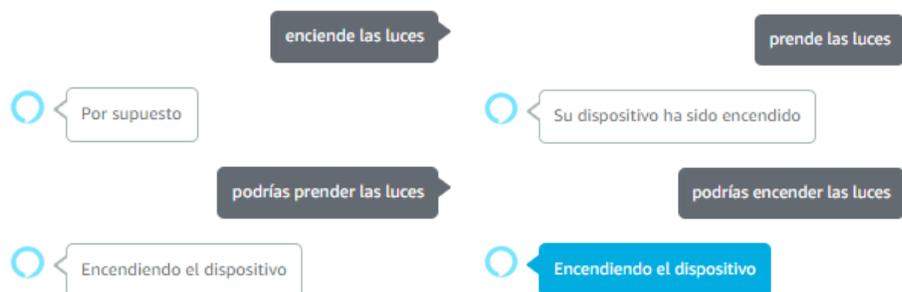


Figura 19. Pruebas funcionales con Alexa

- **Pruebas con Wit.ai**

Al realizar las pruebas de igual forma la plataforma Wit.ai no tuvo mayor dificultad en procesar los audios, se obtuvo el texto en un formato JSON. Los audios enviados se pueden escuchar, leer y modificar de ser necesario dentro de la herramienta Wit. Para la generación de respuestas, el desarrollador programará las respuestas dependiendo de las entidades identificadas.

3.7.4 Resultados de pruebas funcionales

En la Tabla 13, se muestran los resultados obtenidos en la prueba de funcionalidad. En la cual se ingresaron varias órdenes y cada bot dio la una respuesta diferente, esto se debe a la configuración que hace el desarrollador.

Tabla 13

Resultados obtenidos en pruebas de funcionalidad

API	Entrada	Salida
Dialogflow	Enciende luces	Se ha encendido
IFTTT	Prender luces	Con gusto / se ha encendido las luces
Alexa	Podrías prender las luces	Con mucho gusto, por supuesto, su dispositivo ha sido encendido
WIT.AI	Podrías encender las luces	Encendiendo luces (la respuesta depende de la programación)

Se escogió la API Wit.ai por varias razones, si bien es cierto Dialogflow estuvo a la cabeza en la prueba de precisión y es una herramienta poderosa para el desarrollo de un chatbot basado en el lenguaje natural, el cual cuenta con un modelo ASR (reconocimiento automático de voz), que obtuvo mejores resultados al momento de identificar ciertas palabras, pero dependiendo de su uso y el número de interacciones esta herramienta tiene un costo adicional, lo cual no es lo más óptimo para desarrollar el aplicativo.

Por esta razón se escogió a Wit.ai ya que permite un SDK para el desarrollo de un bot de forma gratuita, fácil de implementar e integrar, soporta múltiples idiomas y da la libertad al usuario de poder desarrollar cualquier bot dependiendo de sus necesidades. La herramienta puede responder a comandos escritos o hablados, permitiendo la integración con otras aplicaciones manejando su API HTTP. Tiene mayor facilidad al momento de entrenar al bot el cual detectará entidades o podrá usar las entidades que la herramienta proporciona. Genera un intervalo de confianza al momento de identificar las entidades dentro del texto o voz para facilitar validaciones de ser necesario. Proporciona documentos y ejemplos de código necesarios para el uso de la herramienta.

CAPÍTULO IV

IMPLEMENTACIÓN

Dentro de este capítulo, se hablará sobre la ejecución de la solución propuesta, la cual busca la reducción de accidentes domésticos, asociados a la electricidad dentro del hogar. Para facilitar la comunicación con los dispositivos inteligentes y ejecutar nuestra simulación, hemos implementado la tecnología de OpenHAB. La cual contribuyó con la comunicación y conexión de dispositivos.

4.1 Metodología basada en prototipo

Esta metodología tiene un enfoque basado en un proceso iterativo a partir de un análisis de requisitos inicial donde se realiza prototipos. Dependiendo de la opinión del cliente se incorpora sus sugerencias con respecto al prototipo para la siguiente iteración (Rojas, 2017). Esta metodología conlleva las siguientes etapas:

- **Identificar requerimientos básicos:** En esta etapa se identifica los requerimientos indispensables para el desarrollo del prototipo inicial. El mismo deberá contar con funcionalidades básicas que permita dar soluciones totales o parciales a la problemática.
- **Diseño del prototipo inicial:** Diseño inicial de la interfaz gráfica del prototipo de software.
- **Implementación del prototipo:** Se desarrolla los diseños realizados en la etapa anterior.
- **Pruebas y mejora del prototipo:** En esta etapa se realizan las pruebas correspondientes en cuanto a funcionalidad y cumplimiento de requerimientos que han sido identificados. Se aplicarán los cambios en caso de ser necesario para garantizar la correcta funcionalidad del prototipo.

Se implementó la metodología de desarrollo de software basada en prototipos para cumplir con el desarrollo del aplicativo. Las razones por las cuales se la eligió se detallan a continuación:

- La metodología es óptima para el desarrollo rápido del aplicativo contando con un equipo reducido.
- La aplicación se desarrolló por fases.
- Se realizó una implementación parcial en el chatbot basada en texto, para luego incluir mensajes de voz, posterior a ello se realizaron las respectivas pruebas.
- Los usuarios potenciales pudieron probar la aplicación.

4.2 Primera Fase: Identificar requerimientos básicos

En esta fase se identificaron los requisitos funcionales del prototipo. Para ello se utilizó la especificación de requisitos de usuario, la cual permite determinar requerimientos básicos, contiene información relevante con respecto a lo que el usuario necesita usando un lenguaje coloquial. Para la identificación de los requisitos se utilizó el estándar IEEE 830.

4.2.1 Introducción

4.2.1.1 Propósito

La identificación de requerimientos se la realiza de manera clara y concisa para determinar que funcionalidades debe tener la aplicación y poder brindar una solución a la problemática.

4.2.1.2 Ámbito del sistema

La aplicación fue desarrollada con el fin de ofrecer una solución a la problemática surgida a partir de posibles accidentes domésticos. Los cuales involucra choques eléctricos o incendios, por mal uso de los electrodomésticos o por descuidos de las personas al dejar encendido algún artefacto durante un tiempo prolongado. Repercutiendo en graves consecuencias como pérdidas

materiales. El desarrollo del chatbot fue integrado con el motor de inferencia Wit.ai para el uso del lenguaje natural, donde el usuario podrá interactuar de forma natural con el bot ejecutando la acción requerida hacia un electrodoméstico determinado. Para el control y manejo de los dispositivos se utilizó la plataforma OpenHAB.

4.2.1.3 Alcance del sistema

El presente proyecto abarcó la investigación de los componentes necesarios para implementar la aplicación web/móvil, abordando temas como domótica, OpenHAB, OpenHABian, Raspberry Pi, lenguajes de programación y el motor de inferencia llamado Wit.ai. Cabe recalcar que la aplicación web/móvil es un prototipo funcional, donde se adaptó un smart led simulando que son electrodoméstico o componentes electrónicos como: luces, lavadora, plancha, televisor, etc. El cual se controla de forma remota mediante un chatbot. La aplicación desarrollada cumple con las siguientes funcionalidades:

- Envió de mensajes en forma de texto o voz por parte del usuario hacia el chatbot.
- Identificación de entidades e intenciones del mensaje enviado por el usuario por medio de la API seleccionada.
- Respuestas en tiempo real por parte del chatbot al usuario utilizando al API seleccionada.
- Ejecutar acción requerida en el smart led, por parte del usuario.
- Estado del dispositivo electrónico de forma remota.

4.2.1.4 Definición y acrónimos

Las tablas [14-16], muestran definiciones, acrónimos y abreviaturas que son utilizadas en la presente sección.

Tabla 14*Documento ERS: Definiciones*

TÉRMINO	DEFINICIÓN
Chatbot	Son aplicaciones informáticas basadas en inteligencia artificial que permiten simular la conversación con una persona y entregando respuestas automatizadas.
OpenHAB	Es una plataforma de automatización del hogar de código abierto y tecnología independiente que funciona como el centro del hogar inteligente.
Raspberry Pi	Es un mini ordenador de placa simple a un bajo costo. Destinado principalmente al desarrollo de pequeños prototipos.
Motor de inferencia	Es el componente de un Sistema Experto, encargado de administrar y controlar lógicamente el manejo del conocimiento almacenado en la base.

Tabla 15*Documento ERS: Acrónimos*

ACRÓNIMO	SIGNIFICADO
ERS	Especificación de Requisitos de Software
RF	Requerimiento funcional
RNF	Requerimiento no funcional
API	Interfaz de aplicación de programación
SDK	Kit de desarrollo de software (Software Development Kit)
IDE	Entorno de desarrollo integrado (Integrated Development Environment)
ASR	Reconocimiento automático de voz (Automatic Speech Recognition)

Tabla 16*Documento ERS: Abreviaturas*

ABREVIATURA	SIGNIFICADO
APP	Aplicación
SO	Sistema operativo

4.2.1.5 Referencias

- Especificación de Requisitos según el estándar de IEEE 830.ANSI/IEEE Std. 830-1998.

4.2.1.6 Visión general del ERS

Este documento consta una perspectiva muy general hasta una visión específica, de cada requerimiento y proceso que comprenderá el sistema. También cuenta con toda la información necesaria para entender de manera clara y concisa cualquier duda que se puede generar en el proceso de creación, manipulación o mantenimiento del sistema a tratar en cuestión.

4.2.2 Descripción general

4.2.2.1 Perspectiva del producto

La aplicación web/móvil a desarrollar será un chatbot el cual permitirá la interacción mediante lenguaje natural entre el usuario y dispositivos electrónicos, donde el usuario podrá enviar mensajes de texto y voz para la manipulación y control de los dispositivos.

4.2.2.2 Funciones del producto

La explicación y especificación de esta sección se realizará con la ayuda de la representación de casos de uso para requisitos funcionales del aplicativo. Los casos de uso permiten describir el comportamiento ante la interacción entre el usuario con el sistema en situaciones específicas (Pressman, 2002). La propuesta establecida para la interacción entre el chatbot basado en lenguaje natural, el cual permite el control de los dispositivos electrónicos. Señala al usuario como principal actor, ya que este podrá interactuar con el bot iniciando la conversación vía texto o voz, enviando el mensaje al chat, para que el bot pueda procesar y emitir una respuesta hacia el usuario. Ejecutando la acción que el usuario ha solicitado de forma automática por medio del sistema hacia el dispositivo electrónico. La siguiente Figura 20 detalla las acciones definidas las cuales son secuenciales y muestran el comportamiento del usuario, el sistema y los dispositivos.

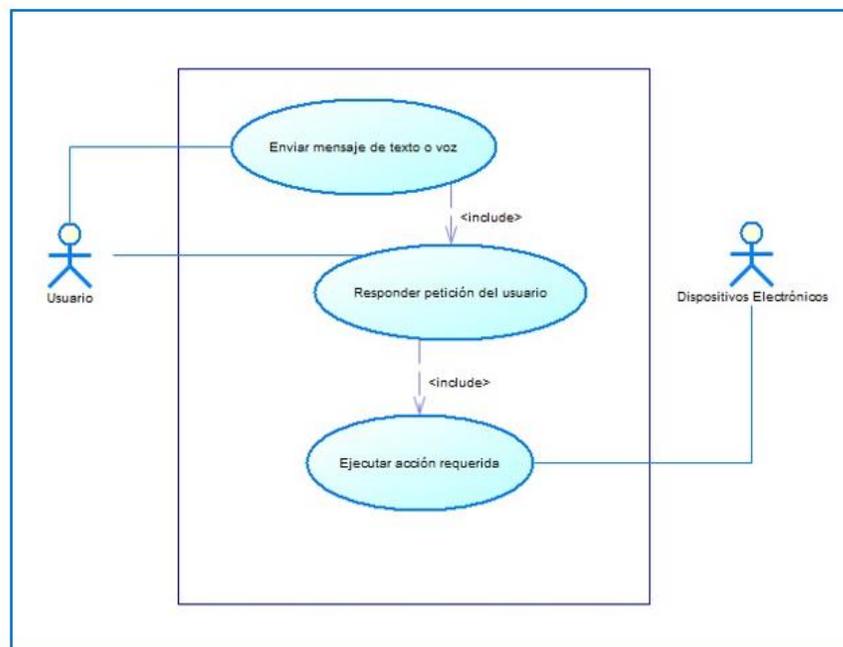


Figura 20. Diagrama de caso de uso.

4.2.2.3 Características del usuario

Los usuarios a los cuales está destinada esta aplicación deberán estar familiarizados con el uso de smartphones para la plataforma Android o aplicaciones web por medio del browser Firefox o Chrome. No existe restricción en cuanto a género o edad.

4.2.2.4 Restricciones

- La aplicación estará disponible para el sistema operativo Android.
- Se utilizó un smart led Mi-light para simular los dispositivos y ejecutar las acciones requeridas por el usuario en este caso encender y apagaras dispositivo.
- El chatbot solo podrá recibir mensajes de texto, o el envío de un audio para su debido procesamiento.
- El usuario solo podrá enviar un mensaje a la vez, donde el chatbot se encargará de contestar su petición en tiempo real.
- Los audios enviados serán en formato .WAV.

- El chatbot solo se limitará a contestar preguntas o peticiones con respecto a la domótica.
- La aplicación solo funciona si el usuario y los dispositivos están conectados a internet.
- Se debe instalar la aplicación “Grabadora de audio” en los smartphones para el envío de audio en tiempo real.
- No se podrá grabar audios desde los navegadores web, pero si enviar archivos de audios previamente grabados.

4.2.2.5 Suposiciones y dependencias

- Los dispositivos electrodomésticos o electrónicos que funcionan por medio de una red deberán estar vinculados a la plataforma de OpenHAB.
- El tiempo que se demoró en ejecutar la acción, dependerá de la red establecida en el hogar.

4.2.3 Requisitos específicos

4.2.3.1 Requisitos funcionales

Los siguientes requisitos descritos en las tablas [17-19], detallan las funcionalidades que realiza el usuario al momento de interactuar con el bot, el cual está basado en lenguaje natural para el control de los dispositivos electrónicos mediante una aplicación web/móvil.

Tabla 17

Requerimiento RF01

Id	RF01
Nombre	Enviar mensajes vía texto o voz al chat.
Descripción	El usuario podrá enviar su petición por medio de un mensaje de texto o mensaje de voz a la aplicación desde su dispositivo móvil, para el control de los dispositivos electrónicos.
Entradas	Solicitud del usuario por medio de voz o texto.
Salidas	Entidades e intenciones identificadas dentro del texto.
Proceso	<ol style="list-style-type: none"> 1. El usuario ingresa a la página web desde su dispositivo móvil. 2. El usuario envía una solicitud (encender/apagar) a un determinado dispositivo, vía texto o voz.

CONTINÚA 

Precondiciones	Conexión entre el software y hardware. Configuración de los dispositivos electrónicos. Conexión entre el chat y la aplicación Wit.ai (motor de inferencia). Ingresar a la página web de la aplicación desde un smartphone.
Post condiciones	Respuesta del bot.
Prioridad	Alta
Rol que lo ejecuta	Usuario

Tabla 18*Requerimiento RF02*

Id	RF02
Nombre	Responder petición del usuario.
Descripción	Según la petición del usuario, el bot dará una respuesta al momento de ejecutar la acción a un determinado dispositivo electrónico.
Entradas	Entidades e intenciones identificadas dentro del texto.
Salidas	Respuesta del bot en forma de texto.
Proceso	<ol style="list-style-type: none"> 1. Captación del texto o voz. 2. Si el mensaje es por medio de voz, el motor de inferencia procederá a realizar el debido análisis para transcribir de voz a texto. 3. Identificación de entidades mediante el motor de inferencia. 4. Validación de respuestas, según la identificación de entidades. 5. Generación de respuesta principal.
Precondiciones	Enviar un mensaje al chatbot.
Post condiciones	Acción del dispositivo.
Prioridad	Media
Rol que lo ejecuta	Usuario

Tabla 19*Requerimiento RF03*

Id	RF03
Nombre	Ejecutar acción por parte de los dispositivos electrónicos.
Descripción	Al generar la respuesta por parte del chatbot, el sistema ejecutará la acción que fue especificada por usuario de forma automática hacia un dispositivo electrónico determinado.
Entradas	Identificación de dispositivo y la acción que desea realizar el usuario.
Salidas	Acción (encender/apagar) que realiza el dispositivo.
Proceso	1. Identificación de dispositivo y la acción.

CONTINÚA 

	2. Ejecución del comando por parte de OpenHab según la acción identificada. 3. Acción realizada por el dispositivo.
Precondiciones	Identificar dispositivo y la acción a ejecutar.
Post condiciones	Si el usuario desea podrá enviar otro mensaje para ejecutar otra actividad o podría dar fin a la conversación.
Prioridad	Baja
Rol que lo ejecuta	Usuario

4.2.3.2 Requisitos no funcionales

Tabla 20

Requisitos no funcionales

Identificador	Nombre	Descripción	Prioridad
RNF01	Interfaz de usuario	La aplicación de chat deberá presentar una interfaz de usuario sencilla y diseño adaptativo a diferentes plataformas	Alta
RNF02	Desempeño	La aplicación de chat deberá responder en tiempo real.	Alta
RNF03	Rendimiento	Se debe garantizar que el uso de la aplicación no afecte el desempeño del dispositivo y tampoco consuma excesivos recursos como la memoria, procesador o genere tráfico de red.	Alta

4.2.3.3 Requisitos comunes de las interfaces

Interfaces de usuario

La interfaz de usuario estará optimizada por medio del diseño responsive, el cual permitirá visualizar la aplicación desde navegadores web y dispositivos Android.

Interfaces de hardware

La aplicación podrá ser ejecutada con todas sus funcionalidades en dispositivos Android y dispositivos con acceso a la web. En la Tabla 21 se presentan los equipos requeridos para el desarrollo del proyecto y los requerimientos que deben cumplir.

Tabla 21

Equipos requeridos para el desarrollo del chatbot

Equipo	Requisitos
Computadora	<ul style="list-style-type: none"> • Sistema Operativo Windows 10 • 8 GB de RAM
Raspberry Pi 3B+	<ul style="list-style-type: none"> • 4 puertos USB 2.0 • Wi-Fi + Bluetooth: 2,4GHz y 5GHz • Micro-SD • Ethernet: Gigabit Ethernet sobre 2.0 • HDMI
Memoria SD	<ul style="list-style-type: none"> • 16 GB de memoria
Televisor	<ul style="list-style-type: none"> • Con entrada HDMI
Teclado	<ul style="list-style-type: none"> • Inalámbrico
Smart Led	<ul style="list-style-type: none"> • 6 vatios de consumo.
Milight	<ul style="list-style-type: none"> • Conexión por medio de wi-fi.

Interfaces de software

Es necesario que los dispositivos sobre los cuales se va a ejecutar la aplicación cuenten con browser de preferencia FireFox o Chrome. La Tabla 22 presenta todas las herramientas utilizadas para el desarrollo del aplicativo con su respectiva versión.

Tabla 22

Herramientas utilizadas para el desarrollo del chatbot

Software	Versión
Node.js	10.0
Node-wit	5.0.0
Node-witai-speech	1.0.2
Preact-cli	2.0.0
Jquery	3.4.0
Pusher.js	4.4.0
Atom (Editor de código)	1.41.0

CONTINÚA 

Sistema Operativo Openhabian	2.4.0
API REST Openhab	

Interfaces de comunicación

La aplicación se conectará al servidor a través de un servicio web basado en el intercambio de mensajes JSON utilizando el protocolo HTTP.

4.3 Segunda Fase: Diseño del prototipo inicial

Una vez identificados los requisitos funcionales, y establecido el diagrama de casos de uso. A continuación, se presentará los diagramas de arquitectura y secuencia con los cuales se muestra la operatividad del aplicativo propuesto.

4.3.1 Diagrama de arquitectura

Un diagrama de arquitectura ayuda a plantear una vista completa del sistema que se va a construir, el cual definirá la estructura y organización de los componentes de software (Pressman, 2002). La Figura 21 muestra la arquitectura, la cual especifica el diseño de la aplicación de un chatbot basado en lenguaje natural para el control de dispositivos electrónicos. Esta comienza con la interacción del usuario con la aplicación web/móvil, el usuario ingresa a la aplicación y envía el mensaje. Este mensaje puede ser de forma textual o puede enviar un archivo de voz. Se recibe el mensaje desde la interfaz de la aplicación y el contenido del mensaje es enviado por medio del método POST del protocolo HTTP, dentro del servidor node.js, para ser enviado a la API Wit.ai, la cual procede con el procesamiento de lenguaje natural, detectando las entidades involucradas dentro del texto.

Una vez identificadas las entidades estas serán comparadas para validar las respuestas que serán generadas a partir de la detección de la entidad apropiada que ha coincidido o se generará

una respuesta predeterminada en el caso de no coincidir con alguna entidad dentro del servidor de la aplicación. La respuesta será enviada a la interfaz de la aplicación la cual será receptada por el usuario. De igual manera, al momento de comparar las entidades, se enviará un comando (ON/OFF) dependiendo de lo que el usuario ha requerido dentro del texto para el control del dispositivo por medio de las reglas preestablecidas en el servidor. Donde se podrá actualizar u obtener el estado del dispositivo consumiendo la API REST que proporciona la plataforma de OPENHAB y de donde el servidor OpenHABian por medio de una Raspberry Pi se encuentra en funcionamiento para la manipulación de los dispositivos electrónico.

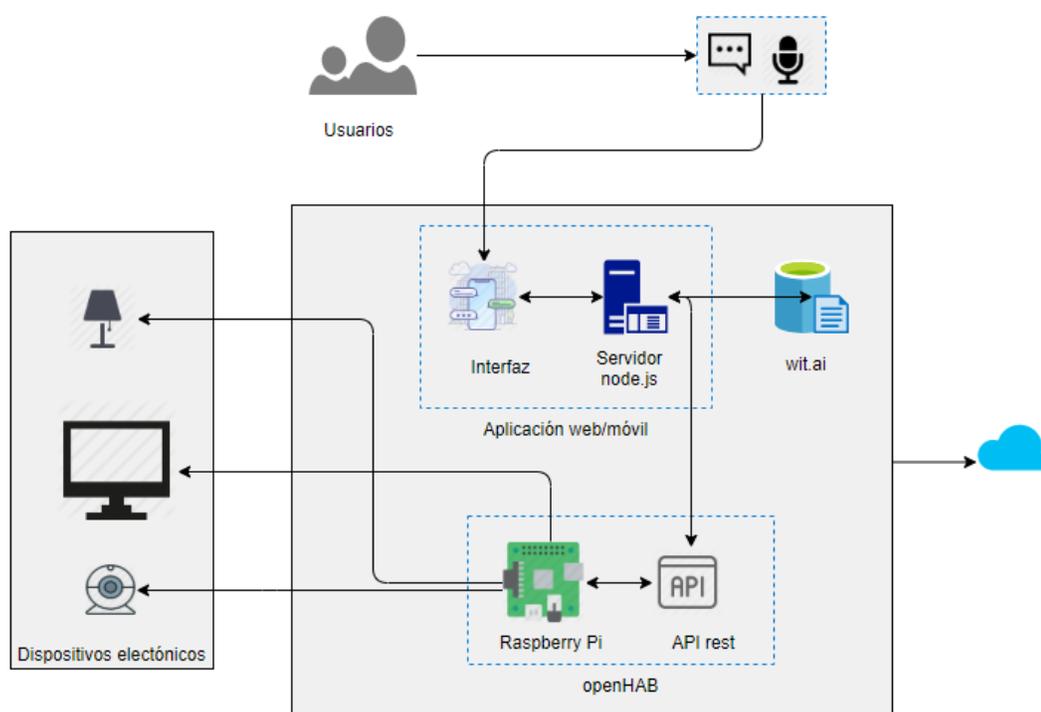


Figura 21. Diagrama de arquitectura

4.3.2 Diagramas de secuencia

Las interacciones entre objetos que conforman un sistema en orden secuencial los cuales ocurren en un determinado escenario se muestran a partir de diagramas de secuencia (Booch,

Rumbaugh, Jacobson, Martínez, & Molina, 1999). El usuario podrá interactuar con los dispositivos por medio de la aplicación propuesta la cual es una interfaz web/móvil. La Figura 22 muestra las secuencias correspondientes a la interacción con el motor de inferencia Wit.ai, quien es el que maneja el procesamiento de lenguaje natural para la identificación de entidades previamente aprendidas y el servidor OpenHABian, el cual es un servicio que proporciona el control y manejo de los dispositivos.

El usuario ingresa a la aplicación web/móvil y deberá enviar un mensaje ya sea texto o voz, una vez realizado esto el servidor de la aplicación captará el mensaje, este mensaje será enviado al motor de inferencia dentro del cual se realizará la detección de las entidades e intenciones de acuerdo al texto ingresado por el usuario, si el mensaje es un archivo de voz se procederá a realizar la traducción de voz a texto. Después de haber realizado el análisis del texto se retornará un Json con las entidades detectadas para ejecutar las validaciones correspondientes al momento de generar la respuesta al usuario dentro del servidor de la aplicación. Al momento de generar la respuesta y enviarla a la interfaz del chat, también se envía el comando a la API REST de OPENHAB para ejecutar la acción por medio del servidor OpenHABian que controla los dispositivos que hayan sido involucrados dentro del texto o mensaje enviado por el usuari

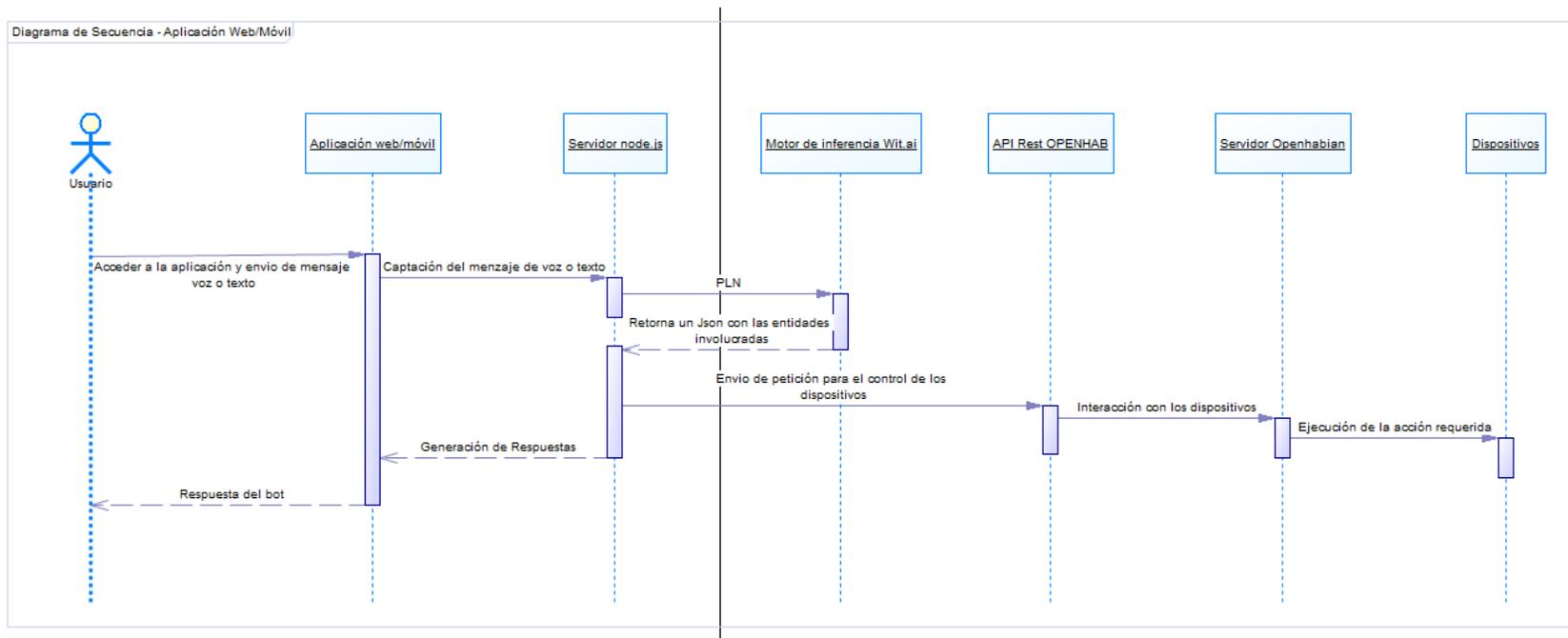


Figura 22. Diagrama de Secuencia – Aplicación Web/Móvil

4.3.3 Sketch board inicial del prototipo

Dentro de esta sección se realizará un prototipo de chatbot. El prototipo mostrará la interacción entre el usuario y el chatbot, mediante mensajes de texto y la importación de un audio de voz. La Figura 23 muestra el flujo de interacciones por medio de mensajes de texto y voz. Se puede apreciar dentro de la imagen la importación de archivos de audio, mediante dos botones. El botón “examinar” permite grabar y seleccionar un archivo de audio, mientras que “subir” permite la carga del archivo hacia el sistema.

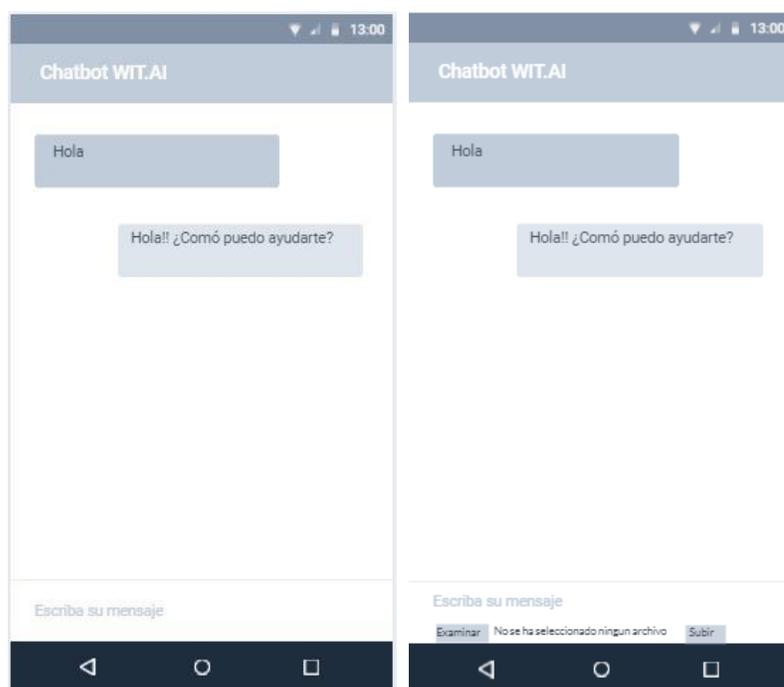


Figura 23. Interfaz inicial del prototipo

4.4 Tercera Fase: Implementación del prototipo inicial

Posterior a definir los requisitos funcionales, historias de usuario y diagramas de caso de uso, además de los diagramas de secuencia y arquitectura. Dentro de esta sección se detallará la

implementación del chatbot basado en lenguaje natural para el control de dispositivos electrónicos.

4.4.1 Selección de herramientas

Se describirán las herramientas seleccionada para el desarrollo del aplicativo:

- **Node**

Entorno que permite la ejecución del lenguaje JavaScript del lado del servidor. Posibilita el manejo de eventos asíncronos orientados al diseño de aplicaciones en redes escalables. Esta herramienta fue seleccionada ya que sirve de intermediario entre la comunicación de la aplicación web/móvil, el motor de inferencia Wit.ai y la ejecución de reglas para el control de los dispositivos. La herramienta maneja algunas librerías que facilitan la integración de la aplicación web/móvil con las plataformas anteriormente mencionadas.

- **Preact**

Biblioteca de JavaScript, la cual se describe como una alternativa rápida ya que se puede obtener una aplicación Preact de forma instantánea, lista para su funcionamiento. Esta herramienta facilitó el desarrollo web de la interfaz mediante la implementación de componentes reutilizables.

- **Wit.ai**

Wit.ai se integra con aplicaciones y sitios web como Facebook Messenger. Se escogió esta herramienta puesto que fue la mejor opción al realizar la comparación y no genera costo alguno. Para poder integrar la aplicación generada en Wit.ai se debe utilizar uno de los clientes oficiales.

La mayoría de ejemplos se encuentran alojados en el repositorio de GitHub donde se puede encontrar el material necesario en diferentes lenguajes de programación.

- **OpenHAB**

Se ha seleccionado openHAB ya que es open source y permite la conexión con el servidor node.js. Facilita la configuración entre los dispositivos eléctricos que el usuario desee controlar. Para esto se ha utilizado la API REST de openHAB, la cual permite tener acceso a todos los datos relacionados a los dispositivos (ítem) y la conexión con estos (bridge), así como la capacidad para invocar acciones que pueden cambiar el estado de los dispositivos o influir en el comportamiento de otros elementos de openHAB. Las interacciones con la API REST se basan en el protocolo http.

4.4.2 Implementación de aplicación web

Las tecnologías web: CSS, HTML y lenguaje JavaScript permitirán el desarrollo del aplicativo. La aplicación se desarrolló a través de preact, donde a partir de ella se construyó un simple chat, el cual permite el ingreso de texto y la incorporación de un archivo de audio. La Figura 24 muestra la interfaz a la cual el usuario accederá. Esta aplicación se encuentra alojada en la nube, servicio que provee Microsoft Azure. Al generar el servidor es importante instalar las dependencias necesarias, se configuró una ruta /chat la cual está definida en la siguiente Tabla 23, con la finalidad de recibir los mensajes desde la interfaz, para ser enviados a la API de Wit.ai.

Tabla 23

Estructura de ruta

Parámetro	Descripción	Ejemplo
App	Instancia de la librería express	app.method(path, handler)
METHOS	Método de solicitud HTTP.	
PATH	Vía de acceso al servidor	app.post('/chat', function (req, res) => {

CONTINÚA 

HANDLER	Función ejecutada cuando se correlaciona la ruta.	<pre>const { message } = req.body; }</pre>
---------	---	--

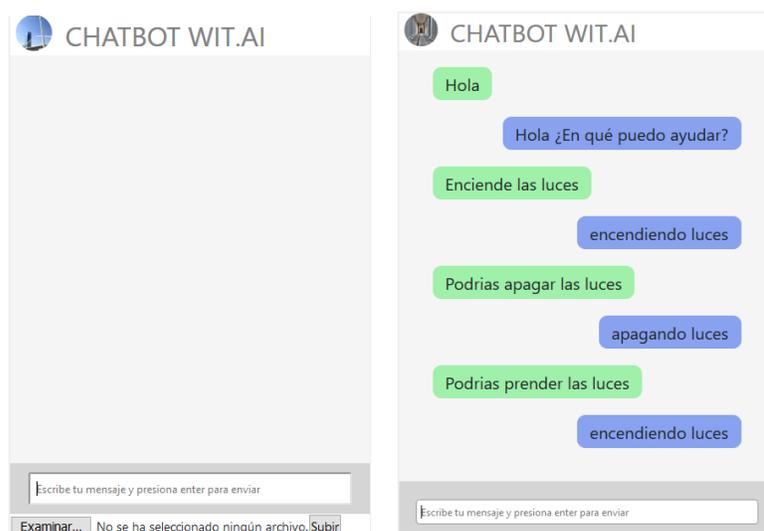


Figura 24. Interfaz de aplicación web.

4.4.3 Implementación Cliente Wit.ai

Para integrar la aplicación y la herramienta Wit.ai se implementó un constructor. El constructor Wit, contiene un parámetro denominado “accessToken”, el cual permite la autenticación de las herramientas mediante un token de acceso generado por el servidor de Wit.ai. Los parámetros del constructor se encuentran definidos en la Tabla 24. De igual forma es necesario configurar un componente en el servidor, para el intercambio de mensajes a través de la interfaz de la aplicación desarrollada, a la herramienta Wit.ai para su debido procesamiento.

Para esto se implementó el SDK para node.js, el cual proporciona una clase Wit definida por el método message. Este método permite enviar el texto a la aplicación para la detección de

entidades e intenciones generadas en el modelo de datos, devolviendo las entidades extraídas de la oración. El método se especificará en la Tabla 25.

Tabla 24*Constructor Wit*

Parámetro	Descripción	Tipo	Ejemplo
accessToken	Token de acceso de la instancia generada en la herramienta Wit.ai.	Obligatorio	process.env.WIT_ACCESS_TOKEN
Logger	Objeto que maneja el registro.	Opcional	
apiVersion	Versión de la API a usar en lugar de la recomendada.	Opcional	

Tabla 25*Método Message*

Parámetro	Descripción	Tipo	Ejemplo
Message	Texto del cual Wit.ai extrae información.	Obligatorio	“enciende las luces”

Ejemplo JSON de Respuesta

```
{ "_text": "enciende las luces", "entities":
{"getAccion": [{"confidence": 0.7802406966934, "value": "Encender", "type": "value"}],
"dispositivo": [{"confidence": 0.99708792971907, "value": "luces", "type": "value"}]},
"msg_id": "1xGDnMOTvseC8fUU" }
```

El formato de respuestas de la herramienta Wit.ai es JSON para solicitudes exitosas, mientras que para una fallida se generaría un error de texto sin formato. La detección de las entidades dentro del texto enviado por el usuario, permitirá validar las respuestas que se generaran según la acción y el dispositivo identificado. El formato de respuestas JSON sigue un formato en general, ya sea en texto o bien sea una onda de audio. Este formato se definirá en la Tabla 26 donde se especifica el uso de varios parámetros establecidos. El nivel de confianza indica cuan preciso es

Wit al detectar las entidades dentro del texto. Mientras mayor sea la aproximación a 1 es extremadamente confiado, y baja confianza al ser menor de 0.5.

Tabla 26

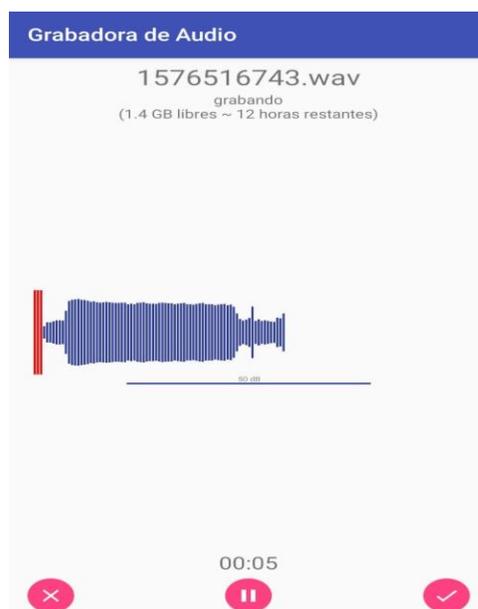
Formato de respuestas Wit.ai

Nombre	Descripción	Tipo	Ejemplo
_text	Texto o audio enviado.	String	<pre>{ "_text": "Hola", "entities": { "saludos": [{ "confidence": 0.68488672992781, "value": "hola", "type": "value" }] }, "msg_id": "1vBvfi7Da0PvcfBbt" }</pre>
Intent	Intención del usuario	String	
Entities	Objeto de entidades. Cada una estará contenida en una matriz de valores.	Object	
confidence	Representa el nivel de confianza del resultado	Number	
msg_id	Un id generada por la API.	String	

4.4.4 Implementación de audio

Wit.ai puede recuperar el significado extraído de un archivo de audio o transmisión. En este caso se ha implementado el módulo “node-witai-speech”. El cual facilita la extracción, y recuperación del significado de un archivo de audio en varios formatos permitidos, como lo son: mpeg, wav, ogg, entre otros, teniendo un máximo de duración de 30 segundos. Para la aplicación desarrollada se utilizó archivos con formato .wav, donde se incorporó la aplicación “Grabadora de voz”. Esta herramienta permitió grabar el audio en este formato como lo muestra la Figura 25. A demás de ello se genera de forma automática el archivo de audio que será enviado al servidor

Figura 25. Implementación audio dentro del chat.



Al enviar el archivo de audio al servidor, los parámetros enviados para realizar su procesamiento por Wit.ai son: la ubicación del archivo, un content-type para establecer el tipo de

formato de audio y el encabezado de autorización que será el token generado por el servidor de Wit.ai. En el caso de generar con éxito la extracción del audio, el código HTTP será 200 donde se mostrará un formato JSON con el texto del audio reconocido y las entidades identificadas. Si surge algún error se notificará con los códigos establecidos en la Tabla 27.

Tabla 27
Códigos de error

Código HTTP	Descripción	Ejemplo
200	Éxito	<pre>{ "_text": "Hola", "entities": { "saludos": [{ "confidence": 0.68488672992781, "value": "hola", "type": "value" }] }, "msg_id": "1vBvfi7Da0PvcfBbt" }</pre>
400	Falta del cuerpo o tipo de contenido, o contenido desconocido, reconocimiento de voz no válido o los parámetros no son válidos.	
401	Incorrecta autenticación de token	
408	Solicitud caducada, audios muy largos	
500-503	Problemas en el servidor	

4.4.5 Implementación de respuestas

La Figura 26 muestra las respuestas que se generan al momento de interactuar el usuario con el bot. Para la implementación de respuestas se ha generado una validación. Donde las entidades, en especial las acciones, los dispositivos y las posibles localidades son comparados. Cada respuesta generada, depende de la condición en la que las acciones y dispositivos se vean involucrados. Dentro del servidor Node.js y de la ruta que se especificó /chat se configuró un objeto denominado responses. El cual se muestra en la Tabla 28, este contiene varias respuestas para cada entidad predefinida en la aplicación de Wit.ai. La función handleMessage(), permite generar la validación y la respuesta a cada mensaje recibido.

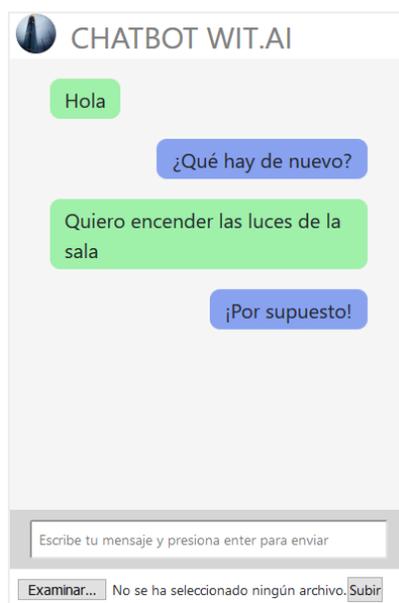


Figura 26. Respuestas del bot

Tabla 28

Implementación de respuestas

```
const responses = {
  greetings: ["Hola, ¿En qué puedo ayudarte ?", "¿Qué hay de nuevo?"],
}
const handleMessage = ({ entities }) => {
  const greetings = firstEntityValue(entities, 'saludos');
  if (greetings) {
    return pusher.trigger('bot', 'bot-response', {
      message:
        responses.greetings[
          Math.floor(Math.random() * responses.greetings.length)
        ],
    });
  }
}
```

Los canales se utilizan para chat, juegos, notificación, IoT entre otros. La implementación y configuración de canales bajo la herramienta Pusher, generará mayor expectativa en el usuario, ya que el bot responderá en tiempo real. Para la publicación de eventos la herramienta proporciona bibliotecas con métodos especiales para desencadenar eventos, realizando una

llamada a la API HTTP. La siguiente Tabla 29 definirá el paso de parámetros a la función trigger, la cual será definida dentro del servidor Node.js

Tabla 29

Parámetros definidos para la comunicación en tiempo real

Nombre	Descripción	Tipo	Ejemplo
Canales	Cadena que identifica un solo canal o una matriz de cadenas.	String	<pre>var Pusher = require('pusher'); var pusher = new Pusher({ appId: 'APP_ID', key: 'APP_KEY', secret: 'APP_SECRET', cluster: 'APP_CLUSTER' }); pusher.trigger(channels, event, data, socketId, callback);</pre>
Evento	Nombre del evento que se activará	String	
Datos	Datos enviados en el evento, en formato JSON.	String	
socket_id	ID de socket de un cliente.	Opcional	
Devolución de llamada	Funcion de devolucion de llamada que se invoca cuando esta se completa.	Opcional	

La conexión de canales es un medio fundamental de comunicación con el servidor. Para esto es necesario establecer una conexión bidireccional donde puede recibir y emitir mensajes desde el servidor. Para abrir la conexión a los canales se implementará el método subscribe(), definiendo un canal llamado “bot”, y el bot escuchará por medio del canal “bot-response”. Para la actualización y estado de la aplicación al momento de recibir un mensaje se aplicó el método bind().

4.4.6 Conexión con openHAB

Se manejó la Raspberry pi 3 b+ para el uso del software OpenHAB. En la cual se montó el sistema operativo openHABian que se basa en el sistema estándar Raspbian Lite. Esta Raspberry pi se la utilizó como servidor para el control de los dispositivos. Para realizar la conexión entre el

bot y los dispositivos electrónicos se utilizó la API REST de openHAB. La cual permite el acceso a los dispositivos y al estado de estos, así como actualizaciones de estado o el envío de comandos para su control. La interfaz REST es muy rápida, por lo que es buena para la interacción en tiempo real con openHAB. De igual forma para el acceso remoto a los dispositivos en tiempo real, se utilizó my.openHAB.org, el cual es un servicio en la nube. Permitiendo al usuario controlar los dispositivos sin necesidad de estar en la misma red, igualmente permite verificar el estado de los elementos expuestos, notificaciones y eventos.

Al utilizar my.openHAB.org se debe registrar con anterioridad. Para acceder a este servicio se utilizó el siguiente url, para iniciar sesión desde el servidor node.js “http://username:password@host:port/rest”. Para obtener una colección de todos los dispositivos declarados se accedió a la siguiente ruta: “http://localhost:8080/rest/items”. Por lo tanto, se puede acceder a dispositivos individualmente. Como por ejemplo un termostato bajo la siguiente url: “http://localhost:8080/rest/items/Temperature_FF_Office”.

Para acceder directamente al estado de un dispositivo se utilizó la siguiente url: “http://localhost:8080/rest/items/Temperature_FF_Office/state”. Del mismo modo, se puede enviar actualizaciones de estado a través de reglas. Las reglas se utilizan para automatizar procesos. Cada regla se puede activar, lo que invoca un script que realiza cualquier tipo de tarea. Por ejemplo: Encender luces, iniciar temporizadores, etc. OpenHAB ofrece crear estas reglas utilizando varios lenguajes de programación, para lo cual se ha elegido jquery ya que es el lenguaje más sencillo de implementar en el aplicativo.

Jquery proporciona el método Ajax para realizar peticiones al servidor. Este método se utilizó para las peticiones get que obtiene el estado actual de un ítem, post envía un comando a un ítem o put establece el estado de un ítem. Para seleccionar entre las distintas opciones, se utiliza un objeto configurable. El cual es un objeto JavaScript con varios valores que permiten configurarlo según las necesidades de la petición Ajax. Se utiliza la función done(), la cual se ejecuta cuando se obtiene una respuesta con éxito del servidor y la función fail(), si ha ocurrido algún error. Se detalla más al respecto en la Tabla 30.

Tabla 30*Solicitud al servidor openHAB*

Nombre	Descripción	Tipo	Ejemplo
Type	tipo de la petición, GET, PUT o POST.	PlainObject	<pre> var request = \$.ajax ({ type : "POST", url: "http://192.168.100.21:8080 /rest/items/MyLight/state", data : txtCommand, headers : {"Content-Type": "text/plain", "Accept": "application/json"}, dataType : "text" }); </pre>
url	dirección a la que se envía la petición.	String	
Data	datos a enviar al servidor.	PlainObject	
Headers	encabezados de solicitud.	PlainObject	
dataType	tipo de datos que se espera del servidor	String	

4.4.7 Configuración de dispositivo

El dispositivo que se ha utilizado para realizar las pruebas es el foco de mi-light iBox. Inicialmente, este dispositivo se lo debe vincular a la red doméstica utilizando la aplicación del fabricante. Una vez hecho esto se podrá configurar el dispositivo a través de openHAB, empleando la interfaz Paper, la cual permite agregar automáticamente el enlace del dispositivo. Este permite la comunicación entre openHAB y el dispositivo como se muestra en la Figura 27.

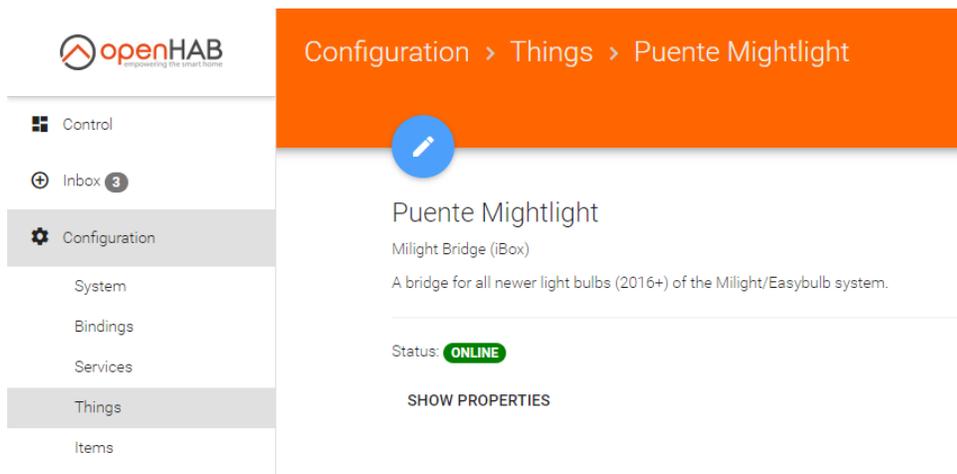


Figura 27. Configuración del enlace del dispositivo.

Una vez realizada la configuración del enlace, se procede con el foco. OpenHAB mostrará una gran variedad de modelos disponibles para el cual se eligió Integrated bulb (iBox). Para la configuración se debe agregar el nombre, el ID del dispositivo, el enlace antes creado y si se desea se puede agregar una zona si la intención es controlar varios dispositivos a la vez. A continuación, en la Figura 28 se muestra la configuración realizada.

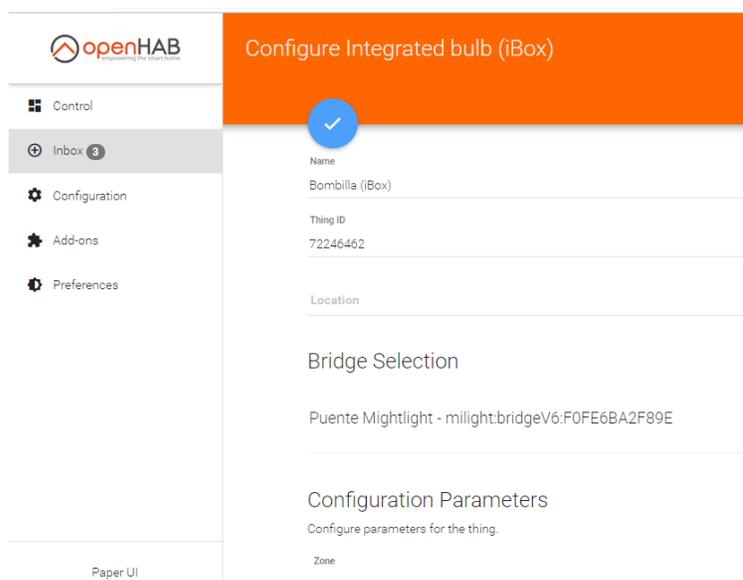


Figura 28. Configuración del dispositivo

4.5 Cuarta Fase: Pruebas y mejora del aplicativo

Las pruebas realizadas al aplicativo se basaron en el enfoque de caja negra, buscando mantener un correcto funcionamiento al momento de ejecutar una tarea hacia el bot, rendimiento y diseño de interfaz amigable, sencilla e intuitiva. Las pruebas realizadas se han decidido dividir las en tres etapas: Pruebas de requerimiento, pruebas de funcionalidad y pruebas de interfaz gráfica.

4.5.1 Etapa I: Pruebas de requerimientos

Se evaluó los requerimientos iniciales y funcionalidades básicos que debe cumplir el aplicativo. Estas pruebas fueron realizadas por un usuario potencial. La Tabla 31 muestra si los requerimientos fueron cumplidos o si hubo algún inconveniente.

Tabla 31

Pruebas de requerimientos

REQUERIMIENTO	CUMPLIÓ CON EL REQUERIMIENTO		COMENTARIOS
	SI	NO	
Enviar mensajes vía texto	X		Aprobado
Enviar mensajes vía voz	X		Solo admite audios con formato .wav y se debe instalar una aplicación móvil para grabar la voz.
Responder petición del usuario	X		Aprobado
Ejecutar acción por parte de los dispositivos electrónicos	X		Aprobado

En gran parte los requerimientos funcionales del aplicativo se ejecutan de forma correcta únicamente existe una limitante al momento de enviar audios en el chat, ya que se requiere que estos estén bajo un formato .wav y se debe instalar un programa para grabar audios. Otra

limitante identificada es que el usuario solo puede enviar un mensaje a la vez para que se realice la acción requerida por parte del dispositivo.

4.5.2 Etapa II: Pruebas de Funcionalidad

Con estas pruebas se buscó verificar si el funcionamiento del aplicativo fue óptimo, y si se obtenían los resultados esperados. Para realizar esta prueba se tomó en cuenta dos escenarios:

- Envío de mensajes de texto
- Envío de mensajes de voz

También se tomó en cuenta que acciones puede realizar el usuario al momento de interactuar con el bot como lo muestra la Tabla 32.

Tabla 32
Pruebas de funcionalidad

ACCIONES	ESCENARIO	FUNCIONÓ CORRECTAMENTE		COMENTARIOS
		SI	NO	
Encender dispositivos	1	X		Al momento de ejecutar la prueba el usuario mencionó que un dispositivo no se encontraba almacenado en el motor de inferencia wit.
	2	X		
Apagar dispositivo	1	X		Los errores lingüísticos al pronunciar algunas palabras, pueden generar problemas al momento de reconocer las acciones.
	2	X		
Verificar estado del dispositivo	1	X		Al realizar esta prueba el usuario no tenía idea de cómo preguntar o verificar el estado del dispositivo.
	2	X		

4.5.1 Etapa III: Pruebas de Interfaz Gráfica

En la Tabla 33 se describe los resultados de las pruebas realizadas respecto al grado de aceptación del usuario en cuanto a la interfaz gráfica del aplicativo:

Tabla 33
Pruebas de Interfaz gráfica

Preguntas	Cumplió con el requerimiento		Observaciones
	SI	NO	
La interfaz es amigable con el usuario	X		Aprobado
La interfaz es intuitiva	X		Aprobado
Cuenta con un diseño adecuado	X		Aprobado
Utiliza controles (botones) adecuados para cada funcionalidad	X		Al momento de ejecutar pruebas con la voz el usuario prefiere que se envíe de forma automática el audio utilizando solo un botón, en cuanto al texto se requiere un botón para enviar.
Resulta fácil de aprender a manejarla	X		Aprobado

CAPÍTULO V

VÁLIDACIÓN DE HIPÓTESIS

Retomando la hipótesis, ya antes mencionada dentro del primer capítulo. La utilización de un aplicativo para el control de artefactos eléctricos basado en un chatbot e interacción con lenguaje natural, disminuirá el riesgo de accidentes domésticos. Para la validación de la solución propuesta se llevó a cabo el análisis de resultados obtenidos al realizar las pruebas de aceptación y usabilidad por parte de los usuarios. La finalidad de este análisis es comprobar si la aplicación desarrollada puede ayudar de cierta forma a disminuir los accidentes domésticos que generan los electrodomésticos en el hogar.

5.1 Estudio realizado

El objetivo principal del aplicativo es determinar si se puede disminuir los accidentes domésticos dentro del hogar. La aplicación desarrollada, permite interactuar con un bot tanto en texto como en voz. Para el control de los dispositivos se ha integrado un smart led el cual simulará los aparatos electrónicos dentro del hogar. Las pruebas de usuario se realizaron con 10 personas. La primera prueba que se realizó fue la de funcionamiento la cual consistió en realizar las siguientes tareas:

- Encender un dispositivo
- Apagar un dispositivo
- Verificar el estado de un dispositivo

Los usuarios interactuaron con el bot enviando mensajes de texto y de voz. Observaron si el dispositivo en este caso el smart led, ejecutaba las tareas asignadas y además la respuesta del bot.

Posterior a ello se procedió a contestar una encuesta sobre el grado de aceptación y usabilidad del aplicativo, sumándose comentarios acerca del mismo, obteniendo ciertas sugerencias para mejorar la aplicación. Para la preparación y ejecución de las pruebas se estableció lo siguiente:

- Definición de perfiles de usuario y tareas asignadas.
- Establecer datos a registrar (comandos de voz y texto del participante).
- Preparación del entorno físico (espacio físico, smartphone, smart led).
- Guion del test. Incluye bienvenida, explicación del test, consignas para tareas, cuestionario posterior y despedida.
- Prueba piloto del test.
- Ajustes de la aplicación, dispositivos de interacción y entorno (Raspberry Pi, Microsoft Azure, Openhab Cloud, Grabadora de voz y Smart led).

5.2 Características de los usuarios

En la Tabla 34 se detalla las características demográficas de los usuarios potenciales. Donde se segmenta por rango de edades a las 10 participantes, obteniendo dos grupos. El primer grupo, con un rango de edad de 18 a 23 años, corresponden a un determinado número de estudiantes de varias carreras universitarias. El segundo grupo, con un rango de edad de 45 a 65 años, corresponde a un pequeño número de profesionales en varios campos distintos a la informática. Cabe recalcar que los participantes deben tener un conocimiento básico en cuanto al uso de smartphones y envío de mensajes por chats.

Tabla 34
Características demográficas de los participantes

Características	Descripción
Número de encuestados	10
Género	Masculino y Femenino
Edad	[18 a 23] y [45 a 65] años

5.3 Encuesta

La encuesta está dividida en tres partes. La primera parte recolecta información sobre la persona y los tiempos al ejecutar las tareas asignadas; La segunda, mide el grado de aceptación de la aplicación en la que el usuario asigna una calificación; La tercera y última parte, los participantes respondieron algunas preguntas evaluando la usabilidad del sistema, adicional a ello se receptaron sugerencias acerca de la aplicación y si la misma podría ayudar a prevenir accidentes domésticos. Todo esto se llevó a cabo con el objetivo de analizar la información recolectada.

5.3.1 Prueba de funcionalidad

Los datos personales de los participantes fueron registrados en esta sección. Posterior a ello se realizaron las pruebas de funcionamiento con el fin de registrar el tiempo (segundos) que se demora el usuario al ejecutar cada tarea. El participante envió mensajes de texto y de voz, mediante el chatbot para controlar un determinado dispositivo. Las tareas realizadas se describen a continuación:

- **Tarea 1. Encender el dispositivo:** Esta tarea consiste en encender el dispositivo ya sea: luces, televisión, refrigeradora, etc.

- **Tarea 2. Apagar el dispositivo:** La tarea consiste en apagar el dispositivo ya sea: luces, televisión, refrigeradora, etc.
- **Tarea 3. Obtener el estado del dispositivo:** Esta tarea consiste en obtener el estado del dispositivo ya sea: luces, televisión, refrigeradora, etc.

Dentro de esta sección se obtuvo el porcentaje de culminación de tareas asignadas (CTA). El cual se calculó mediante la sumatoria de todas las tareas realizadas (TTR) para el total de las tareas asignadas (TTA) multiplicado por 100 como lo muestra la siguiente fórmula.

$$\%CTA = \frac{\sum TTR}{TTA} * 100$$

De igual forma se registró en número de errores al realizar cada tarea y las asistencias o ayudas que se brindó a cada participante. Por último se determinó el tiempo promedio para culminación de cada tarea asignada, ya sea por voz o texto. A continuación se muestra la primera parte de la encuesta realizada.

Nombre:

Datos:

Edad: Sexo:

Trabajo/Estudios:

Uso de internet:

1 vez a la semana

5 veces como mínimo a la semana

más de una hora al día

Tarea 1: Encender el dispositivo

	Voz	Texto
	<input style="width: 40px; height: 20px;" type="text"/>	<input style="width: 40px; height: 20px;" type="text"/>
Tiempo	Seg.	Seg.

Errores:

Comentarios

Tarea 2: Apagar el dispositivo

	Voz	Texto
	<input style="width: 40px; height: 20px;" type="text"/>	<input style="width: 40px; height: 20px;" type="text"/>
Tiempo	Seg.	Seg.

Errores:

Comentarios

Tarea 3: Obtener el estado del dispositivo

	Voz	Texto
	<input style="width: 40px; height: 20px;" type="text"/>	<input style="width: 40px; height: 20px;" type="text"/>
Tiempo	Seg.	Seg.

Errores:

Comentarios

5.3.2 Prueba de grado de aceptación

Dentro de esta sección se midió el grado de aceptación del aplicativo y se receptó opiniones o sugerencias del participante. Para evaluar el grado de aceptación, el participante calificó con un valor de 1 a 5, si está totalmente en desacuerdo o totalmente de acuerdo respectivamente. Para

mayor análisis se procedió a calcular el porcentaje de aceptación por cada grupo de usuario. A continuación, se mostrará la segunda parte de la encuesta.

1. En general estoy satisfecho con la facilidad de realizar estas tareas

Totalmente en desacuerdo..... Totalmente de acuerdo

12345

2. En general estoy satisfecho con el tiempo empleado en realizar estas tareas

Totalmente en desacuerdo..... Totalmente de acuerdo

12345

3. En general estoy satisfecho con los mensajes de ayuda al realizar estas tareas

Totalmente en desacuerdo..... Totalmente de acuerdo

12345

¿Cuál es su opinión con respecto a la aplicación?

5.3.3 Usabilidad del sistema

En la tercera sección se analizó la usabilidad del aplicativo, donde se adaptó la escala de usabilidad del sistema (SUS), la cual consiste en un cuestionario de 9 preguntas con cinco opciones de respuestas para los encuestados. Se utilizó estas preguntas, ya que se puede ejecutar en muestras pequeñas con resultados confiables. Para obtener la puntuación SUS, se suman todas las preguntas impares (PI) y se resta cinco. $x_{impares} = \sum PI - 5$. De igual forma se procede a sumar las preguntas pares (PP) y se resta veinticinco $y_{pares} = \sum PP - 25$. Estos resultados son sumados, para ser multiplicados por 2,5.

$$\text{Puntuación SUS} = (x_{\text{impares}} + y_{\text{pares}}) * 2,5$$

Se calculó el porcentaje de usabilidad, obteniendo el promedio de todas las puntuaciones SUS alcanzadas por cada participante. Después de realizar este cuestionario (Véase Tabla 35), los participantes respondieron dos preguntas con respecto a la problemática planteada para determinar si el aplicativo puede ayudar en gran parte a disminuir accidentes domésticos. A continuación, se mostrará la última parte de la encuesta realizada.

Responda las siguientes preguntas con la puntuación del 1 al 5, donde 1 es totalmente desacuerdo y 5 es totalmente de acuerdo.

Tabla 35
Cuestionario

Preguntas	Puntuación
1. ¿Le gustaría usar esta aplicación con frecuencia?	
2. ¿Piensa que la aplicación es fácil de usar?	
3. ¿Cree que es necesario el apoyo de una persona técnica para poder utilizar la aplicación?	
4. ¿Piensa que las diversas funciones en la aplicación están bien integradas?	
5. ¿Cree que hubo errores en la aplicación?	
6. ¿Piensa que la mayoría de las personas aprenderían usar la aplicación rápidamente?	
7. ¿Se siente incómodo usando la aplicación?	
8. ¿Se siente seguro usando la aplicación?	
9. ¿Cree que necesita tener conocimientos previos antes de utilizar la aplicación?	

¿Cree que con el uso de la aplicación reducirían los accidentes domésticos? Si, No ¿Por qué?

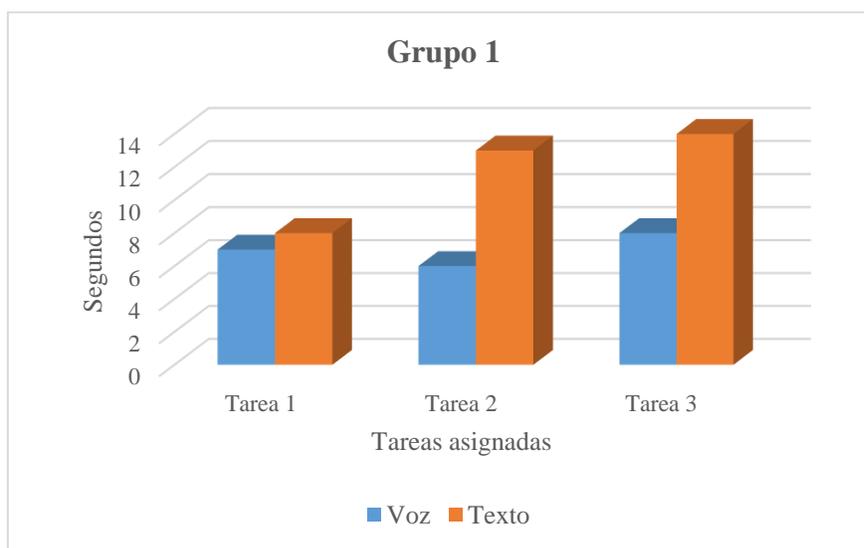
5.4 Resultados obtenidos

Los resultados obtenidos de la primera parte de la encuesta, se resume en la Tabla 36. En la citada tabla se describe el número de asistencias y número de errores suscitados al momento de ejecutar las pruebas. A demás se promedió el tiempo de culminación de cada tarea asignada, tanto en texto como en voz por parte de los dos grupos.

Tabla 36*Resultados obtenidos en ejecución de pruebas*

Grupo de Usuario	% Culminación de tareas asistidas	Número de errores	Número de asistencias	Tiempo de culminación de tarea 1 Promedio		Tiempo de culminación de tarea 2 Promedio		Tiempo de culminación de tarea 3 Promedio	
				Voz	Texto	Voz	Texto	Voz	Texto
				Grupo 1	100 %	4	4	7	8
Grupo 2	100 %	3	4	8	22	11	11	9	17

Las Figuras 29 y 30 muestran los datos recolectados dentro del grupo 1 y grupo 2. En los cuales se evidencia el tiempo promedio de ejecución de cada tarea asignada ya sea en voz o texto. Con los resultados obtenidos se determinó que los participantes se demoran más al enviar mensajes de texto que de voz; sin embargo, las tareas que se asignaron se pudieron realizar sin problemas.

**Figura 29.** Tareas realizadas por Grupo 1 (Voz y texto).

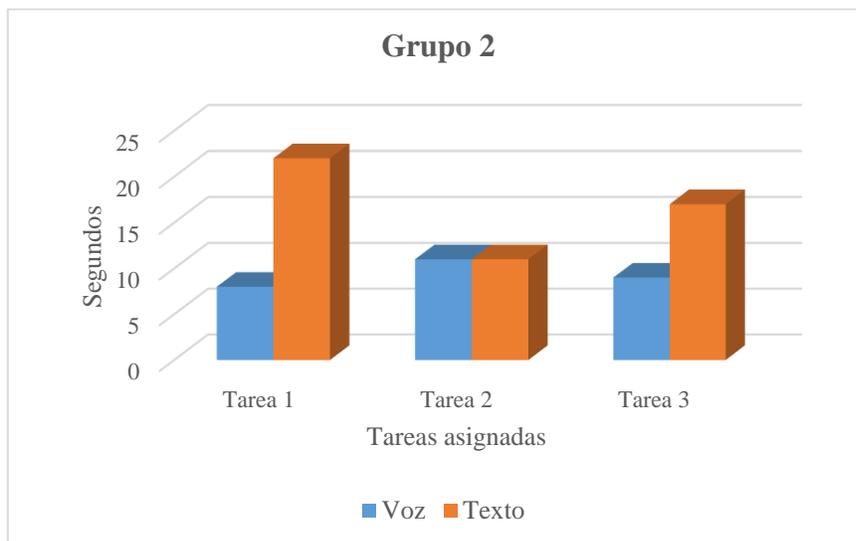


Figura 30. Tareas realizadas por Grupo 2 (Voz y texto)

En la segunda sección al analizar el grado de aceptación del usuario, se obtuvo los siguientes resultados: Para el grupo 1 se alcanzó el 71% (totalmente de acuerdo) y 29% (de acuerdo) en la facilidad de realizar las tareas, el tiempo empleado y mensajes de ayuda. Mientras que para el grupo 2 se obtuvo un 100%. En la Figura 31 se representa los porcentajes obtenidos.

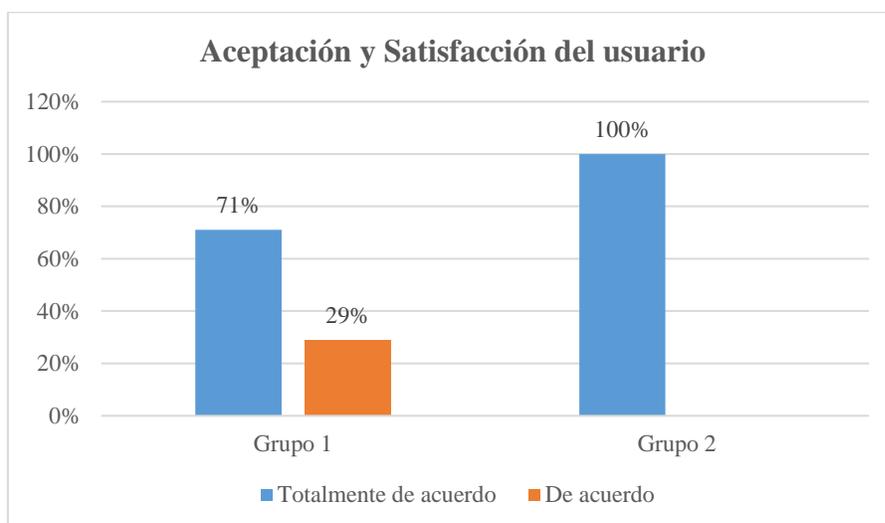


Figura 31. Porcentaje obtenido con respecto a la aceptación y satisfacción de uso

En la tercera parte de la encuesta se presentó un porcentaje de usabilidad del 80% para el grupo 1. Mientras que para el grupo 2 se alcanzó el 94%, lo que permite determinar que la usabilidad del sistema es buena, sin embargo, el aplicativo puede mejorarse en algunos aspectos como es el caso del envío automático del audio. La Figura 32 muestra los resultados obtenidos.

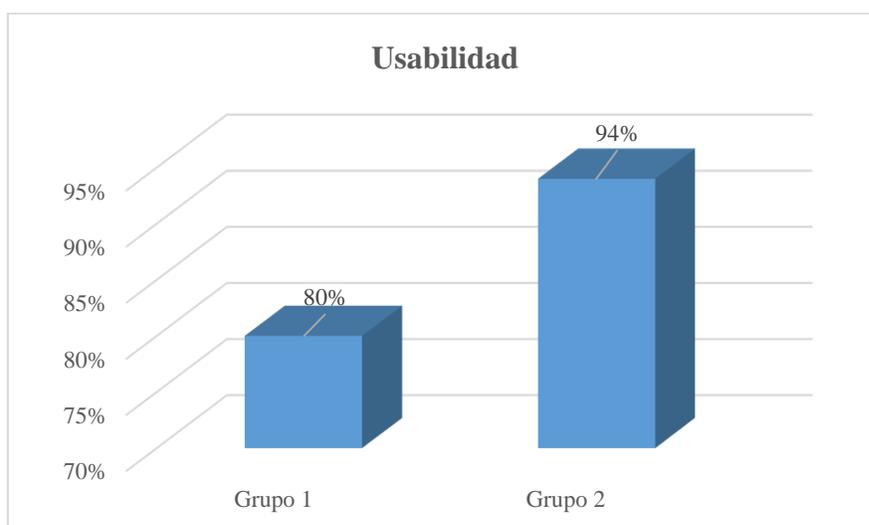


Figura 32. Porcentaje obtenido en análisis de usabilidad

Para concluir con el análisis de resultados, se procedió a validar si el aplicativo desarrollado puede ayudar a disminuir accidentes domésticos. Para esto en el cuestionario se realizó una pregunta en la cuales los usuarios potenciales debían responderían con un SI o NO, argumentando su respuesta. De la evaluación y análisis de las contestaciones se obtuvieron los siguientes resultados: el 80% piensa que el aplicativo podría ayudar a reducir accidentes domésticos. Las Figuras 33 resumen los resultados obtenidos.

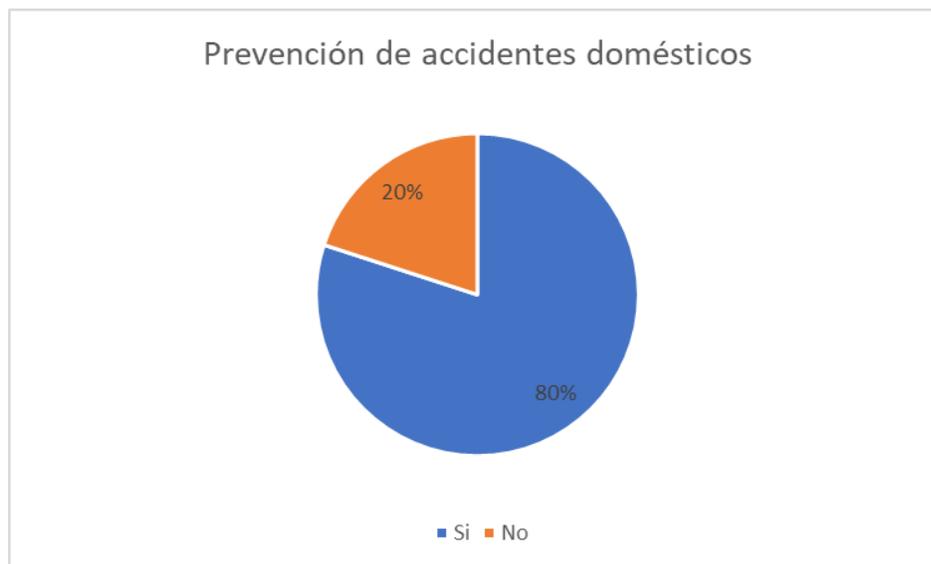


Figura 33. Resultados obtenidos en cuanto a la prevencción de accidentes domésticos

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

A continuación, se presentarán las conclusiones y recomendaciones obtenidas en el desarrollo del aplicativo.

6.1 Conclusiones

El estudio de literatura que se realizó, pudo reflejar varias soluciones en cuanto a accidentes domésticos y el control de artefactos eléctricos. Una de las soluciones más relevante para el estudio se centró en el área de IoT, manejo de hardware como Raspberry Pi y protocolos de comunicación inalámbrica como ZigBee y Wifi. Mientras que las demás se centraron en sensores, microcontroladores y dispositivos para controlar la electricidad. Estas opciones manejan sitios web o hardware para el control de los dispositivos eléctricos; sin embargo, la interacción con el usuario no suele ser muy amigable. Ninguno de los estudios primarios arrojó resultados en cuanto a la implementación de chatbots y la problemática planteada. La incorporación de chatbots a sistemas domóticos, permite una mayor interacción con el usuario, facilitando su uso por medio de mensajes basados en un lenguaje natural.

Para la definición de la semántica se incorporó un corpus de datos (base de conocimiento) en el idioma español. El corpus creado está basado en un modelo de interacción “Smart Home” preestablecido por la herramienta Dialogflow. El cual facilita la comunicación en un lenguaje natural, entre el usuario y los dispositivos electrónicos. Para lograr esta comunicación, la incorporación del procesamiento de lenguaje natural en los chatbots, permite que la máquina

interprete el lenguaje humano. Extrayendo información y significado del idioma tanto en texto como en voz, basándose en técnicas para extraer o identificar información relevante.

Se hizo el estudio comparativo entre las APIs: Dialogflow, Wit.ai, IFTTT y Alexa. Donde se comparó la usabilidad y características generales de cada plataforma al momento de desarrollar un chatbot. De igual forma se realizó pruebas de precisión en cuanto al reconocimiento de voz, y pruebas funcionales del modelo de interacción. La herramienta que se escogió fue Wit.ai ya que ofrece mejores prestaciones en cuanto al costo al ser open source, provee integración con otros sistemas mediante el uso de su API HTTP, facilita el aprendizaje del bot, soporta múltiples idiomas, responde a comandos escritos o hablados y proporciona el material necesario para el uso de la herramienta.

Para el desarrollo del aplicativo se adaptó un diseño responsive, el cual es multiplataforma y permite su uso tanto en web como en plataformas móviles. El aplicativo desarrollado se comunica con el motor de inferencia Wit.ai, por medio del protocolo HTTP. Este motor de inferencia permite el aprendizaje automático del bot y la identificación de entidades e intenciones, que son datos primordiales para la generación de respuesta.

Para establecer la conexión entre el aplicativo desarrollado y el control de los dispositivos, se utilizó una Raspberry Pi, para el funcionamiento de la plataforma OpenHAB. El cual permite integrar una gran cantidad de dispositivos dentro del hogar, facilitando la automatización del mismo a un bajo costo y sin depender de proveedores. La API REST que ofrece OpenHAB permitió consumir los servicios, donde se puede cambiar o verificar el estado de un dispositivo, a

través de reglas implementadas en el aplicativo desarrollado y por medio del protocolo de comunicación inalámbrica Wifi.

Para la validación de la propuesta, se realizaron pruebas de funcionalidad, grado de aceptación del aplicativo y el puntaje SUS obtenido en cuanto a la usabilidad. Dentro de las pruebas de funcionalidad, se pudo constatar que los participantes se demoran menos tiempo al momento de ejecutar las tareas por medio de voz. Cabe recalcar que las tareas fueron realizadas con éxito, no obstante, se tuvo varios problemas en el reconocimiento de voz, al pronunciar mal las palabras o cerrar muy rápido el micrófono. La mayoría de participantes concuerda que la solución propuesta puede ayudar en cierta forma a disminuir los accidentes domésticos verificando el estado de los dispositivos y actuando sobre ellos, pero se debería garantizar mayor seguridad en las conexiones.

6.2 Recomendaciones

Con la implementación de chatbots, el usuario podría interactuar de mejor manera con el sistema; puesto que, la comunicación se presenta de forma natural. Además los chatbots podrían ser incorporados en múltiples áreas como entretenimiento, áreas administrativas, entre otras, haciendo uso de varias tareas. En el caso de la aplicación propuesta, se podría añadir temporizadores al momento de encender o apagar determinados dispositivos.

Al momento de controlar los dispositivos electrónicos por medio de una red inalámbrica bajo el protocolo Wifi, surgieron varios problemas en cuanto al tiempo de ejecución de la orden al existir tráfico en la red por lo que se sugiere investigar sobre la implementación del protocolo

Zigbee en la plataforma OpenHAB. Con el fin de establecer una comunicación segura y controlar el acceso de seguridad de la red.

REFERENCIAS BIBLIOGRÁFICAS

Abdul-Kader, S. A., & Woods, J. C. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7).

Abdulrazak, B., Yared, R., Tessier, T., & Mabileau, P. (2015). Toward pervasive computing system to enhance safety of ageing people in smart kitchen. *In ICT4AgeingWell*, (págs. 17-28).

Abe, T., Fukagawa, K., Mizuno, Y., & Yoshida, A. (2016). Arc discharge detection caused by short-circuit in ac power supply cord. *In 2016 IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP)* (págs. 420-423). IEEE.

Alcaraz, M. (2014). Internet de las cosas. *Universidad Católica Nuestra Señora de la Asunción*, 2-3.

Álvarez, D. (Marzo de 2015). Evaluación de los impactos en el consumo de energía eléctrica asociados al uso de refrigeradores eficientes en ecuador. Cuenca, Azuay, Ecuador.

Amazon. (14 de Noviembre de 2019). *Alexa skill kit docs*. Obtenido de Alexa skill kit docs: <https://developer.amazon.com/es/docs/custom-skills/speech-synthesis-markup-language-ssml-reference.html>

- Amazon. (2019). *Amazon alexa*. Obtenido de Amazon alexa: <https://developer.amazon.com/es-ES/alexa/alexa-skills-kit>
- ARCONEL. (2018). *Estadística anual y multianual del sector eléctrico ecuatoriano*. Agencia de Regulación y Control de Electricidad.
- Baby, C. J., Khan, F. A., & Swathi, J. (Abril de 2017). Home automation using iot and a chatbot using natural language processing. *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, 1-6.
- Baby, C. J., Munshi, N., Malik, A., Dogra, K., & Rajesh, R. (Agosto de 2017). Home automation using web application and speech recognition. *In 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS)*, 1-6.
- Baskerville, R., & Pries-Heje, J. (1999). Grounded action research: a method for understanding it in practice. *Accounting, Management and Information Technologies*, 9(1), 1-23.
- Beltrán, M., Barreto, J., Salazar, H., & Villareal, C. (2019). Bibliobot. Bogota.
- Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. J. (1999). *El lenguaje unificado de modelado* (Vol. 1). Madrid: Addison Wesley.
- Cahn, J. (26 de Abril de 2017). CHATBOT: Architecture, design, and development. *University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science*.

- Calugay, A. A., Santos, G. D., Santos, J. I., Tolentino, R. T., & Zhuo, E. R. (2016). Web-based power board using raspberry pi. *In 2016 International Conference on Integrated Circuits and Microsystems (ICICM)* (págs. 50-54). IEEE.
- Carrero, D. (2017). Los 10 electrodomésticos que consumen más energía apagados. *Decoración 2.0*.
- Carvajal, L., Quesada, L., López, G., & Brenes, J. A. (Julio de 2017). Developing a proxy service to bring naturality to amazon's personal assistant "alexa". *In International Conference on Applied Human Factors and Ergonomics*, 260-270.
- Chaparro, P. (1999). Traumas y accidentes. *Revista de Salud Pública*, 1(3), 274-285.
- Chaumon, M., Cuvillier, B., Body, S., & Cros, F. (1 de Noviembre de 2016). Detecting falls at home user-centered desing of a pervasive technology. *Human Technology*(2), 12.
- Cobos Torres, J. C. (2013). *Integración de un chatbot como habilidad de un robot social con gestor de diálogos*. Madrid, España: Facultad de Ingeniería de Sistemas y Automática - Universidad Carlos III.
- CONELEC. (2011). Estadística del sector eléctrico ecuatoriano.
- Da Rocha Loureiro, R. J. (21 de Julio de 2017). Chatbot for domotics.
- Dale, R. (2016). The return of the chatbots. *Natural Language Engineering*, 22(5), 811-817.
- Del Campo, J., Quiñonez, E., García, E., & Magnífico, R. (Enero de 2009). Sistema doméstico para prevención de accidentes y supervisión en hogares. Bogotá.

Dhianeswar, R., Sumathi, S., & Joshitha, K. L. (2018). Automatic gas controller. En IEEE (Ed.), *In 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, (págs. 215-218).

Dones Téllez, I. (2019). *Integración de chatbot con smart speakers para asistente de compras*.

El Kaed, C., Ponnouradjane, A., & Shah, D. (Junio de 2018). A semantic based multi-platform iot integration approach from sensors to chatbots. *In 2018 Global Internet of Things Summit (GIoTS)*, 1-6.

Facebook. (14 de Noviembre de 2019). *Facebook for developers*. Obtenido de Facebook for developers: https://developers.facebook.com/docs/pages/realtime?locale=es_ES

Fiorini, L., Bonaccorsi, M., Betti, S. D., & Cavallo, F. (2016). User indoor localisation system enhances activity recognition: A proof of concept. *In Italian Forum of Ambient Assisted Living* (págs. pp. 251-268). Springer, Cham.

Flores Garate, A. L. (2018). Estudio de las técnicas que permitan el ahorro del consumo de energía eléctrica, tanto en forma doméstica como en forma institucional, para disminuir los gastos económicos que afectan negativamente los presupuestos integrales. Arequipa.

Følstad, A., & Brandtzæg, P. B. (2017). Chatbots and the new world of hci. *interactions*. 24(4), 38-42.

Freitas, D. J., Marcondes, T. B., Nakamura, L. H., Ueyama, J., Gomes, P. H., & Meneguette, R. I. (2015). Combining cell phones and wsns for preventing accidents in smart-homes with

disabled people. *In 2015 7th International Conference on New Technologies, Mobility and Security (NTMS)* (págs. pp. 1-5). IEEE.

Gomes, J. B., Rodrigues, J. J., Al-Muhtadi, J., Arunkumar, N., Rabêlo, R. A., & Furtado, V. (2018). An iot-based smart solution for preventing domestic co and lpg gas accidents. *In 2018 IEEE 10th Latin-American Conference on Communications (LATINCOM)* (págs. 1-6). IEEE.

Google. (28 de Octubre de 2019). *Google cloud*. Obtenido de Google cloud: <https://cloud.google.com/dialogflow/docs/?authuser=1>

Greyling, C. (14 de Agosto de 2019). *Medium*. Obtenido de Chatbots: from unstructured data to conversation: <https://medium.com/@CobusGreyling/chatbots-from-unstructured-data-to-conversation-4bef5b014c47>

Guzmán Villagómez, O. P. (2018). Desarrollo de un modelo para el análisis de vulnerabilidades de dispositivos inteligentes de reconocimiento de voz, caso de estudio amazon echo (alexa).

Haack, W., Severance, M., Wallace, M., & Wohlwend, J. (2017). Security analysis of the amazon echo. *Allen Institute for Artificial Intelligence*.

Hamdan, O., Shanableh, H., Zaki, I., Al-Ali, A. R., & T, S. (Enero de 2019). IoT-based interactive dual mode smart home automation. *In 2019 IEEE International Conference on Consumer Electronics (ICCE)*, 1-2.

- Herrera, L. (2005). Viviendas inteligentes (domótica). *Revista Ingeniería e Investigación*, 25(2), 47-53.
- IFTTT. (2019). *IFTTT*. Obtenido de IFTTT: <https://platform.ifttt.com/docs>
- Jamil, M. M., & Ahmad, M. S. (2015). A pilot study: Development of home automation system via raspberry Pi. In *2015 2nd International Conference on Biomedical Engineering (ICoBE)*, 1-4.
- Jivani, F. D., Malvankar, M., & Shankarmani, R. (2018). A Voice Controlled Smart Home Solution With a Centralized Management Framework Implemented Using AI and NLP. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)* (págs. pp. 1-5). IEEE.
- Junestrand, S., Passaret, X., & Vázquez, D. (2004). *Domótica y hogar digital*. Editorial Paraninfo.
- Kaiwen, C., Kumar, A., Xavier, N., & Panda, S. K. (2016). An intelligent home appliance control-based on WSN for smart buildings. In *2016 IEEE International Conference on Sustainable Energy Technologies (ICSET)*, 282-287.
- Kaveeya, G. S., Gomathi, S., Kavipriya, K., Selvi, A. K., & Sivakumar, S. (2017). Automated unified system for LPG using load sensor. In *2017 International Conference on Power and Embedded Drive Control (ICPEDC)* (págs. 459-462). IEEE.
- Khan, M. A., & Salah, K. (2018). IoT security review, blockchain solutions, and open challenges. *Future Generation Computer Systems*, 82, 395-411.

- Khanna, A., Das, B., Pandey, B., Hussain, D., & Jain, V. (2016). A discussion about upgrading the quick script platform to create natural language based iot systems. *Indian Journal of Science and Technology*, 9(46).
- Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Pradeepkumar, B., & Das, T. (2015). A study of today's ai through chatbots and rediscovery of machine intelligence. *Int. J. ue Serv. Sci. Technol*, 8(7), 277-284.
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature. *Software Engineering Group*.
- Lee, I., & Lee, K. (2015). The internet of things applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431-440.
- Levac, D., & Colquhoun, H. (2010). Scoping studies advancing the methodology. *Implementation science*.
- Li, R. Y., Li, H., Mak, C., & Tang, T. (2016). Sustainable smart home and home automation big data analytics approach. *International Journal of Smart Home*, 10(8), 177-187.
- López, G., Quesada, L., & Guerrero, L. A. (Julio de 2017). Alexa vs. siri vs. cortana vs. google assistant: a comparison of speech-based natural user interfaces. *In International Conference on Applied Human Factors and Ergonomics*, 241-250.
- Lucci, G., & Paternò, F. (Mayo de 2015). Analysing how users prefer to model contextual event-action behaviours in their smartphones. *In International Symposium on End User Development*, 186-191.

- Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of things a literature review. *Journal of Computer and Communications*, 3(5), 164.
- Mallap, M. P., Rao, G. J., Sahu, B. P., Kumar, J. S., & Krishna, S. (2018). A Novel Approach for Home Automation. *International Journal of Scientific Research in Science*, 962-966.
- Mamgain, D. (10 de Mayo de 2019). *Kommunicate*. Obtenido de *Kommunicate*: <https://www.kommunicate.io/blog/dialogflow-vs-lex-vs-watson-vs-wit-vs-azure-bot/>
- Metro. (27 de Marzo de 2018). Estos son los artefactos que consumen más energía cuando están apagados. *Metro Ecuador*.
- Minda Almagor, A. C. (2013). *Factores influyentes asociados en la aparición de accidentes domésticos en niñas y niños menores de 5 años, atendidos en el servicio de emergencia del hospital san vicente de paúl, de la ciudad de ibarra en el período noviembre 2010 a julio del 2011*. Ibarra.
- Morales, G. (2011). La domótica como herramienta para un mejor confort, seguridad y ahorro energético. *Ciencia e Ingeniería*, 32(1), 39-42.
- Nereida Pacios, A., & Salazar, H. (1998). Factores de riesgo relacionados con los accidentes domésticos. *Revista Cubana de Medicina General e Integral*, 14(5), 440-444.
- Nereida Pacios, A., & Salazar, H. (1999). Accidentes en el hogar. *Revista Cubana de Medicina General Integral*, 15(2), 123-127.

- Nurfaif, M. B., Sulistiyanti, S. R., Komarudin, M., & Nama, G. F. (2017). Telemetry and tele-control of electronic appliances for smart-home-system. *In 2017 International Symposium on Electronics and Smart Devices (ISESD)*, 18-23.
- Orejuela, V., Arias, D., & Aguila, A. (2015). Response of residential electricity demand against price signals in ecuador. *Proc. 2015 IEEE Thirty Fifth Cent. Am. Panama Conv*, 373-378.
- Parthornratt, T., Kitsawat, D., Putthapipat, P., & Koronjaruwat, P. (Julio de 2018). A Smart Home Automation Via Facebook Chatbot and Raspberry Pi. *In 2018 2nd International Conference on Engineering Innovation (ICEI)*, 52-56.
- Pavithra, D., & Balakrishnan, R. (Abril de 2015). Iot based monitoring and control system for home automation. *2015 Global Conference on Communication Technologies (GCCT)*, 169-173.
- Peng, C., & Chen, R. C. (Marzo de 2018). Voice recognition by Google Home and Raspberry Pi for smart socket control. *In 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)* (págs. 324-329). IEEE.
- Pérez Vallejo, P. J. (2019). Implementación de un agente conversacional para negocio de repuestos automotrices integrado a plataformas de mensajería instantánea (Bachelor's thesis, PUCE-Quito).
- Pressman, R. (2002). *Ingeniería del software un enfoque práctico*. México: The McGraw-Hill.
- Quishpe Gaibor, J., & Quishpe Freire, J. (Septiembre de 2018). Deontología aplicada al consumo de electricidad en hogares del Ecuador. *Revista Caribeña de Ciencias Sociales*, 1-8.

- Rahman, A. M., Al Mamun, A., & Islam, A. (Diciembre de 2017). Programming challenges of chatbot current and future prospective. *In 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 75-78.
- Ramos, F., & Velez, J. I. (2016). Integración de técnicas de procesamiento de lenguaje natural a través de servicios web.
- Reddy, P. S., Reddy, K. T., Reddy, P. A., Ramaiah, G. K., & Kishor, S. N. (Octubre de 2016). An IoT based home automation using android application. *In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs)*, 285-290.
- Robalino, M. (2017). *Desarrollo de un prototipo de "bot conversacional" empleando procesamiento de lenguaje natural*. (Bachelor's thesis, Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Electrónica y Telecomunicaciones).
- Rojas, J. P. (2017). *Desarrollo de un prototipo funcional para la aplicación móvil q-bus para la plataforma ios que brinde información de las rutas de transporte público en la ciudad de quito utilizando bluetooth low energy, códigos qr y geo posicionamiento*. (Bachelor's thesis, CIENCIAS DE LA INGENIERÍA E INDUSTRIAS FACULTAD: INGENIERÍA INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN).
- Salamanca, U., Eugenia, C., & Macq, B. (2019). Study and design of a chatbot as support of foreign languages learning within an e-learning platform.
- Sandoval, V. V. (2019). Reconocimiento de emociones por medio de voz.

- Sara Rodríguez, G., Noelia, H. D., Marta, P., S., M. M., Z., M. P., & Irene, C. P. (s.f.). Investigación acción. *Investigación Acción*. Madrid. Obtenido de https://mestrado.prpg.ufg.br/up/97/o/IA._Madrid.pdf
- Saygin, A. P., Cicekli, I., & Akman, V. (2000). *Turing test 50 years later* (Vol. 10). Minds and machines.
- Serna, A., Acevedo, E., & Serna, E. (2017). Principios de la inteligencia artificial en las ciencias computacionales. *Desarrollo e Innovación en Ingeniería*, 161.
- Sivaranjani, S., Gowdhami, D., & Karthikkannan, P. (2016). An appraisal on gas leakage detection and controlling system in smart home using IoT. En *IIOAB Journal* (Vol. 7, págs. 608-615).
- Solans, A. C. (2009). *Hacia la era de la inteligencia*. (Vol. 81). Fundación Telefónica Patronato de Fundación Telefónica.
- Stergiou, C., Psannis, K. E., Kim, B. G., & Gupta, B. (2018). Secure integration of iot and cloud computing. *Future Generation Computer Systems*, 78, 964-975.
- Susman, G., & Evered, R. D. (1978). An assessment of the scientific merits of action research. *Administrative Science Quarterly*, 582-603.
- Tarifasdeluz. (2019). *Tarifas de luz*. Obtenido de <https://www.tarifasdeluz.com/faqs/electrodomesticos-que-consumen-mas-energia/>

- Teng-Fa, T., & Cheng-Chien, K. (2013). A smart monitoring and control system for the household electric power usage. *In 2013 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)* (págs. 1-4). IEEE.
- Terada, T., Watanabe, R., & Tsukamoto, M. (2013). A user recognition method using accelerometer for electric appliances. *In 2013 16th International Conference on Network-Based Information Systems* (págs. 350-355). IEEE.
- Toledo, A. (2018). Desarrollo de un chatbot que ayude a responder a preguntas frecuentes referentes a becas en la universidad técnica particular de loja. (Bachelor's thesis).
- Torres, E. (2018). Academic-bot diseño, definición y documentación de un bot conversacional de uso académico. Madrid.
- Torruella, J., & Llisterri, J. (1999). Diseño de corpus textuales y orales. *Filología e informática. Nuevas tecnologías en los estudios filológicos* , 45-77.
- Tseng, C. L., Cheng, C. S., & Hsu, Y. H. (Octubre de 2018). An IoT-Based Home Automation System Using Wi-Fi Wireless Sensor Networks. *In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2430-2435.
- Ulloa Airzaga, E. S. (2015). *Eficiencia Del Consumo Eléctrico en el Sector Residencial Urbano de Cuenca*. Cuenca.
- Umer, S., Kaneko, M., Tan, Y., & Lim, A. O. (Febrero de 2014). Priority based maximum consuming power control in smart homes. *In ISGT 2014* , 1-5.

- Vásquez, A. C., Quispe, J. P., & Huayna, A. M. (2009). Procesamiento de lenguaje natural. *Revista de investigación de Sistemas e Informática*, 6(2), 45-54.
- Verduga, J., & Zambrano, W. (2017). Análisis técnico sobre los efectos generados por equipos de inducción en el consumo eléctrico residencial del edificio holsol.
- Wit.ai. (15 de Octubre de 2019). *Wit* . Obtenido de Wit : <https://wit.ai/docs/quickstart>
- Wohlin, C., Runeson, P., Neto, P., Engström, E., do Carmo Machado, I., & De Almeida, E. S. (2013). On the reliability of mapping studies in software engineering. *The Journal of Systems and Software.*, 86(10), 2594-2610.
- Yang, X., Chen, Y. N., Hakkani-Tür, D., Crook, P., Li, X., Gao, J., & Deng, L. (Marzo de 2017). End-to-end joint learning of natural language understanding and dialogue manager. *In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5690-5694.
- Yerragolla, M., Pallela, K., & Gera, I. P. (2016). Intelligent security system for residential and industrial automation. *In 2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)* (págs. 229-234). IEEE.
- Zhou, Y., Chen, Y., Xu, G., & Zheng, C. (Mayo de 2014). Home energy management from demand side in smart grid. *In 2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, 180-185.
- Zuñiga, Z., & Humberto, O. (2018). *Implementación de un chatbot con botframework caso de estudio, servicios a clientes del área de fianzas de seguros equinoccial*. Quito.

