



Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

DESARROLLO DE UN CHATBOT WEB COMO ASISTENTE DE VENTAS

Trabajo Fin de Máster

Presentado por: Machado Redrobán, Luis Felipe

Director: Dr. Salvador Cobos Guzmán

Ciudad: Quito

Fecha: 20-02-2019

Resumen

En este proyecto se desarrolla un chatbot para una empresa en Ecuador. Dicho chatbot tendrá como objetivo atender a los visitantes del sitio web, cuando estos establezcan una comunicación a través de un chat en línea. Se utiliza metodología SCRUM, para el desarrollo del software. El chatbot será específicamente, un asistente de ventas, que proporcionará información a los potenciales clientes y tratará de tomar los datos de los mismos, como nombre, correo electrónico y/o teléfono.

El chatbot se desarrolla, utilizando librerías y frameworks libres y gratuitos que permiten procesamiento de lenguaje natural y estructurar contextos de conversaciones. Para el entrenamiento se redacta un corpus de acuerdo a la visión, misión, productos, servicios y objetivos de la empresa. El chatbot es entrenado y es integrado al chat en línea del sitio web de la empresa. Finalmente se compara con otros chatbots de otras páginas web de empresas en el Ecuador.

Palabras clave: chatbot, inteligencia artificial, procesamiento de lenguaje natural, sitio web, software.

Abstract

In this project a chatbot is developed for a company in Ecuador. This chatbot will aim to serve the visitors of the website, when they establish a communication through an online chat. SCRUM methodology is used for software development. The chatbot will specifically be a sales assistant, who will provide information to potential customers and try to take their data, such as name, email and / or telephone.

The chatbot is developed, using free and free libraries and frameworks that allow natural language processing and structure conversation contexts. For training, a corpus is written according to the vision, mission, products, services and objectives of the company. The chatbot is trained and integrated into the online chat of the company's website. Finally, it is compared with other chatbots from other websites of companies in Ecuador.

Keywords: artificial intelligence, chatbot, natural language processing, software, website.

INDICE

1 Contenido

1	Introducción.....	10
1.1	Justificación.....	12
1.2	Planteamiento del trabajo	12
1.3	Estructura de la memoria.....	13
2	Estado del arte.....	15
2.1	Chatbot.....	15
2.2	Procesamiento del lenguaje natural (PLN).....	15
2.3	Enfoque tradicional del procesamiento del lenguaje natural (PLN)	17
2.3.1	Desventajas del método tradicional.....	18
2.3.2	Enfoque de <i>Deep learning</i> del Procesamiento del Lenguaje Natural (PLN).....	18
2.4	<i>Deep learning</i> en Chatbots	18
2.4.1	Historia del <i>deep learning</i>	18
2.4.2	Redes Neuronales para el procesamiento de lenguaje natural	21
2.5	Funciones del chatbot.....	22
2.6	Historia de los chatbots	23
2.6.1	Chatbots históricos.....	23
2.6.2	Aplicaciones de un chatbot	23
2.6.3	Tipos de Chatbots	24
2.6.4	Ventajas del uso de chatbots	24
2.6.5	Limitaciones	25
2.6.6	Chatbots actuales.....	25
2.7	<i>Frameworks</i> para el desarrollo de Chatbots	26
2.7.1	DIALOGFLOW	29
2.8	Conclusiones del estado del arte.....	31
2.9	Contribución del TFM.....	32
3	Objetivos y metodología.....	34
3.1	Objetivo General.....	34
3.2	Objetivos Específicos.....	34
3.3	Metodología de trabajo.....	34

3.3.1	Organización	35
3.3.2	Procesos de SCRUM.....	36
4	DISEÑO DE UN CHATBOT PARA ASISTENTE DE VENTAS.....	37
4.1	Proceso de Inicio	37
4.1.1	Visión del producto.....	37
4.1.2	Roles	42
4.1.3	<i>Backlog</i> de proyecto.....	42
4.2	Proceso de planificación y estimación	44
4.2.1	Historias de usuarios	44
4.2.2	Lista de tareas.....	44
	Creación del proyecto.....	44
	Entrenamiento de corpus de servicio	44
	Entrenamiento de corpus de técnico.....	45
	Contexto de la conversación	45
	Errores de la conversación	45
	Detección de visitantes no deseados.....	45
	Corpus Avanzado	45
	Integración.....	45
4.3	Proceso de creación del chatbot de ventas	46
4.3.1	Creación del proyecto.....	46
4.3.2	Desarrollo de corpus.....	46
4.3.3	Entrenamiento con DIALOGFLOW.....	56
4.3.4	Entrenamiento con KERAS.....	57
4.3.5	Corpus complementario	58
4.3.6	Contextos.....	61
4.3.7	Integraciones.....	68
4.4	Proceso de Pruebas y Lanzamiento	69
5	EVALUACIÓN DEL CHATBOT	75
5.1	Chatbots en páginas web.....	75
5.1.1	Chatbot empresa Claro Ecuador	75
5.1.2	Chatbot empresa Salud SA	77
5.1.3	Chatbot empresa Banco Pacífico	79
5.1.4	Chatbot empresa Banco de Loja	81
5.2	Chatbots en páginas web vs ContactBot.....	83

6	CONCLUSIONES Y RECOMENDACIONES.....	85
6.1	CONCLUSIONES.....	85
6.2	RECOMENDACIONES	87
7	Bibliografía.....	88
	Entrenamiento de corpus técnico	95
	Visitantes no deseados	106

TABLA DE FIGURAS

Figura 1 Taxonomía de las aplicaciones de PLN (Ganegedara., May 2018).....	17
Figura 2 Enfoque general de un PLN clásico (Ganegedara., May 2018).....	17
Figura 3 Representación de McCulloch y Pitts (Wallis)	19
Figura 4 Perceptrón (Wallis).....	20
Figura 5 Celda básica LSTM (Bhagwat, 2018).....	21
Figura 6 Modelo Sec2Sec (Bhagwat, 2018)	22
Figura 7 Creación de un Agente (Google)	29
Figura 8 INTENT (Google).....	30
Figura 9 Contextos (Google)	31
Figura 10 Organización en SCRUM (SCRUMstudy, 2013).....	35
Figura 11 Web Demo Dialogflow (Google).....	40
Figura 12 Demo de conversación (Google)	41
Figura 13 Creación de nuevo Agente (Google).....	46
Figura 14 INTENTS en DIALOGFLOW (Google).....	57
Figura 15 Entrenamiento Keras I (Google)	58
Figura 16 Entrenamiento KERAS II (Google)	59
Figura 17 Entrenamiento KERAS III (Google)	59
Figura 18 Entrenamiento KERAS IV (Google).....	60
Figura 19 Pruebas del modelo entrenado con KERAS (Google)	60
Figura 20 Entrenamiento KERAS V (Google).....	61
Figura 21 Contexto inicial.....	62
Figura 22 Contextos Contactvox-UCS (Google)	63
Figura 23 Contextos CONTACTVOX UCS (Google).....	63
Figura 24 Ejemplo de conversación bajo contexto	64
Figura 25 Entrenamiento de obtención de datos del visitante (Google).....	66
Figura 26 Entrenamiento de datos del visitante NOMBRE (Google)	67
Figura 27 Entrenamiento de datos del visitante E-MAIL (Google)	67
Figura 28 Integración chatbot.....	68
Figura 29 Prueba de Chatbot I (Google)	69
Figura 30 Prueba de Chatbot II (Google)	69
Figura 31 Prueba de Chatbot III (Google)	70
Figura 32 Prueba de Chatbot IV (Google).....	70
Figura 33 Prueba de Chatbot V (Google).....	71
Figura 34 Prueba de Chatbot VI (Google).....	71
Figura 35 Prueba de Chatbot VII (Google).....	71
Figura 36 Prueba de Chatbot VIII (Google)	72
Figura 37 Prueba de Chatbot IX (Google).....	72
Figura 38 Prueba de Chatbot X (Google).....	73
Figura 39 Prueba de Chatbot XI (Google).....	73

Figura 40 Prueba de Chatbot XII (Google).....	74
Figura 41 Prueba de Chatbot XIII (Google)	74
Figura 42 Asistente virtual I (CLARO).....	76
Figura 43 Asistente virtual VI (Saludsa)	78
Figura 44 Asistente virtual XV (Banco del Pacifico).....	80
Figura 45 Asistente virtual XXIII (FACEBOOK).....	82
Figura 46 Entrenamiento de saludos (Google).....	90
Figura 47 Respuestas a saludos (Google).....	90
Figura 48 Prueba de respuesta I (Google).....	91
Figura 49 Prueba de respuesta II (Google).....	91
Figura 50 Entrenamiento de despedidas (Google)	92
Figura 51 Prueba de despedidas I (Google).....	93
Figura 52 Prueba de despedidas II (Google).....	93
Figura 53 Entrenamiento de datos de la empresa (Google)	94
Figura 54 Prueba de datos de la empresa I (Google)	95
Figura 55 Entrenamiento de producto CONTACTVOX UCS (Google).....	96
Figura 56 Prueba de producto CONTACTVOX UCS (Google).....	96
Figura 57 Entrenamiento de producto CONTACTVOX CC (Google)	97
Figura 58 Prueba de producto CONTACTVOX CC.....	98
Figura 59 Entrenamiento producto CONTACTVOX CRM (Google)	99
Figura 60 Pruebas de producto CONTACTVOX CRM (Google).....	99
Figura 61 Entrenamiento de producto CONTACTVOX BI (Google)	100
Figura 62 Prueba de producto CONTACTVOX BI (Google)	101
Figura 63 Entrenamiento de producto CONTACTVOX VIDEO (Google).....	101
Figura 64 Prueba de producto CONTACTVOX VIDEO (Google).....	102
Figura 65 Entrenamiento de producto CONTACTVOX CALLMATIC (Google).....	103
Figura 66 Prueba de producto CONTACTVOX CALLMATIC	103
Figura 67 Entrenamiento de producto CONTACTVOX CONTACTWEB (Google).....	104
Figura 68 Pruebas de producto CONTACTVOX CONTACTWEB (Google).....	105
Figura 69 Entrenamiento de Errores (Google)	105
Figura 70 Entrenamiento visitantes no deseados (Google)	106
Figura 71 Prueba de obtención de datos de visitante I (Google)	106
Figura 72 Prueba de obtención de datos de visitante II (Google)	107
Figura 73 Prueba de obtención de datos de visitante III (Google).....	107
Figura 74 Asistente virtual V (CLARO)	113
Figura 75 Asistente virtual (CLARO)	113
Figura 76 Asistente virtual II (CLARO).....	114
Figura 77 Asistente virtual III (CLARO).....	114
Figura 78 Asistente virtual IV (CLARO)	114
Figura 79 Asistente virtual IV (CLARO)	115
Figura 80 Asistente virtual VII (Saludsa)	115
Figura 81 Asistente virtual VIII (Saludsa)	116
Figura 82 Asistente virtual IX (Saludsa)	116
Figura 83 Asistente virtual X (Saludsa).....	116
Figura 84 Asistente virtual XI (Saludsa)	117

Figura 85 Asistente virtual XIV (Saludsa).....	117
Figura 86 Asistente virtual XIII (Saludsa)	117
Figura 87 Asistente virtual XVI (Banco del Pacifico)	118
Figura 88 Asistente virtual XVII (Banco del Pacifico)	119
Figura 89 Asistente virtual XVIII (Banco del Pacifico)	119
Figura 90 Asistente virtual XIX (Banco del Pacifico)	120
Figura 91 Asistente virtual XX (Banco del Pacifico).....	120
Figura 92 Asistente virtual XXI (Banco del Pacifico)	121
Figura 93 Asistente virtual XXII (Banco del Pacifico)	121
Figura 94 Asistente virtual XXIV (FACEBOOK)	122
Figura 95 Asistente virtual XXV (FACEBOOK).....	122
Figura 96 Asistente virtual XXVI (FACEBOOK)	123
Figura 97 Asistente virtual XXVII (FACEBOOK)	123
Figura 98 Asistente virtual XXVIII (FACEBOOK)	123
Figura 99 Asistente virtual XXIX (FACEBOOK)	124
Figura 100 Asistente virtual XXX (FACEBOOK)	124

1 Introducción

De acuerdo al diccionario, la comunicación es la “Transmisión de señales mediante un código común entre el emisor y el receptor.” Los seres humanos utilizamos un complejo sistema de códigos al que le denominamos lenguaje.

El lenguaje nos permite comunicarnos, describir el mundo que nos rodea y transmitirlo a otros seres humanos. El lenguaje está íntimamente relacionado con el pensamiento, por lo que a través del lenguaje podemos comunicar nuestras ideas, creencias, deseos, sentimientos, etc.

En la actualidad la comunicación, no solo se establece entre seres humanos, es decir el emisor y receptor ya no son solamente seres humanos, ahora pueden ser humanos y máquinas. Desde el diseño y construcción de computadoras el ser humano ha desarrollado varios lenguajes para poder comunicarse con las máquinas y poder programar su funcionamiento. Estos lenguajes existen en varios niveles, los de nivel más bajo, podemos indicar que son señales y pulsos eléctricos, representados con sistema binario de 0 y 1, hasta llegar a lenguaje de alto nivel, mucho más intuitivo para el ser humano y compuesto en parte por palabras propias de un lenguaje humano. Adicionalmente, se han desarrollado otros tipos de lenguaje para que las máquinas puedan comunicarse entre sí. Estos lenguajes denominados protocolos permite el intercambio de información entre computadoras.

Sin embargo el lenguaje de comunicación entre los seres humanos y las máquinas no es el lenguaje natural del ser humano, ya que la maquina no entiende nuestra forma de expresarnos naturalmente, como lo hacemos con otros seres humanos. Es así que a partir de la época de 1950, se empieza un estudio, a través de una serie de investigaciones para lograr el objetivo de que las computadoras puedan entender el lenguaje natural de los seres humanos.

La posibilidad de que las máquinas entiendan el lenguaje natural de los seres humanos permite satisfacer muchas necesidades identificadas, como por ejemplo el procesamiento de datos y la clasificación de información. Según un estudio realizado por IBM, 2.5 EXABYTES de datos fueron generados cada día en 2017 (Ganegedara., May 2018) y en la actualidad esa cifra ha ido en aumento. Por supuesto no toda los datos tiene el interés de un ser humano, aún así si se reduce los datos a los temas de interés, sería imposible para un ser humano procesar toda la cantidad de datos que se genera, leer todos los documentos, obtener resúmenes, analizar los datos para convertirlos en información, clasificar la información más relevante o de interés, etc.

Adicionalmente, el poder interactuar con las máquinas de manera natural y rápida, para que éstas, sirvan de ayuda para resolver problemas, brinden servicios o atiendan requerimientos del ser humano.

Por estos motivos se ha fortalecido el desarrollo tecnologías y aplicaciones que permitan a las máquinas entender de mejor manera el lenguaje de los seres humanos, ya sea que dicho lenguaje esté contenido en un texto de manera escrita, en una grabación de audio o que se esté llevando a cabo una comunicación en tiempo real, en la cual una máquina asuma el rol como emisor y/o receptor. En este último caso se han creado aplicaciones denominadas “agentes conversacionales” o como se los conoce coloquialmente: “chatbots”.

Un chatbot es un agente inteligente que puede comunicarse con los seres humanos a través del lenguaje natural del ser humano. Precisamente de esto trata este proyecto, del estudio de chatbots y el diseño y construcción de un chatbot capaz de atender y brindar un servicio al cliente en un sitio web de una empresa, asumiendo el rol de un asistente de ventas. Este trabajo tiene como objetivo principal, desarrollar un chatbot, para que el visitante de una página web de una empresa, pueda ser atendido de manera inmediata por un agente inteligente mediante comunicación escrita.

Con el auge de la Inteligencia Artificial, los humanos desean comunicarse con las maquinas utilizando el lenguaje natural propio de los seres humanos. En consecuencia se han desarrollado tecnologías capaces de hacer que las máquinas aprendan y procesen el lenguaje natural de los seres humanos. Sin embargo este trabajo apenas inicia, ya que las expresiones que se utilizan en conversaciones entre seres humanos son muy complejas.

En este trabajo se analiza el estado del arte de los chatbots y se describen algunas librerías, para el procesamiento de lenguaje natural y *frameworks* capaces de permitir la creación de chatbots.

Sin embargo, como veremos a través de este trabajo, existen formas de expresión muy complejas de los seres humanos, que hacen muy difícil que el chatbot pueda responder con coherencia y dificulta el trabajo de entrenamiento. Por tal motivo se pretende entrenar el chatbot, tomando en consideración las diferentes formas de expresión de un ser humano, para buscar incrementar la calidad de las respuestas que el chatbot emita dentro de una conversación y para esto se desarrollará un corpus de frases.

Este trabajo está enfocado en el diseño un chatbot como asistente de ventas, utilizando *frameworks* y librerías existentes y enfocándose estrictamente en esa área profesional, limitando así, la capacidad de respuestas dentro de dicha área.

1.1 Justificación

Las pequeñas y medianas empresas en Ecuador cuentan con un sitio web en el cual se encuentra información de las mismas. Los visitantes a menudo buscan información de la empresa en el sitio, pero muchas veces es necesario que dichas personas se comuniquen con los miembros de la empresa, para realizar consultas o solicitar más información de los servicios y productos que ofrece, sin embargo al ser una pequeña empresa, el personal de ventas no siempre está disponible. La empresa tiene la necesidad de brindar atención inmediata a los visitantes de su sitio web, con el objetivo de potencializar sus oportunidades de ventas. Por dicho motivo es necesario la incorporación de un sistema de inteligencia artificial (chatbots), que guíe al visitante y que pueda atender o responder las preguntas que tengan referencia con los servicios o productos que la empresa comercializa. Aunque la necesidad de atención inmediata, es parte de todas las empresas, las más grandes si cuentan con chats en línea con agentes humanos, para servicio al cliente y de hecho algunas de ellas ya están implementando chatbots. Sin embargo dichos chats no tiene gran capacidad de respuesta, sino se evidencia algunos chats como si se tratase de un IVR (*Interactive Voice Response*), en el cual se solicita la elección de opciones en lugar de tratar de establecer una comunicación más abierta.

El proyecto busca que los visitantes del sitio puedan establecer una conversación escrita con un agente inteligente dentro de un ámbito específico. Para esto se analizará el estado del arte del procesamiento de lenguaje natural, algunas herramientas existentes para la construcción de chatbots y las técnicas utilizadas hoy en día con la ayuda de *deep learning*. Para probar el funcionamiento del proyecto el chatbot se desplegará en el sitio web: <https://www.contactvox.com>.

Adicionalmente en este proyecto, se pone a prueba el chatbot desarrollado versus algunos chatbots, instalados en algunas páginas web de empresas ecuatorianas, con el objetivo de comparar y evaluar el comportamiento del chatbot. Se analiza el tipo de respuestas, si dicha respuesta es un conjunto de opciones o el chatbot permite establecer una comunicación más abierta, si entiende frases más elaboradas, frases con faltas ortográficas o mal escritas, entre otros criterios.

Finalmente en este proyecto no se tratará de encontrar respuestas al por qué las empresas no tienen implementados chatbots o las que si tienen que tipo de chatbot han implementado y cuanto les ha costado la inversión.

1.2 Planteamiento del trabajo

El presente trabajo propone el diseño y programación de un chatbot como asistente de ventas. En el estado del arte se analiza las diferentes tecnologías existentes para la

programación de chatbots, en busca de encontrar la mejor y con la cual poder cumplir los objetivos del presente trabajo.

Se parte del conocimiento general, de la evolución de los sistemas de inteligencia artificial enfocado en el tema de chatbots y se describen algunas herramientas como librerías y *frameworks* para la construcción de chatbots.

El trabajo consta de:

- Hacer un estudio del estado del arte de los chatbots.
- Hacer un breve estudio de las tecnologías de inteligencia artificial.
- Describir distintas herramientas o *frameworks* para la construcción de chatbots.
- Entrenar el chatbot utilizando DIALOGFLOW (Google) y KERAS (KERAS).
- Validar con frases simples.
- Validar con frases más complejas.
- Desarrollar un flujo de conversación.
- Evaluar los resultados.
- Comparar con chatbots existentes en páginas web de empresas ecuatorianas.
- Obtener conclusiones.

1.3 Estructura de la memoria

- En el capítulo 1 se realiza una introducción al tema propuesto. Se presenta la justificación del trabajo y el planteamiento del mismo.
- En el capítulo 2 se analiza el estado del arte de los chatbots. Los chatbots son una aplicación de los sistemas de procesamiento del lenguaje natural. Se analizan diferentes tecnologías que se utilizan en el procesamiento del lenguaje natural, principalmente el *deep learning*. También se presentan algunas librerías que implementan algoritmos de *deep learning* y que son utilizadas en los actuales momentos para construir aplicaciones de procesamiento del lenguaje natural. Se describen brevemente algunos chatbots existentes, así como *frameworks* que brindan entornos de desarrollo para crear chatbots.
- En el capítulo 3 se realiza el planteamiento de los objetivos del proyecto, tanto generales como específicos. Adicionalmente se describe brevemente la metodología utilizada en el análisis de requerimientos y desarrollo del chatbot.
- En el capítulo 4 se desarrolla un chatbot, orientado a brindar servicio de ventas dentro de un sitio web. Se desarrolla un corpus y se entrena el chatbot, tomando en consideración en el entrenamiento las frases de las conversaciones que pudiese

establecer con los visitantes del sitio web. Se describe el entrenamiento utilizando la librería DIALOGFLOW.

El chatbot se pone a prueba, para validar la capacidad de respuesta tiene frente a un visitante humano. Se evalúan distintos tipos de conversaciones o expresiones, desde frases bien escritas, hasta frases incompletas y con faltas ortográficas. Adicionalmente se entrena el corpus utilizando una librería que utiliza redes neuronales, utilizando el *framework* KERAS.

- En el capítulo 5 se evalúa el funcionamiento y capacidad de comprensión y respuesta del chatbot desarrollados y se realiza un comparativo con otros chatbots existentes en páginas web ecuatorianas.
- Finalmente se enlistan las conclusiones del trabajo realizado y se adjuntan anexos de código.

2 Estado del arte

2.1 Chatbot

La palabra chatbot se compone de dos palabras: chat y bot. Según el diccionario **chat**, es “El intercambio de mensajes en tiempo real, entre uno o más usuarios, en una red de computadoras” (DICTIONARIES) y **bot**, es una abreviatura de “robot”, que puede actuar como un programa autónomo en la red de computadoras e interactuar con otros usuarios. (Vogel, 2017). Un chatbot es un sistema de dialogo en línea entre un humano y un computador utilizando el lenguaje natural, la comunicación puede realizarse a través de lenguaje oral y escrito (Jía, 2003). Los chatbots o agentes conversacionales son una aplicación del procesamiento del lenguaje natural.

2.2 Procesamiento del lenguaje natural (PLN)

El Procesamiento de Lenguaje Natural es un área de estudio de la informática, que incluye a la inteligencia artificial y lingüística, cuyo objetivo es que las máquinas entiendan el lenguaje como lo hace otro ser humano y así se pueda establecer una comunicación entre los humanos y máquinas mediante el uso del lenguaje natural. El lenguaje natural es el idioma o lengua hablada o escrita utilizada por los seres humanos para la comunicación (Mateos Martín F. J., 2013).

De acuerdo a (Ganegedara., May 2018), con la ayuda del PLN se puede tener aplicaciones como:

- *Tokenization*

Es la tarea de separar un texto (corpus) en unidades denominadas tokens. Los tokens de una oración por ejemplo puede ser: palabras.

Ejemplo: “La canción de navidad”

Tokens:

- La
- canción
- de
- navidad

- *Desambiguar de palabras (WSD Word-sense Disambiguation)*

Es la tarea de identificar correctamente el significado de una palabra. Por ejemplo:

- Esteban es blanco de todas las críticas.
- Esteban usa un pantalón blanco.

- Reconocimiento de nombre de una entidad (*NER Named Entity Recognition*)

Es una tarea de identificar entidades dentro del texto, por ejemplo extraer personas, localidades y organizaciones.

- Etiquetado de partes de la oración (*PoS Part-of-Speech*)

Es una tarea de realizar un análisis sintáctico de la oración y etiquetar a las palabras que son nombres, verbos, determinantes, entre otros.

- Clasificación de sinopsis (*Sentence/Synopsis classification*)

Es la tarea de clasificar un texto, de acuerdo a su contenido. Por ejemplo detección de spam, clasificación de artículos de deportes, política, entre otros.

- Generación de Lenguaje (*Language Generation*)

Es la tarea de predecir nuevo texto a partir de un texto dado. Por ejemplo recomendaciones de búsquedas.

- Preguntas y Respuestas (*QA Question Answering*)

Es una tarea de producir respuesta a partir de preguntas. Estas aplicaciones son muy utilizadas en el área comercial, ya que estas técnicas sirven para sistemas de chatbots, o asistentes como Google Assistant o Apple Siri.

- Traductores automáticos (*MT Machine Translation*)

Es la tarea de transformar una oración o frase de un lenguaje a un nuevo lenguaje. Es una tarea complicada, ya que los lenguajes tienen estructuras morfológicas complejas y para esto se necesita utilizar redes neuronales artificiales.

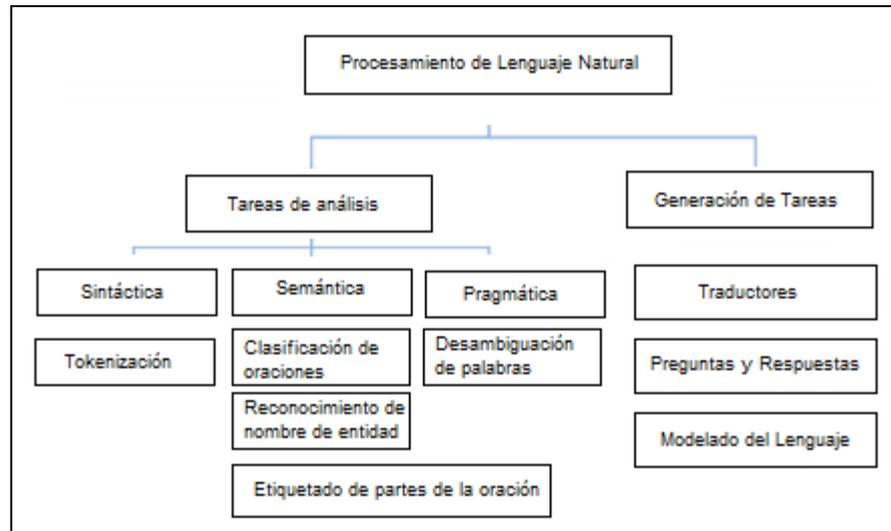


Figura 1 Taxonomía de las aplicaciones de PLN (Ganegedara., May 2018)

2.3 Enfoque tradicional del procesamiento del lenguaje natural (PLN)

El enfoque tradicional, los problemas de PLN se resuelven utilizando algoritmos de *machine learning*. Dichos algoritmos, se enfocan en la extracción de características de las representaciones de los datos para realizar el entrenamiento de un modelo. Para este entrenamiento se requieren dos tipos de datos: datos de test y datos de validación. Así se pone a prueba el modelo entrenado (Ganegedara., May 2018).

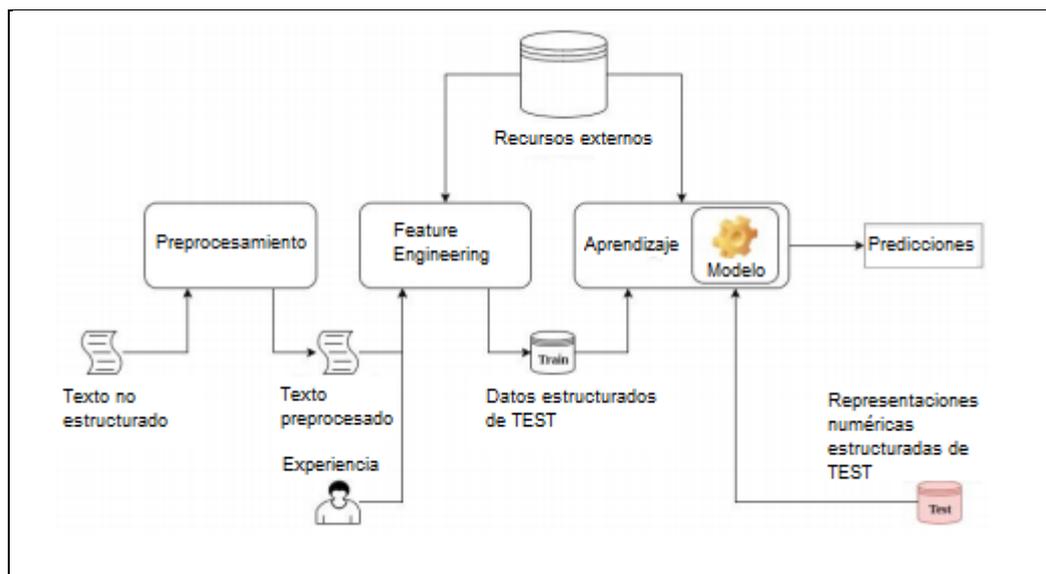


Figura 2 Enfoque general de un PLN clásico (Ganegedara., May 2018)

El enfoque general se basa en una serie de tareas que se deben realizar, empezando por obtener un corpus lo suficientemente grande que permita entrenar el modelo de manera adecuada. Entre los corpus más importantes podemos mencionar a WordNet, que es una base de datos léxica para el idioma Inglés (PRINCETON).

2.3.1 Desventajas del método tradicional

De acuerdo a (Ganegedara., May 2018), se tienen varios problemas con el método tradicional del procesamiento de lenguaje, entre ellos se mencionan:

- El pre procesamiento de la información es muy importante, convertir los datos no estructurados, en una representación que el modelo pueda utilizar de mejor manera para generar el aprendizaje.
- El *Feature Engineering* de los datos se debe realizar manualmente, lo que le convierte en una tarea compleja y tediosa. Adicionalmente debe existir una experiencia de dominio, que para ciertas áreas puede ser escaso.
- Los recursos externos son muy necesarios para un buen rendimiento, pero son escasos. algunas ocasiones, es mejor generar manualmente dichos recursos utilizando bases de datos o recolectando información por mucho tiempo.

2.3.2 Enfoque de *Deep learning* del Procesamiento del Lenguaje Natural (PLN)

El *Deep learning* está revolucionando los campos de Visión de Computador, Reconocimiento de Habla y por supuesto Procesamiento del Lenguaje Natural.

Gracias a los algoritmos de *deep learning*, se pueden crear modelos que permiten analizar gran cantidad de datos eliminando la complejidad de *Feature Engineering* y pasar del tradicional modelo de *machine learning* a un nuevo modelo que permita tener un mejor proceso de aprendizaje. (Ganegedara., May 2018).

2.4 *Deep learning* en Chatbots

2.4.1 Historia del *deep learning*

En 1943 los neurocientíficos Warren McCulloch y Walter Pitts publican un artículo “*A Logical Calculus of Ideas Immanent in Nervous Activity*”, en el cual describen una analogía entre una neurona con una función e activación y oraciones lógicas de primer orden. (Wallis)

El modelo de Warren McCulloch y Walter Pitts, puede representarse en la siguiente figura:

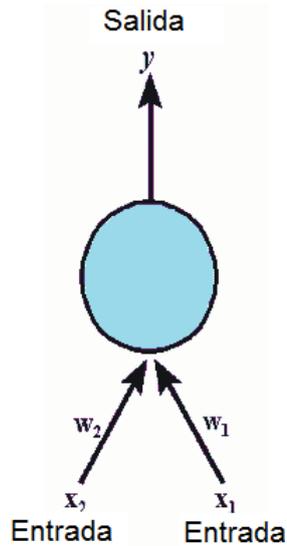


Figura 3 Representación de McCulloch y Pitts (Wallis)

En 1949 Donal Hebb, en su libro "*The Organization of Behavior*" descubre que el axón de una neurona puede participar en el proceso de activación de otra neurona, esta activación de las dos neuronas, es una operación fundamental para el aprendizaje y memoria. (Wallis)

En 1960, cuando Hubel y Weisel, realizan un experimento y descubren que la corteza visual de un gato es hecha de celdas simples y complejas (neuronas) organizadas de forma jerárquica. Es modelo inspiró a los investigadores a replicar un comportamiento similar en las máquinas, naciendo el concepto de *deep learning*. (Wallis)

En 1962 Frank Rosenblat, utilizando la neurona de McCulloch y Pitts, con los descubrimientos de Hebb, propone en su libro "*Principles of Nuerodynamics*", una representación básica de un perceptrón. (Wallis)

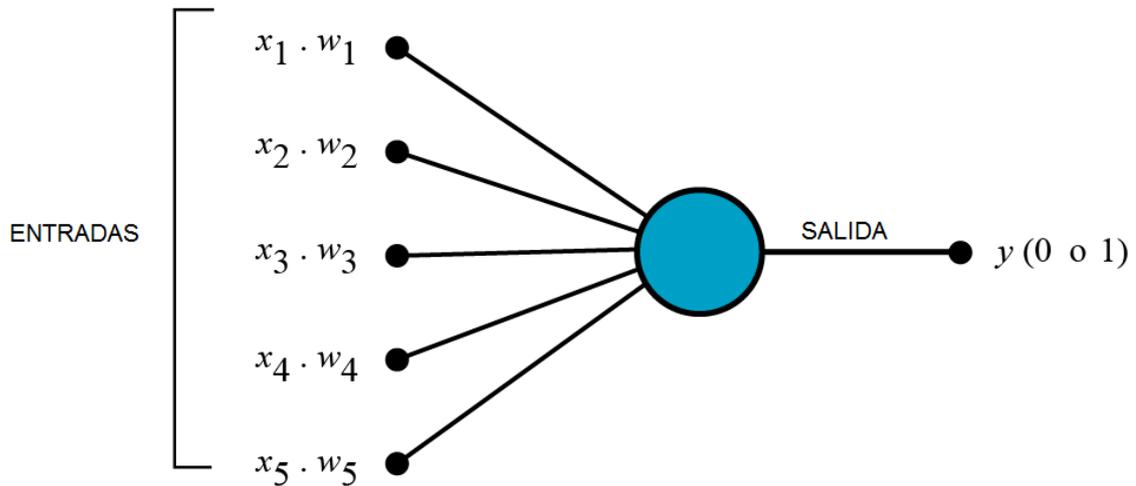


Figura 4 Perceptrón (Wallis)

Posteriormente, el perceptrón simple es combinado, para formar una red denominada perceptrón multicapa o red neuronal artificial, que consta de una capa de entrada, una de salida y una o varias capas ocultas.

De acuerdo a (Ganegedara., May 2018), las redes neuronales artificiales, llamaron la atención de muchos investigadores y así, basándose en el modelo del Perceptrón en 1965 se presenta un método conocido como *Group Method of Data Handling* (GMDH) para entrenar una red neuronal.

En 1979 Fukushima introduce el concepto de "*Neocognitron*", que sirvió de base para las Redes Neuronales Convolucionales, a diferencia de que un perceptrón tenía una entrada de una dimensión, el *Neocognitron* puede procesar entradas de dos dimensiones.

En los años 80 nace la idea del algoritmo de back-propagation, que sirve para el entrenamiento de las redes neuronales artificiales.

En la década de los 90 nacen las redes Neuronales Recurrentes y las Redes Neuronales Secuenciales (LSTM).

Sin embargo las redes neuronales no pueden alcanzar su máximo desempeño debido a varios problemas en el algoritmo de back-propagation que resultaba en un *vanish gradient* (Chandrashekar Lakshminarayanan, 2020), debido al número limitado de capas de la red. En 2016 gracias a los aporte de Geoffrey Hinton, Yoshua Bengio y Yann LeCun, se puede solventar dichos problema y las redes neuronales vuelven a ocupar el centro de atención de los investigadores.

Y es a partir del 2012, las redes neuronales alcanzan un auge extraordinario, gracias a dos factores:

- La cantidad de datos disponibles para entrenamiento.
- La capacidad de procesamiento de las computadoras.

Para el procesamiento del lenguaje natural se ha experimentado con varias redes

2.4.2 Redes Neuronales para el procesamiento de lenguaje natural

De acuerdo a (Junjie Yin), la red neuronal recurrente y las redes neuronales convolucionales son predominantes en el manejo de tareas de procesamiento de lenguaje natural. En principio las redes convolucionales han sido utilizadas en visión artificial pero se han utilizado para el procesamiento de textos, obteniendo óptimos resultados.

2.4.2.1 LSTM (Long Short Term Memory)

Son una variante de las redes recurrentes que solventan el problema de vanish gradient. Están diseñadas para recordar estados por largo tiempo, es decir aprender las relaciones entre elementos lejanos a lo largo de períodos largos. (Bhagwat, 2018)

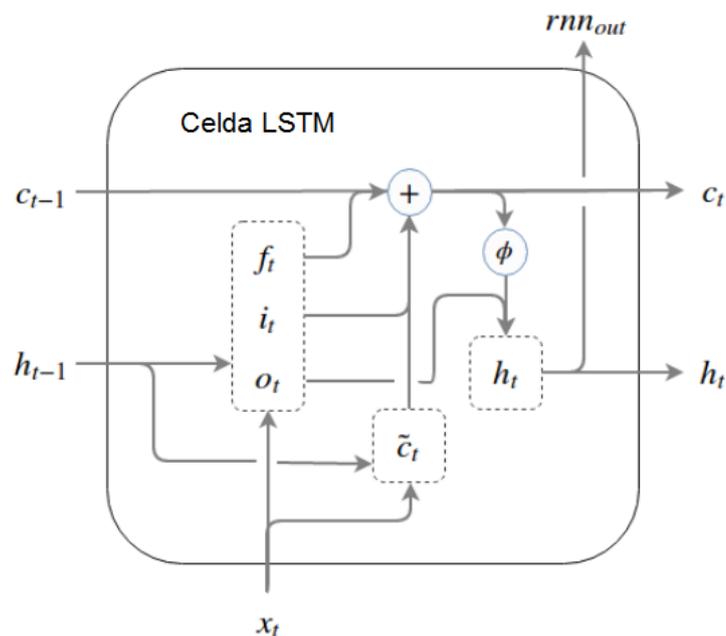


Figura 5 Celda básica LSTM (Bhagwat, 2018)

Actualmente estas redes son muy utilizadas en etiquetado de partes de la oración, obteniendo muy buenos resultados, así como en tareas de generación de texto.

2.4.2.2 Modelos Secuencia a Secuencia Sec2Sec

Los modelos Sec2Sec, están basados en Redes Neuronales Recurrentes y consisten en dos redes conectadas que actúan como codificador (entrada) y decodificador (salida). El nodo decodificador produce una salida en base al nodo codificador

Este modelo es muy robusto para traducción y para generación del lenguaje, de tal manera que está siendo utilizado en chatbots modernos (Bhagwat, 2018).

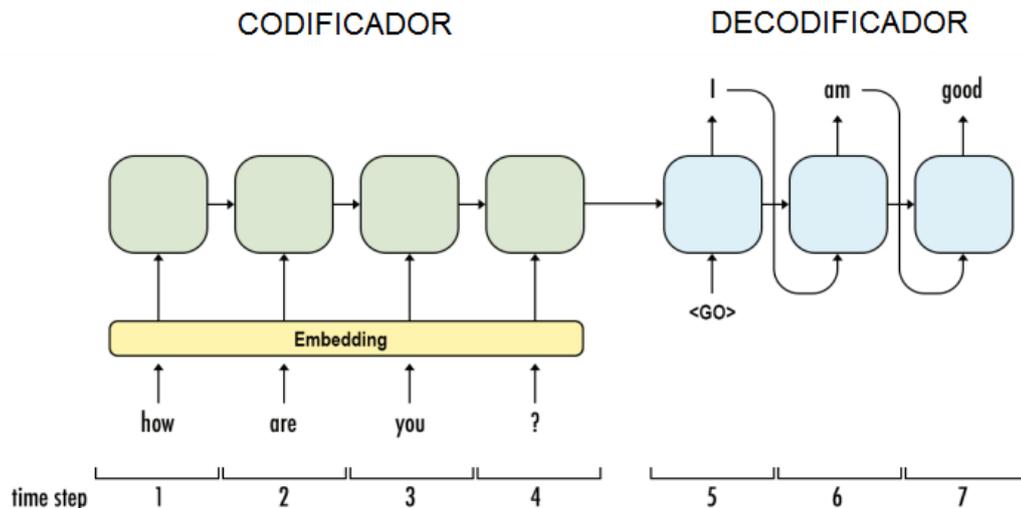


Figura 6 Modelo Sec2Sec (Bhagwat, 2018)

2.5 Funciones del chatbot

Según (Cahn, 2017) un chatbot debe tener tres funciones principales:

- **Agente de Diálogo:** Proporciona la función de comprensión, que permite al usuario a través de una interfaz GUI (Graphical User Interface) ingresar texto para obtener ayuda.
- **Agente Racional:** Proporciona la función y competencia, para administrar las solicitudes del usuario y emitir las respuestas.
- **Agente encarnado:** Proporciona la función de presencia, personificando a un ser humano, para generar confianza en el usuario, es por eso que debe tener un nombre de persona como por ejemplo los famosos agentes ELIZA, ALICE, etc.

2.6 Historia de los chatbots

En 1950 Alan Turing, plantea la siguiente pregunta ¿Las máquinas pueden pensar? Para resolver esta pregunta, se diseña un juego en el cual se realizan preguntas a seres humanos y máquinas, si las personas que realizan las preguntas no pueden diferenciar si es una persona la que responde o es una máquina, entonces, la respuesta es “Las maquinas pueden pensar”. (Cahn, 2017)

Así nace ELIZA en 1966, el primer chatbot que podía responder preguntas y superar el denominado “*Turing Test*”, creada por Joseph Weizenbaum en el MIT

2.6.1 Chatbots históricos

- **ELIZA** fue el primer paso para el procesamiento del lenguaje natural, basado en un modelo simple de expresiones regulares y comodines (*wilcards*) que buscaba reconocer palabras, siguiendo ciertas reglas dentro de una pregunta y responder con frases predefinidas. ELIZA ha servido de inspiración para la creación de nuevos y más modernos chatbots (Cahn, 2017).
- **PARRY**, creado en 1972, por Kenneth Colby de Stanford, un chatbot que suplantó un paciente con un cuadro de esquizofrenia y paranoia. (Colby, 1981).
- **A.L.I.C.E.** (*Artificial Linguistic Internet Computer Entity*), desarrollado en 1955 por Dr. Richard Wallace, es un chatbot inspirado en ELIZA. A.L.I.C.E. fue escrito en AIML (*Artificial Intelligence Markup Language*). AIML es una extensión de XML, que fue desarrollada por el mismo Richard Wallace. Este chatbot ha sido 3 veces ganador del premio Loebner, que es una competición de chatbots que deben superar el “*Turing Test*” y demostrar ser lo más inteligentes. (Wallace). Ahora es parte del proyecto PANDORA, un proyecto comercial que permite crear chatbots para diferentes aplicaciones. (Cahn, 2017)

2.6.2 Aplicaciones de un chatbot

Existen muchas aplicaciones en la actualidad en la que los chatbots cumplen importante funciones, entre ellas podemos mencionar:

- Asistentes en teléfonos inteligentes o PCs (*Personal Computers*).
- Asistentes de sitios web.
 - Atención de servicio al cliente.
 - Ventas.
 - Procesos de soporte.
- Asistentes telefónicos.
- Agentes de SMS.

- Agentes de redes sociales (WhatsApp, Telegram, Facebook, Twitter, entre otras) (Falko Koetter). De acuerdo a (Sugisaki), más de 300.000 chatbots se han desarrollado en Facebook Messenger desde mediados de 2018.
- Asistentes en juegos.
- Asistentes de respuestas en email, entre otras.
- Traductores.

2.6.3 Tipos de Chatbots

De acuerdo al incremento de aplicaciones, los desarrolladores han creado diferentes tipos de chatbots, en base a lo mencionado, se puede citar una clasificación propuesta por la revista chatbotmagazine. (chatbotmagazine)

- **Single-feature Chatbot:** chatbots, que poseen una sola característica. Tienen limitadas funcionalidades para un uso específico y simple. Por ejemplo, un traductor de texto simplificado.
- **Proactive Chatbots:** chatbots, que colocan información al usuario, en lugar de esperar preguntas. No son interactivos, simplemente ponen a disposición del usuario cierta información en determinados momentos.
- **Group Chatbots:** chatbots que interactúan con un grupo de usuarios. Responden a requerimientos específicos de los usuarios. Por ejemplo en juegos en línea, asignan recursos a los usuarios.
- **Simplification Chatbots:** chatbots con una interfaz simple, pero que permite resolver trámites automatizados. Por ejemplo pago de tickets de parqueadero.
- **Entertainment Chatbots:** chatbots, cuya funcionalidad es establecer una conversación con el usuario.
- **Personal Assistants:** chatbots que combinan diferentes funcionalidades y sirve como asistente de tareas para un usuario. Por ejemplo asistentes de teléfonos inteligentes (*smartphones*).
- **Optimization Chatbots:** chatbots utilizados en sitios web. Estos chatbots son entrenados como asistentes de ventas o servicio al cliente, interactúan con los usuarios a través de conversaciones dentro de un ámbito específico.

2.6.4 Ventajas del uso de chatbots

Como se había comentado anteriormente, existen varias aplicaciones en las cuales el uso de chatbots optimiza los recursos y facilita las comunicaciones. La principal ventaja es que los mensajes se transmiten utilizando el lenguaje natural del ser humano, ya sea escrito o hablado. Es decir se puede comunicar a las máquinas los pensamientos u órdenes en la

forma natural de hacerlo de un ser humano. De esta manera, la visión de un chatbot sería: “Que las máquinas tengan la capacidad de dar respuestas a preguntas y poder establecer una conversación como un humano lo haría”.

Entre otras ventajas tenemos

- Automatización de comunicación.
- Ahorro de recursos.
- Ahorro de tiempo.

Adicionalmente se pueden entrenar chatbots con ciertos perfiles, que permitan imitar pensamientos de acuerdo a realidades, creencias o incluso enfermedades mentales, que sirven para el análisis de ciencias médicas.

2.6.5 Limitaciones

Los chatbots son aplicaciones que corren sobre las redes de datos y que requieren el intercambio de información entre el usuario y el servidor, en los cuales se aloja su código fuente, de ahí que siempre va a ser necesario que exista conectividad entre los diferentes dispositivos.

Para que los chatbots puedan responder de manera adecuada, estos deben ser muy bien entrenados. El proceso de entrenamiento puede ser complejo y siempre se requerirá de nuevos entrenamientos. Si el proceso de entrenamiento ha sido realizado de manera satisfactoria, el chatbot tendrá una gran capacidad de respuestas. Adicionalmente, el hecho de que un chatbot pueda responder a preguntas, no significa que pueda establecer una conversación con un ser humano. La conversación es un proceso complejo de intercambio de frases que no necesariamente está basada en preguntas y respuestas, sino más bien muchas de estas frases son ideas, pensamientos, conclusiones que están relacionadas entre sí. Por este motivo una gran limitante a vencer es que el chatbot pueda entender el contexto de una frase que le permita ubicarse dentro del sentido de la conversación.

2.6.6 Chatbots actuales

Actualmente existe un gran número de chatbots, especialmente comerciales, entre los más populares podemos mencionar los de las grandes marcas de INTERNET:

- Amazon Echo and Alexa,
- Apple's Siri,
- Microsoft's Cortana,
- GOOGLE NOW, entre otros.

Esta nueva era de chatbots, con mejores resultados, un procesamiento del lenguaje natural más avanzado y mejores respuestas, tiene lugar gracias al avance de modelos *machine learning* y el *deep learning*. Es así que hoy en día, una de las prácticas, para el desarrollo de chatbots es el uso de Seq2Seq, un algoritmo de SMT (*statistical machine translation*), que utiliza RNNs (*recurrent neural networks*) para codificar y decodificar entradas en respuestas. (Jurafsky).

2.7 Frameworks para el desarrollo de Chatbots

Un *framework* en *deep learning*, es una librería o conjunto de librerías que facilita el desarrollo e implementación de algoritmos de *machine learning* y *deep learning*.

Existen varias ventajas de utilizar un *framework*, entre ellas podemos mencionar:

- Optimización de código.
- Facilidad del manejo de GPU (*Graphics Processing Unit*) y ambientes distribuidos.
- Optimización de cálculos matemáticos.
- Optimización de entrenamiento.
- Facilidad de uso de modelos.

En *deep learning* tenemos varios *frameworks*, entre los más importantes podemos mencionar:

- **TENSORFLOW:** Es un *framework open source*, que utiliza grafos de computación en forma de tensores. Fue desarrollado por GOOGLE y liberado en 2015. Ofrece una API (*Application Programming Interface*) en Python para su utilización, aunque la gran mayoría de su código está desarrollado en C++. Es uno de los *frameworks* más populares hoy en día (TENSORFLOW).
- **THEANO:** Es uno de los primeros *frameworks* para *machine learning* y *deep learning*. Escrito en Python y creado en 2007 por Yoshua Bengio. Aunque su desarrollo fue interrumpido en el 2017, THEANO ha servido de modelo para otros *frameworks* actuales (LAB).
- **PYTORCH:** Es un *framework open source*, escrito en Python. Está basado en TORCH (*framework* clásico como THEANO) Fue desarrollado por FACEBOOK y está alcanzado gran uso en estos tiempos (PYTORCH).
- **CAFFE 2:** Basado en el *framework* CAFFE, también propiedad de FACEBOOK. Está escrito en C++, con una API en Python. Actualmente el proyecto CAffe2 es parte de PYTORCH (Source).

- **CNTK:** *Microsoft Cognitive Toolkit* es un *framework* de la empresa MICROSOFT, que permite entrenar algoritmos de *deep learning*. Existe una versión *open source* y una comercial, con funcionalidades avanzadas (Microsoft, Microsoft Docs).
- **MXNET:** Apache MXNet es un *framework open source* para entrenar modelos de *deep learning*, sin embargo que a diferencia de los antes mencionados nos da la posibilidad de tener varias APIs (*Application Programming Interface*) para su uso, entre ellas están las de *PYTHON*, *SCALA*, *R*, *PERL*, *C++* entre otras, facilitando así su uso (APACHE).
- **KERAS:** Keras es un *framework open source*, escrito en Python, que facilita el entrenamiento de modelos, simplificando el código. Keras se enfoca en tres objetivos:
 - Permitir un uso fácil y rápido de prototipos.
 - Soportar redes convolucionales y recurrentes y sus combinaciones.
 - Entrenar modelos utilizando CPU (*Central Processing Unit*) y GPU (*Graphics Processing Unit*).

Keras utiliza como base otros *frameworks* como TensorFlow, Theano o CNTK (KERAS).

Estos *frameworks* pueden ser utilizados para realizar el procesamiento del lenguaje natural y así poder empezar a construir un chatbot. Sin embargo un chatbot requiere de algunos otros elementos, que le permitan cumplir con sus objetivos de manera eficaz y eficiente. Un chatbot no solo es capaz de procesar el lenguaje natural sino también de poder emitir respuestas coherentes ya sean constantes o variables, es decir debe tener la capacidad de establecer conexiones con bases de datos para realizar consultas y mejorar su capacidad de respuestas, así mismo debe ser capaz de almacenar variables en memoria y poder utilizarlas durante la conversación, como por ejemplo preguntar al inicio de la conversación un nombre y poder recordarlo al final de la misma.

Uno de los aspectos más importantes y complejos de un chatbot, es la capacidad de mantener una conversación coherente a medida de lo posible, es decir poder seguir un hilo de conversación. Las expresiones de los seres humanos son muy ambiguas, por ejemplo una respuesta como "SI", "NO", "DE ACUERDO", entre muchas más, dependerán del contexto de la conversación.

Es por este motivo que no solo se requiere de una librería capaz de procesar el lenguaje natural sino toda una plataforma de construcción de conversaciones.

De acuerdo a un artículo publicado por (Verdugo) y también mencionadas en el artículo de (Xingkun Liu), entre las plataformas que brindan la programación y construcción de chatbots se tiene:

- **MICROSOFT BOT FRAMEWORK:** Es un *framework* de propiedad de la empresa Microsoft, para aplicaciones de inteligencia artificial, que permite crear chatbot a nivel empresarial. Dichos chatbots pueden ser utilizados para responder preguntas frecuentes o asistentes virtuales más sofisticados. Utiliza un SDK (Software Development Kit) de código abierto que permite diferentes tipos de integraciones con canales de comunicaciones o dispositivos, como son sitios web y redes sociales, además ofrece la capacidad de entender texto escrito como hablado y de brindar respuestas en texto o mediante voz (Microsoft, Microsoft Bot Framework).
- **AMAZON LEX:** Es un *framework* de propiedad de la empresa AMAZON, permite crear interfaces de conversación de voz y texto. Utilizando tecnología de *deep learning*, tiene la capacidad de realizar procesamiento de lenguaje natural y la creación de chatbots sofisticados. Un ejemplo de esta tecnología es ALEXA (Amazon).
- **DIALOGFLOW:** Es un sistema que permite crear interfaces conversacionales de voz y texto, con la capacidad de procesamiento de lenguaje natural. Además permite integrarse con varios canales de comunicación, como son sitios web, aplicaciones móviles, redes sociales como Facebook Messenger, Amazon Alexa, entre otras (Google).
- **IBM WATSON:** Es una suite de herramientas de inteligencia artificial con aplicaciones prácticas en varias industrias. Consta de varios módulos de procesamiento de lenguaje natural, Text to Speech, Speech to Text, Traductores, entre otros (IBM).
- **WIT.AI:** Son un conjunto de librerías, orientadas a desarrolladores que desean incorporar en sus aplicaciones procesamiento del lenguaje natural. Permite la creación de chatbots capaces de procesar voz o texto (WIT.AI).
- **OTRAS SUITS COMERCIALES:** Existen, muchos otras plataformas que permiten la creación de chatbots, para redes sociales y sitios web, entre ellos podemos mencionar BOTFUEL, PANDORABOTS, SAP CONVERSATIONAL AI, entre otros. En general todos muestran características similares que permiten la creación de chatbots con interfaces de programación de alto nivel, muchas de ellas utilizan las suites de herramientas como DIALOGFLOW o WIT, con una capa de presentación más amigable y fácil de configurar para el usuario.

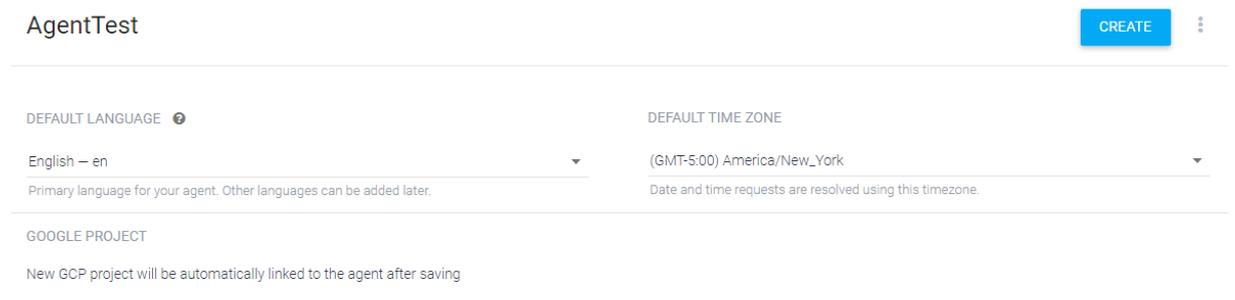
A continuación se describe brevemente, una de las herramientas más utilizadas para la creación de chatbots: **DIALOGFLOW**. Se elige esta herramienta, por la cantidad de documentación existente y además porque provee un entorno de desarrollo libre y gratuito.

2.7.1 DIALOGFLOW

DIALOGFLOW es una plataforma de procesamiento de lenguaje natural que permite el diseño y desarrollo de interfaces de conversación e integración con diferentes canales de comunicación (Google).

2.7.1.1 AGENTS

Un agente maneja conversaciones con los usuarios finales. Un agente es una simulación de un agente humano, que tiene capacidad de procesar el lenguaje natural y estructurar los datos, ya sea de mensajes escritos o audio. Un agente será entrenado para manejar un lenguaje como primario, pero puede soportar varios lenguajes durante su entrenamiento.



The screenshot shows the configuration page for a new agent named "AgentTest". At the top right, there is a blue "CREATE" button and a vertical ellipsis menu icon. Below the agent name, there are two dropdown menus: "DEFAULT LANGUAGE" set to "English -- en" and "DEFAULT TIME ZONE" set to "(GMT-5:00) America/New_York". Below these are two sections: "GOOGLE PROJECT" with a note "New GCP project will be automatically linked to the agent after saving".

Figura 7 Creación de un Agente (Google)

2.7.1.2 INTENTS

Un *INTENT* clasifica la intención de comunicación de un usuario. Los *INTENTS* deben ser entrenados con frases esperadas (CORPUS de frases) y dentro de un ámbito específico, por ejemplo saludos, productos, despedidas, etc.

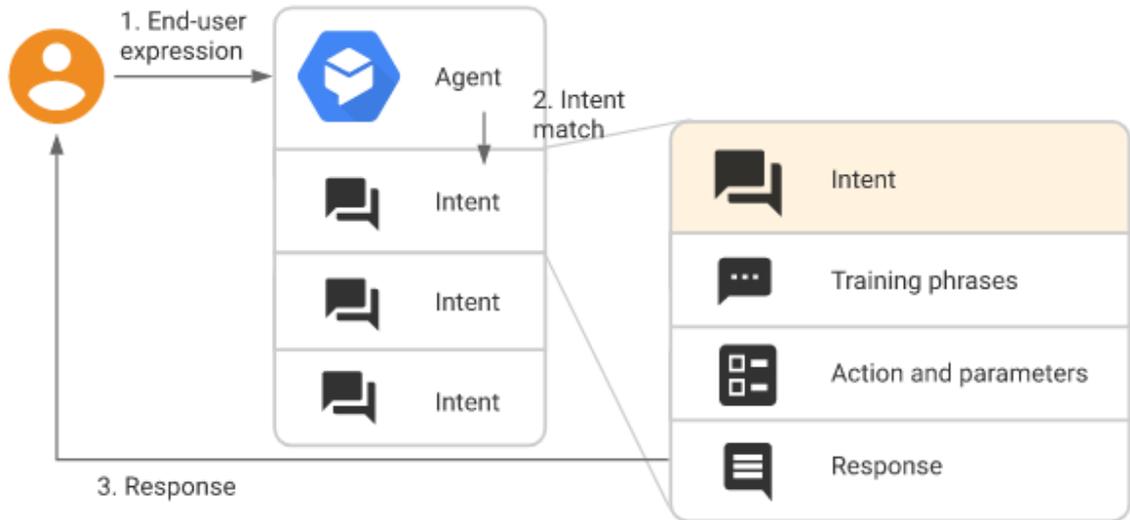


Figura 8 INTENT (Google)

Durante el proceso de entrenamiento, el *INTENT* tiene la capacidad de extraer variables y emitir respuestas de acuerdo al entrenamiento realizado.

2.7.1.3 ENTITIES

Los *ENTITIES* son tipos de variables que se pueden extraer de las frases que se entrenan en los *INTENTS*. Estos tipos permiten limitar el tipo de respuesta y mejorar la capacidad de comprensión.

Existen varios de *ENTITIES* predefinidos, como por ejemplo, email, país, ciudad, número, etc., sin embargo se pueden programar *ENTITIES* adicionales dependiendo de las respuestas que se esperan.

2.7.1.4 CONTEXTS

Los *CONTEXTS*, permite generar un flujo de conversación, darle sentido y un hilo a la conversación. Permiten adicionalmente la exportación e importación de variables reconocidas en algún *INTENT* y sobretodo desambiguar expresiones comunes como un SI o NO, ya que dependiendo del contexto se conocerá a que obedece cada respuesta.

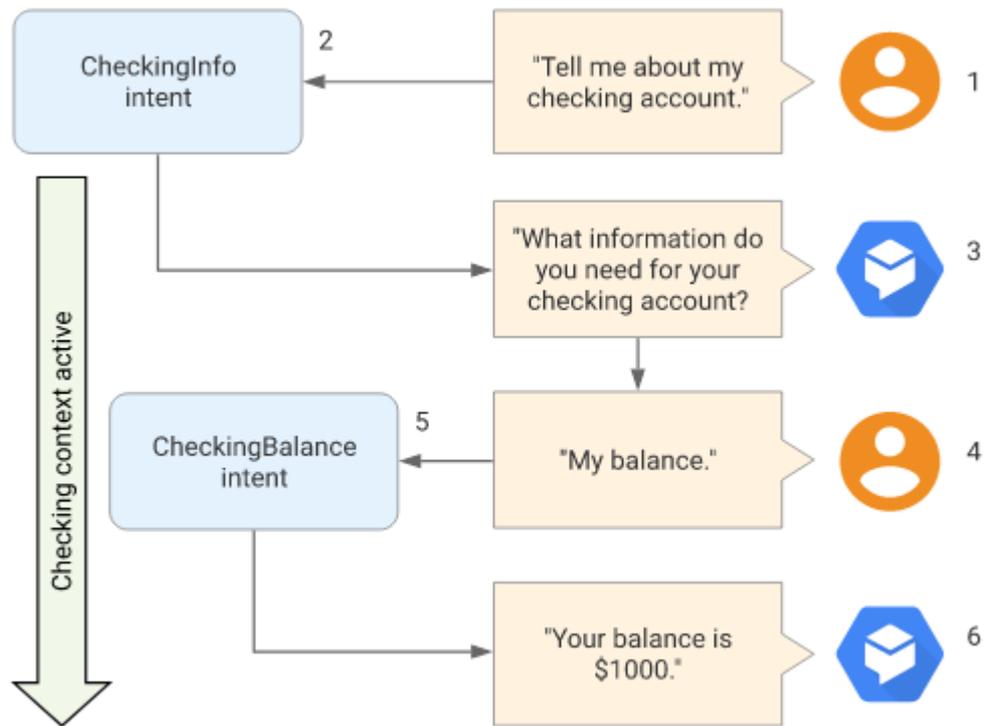


Figura 9 Contextos (Google)

2.7.1.5 INTEGRATIONS

El módulo de *INTEGRATIONS*, permite incorporar canales de comunicación con un sitio web, Facebook Messenger, Telegram, entre otros, es decir los medio comunes por los cuales los usuarios interactúan con el agente.

2.7.1.6 FULLFIMENTS

Este módulo permite ampliar la funcionalidad del agente, ya que permite integrarse con otras plataformas de desarrollo, especialmente con bases de datos u otros módulos que sean necesarios para procesar la información.

2.8 Conclusiones del estado del arte

Un chatbot es más que un sistema de procesamiento de lenguaje natural, es un agente conversacional, es decir que debe tener la capacidad de generación de respuestas dentro de un contexto adecuado.

Existen dos obstáculos principales a superar en el desarrollo de un chatbot:

- Procesamiento de lenguaje natural
- Contexto de una conversación

Para los entrenamientos se requiere la redacción de un corpus de frases, que dependiendo la aplicación del chatbot podrán o no existir. Para el idioma español no existen corpus estructurados que permitan realizar un entrenamiento adecuado, el objetivo de estructurar un corpus es muy amplio y se debería considerar como un proyecto independiente.

El problema de las conversaciones humanas, es que las frases que las componen son muy ambiguas, este es uno de los obstáculos más grandes a vencer, ya que las frases se deben interpretar desde un contexto dentro de la conversación.

Adicionalmente el chatbot no puede existir en un ambiente aislado es necesario que se integre dentro de un sistema de comunicaciones con el cual interactúan los usuarios, este puede ser una página web o una red social. Finalmente un chatbot también debe tener capacidad de integrarse con otros sistemas tradicionales para el procesamiento de datos, que le permita realizar cálculos o consultas en bases de datos u otros medios de almacenamiento digital.

Para el desarrollo de chatbots, primero se debe especificar cuál va a ser el objetivo del mismo, es decir establecer claramente el ámbito de repuestas del chatbot, en otras palabras, se debe profesionalizar al chatbot para que cumpla con ciertos objetivos específicos y tenga capacidad de respuesta en cierta área específica del conocimiento.

Finalmente, si un chatbot fuese muy bien entrenado para que tenga gran capacidad de respuesta, esto no significa que el chatbot tenga un pensamiento propio, es decir que pueda emitir ideas por su cuenta, característica propia de los seres humanos.

Para la construcción de un chatbot existen varias alternativas, desde librerías que permiten la construcción de redes neuronales para el procesamiento del lenguaje natural y generación de respuestas hasta plataformas que facilitan el desarrollo de chatbots incorporando módulos de contextos, integración con sitios web y redes sociales.

Es necesario el uso de estas librerías que ya poseen pre entrenamientos, con corpus extensos y en infraestructuras de hardware muy bien equipadas.

2.9 Contribución del TFM

El proyecto desarrollado en este trabajo realiza las siguientes contribuciones:

- Detallar el estado de arte de los chatbots, para poner en conocimiento del lector las posibles tecnologías y herramientas existentes, así como los beneficios, ventajas y limitaciones de los chatbots en la actualidad.

- Redactar un corpus orientado a ser un asistente de ventas. No existe un corpus para este trabajo, ya que es muy específico el tema del conocimiento. Este corpus puede servir de ejemplo base o guía para otros proyectos de similares características.
- Realizar un entrenamiento detallado y muy bien documentado utilizando dos herramientas y mostrar el cómo se realiza el entrenamiento en cada una de ellas. Al incorporar un segunda herramienta de entrenamiento se potencializa la capacidad de procesamiento de lenguaje natural del chatbot y al mismo tiempo se resuelve en cierta medida el problema de ambigüedad.
- El desarrollo de un flujo de conversación que permita tener el control de la misma al chatbot. El chatbot debe ser consistente en sus respuestas y guiar la conversación, caso contrario la comunicación no fluirá y el visitante se desanimará de continuar interactuando con un robot.
- El principal aporte es el dotar de una herramienta de inteligencia artificial a la organización que auspició este proyecto, ya que la misma no cuenta con un sistema de esta naturaleza, permitiéndole potencializar su sitio web e incrementar sus probabilidades de cierre de ventas, al mostrar innovación y atención inmediata. No obstante, el chatbot desarrollado en este proyecto sirve de base para otras organizaciones de similares características que deseen incorporar a sus sitios web, inteligencia artificial. Por lo cual este trabajo se convierte en una guía indispensable que toda pequeña empresa ya sea de Ecuador u otro país, puede seguir para incorporar un chatbot en su sitio web.

3 Objetivos y metodología

3.1 Objetivo General

Desarrollar un chatbot, que cumplirá la función de un asistente de ventas para los visitantes de la página web, propiedad de la empresa SIDEVOX S.A.::

<https://www.contactvox.com>.

3.2 Objetivos Específicos

- Incorporar las funcionalidades de un chatbot al sitio web de la empresa SIDEVOX S.A. para la atención de ventas.
- Probar varias herramientas que permiten la construcción de chatbots, con el objetivo de desarrollar un chatbot que brinde buena comprensión del lenguaje natural.
- Integrar de ser necesario más de una herramienta en el desarrollo del chatbot, para mejorar la comprensión del lenguaje natural y la generación de respuestas.
- Integrar el chatbot desarrollado al sistema de comunicaciones existente en dicha empresa.
- Brindar un servicio automático de respuestas 24/7.
- Brindar una guía rápida de los productos y servicios que la empresa brinda a los visitantes de su sitio web.
- Brindar respuestas a preguntas frecuentes de los productos y servicios que la empresa brinda a los visitantes de su sitio web.
- Reducir lo más posible repuestas como “no entiendo”, “podrías repetirlo”, es decir tratar la máximo que el chatbot ofrezca fluidez en la conversación
- Brindar respuestas inmediatas y guiar en el acercamiento a la empresa a los visitantes de su sitio web.
- Comparar el chatbot desarrollados con otros chatbots de páginas ecuatorianas.

3.3 Metodología de trabajo

La metodología que se aplica para el desarrollo del chatbot es SCRUM. De acuerdo a (SCRUMstudy, 2013), “Un proyecto SCRUM implica un esfuerzo de colaboración para crear un nuevo Producto o, servicio, o cualquier otro resultado como se define en el Declaración de la Visión del Proyecto. Los proyectos se ven afectados por las limitaciones de tiempo, costo, alcance, calidad, recursos, capacidades organizativas, y otras limitaciones que los hacen difíciles de planificar, ejecutar, administrar y finalmente tener éxito. Sin embargo, la implementación exitosa de los resultados de un proyecto acabado le proporciona ventajas económicas significativas a una organización”.

3.3.1 Organización

Los roles de SCRUM se dividen en dos categorías:

1. Roles principales:
 - a. Producto Owner
 - b. Scrum Master
 - c. Equipo Scrum

Para esto proyecto unipersonal, yo seré quien cumpla con los tres roles y capte las necesidades del cliente (SIDEVOX S.A.)

2. Roles no esenciales:
 - a. Stakeholders

Para este proyecto solo se requiere la participación de un stakeholder que está representado por el cliente (SIDEVOX S.A.)

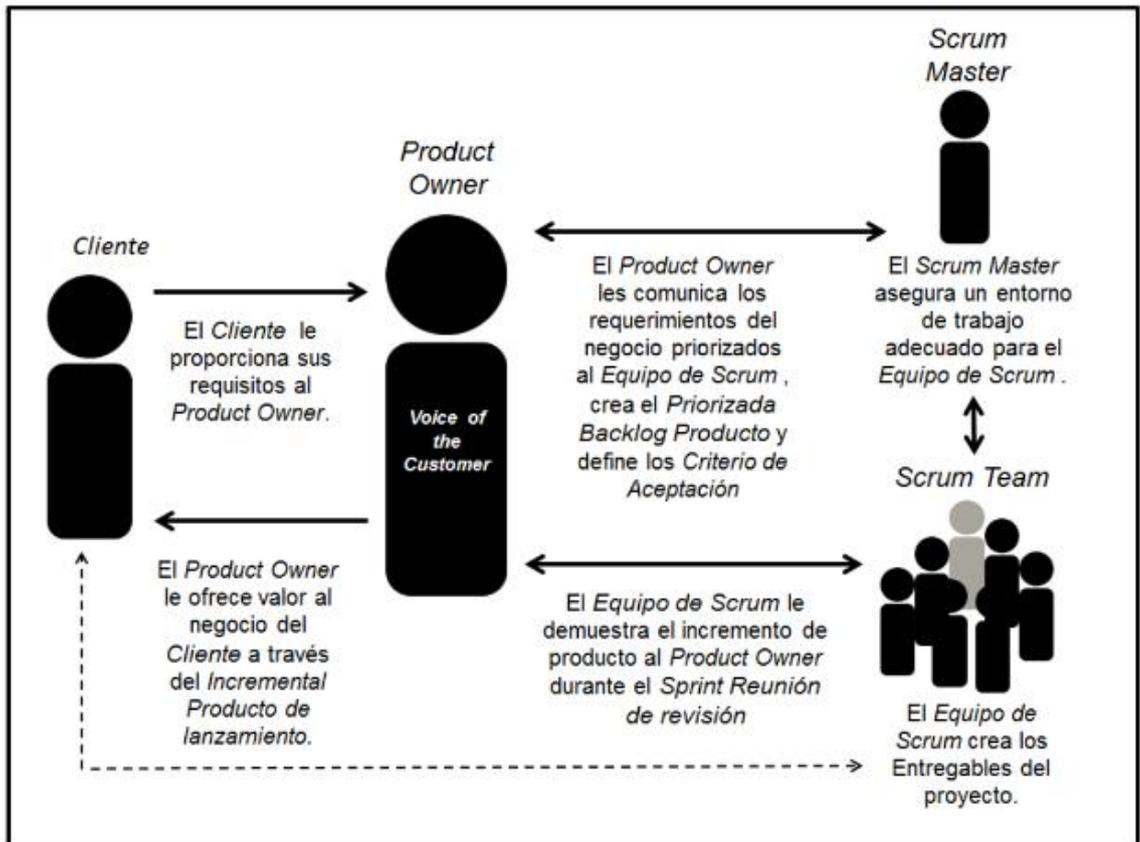


Figura 10 Organización en SCRUM (SCRUMstudy, 2013)

3.3.2 Procesos de SCRUM

De acuerdo a (SCRUMstudy, 2013), los procesos SCRUM abordan las actividades y el flujo específico de un proyecto Scrum. En SCRUM hay 19 procesos que se agrupan en cinco fases:

1. INICIAR
 - Crear la visión del producto
 - Identificar el Scrum Master and Stakeholder(s)
 - Formar el equipo Scrum
 - Desarrollo de Épica(s)
 - Crear la lista de pendientes del producto
 - Realizar la planificación del *release*
2. PLANEAR Y ESTIMAR
 - Crear historias de usuarios
 - Aprobar, estimar y comprometerse a las historias de los usuarios
 - Crear tareas
 - Estimar el trabajo
 - Crear la lista de pendientes del Sprint
3. IMPLEMENTAR
 - Crear el producto final
 - Realizar un *standup* diario
 - Mantenimiento priorizado de los pendientes del producto
4. REVISIÓN Y RETROSPECTIVA
 - Convocar Scrum de Scrums
 - Demostrar y validar el Sprint
 - Retrospectiva del Sprint
5. Lanzamiento
 - Entrega del producto final
 - Retrospectiva del proyecto

4 DISEÑO DE UN CHATBOT PARA ASISTENTE DE VENTAS

Para el desarrollo del chatbot se considera utilizar como guía la metodología SCRUM, debido a que el proyecto es personal no se tomarán en cuenta algunos roles ni procesos que se consideren innecesarios para el desarrollo del mismo.

4.1 Proceso de Inicio

4.1.1 Visión del producto

Para tener una visión clara del producto se tiene que considerar las siguientes entradas:

4.1.1.1 *Business Case*

Para la redacción del *business case* se toma en consideración las siguientes herramientas:

4.1.1.1.1 Reunión de la visión del proyecto

En esta reunión con el stakeholder (cliente), se identifica el contexto empresarial de la empresa SIDEVOX S.A.

SIDEVOX es una pyme ecuatoriana, que desarrolla software de comunicaciones y gestión empresarial. Su enfoque de negocios es B2B (*Business To Business*), es decir realiza transacciones comerciales con otras empresas. Las personas encargadas de tomar la decisión de compra de los servicios de SIDEVOX son profesionales enrolados en el área de tecnología y/o marketing. Dichas personas son los visitantes del sitio web, en el cual se exponen los servicios y productos de la empresa. El negocio requiere por un lado brindar una atención oportuna y rápida a dichos visitantes y adicionalmente la empresa desea proyectar innovación a través de la incorporación de herramientas de inteligencia artificial.

4.1.1.1.2 Diseño de aplicación conjunto

Alcance

El desarrollo del chatbot debe permitir establecer una conversación lo más coherente posible con el visitante de la web, ser capaz de responder a preguntas y de ser posible tomar datos del visitante. Es decir se requiere un sistema de inteligencia artificial, capaz de atender al visitante de la web de manera rápida y eficaz, a través de mensajería instantánea.

Objetivos

Los objetivos de este proyecto se enlistan en el capítulo 3, sección 3.1 y 3.2

Especificaciones

El chatbot debe atender al visitante de manera inmediata siempre y cuando el visitante inicie la conversación escrita. La atención se realiza a través del sistema de mensajería instantánea existente en el sitio web de la empresa.

4.1.1.1.3 Análisis FODA

Fortalezas

- La empresa ya cuenta con un sistema de mensajería instantánea instalada en su sitio web
- La empresa fue quien desarrolló su propio sistema de mensajería instantánea.

Debilidades

- La empresa atiende en un horario laboral y fuera del mismo, no tiene la capacidad de respuestas a sus visitantes. Incluso dentro del horario laboral los visitantes no son atendidos cuando el personal se encuentra ocupado en otras actividades.
- La empresa no cuenta con el personal suficiente para dar respuestas y atención a los visitantes del sitio web

Oportunidades

- Existen herramientas gratuitas y de fácil uso e integración que permiten incorporar sistemas de inteligencia artificial, dentro de sitios web.
- Existen profesionales de estudios de ingeniería o maestría que requieren poner en marcha proyectos de esta naturaleza con el objetivo de cumplir con sus proyectos de titulación.
- Las personas se están familiarizando con la interacción con robots e incluso generando una preferencia a su uso, por la rapidez y facilidad de consultas.

Amenazas

- Las políticas de uso de las diferentes librerías o *frameworks* para el desarrollo del proyecto, pueden cambiar con el tiempo. Se utilizaran herramientas libres y sin costo alguno.
- El estado del arte en, los chatbots, aún está en desarrollo, las técnicas de procesamiento de lenguaje natural siguen mejorando y es muy complicado llegar a desarrollar un robot capaz de establecer una conversación como un ser humano lo haría.
- No todas las personas desean ser atendidas por robots, prefieren tratar directamente con seres humanos

- Las personas tiende a tener expresiones escritas complejas, no suelen expresar bien sus ideas o expresan más de una idea en una sola frase, sin signos de puntuación y muchas veces con faltas ortográficas o palabras incompletas.

4.1.1.1.4 Análisis de brechas

Estado Actual

Chat atendido por personas.

Estado deseado

Chat atendido por robots

4.1.1.1.5 Declaración del *business case*

La empresa SIDEVOX S.A., posee una página web en la siguiente url: <https://www.contactvox.com>, en la cual presenta sus productos y servicios a través de su marca registrada CONTACTVOX. La empresa, tiene instalado en su página, un sistema de comunicaciones unificadas, que les permite a los visitantes, comunicarse con la empresa a través de chat, llamadas y video llamadas. Los chats con atendidos por personas que trabajan en la empresa, sin embargo dichas personas no siempre están disponibles durante el horario laboral y fuera de éste horario no se brinda ningún tipo de servicio de comunicación. El objetivo principal del proyecto es desarrollar un chatbot que cumpla las funciones de un asistente de ventas y que se integre con el sistema de comunicaciones existente. Los objetivos de este proyecto se enlistan en el capítulo 3, sección 3.1 y 3.2

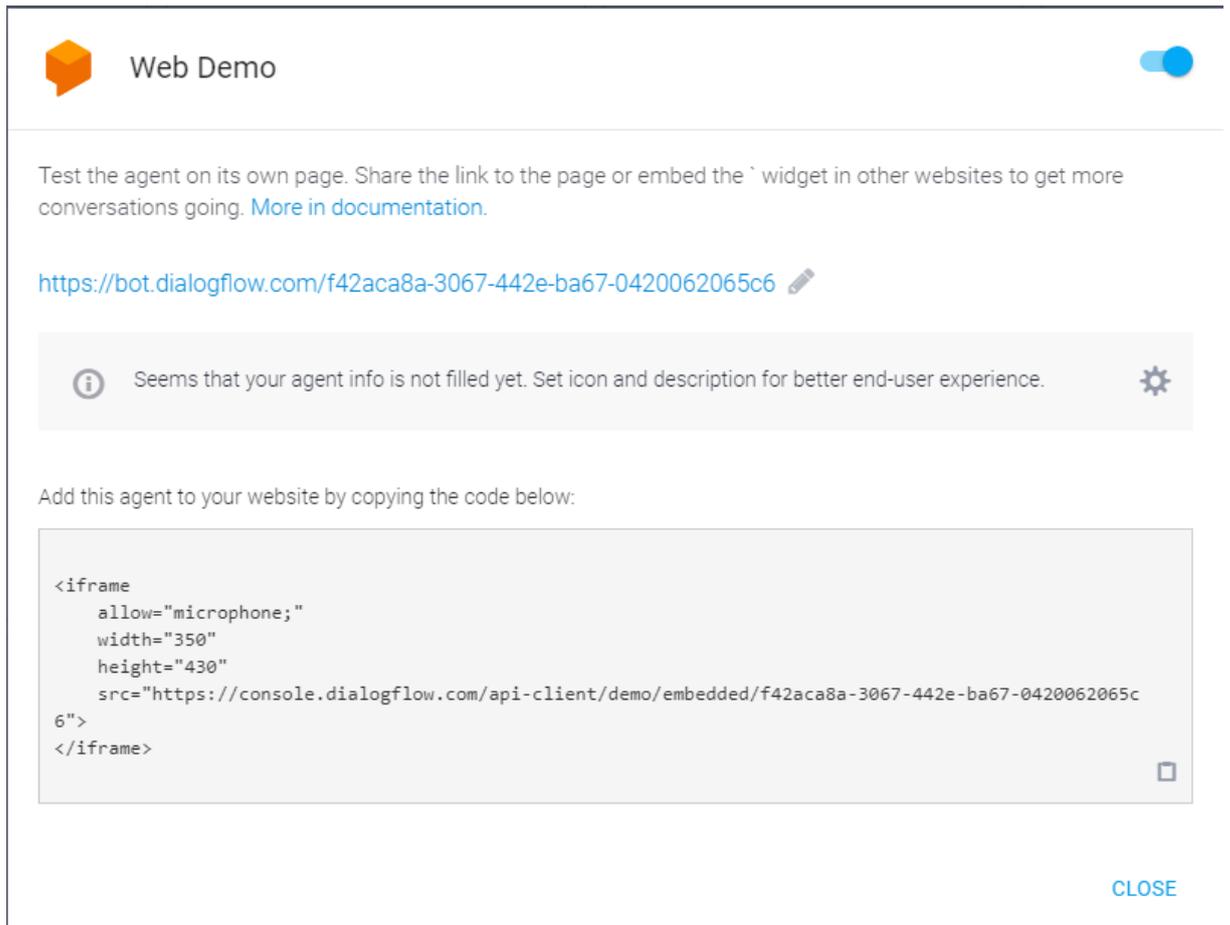
El resultado deseado es un chatbot que permita brindar respuestas rápidas y coherentes a los visitantes del sitio web, que sirva de guía para el visitante, cuando éste pregunte por productos o servicios de la empresa.

El proyecto no tendrá ningún tipo de costo para la empresa, ya que se trata de un proyecto de titulación. La empresa colaborara con toda información que sea necesaria, así como herramientas ya existentes y facilidades de integración con las mismas.

4.1.1.2 Trial

Para que el cliente tenga una visión más clara del funcionamiento del chatbot, se va a realizar un breve desarrollo en la plataforma DIALOGFLOW, que permite crear chatbots de manera rápida y sin costo.

En DIALOGFLOW, se crea un proyecto, llamado **ContactBot** y se habilita las opciones de integración WEB. DIALOGFLOW genera un link, de una dirección URL (*Uniform Resource Locator*), para la demostración.



Web Demo

Test the agent on its own page. Share the link to the page or embed the widget in other websites to get more conversations going. [More in documentation.](#)

<https://bot.dialogflow.com/f42aca8a-3067-442e-ba67-0420062065c6>

Seems that your agent info is not filled yet. Set icon and description for better end-user experience.

Add this agent to your website by copying the code below:

```
<iframe
  allow="microphone;"
  width="350"
  height="430"
  src="https://console.dialogflow.com/api-client/demo/embedded/f42aca8a-3067-442e-ba67-0420062065c6">
</iframe>
```

CLOSE

Figura 11 Web Demo Dialogflow (Google)

Se ingresa al link:

<https://bot.dialogflow.com/f42aca8a-3067-442e-ba67-0420062065c6>

En el cual podemos observar el chatbot en funcionamiento. Se realizan varias pruebas con saludos y frases sencillas y se obtiene las primeras respuestas del chatbot. Este demo se le presenta al cliente para que tenga una mejor percepción de cómo funcionaría el chatbot.

Se entrena al chatbot con un saludo inicial, el cual se responderá cuando el visitante de la página escriba un saludo como "hola", "buen día", "cómo estan", entre otros.

Cabe mencionar que el chatbot resultado de este proyecto se debe integrar al sitio web del cliente y a su plataforma de comunicaciones existente, es decir la interfaz gráfica de interacción con el visitante es la del cliente y no la de DIALOGFLOW.

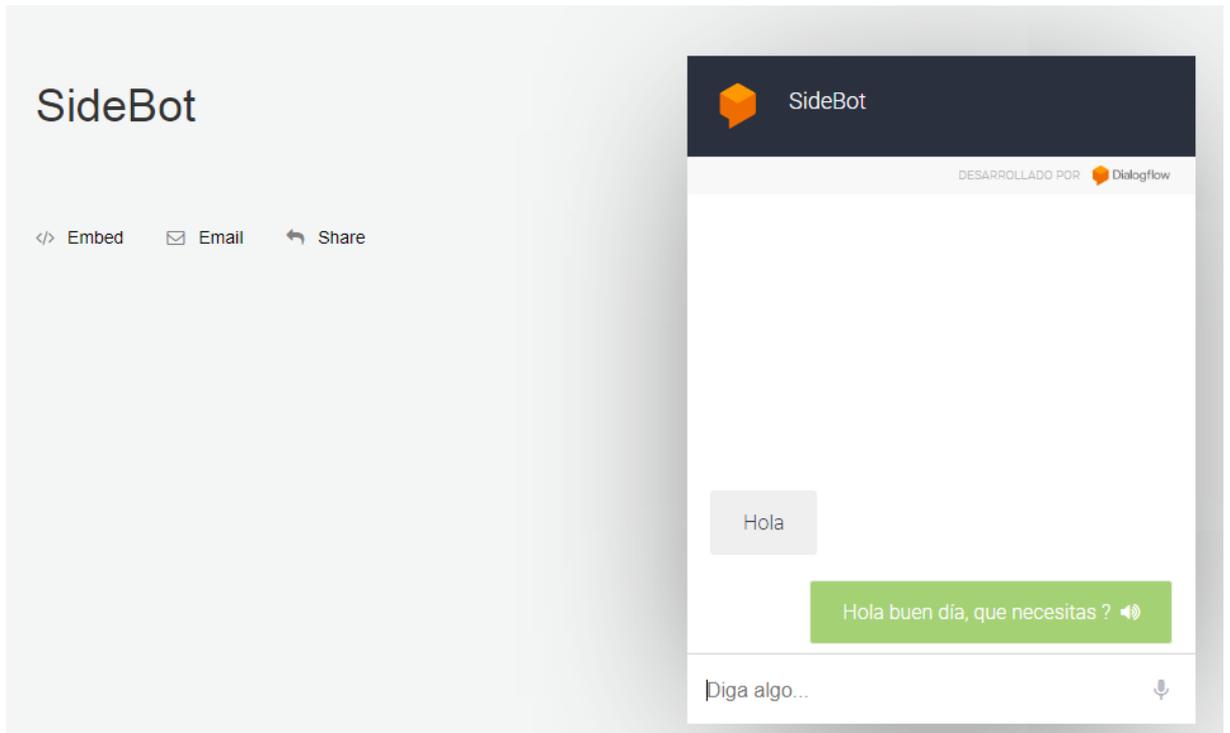


Figura 12 Demo de conversación (Google)

4.1.1.3 Visión de la empresa

Ser una empresa reconocida, distinguida, renombrada y demandante, en el área del software de comunicaciones (Telefonía, Contact Centers, Comunicaciones Unificadas, Social Bussines, entre otras), gracias a la buena reputación y calidad de sus productos y servicios, logrando así enfrentar mercados internacionales.

4.1.1.4 Misión de la empresa

SIDEVOX es una empresa de innovación de software de comunicaciones, orientada desarrollar productos y servicios que permitan una mejor y nueva experiencia de comunicación entre los miembros de la empresa u organización, clientes, proveedores y otros actores de importancia para la misma y adicionalmente que dichos actores accedan y utilicen nuevos y mejores servicios de comunicación.

4.1.1.5 Declaración de la visión del producto

SIDEVOX S.A. es una PYME, que desarrolla software de comunicaciones y gestión empresarial, quiere incorporar herramientas de inteligencia artificial en su sitio web para brindar servicio y mejorar su imagen corporativa a nivel nacional y posteriormente internacional.

Visión: Desarrollar un chatbot que brinde atención inmediata a sus visitantes, mediante mensajería instantánea, que sea capaz de establecer una conversación coherente y guiar al visitante a través de los productos y servicios que brinda la empresa, transmitiendo calidad de servicio e innovación.

4.1.2 Roles

Los roles están definidos en el capítulo 3 sección 3.1

4.1.3 Backlog de proyecto

Los requisitos para poder cumplir con el desarrollo del chatbot son los siguientes:

Proyecto inicial

REQUISITOS	TIEMPO ESTIMADO
Crear un nuevo proyecto en DIALOGFLOW	1 DIA

Entrenamiento de corpus de servicio

REQUISITOS	TIEMPO ESTIMADO
Desarrollar un corpus de entrenamiento de saludos, bienvenida, despedida y obtención de datos del visitante	1 SEMANA
Entrenar el corpus	1 SEMANA

Entrenamiento de corpus técnico

REQUISITOS	TIEMPO ESTIMADO
Análisis del sitio WEB que contiene información de los productos y servicios	1 SEMANA
Obtener información adicional de productos o servicios, que no consten en el sitio web	1 SEMANA
Desarrollar un corpus de entrenamiento de preguntas y respuestas	1 SEMANA
Entrenar el corpus	1 SEMANA

Contextos

REQUISITOS	TIEMPO ESTIMADO
Diseñar un diagrama de flujo de la conversación	3 SEMANAS
Verificar posibles variantes del flujo	1 SEMANA

Errores

REQUISITOS	TIEMPO ESTIMADO
Desarrollar el corpus con errores para frases mal escritas o con faltas de ortografía	3 SEMANAS
Entrenar el corpus	1 SEMANA

Visitantes no deseados

REQUISITOS	TIEMPO ESTIMADO
Desarrollar un corpus de entrenamiento de frases hostiles o soeces	3 SEMANAS
Entrenar el corpus	1 SEMANA

Entrenamiento de corpus complementario

REQUISITOS	TIEMPO ESTIMADO
Desarrollar un corpus de entrenamiento de frases complementarias cortas	3 SEMANAS
Entrenar el corpus	1 SEMANA

Integración

REQUISITOS	TIEMPO ESTIMADO
Leer los mensajes de la conversación	1 SEMANA
Generar respuestas	1 SEMANA

4.2 Proceso de planificación y estimación

4.2.1 Historias de usuarios

Las historias de usuarios permiten identificar la funcionalidad esperada desde el punto de vista del usuario final. Una historia indica tres cosas ¿Quién, qué y por qué?

A continuación se enlistan las historias de los usuarios, que servirán de base para la creación de tareas.

Como visitante debería:

- Tener el servicio de un asistente robot que responda el chat de manera inmediata a cualquier hora.
- Tener respuestas coherentes.
- Tener una asesoría de productos y servicios.
- Tener respuestas de la empresa.
- Tener respuestas concretas a preguntas.

Como dueño del sitio debería

- Atender de manera inmediata a un *INTENT* de comunicación escrita en el sitio web.
- Brindar una asesoría de los productos y servicios de la empresa.
- Captar datos del visitante del sitio.
- Guardar los datos del visitante en una base de datos.

4.2.2 Lista de tareas

Una vez detalladas las historias y las prioridades del *backlog*, se procede a dividir el proyecto en tareas.

Creación del proyecto

- Crear una cuenta de GOOGLE.
- Crear un proyecto en DIALOGFLOW.

Entrenamiento de corpus de servicio

- Redactar el corpus de saludos.
- Redactar el corpus de despedidas.
- Redactar el corpus de información general de la empresa.
- Entrenar el corpus de saludos.

- Entrenar el corpus de despedidas.
- Entrenar el corpus de información general de la empresa.
- Probar el corpus de saludos.
- Probar el corpus de despedidas.
- Probar el corpus de información general de la empresa.

Entrenamiento de corpus de técnico

- Identificar los productos y servicios.
- Redactar el corpus de productos.
- Redactar el corpus de servicios.
- Entrenar el corpus de productos.
- Entrenar el corpus de servicios.
- Probar el corpus de productos.
- Probar el corpus de servicios.

Contexto de la conversación

- Desarrollar contexto de la conversación.
- Diagramar los contextos
- Analizar frases fuera de contexto.

Errores de la conversación

- Redactar el corpus de errores de conversación.
- Entrenar el corpus de errores de conversación.
- Probar el corpus de errores de conversación.

Detección de visitantes no deseados

- Redactar el corpus de errores de conversación.
- Entrenar el corpus de errores de conversación.
- Probar el corpus de errores de conversación.

Corpus Avanzado

- Redactar el corpus de frases más elaboradas.
- Entrenar el corpus de frases más elaboradas.
- Probar el corpus frases más elaboradas.

Integración

- Diagramar el esquema de integración
- Incorporar el chatbot con la plataforma existente.

4.3 Proceso de creación del chatbot de ventas

Basados en el proceso de planificación y estimación definidas en la sección anterior, se procede a la realización de todas las tareas.

4.3.1 Creación del proyecto

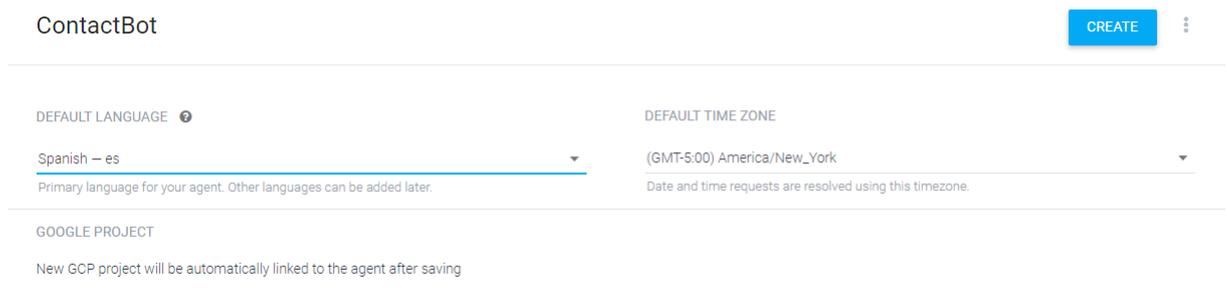
La empresa SIDEVOX ya posee creada una cuenta en GOOGLE, por tal motivo, se va a utilizar dicha cuenta para la creación del proyecto en DIALOGFLOW.

Para ingresar a la consola de DIALOGFLOW se ingresa a la siguiente dirección:

<https://dialogflow.cloud.google.com/>

En pantalla se observará la consola de administración. Elegimos del menú la opción **Crear nuevo Agente**.

El nuevo agente tendrá de nombre **ContactBot** y será entrenado para el lenguaje español.



The screenshot shows the Google Dialogflow console interface for creating a new agent. At the top, the agent name 'ContactBot' is displayed next to a blue 'CREATE' button and a vertical ellipsis menu icon. Below this, there are two main configuration sections: 'DEFAULT LANGUAGE' and 'DEFAULT TIME ZONE'. The 'DEFAULT LANGUAGE' section shows 'Spanish - es' selected, with a note that it is the primary language and other languages can be added later. The 'DEFAULT TIME ZONE' section shows '(GMT-5:00) America/New_York' selected, with a note that date and time requests are resolved using this timezone. At the bottom, the 'GOOGLE PROJECT' section indicates that a new GCP project will be automatically linked to the agent after saving.

Figura 13 Creación de nuevo Agente (Google)

4.3.2 Desarrollo de corpus

No existe un corpus entrenado de acuerdo a la declaración del *business case*, por tal motivo, se debe redactar un corpus que se entrenará en el chatbot. A continuación se detalla una parte del corpus general del chatbot.

Para los saludos se toma en consideración la forma de expresión de las personas del Ecuador principalmente, sin embargo también se debe considerar que la empresa desea que personas de otros lugares externos al Ecuador puedan establecer contacto con la empresa. Así tenemos algunas frases esperadas:

- Hola
- Buen día
- Buenos días
- Buenas tardes

- Buena tarde
- Buenas noches
- Hola cómo están
- Buenas
- Cómo están
- Como han pasado

Las respuestas que el chatbot deberá generar son:

- Gracias por comunicarse con nuestra empresa. Mi nombre es ContactBot y seré su asistente durante su visita en nuestro sitio web ¿En qué podemos servirle?
- Gracias por su comunicación. Mi nombre es ContactBot, esperamos que su visita sea placentera, yo seré su asistente virtual durante la misma. ¿En qué podemos servirle?
- Gracias por visitar nuestro sitio web. Soy ContactBot y estaré gustoso de asistirle. ¿Tiene alguna pregunta o requerimiento en especial?

Para las despedidas se toma en consideración la forma de expresión de las personas del Ecuador principalmente, sin embargo también se debe considerar que la empresa desea que personas de otros lugares externos al Ecuador puedan establecer contacto con la empresa. Así tenemos algunas frases esperadas:

- A sido un placer
- adiós
- nos vemos luego
- bye bye
- Hasta otro momento
- gracias hasta luego
- gracias por la información

Las respuestas que el chatbot deberá generar son:

- fue un placer atenderlo, hasta pronto.
- ha sido un gusto poder acompañarle en su visita, espero atenderlo pronto.
- hasta pronto, que tenga muchos éxitos.

Para el entrenamiento de datos de la empresa, se toma en consideración las secciones de la página web, que tienen información de la misma, la misión, visión, valores, experiencia. Se espera que el visitante realice las siguientes preguntas:

- ¿Cuántos años están en el mercado?
- ¿A qué se dedican?
- ¿Cuáles son sus servicios?
- ¿De donde son?
- ¿Quiénes son Uds.?
- ¿Qué servicios ofrecen?
- ¿En dónde están ubicados?
- ¿Qué hacen?
- ¿Cuántos años de experiencia tienen?

Las respuestas que el chatbot deberá generar son:

- Gracias por tu pregunta, debo indicar que somos una empresa de R&D y llevamos 11 años en el mercado, posicionando nuestros productos y servicios en 4 países en la actualidad.
- Gracias por su comunicación, déjame contarte que SIDEVOX es una empresa de R&D y llevamos 11 años presentes en el mercado. Nuestros productos y servicios ahora están presentes en 4 países.
- Gracias por preguntar. Permíteme describir brevemente a nuestra empresa. SIDEVOX es una empresa de R&D y estamos trabajando desde 11 años y en la actualidad estamos presentes en el mercado nacional e internacional.

El corpus de técnico, consta de productos y servicios. Se considera que el visitante preguntará por los productos y servicios que ofrece la empresa. Esta información se la encuentra en el sitio web. La empresa, SIDEVOX S.A. comercializa varios tipos de productos.

Contact-UCS: este sistema permite la comunicación telefónica interna y externa, comunicaciones de video y chat entre los usuarios de la organización. Se esperan frases como:

- ¿Uds. venden telefonía IP?
- ¿Tienen sistemas de comunicaciones unificadas?
- ¿Tienen centrales telefónicas?
- ¿Estoy interesado en comprar una central?
- ¿Me pueden asesorar, quiero telefonía?
- ¿Deseo una PBX?
- ¿Tengo una central Panasonic y quiero cambiar a central IP?

- ¿Quiero cambiar mi central IP?
- ¿Tengo una central analógica?
- ¿Uds. venden Grandstream?
- ¿Que tienen de solución en telefonía?
- ¿Venden Panasonic?
- ¿Qué tipo de centrales tienen?
- ¿Deseo un sistema de voip?
- ¿Quiero integrar mi sistema de telefonía?
- ¿Deseo un software de voip?
- ¿Venden Isabel?
- ¿Me pueden ayudar con un sistema Elastix?
- ¿Quiero cambiar mi sistema Elastix por uno nuevo?

Las respuestas que el chatbot deberá generar son:

- Nuestra empresa es especialista en sistemas de comunicaciones unificadas, te podemos ofrecer nuestro sistema de telefonía IP CONTACTVOX UCS
- Permíteme presentarte nuestro sistema de comunicaciones CONTACTVOX UCS
- Somos especialistas en telefonía IP, podemos recomendarte, nuestro sistema de comunicaciones CONTACTVOX UCS
- Con mucho gusto podemos ofrecerte el mejor sistema de comunicaciones telefónicas CONTACTVOX UCS

Contact-CC: Este sistema permite la comunicación telefónica interna y externa, principalmente orientados a centro de atención al cliente.

Se esperan frases como:

- ¿Uds. venden callcenter?
- ¿Tienen sistemas de callcenter?
- ¿Tienen call centers?
- ¿Estoy interesado en comprar un callcenter?
- ¿Me pueden asesorar, quiero instalar un callcenter?
- ¿Deseo una contactcenter?
- ¿Tengo un sistema de contactcenter y quiero cambiarlo?
- ¿Quiero cambiar mi contact center?
- ¿Tengo un contactcenter antiguo?
- ¿Uds. venden Génesis?

- ¿Que tienen de solución en callcenter o contactcenter?
- ¿Venden software para callcenter?
- ¿Qué tipo de callcenters tienen?
- ¿Deseo un sistema de call contact center?
- ¿Quiero integrar mi sistema de callcenter?
- ¿Deseo un software de contactcenter?
- ¿Venden Génesis?
- ¿Me pueden ayudar con un sistema Génesis?
- ¿Quiero cambiar mi sistema Génesis por uno más económico?

Las respuestas que el chatbot deberá generar son:

- Nuestra empresa es especialista en sistemas de contactcenter, te podemos ofrecer nuestro sistema CONTACTVOX CC
- Permíteme presentarte nuestro sistema de comunicaciones CONTACTVOX CC.
- Somos especialistas en contactcenter, podemos recomendarte, nuestro sistema de comunicaciones CONTACTVOX CC
- Con mucho gusto podemos ofrecerte el mejor sistema de contactcenter CONTACTVOX CC

Contact-CRM: Este sistema permite la gestión de los clientes de una organización. El CRM (*Customer Relationship Management*) tiene varios módulos orientados a venta y postventa. Se puede utilizar en muchos giros de negocios y posee características especiales para control de tareas, calendarios y documentos.

Se esperan frases como:

- ¿Uds. venden crm?
- ¿Tienen sistemas de crm?
- ¿Tienen crm?
- ¿Estoy interesado en comprar un crm?
- ¿Me pueden asesorar, quiero instalar un crm?
- ¿Deseo un crm?
- ¿Tengo un sistema de crm y quiero cambiarlo?
- ¿Quiero cambiar mi crm?
- ¿Tengo un crm y quiero potencializarlo?
- ¿Uds. venden Sales Force?
- ¿Que tienen de solución en crm para ventas?

- ¿Venden software para ventas y cobranzas?
- ¿Qué tipo de crm tienen?
- ¿Deseo un sistema de cobranzas?
- ¿Deseo un sistema de gestión de atención al cliente?
- ¿Deseo gestionar mi atención al cliente?
- ¿Quiero integrar mi ERP con un sistema de ventas?
- ¿Deseo un software de ventas?
- ¿Venden Sales Force?
- ¿Me pueden ayudar con un sistema Microsoft Dynamics?
- ¿Quiero cambiar mi sistema Dynamics por uno más económico?

Las respuestas que el chatbot deberá generar son:

- Nuestra empresa es especialista en sistemas de crm para venta y postventa, te podemos ofrecer nuestro sistema CONTACTVOX CRM
- Permíteme presentarte nuestro sistema de crm CONTACTVOX CRM
- Somos especialistas en sistemas de gestión empresarial, podemos recomendarte, nuestro sistema CONTACTVOX CRM
- Con mucho gusto podemos ofrecerte el mejor sistema de gestión de ventas y postventa CONTACTVOX CRM

Contact-BI: este software permite organizar los datos de los otros sistemas como CRM, CALLCENTER, ERP y construir la *data warehouse*, *datamart* y la capa de presentación.

Se esperan frases como:

- ¿Uds. Venden BI?
- ¿Tienen sistemas de BI?
- ¿Tienen *business intelligence*?
- ¿Estoy interesado en comprar un *business intelligence*?
- ¿Me pueden asesorar, quiero instalar un BI?
- ¿Deseo un BI?
- ¿Tengo un sistema de reportes en Excel y quiero cambiarlo?
- ¿Quiero cambiar mis reportes?
- ¿Tengo un BI y quiero potencializarlo?
- ¿Uds. venden Power BI?
- ¿Que tienen de solución en *business intelligence*?
- ¿Venden software para reportería y estadísticas?

- ¿Qué tipo de *business intelligence* tienen?
- ¿Deseo un sistema de *business intelligence*?
- ¿Deseo un sistema de reportes?
- ¿Deseo gestionar mis reportes de manera eficiente?
- ¿Quiero integrar mis datos con un sistema de BI o reportes?
- ¿Deseo un software de *business intelligence*?
- ¿Venden Qlik Sense?
- ¿Me pueden ayudar con un sistema Power BI?
- ¿Quiero cambiar mi sistema Power BI?

Las respuestas que el chatbot deberá generar son:

- Nuestra empresa es especialista en sistemas de *business intelligence* que permite integrar tus datos y mostrarlos en una capa de presentación intuitiva
- Permíteme presentarte nuestro sistema de BI CONTACTVOX BI
- Somos especialistas en sistemas de *business intelligence*, podemos recomendarte, nuestro sistema CONTACTVOX BI

Contact-Video: sistema orientado a las grandes organizaciones tanto públicas como privadas. Permite establecer comunicaciones a distancia ahorrando costos de viajes y movilizaciones.

Se esperan frases como:

- ¿Uds. venden video conferencia?
- ¿Tienen sistemas de video?
- ¿Tienen colaboración?
- ¿Estoy interesado en comprar un sistema de video conferencia?
- ¿Me pueden asesorar, quiero instalar una videoconferencia?
- ¿Deseo un sistema de colaboración?
- ¿Tengo un sistema de video y quiero cambiarlo?
- ¿Quiero cambiar mi sistema de video por uno mejor?
- ¿Tengo un sistema Polycon y quiero actualizarlo?
- ¿Uds. venden Webex?
- ¿Que tienen de solución en video conferencia?
- ¿Venden software para colaboración y video conferencia?
- ¿Qué tipo de sistemas de video tienen?
- ¿Deseo un software de video conferencia?

- ¿Deseo un sistema de colaboración y video?
- ¿Deseo establecer comunicación remota de video con mis socios?
- ¿Quiero crear salas de video para colaboración?
- ¿Deseo un software de teleconferencia?
- ¿Venden teleconferencia?
- ¿Me pueden ayudar con un sistema de video conferencia?
- ¿Quiero cambiar mi sistema Webex?

Las respuestas que el chatbot deberá generar son:

- Nuestra empresa es especialista en sistemas de video conferencia que permite integrar realizar comunicaciones a distancia con alta definición
- Permíteme presentarte nuestros sistema de BI CONTACTVOX VIDEO
- Somos especialistas en sistemas de video conferencia, podemos recomendarte, nuestro sistema CONTACTVOX VIDEO

Contact-Mobile: Es un sistema de comunicaciones telefónicas en la nube. Este sistema está orientado a las empresas PYMES, ya que les brinda las ventajas de un sistema tradicional de PBX y al mismo tiempo les permite ahorrar recursos en infraestructura, instalación y mantenimiento.

Este producto es de similares características al de CONTACTVOX UCS, por tal motivo, el entrenamiento del producto mencionado va a servir para este nuevo producto. Sin embargo al ser dos productos diferentes, la distinción de los mismos se la deberá realizar, al establecer un contexto en la conversación. Los contextos están desarrollados más adelante.

Contact-Matic: Es un sistema de publicidad masiva telefónica, que sirve para transmitir mensaje de audio a través de llamadas de carácter informativo y publicitario. Utiliza marcadores automáticos y tecnología TTS.

Se esperan frases como:

- ¿Uds. venden sistemas de telemarketingo?
- ¿Tienen sistemas de tele marketingo?
- ¿Tienen envío masivo de llamadas?
- ¿Estoy interesado en comprar un sistema de envío masivo de mensajes y llamadas?
- ¿Me pueden asesorar, quiero instalar un software de marcación automática?
- ¿Deseo un sistema de marcación automática?

- ¿Tengo un sistema de marcación y quiero mejorarlos?
- ¿Quiero hacer llamadas masivas?
- ¿Quiero enviar mensajes masivos?
- ¿Uds. venden sistemas de marcadores automátcos?
- ¿Que tienen de solución de text to speech?
- ¿Venden software para llamadas con texto speech?
- ¿Qué tipo de sistemas de TTS tienen?
- ¿Deseo un software de marcación con TTS?
- ¿Deseo un sistema de texto speech y envío masivo de mensajes?
- ¿Deseo establecer comunicación con mis clientes de manera masiva?
- ¿Quiero crear campañas de marcación automática?
- ¿Deseo un software de telemarketing?
- ¿Venden envío de mensajes masivos?
- ¿Me pueden ayudar con un sistema de llamadas y mensajes?
- ¿Quiero cambiar mi sistema de telemarketing?

Las respuestas que el chatbot deberá generar son:

- Nuestra empresa es especialista en sistemas de tele marketing que permite realizar llamadas automáticas y puede incluir tecnología TTS
- Permíteme presentarte nuestro sistema de marcación automática CONTACTVOX CALLMATIC
- Somos especialistas en sistemas de marcación automática para campañas de marketing, podemos recomendarte, nuestro sistema CONTACTVOX CALLMATIC

Contact-Web: Es la nueva forma de comunicación a través de Internet. Nuevos canales de datos, voz y video que permiten el intercambio de información de manera rápida y oportuna. Incorpora chatbots web para atención rápida y oportuna.

Se esperan frases como:

- ¿Uds. venden sistemas de chatbots?
- ¿Tienen chatbots?
- ¿Tienen llamadas web?
- ¿Estoy interesado en comprar un sistema de chat web?
- ¿Me pueden asesorar, quiero bajar costos de 1800?
- ¿Deseo un sistema de comunicación web?
- ¿Tengo un chat en mi página y quiero mejorarlo?

- ¿Quiero que mis clientes se contacten por la web?
- ¿Quiero que me envíen mensajes por la web?
- ¿Uds. venden sistemas de chat?
- ¿Que tienen de solución de chat?
- ¿Venden software para chatweb?
- ¿Qué tipo de chats tienen?
- ¿Deseo un software de llamadas web?
- ¿Deseo un sistema de chatweb y llamadas web?
- ¿Deseo que mis clientes me llame por mi página web?
- ¿Quiero hablar con mis clientes por la web?
- ¿Deseo un software de de bots?
- ¿Venden chatbots?
- ¿Me pueden ayudar con un sistema de chatweb?
- ¿Quiero cambiar mi 1800?
- ¿Necesito una línea de respaldo del 1800?
- ¿Venden troncales SIP?
- Tienen SIP trunks?
- ¿Tienen webrtc?
- ¿Quiero llamadas webrtc?

Las respuestas que el chatbot deberá generar son:

- Nuestra empresa es especialista en sistemas web, a través de tecnología WebRtc, sus clientes podrán comunicarse con su empresa por su página web a través de llamadas o chat y serán atendidas por un chatbot
- Permíteme presentarte nuestro sistema de comunicación web CONTACTVOX CONTACTWEB, un sistema que incorpora WebRtc e Inteligencia Artificial
- Somos especialistas en sistemas de comunicación, podemos recomendarte, nuestro sistema CONTACTVOX CONTACTWEB, un sistema de vanguardia con inteligencia artificial

Adicionalmente existirán en muchas ocasiones visitantes que no tenga la más mínima intención de obtener información y empiecen a jugar con el chatbot, incluso transmitiendo frases insultantes y soeces. Para esto se entrenará un corpus de frases no deseadas:

- eres un tonto
- robot imbécil

- que idiota
- no sirves para nada

Así entrenamos este corpus y la respuesta del chatbot será:

- Creo que no deberías expresarte de esa manera, lo siento pero creo que no puedo atenderte
- Me gustaría ayudarte, pero pienso que tu intención es la de molestar, adiós

Finalmente, cuando el chatbot no responda como debe o no entienda, es muy probable que el usuario escriba frases como:

- no me entiendes
- no me ayudas
- me estás malentendiendo las cosas

Así entrenamos este corpus y la respuesta del chatbot será:

- Disculpa aún estoy aprendiendo, por eso te solicito que seas muy claro en tus respuestas, por favor
- Por favor, puedes volver a escribir, estoy en proceso de aprendizaje

Es muy importante que ante frases que el chatbot no entienda, explique que se encuentra en proceso de aprendizaje para no transmitir una mala impresión al usuario y evitar que éste quiera hablar de manera inmediata con un ser humano perdiendo confianza en el chatbot.

4.3.3 Entrenamiento con DIALOGFLOW

Para entrenar el corpus en DIALOGFLOW, se deben crear *INTENTS*, con las frases antes enlistadas y generar una respuesta a dichas frases. Se crean los siguientes *INTENTS*:

- **Default Welcome INTENT:** entrenamiento de saludos
- **Despedidas:** entrenamiento de despedidas
- **Datos de la empresa:** entrenamiento de datos de la empresa
- **Comunicaciones Unificadas:** entrenamiento de producto Contact-UCS
- **Contactcenter:** entrenamiento del producto Contact-CC
- **Contactcrm:** entrenamiento del producto Contact-CRM
- **Contactbi:** entrenamiento del producto Contact-BI
- **Contactvideo:** entrenamiento del producto Contact-Video

- **Contactweb:** entrenamiento del producto Contact-Web
- **Contactmatic:** entrenamiento del producto Contact-Matic
- **Visitantes no deseados:** entrenamiento de frases soeces y de insultos
- **Errores:** entrenamiento de frases de rechazo

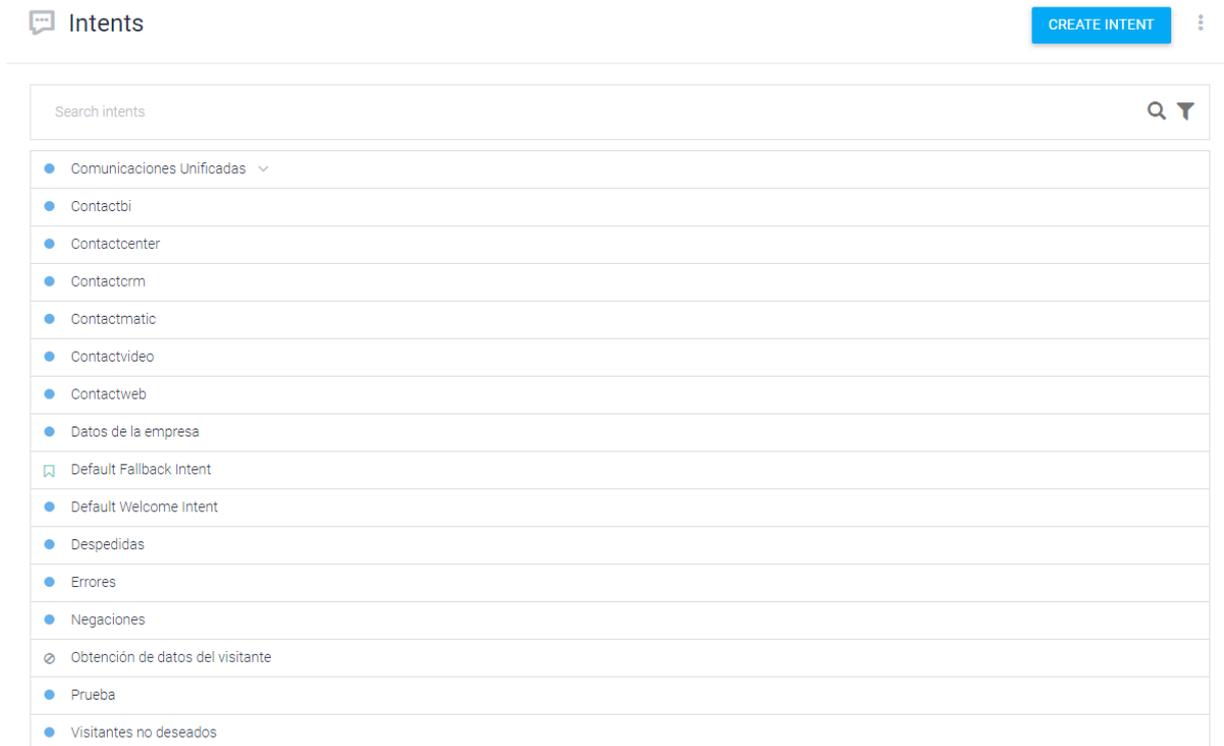


Figura 14 INTENTS en DIALOGFLOW (Google)

4.3.4 Entrenamiento con KERAS

El entrenamiento con los *INTENTS* de DIALOGFLOW, facilitan el procesamiento del lenguaje natural, sin embargo para evitar ambigüedades se requiere utilizar otra librería que permita realizar un entrenamiento similar en caso de nuevas expresiones o para el tratamiento de un corpus de en el cual las ideas no se expongan de manera clara. Este corpus no lo vamos a entrenar con DIALOGFLOW, vamos a introducir un recurso más complejo pero que puede dar cierta flexibilidad cuando se trate de corpus extensos. Para esto se desarrolla un corpus complementario con frases que el visitante pueda tener, sin embargo se entrenará un corpus limitado que sirva como ejemplo de entrenamiento.

El entrenamiento se va a realizar entrenando una red neuronal, se utiliza el *framework* KERAS, en base a TENSORFLOW. Este ejemplo se basa en un proyecto de GITHUB KERAS EXAMPLES. (GITHUB)

A continuación se enlistan algunas frases que serán entrenadas:

- Hola, lo que pasa es que mi central se quemó y no se que será
- Verá, lo que pasa es que no he podido configurar mi central telefónica
- No, se si me pueda ayudar, lo que pasa es que no tengo telefonía
- Verá yo no puedo llamar pero no se que pasa, ud me puede ayudar
- Lo que pasa es que yo compre un portero ip y quiero integrarlo con mi central

4.3.5 Corpus complementario

```
from __future__ import print_function

from keras.models import Model
from keras.layers import Input, LSTM, Dense
import numpy as np

batch_size = 64 # Batch size for training.
epochs = 100 # Number of epochs to train for.
latent_dim = 256 # Latent dimensionality of the encoding space.
num_samples = 200 # Number of samples to train on.
# Path to the data txt file on disk.
data_path = 'contactvox.txt'
```

Figura 15 Entrenamiento Keras I (Google)

Creamos un archivo de texto con las frases y respuestas a entrenar. Se realiza la lectura del archivo de texto, que contiene el corpus a entrenar y se crean los arreglos de entrada y salida del texto

```

▶ # Vectorize the data.
input_texts = []
target_texts = []
input_characters = set()
target_characters = set()
with open(data_path, 'r', encoding='utf-8') as f:
    lines = f.read().split('\n')
for line in lines[: min(num_samples, len(lines) - 1)]:
    input_text, target_text = line.split('\t')
    # We use "tab" as the "start sequence" character
    # for the targets, and "\n" as "end sequence" character.
    target_text = '\t' + target_text + '\n'
    input_texts.append(input_text)
    target_texts.append(target_text)
    for char in input_text:
        if char not in input_characters:
            input_characters.add(char)
    for char in target_text:
        if char not in target_characters:
            target_characters.add(char)

input_characters = sorted(list(input_characters))
target_characters = sorted(list(target_characters))
num_encoder_tokens = len(input_characters)
num_decoder_tokens = len(target_characters)
max_encoder_seq_length = max([len(txt) for txt in input_texts])
max_decoder_seq_length = max([len(txt) for txt in target_texts])

print('Number of samples:', len(input_texts))
print('Number of unique input tokens:', num_encoder_tokens)
print('Number of unique output tokens:', num_decoder_tokens)
print('Max sequence length for inputs:', max_encoder_seq_length)
print('Max sequence length for outputs:', max_decoder_seq_length)

```

Figura 16 Entrenamiento KERAS II (Google)

Se utiliza el modelo seq2seq en los cuales se deben definir encoders y decoders

```

input_token_index = dict(
    [(char, i) for i, char in enumerate(input_characters)])
target_token_index = dict(
    [(char, i) for i, char in enumerate(target_characters)])

encoder_input_data = np.zeros(
    (len(input_texts), max_encoder_seq_length, num_encoder_tokens),
    dtype='float32')
decoder_input_data = np.zeros(
    (len(input_texts), max_decoder_seq_length, num_decoder_tokens),
    dtype='float32')
decoder_target_data = np.zeros(
    (len(input_texts), max_decoder_seq_length, num_decoder_tokens),
    dtype='float32')

for i, (input_text, target_text) in enumerate(zip(input_texts, target_texts)):
    for t, char in enumerate(input_text):
        encoder_input_data[i, t, input_token_index[char]] = 1.
    for t, char in enumerate(target_text):
        # decoder_target_data is ahead of decoder_input_data by one timestep
        decoder_input_data[i, t, target_token_index[char]] = 1.
        if t > 0:
            # decoder_target_data will be ahead by one timestep
            # and will not include the start character.
            decoder_target_data[i, t - 1, target_token_index[char]] = 1.

```

Figura 17 Entrenamiento KERAS III (Google)

Se define el modelo seq2seq, con los parámetros definidos

```

# Define an input sequence and process it.
encoder_inputs = Input(shape=(None, num_encoder_tokens))
encoder = LSTM(latent_dim, return_state=True)
encoder_outputs, state_h, state_c = encoder(encoder_inputs)
# We discard `encoder_outputs` and only keep the states.
encoder_states = [state_h, state_c]

# Set up the decoder, using `encoder_states` as initial state.
decoder_inputs = Input(shape=(None, num_decoder_tokens))
# We set up our decoder to return full output sequences,
# and to return internal states as well. We don't use the
# return states in the training model, but we will use them in inference.
decoder_lstm = LSTM(latent_dim, return_sequences=True, return_state=True)
decoder_outputs, _, _ = decoder_lstm(decoder_inputs,
                                     initial_state=encoder_states)
decoder_dense = Dense(num_decoder_tokens, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)

# Define the model that will turn
# `encoder_input_data` & `decoder_input_data` into `decoder_target_data`
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

# Run training
model.compile(optimizer='rmsprop', loss='categorical_crossentropy')
model.fit([encoder_input_data, decoder_input_data], decoder_target_data,
        batch_size=batch_size,
        epochs=epochs,
        validation_split=0.2)

# Save model
model.save('s2s.h5')

```

Figura 18 Entrenamiento KERAS IV (Google)

Se realiza pruebas de respuesta

```

[ ] message = 'Hola.'
    seq_index = input_texts.index(message)
    input_seq = encoder_input_data[seq_index:seq_index + 1]
    decoded_sentence = decode_sequence(input_seq)
    print('-')
    print('Input sentence:', input_texts[seq_index])
    print('Decoded sentence:', decoded_sentence)

```

```

↳ -
   Input sentence: Hola.
   Decoded sentence: Hola, en qué puedo ayudarte.

```

Figura 19 Pruebas del modelo entrenado con KERAS (Google)

```

↳ Number of samples: 200
   Number of unique input tokens: 40
   Number of unique output tokens: 27
   Max sequence length for inputs: 32
   Max sequence length for outputs: 39
   Train on 160 samples, validate on 40 samples
   Epoch 1/100
   160/160 [=====] - 3s 20ms/step - loss: 2.5403 - val_loss: 2.2344
   Epoch 2/100
   160/160 [=====] - 0s 2ms/step - loss: 2.4354 - val_loss: 1.9555
   Epoch 3/100
   160/160 [=====] - 0s 2ms/step - loss: 2.0456 - val_loss: 1.8901
   Epoch 4/100
   160/160 [=====] - 0s 2ms/step - loss: 1.9959 - val_loss: 1.7688
   Epoch 5/100
   160/160 [=====] - 0s 2ms/step - loss: 1.8851 - val_loss: 1.8145
   Epoch 6/100
   160/160 [=====] - 0s 2ms/step - loss: 1.9155 - val_loss: 1.6708
   Epoch 7/100
   160/160 [=====] - 0s 2ms/step - loss: 1.7678 - val_loss: 1.6053
   Epoch 8/100
   160/160 [=====] - 0s 2ms/step - loss: 1.8400 - val_loss: 1.6764
   Epoch 9/100
   160/160 [=====] - 0s 2ms/step - loss: 1.7541 - val_loss: 1.5558
   Epoch 10/100
   160/160 [=====] - 0s 2ms/step - loss: 1.7286 - val_loss: 1.6075
   Epoch 11/100
   160/160 [=====] - 0s 2ms/step - loss: 1.6813 - val_loss: 1.5150
   Epoch 12/100
   160/160 [=====] - 0s 2ms/step - loss: 1.6395 - val_loss: 1.6427
   Epoch 13/100

```

Figura 20 Entrenamiento KERAS V (Google)

4.3.6 Contextos

El mayor problema a resolver con un chatbot es que el visitante no siga un hilo de conversación. Con la ayuda de contextos podemos establecer un camino controlado por el chatbot, que funciona de manera adecuada si el visitante decide seguir ese camino, sin embargo si el visitante no decide seguir el camino, si hace preguntas fuera del contexto o introduce nuevos elementos a la conversación es importante tener una respuesta adecuada y volver a encaminar al chatbot en el camino correcto.

Con el entrenamiento realizado en la sección anterior, el chatbot va a realizar el procesamiento del lenguaje natural y emitir respuestas de acuerdo a las intenciones de comunicación del visitante. Sin embargo esta característica no es suficiente para poder establecer una conversación entre el chatbot y el visitante. Se requiere que la conversación pueda seguir un lineamiento, que tenga coherencia desde el punto de vista del visitante, no se trata solo de preguntar y responder. Por tal motivo en esta sección se diseñan los contextos de la conversación, con el objetivo de potencializar las respuestas del chatbot.

Como se había analizado en los primeros capítulos, las expresiones humanas son tan complejas y a la vez desordenadas, un visitante podría hacer preguntas y cambiar el tema de la conversación y topa temas no relevantes o sin importancia.

La estrategia es hacer que el chatbot controle la conversación y cierre las posibilidades de preguntas del cliente, para que la conversación tenga fluidez. No se trata de hacer un chatbot que lo sabe todo o que es capaz de responder a todo, se trata de responder coherentemente y de encaminar al visitante dentro de una línea de conversación.

Los contextos de la conversación están diseñados en base a la conexión que tendrán los *INTENTS*.



Figura 21 Contexto inicial

El visitante inicia con un saludo y el chatbot responde con un saludo de bienvenida y haciendo la pregunta: ¿Qué deseas?, ésta pregunta puede dar lugar a ambigüedades, ya que se abre todo un abanico de respuestas. Sin embargo el chatbot de este proyecto va a tratar de ser al máximo ser un agente de conversación y no solo un agente de preguntas y respuestas.

Se espera entonces que el visitante pregunte por alguno de los servicios de la empresa. Si es el caso el chatbot responderá de manera adecuada, si no lo es, existen *INTENTS* de respuesta para tomar el control de la conversación y poder guiar al visitante de mejor manera. Los contexto permiten generar un hilo de conversación, desde el punto de vista de la programación de *DIALOGFLOW*, los contextos se traducen en *INTENTS* anidados. Es decir *INTENTS* dentro de *INTENTS*.

Por ejemplo, si el visitante pregunta por el producto *CONTACTVOX UCS* se define el siguiente contexto:

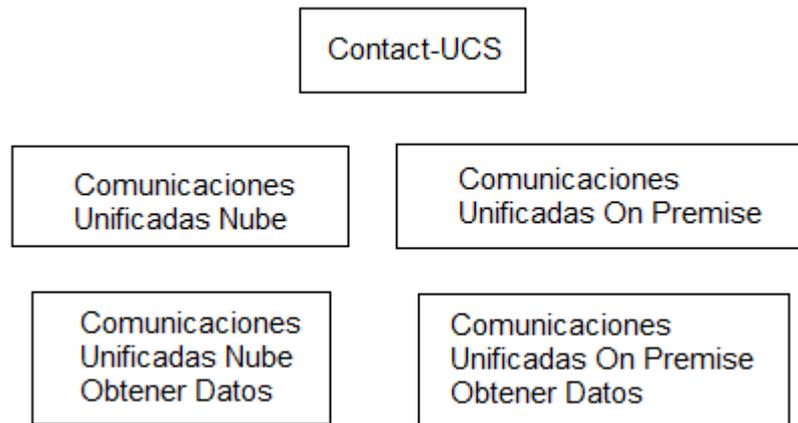


Figura 22 Contextos Contactvox-UCS (Google)

Se crean los *INTENTS*:

- Comunicaciones Unificadas nube
 - Comunicaciones Unificadas nube obtener datos
- Comunicaciones Unificadas on premise
 - Comunicaciones Unificadas nube obtener datos

En cada uno de estos se realizan entrenamientos de corpus que permiten mantener un flujo de conversación y se obtienen datos del negocio y del cliente.



Figura 23 Contextos CONTACTVOX UCS (Google)

Con este contexto se crea un flujo de conversación, entre el cliente y el chatbot. A continuación se diagrama un ejemplo de una conversación basada en este contexto:

visitante	chatbot
<p>hola, me pueden ayudar</p>	<p>Gracias por su comunicación. Mi nombre es ContactBot, esperamos que su visita sea placentera, yo seré su asistente virtual durante la misma. ¿En qué podemos servirle?</p>
<p>¿Tienen sistemas de comunicaciones unificadas?</p>	<p>Somos especialistas en telefonía IP, podemos recomendarte, nuestro sistema de comunicaciones CONTACTVOX UCS</p>
<p>creo que puede ser la nube</p>	<p>Tenemos dos tipos de servicios en la nube y on premise, de ¿cuál de ellas requiere información?</p> <p>nuestro sistema se aloja en la nube de Amazon Ohio</p> <p>Para poder asesorarte de mejor manera necesito conocer algunos datos, ¿Estás de acuerdo ?</p>

Figura 24 Ejemplo de conversación bajo contexto

Como se puede evidenciar en la figura que antecede, el visitante sigue la conversación propuesta por el chatbot, en principio solicitó información del sistema de comunicaciones unificadas y a continuación se empieza a desarrollar el diálogo, incluso se evidencia que el chatbot le solicita una contestación afirmativa y pasar a una toma de datos, que para la parte comercial es muy importante debido a que el visitante puede ser un potencial cliente cumpliendo así con uno de los objetivos propuestos para este chatbot.

Para captar los datos del potencial cliente Creamos un nuevo *INTENT* denominado **Obtención de datos del visitante**. Para obtener los datos del visitante, el chatbot terminó su intervención con preguntas como:

- ¿Para poder brindarle una mejor atención, me puedes ayudar con tus datos?
- ¿Me gustaría conocer tus datos para poder asesorarte de mejor manera, estás de acuerdo?

A lo cual se esperan dos posibles respuestas SI y NO. Si el visitante está de acuerdo se procede a tomar los datos. Las frases a entrenar son las de la respuesta afirmativa

- está bien
- claro
- no hay problema
- claro, sin problema
- por supuesto
- de acuerdo
- si
- si, por supuesto
- si acepto
- acepto
- estoy de acuerdo
- claro que si

• Obtención de datos del visitante SAVE

Contexts ? v

Events ? v

Training phrases ? Search training phrases 🔍 ^

” Add user expression

” esta bien

” claro

” no hay problema

” claro, no hay problema

” por supuesto

” de acuerdo

” si

Figura 25 Entrenamiento de obtención de datos del visitante (Google)

Cuando la respuesta es reconocida como afirmativa, se procede a preguntar por los datos de nombre y correo electrónico. Para esto, definimos las variables como requeridas y entrenamos las frases dentro de la opción de *PROMPTS*

Frases de obtención del nombre:

- ¿Cuál es tu nombre?
- ¿Me puedes indicar tu nombre?
- ¿Cómo te llamas?

Frases de obtención de la dirección del correo electrónico:

- ¿Cuál es la dirección de e-mail?
- ¿Me puedes indicar la dirección de tu e-mail?
- ¿Cómo es la dirección de tu correo electrónico?

Prompts for "name"

NAME	ENTITY	VALUE
name	@sys.any	

PROMPTS	
1	Cuál es tu nombre ?
2	Me puedes indicar tu nombre ?
3	Cómo te llamas ?
4	Enter a prompt variant

CLOSE

Figura 26 Entrenamiento de datos del visitante NOMBRE (Google)

Prompts for "email"

NAME	ENTITY	VALUE
email	@sys.email	

PROMPTS	
1	Cúal es tu dirección de e-mail ?
2	Me puedes indicar tu dirección e-mail ?
3	Cuál es la dirección de tu correo electrónico ?
4	Enter a prompt variant

CLOSE

Figura 27 Entrenamiento de datos del visitante E-MAIL (Google)

Por cada uno de los servicios se desarrolla un modelo de contextos que permite guiar al visitante y al final obtener sus datos

4.3.7 Integraciones

El chatbot deberá integrarse a la infraestructura de comunicaciones que tienen el cliente, en su sitio web. Los mensajes son enviados por el cliente desde una interfaz web, a través de un *websocket* en mensajes SIP.

Los mensajes son procesados por el servidor de señalización SIP SERVER y enviado a una interface de la base de datos, la cual almacena los mensajes y envía al chatbot. El chatbot realiza el procesamiento del lenguaje natural, genera las respuestas y responde con mensajes que el SIP SERVER envía al usuario.

Los mensajes son enviados a DIALOGFLOW, este envío se hace utilizando un API (*Application Programming Interface*) de funciones provista por GOOGLE (Google). DIALOGFLOW, procesará el mensaje, si, el mismo es reconocido por un *INTENT*, se generará la respuesta, de acuerdo al entrenamiento de dicho *INTENT*. Ahora, si la frase no ha sido entrenada, es decir no es reconocida, DIALOFFLOW enviará dicha frase a un *INTENT* denominado *DEFAULT FALLBACK INTENT*. A continuación se reenvía el mensaje hacia otro sistema de procesamiento que fue entrenado con KERAS, para que se emita la respuesta.

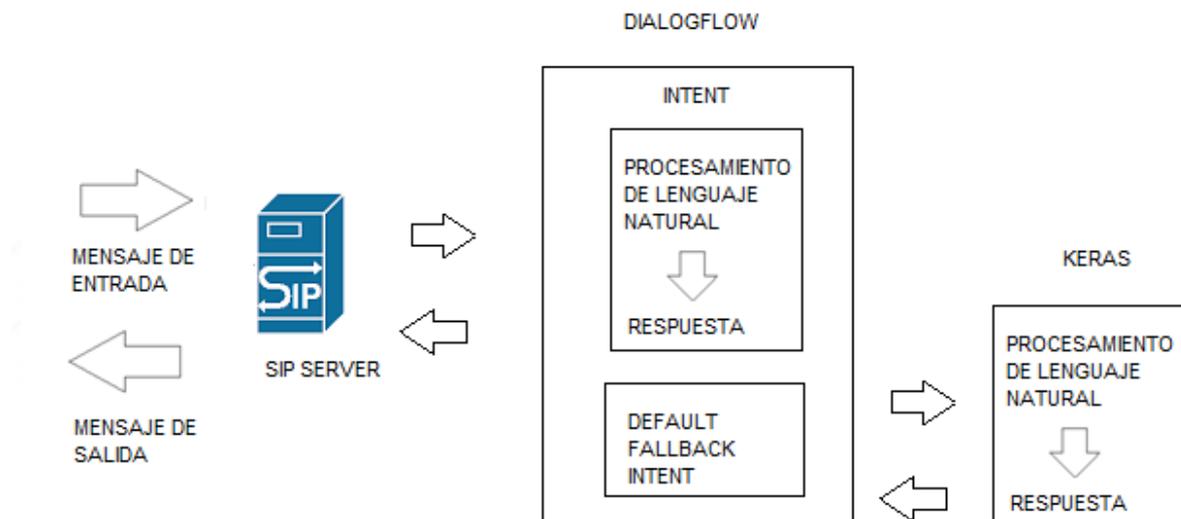


Figura 28 Integración chatbot

4.4 Proceso de Pruebas y Lanzamiento

Se procede a probar el chatbot en la interfaz de DIALOGFLOW, para comprobar algunos flujos de conversación.

En este ejercicio observamos que el visitante sigue el flujo de la conversación, indicando que requiere y compartiendo sus datos e información relevante para la asesoría.

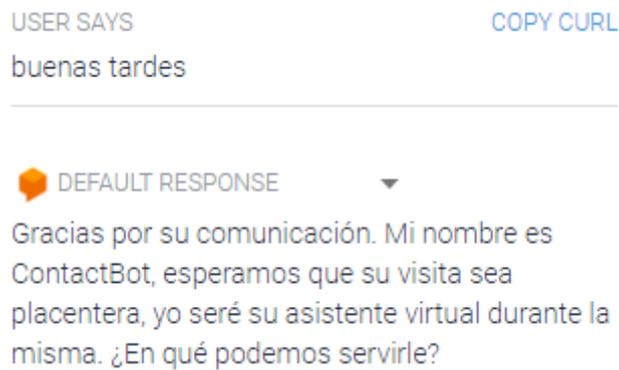


Figura 29 Prueba de Chatbot I (Google)

En este contexto el visitante requiere información de telefonía

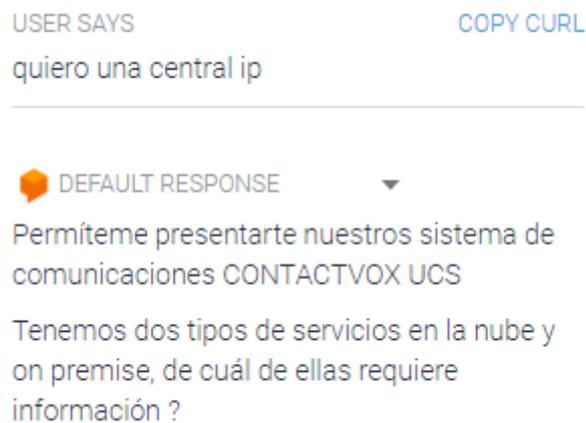


Figura 30 Prueba de Chatbot II (Google)

Las preguntas que vienen a continuación son porque el contexto de la conversación se encaja dentro de los sistemas de telefonía, cuando el visitante acepta responder a las preguntas, estas estarán relacionadas con el contexto de telefonía.

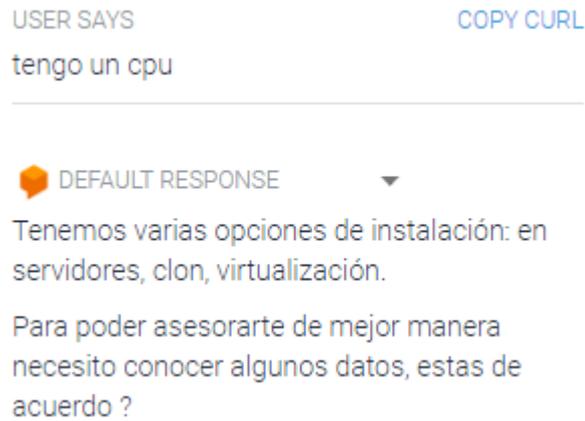


Figura 31 Prueba de Chatbot III (Google)

A partir de este momento el chatbot pregunta al visitante, acerca de requerimientos de telefonía que es necesario conocer para armar el proyecto.

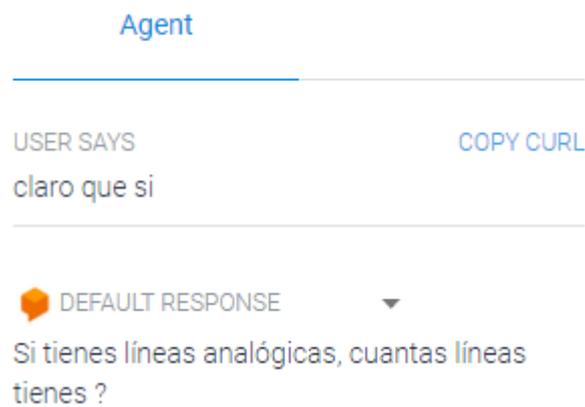


Figura 32 Prueba de Chatbot IV (Google)

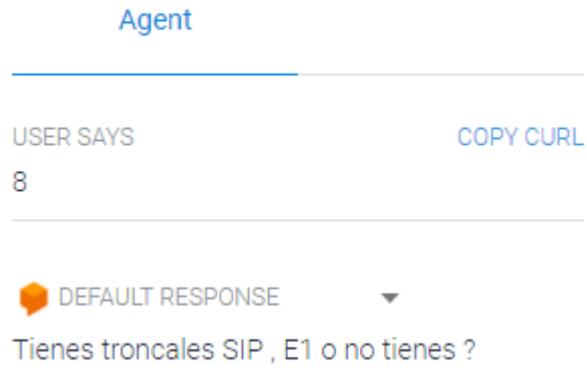


Figura 33 Prueba de Chatbot V (Google)

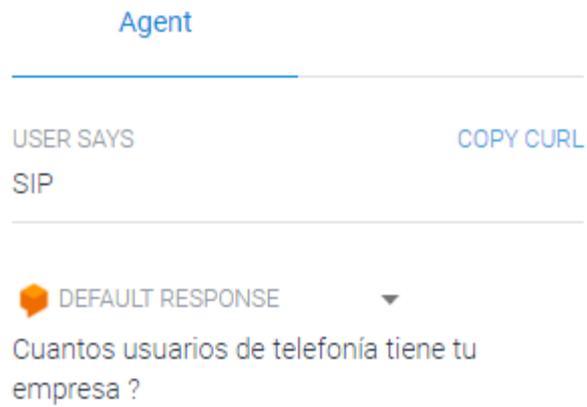


Figura 34 Prueba de Chatbot VI (Google)

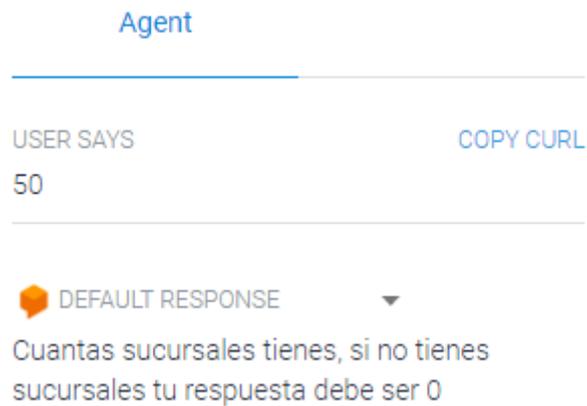


Figura 35 Prueba de Chatbot VII (Google)

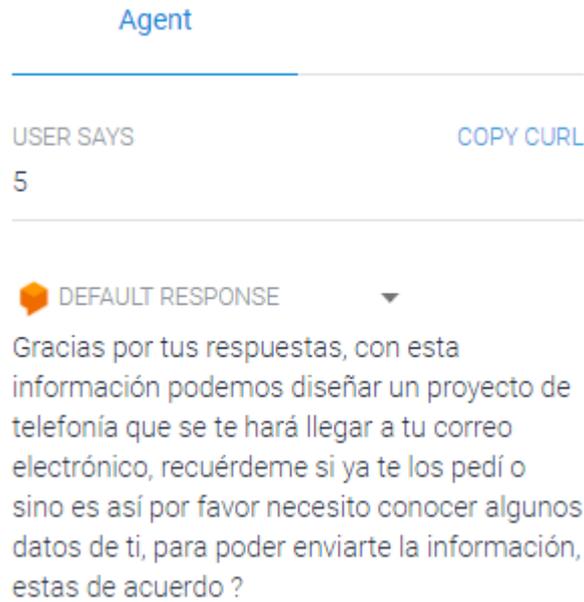


Figura 36 Prueba de Chatbot VIII (Google)

De manera seguida el chatbot intentará obtener datos del visitante para poder realizar el seguimiento respectivo.

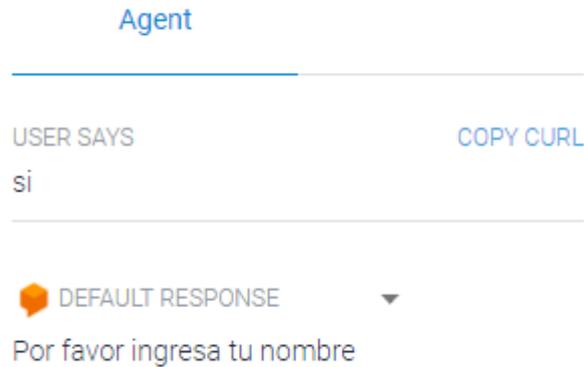


Figura 37 Prueba de Chatbot IX (Google)

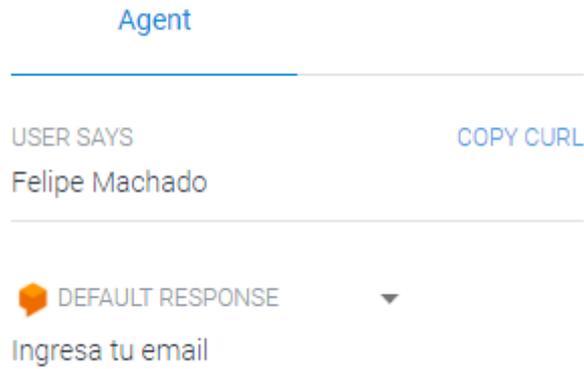


Figura 38 Prueba de Chatbot X (Google)

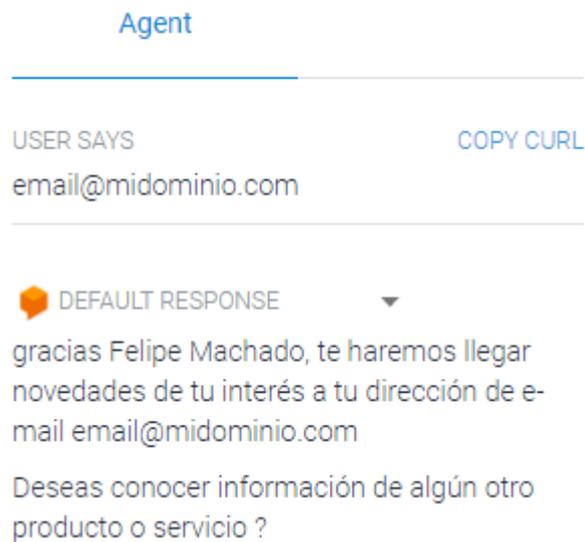


Figura 39 Prueba de Chatbot XI (Google)

El chatbot no pierde la oportunidad de obtener más información del visitante y le sugiere consultar sobre otro tema.

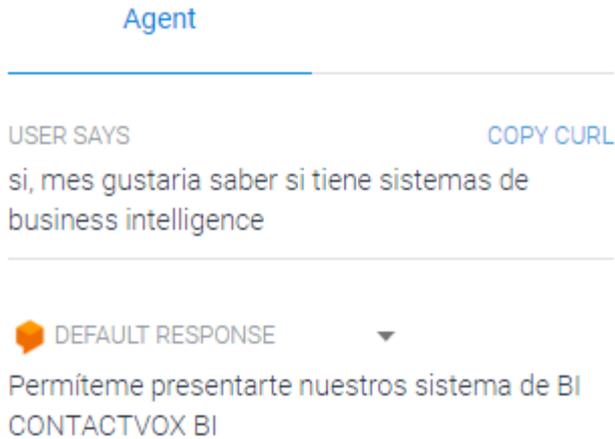


Figura 40 Prueba de Chatbot XII (Google)

En la siguiente figura podemos observar la respuesta negativa del visitante a dar sus datos. El chatbot debe entrenarse con la respuesta adecuada y tratar de persuadir al visitante en un futuro cercano de la conversación o proporcionar información de la empresa para que el visitante se comunique cuando desee.

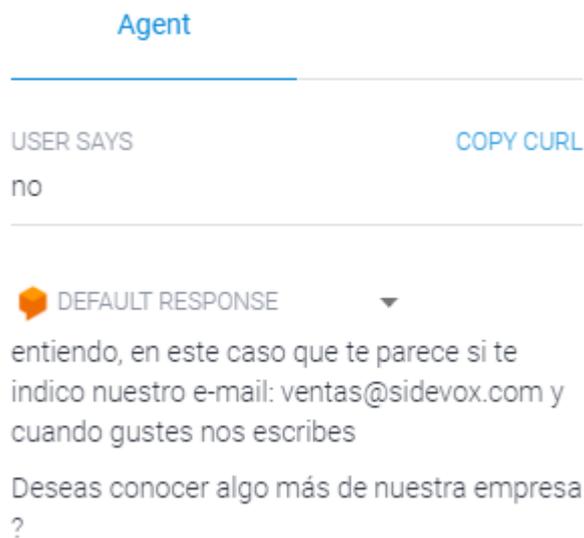


Figura 41 Prueba de Chatbot XIII (Google)

5 EVALUACIÓN DEL CHATBOT

5.1 Chatbots en páginas web

A continuación se analizan varios chatbots que se encuentran en producción en páginas de empresas del Ecuador y que se desempeñan como asistentes de ventas.

El análisis se lo va a realizar, tomando en consideración algunos criterios como:

- Saludos e inicio de conversación
- Capacidad de conversación
- Frases mal escritas, incompletas
- Faltas de ortografía
- Expresiones fuera de contexto
- Seguir a cabalidad la conversación propuesta por el chatbot
- Salir de la conversación y tratar de imponer un guión de usuario
- Frases simples
- Frases complejas
- Dos expresiones, en la misma frase
- Preguntas y respuestas
- Despedidas
- Latencia de conversación
- Experiencia de usuario

Los candidatos para realizar las pruebas son elegidos de empresas grandes que incorporen un chatbot en su página web.

5.1.1 Chatbot empresa Claro Ecuador

- El chatbot inicia la conversación presentándose e indicando que es un asistente virtual y sugiere que se introduzca palabras clave.



Figura 42 Asistente virtual I (CLARO)

- Presenta a una animación gráfica, de simulación de un humano.
- Al introducir palabras y el chatbot responde con un listado de posibles opciones, que coinciden con las palabras clave. Dichas opciones se presentan a manera de un menú que le permite al usuario elegir haciendo click en cualquier ítem.
- Al elegir un menú, el mismo que presenta un hipervínculo que hace referencia a más información y re direcciona la el sitio a otra página con más información del producto.
- El chatbot presenta una encuesta de satisfacción, para validar si la información brindada fue correcta y lo esperado por el visitante.
- El chatbot presenta sugerencias, las mismas que se enlistan y al que presenta un hipervínculo que hace referencia a más información y re direcciona la el sitio a otra página con más información de la elección.
- Al elegir otras opciones el chatbot responde con respuestas simples.
- El chatbot si pasa por alto faltas ortográficas.
- Al ingresar frases mal escritas el chatbot no es capaz de reconocer lo que el usuario desea, en algunos casos confunde los requerimientos.
- Cuando se colocan frases negativas las entienden como positivas: Ejemplo “no quiero un plan”, el chatbot entiende que quiero un plan.

- El chatbot siempre responde y presenta al final la encuesta de satisfacción y sugerencias de nuevos requerimientos.
 - Si el chatbot detecta intención de compra de un equipo, responde con la opción de contactar con un asistente de ventas humano.
 - Al tratar de establecer un guión de conversación, el chatbot siempre responde con opciones. Dichas opciones se presentan basándose en las palabras clave introducidas en la frase del visitante, pero sin analizar ningún tipo de contexto, sino solo por coincidencia de las palabras.
 - El chatbot no es capaz de entender frases negativas. No es capaz de reconocer expresiones como: “No me estas ayudando”, “No me entiendes” a las mismas responde con “Tienes problemas con nuestro productos y/o servicio”
-
- El chatbot no es capaz de establecer una conversación simple.
 - Es un chatbot de tipo: preguntas y respuestas.
 - El chatbot responde bien a las preguntas elegidas (no se deben escribir, solo elegir).
 - El chatbot entiende frases simples. “Me gustaría comprar un equipo”, pero presenta las respuestas como “Te gustaría comprar un equipo” SI - NO, al escribir, “si, me gustaría”, no lo entiende, vuelve a responder para elegir las opciones SI – NO.
 - El chatbot no entiende frases más complejas, como “ayer fui a comprar, pero el equipo estaba en mal estado”. El chatbot responde con “Quieres conocer nuestros equipos”.
 - Si el chatbot no entiende varias frases sugiere la comunicación con un asistente humano.
 - El chatbot no es susceptible a la latencia de las respuestas del usuario. La respuesta del usuario ante una pregunta sigue la secuencia lógica propuesta por el chatbot aún cuando el usuario se haya tardado unos 10 min en responder.
 - Desde un punto de vista más técnico este chatbot parecería que está programado con expresiones regulares y solo permite reconocer palabras, sin entender el contexto de las frases.
 - La interfaz del chatbot es atractiva, muy amigable, desde el primer momento se presenta ante el usuario como un robot.

5.1.2 Chatbot empresa Salud SA

- El chatbot inicia la conversación presentándose e indicando que es un asistente virtual y sugiere algunas opciones a manera de menú.

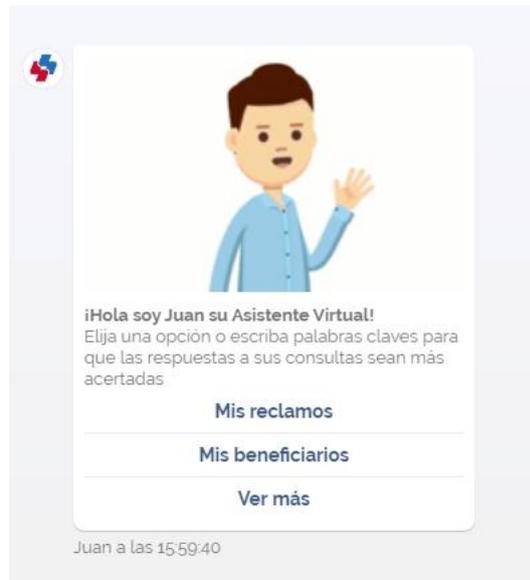


Figura 43 Asistente virtual VI (Saludsa)

- Presenta a una animación gráfica, de simulación de un humano, pero más parecido a una caricatura.
- Al introducir palabras y el chatbot responde con un listado de posibles opciones, que coinciden con las palabras clave. Dichas opciones se presentan a manera de un menú que le permite al usuario elegir haciendo click en cualquier ítem.
- Al elegir un menú, el mismo que presenta un hipervínculo que hace referencia a más información y re direcciona la el sitio a otra página con más información del producto.
- El chatbot presenta una encuesta de satisfacción, para validar si la información brindada fue correcta y lo esperado por el visitante.
- Al elegir otras opciones el chatbot responde con respuestas simples.
- El chatbot si pasa por alto faltas ortográficas.
- Al ingresar frases mal escritas el chatbot no es capaz de reconocer lo que el usuario desea, en algunos casos confunde los requerimientos.
- Cuando se colocan frases negativas las entienden como positivas: Ejemplo “no quiero un plan”, el chatbot entiende que quiero un plan.
- El chatbot siempre responde y presenta al final la encuesta de satisfacción y sugerencias de nuevos requerimientos.
- Si el chatbot detecta intención de compra de un servicio, responde con la opción de contactar con un asistente de ventas humano.
- Al tratar de establecer un guión de conversación, el chatbot siempre responde con opciones. Dichas opciones se presentan basándose en las palabras clave

introducidas en la frase del visitante, pero sin analizar ningún tipo de contexto, sino solo por coincidencia de las palabras.

- El chatbot es capaz de reconocer expresiones como: “No me estas ayudando”, “No me entiendes” a las mismas responde con una disculpa y un listado de opciones.
- Es un chatbot de tipo: preguntas y respuestas.
- El chatbot responde bien a las preguntas elegidas.
- El chatbot entiende frases simples. “Me gustaría comprar un plan”.
- El chatbot no entiende frases más complejas, como “me atendieron mal en sus oficinas”. El chatbot responde con “Quieres conocer nuestras oficinas”.
- Si el chatbot no entiende varias frases sugiere la comunicación con un asistente humano.
- El chatbot no es susceptible a la latencia de las respuestas del usuario. La respuesta del usuario ante una pregunta sigue la secuencia lógica propuesta por el chatbot aún cuando el usuario se haya tardado unos 10 min en responder.
- Desde un punto de vista más técnico este chatbot parecería que está programado con expresiones regulares y solo permite reconocer palabras, sin entender el contexto de las frases.
- El chatbot no es capaz de establecer una conversación simple.
- La interfaz del chatbot es atractiva, muy amigable, desde el primer momento se presenta ante el usuario como un robot.

5.1.3 Chatbot empresa Banco Pacífico

- El chatbot inicia la conversación presentándose e indicando que es un asistente virtual.



Figura 44 Asistente virtual XV (Banco del Pacífico)

- Presenta a una animación gráfica, con la imagen de un robot.
- Al introducir palabras y el chatbot responde con un listado de posibles opciones, que coinciden con las palabras clave. Dichas opciones se presentan a manera de un menú que le permite al usuario elegir haciendo click en cualquier ítem.
- Al elegir un menú, el mismo que presenta un hipervínculo que hace referencia a más información y re direcciona la el sitio a otra página con más información del producto.
- Se elije la opción de “Ubicación de Cajeros”. El chatbot permite captar información de geo-referenciación.
- Al entrar en esta opción no permite regresar al ingresar frases como “no deseo esta opción” o “no deseo esto” se *INTENT* ó con varias frases como “atrás”, “salir” sin embargo el chatbot insiste en que se ingrese la posición y si no se realiza esa acción la conversación se queda un ciclo sin salida.
- Al ingresar la posición responde de manera adecuada, con las ubicaciones de las agencias.
- El chatbot no presenta una encuesta de satisfacción, para validar si la información brindada fue correcta y lo esperado por el visitante, como los chatbots analizados anteriormente.
- Al elegir otras opciones el chatbot responde con respuestas simples.
- El chatbot no entiende el hecho de escribir con minúsculas una opción que se presenta con la primera letra en mayúscula.
- Al ingresar frases mal escritas el chatbot no es capaz de reconocer lo que el usuario desea, en algunos casos confunde los requerimientos.
- Cuando se colocan frases negativas las entienden como positivas: Ejemplo “no quiero saber mi saldo”, el chatbot entiende que quiero saber mi saldo.

- Si el chatbot detecta intención de compra de un servicio, responde con la opción de contactar con un asistente de ventas humano.
- Al tratar de establecer un guión de conversación, el chatbot siempre responde con opciones. Dichas opciones se presentan basándose en las palabras clave introducidas en la frase del visitante, pero sin analizar ningún tipo de contexto, sino solo por coincidencia de las palabras.
- El chatbot es capaz de reconocer expresiones como: “No me estas ayudando”, “No me entiendes” a las mismas responde con una disculpa y sugiere la transferencia a un asistente humano.
- Es un chatbot de tipo: preguntas y respuestas.
- El chatbot responde bien a las preguntas elegidas.
- El chatbot entiende frases simples. “Me gustaría saber mi saldo”. “Puedo pagar la luz”, “Quiero invertir mi dinero”, a todas estas preguntas responde con un menú u de opciones, muy bien relacionadas con las preguntas.
- El chatbot no entiende frases más complejas, como “me atendieron mal en su oficina de la Amazonas”. El chatbot responde con la opción de que ingrese mi posición para enviarme la ubicación de las oficinas.
- Si el chatbot no entiende varias frases sugiere la comunicación con un asistente humano.
- El chatbot se disculpa cuando no entiende alguna frase.
- El chatbot no es susceptible a la latencia de las respuestas del usuario. La respuesta del usuario ante una pregunta sigue la secuencia lógica propuesta por el chatbot aún cuando el usuario se haya tardado unos 10 min en responder.
- Desde un punto de vista más técnico este chatbot parecería que está programado con expresiones regulares y solo permite reconocer palabras, sin entender el contexto de las frases.
- El chatbot no es capaz de establecer una conversación simple.
- La interfaz del chatbot es atractiva, muy amigable, responde con menús incluso fotos o tarjetas.
- Desde el primer momento se presenta ante el usuario como un robot.

5.1.4 Chatbot empresa Banco de Loja

- El chatbot inicia la conversación presentándose e indicando que es un asistente virtual.
- El chatbot utiliza un plugin de Facebook, por lo que requiere que el usuario inicie su sesión de Facebook para empezar la conversación.



Figura 45 Asistente virtual XXIII (FACEBOOK)

- Presenta a una animación gráfica, con la imagen de un robot.
- El chatbot solicita que se acepten términos y condiciones antes de empezar la conversación
- Al introducir palabras y el chatbot responde con un listado de posibles opciones, que coinciden con las palabras clave. Dichas opciones se presentan a manera de un menú que le permite al usuario elegir haciendo click en cualquier ítem.
- Al elegir un menú, el mismo que presenta un hipervínculo que hace referencia a más información y re direcciona la el sitio a otra página con más información del producto.
- El chatbot no presenta una encuesta de satisfacción, para validar si la información brindada fue correcta y lo esperado por el visitante.
- Al elegir otras opciones el chatbot responde con respuestas simples.
- El chatbot si permite el ingreso de opciones escritas en minúsculas o mayúsculas.
- Al ingresar frases mal escritas el chatbot no es capaz de reconocer lo que el usuario desea, en algunos casos confunde los requerimientos o pide disculpas.
- Cuando se colocan frases negativas las entienden como positivas: Ejemplo “no deseo una cuenta”, el chatbot entiende que deseo una cuenta.
- Si el chatbot detecta intención de compra de un servicio, responde con la opción de contactar con un asistente de ventas humano.
- Al tratar de establecer un guión de conversación, el chatbot siempre responde con opciones. Dichas opciones se presentan basándose en las palabras clave introducidas en la frase del visitante, pero sin analizar ningún tipo de contexto, sino solo por coincidencia de las palabras.

- El chatbot es capaz de reconocer expresiones como: “No me estas ayudando”, “No me entiendes” a las mismas responde con una disculpa y una imagen del robot apenado.
- Es un chatbot de tipo: preguntas y respuestas.
- El chatbot responde bien a las preguntas elegidas.
- El chatbot entiende frases simples. “Me gustaría saber mi saldo”. “Puedo pagar la luz”, “Quiero invertir mi dinero”, a todas estas preguntas responde con un menú de opciones, muy bien relacionadas con las preguntas.
- El chatbot no entiende frases más complejas, como “quisiera poder hablar con alguien”. El chatbot no entiende esta frase.
- Si el chatbot no sugiere la comunicación con un asistente humano.
- El chatbot se disculpa cuando no entiende alguna frase.
- El chatbot no es susceptible a la latencia de las respuestas del usuario. La respuesta del usuario ante una pregunta sigue la secuencia lógica propuesta por el chatbot aún cuando el usuario se haya tardado unos 10 min en responder.
- Desde un punto de vista más técnico este chatbot parecería que está programado utilizando algún *framework* de inteligencia artificial.
- El chatbot no es capaz de establecer una conversación simple.
- La interfaz del chatbot es atractiva, muy amigable, responde con menús incluso fotos o tarjetas.
- Desde el primer momento se presenta ante el usuario como un robot.

5.2 Chatbots en páginas web vs ContactBot

De las pruebas realizadas, podemos concluir

- El chatbot de este trabajo tiene una interfaz gráfica solo en texto (Debería en un futuro tener un GUI).
- El chatbot no presenta lista de opciones por lo cual abre la posibilidad de una conversación abierta como lo haría un humano.
- Es capaz de responder a preguntas.
- El chatbot solicita datos para seguimiento de la interacción en tiempos futuros.
- Es capaz de detectar usuarios no deseados.
- Entiende cuando un usuario no quiere un servicio y no se equivoca presentando lista de servicios.

- Permite establecer una conversación como lo haría un humano, siempre y cuando el humano con el que se interactúe, siga el guión establecido, que en sí es bastante flexible y no es tan rígido.
- No se integra con bases de datos ni otros sistemas (Debería en un futuro).

6 CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Existen muy pocos sitios web de empresas en Ecuador con chatbots, se busca en alrededor de 20 empresas grandes pero no tienen implementado un chatbot. Se realiza la búsqueda en páginas de bancos, supermercados, aseguradoras, empresas de telecomunicaciones.

Los chatbots encontrados en las pocas páginas que los contenían, tienen una interfaz gráfica interesante pero se enfocan en ser chatbots de preguntas y respuestas, no se puede establecer una conversación como con un ser humano.

El estado del arte ha avanzado mucho en los últimos años. Las plataformas de desarrollo de chatbots están mejorando la forma de construir este tipo de aplicaciones, sin embargo sigue siendo un trabajo complejo. Existen herramientas actuales que facilitan el procesamiento del lenguaje natural, un aspecto bastante complicado, pero gracias a estas herramientas se vuelve más sencillo.

Las aplicaciones de un chatbot son bastante amplias, pero el corpus debe ser entrenado de acuerdo a dichas aplicaciones. Es decir, en el ámbito en el que el chatbot se va a desempeñar hay que desarrollar un corpus exclusivo, ya que el mismo obedece a la visión o misión del proyecto y si es un chatbot comercial para una empresa, se debe alinear a la misma y aprender de sus productos, servicios y su cultura corporativa que es la que se espera transmitir en las comunicaciones. Es muy importante que el chatbot se personalice y profesionalice, es decir que tenga un nombre y se presente de manera adecuada, generando un estado de presencia que genera seguridad en un ser humano. Adicionalmente la profesionalización le brinda la guía del entrenamiento. La idea no es engañar al usuario sino brindarle confianza, para que lo expresado por el chatbot, sea aceptado por el usuario. Aunque el usuario sea consciente de que está conversando con un robot, debe conocer que el mismo tiene un nombre y asume el rol de una persona, lo cual facilitará la comunicación.

Un aspecto bastante complicado del chatbot, fue el diseñar los contextos y la forma de entrenamiento de los *INTENTS*, aparentemente es sencillo crear un *INTENT*, pero cuando se tienen conversaciones extensas, se tiende a caer en ambigüedades y controlar la gran mayoría de éstas requiere un esfuerzo grande, para que el agente inteligente no pierda el control del diálogo y pueda obtener las respuestas correctas de

los usuarios. Este puede ser uno de los retos más grandes a vencer el desarrollo de chatbots, el hecho de diseñar un sistema capaz de mantener una conversación coherente con un usuario. El chatbot no es solo un sistema que realiza el procesamiento del lenguaje natural, se requiere establecer contextos, flujos de conversación que siguen siendo impredecibles debido a las respuestas propias de un ser humano. Se debe tratar de controlar al máximo el hilo de la conversación y volver a encaminarla.

El agente debe conocer en todo momento por donde está encaminada la conversación, hacia donde se dirige y es más debe ser el agente quien dirija la misma y no darle tanta libertad al usuario. Por este motivo la mayoría de chatbots comerciales se restringen a ser agentes de preguntas y respuestas, es decir no establecen una conversación con el usuario, sino solo se limitan a responder con opciones a las preguntas que hace un usuario.

La obtención de los datos para el entrenamiento, es uno de los aspectos más complicados, ya que la empresa tenía cierta información y muchas veces no sabe lo que quiere y transmitir.

DIALOGFLOW, facilita mucho el entrenamiento, ya que no ayuda a procesar el texto, se asume que DIALOGFLOW lo hace de manera interna. Específicamente el procesamiento del texto, debería consistir en tomar la frase y generar variantes de la misma, como suprimir artículos, agregar o eliminar signos de puntuación, entrenar la frase con faltas ortográficas, en mayúsculas, minúsculas, entre otras. Para el entrenamiento con KERAS este trabajo se debe realizar de manera manual o en su defecto se debe construir un módulo adicional para procesar el texto y generar estas variantes de manera automática. Por este motivo el entrenamiento con KERAS es muchas más complicado.

Un dato curioso que se evidenció con algunas pruebas realizadas, es que cuando no se le indica a un usuario que se trata de un agente inteligente, la conversación tiende a fluir de manera natural y pudiendo esta llegar a pasar desapercibo que se trata de una conversación entre una persona y un agente inteligente. Sin embargo cuando el usuario conoce que se trata de un chatbot, tiende a buscar la falla del mismo, tratando de poner a prueba el software incluso de una manera absurda, que no lo haría si supiese que se trata de una persona que es con la que está conversando. Por este motivo se consideró desarrollar el módulo de detección de visitantes no deseados y se generan ciertas preguntas respecto al comportamiento de los seres humanos frente a las máquinas.

Finalmente el chatbot debe estar muy bien entrenado, para que pueda ser capaz de responder al usuario y continuar el diálogo. El corpus fue prediseñado manualmente tomando en cuenta algunas frases, que sirvan para el proyecto. Se escribieron apenas alrededor de 100 frases con el objetivo de que sea una muestra. Mientras más ejemplos se tengan, el modelo se puede entrenar de mejor manera. Por lo tanto una de las tareas que demanda mayor esfuerzo es la de armar un corpus. Esta tarea es vital si queremos que un chatbot tenga buenas capacidades de respuesta.

6.2 RECOMENDACIONES

Un chatbot se debe desarrollar de acuerdo a un objetivo en concreto, por este motivo es muy importante construir un corpus que el chatbot se entrene de manera adecuada.

El chatbot debe seguir un proceso de mejora, a través del desarrollo de nuevas versiones. Para una segunda versión del chatbot, se recomienda:

- Incorporar un corpus en español y una herramienta de procesamiento de lenguaje natural como NLTK, pero con el corpus en español. Existe un corpus desarrollado por la Universidad de Standford que permite realizar, etiquetado de las partes de la oración (POS tagging), para potencializar las capacidades de respuesta del chatbot y para agregar más corpus, se deberá realizar un procesamiento bastante completo de procesamiento de datos, ya que no existe por el momento corpus en español bien estructurado.
- Mejorar la interfaz gráfica de usuario, es decir tener la capacidad de mostrar contenido multimedia.
- Incorporar nuevos elementos de entrenamiento utilizando otras redes neuronales, para poder procesar textos más complejos.
- Los corpus deberían ser extraídos de las páginas web en la cual va a estar alojado el bot. Se podría utilizar una técnica como *web scraping*, para poder ensamblar corpus de manera automática y robusta.
- Se recomienda potencializar el chatbot permitiéndole realizar consultas a bases de datos u otros medios de procesamiento de datos. Por ejemplo si el cliente ya dio sus datos tener almacenados en variables para evitar pedírselos otra vez.

7 Bibliografía

- Amazon. (s.f.). *Amazon Web Services*. Obtenido de <https://aws.amazon.com/es/lex/>
- APACHE. (s.f.). *MXNET*. Obtenido de <https://mxnet.apache.org/>
- Banco del Pacifico. (s.f.). *Banco del Pacífico*. Obtenido de <https://www.bancodelpacifico.com/>
- Bhagwat, V. A. (2018). *Deep Learning for Chatbots*.
- Cahn, J. (2017). *CHATBOT: Architecture, Design, & Development*.
- Chandrashekar Lakshminarayanan, A. V. (2020). Deep Gated Networks: A framework to understand training and generalisation in deep learning.
- chatbotsmagazine. (s.f.). <https://chatbotsmagazine.com/>. Recuperado el 24 de 12 de 2018, de <https://chatbotsmagazine.com/7-types-of-bots-8e1846535698>
- CLARO. (s.f.). *CLARO*. Obtenido de <https://www.claro.com.ec/>
- Colby, K. M. (1981). *Modeling a paranoid mind*. *Behavioral and Brain Sciences*, 4(04), 515.
- DICTIONARIES, E. O. (s.f.). <https://en.oxforddictionaries.com>. Obtenido de <https://en.oxforddictionaries.com/definition/chat>
- FACEBOOK. (s.f.). *Facebook Banco del Loja*. Obtenido de <https://www.facebook.com/BancodeLoja/>
- Falko Koetter, M. B. (s.f.). *Conversational Agents for Insurance Companies: From Theory to Practice*.
- Ganegedara., T. (May 2018). *Natural Language Processing with TensorFlow*.
- GITHUB. (s.f.). Obtenido de https://github.com/keras-team/keras/blob/master/examples/lstm_seq2seq.py
- Google. (s.f.). *Colab*. Obtenido de <https://colab.research.google.com/drive/19h8lagWaaV6oqx8N8LKAs624AB1m8GOy>
- Google. (s.f.). *Dialogflow*. Obtenido de <https://dialogflow.com/>
- Google. (s.f.). *DialogFlow console*. Obtenido de <https://dialogflow.cloud.google.com/>
- IBM. (s.f.). *Watson*. Obtenido de <https://www.ibm.com/watson>
- Jía, J. (2003). *The Study of the Application of a Keywords-based Chatbot System on the Teaching of Foreign Languages*.
- Junjie Yin, Z. C. (s.f.). *A Deep Learning Based Chatbot for Campus Psychological Therapy*.

Jurafsky, D. C. (s.f.). Obtenido de <http://web.stanford.edu/class/cs124/lec/chatbot.pdf>

KERAS. (s.f.). *KERAS DOCUMENTATION*. Obtenido de <https://keras.io/>

LAB, L. (s.f.). *THEANO*. Recuperado el 05 de 12 de 2019, de <http://deeplearning.net/software/theano/>

Mateos Martín F. J., R. R. (2013). *Dpto. de Ciencias de la Computación e Inteligencia Artificial*. Recuperado el 05 de 12 de 2019, de <https://www.cs.us.es/>

Microsoft. (s.f.). *Microsoft Bot Framework*. Obtenido de <https://dev.botframework.com/>

Microsoft. (s.f.). *Microsoft Docs*. Obtenido de <https://docs.microsoft.com/en-us/cognitive-toolkit/>

PANDORABOTS. (s.f.). *Pandorabots*. Obtenido de <https://home.pandorabots.com/home.html>

PRINCETON. (s.f.). *WORDNET*. Obtenido de <https://wordnet.princeton.edu/>

PYTORCH. (s.f.). *PyTorch*. Obtenido de <https://pytorch.org/>

Saludsa. (s.f.). *Saludsa*. Obtenido de <https://www.saludsa.com/>

SCRUMstudy. (2013). *Una guía para el conocimiento de SCRUM*. Phoenix.

SIDEVOX. (s.f.). *CONTACTVOX*. Obtenido de <https://www.contactvox.com>

Source, F. O. (s.f.). *Caffe2*. Obtenido de <https://caffe2.ai/>

Sugisaki, K. (s.f.). Chat-Bot-Kit: A web-based tool to simulate text-based interactions between humans and with computers.

TENSORFLOW. (s.f.). *TENSORFLOW*. Recuperado el 05 de 12 de 2019, de <https://www.tensorflow.org/>

Verdugo, P. (s.f.). *planetachatbot*. Obtenido de <https://planetachatbot.com/mejores-frameworks-de-desarrollo-de-chatbots-7160c59b1ac4>

Vogel, J. (2017). *Chatbots: Development and Applications*.

Wallace, R. S. (s.f.). Obtenido de <https://www.pandorabots.com>

Wallis. (s.f.). <https://web.csulb.edu/>. Obtenido de <https://web.csulb.edu/~cwallis/artificialn/History.htm>

WIT.AI. (s.f.). *wit.ai*. Obtenido de <https://wit.ai/>

Xingkun Liu, A. E. (s.f.). Benchmarking Natural Language Understanding Services for building Conversational Agents.

ANEXOS

Entrenamiento de INTENTS

• Default Welcome Intent SAVE ⋮

Welcome Add event

Training phrases ⓘ Search training phrases 🔍 ^

” Add user expression
” Como han pasado
” Cómo están Uds.
” Buenas
” Hola cómo están
” Buenas noches
” Buena tarde
” Buenas tardes
” Buen día
” Hola

Figura 46 Entrenamiento de saludos (Google)

Responses ⓘ ^

DEFAULT +

Text Response ⓘ 🗑

- 1 Gracias por su comunicación. Mi nombre es ContactBot, esperamos que su visita sea placentera, yo seré su asistente virtual durante la misma. ¿En qué podemos servirle?
- 2 Gracias por comunicarse con nuestra empresa. Mi nombre es ContactBot y seré sus asistente durante su visita en nuestro sitio web. ¿En qué podemos servirle?
- 3 Gracias por visitar nuestro sitio web. Soy ContactBot y estaré gustoso de asistirle. ¿Tiene alguna pregunta o requerimiento en especial?
- 4 Enter a text response variant ⌵

ADD RESPONSES

Set this intent as end of conversation ⓘ

Figura 47 Respuestas a saludos (Google)

Prueba del entrenamiento de saludos

Se ingresan algunas frases, para ver cómo responde el chatbot.

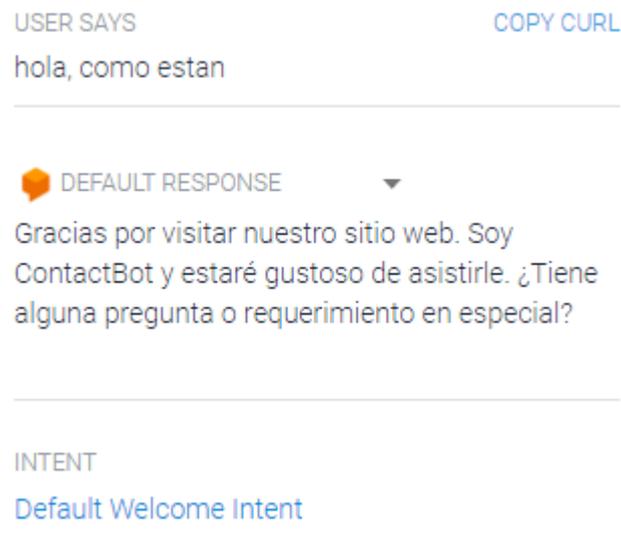


Figura 48 Prueba de respuesta I (Google)



Figura 49 Prueba de respuesta II (Google)

Entrenamiento del corpus de despedidas

Se crea un nuevo *INTENT* al cual denominaremos **Despedidas**, con las frases antes enlistadas.

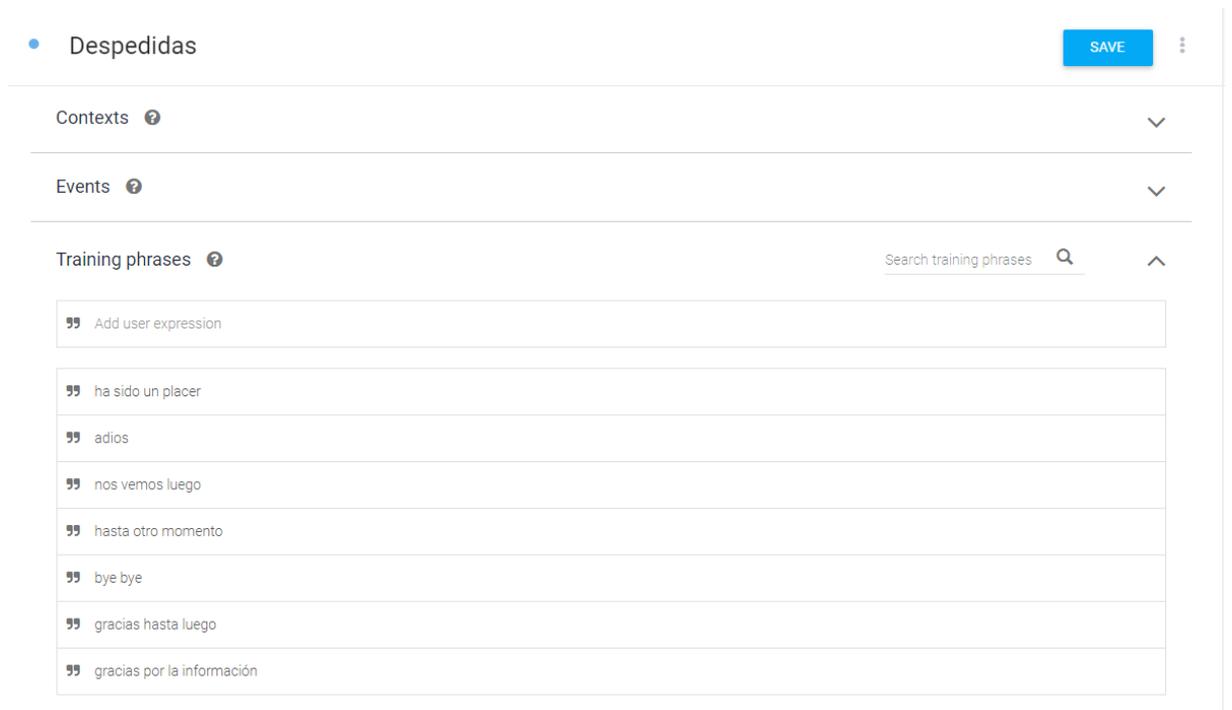


Figura 50 Entrenamiento de despedidas (Google)

Prueba del entrenamiento de despedidas

Se ingresan algunas frases, para ver cómo responde el chatbot.

- hasta luego
- bye

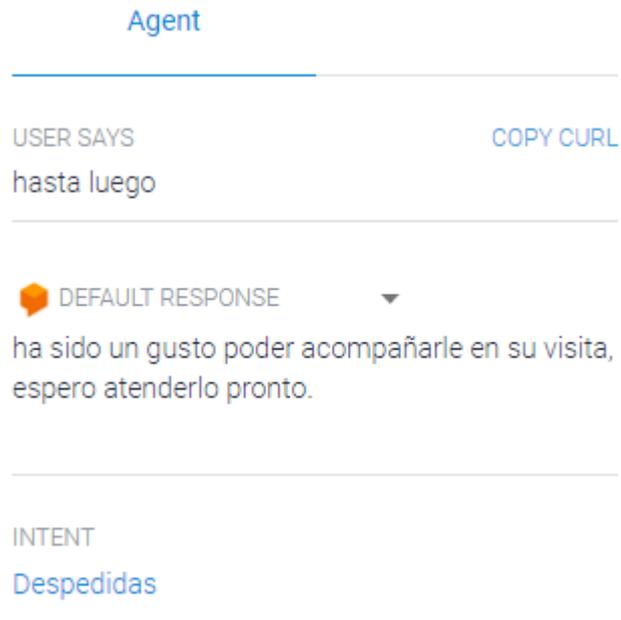


Figura 51 Prueba de despedidas I (Google)

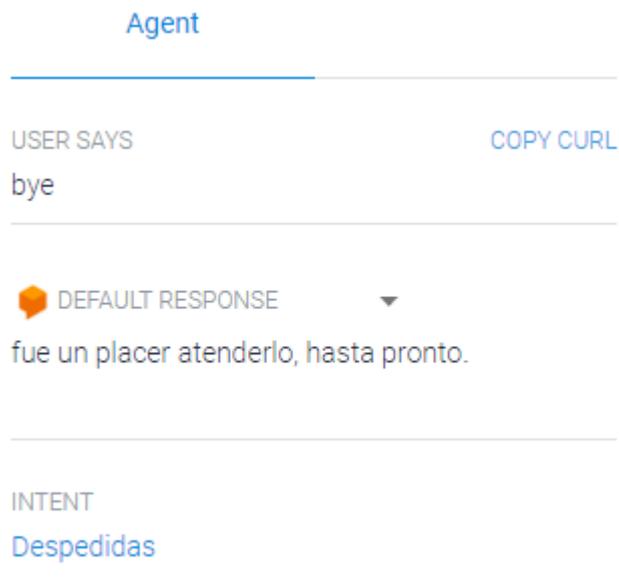


Figura 52 Prueba de despedidas II (Google)

Entrenamiento del corpus de datos de la empresa

Creamos un nuevo *INTENT* denominado **Datos de la empresa**. Para poder responder de manera apropiada a preguntas generales relacionadas con la empresa:

The screenshot shows the Google Assistant training interface for a new intent named "Datos de la empresa". At the top, there is a blue "SAVE" button and a menu icon. Below this, there are three main sections: "Contexts", "Events", and "Training phrases". The "Training phrases" section is expanded, showing a search bar and a list of training phrases. The first phrase is "Add user expression". The following phrases are: "¿Cuántos años de experiencia tienen?", "¿Qué hacen?", "¿En dónde están ubicados?", "¿Qué servicios ofrecen?", "¿Quiénes son Uds.?", "¿De donde son?", "¿Cuáles son sus servicios?", "¿A qué se dedican?", and "¿Cuántos años están en el mercado?".

Training phrases
Add user expression
¿Cuántos años de experiencia tienen?
¿Qué hacen?
¿En dónde están ubicados?
¿Qué servicios ofrecen?
¿Quiénes son Uds.?
¿De donde son?
¿Cuáles son sus servicios?
¿A qué se dedican?
¿Cuántos años están en el mercado?

Figura 53 Entrenamiento de datos de la empresa (Google)

Prueba del entrenamiento de datos de la empresa

Se ingresan algunas frases, para ver cómo responde el chatbot.

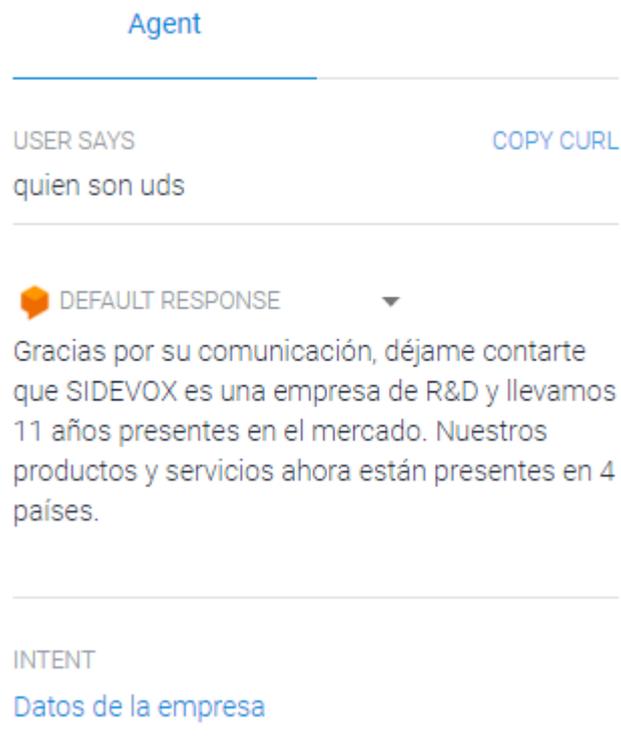


Figura 54 Prueba de datos de la empresa I (Google)

Entrenamiento de corpus técnico

Entrenamiento del corpus de producto CONTACTVOX UCS

Creamos un nuevo *INTENT* denominado **Comunicaciones Unificadas**

• Comunicaciones Unificadas SAVE

Contexts ⓘ ▼

Events ⓘ ▼

Training phrases ⓘ Search training phrases 🔍 ^

” Add user expression
” Quiero cambiar mi sistema Elastix por uno nuevo
” Me pueden ayudar con un sistema Elastix
” Venden Isabel
” Deseo un software de voip?
” Quiero integrar mi sistema de telefonía
” Deseo un sistema de voip
” Que tipo de centrales tienen
” Venden Panasonic
” Que tienen de solución en telefonía
” Uds venden Grandstream

Figura 55 Entrenamiento de producto CONTACTVOX UCS (Google)

Prueba del corpus de producto CONTACTVOX UCS

Se ingresan algunas frases, para ver cómo responde el chatbot.

Agent

USER SAYS COPY CURL

Quiero una central Elastix

📌 DEFAULT RESPONSE ▼

Somos especialistas en telefonía IP, podemos recomendarte, nuestro sistema de comunicaciones CONTACTVOX UCS

INTENT

Comunicaciones Unificadas

Figura 56 Prueba de producto CONTACTVOX UCS (Google)

CONTACTVOX CC

Entrenamiento del corpus de producto CONTACTVOX CC

Creamos un nuevo *INTENT* denominado **Contactcenter**

The screenshot shows the training interface for the 'Contactcenter' intent. At the top, there is a blue 'SAVE' button and a vertical ellipsis menu. Below this, there are sections for 'Contexts', 'Events', and 'Training phrases'. The 'Training phrases' section is expanded, showing a list of phrases with a search bar and a magnifying glass icon. The phrases are:

- ” Add user expression
- ” ¿Tengo un contactcenter antiguo?
- ” ¿Quiero cambiar mi contact center?
- ” ¿Tengo un sistema de contactcenter y quiero cambiarlo?
- ” ¿Deseo una contactcenter?
- ” ¿Me pueden asesorar, quiero instalar un callcenter?
- ” ¿Estoy interesado en comprar un callocenter?
- ” ¿Tienen call centers?
- ” ¿Tienen sistemas de callocenter?
- ” ¿Uds. venden callocenter?

Figura 57 Entrenamiento de producto CONTACTVOX CC (Google)

Prueba del corpus de producto CONTACTVOX CC

Se ingresan algunas frases, para ver cómo responde el chatbot.

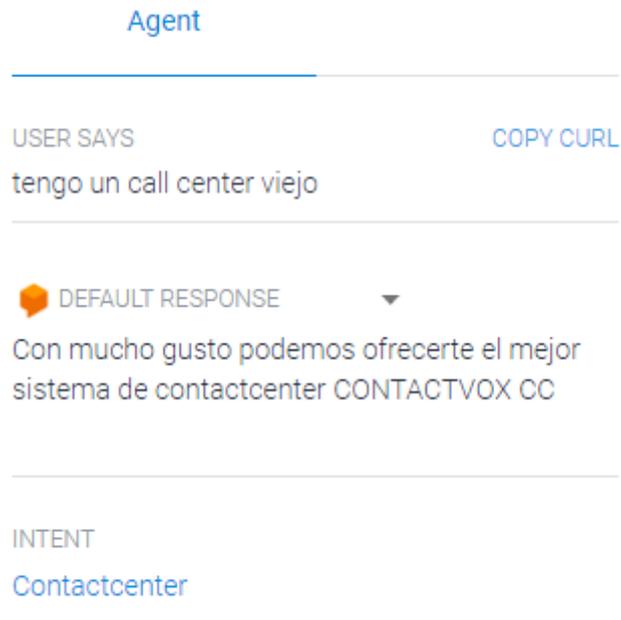


Figura 58 Prueba de producto CONTACTVOX CC

CONTACTVOX CRM

Entrenamiento del corpus de producto CONTACTVOX CRM

Creamos un nuevo *INTENT* denominado **Contactcrm**

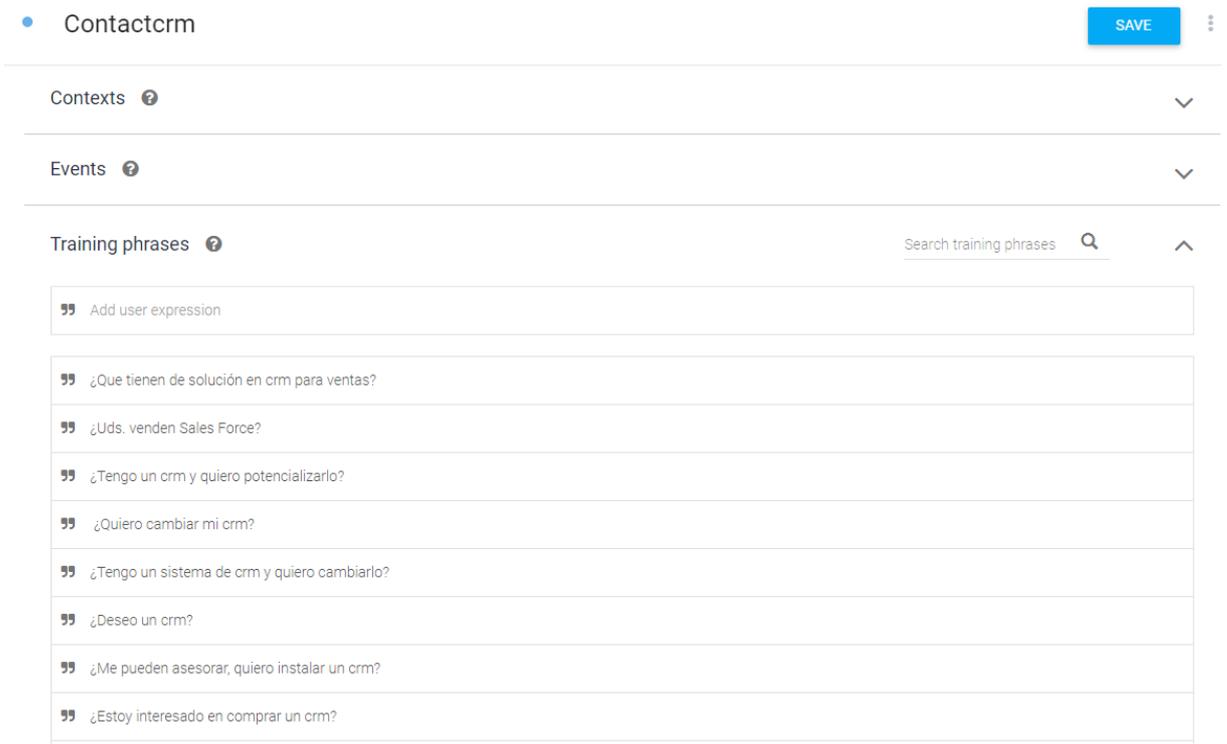


Figura 59 Entrenamiento producto CONTACTVOX CRM (Google)

Prueba del corpus de producto CONTACTVOX CC

Se ingresan algunas frases, para ver cómo responde el chatbot.

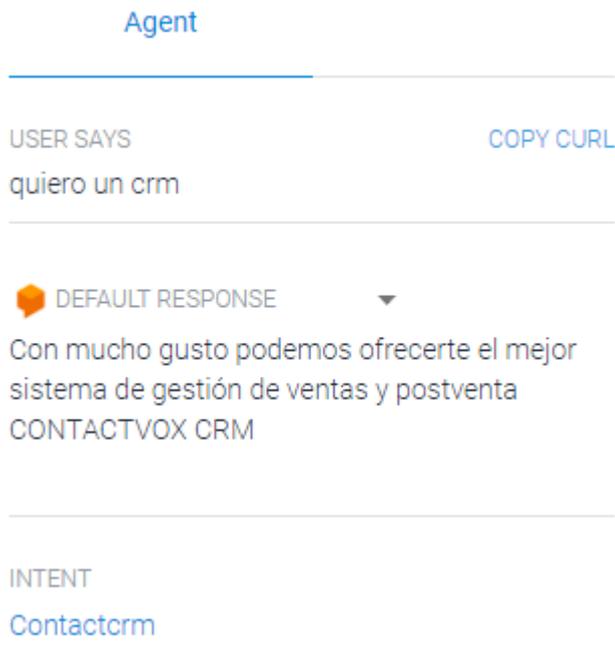


Figura 60 Pruebas de producto CONTACTVOX CRM (Google)

7.1.1.1 CONTACTVOX BI

7.1.1.1.1 Entrenamiento del corpus de producto CONTACTVOX BI

Creamos un nuevo *INTENT* denominado **Contactbi**

The screenshot shows the configuration page for the 'Contactbi' intent. At the top, there is a blue 'SAVE' button and a vertical ellipsis menu. Below this, there are three main sections: 'Contexts', 'Events', and 'Training phrases'. The 'Training phrases' section is expanded, showing a search bar and a list of training phrases. The phrases are:

- ¿Deseo un sistema de reportes?
- ¿Deseo un sistema de business intelligence?
- ¿Qué tipo de business intelligence tienen?
- ¿Venden software para reportería y estadísticas?
- ¿Que tienen de solución en business intelligence?
- ¿Uds. venden Power BI?
- ¿Tengo un BI y quiero potencializarlo?
- ¿Quiero cambiar mis reportes?

Figura 61 Entrenamiento de producto CONTACTVOX BI (Google)

Prueba del corpus de producto CONTACTVOX BI

Se ingresan algunas frases, para ver cómo responde el chatbot.

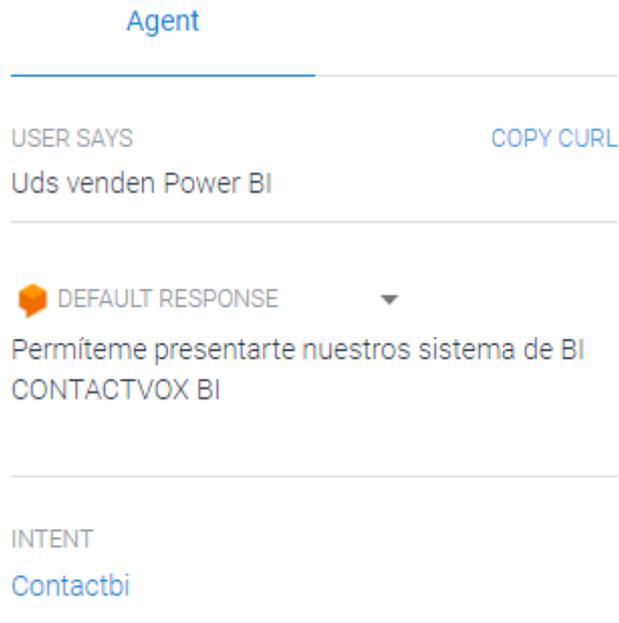


Figura 62 Prueba de producto CONTACTVOX BI (Google)

CONTACTVOX VIDEO

Entrenamiento del corpus de producto CONTACTVOX VIDEO

Creamos un nuevo *INTENT* denominado **Contactvideo**

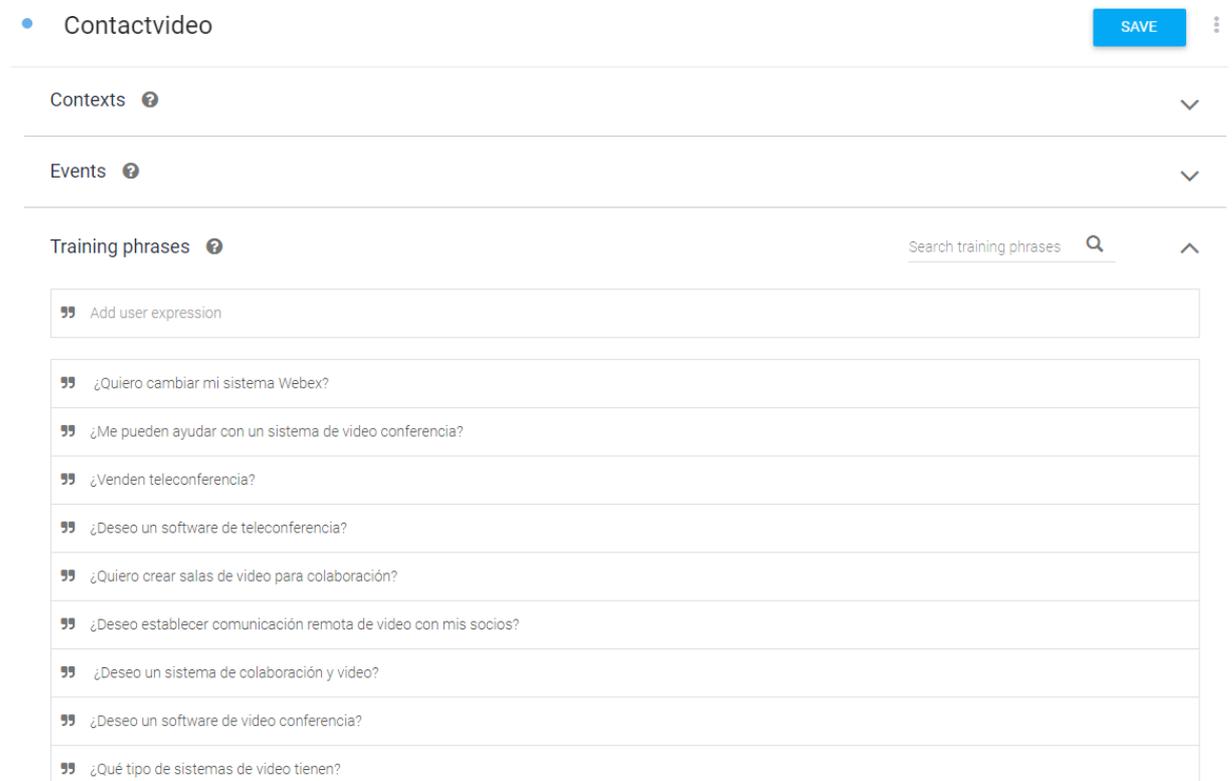


Figura 63 Entrenamiento de producto CONTACTVOX VIDEO (Google)

7.1.1.1.2 Prueba del corpus de producto CONTACTVOX VIDEO

Se ingresan algunas frases, para ver cómo responde el chatbot.

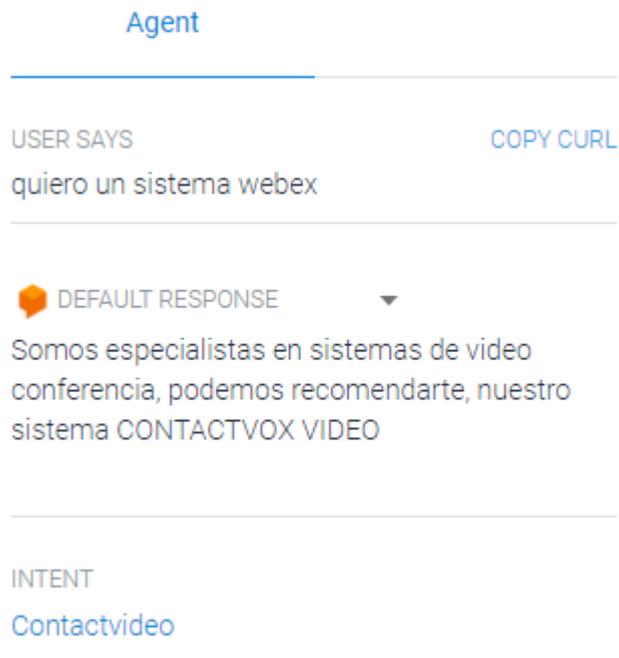


Figura 64 Prueba de producto CONTACTVOX VIDEO (Google)

CONTACTVOX CALLMATIC

Entrenamiento del corpus de producto CONTACTVOX CALLMATIC

Creamos un nuevo *INTENT* denominado **Contactmatic**

The screenshot shows the 'Contactmatic' training interface. At the top, there is a 'SAVE' button and a menu icon. Below are sections for 'Contexts', 'Events', and 'Training phrases'. The 'Training phrases' section includes a search bar and a list of phrases:

- Add user expression
- ¿Venden software para llamadas con texto speech?
- ¿Uds. venden sistemas de telemarketingo?
- ¿Quiero cambiar mi sistema de telemarketingo?
- ¿Me pueden ayudar con un sistema de llamadas y mensajes?
- ¿Venden envío de mensajes masivos?
- ¿Deseo un software de telemarketingo?
- ¿Quiero crear campañas de marcación automática?

Figura 65 Entrenamiento de producto CONTACTVOX CALLMATIC (Google)

Prueba del corpus de producto CONTACTVOX CALLMATIC

Se ingresan algunas frases, para ver cómo responde el chatbot.

The screenshot shows the chatbot interface. At the top, the word 'Agent' is displayed. Below it, the user input is shown as 'quiero crear una campaña de telemarketingo'. The chatbot's default response is: 'Nuestra empresa es especialista en sistemas de tele marketing que permite realizar llamadas automáticas y puede incluir tecnología TTS'. Below the response, the intent is identified as 'Contactmatic'.

Figura 66 Prueba de producto CONTACTVOX CALLMATIC

CONTACTVOX CONTACTWEB

Entrenamiento del corpus de producto CONTACTVOX CONTACTWEB

Creamos un nuevo *INTENT* denominado **Contactweb**

- Contactweb SAVE

Contexts ⓘ

Events ⓘ

Training phrases ⓘ 🔍

” Add user expression
” ¿Tienen webrtc?
” ¿ Tienen SIP trunks?
” ¿Venden troncales SIP?
” ¿Necesito una línea de respaldo del 1800?
” ¿Quiero cambiar mi 1800?
” ¿Me pueden ayudar con un sistema de chatweb?
” ¿Venden chatbots?
” ¿Deseo un software de de bots?
” ¿Quiero hablar con mis clientes por la web?

Figura 67 Entrenamiento de producto CONTACTVOX CONTACTWEB (Google)

Prueba del corpus de producto CONTACTVOX CONTACTWEB

Se ingresan algunas frases, para ver cómo responde el chatbot.

Agent

USER SAYS COPY CURL

venden chatbots

 DEFAULT RESPONSE ▼

Nuestra empresa es especialista en sistemas web, a través de tecnología WebRtc, sus clientes podrán comunicarse con su empresa por su página web a través de llamadas o chat y serán atendidas por un chatbot

INTENT

Contactweb

Figura 68 Pruebas de producto CONTACTVOX CONTACTWEB (Google)

• Errores SAVE 

Contexts  ▼

Events  ▼

Training phrases  Search training phrases  ^

 Add user expression
 no puedes responder
 no sabes nada
 no sabes
 no funcionas
 no creo que este este funcionando
 no entiendes verdad
 no me estas entendiendo
 creo que me explique mal
 no necesito eso
 no me entiendes

Figura 69 Entrenamiento de Errores (Google)

Visitantes no deseados

• Visitantes no deseados ⓘ SAVE ⋮

Contexts ⓘ ▼

Events ⓘ ▼

Training phrases ⓘ Search training phrases 🔍 ▲

” Add user expression

” torpe

” idiota

” me haces perder el tiempo

” que imbecil

” no sabes nada ⓘ

” eres un tonto

Figura 70 Entrenamiento visitantes no deseados (Google)

Prueba del entrenamiento de la obtención de datos del visitante
 Se ingresan algunas frases, para ver cómo responde el chatbot.

USER SAYS
COPY CURL

por supuesto

🗨️
DEFAULT RESPONSE
▼

Cómo te llamas ?

Figura 71 Prueba de obtención de datos de visitante I (Google)

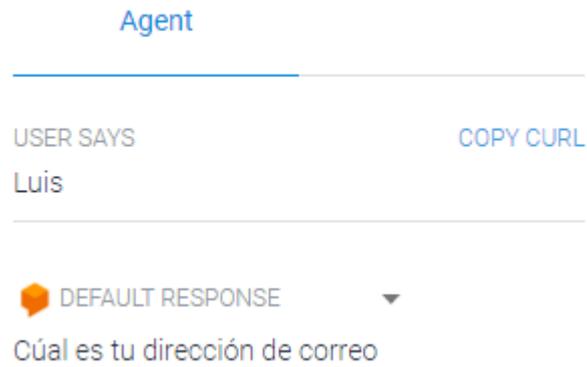


Figura 72 Prueba de obtención de datos de visitante II (Google)

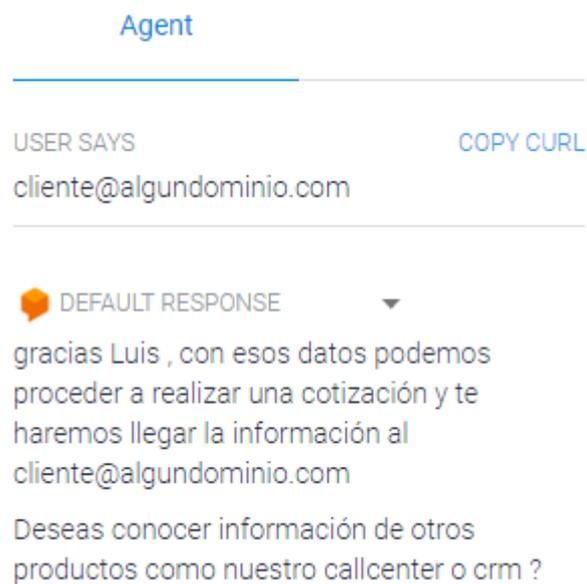


Figura 73 Prueba de obtención de datos de visitante III (Google)

ANEXO 2

CODIGO DE ENTRENAMIENTO EN KERAS, BASADO EN EL PROYECTO DE (GITHUB)

```
from __future__ import print_function

from keras.models import Model
from keras.layers import Input, LSTM, Dense
import numpy as np

batch_size = 64 # Batch size for training.
epochs = 100 # Number of epochs to train for.
latent_dim = 256 # Latent dimensionality of the encoding space.
num_samples = 200 # Number of samples to train on.
# Path to the data txt file on disk.
data_path = 'corpus.txt'

# Vectorize the data.
input_texts = []
target_texts = []
input_characters = set()
target_characters = set()
with open(data_path, 'r', encoding='utf-8') as f:
    lines = f.read().split('\n')
for line in lines[: min(num_samples, len(lines) - 1)]:
    input_text, target_text = line.split('\t')
    # We use "tab" as the "start sequence" character
    # for the targets, and "\n" as "end sequence" character.
    target_text = '\t' + target_text + '\n'
    input_texts.append(input_text)
    target_texts.append(target_text)
    for char in input_text:
        if char not in input_characters:
            input_characters.add(char)
    for char in target_text:
        if char not in target_characters:
            target_characters.add(char)
```

```
input_characters = sorted(list(input_characters))
target_characters = sorted(list(target_characters))
num_encoder_tokens = len(input_characters)
num_decoder_tokens = len(target_characters)
max_encoder_seq_length = max([len(txt) for txt in input_texts])
max_decoder_seq_length = max([len(txt) for txt in target_texts])

print('Number of samples:', len(input_texts))
print('Number of unique input tokens:', num_encoder_tokens)
print('Number of unique output tokens:', num_decoder_tokens)
print('Max sequence length for inputs:', max_encoder_seq_length)
print('Max sequence length for outputs:', max_decoder_seq_length)

input_token_index = dict(
    [(char, i) for i, char in enumerate(input_characters)])
target_token_index = dict(
    [(char, i) for i, char in enumerate(target_characters)])

encoder_input_data = np.zeros(
    (len(input_texts), max_encoder_seq_length, num_encoder_tokens),
    dtype='float32')
decoder_input_data = np.zeros(
    (len(input_texts), max_decoder_seq_length, num_decoder_tokens),
    dtype='float32')
decoder_target_data = np.zeros(
    (len(input_texts), max_decoder_seq_length, num_decoder_tokens),
    dtype='float32')

for i, (input_text, target_text) in enumerate(zip(input_texts, target_texts)):
    for t, char in enumerate(input_text):
        encoder_input_data[i, t, input_token_index[char]] = 1.
    for t, char in enumerate(target_text):
        # decoder_target_data is ahead of decoder_input_data by one timestep
        decoder_input_data[i, t, target_token_index[char]] = 1.
    if t > 0:
```

```

# decoder_target_data will be ahead by one timestep
# and will not include the start character.
decoder_target_data[i, t - 1, target_token_index[char]] = 1.

# Define an input sequence and process it.
encoder_inputs = Input(shape=(None, num_encoder_tokens))
encoder = LSTM(latent_dim, return_state=True)
encoder_outputs, state_h, state_c = encoder(encoder_inputs)
# We discard `encoder_outputs` and only keep the states.
encoder_states = [state_h, state_c]

# Set up the decoder, using `encoder_states` as initial state.
decoder_inputs = Input(shape=(None, num_decoder_tokens))
# We set up our decoder to return full output sequences,
# and to return internal states as well. We don't use the
# return states in the training model, but we will use them in inference.
decoder_lstm = LSTM(latent_dim, return_sequences=True, return_state=True)
decoder_outputs, _, _ = decoder_lstm(decoder_inputs,
                                     initial_state=encoder_states)
decoder_dense = Dense(num_decoder_tokens, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)

# Define the model that will turn
# `encoder_input_data` & `decoder_input_data` into `decoder_target_data`
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

# Run training
model.compile(optimizer='rmsprop', loss='categorical_crossentropy')
model.fit([encoder_input_data, decoder_input_data], decoder_target_data,
        batch_size=batch_size,
        epochs=epochs,
        validation_split=0.2)

# Save model
model.save('s2s.h5')

# Next: inference mode (sampling).

```

```

# Here's the drill:
# 1) encode input and retrieve initial decoder state
# 2) run one step of decoder with this initial state
# and a "start of sequence" token as target.
# Output will be the next target token
# 3) Repeat with the current target token and current states

# Define sampling models
encoder_model = Model(encoder_inputs, encoder_states)

decoder_state_input_h = Input(shape=(latent_dim,))
decoder_state_input_c = Input(shape=(latent_dim,))
decoder_states_inputs = [decoder_state_input_h, decoder_state_input_c]
decoder_outputs, state_h, state_c = decoder_lstm(
    decoder_inputs, initial_state=decoder_states_inputs)
decoder_states = [state_h, state_c]
decoder_outputs = decoder_dense(decoder_outputs)
decoder_model = Model(
    [decoder_inputs] + decoder_states_inputs,
    [decoder_outputs] + decoder_states)

# Reverse-lookup token index to decode sequences back to
# something readable.
reverse_input_char_index = dict(
    (i, char) for char, i in input_token_index.items())
reverse_target_char_index = dict(
    (i, char) for char, i in target_token_index.items())

def decode_sequence(input_seq):
    # Encode the input as state vectors.
    states_value = encoder_model.predict(input_seq)

    # Generate empty target sequence of length 1.
    target_seq = np.zeros((1, 1, num_decoder_tokens))
    # Populate the first character of target sequence with the start character.
    target_seq[0, 0, target_token_index['\t']] = 1.

```

```
# Sampling loop for a batch of sequences
# (to simplify, here we assume a batch of size 1).
stop_condition = False
decoded_sentence = ""
while not stop_condition:
    output_tokens, h, c = decoder_model.predict(
        [target_seq] + states_value)

    # Sample a token
    sampled_token_index = np.argmax(output_tokens[0, -1, :])
    sampled_char = reverse_target_char_index[sampled_token_index]
    decoded_sentence += sampled_char

    # Exit condition: either hit max length
    # or find stop character.
    if (sampled_char == '\n' or
        len(decoded_sentence) > max_decoder_seq_length):
        stop_condition = True

    # Update the target sequence (of length 1).
    target_seq = np.zeros((1, 1, num_decoder_tokens))
    target_seq[0, 0, sampled_token_index] = 1.

    # Update states
    states_value = [h, c]

return decoded_sentence
```

ANEXO 3

CHATS EN PÁGINAS WEB

Chat de CLARO



Figura 74 Asistente virtual V (CLARO)



Figura 75 Asistente virtual (CLARO)



Figura 76 Asistente virtual II (CLARO)



Figura 77 Asistente virtual III (CLARO)



Figura 78 Asistente virtual IV (CLARO)



Figura 79 Asistente virtual IV (CLARO)

Chat SALUDSA

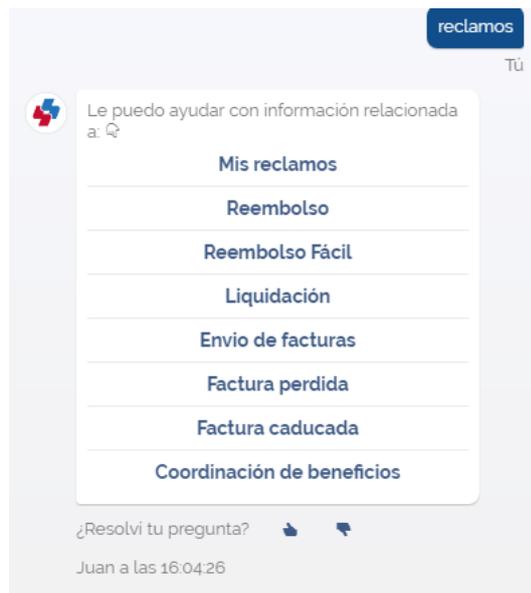


Figura 80 Asistente virtual VII (Saludsa)

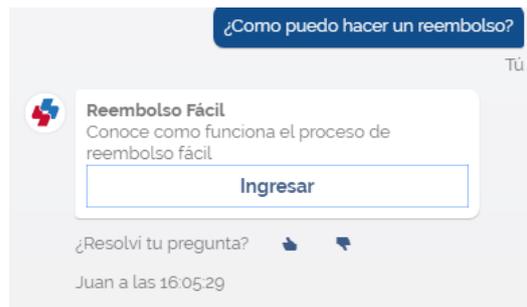


Figura 81 Asistente virtual VIII (Saludsa)

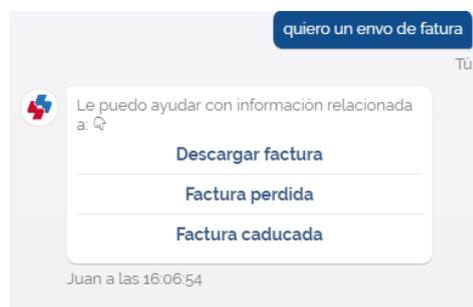


Figura 82 Asistente virtual IX (Saludsa)

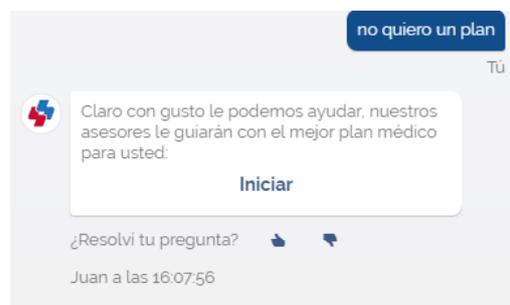


Figura 83 Asistente virtual X (Saludsa)

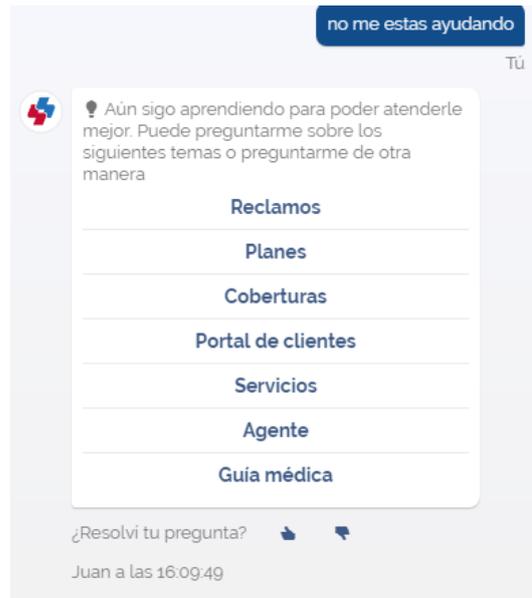


Figura 84 Asistente virtual XI (Saludsa)

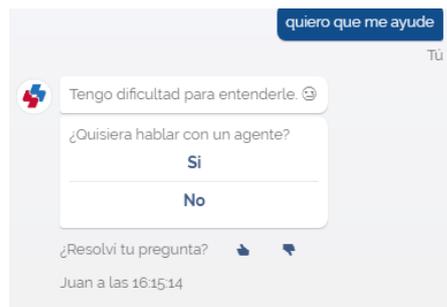


Figura 85 Asistente virtual XIV (Saludsa)



Figura 86 Asistente virtual XIII (Saludsa)

Chat Banco Pacífico



Figura 87 Asistente virtual XVI (Banco del Pacífico)



Figura 88 Asistente virtual XVII (Banco del Pacífico)

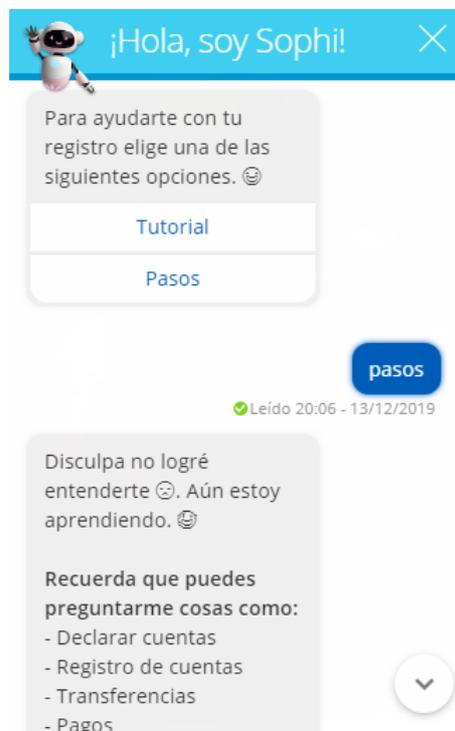


Figura 89 Asistente virtual XVIII (Banco del Pacífico)

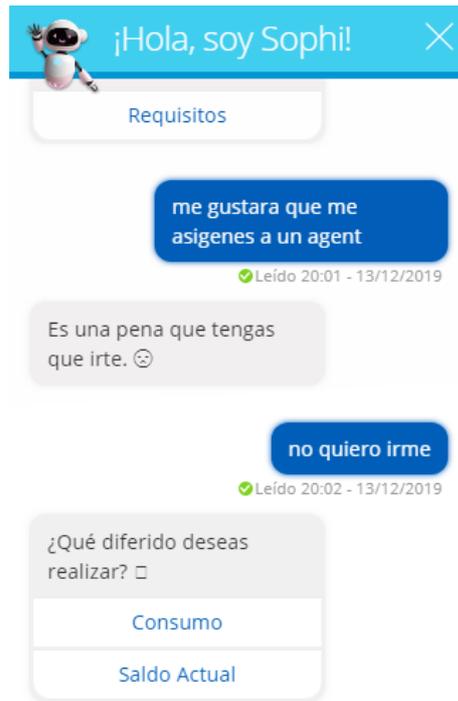


Figura 90 Asistente virtual XIX (Banco del Pacifico)



Figura 91 Asistente virtual XX (Banco del Pacifico)

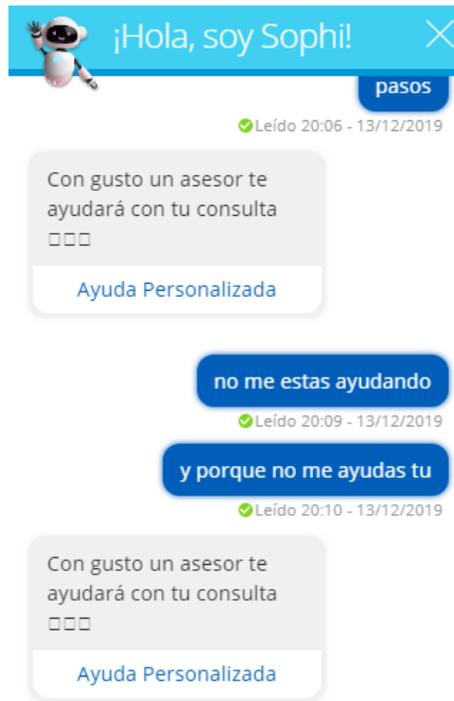


Figura 92 Asistente virtual XXI (Banco del Pacifico)



Figura 93 Asistente virtual XXII (Banco del Pacifico)

Chat Banco de Loja

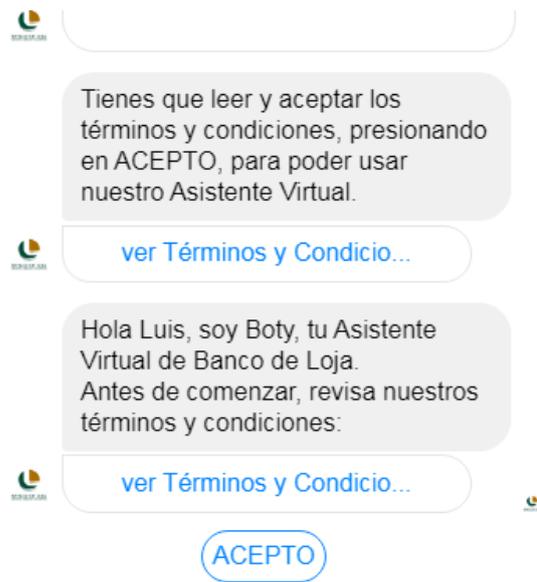


Figura 94 Asistente virtual XXIV (FACEBOOK)



Figura 95 Asistente virtual XXV (FACEBOOK)

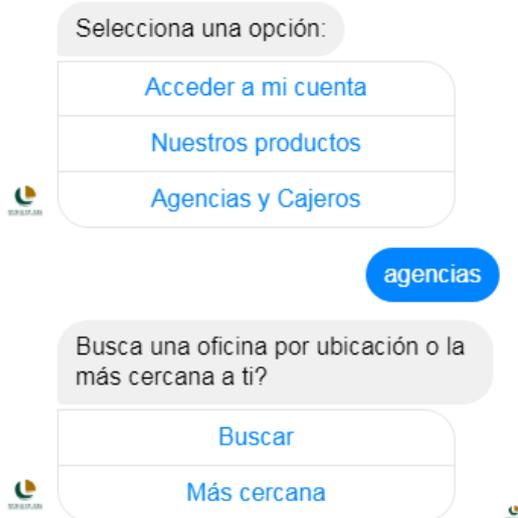


Figura 96 Asistente virtual XXVI (FACEBOOK)



Figura 97 Asistente virtual XXVII (FACEBOOK)



Figura 98 Asistente virtual XXVIII (FACEBOOK)



Figura 99 Asistente virtual XXIX (FACEBOOK)

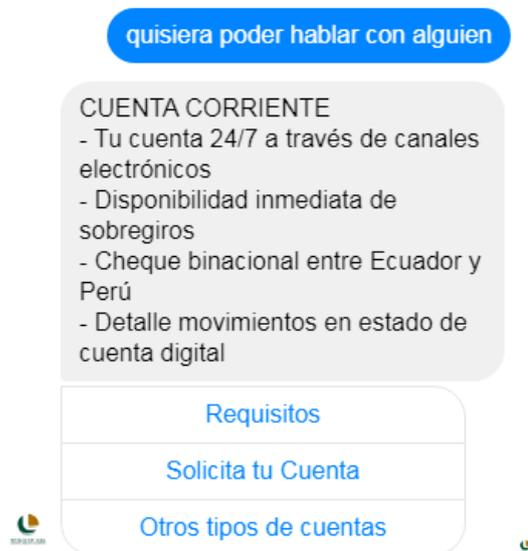


Figura 100 Asistente virtual XXX (FACEBOOK)

DESARROLLO DE UN CHATBOT WEB COMO ASISTENTE DE VENTAS

Luis Felipe Machado Redrobán

Universidad Internacional de la Rioja, Logroño (España)

20 de Febrero de 2020



RESUMEN

En este proyecto se desarrolla un chatbot para una empresa en Ecuador. Dicho chatbot tendrá como objetivo atender a los visitantes del sitio web, cuando estos establezcan una comunicación a través de un chat en línea. Se utiliza metodología SCRUM, para el desarrollo del software. El chatbot será específicamente, un asistente de ventas, que proporcionará información a los potenciales clientes y tratará de tomar los datos de los mismos, como nombre, correo electrónico y/o teléfono. El chatbot se desarrolla, utilizando librerías y frameworks libres y gratuitos que permiten procesamiento de lenguaje natural y estructurar contextos de conversaciones. Para el entrenamiento se redacta un corpus de acuerdo a la visión, misión, productos, servicios y objetivos de la empresa. El chatbot es entrenado y es integrado al chat en línea del sitio web de la empresa. Finalmente se compara con otros chatbots de otras páginas web de empresas en el Ecuador.

PALABRAS CLAVE

Chatbot, inteligencia artificial, procesamiento de lenguaje natural, sitio web, software.

I. INTRODUCCIÓN

De acuerdo al diccionario, la comunicación es la “Transmisión de señales mediante un código común entre el emisor y el receptor.” Los seres humanos utilizamos un complejo sistema de códigos al que le denominamos lenguaje.

El lenguaje nos permite comunicarnos, describir el mundo que nos rodea y transmitirlo a otros seres humanos. En la actualidad la comunicación, no solo se establece entre seres humanos, es decir el emisor y receptor ya no son solamente seres humanos, ahora pueden ser humanos y máquinas. Desde el diseño y construcción de computadoras el ser humano ha desarrollado varios lenguajes para poder comunicarse con las máquinas y poder programar su funcionamiento. Sin embargo el lenguaje de comunicación entre los seres humanos y las máquinas no es el lenguaje natural del ser humano, ya que la máquina no entiende nuestra forma de expresarnos naturalmente, como lo hacemos con otros seres humanos. Es así que a partir de la época de 1950, se empieza un estudio, a través de una serie de investigaciones para lograr el objetivo de que las computadoras puedan entender el lenguaje natural de los seres humanos.

La posibilidad de que las máquinas entiendan el lenguaje natural

de los seres humanos permite satisfacer muchas necesidades identificadas, como por ejemplo el procesamiento de datos y la clasificación de información. Según un estudio realizado por IBM, 2.5 EXABYTES de datos fueron generados cada día en 2017 [1] y en la actualidad esa cifra ha ido en aumento. Por supuesto no toda la información tiene el interés de un ser humano, aún así si se reduce los datos a los temas de interés, sería imposible para un ser humano procesar toda la cantidad de datos que se genera, leer todos los documentos, obtener resúmenes, analizar los datos para convertirlos en información, clasificar la información más relevante o de interés, etc.

Adicionalmente, el poder interactuar con las máquinas de manera natural y rápida, para que éstas, sirvan de ayuda para resolver problemas, brinden servicios o atiendan requerimientos del ser humano.

Por estos motivos se ha fortalecido el desarrollo tecnologías y aplicaciones que permitan a las máquinas entender de mejor manera el lenguaje de los seres humanos, ya sea que dicho lenguaje esté contenido en un texto de manera escrita, en una grabación de audio o que se esté llevando a cabo una comunicación en tiempo real, en la cual una máquina asuma el rol como emisor y/o receptor. En este último caso se han creado aplicaciones denominadas “agentes conversacionales” o como se los conoce coloquialmente: “chatbots”.

Un chatbot es un agente inteligente que puede comunicarse con los seres humanos a través del lenguaje natural del ser humano. Precisamente de esto trata este proyecto, del estudio de chatbots y el diseño y construcción de un chatbot capaz de atender y brindar un servicio al cliente en un sitio web de una empresa, asumiendo el rol de un asistente de ventas. Este trabajo tiene como objetivo principal, desarrollar un chatbot, para que el visitante de una página web de una empresa, pueda ser atendido de manera inmediata por un agente inteligente mediante comunicación escrita.

Con el auge de la Inteligencia Artificial, los humanos desean comunicarse con las máquinas utilizando el lenguaje natural propio de los seres humanos. En consecuencia se han desarrollado tecnologías capaces de hacer que las máquinas aprendan y procesen el lenguaje natural de los seres humanos. Sin embargo este trabajo apenas inicia, ya que las expresiones que se utilizan en conversaciones entre seres humanos son muy complejas.

En este trabajo se analiza el estado del arte de los chatbots y se describen algunas librerías, para el procesamiento de lenguaje natural y frameworks capaces de permitir la creación de chatbots.

Este trabajo está enfocado en el diseño un chatbot como asistente de ventas, utilizando frameworks y librerías existentes y enfocándose estrictamente en esa área profesional, limitando así, la capacidad de respuestas dentro de dicha área. Es importante entrenar el chatbot para que tenga buena capacidad de respuesta [2]

El trabajo consta de:

- Hacer un estudio del estado del arte de los chatbots.
- Hacer un breve estudio de las tecnologías de inteligencia artificial.
- Describir distintas herramientas o frameworks para la construcción de chatbots.
- Entrenar el chatbot utilizando DIALOGFLO [3] y KERAS [4]
- Validar con frases simples.
- Validar con frases más complejas.
- Desarrollar un flujo de conversación.
- Evaluar los resultados.
- Comparar con chatbots existentes en páginas web de empresas ecuatorianas.
- Obtener conclusiones.

La palabra chatbot se compone de dos palabras: chat y bot. Según el diccionario **chat**, es “El intercambio de mensajes en tiempo real, entre uno o más usuarios, en una red de computadoras” [5] y **bot**, es una abreviatura de “robot”, que puede actuar como un programa autónomo en la red de computadoras e interactuar con otros usuarios [6]. Un chatbot es un sistema de dialogo en línea entre un humano y un computador utilizando el lenguaje natural, la comunicación puede realizarse a través de lenguaje oral y escrito (Jía, 2003). Los chatbots o agentes conversacionales son una aplicación del procesamiento del lenguaje natural.

El Procesamiento de lenguaje natural es un área de estudio de la informática, que incluye a la inteligencia artificial y lingüística, cuyo objetivo es que las máquinas entiendan el lenguaje como lo hace otro ser humano y así se pueda establecer una comunicación entre los humanos y máquinas mediante el uso del lenguaje natural. El lenguaje natural es el idioma o lengua hablada o escrita utilizada por los seres humanos para la comunicación [7].

De acuerdo a Ganegedara [1], con la ayuda del PLN se puede tener aplicaciones como:

Tokenization

- Desambiguar de palabras (*WSD Word-sense Disambiguation*)
- Reconocimiento de nombre de una entidad (*NER Named Entity Recognition*)
- Etiquetado de partes de la oración (*POS Part-of-Speech*)
- Clasificación de sinopsis (*Sentence/Synopsis classification*)
- Generación de Lenguaje (*Language Generation*)
- Preguntas y Respuestas (*QA Question Answering*)
- Traductores automáticos (*MT Machine Translation*)

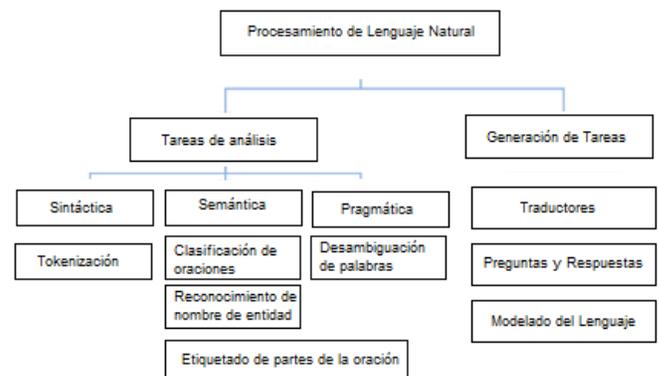


Figura 1 Procesamiento de Lenguaje Natural [1]

El enfoque tradicional, los problemas de PLN se resuelven utilizando algoritmos de *machine learning*. Dichos algoritmos, se enfocan en la extracción de características de las representaciones de los datos para realizar el entrenamiento de un modelo. Para este entrenamiento se requieren dos tipos de datos: datos de test y datos de validación. Así se pone a prueba el modelo entrenado [1].

Sin embargo el *deep learning* está revolucionando los campos de Visión de Computador, Reconocimiento de Habla y por supuesto Procesamiento del Lenguaje Natural. [8]

Gracias a los algoritmos de *deep learning*, se pueden crear modelos que permiten analizar gran cantidad de datos eliminando la complejidad de *Feature Engineering* y pasar del tradicional modelo de *machine learning* a un nuevo modelo que permita tener un mejor proceso de aprendizaje [1].

Según, Cahn [9] un chatbot debe tener tres funciones principales:

- **Agente de Diálogo:** Proporciona la función de comprensión, que permite al usuario a través de una interfaz GUI (*Graphical User Interface*) ingresar texto para obtener ayuda.
- **Agente Racional:** Proporciona la función y competencia, para administrar las solicitudes del usuario y emitir las respuestas.
- **Agente encarnado:** Proporciona la función de presencia, personificando a un ser humano, para generar confianza en el usuario, es por eso que debe tener un nombre de persona como por ejemplo los famosos agentes ELIZA, ALICE, etc.

En 1950 Alan Turing, plantea la siguiente pregunta ¿Las máquinas pueden pensar? Para resolver esta pregunta, se diseña un juego en el cual se realizan preguntas a seres humanos y máquinas, si las personas que realizan las preguntas no pueden diferenciar si es una persona la que responde o es una máquina, entonces, la respuesta es “Las máquinas pueden pensar” [9].

Así nace ELIZA en 1966, el primer chatbot que podía responder preguntas y superar el denominado “*Turing Test*”, creada por Joseph Weizenbaum en el MIT

Chatbots histories:

- **ELIZA** fue el primer paso para el procesamiento del lenguaje natural, basado en un modelo simple de expresiones regulares y comodines (*wilcards*) que buscaba reconocer palabras, siguiendo ciertas reglas dentro de una pregunta y responder con frases predefinidas. ELIZA ha

servido de inspiración para la creación de nuevos y más modernos chatbots [9].

- **PARRY**, creado en 1972, por Kenneth Colby de Stanford, un chatbot que suplantó un paciente con un cuadro de esquizofrenia y paranoia [10]
- **A.L.I.C.E.** (*Artificial Linguistic Internet Computer Entity*), desarrollado en 1955 por Dr. Richard Wallace, es un chatbot inspirado en ELIZA. A.L.I.C.E. fue escrito en AIML (*Artificial Intelligence Markup Language*). AIML es una extensión de XML, que fue desarrollada por el mismo Richard Wallace. Este chatbot ha sido 3 veces ganador del premio Loebner, que es una competición de chatbots que deben superar el “*Turing Test*” y demostrar ser lo más inteligentes. (Wallace). Ahora es parte del proyecto PANDORA, un proyecto comercial que permite crear chatbots para diferentes aplicaciones [9].

Aplicaciones de un chatbot:

Existen muchas aplicaciones en la actualidad en la que los chatbots cumplen importante funciones, entre ellas podemos mencionar:

- Asistentes en teléfonos o computadoras.
- Asistentes de sitios web.
 - Atención de servicio al cliente.
 - Ventas.
 - Procesos de soporte.
- Asistentes telefónicos.
- Agentes de SMS.
- Agentes de redes sociales (WhatsApp, Facebook, Twitter, etc).
- Asistentes en juegos.
- Asistentes de respuestas en email, entre otras.
- Traductores.

Tipos de Chatbots:

De acuerdo al incremento de aplicaciones, los desarrolladores han creado diferentes tipos de chatbots, en base a lo mencionado, se puede citar una clasificación propuesta por la revista *chatbots magazine* [11].

- **Single-feature Chatbot:** chatbots, que poseen una sola característica. Tienen limitadas funcionalidades para un uso específico y simple. Por ejemplo, un traductor de texto simplificado.

- **Proactive Chatbots:** chatbots, que colocan información al usuario, en lugar de esperar preguntas. No son interactivos, simplemente ponen a disposición del usuario cierta información en determinados momentos.
- **Group Chatbots:** chatbots que interactúan con un grupo de usuarios. Responden a requerimientos específicos de los usuarios. Por ejemplo en juegos en línea, asignan recursos a los usuarios.
- **Simplification Chatbots:** chatbots con una interfaz simple, pero que permite resolver trámites automatizados. Por ejemplo pago de tickets de parqueadero.
- **Entertainment Chatbots:** chatbots, cuya funcionalidad es establecer una conversación con el usuario.
- **Personal Assistants:** chatbots que combinan diferentes funcionalidades y sirve como asistente de tareas para un usuario. Por ejemplo asistentes de teléfonos inteligentes (*smartphones*).
- **Optimization Chatbots:** chatbots utilizados en sitios web. Estos chatbots son entrenados como asistentes de ventas o servicio al cliente, interactúan con los usuarios a través de conversaciones dentro de un ámbito específico.

Los chatbots son aplicaciones que corren sobre las redes de datos y que requieren el intercambio de información entre el usuario y el servidor, en los cuales se aloja su código fuente, de ahí que siempre va a ser necesario que exista conectividad entre los diferentes dispositivos.

Para que los chatbots puedan responder de manera adecuada, estos deben ser muy bien entrenados. El proceso de entrenamiento puede ser complejo y siempre se requerirá de nuevos entrenamientos. Si el proceso de entrenamiento ha sido realizado de manera satisfactoria, el chatbot tendrá una gran capacidad de respuestas. Adicionalmente, el hecho de que un chatbot pueda responder a preguntas, no significa que pueda establecer una conversación con un ser humano. La conversación es un proceso complejo de intercambio de frases que no necesariamente está basada en preguntas y respuestas, sino más bien muchas de estas frases son ideas, pensamientos, conclusiones que están relacionadas entre sí. Por este motivo, una gran limitante a vencer es que el chatbot pueda entender el contexto de una frase que le permita ubicarse dentro del sentido de la conversación.

III. OBJETIVOS Y METODOLOGÍA

Objetivo general:

Desarrollar un chatbot, que cumplirá la función de un asistente de ventas para los visitantes de la página web, propiedad de la empresa SIDEVOX S.A.:

<https://www.contactvox.com>.

Objetivos específicos:

- Incorporar las funcionalidades de un chatbot al sitio web de la empresa SIDEVOX S.A. para la atención de ventas.
- Probar varias herramientas que permiten la construcción de chatbots, con el objetivo de desarrollar un chatbot que brinde buena comprensión del lenguaje natural.
- Integrar de ser necesario más de una herramienta en el desarrollo del chatbot, para mejorar la comprensión del lenguaje natural y la generación de respuestas.
- Integrar el chatbot desarrollado al sistema de comunicaciones existente en dicha empresa.
- Brindar un servicio automático de respuestas 24/7.
- Brindar una guía rápida de los productos y servicios que la empresa brinda a los visitantes de su sitio web.
- Brindar respuestas a preguntas frecuentes de los productos y servicios que la empresa brinda a los visitantes de su sitio web.
- Reducir lo más posible repuestas como “no entiendo”, “podrías repetirlo”, es decir tratar la máximo que el chatbot ofrezca fluidez en la conversación
- Brindar respuestas inmediatas y guiar en el acercamiento a la empresa a los visitantes de su sitio web.
- Comparar el chatbot desarrollados con otros chatbots de páginas ecuatorianas.

Metodología de trabajo

La metodología que se aplica para el desarrollo del chatbot es SCRUM. De acuerdo al libro ScrumStudy [12], “Un proyecto SCRUM implica un esfuerzo de colaboración para crear un nuevo Producto o, servicio, o cualquier otro resultado como se define en el Declaración de la Visión del Proyecto. Los proyectos se ven afectados por las limitaciones de tiempo, costo, alcance, calidad, recursos, capacidades organizativas, y otras limitaciones que los hacen difíciles de planificar, ejecutar, administrar y finalmente tener éxito. Sin embargo, la implementación exitosa de los resultados de un proyecto acabado le proporciona ventajas económicas significativas a una organización”.

De acuerdo ScrumStudy [12], los procesos SCRUM abordan las actividades y el flujo específico de un proyecto dividido en 19 procesos que se agrupan en cinco fases, 3 fases se abordan en esta sección y dos en las siguientes:

Iniciar

Visión del producto: desarrollar un chatbot que brinde atención inmediata a sus visitantes, mediante mensajería instantánea, que sea capaz de establecer una conversación coherente y guiar al visitante a través de los productos y servicios que brinda la empresa, transmitiendo calidad de servicio e innovación.

Planear y estimar

Como visitante debería:

- tener el servicio de un asistente robot que responda el chat de manera inmediata a cualquier hora.
- tener respuestas coherentes.
- Tener una asesoría de productos y servicios.
- Tener respuestas de la empresa.
- Tener respuestas concretas a preguntas.

Como dueño del sitio debería

- Atender de manera inmediata a un *INTENT* de comunicación escrita en el sitio web.
- Brindar una asesoría de los productos y servicios de la empresa.
- Captar datos del visitante del sitio.
- Guardar los datos del visitante en una base de datos.

Creación del proyecto

- Crear una cuenta de GOOGLE.
- Crear un proyecto en DIALOGFLOW.

Entrenamiento de corpus de servicio

- Redactar el corpus de saludos.
- Redactar el corpus de despedidas.
- Redactar el corpus de información general de la empresa.
- Entrenar el corpus de saludos.
- Entrenar el corpus de despedidas.
- Entrenar el corpus de información general de la empresa.
- Probar el corpus de saludos.
- Probar el corpus de despedidas.
- Probar el corpus de información general de la empresa.

Entrenamiento de corpus de técnico

- Identificar los productos y servicios.
- Redactar el corpus de productos.
- Redactar el corpus de servicios.
- Entrenar el corpus de productos.
- Entrenar el corpus de servicios.
- Probar el corpus de productos.
- Probar el corpus de servicios.

Contexto de la conversación

- Desarrollar contexto de la conversación.
- Diagramar los contextos
- Analizar frases fuera de contexto.

Errores de la conversación

- Redactar el corpus de errores de conversación.
- Entrenar el corpus de errores de conversación.
- Probar el corpus de errores de conversación.

Detección de visitantes no deseados

- Redactar el corpus de errores de conversación.
- Entrenar el corpus de errores de conversación.
- Probar el corpus de errores de conversación.

Corpus Avanzado

- Redactar el corpus de frases más elaboradas.
- Entrenar el corpus de frases más elaboradas.
- Probar el corpus frases más elaboradas.

Integración

- Diagramar el esquema de integración
- Incorporar el chatbot con la plataforma existente.

Implementar

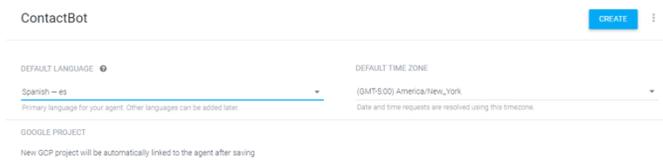


Figura 2 Creación de un Agente [13]

No existe un corpus entrenado de acuerdo a la declaración del *business case*, por tal motivo, se debe redactar un corpus

que se entrenará en el chatbot. A continuación se detalla una parte del corpus general del chatbot.

Ejemplo:

- Hola
- Buen día
- Buenos días
- Buenas tardes
- ¿Cuántos años están en el mercado?
- ¿A qué se dedican?
- ¿Cuáles son sus servicios?
- ¿De donde son?
- ¿Quiénes son Uds.?

Se realizaron dos tipos de entrenamientos, el primero en DIALOGFLOW y un segundo corpus fue entrenado en KERAS. Para este último se configuró una red neuronal, con el modelo seq2seq tomando del ejemplo del proyecto KERAS en GITHUB [14]

```
Number of samples: 200
Number of unique input tokens: 40
Number of unique output tokens: 27
Max sequence length for inputs: 32
Max sequence length for outputs: 39
Train on 160 samples, validate on 40 samples
Epoch 1/100
160/160 [=====] - 3s 20ms/step - loss: 2.5403 - val_loss: 2.2344
Epoch 2/100
160/160 [=====] - 0s 2ms/step - loss: 2.4354 - val_loss: 1.9555
Epoch 3/100
160/160 [=====] - 0s 2ms/step - loss: 2.0456 - val_loss: 1.8901
Epoch 4/100
160/160 [=====] - 0s 2ms/step - loss: 1.9959 - val_loss: 1.7688
Epoch 5/100
160/160 [=====] - 0s 2ms/step - loss: 1.8851 - val_loss: 1.8145
Epoch 6/100
160/160 [=====] - 0s 2ms/step - loss: 1.9155 - val_loss: 1.6788
Epoch 7/100
160/160 [=====] - 0s 2ms/step - loss: 1.7678 - val_loss: 1.6053
Epoch 8/100
160/160 [=====] - 0s 2ms/step - loss: 1.8400 - val_loss: 1.6764
Epoch 9/100
160/160 [=====] - 0s 2ms/step - loss: 1.7541 - val_loss: 1.5558
Epoch 10/100
160/160 [=====] - 0s 2ms/step - loss: 1.7286 - val_loss: 1.6075
Epoch 11/100
160/160 [=====] - 0s 2ms/step - loss: 1.6813 - val_loss: 1.5150
Epoch 12/100
160/160 [=====] - 0s 2ms/step - loss: 1.6395 - val_loss: 1.6427
```

Figura 3 Entrenamiento en KERAS [15]

IV. CONTRIBUCIÓN

El proyecto desarrollado en este trabajo realiza las siguientes contribuciones:

- Detallar el estado de arte de los chatbots, para poner en conocimiento del lector las posibles tecnologías y herramientas existentes, así como los beneficios, ventajas y limitaciones de los chatbots en la actualidad.
- Redactar un corpus orientado a ser un asistente de ventas. No existe un corpus para este trabajo, ya que es muy específico el tema del conocimiento. Este corpus puede servir de ejemplo base o guía para otros proyectos de similares características.

- Realizar un entrenamiento detallado y muy bien documentado utilizando dos herramientas, y mostrar el cómo se realiza el entrenamiento en cada una de ellas. Al incorporar una segunda herramienta de entrenamiento se potencializa la capacidad de procesamiento de lenguaje natural del chatbot y al mismo tiempo se resuelve en cierta medida el problema de ambigüedad.
- El desarrollo de un flujo de conversación que permita tener el control de la misma al chatbot. El chatbot debe ser consistente en sus respuestas y guiar la conversación, caso contrario la comunicación no fluirá y el visitante se desanimará de continuar interactuando con un robot.
- El principal aporte es el dotar de una herramienta de inteligencia artificial a la organización que auspició este proyecto ya que la misma no cuenta con un sistema de esta naturaleza, permitiéndole potencializar su sitio web e incrementar sus probabilidades de cierre de ventas, al mostrar innovación y atención inmediata. No obstante, el chatbot desarrollado en este proyecto sirve de base para otras organizaciones de similares características que deseen incorporar a sus sitios web, inteligencia artificial. Por lo cual este trabajo se convierte en una guía indispensable que toda pequeña empresa ya sea de Ecuador u otro país, puede seguir para incorporar un chatbot en su sitio web.

V. RESULTADOS EVALUACIÓN

De acuerdo a la metodología SCRUM [12] en la fase 4 de REVISIÓN se pone a prueba el producto y se obtienen los siguientes resultados:

- El chatbot de este trabajo tiene una interfaz gráfica solo en texto (Debería en un futuro tener un GUI).
- El chatbot no presenta lista de opciones por lo cual abre la posibilidad de una conversación abierta como lo haría un humano.
- Es capaz de responder a preguntas.
- Solicita datos.
- Es capaz de detectar usuarios no deseados.
- Entiende cuando un usuario no quiere un servicio y no se equivoca presentando lista de servicios.
- Permite establecer una conversación como lo haría un humano.
- No se integra con bases de datos ni otros sistemas (Debería en un futuro).

VI. DISCUSIÓN O ANÁLISIS DE RESULTADOS

Finalmente en la fase 5 de la metodología SCRUM [12] se entrega el resultado.

Los resultados nos muestran un chatbot capaz de establecer una conversación con un ser humano. El chatbot cumple con el objetivo de ser un asistente de ventas, responde preguntas de los diferentes productos y servicios y toma datos del potencial cliente.

Si el visitante sigue la lógica de la conversación que el chatbot propone, dicha conversación se desarrollará de manera natural y el chatbot habrá cumplido su objetivo de manera adecuada.

El problema se presenta cuando un visitante no sigue la lógica de la conversación, expresa mal sus ideas, mezcla las ideas o entra en discusión. En este caso el entrenamiento del chatbot se basa en generar diseños de contextos que le permitan retomar el control de la conversación y estar preparado para frases inesperadas.

VII. CONCLUSIONES

Existen muy pocos sitios web de empresas en Ecuador con chatbots, se busca en alrededor de 20 empresas grandes pero no tienen implementado un chatbot. Se realiza la búsqueda en páginas de bancos, supermercados, aseguradoras, empresas de telecomunicaciones.

Los chatbots encontrados en las pocas páginas que los contenían, tienen una interfaz gráfica interesante pero se enfocan en ser chatbots de preguntas y respuestas, no se puede establecer una conversación como con un ser humano.

El estado del arte ha avanzado mucho en los últimos años. Las plataformas de desarrollo de chatbots están mejorando la forma de construir este tipo de aplicaciones, sin embargo sigue siendo un trabajo complejo. Existen herramientas actuales que facilitan el procesamiento del lenguaje natural, un aspecto bastante complicado, pero gracias a estas herramientas se vuelve más sencillo.

Las aplicaciones de un chatbot son bastante amplias, pero el corpus debe ser entrenado de acuerdo a dichas aplicaciones. Es decir en el ámbito en el que el chatbot se va a desempeñar hay que desarrollar un corpus exclusivo, ya que el mismo obedece a la visión o misión del proyecto y si es un chatbot comercial para una empresa, se debe alinear a la misma y aprender de sus productos, servicios y su cultura corporativa que es la que se

espera transmitir en las comunicaciones. Es muy importante que el chatbot se personalice y profesionalice, es decir que tenga un nombre y se presente de manera adecuada, generando un estado de presencia que genera seguridad en un ser humano. Adicionalmente la profesionalización le brinda la guía del entrenamiento. La idea no es engañar al usuario sino brindarle confianza, para que lo expresado por el chatbot, sea aceptado por el usuario. Aunque el usuario sea consciente de que está conversando con un robot, debe conocer que el mismo tiene un nombre y asume el rol de una persona, lo cual facilitará la comunicación.

Un aspecto bastante complicado del chatbot, fue el diseñar los contextos y la forma de entrenamiento de los *INTENTS*, aparentemente es sencillo crear un *INTENT*, pero cuando se tienen conversaciones extensas, se tiende a caer en ambigüedades y controlar la gran mayoría de éstas requiere un esfuerzo grande, para que el agente inteligente no pierda el control del diálogo y pueda obtener las respuestas correctas de los usuarios. Este puede ser uno de los retos más grandes a vencer el desarrollo de chatbots, el hecho de diseñar un sistema capaz de mantener una conversación coherente con un usuario. El chatbot no es solo un sistema que realiza el procesamiento del lenguaje natural, se requiere establecer contextos, flujos de conversación que siguen siendo impredecibles debido a las respuestas propias de un ser humano. Se debe tratar de controlar al máximo el hilo de la conversación y volver a encaminarla.

El agente debe conocer en todo momento por donde está encaminada la conversación, hacia donde se dirige y es más debe ser el agente quien dirija la misma y no darle tanta libertad al usuario. Por este motivo la mayoría de chatbots comerciales se restringen a ser agentes de preguntas y respuestas, es decir no establecen una conversación con el usuario, sino solo se limitan a responder con opciones a las preguntas que hace un usuario.

La obtención de los datos para el entrenamiento, es uno de los aspectos más complicados, ya que la empresa tenía cierta información y muchas veces no sabe lo que quiere y transmitir.

Un dato curioso que se evidenció con algunas pruebas realizadas, es que cuando no se le indica a un usuario que se trata de un agente inteligente, la conversación tiende a fluir de manera natural y pudiendo esta llegar a pasar desapercibido que se trata de una conversación entre una persona y un agente inteligente. Sin embargo cuando el usuario conoce que se trata de un chatbot, tiende a buscar la falla del mismo, tratando de poner a prueba el software incluso de una manera absurda, que

no lo haría si supiese que se trata de una persona que es con la que está conversando. Por este motivo se consideró desarrollar el módulo de detección de visitantes no deseados y se generan ciertas preguntas respecto al comportamiento de los seres humanos frente a las máquinas.

Finalmente el chatbot debe estar muy bien entrenado, para que pueda ser capaz de responder al usuario y continuar el diálogo. El corpus fue prediseñado manualmente tomando en cuenta algunas frases, que sirvan para el proyecto. Se escribieron apenas alrededor de 100 frases con el objetivo de que sea una muestra. Mientras más ejemplos se tengan, el modelo se puede entrenar de mejor manera. Por lo tanto una de las tareas que demanda mayor esfuerzo es la de armar un corpus. Esta tarea es vital si queremos que un chatbot tenga buenas capacidades de respuesta.

APÉNDICES

Apéndice A

Resultados del entrenamiento del chatbot:

USER SAYS COPY CURL
buena tarde



DEFAULT RESPONSE

Gracias por su comunicación. Mi nombre es ContactBot, esperamos que su visita sea placentera, yo seré su asistente virtual durante la misma. ¿En qué podemos servirle?

Figura 4 Prueba de saludo [13]

Agent

USER SAYS COPY CURL
quien son uds



DEFAULT RESPONSE

Gracias por su comunicación, déjame contarte que SIDEVOX es una empresa de R&D y llevamos 11 años presentes en el mercado. Nuestros productos y servicios ahora están presentes en 4 países.

Figura 5 Prueba de datos de empresa [13]

visitante	chatbot
hola, me pueden ayudar	Gracias por su comunicación. Mi nombre es ContactBot, esperamos que su visita sea placentera, yo seré su asistente virtual durante la misma. ¿En qué podemos servirle?
¿Tienen sistemas de comunicaciones unificadas?	Somos especialistas en telefonía IP, podemos recomendarte, nuestro sistema de comunicaciones CONTACTVOX UCS Tenemos dos tipos de servicios en la nube y on premise, de ¿cuál de ellas requiere información?
creo que puede ser la nube	nuestro sistema se aloja en la nube de Amazon Ohio Para poder asesorarte de mejor manera necesito conocer algunos datos, ¿Estás de acuerdo ?

Figura 6 Prueba de contextos [13]

REFERENCIAS

VIII. BIBLIOGRAFÍA

- [1] Ganegedara., T. (May 2018). *Natural Language Processing with TensorFlow*.
- [2] Bang Liu, H. W. (n.d.). Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus.
- [3] Google. (n.d.). *Dialogflow*. Retrieved from <https://dialogflow.com/>
- [4] KERAS. (n.d.). *KERAS DOCUMENTATION*. Retrieved from <https://keras.io/>
- [5] DICTIONARIES, E. O. (n.d.). <https://en.oxforddictionaries.com>. Retrieved from <https://en.oxforddictionaries.com/definition/chat>
- [6] Vogel, J. (2017). *Chatbots: Development and Applications*.
- [7] Mateos Martín F. J., R. R. (2013). *Dpto. de Ciencias de la Computación e Inteligencia Artificial*. Retrieved 12 05, 2019, from <https://www.cs.us.es/>
- [8] Bhagwat, V. A. (2018). *Deep Learning for Chatbots*.
- [9] Cahn, J. (2017). *CHATBOT: Architecture, Design, & Development*.
- [10] Colby, K. M. (1981). *Modeling a paranoid mind. Behavioral and Brain Sciences, 4(04), 515*.
- [11] chatbotsmagazine. (n.d.). <https://chatbotsmagazine.com/>. Retrieved 12 24, 2018, from <https://chatbotsmagazine.com/7-types-of-bots-8e1846535698>
- [12] SCRUMstudy. (2013). *Una guía para el conocimiento de SCRUM*. Phoenix.
- [13] Google. (n.d.). *DialogFlow console*. Retrieved from <https://dialogflow.cloud.google.com/>
- [14] GITHUB. (n.d.). Retrieved from https://github.com/keras-team/keras/blob/master/examples/lstm_seq2seq.py
- [15] Google. (n.d.). *Colab*. Retrieved from <https://colab.research.google.com/drive/19h8lagWaaV66qx8N8LKAs624AB1m8GOy>