



**Uso del procesamiento del lenguaje natural (PLN) en la implementación de
agentes conversacionales (bots) para perfiles de Facebook**

Ruiz Yaguachi, Verónica Belén

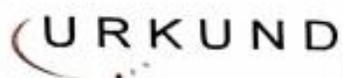
Departamento de Ciencias de la Computación

Carrera de Ingeniería de Sistemas e Informática

Trabajo de titulación, previo a la obtención del título de Ingeniera en Sistemas e Informática

Ing. Coral Coral, Henry Ramiro

2 de septiembre del 2020



Urkund Analysis Result

Analysed Document: Tesis Veronica Ruiz.docx (D79193073)
Submitted: 9/15/2020 6:31:00 PM
Submitted By: HRCORAL@ESPE.EDU.EC
Significance: 1 %

Sources included in the report:

[https://iso25000.com/index.php/normas-iso-25000/iso-25010/24-fiabilidadjalife,](https://iso25000.com/index.php/normas-iso-25000/iso-25010/24-fiabilidadjalife)

Instances where selected sources appear:

1

Firma:

A handwritten signature in blue ink, appearing to read "Coral Coral Henry Ramiro". The signature is written over a horizontal line.

Ing. Coral Coral Henry Ramiro

C.C.:171486483-0



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Certificación

Certifico que el trabajo de titulación, **“Uso del procesamiento del lenguaje natural (PLN) en la implementación de agentes conversacionales (bots) para perfiles de Facebook”** fue realizado por la señorita Ruiz Yaguachi, Verónica Belén el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 2 de septiembre del 2020

Firma:

Ing. Coral Coral Henry Ramiro

C.C.:171486483-0



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

RESPONSABILIDAD DE AUTORÍA

Yo, **Ruiz Yaguachi, Verónica Belén**, con cédula de ciudadanía n° 171848810-7, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Uso del procesamiento del lenguaje natural (PLN) en la implementación de agentes conversacionales (bots) para perfiles de Facebook”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 2 de septiembre del 2020

Firma:

Ruiz Yaguachi, Verónica Belén

C.C.: 171848810-7



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Ruiz Yaguachi, Verónica Belén**, con cédula de ciudadanía n° 171848810-7, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **"Uso del procesamiento del lenguaje natural (PLN) en la implementación de agentes conversacionales (bots) para perfiles de Facebook"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 2 de septiembre del 2020

Firma:

Ruiz Yaguachi, Verónica Belén

C.C.: 171848810-7

Dedicatoria

Quiero dedicar a mis padres por haberme forjado en la persona que soy actualmente, con reglas y libertades que a la final me ayudaron a ser responsable con mi vida, a aprender a tomar buenas o malas decisiones pero a salir adelante a pesar de las dificultades, a mis hermanos que los quiero mucho, ser un ejemplo para ellos y que pueden contar conmigo si eso les ayuda a llegar más lejos en sus metas y en general a toda mi familia que son lo más importante en mi vida.

A todas aquellas personas que dudaron y no creyeron en mí, me subestimaron y juzgaron por dejarse de llevar de las apariencias pero que con lucha y energías positivas pude llegar a la final.

Verónica Belén Ruiz Yaguachi

Agradecimiento

Agradezco primero a Dios por guiarme en este camino de lucha, donde me permitió vivir la mejor etapa que es la universidad, donde tuve la fortuna de conocer compañeros y los mejores ingenieros que formaron parte en mi formación académica.

A todas las personas que han estado involucradas en este proceso y han aportado para su culminación. A mis padres que han estado siempre conmigo para brindarme su apoyo incondicional, paciencia y guía permanente. A mi novio Diego Pacheco por estar siempre presente apoyándome y brindándome todo su amor y cariño para lograr culminar con esta meta.

A la empresa QUEST por su colaboración en el desarrollo de este proyecto, permitiéndome poner siempre como prioridad principal la universidad.

Verónica Belén Ruiz Yaguachi

Índice de contenidos

<i>Urkund</i>	2
<i>Certificación</i>	3
<i>Responsabilidad de autoría</i>	4
<i>Autorización de publicación</i>	5
<i>Dedicatoria</i>	6
<i>Agradecimiento</i>	7
<i>Índice de contenidos</i>	8
<i>Índice de tablas</i>	12
<i>Índice de figuras</i>	13
<i>Resumen</i>	16
<i>Abstract</i>	17
<i>Capítulo I Introducción</i>	18
<i>Antecedentes</i>	18
<i>Planteamiento del problema</i>	19
<i>Justificación</i>	21
<i>Objetivos</i>	22
<i>Objetivo general</i>	22
<i>Objetivos específicos</i>	22
<i>Alcance</i>	22

<i>Capitulo II Agentes de auto-respuesta</i>	24
Inteligencia Artificial	24
Aprendizaje automático	24
Aprendizaje supervisado	25
Aprendizaje no supervisado	25
Sistemas Expertos	25
Basados en reglas.....	26
Basados en casos.....	27
Basados en probabilidades.....	28
Procesamiento del Lenguaje Natural.....	29
Historia	30
Técnicas y modelos del procesamiento del lenguaje natural.....	31
Implementación de agentes de auto-respuesta.....	39
Tipos de agentes de auto-respuesta	41
Características de un agente de auto-respuesta	42
Definición de Dominio.....	43
Construcción de personalidades.....	43
<i>Capitulo III Implementación de agente de auto-respuesta para empresas comerciales</i>	45
Levantamiento de información	45
Definición de Dominio.....	46
Herramientas para el desarrollo.....	48
Visual Studio Code.....	48

	10
Nodejs.....	49
Mongodb	50
Ngrok	51
Desarrollo del agente conversacional	52
Capa de datos	52
Capa de aprendizaje supervisado	54
GLN del agente conversacional.....	60
<i>Capitulo IV Integración con Facebook.....</i>	<i>63</i>
Facebook	63
Perfiles públicos de empresas	64
Facebook for Developers.....	64
Backend de mensajería del agente conversacional	65
Variables de entorno.....	65
Sendapi de Messenger	66
Webhook de Messenger.....	70
Integración del agente conversacional en Facebook.....	72
Crear una cuenta en Facebook for Developers.....	72
Crear aplicación en Facebook for Developers	75
Crear página de Facebook	76
Agregar productos a la aplicación	78
Agregar páginas a la aplicación	78
Suscripción del Webhook	80

<i>Capítulo V Evaluación de resultados</i>	87
Metodología	87
Definición de parámetros de evaluación.....	88
Desempeño teórico del agente conversacional.....	88
Desempeño del agente conversacional con usuarios reales	91
Satisfacción de los usuarios	92
Usabilidad del sistema.....	93
Confiabilidad del sistema.....	94
Ejecución de pruebas controladas y recolección de información	94
Evaluación de resultados.....	97
Análisis del desempeño del sistema	97
Análisis de la métrica de satisfacción.....	98
Análisis de la métrica de usabilidad	102
Confiabilidad del agente conversacional.....	104
Resumen de la evaluación de resultados	104
<i>Capítulo VI Conclusiones y recomendaciones</i>	105
Conclusiones.....	105
Recomendaciones.....	105
Trabajo Futuro	106
<i>Bibliografía</i>	107

Índice de tablas

Tabla 1 <i>Ejemplos de prefijos</i>	31
Tabla 2 <i>Ejemplos de sufijos</i>	31
Tabla 3 <i>Etiquetas del análisis sintáctico</i>	32
Tabla 4 <i>Morfemas sufijos ejemplos</i>	36
Tabla 5 <i>Morfemas prefijos ejemplos</i>	36
Tabla 6 <i>Preguntas frecuentes de una librería</i>	45
Tabla 7 <i>Clasificación de PFs por Intención</i>	46
Tabla 8 <i>Clasificación de Intención con sus respuestas</i>	47
Tabla 9 <i>Peticiones no comunes</i>	47
Tabla 10 <i>Propiedades del objeto JSON de mensajería</i>	67
Tabla 11 <i>Parámetros del objeto Recipient</i>	68
Tabla 12 <i>Componentes del objeto Message</i>	68
Tabla 13 <i>Propiedades de mensaje de archivo adjunto</i>	69
Tabla 14 <i>Matriz de confusión de dimensiones 3x3</i>	89
Tabla 15 <i>Modelo de encuesta de satisfacción de usuario</i>	92
Tabla 16 <i>Modelo de encuesta de satisfacción de usuario</i>	93
Tabla 17 <i>Matriz de confusión del agente conversacional</i>	95
Tabla 18 <i>Datos de preguntas de usuarios</i>	96
Tabla 19 <i>Resultados de la experiencia de usuarios</i>	98
Tabla 20 <i>Resumen de parámetros de confiabilidad</i>	104

Índice de figuras

Figura 1 <i>Estructura de un Sistema Experto</i>	26
Figura 2 <i>Ejemplo de detección de oraciones</i>	34
Figura 3 <i>Ejemplo de Segmentación de palabras</i>	34
Figura 4 <i>Ejemplo de Radicación</i>	35
Figura 5 <i>Ejemplo de Etiquetado Gramatical</i>	36
Figura 6 <i>Ejemplo Bolsa de Palabras descrito las palabras vacías y lematización</i>	37
Figura 7 <i>Ejemplo de REN</i>	38
Figura 8 <i>Ejemplo de Expresión Regular</i>	38
Figura 9 <i>Ejemplo Dependencia Gramatical</i>	39
Figura 10 <i>Componentes de un enunciado</i>	40
Figura 11 <i>Logo de Visual Studio Code</i>	48
Figura 12 <i>Logo de Nodejs</i>	49
Figura 13 <i>Función Node.js</i>	50
Figura 14 <i>Logo de MongoDB</i>	50
Figura 15 <i>Logo Ngrok</i>	51
Figura 16 <i>Diagrama de Arquitectura</i>	52
Figura 17 <i>Estructura del corpus del agente conversacional.</i>	53
Figura 18 <i>Estructura de un documento de la colección libros.</i>	54
Figura 19 <i>Diagrama de flujo del bloque de procesamiento del lenguaje natural</i>	55
Figura 20 <i>Diagrama de flujo del generador del lenguaje natural</i>	61
Figura 21 <i>Diagrama de flujo del clasificador de Naive Bayes</i>	62
Figura 22 <i>Variables de entorno de la aplicación</i>	66
Figura 23 <i>Objeto JSON enviado desde la API de Messenger</i>	67

Figura 24	<i>Código de envío de mensaje hacia la API Messenger</i>	69
Figura 25	<i>Creación del servidor express HTTP</i>	70
Figura 26	<i>Código fuente para la verificación de webhook</i>	71
Figura 27	<i>Código fuente de la asignación del extremo del webhook</i>	72
Figura 28	<i>Plataforma Facebook for Developers</i>	73
Figura 29	<i>Formulario inicio de Sesión y de registro de usuario</i>	73
Figura 30	<i>Verificación de Cuenta e Información sobre el usuario</i>	74
Figura 31	<i>Ventana de Bienvenida a FBFD</i>	74
Figura 32	<i>Crear app en FBFD</i>	75
Figura 33	<i>Crear identificador de App</i>	75
Figura 34	<i>Identificador de la app</i>	76
Figura 35	<i>Ruta para crear nueva página de FB</i>	76
Figura 36	<i>Creación de una nueva página de FB</i>	77
Figura 37	<i>Mensajería avanzada sin apps sincronizadas</i>	77
Figura 38	<i>Agregar productos a la app</i>	78
Figura 39	<i>Agregar páginas a la aplicación</i>	78
Figura 40	<i>Agregar página de FB a la aplicación</i>	79
Figura 41	<i>Autorizar acceso a Messenger</i>	79
Figura 42	<i>Identificador y Clave secreta de la app</i>	80
Figura 43	<i>Identificador y token de la página</i>	81
Figura 44	<i>Page Token Access generado</i>	81
Figura 45	<i>Ejecución de Ngrok</i>	82
Figura 46	<i>Archivo de variables de entorno</i>	83
Figura 47	<i>Agregar URL de devolución de llamada</i>	83

Figura 48 <i>URL de devolución de llamada.</i>	84
Figura 49 <i>Webhook de FBFD configurado.</i>	84
Figura 50 <i>Suscripciones de webhooks.</i>	85
Figura 51 <i>Campos seleccionados de Webhooks.</i>	85
Figura 52 <i>Verificación de Webhook</i>	86
Figura 53 <i>Apps conectadas a la página.</i>	86
Figura 54 <i>Escala de medidas de la satisfacción. Adjetivos y rangos de aceptabilidad</i>	94
Figura 55 <i>Resultados pregunta 1</i>	99
Figura 56 <i>Resultado pregunta 2.</i>	99
Figura 57 <i>Resultados pregunta 3</i>	100
Figura 58 <i>Resultados pregunta 4</i>	100
Figura 59 <i>Resultados pregunta 5</i>	101
Figura 60 <i>Resultados pregunta 7</i>	101
Figura 61 <i>Promedio de calificaciones de preguntas de usabilidad</i>	102
Figura 62 <i>Promedio de calificaciones de usabilidad con ítems negativos revertidos.</i>	103

Resumen

El procesamiento de lenguaje natural es una rama de la inteligencia artificial que está enfocada a la investigación de sistemas computacionales que permiten la comunicación entre personas y máquinas con el uso del lenguaje natural. Por otra parte, los agentes de auto respuesta “chatbot” son software que imitan una conversación con un usuario a través de canales de mensajería como: Facebook Messenger, Whatsapp, correos electrónicos, entre otros. Para tales fines, se hace uso de técnicas computacionales tales como: procesamiento de lenguaje natural, aprendizaje automático. El presente documento describe el diseño de una herramienta de comunicación automática integrada en la plataforma Messenger de Facebook, para la página de una tienda de libros. El objetivo de este agente conversacional es responder a preguntas frecuentes de potenciales clientes. Su funcionamiento se fundamenta en un sistema experto basado en probabilidades, mismo que integra el algoritmo de clasificación Naive Bayes, el cual, es el más utilizado de los algoritmos de redes bayesianas. El entrenamiento de este sistema utiliza el método de aprendizaje supervisado, mismo que toma como base de conocimiento las etiquetas de posibles variantes del lenguaje que puede tener una misma pregunta frecuente. La ventaja de este tipo de agente conversacional es la alta confiabilidad y la satisfacción del cliente, esto a un bajo costo de implementación, además de la facilidad de escalamiento en su base de conocimiento.

– Palabras clave:

- **PROCESAMIENTO DE LENGUAJE NATURAL**
- **CLASIFICADOR DE NAIVES BAYES**
- **APRENDIZAJE AUTOMÁTICO**
- **AGENTE CONVERSACIONAL**
- **MESSENGER DE FACEBOOK**

Abstract

Natural language processing is a branch of Artificial Intelligence, that is focused on the investigation of computer systems that allow communication between people and machines using natural language. Otherwise, auto-response agent "chatbot", are software system, that imitate a conversation with a human user through messaging channels such as: Facebook Messenger, WhatsApp, e-mails, among others. For such purposes, computational techniques such as: natural language processing, machine learning, are used. This document describes the design of an automatic communication tool integrated into the Facebook Messenger platform for a book store page. The main purpose of this auto-response agent is to answer frequently asked questions from potential clients of the book store. This chatbot is an expert system based on probabilities, which integrates the Naive Bayes classifier algorithm, which is the most widely used of the Bayesian network algorithms. The training of this system uses the supervised machine learning method, which has a knowledge base with labels that include possible language variants of the frequently asked questions. The advantage of this type of chatbot is high reliability and customer satisfaction, at a low cost of implementation and maintenance, also, the ease of scaling the knowledge base.

– Key words:

- **NATURAL LANGUAGE PROCESSING**
- **NAIVE BAYES CLASIFIER**
- **MACHINE LEARNING**
- **CHATBOT**
- **FACEBOOK MESSENGER**

Capítulo I

Introducción

Antecedentes

El Lenguaje Natural es el que se utiliza en la comunicación humana, y puede ser escrito, hablado o signado. El procesamiento de lenguaje natural es una disciplina de la Inteligencia Artificial que se ocupa de la formulación e investigación de mecanismos computacionales para la comunicación entre personas y máquinas mediante uso del Lenguaje Natural (F. J. Martín Mateos, 2012 - 2013).

El Procesamiento de Lenguaje Natural tiene su origen desde los años 50, en sus inicios las primeras aplicaciones desarrolladas para ejecutarse en computadores, con limitadas capacidades de memoria y procesamiento, estaban orientadas a la traducción del ruso al inglés (aplicación con fines militares). En las décadas de los años 70s y 80s, se desarrolló el primer sistema de comprensión completa de un dominio limitado y se avanzó más en la traducción automática entre varios idiomas. En la década de los 90s al utilizar computadores de mejores características se desarrollaron aplicaciones de estado finito y se inició la investigación del uso de métodos probabilísticos para procesar el lenguaje natural de forma automática; finalmente desde los años 2000 y con el uso de algunas técnicas de inteligencia artificial modernas se ha llegado a analizar textos poniendo un especial énfasis en el análisis semántico y el estilo de escritura; para analizar la autoría de textos, el significado de discursos y principalmente en la implementación de diálogo hombre-máquina (Grampln, 2012) .

Los “Agentes de Auto-Respuesta” también conocidos como “Chat Bots” constituyen una de las mejores herramientas de comunicación disponibles para para las pequeñas y medianas empresas que no cuentan con los recursos económicos suficientes para contratar personal dedicado a solventar las inquietudes y necesidades de los clientes.

La interacción de los usuarios con agentes de auto-respuesta permite la detonación de eventos en sistemas externos como ejecutar servicios expuestos de consulta e incluso en algunos casos se pueden ejecutar transacciones comerciales o bancarias. El uso más común de los agentes de auto-respuestas es responder a los mensajes entrantes de forma automática con un mensaje definido por el usuario (Altn, 2019); sin embargo, esta funcionalidad puede ser contraproducente al sentir los usuarios que no están comunicándose con un ser humano.

En la actualidad la presencia de las empresas en las redes sociales como Facebook o Instagram es un estándar de facto en sus políticas de marketing; atrás ha quedado los tiempos en los que toda empresa buscaba implementar un sitio web para darse a conocer. Las redes sociales tienen millones de usuarios activos y segmentados a los cuales se puede acceder con la contratación de planes de promoción que están al alcance de todos los presupuestos (Estevez, 2020).

Planteamiento del problema

En la actualidad varias empresas que son clasificadas como Pymes utilizan las redes sociales como Facebook, Instagram y Whatsapp para promocionarse, estas redes sociales cuentan con un gran número de usuarios activos (seguidores) y además proveen herramientas de marketing directo a bajo costo, lo que las convierte en una gran opción para ofertar sus productos o servicios a un público segmentado.

Cuando se utiliza Facebook de forma adecuada en la estrategia de Mercadeo en Redes Sociales, se logra generar un importante tráfico de audiencia de valor hacia los sitios web de pequeñas y medianas empresas. De acuerdo con el estudio de (We Are Social Ltd's, 2018) “el 95.1% de los usuarios que usan Facebook lo hacen desde dispositivos móviles (tablets y celulares)”; eso hace importante que todos los enlaces desde esta red social al sitio web o tienda electrónica deben tener buena experiencia de navegación en este tipo de dispositivos.

Hasta la década pasada, las llamadas telefónicas eran la forma dominante de comunicación de los clientes con las empresas; sin embargo, todo ha cambiado con el auge del Internet, los dispositivos móviles y las redes sociales. Las cifras actuales demuestran que la telefonía convencional ha sido desplazada por los mensajes de texto mandados y recibidos por servicios de mensajería instantánea, tales como Facebook Messenger y WhatsApp Messenger. Tan solo en Centroamérica y el Caribe se ha reportado un aumento exponencial en el uso de este tipo de herramientas en los últimos 3 años (Estevez, 2020).

Este cambio en la forma de interacción de las empresas con sus clientes, ha sido beneficioso; sin embargo, conforme crece el número de clientes las pequeñas y medianas empresas han visto que se ha generado un nuevo problema y es que no cuentan con personal suficiente para entablar comunicación directa con todos los potenciales clientes; lo que muchas veces se vuelve una situación contraproducente ya que el usuario puede llegar a sentir que no es tomado en cuenta, o incluso llegar a generar una percepción de falta de seriedad por parte de la misma (Viloria, 2011).

El problema expuesto, afecta en mayor grado a empresas grandes como operadoras de telefonía celular, suscriptoras de televisión pagada, o internet y entidades bancarias y financieras; estas empresas muchas veces invierten en la contratación de empresas de atención al cliente y call center; para dar una buena atención a sus clientes; sin embargo, con la cantidad de tráfico que aumenta a diario los clientes tienen una tasa de espera de atención cada vez más alta; por lo que se debe buscar soluciones más tecnológicas antes que de recursos humanos por el costo que representa el mantenimiento de estos (Ramírez, 2013).

Para abordar esta problemática y su solución se propone el desarrollo de Agente de Auto-respuesta o Conversacionales, “un agente conversacional es una entidad viviente artificial diseñada para tener conversaciones con seres humanos reales. Estas conversaciones se pueden

expresar vía texto, de forma oral o incluso una conversación no verbal.” Los agentes conversacionales también son conocidos como “chatbots” (Morales-Rodríguez & Domínguez-Martínez, 2011).

De acuerdo a (Morales Rodríguez, 2011) “Los agentes inteligentes tienen las características: Autonomía, Proactividad, Sociabilidad; esto quiere decir que un agente debe tener: comunicación, cooperación y negociación. Sin embargo, cuando los agentes presentan personalidad y emociones se considera un agente conversacional”.

Con la inclusión de un agente conversacional ligado al perfil público de una empresa en una red social como Facebook, se podría atender varias inquietudes (previamente definidas y relacionadas con la empresa) de los usuarios de una forma fácil y rápida; sin recurrir a la interacción de los clientes con seres humanos (Estevez, 2020).

Justificación

En la actualidad muchas Pymes no cuentan con los recursos económicos para contratar agentes que interactúen con clientes o potenciales clientes en sus perfiles de redes sociales; incluso grandes empresas comercializadoras como empresas de telefonía celular o suscripción por cable cuentan con recursos limitados para atender a los usuarios que realizan consultas por medio de redes sociales.

Con el desarrollo de este tema de investigación se espera suplir esta falencia de las Pymes, ya que la interacción con un agente de auto-respuesta el cual puede entender y responder un número limitado de preguntas en un contexto claramente definido, permitirá mejorar la percepción de los clientes o potenciales clientes, manteniendo así el interés de los mismos y de ser el caso redirigiéndolos a una comunicación directa con un agente humano.

El motor de inteligencia artificial que se pretende implementar será fácilmente adaptable a diferentes empresas dada su generalidad; con un costo de implementación y mantenimiento

muy bajo, lo que permitirá dar una ventaja competitiva a las pymes que decidan utilizarlo manteniendo el interés de los clientes o potenciales clientes que llegan al perfil de Facebook de las empresas.

Objetivos

Objetivo general

Implementar un Agente conversacional (BOT) utilizando el Procesamiento de Lenguaje Natural (PLN) que será asociado a un perfil de empresa que se dedica a la venta de libros, en la red social Facebook que entienda y atienda inquietudes de los potenciales clientes.

Objetivos específicos

- i. Levantamiento de información de las necesidades que están ligadas a las preguntas frecuentes de la empresa.
- ii. Implementar el algoritmo de Procesamiento de Lenguaje Natural basado en Corpus.
- iii. Integrar el algoritmo de Procesamiento de Lenguaje Natural a un Agente Conversacional de Facebook
- iv. Evaluar la precisión y confiabilidad del algoritmo de Procesamiento de Lenguaje Natural desarrollado en base a la satisfacción de los clientes de la empresa.

Alcance

Esta investigación comprende en primer lugar la definición e implementación de un algoritmo de Procesamiento de Lenguaje Natural (PLN) basado como un Servicio Web que se ejecutará en un servidor de aplicaciones.

Este proyecto está orientado a ser de utilidad para Pymes; se realizará un proceso de recolección de información de las necesidades de una empresa comercial de la ciudad de Quito, dedicada a la venta de libros al por menor, de la cual se reserva su nombre por seguridad; las necesidades de esta empresa están ligadas a las preguntas frecuentes que los clientes o

potenciales clientes hacen en el perfil de Facebook de la empresa y que muchas veces no son respondidas de forma inmediata. Con la información recolectada se definirá el “contexto” del BOT; el cual sirve para definir el corpus de entrenamiento y posible aprendizaje del mismo.

La idea de exponer el algoritmo de PLN como un servicio web; es que el mismo pueda ser consumido por la API de la red social Facebook; la cual permite integrar de forma estandarizada la ventana de “Chat” de un perfil relacionado a un página de seguidores “fan page” de la red social con un motor propio de inteligencia artificial. En este caso el “motor de inteligencia artificial” está compuesto por el algoritmo de procesamiento de lenguaje natural y un clasificador probabilístico para responder de acuerdo a las iteraciones de los usuarios de Facebook.

Finalmente, una vez que se tenga la integración entre el perfil de Facebook de las empresas y el agente conversacional desarrollado; se realizará una evaluación de la fiabilidad y confiabilidad del mismo, así como de la satisfacción de los clientes o potenciales clientes al interactuar con este.

Capítulo II

Agentes de auto-respuesta

Inteligencia Artificial

La inteligencia artificial (IA), tiene como objetivo estudiar y analizar la comprensión y percepción del ser humano a la hora de resolver problemas y tomar decisiones, y en base a esto desarrollar sistemas informáticos que simulen las actividades intelectuales del ser humano (Suarez, 2006).

El termino IA fue acuñado por el informático estadounidense John McCarthy en el año de 1956, durante la Conferencia de Dartmouth, por primera vez recibe un significado (Serna , Acevedo , & Serna, 2017). Actualmente la IA es utilizada en varios campos tales como: automatización de procesos robóticos, identificación de patrones en datos (España, 2019), análisis de solvencia empresarial (Suarez, 2006), entre otros.

Dentro de las disciplinas de la IA, se puede mencionar las siguientes: Aprendizaje automático, Procesamiento de Lenguaje Natural (PLN), Sistemas Expertos, Lógica Difusa o “Fuzzy Logic”, Redes Neuronales Artificiales (Escobar Macías, 2019).

Aprendizaje automático

El aprendizaje automático o mejor conocida como “Machine Learning” es una disciplina de la IA, que consiste en la construcción de algoritmos los cuales tienen la capacidad de aprender y hacer predicciones de un conjunto de datos. Su funcionamiento está basado en la construcción de un modelo, que recibe ciertas entradas de ejemplo, y las asocia con una salida, esto con el fin de realizar predicciones o elecciones basadas en datos, en lugar de seguir instrucciones estáticas de un programa (Zarabia Zuñiga, Abril, 2018). En el aprendizaje automático se puede encontrar dos modelos de aprendizaje: el modelo de aprendizaje supervisado, y el de aprendizaje no supervisado, estos modelos son descritos a continuación.

Aprendizaje supervisado.

El Aprendizaje Supervisado “Supervised Machine Learning”, trabaja en un conjunto predefinido de ejemplos de entrenamiento que son etiquetados por categorías, por lo que su capacidad de llegar a una conclusión al recibir datos de entrada es más sencilla. Algunos modelos fundamentados en este tipo de aprendizaje son: redes neuronales artificiales, arboles de decisión, máquinas de vectores, redes bayesianas entre otros (A. Simon, 2016).

Uno de los sistemas más utilizado con aprendizaje supervisado es Servicio Inteligente de Comprensión del Lenguaje “Language Understanding Intelligent Service” (LUIS) de Microsoft, el cual crea una comprensión de lenguaje natural que puede ser utilizado en aplicaciones, bots y dispositivos IoT (Internet de las cosas) (Zarabia Zuñiga, Abril, 2018). La configuración de este servicio se basa en la definición de intenciones dentro de un dominio específico, las cuales contienen expresiones íntegramente relacionadas a la intención (Microsoft, Microsoft , 2018).

Aprendizaje no supervisado.

El Aprendizaje No Supervisado “Unsupervised Machine Learning”, trabaja con gran cantidad de datos, los cuales son analizados para encontrar patrones y relaciones con el fin de poder agruparlos. La principal diferencia con el modelo de aprendizaje supervisado es que no existe una base de conocimiento. Algunos modelos fundamentados en este tipo de aprendizaje son: K-means, clustering, cadena de markov y análisis de componentes de independientes (Fernández Torres, Sandoya Villafuerte, & Crespo Torres, 2019).

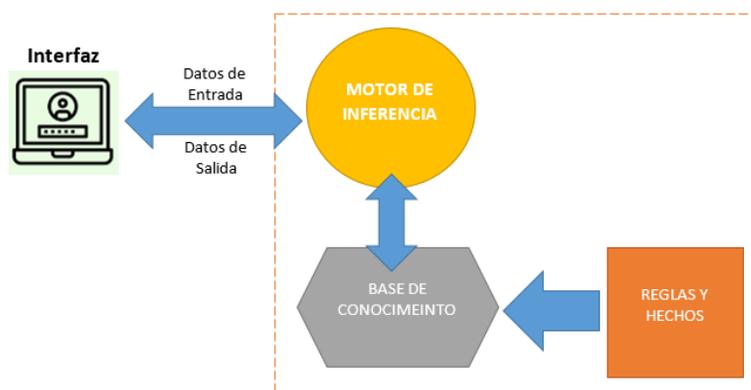
Sistemas Expertos

Un sistema experto es un sistema informático, que simula a los expertos humanos dentro de un área de especialización. Estos sistemas pueden ser capaces de aprender, razonar, procesar y memorizar información, e incluso, lograr comunicarse con las personas u otros sistemas expertos para así tomar decisiones más apropiadas (Castillo, Gutierrez, & Hadi, 1997).

Los primeros sistemas expertos se conocían como Sistemas Basados en Conocimientos (SBC). Su estructura tiene dos partes principales: Base de Conocimiento la cual posee una serie de datos en forma de reglas y hechos que un especialista aporta para la programación del sistema; y el Motor de Inferencia el cual se encarga del razonamiento a partir de la información registrada la base de conocimiento. Estos componentes interactúan entre sí para obtener una solución a un problema de un dominio específico. Los SBC son principalmente usados para: contestar preguntas, resolver problemas rápidamente y realizar operaciones monótonas (Zarabia Zuñiga, Abril, 2018).

Figura 1

Estructura de un Sistema Experto.



Nota. La arquitectura básica de un sistema experto.

Los sistemas expertos pueden ser clasificados en tres tipos principales, tales como: basados en reglas, basados en hechos y basados en probabilidades. El detalle de estos tipos es abordado a continuación en los apartados posteriores (Zarabia Zuñiga, Abril, 2018).

Basados en reglas

Estos sistemas sacan sus conclusiones basándose en un conjunto de reglas determinadas y una base de hechos también conocida como memoria de trabajo, que utiliza un mecanismo de razonamiento lógico (Zarabia Zuñiga, Abril, 2018).

Estas reglas determinadas o afirmación lógica normalmente identificada como “Si premisa entonces una conclusión” que pueden contener una lógica expresión simple o expresión lógica compuesta, las cuales están definidas en la base de conocimiento que también contienen los datos dinámicos, es decir las evidencias o los hechos conocidos por algún escenario particular que puede cambiar. Pero que permanece de manera fija y estática una vez guardada en la base de conocimiento, a menos que se incorpore al sistema experto elementos de aprendizaje. El motor de inferencia saca las conclusiones en base a estas reglas aplicando la lógica de conocimiento, es decir si la afirmación de una regla es cierta entonces la conclusión de la regla en base a ese conocimiento es verdadero. Los datos pueden incrementar incluyendo nuevas conclusiones, las cuales, del mismo modo pueden ser simples de una sola regla o compuestas más de una regla. Para finalizar, este sistema experto cuenta con un control de coherencia para situaciones complejas donde se ingrese información endeble ya sea en las reglas o hechos no factibles, para un mejor control durante el desarrollo de la base de conocimiento evitando conclusiones absurdas ayudando al usuario (Castillo, Gutierrez, & Hadi, 1997).

Basados en casos

Este contiene las experiencias, situaciones previas o similares al actual, con las cuales el sistema realiza sus inferencias para comprender, tomar decisiones y resolver problemas (Rodríguez, González, & Hernández, 2011).

Formado por 3 partes: la base de casos donde el sistema inicia con los problemas solucionados almacenados en la misma, el módulo de recuperación comprende y conforma el dominio de la aplicación del nuevo problema a resolver y el módulo de adaptación que realiza la búsqueda en la base de casos para encontrar la solución al nuevo problema en base a problemas o casos similares resueltos, ya sea transformando para ajustarlo a los requerimientos del nuevo

problema o la composición aparte de varios casos similares (Moya Rodríguez, Becerra Ferreiro, & Chagoyén Méndez, 2012).

Basados en probabilidades

Estos sistemas tienen una medida de incertidumbre para poder tratar situaciones en que la incertidumbre es lo común y no la excepción. La certeza se basa en la cantidad de información que maneja y que tan probable es la respuesta (Huarote Zegarra, 2013). Dentro de este tipo de sistema experto se tiene a las redes bayesianas, las cuales son comúnmente utilizadas en problemas de clasificación y aprendizaje. La base de este clasificador es la aplicación del teorema de Bayes, con el cual a partir de las probabilidades priori, se puede calcular las probabilidades a posteriori (Fuente & Díaz, 2006).

Un tipo de clasificador de redes bayesianas es el bayesiano ingenuo “Naive Bayes”, este es el método probabilista más utilizado por su simplicidad. Para poder encontrar las probabilidades posteriori se hace uso de la fórmula planteada en la ec. 2.1 (Russel & Norvig, 2004).

$$P(A|B_1, \dots, B_n) = P(A) \times \prod_{i=0}^n P(B_i|A) \quad (2.1)$$

El algoritmo de Naive Bayes asume una independencia entre variables de la clase dada, con lo que el aprendizaje es más simple, es decir, cada variable contribuye de manera independiente a la probabilidad de clasificar dicha variable en una de las clases del problema, consiguiendo así la forma de calcular la probabilidad posteriori (Rish, 2001).

Donde $P(A_i)$ son las probabilidades a priori $P(B|A_i)$ es la probabilidad de B en la hipótesis A_i y $P(A|B_i)$ probabilidades posteriori.

Una de las ventajas de este clasificador es que funciona correctamente con un aprendizaje supervisado, y no necesita de una gran cantidad de datos para ser entrenado, y de esta forma

lograr clasificar rápidamente cada modelo. Este clasificador es comúnmente usado en el área del procesamiento de lenguaje natural (Merchán Flores, Galar Idoate, & Sesma, 2020).

Procesamiento del Lenguaje Natural

El procesamiento de lenguaje natural (PLN) es una disciplina de la IA que se centra en la investigación y desarrollo de sistemas computacionales para la comunicación entre personas y máquinas haciendo uso Lenguaje Natural (Mantilla Sierra & Mauro, 2014). El trabajo de (Mg. Augusto Cortez, 2009) presenta una ventaja y desventaja del PLN, como ventaja se tiene que el locutor no debe aprender el medio de comunicación con la máquina, como lo hacen los lenguajes de comando o las interfaces gráficas. Sin embargo, su desventaja es la limitada capacidad de un computador en cuanto a la comprensión del lenguaje. Por ejemplo, el usuario no puede esperar que el computador responda correctamente si este recibió una entrada que la persona sobrentiende, o también al utilizar palabras que están fuera del léxico del computador, es decir lo que un humano considera normal dentro de su lenguaje natural, puede convertirse en un reto para la máquina a la hora del tratamiento computacional (Mg. Augusto Cortez, 2009).

El PLN cuenta con subconjuntos que unidos ayudan a la mejora del desempeño de una máquina, tales como Comprensión del Lenguaje Natural (CLN), Generación del Lenguaje Natural (GLN), entre otros (Fernandes, 2018).

La CLN o “Natural Language Understanding” (NLU), permite comprender la intención dentro de una frase, al manejar las entradas no estructuradas de esa frase y convertirlas en una forma estructurada para que la máquina pueda entender y actuar. Además, puede utilizar el aprendizaje no supervisado para mejorar continuamente el entendimiento (Fernandes, 2018).

La GLN O “Natural Language Generation” (NLG), es un proceso en el cual se genera la respuesta del agente conversacional, es decir, cuando las computadoras responden con el lenguaje natural, convirtiendo los datos estructurados en forma texto (Fernandes, 2018) .

Historia

La historia del PLN empieza desde 1950, aunque se ha encontrado trabajos anteriores. Alan Turing publicó “Computing machinery and intelligence” en el cual proponía el test de Turing como criterio de inteligencia. El experimento de Georgetown en 1954 involucró traducción automática de más de sesenta oraciones del Ruso al Inglés. Los autores sostuvieron que en tres o cinco años la traducción automática sería un problema resuelto. El progreso real en traducción automática fue más lento y después del reporte ALPAC en 1996, se constató que la investigación había tenido un bajo desempeño. Más tarde, una investigación a menor escala en traducción automática se realizó a finales de 1980, cuando se desarrollaron los primeros sistemas de traducción automática estadística. Esto se debió tanto al aumento constante del poder de cómputo resultante de la Ley de Moore y la disminución gradual del predominio de las teorías lingüísticas de Noam Chomsky como la Gramática Transformacional, cuyos fundamentos teóricos desalentaron el tipo de lingüística de corpus, que se basa el enfoque de aprendizaje de máquinas para el procesamiento del lenguaje (Benjamín, 2015).

Algunos de los primeros algoritmos de aprendizaje automático utilizados, tales como árboles de decisión, sistemas producidos de sentencias IF-THEN similares a las reglas escritas a mano. Se puede consultar un resumen de la historia de 50 años de procesamiento automático de publicaciones después del proyecto NLP4NLP en forma de una publicación doble en *Frontiers in Research Metrics and Analytics* (Libre, 2019) .

En los últimos años, las aportaciones a este componente de la IA han mejorado, permitiendo el procesamiento de gran cantidad de información en formato texto con un grado de eficacia aceptable. Muestra de ello es la aplicación de estas técnicas como parte fundamental en los motores de búsqueda web, en las herramientas de traducción automática, o en la generación automática de resúmenes (Zarabia Zuñiga, Abril, 2018).

Técnicas y modelos del procesamiento del lenguaje natural

Niveles de análisis del lenguaje.

La estructura del estudio del lenguaje natural generalmente contempla cuatro niveles de análisis tales como: morfológico, semántico, sintáctico y pragmático. Sin embargo, puede incluir otros niveles como información fonológica, análisis de discurso y conocimiento del mundo (Mg. Augusto Cortez, 2009). A continuación, se detalla cada uno de los principales niveles.

El análisis morfológico consiste en detectar como las palabras se construyen a partir de unas unidades de significado más pequeñas llamadas morfemas, es decir, el reconocimiento de sufijos y prefijos (Mg. Augusto Cortez, 2009). En la Tabla 1 y 2 se detallan prefijos y sufijos del idioma castellano.

Tabla 1

Ejemplos de prefijos

Prefijos	Significado	Ejemplo
A-	No	Amoral
Anti-	En contra de	Antinuclear
Des-	Negación	Deshacer
Entre-	Situación intermedia	Entresuelo
Ex	Que ya no es	Ex ministro
Extra-	Que pertenece a	Extraoficial
	Superior a	Extraordinaria
In-, im-, i-	No	Intocable
Post-	Después de	Postventa
Re-	Volver a	Rehacer
Sub-	Por debajo de	Subacuático

Nota. Prefijos del idioma castellano.

Tabla 2

Ejemplos de sufijos

Sufijos	Ejemplo
-able	Tocable
-ible	Bebible
-anza	Esperanza
-ción	Circulación

Sufijos	Ejemplo
-dad	Crueldad
-dor	Comprador
-ero	Camarero
-ez	Dejadedez
-ía	Alegría

Nota. Sufijos del idioma castellano.

La función del análisis sintáctico es estudiar los componentes sintácticos y como las palabras pueden unirse para formar una oración, al asignar una etiqueta estructural para la función de cada palabra que compone a la oración y que sintagmas son partes de otros sintagmas (Mg. Augusto Cortez, 2009). Esto mediante el uso de una gramática y/o diccionarios. Este análisis tiene un doble objetivo: permite separar las unidades lingüísticas con sentidos simples o compuestos, y por otro lado, permite desambiguar las categorías gramaticales asignadas por el analizador morfológico y al mismo tiempo enriquecer y autogenerar los diccionarios de aplicación (Eva M Méndez Rodríguez, 1999).

Tabla 3

Etiquetas del análisis sintáctico

Nombre	Categorías	Componente
Sintagma nominal	SN	Grupos de palabras cuyo núcleo es un sustantivo o pronombre
Sintagma verbal	SV	Grupo de palabras cuyo núcleo es un verbo
Sintagma Preposicional	SP	Grupo de palabras cuyo núcleo es una preposición
Determinante	D	Determinantes
Núcleo Sintáctico	N	Nombre, adjetivo o pronombre
Verbo	V	Verbo
Preposición	P	Preposición

El objetivo del análisis semántico consiste en encontrar el significado de las palabras y de cómo estos influyen en el significado general de una oración (Mg. Augusto Cortez, 2009). Además, este estudio incluye el reconocimiento de sinónimos e hiperónimos (Eva M Méndez Rodríguez,

1999) . A continuación, se muestra un ejemplo que plantea la desambiguación de la palabra “vela” a partir de los dos siguientes significados:

1. Cilindro de cera o sebo, atravesado por una mecha que se prende para alumbrar.
2. Pieza de lona o lienzo fuerte que, atada al mástil, recibe el viento que impulsa la nave.

Esta palabra puede estar incluida en oraciones, en las que su contexto puede cambiar su significado tal como se muestra en las siguientes oraciones:

1. Cristóbal Colon uso barcos de **vela** en su viaje a América.
2. Nosotros pusimos una **vela** al santo de la iglesia de San Francisco.

El análisis pragmático es el nivel más complejo de automatizar, pues su objetivo es incorporar al análisis semántico una aportación explicativa que implica el entendimiento del contexto entre oraciones, lo que puede producir una variación en la evolución de un discurso. Es decir, el significado de una oración se ve afectado por el análisis semántico de las oraciones previas (Mg. Augusto Cortez, 2009).

Técnicas de análisis del lenguaje.

Dentro del PLN existen varias técnicas que ayudan a la extracción de información de un texto. En el procesamiento de texto escrito existen técnicas que están relacionadas a una o más niveles descritos en la sección 2.2.2.1, entre las cuales se tienen: detección de oraciones, segmentación por palabras, Normalización, Radicación, etiquetado gramatical, segmentación morfológica, eliminación de palabras vacías o “stop words”, reconocimiento de entidades nombradas, análisis semántico latente, expresiones regulares y creación de estructuras de datos gramaticales (Ramos & Velez, 2016). A continuación, una explicación de cada una de ellas:

- Detección de oraciones: esta técnica recorta una secuencia de caracteres, es decir, un texto completo en oraciones que se encuentren entre dos signos de puntuación junto a un espacio en blanco, esta técnica puede tener su grado de dificultad en ciertos casos cuando

no siguen un patrón de texto plano. En la figura 2 se puede observar un ejemplo de la localización de las oraciones dentro de un texto (Ramos & Velez, 2016).

Figura 2

Ejemplo de detección de oraciones

“Al incorporarse y mirar mejor, le pareció ver a una hermosa mujer resplandeciente. Una inmensa luz lo cubría todo: las flores, el balcón y la bella y misteriosa dama.”

```
<Parrafo inicio="0" fin="166">
<Oracion inicio="0" fin="82" />
<Oracion inicio="83" fin="166" />
```

Nota. En el párrafo se detecta dos oraciones la primera empieza en el carácter 0 con la letra “A” y termina en el carácter 82 con él “.”. Segunda oración empieza en el carácter 83 con la letra “U” y termina con el punto final en el carácter 166.

- Segmentación de palabras o mejor conocida como “tokenización”, la oración es dividida en tokens, representando sus componentes: palabras, signos de puntuación, números, etc., en tokens individuales, basándose en un separador de espacios en blanco. Esta es una de las técnicas que más usaremos a lo largo del desarrollo del agente conversacional. Se realiza una segmentación del corpus de texto en unidades menores, asignándoles una serie de identificadores que serán utilizados como puntos de referencia en los diferentes análisis posteriores (Eva M Méndez Rodríguez, 1999).

Figura 3

Ejemplo de Segmentación de palabras

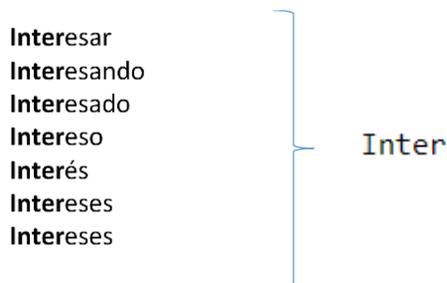
“Al incorporarse y mirar mejor, le pareció ver a una hermosa mujer resplandeciente. Una inmensa luz lo cubría todo: las flores, el balcón y la bella y misteriosa dama.”

```
...
<Palabra inicio="0" fin="1">
<Palabra inicio="3" fin="14" />
<Palabra inicio="16" fin="16" />
<Palabra inicio="18" fin="22" />
<Palabra inicio="24" fin="28" />
...
```

- Normalización o “Normalization”: el texto se divide en fragmentos de texto; fragmentos de texto se dividen en oraciones, oraciones en palabras y signos de puntuación. Los clasificadores de aprendizaje automático están entrenados en texto para aprender la diferencia entre las puntuaciones con o sin terminación. La normalización también incluye la conversión de palabras no estándar en lenguaje natural: números, abreviaturas y acrónimos. Finalmente, implica la desambiguación de los homógrafos, o palabras que se pronuncian de manera diferente y contienen un significado diferente según el contexto. Para estas palabras se puede ejecutar el análisis de etiquetado gramatical para etiquetar la parte del discurso del homógrafo, y al hacerlo, desambíguate su pronunciación (Cahn, 2017).
- Radicación o “Stemming”: similar a la lematización, pero más simple donde se quita el sufijo al final de la palabra, en reducir la palabra flexionada o derivar a su forma más básica. Los dos procesos implican análisis morfológico donde los tallos y afijos llamados morfemas se extraen y se usan para reducir las inflexiones a su forma básica (Lovins, 1968).

Figura 4

Ejemplo de Radicación



- Etiquetado Gramatical (EG) o “Part of Speech” (POS): etiqueta cada palabra en la cadena de entrada con su etiqueta (por ejemplo, sustantivo, verbo, adjetivo, etc.). Estas etiquetas pueden estar basadas en reglas creadas manualmente para especificar parte del discurso para palabras ambiguas dado su contexto. También se pueden crear usando modelos

estocásticos que se entrenan en oraciones etiquetadas con el EG correcto. Del mismo modo en la generación de respuestas para indicar el tipo de objeto EG de la respuesta deseada (Ramos & Velez, 2016).

Figura 5

Ejemplo de Etiquetado Gramatical

Det Sust. Adj. Verb. Det N. E. Det N.
Un hombre viejo compra un libro en su coche

- Segmentación morfológica: esta técnica se encarga de tomar cada palabra y dejarla en una unidad mínima o mejor conocida como morfemas la cual pueden tener una o más, capaz de expresar un significado de un idioma, como se puede observar en la Tabla 4 y 5 algunos ejemplos. Una palabra es independiente, cuando se sostiene por si misma se considera que esta en su raíz porque tiene su significado propio, a diferencia de si depende de otro morfema es un afixo y tiene una función gramatical (Ramos & Velez, 2016).

Tabla 4

Morfemas sufijos ejemplos

Morfema Sufijos	Significado	Ejemplo
-arquía	Refiere formas de gobierno	Monarquía, oligarquía
-cida, -cidio	Refiere al acto de matar	Insecticida, genocidio
-isimo	Indica grado superlativo	Buenísimo, altísimo
-ismo	Refiere a una doctrina política, religión	Budismo, cubismo
-itis	Indica inflamación u otra afección	Artritis, gastritis
-terapia	Formas de tratamiento	Radioterapia, musicoterapia
-logía	Significa ciencia	Ecología, geología

Nota. Sufijos del idioma castellano.

Tabla 5

Morfemas prefijos ejemplos

Morfemas Prefijos	Significado	Ejemplos
An-, a-	Indica negación	Anomia, analfabeto
Auto-	Significado de uno mismo o por uno mismo	Autónomo, autocrítico

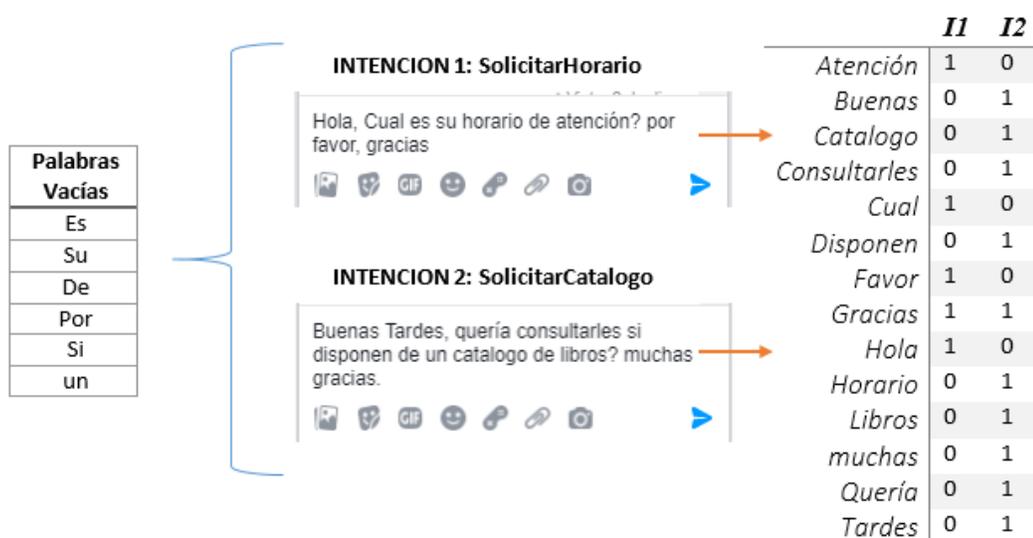
Anti-	Indica contrariedad u oposición	Antídoto, antinomia
Bi-	Significa doble o dos	Bicicleta, bisexual
Co-	Señala participación o unión	Coautor, cooperar
Hemi-	Significa la mitad de algo	Hemisferio, hemiciclo
Hipo-	Indica escasez	Hipotermia, hipócrita

Nota. Sufijos del idioma castellano.

- Palabras Vacías o “Stop Word” se identifican a las palabras sin significado como a, o, para, etc; que son los artículos, pronombres y preposiciones. La cual es muy utilizada en la técnica conocida como: bolsa de palabras (Gamallo Otero & García González, 2012).

Figura 6

Ejemplo Bolsa de Palabras descrito las palabras vacías y lematización



- Bolsa de palabras (BdP) o “Bag of Words” (BoW): las palabras son comparadas con la base de datos de conocimientos del agente conversacional para encontrar palabras claves similares. Se forma un modelo de espacio vectorial, en la que las palabras vacías se eliminan y las variantes morfológicas como puedo, pudo, podrían (conjugaciones) entran en un proceso de lematización que se encarga de realizar que cada uno de estos segmentos de palabras se pueda reducir a su forma base como el verbo poder. Ignoramos la estructura

de las oraciones, el orden y la sintaxis, y contamos el número de ocurrencias de cada palabra (Cahn, 2017) .

- Reconocimiento de entidades nombradas (REN) o “Named/Relation Entity Recognition” (NER): básicamente el reconocimiento de nombres de personas, lugares, grupos, ubicaciones, etc, se extraen y etiquetan en consecuencia. En la figura 7 se puede observar un ejemplo de REN, la cual utiliza diferentes aproximaciones para las diferentes entidades como patrones codificados con expresiones regulares para encontrar fechas, tiempo, velocidad, etc., listas ordenadas para nombres de personas, lugares , etc., y algoritmos de entropía máxima para clasificar tokens para entidades especiales (Ramos & Velez, 2016).

Figura 7

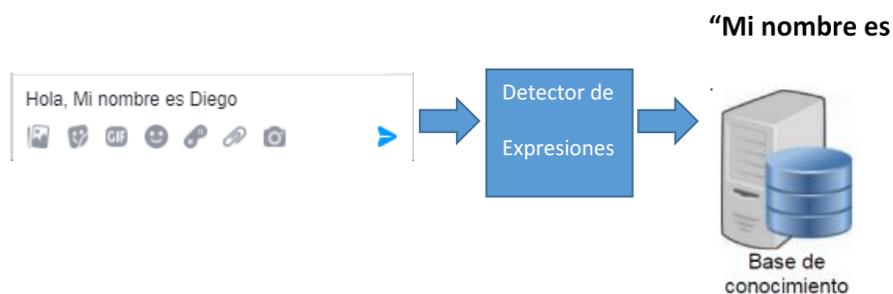
Ejemplo de REN



- Análisis semántico latente o “Latent Semantic Analysis” (LSA): los grupos de palabras que coexisten con frecuencia se agrupan. Se crea una matriz, la fila representa una palabra única, la columna representa un documento y el valor de cada celda es la frecuencia de la palabra en el documento (Pérez, Alfonseca, Gliozzo, & Rodríguez, 2005).
- Expresión regular o “Regular Expressions”: las oraciones se tratan como expresiones regulares para comparar patrones con la base de datos de conocimientos del agente conversacional (Chang & Manning, 2014).

Figura 8

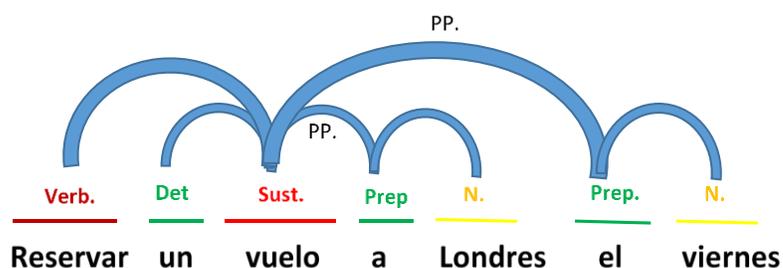
Ejemplo de Expresión Regular



- Creación de estructuras de datos gramaticales o “Creation of Grammatical Data Structures”: las oraciones se pueden almacenar en una forma estructurada en formalismos gramaticales como gramáticas libres de contexto que son datos en forma de árbol de una oración que tienen frases nominales y verbales que contienen sustantivos, verbos, etc., y dependencia gramatical la cual se centra en la relación entre palabras (Cahn, 2017).

Figura 9

Ejemplo Dependencia Gramatical



Implementación de agentes de auto-respuesta

Un agente de auto respuesta es un software que imita una conversación inteligente con un usuario (A. Augello, 2012). Este integra técnicas computacionales tales como: procesamiento de lenguaje natural, inteligencia artificial, sistemas expertos con una base de conocimiento (corpus), para de esta forma interpretar y dar respuesta a las peticiones realizadas por los usuarios a través de canales de mensajería como: Facebook Messenger, Whatsapp, Correos electrónicos, entre otros (Gregoire , y otros, 2015) y (Tear, 2018).

El funcionamiento general de un agente conversacional cuenta tres procesos generales: recibir petición, analizar petición, y responder ante la petición. En el análisis de la petición se examinan los enunciados que la componen, y de los cuales se extraen intenciones y entidades, para así ofrecer una respuesta a la petición (McTear, 2018).

Un enunciado o “Utterance”, es cualquier cosa que el usuario escriba al agente conversacional, puede ser una oración completa o una expresión más pequeña. La intención o “Intent”, es el deseo del enunciado que ingresó el usuario; por lo general suele estar constituida por un verbo y un sustantivo. Mientras que la entidad también conocida como “Entity”, actúa como modificador de la intención de un enunciado, logrando así obtener una intención más implícita (Gregoire , y otros, 2015).

A continuación, en la Figura 10 se presenta un ejemplo de cómo un agente conversacional interpreta una petición ingresada por el usuario.

Figura 10

Componentes de un enunciado



Se puede observar que del enunciado: “Yo deseo comprar un libro del autor J.K. Rowling”, las palabras: “deseo, comprar, libro” forman parte de la Intención: “ComprarLibro”. Mientras que “autor, J.K. Rowling” pasan a ser la Entidad del enunciado. Entonces, un agente conversacional debe responder a la petición “Comprar libro de J.K. Rowling”, con las respuestas que se han establecido en base de conocimiento.

Tipos de agentes de auto-respuesta

Los agentes conversacionales pueden ser categorizados según: la tecnología que utilizan, su diseño de interfaz, y el tipo de servicio al cuál están orientado. A continuación, se detallan los tipos que de agentes que pueden existir dentro de las categorías anteriormente mencionadas.

Agentes conversacionales según la tecnología utilizada.

- Agente Conversacional Simple: son agentes sencillos cuyo funcionamiento se basa en una encontrar coincidencias de patrones básicos que están asignados a una respuesta (Zarabia Zuñiga, Abril, 2018).
- Agente Conversacional Complejo: son más sofisticados pues utilizan técnicas de IA con un complejo seguimiento del estado conversacional, además de la integración a servicios que la empresa ofrece (Zarabia Zuñiga, Abril, 2018).

Agentes conversacionales según su diseño de Interfaz.

- Con interfaz de solo texto: estos agentes se comunican con los clientes haciendo uso únicamente de entradas y salidas de texto (Zarabia Zuñiga, Abril, 2018).
- Con interfaz combinada texto, botones e imágenes: interactúan con los usuarios mediante diálogos que pueden contener texto, imágenes y botones de acción, que permiten a los clientes tener una comunicación más guiada durante la experiencia.

Agentes conversacionales según el tipo de servicio.

- Informativos: estos cumplen funciones específicas complejidad mínima, haciendo uso de un sistema de pregunta-respuesta (QA), el cual está diseñado para dar respuesta a las preguntas frecuentes (PF) de una empresa, también son conocidos como “frequent asked question” (FAQ) (Microsoft, Microsoft , 2018).

- Operativos o Empresariales: ayudan o facilitan el acceso a los servicios de una empresa, logrando así, mejorar los tiempos de respuesta en canales de mensajería (Zarabia Zuñiga, Abril, 2018).
- Comercio electrónico o “e-commerce”: son asistentes virtuales comerciales, donde su propósito es ayudar a los clientes en la compra de productos de la empresa.

Características de un agente de auto-respuesta

Un agente conversacional puede tener una o más características que le permiten desempeñar múltiples funciones, y así, mejorar la experiencia del cliente. Entre las más relevantes se tiene a: Adaptabilidad, Expresividad, Racionalidad, Proactividad, y Personalizable (Torres, 2013). Estas características son descritas a continuación.

- Adaptabilidad: es la capacidad de aprendizaje que puede tener un agente conversacional, por lo que puede variar su comportamiento en función de lo aprendido (Torres, 2013).
- Expresividad: aprovecha los elementos que poseen las canales de mensajería para poder utilizar gráficos, videos y sonidos para así tener una experiencia más expresiva (Torres, 2013).
- Racionalidad: intenta dar la respuesta más apropiada a partir de los datos de entrada (Torres, 2013).
- Proactividad: es capaz de exhibir un comportamiento dirigido a un objetivo, al tomar la iniciativa en una conversación (Torres, 2013).
- Personalizable: el agente puede tener las de características que su programador desee, tales como: mostrar emociones, interpretar sentimientos, intenciones, o tener un comportamiento no verbal (Torres, 2013).

Definición de Dominio

Los agentes de auto-respuesta cumplen funciones basadas en el dominio para el que fueron programados. Existen dos clases de dominios: Dominio abierto o “Open Domain” y Dominio cerrado o “Closed Domain”.

Los agentes conversacionales de dominio abierto poseen la capacidad de conversar libremente en lenguaje natural, es decir estos pueden entablar una conversación sobre cualquier tema. Algunos ejemplos como: Milabot, Xiaolce muestran características similares a los seres humanos, pero se basan en marcos complejos como administradores de diálogos con sistemas basados en conocimiento, recuperación o reglas. La desventaja de los agentes conversacionales de dominio abierto es que a menudo responden a las preguntas abiertas de forma incongruente, o con respuestas que son vagas y genéricas (Adiwardana, y otros, 2020).

A diferencia de los agentes conversacionales de dominio abierto, los de dominio cerrado responden a palabras clave o intentos de realizar tareas específicas. Estas palabras clave pertenecen a expresiones que se encuentran categorizadas en un corpus lingüístico. Generalmente cuando un humano pregunta algo fuera de su dominio responden con una disculpa (Ono, Takeda, Nichols, Nakano², & Komatani, 2016). Aunque los agentes conversacionales de dominio cerrado tienen un amplio conocimiento léxico en sus dominios, no es posible prepararlos por adelantado a términos que estén fuera del vocabulario del sistema (Tamayo & Pérez-Marín, 2013).

Construcción de personalidades

Según la Real Academia Española (RAE), el significado de la palabra “personalidad” es: “Conjunto de características o cualidades originales que destacan en algunas personas” (Real Academia de la Lengua, 2020). Sin embargo, en la actualidad es muy común usar este término en agentes de auto respuesta, por lo tanto, su personalidad está formada por el conjunto de

características tales como: vocabulario, tono, estilo, conocimiento. Además, es posible crear una personalidad mucho más humana al asignar al agente conversacional un nombre, edad y género (Moreno & Antón, 2018).

Por lo general la personalidad de los agentes conversacionales está diseñada para llegar al público objetivo de la empresa, además del propósito para el cual está programado. No obstante, en algunos casos los agentes conversacionales adaptan su personalidad a la forma en la que el usuario realiza sus peticiones. Cabe mencionar que la personalidad del agente conversacional influye en la conexión emocional con los clientes, pues de la calidad de la interacción puede llegar a ser un factor importante en el índice de ventas de la empresa (Yorita, Oakman, CHan, & Kubota, 2019).

Capítulo III

Implementación de agente de auto-respuesta para empresas comerciales

Este capítulo se aborda el desarrollo del agente de auto respuesta para empresas comerciales, en específico una tienda de libros, partiendo desde el levantamiento de información de los requerimientos de la empresa, la definición del dominio del agente, y la estructura general del sistema.

Levantamiento de información

Para el levantamiento de información, se solicitó a la tienda, que proporcione una lista de las preguntas y frases más comunes que recibe su establecimiento “FAQ-Frequently Asked Questions” o (PF) Preguntas Frecuentes como se le denominó en el Capítulo II, las cuales se muestran a continuación en la Tabla 6. Las solicitudes del local, son: Responder a preguntas frecuentes con el trato al cliente de manera culta. En base a esta información se procede a la definición del dominio del agente conversacional, misma que se puede encontrar en la sección 3.2.

Tabla 6

Preguntas frecuentes de una librería

N°	Frases y preguntas frecuentes de una librería
1	¿Dónde están ubicados?
2	¿Cuál es su horario de atención?
3	¿Cuál es su número de teléfono?
4	¿Hacen entregas a domicilio?
5	¿Disponen de un catálogo?
6	¿Venden libros en formato digital?
7	¿Venden audiolibros?
8	¿Venden libros de segunda mano?
9	¿Tienen el libro “nombre del libro”?
10	Quiero comprar el libro “Nombre del libro”
11	¿Tienen club de lectura

Definición de Dominio

En función a los requerimientos de la empresa es necesario iniciar el desarrollo del agente conversacional a partir de un dominio cerrado. Es decir, está enfocado en responder a: saludos, despedidas y las PFs de la librería, utilizando un lenguaje cordial. A continuación, se detalla en la Tabla 7 la clasificación de las intenciones para cada PF, asignado un nombre único para su distinción y una posible respuesta para las mismas.

Con la finalidad de dar un corpus más variado y natural al entrenamiento del agente conversacional, se realizó una encuesta a un grupo de personas, en la cual se solicitó el replanteamiento de cada una de las PFs. Por esta razón, cada Intención, tiene variaciones de sus PFs correspondientes como se observa en la Tabla 7. La encuesta mencionada se la puede encontrar en el siguiente enlace: <https://forms.gle/BsBDCBHKwmDm852D8>.

Tabla 7

Clasificación de PFs por Intención.

Intención	Frases y preguntas frecuentes de una librería
Saludar	Hola, Buenas tardes, Buenos días, etc.
Despedir	Adiós, Hasta pronto, gracias, etc.
SolicitarUbicacion	¿Dónde están ubicados?, etc.
SolicitarHorario	¿Cuál es su horario de atención?, etc.
SolicitarTelefono	¿Cuál es su número de teléfono?, etc.
SolicitarDomicilio	¿Hacen entregas a domicilio?, etc.
VerCatalogo	¿Disponen de un catálogo?, etc.
SolicitarFormatoDigital	¿Venden libros en formato digital?, etc.
SolicitarAudioLibro	¿Venden audiolibros?, etc.
VerLibrosUsados	¿Venden libros de segunda mano?, etc.
BuscarLibro	¿Tienen el libro “nombre del libro”?, etc.
SolicitarClub	Quiero comprar el libro “Nombre del libro”, etc. ¿Tienen club de lectura?, etc.

Una vez establecido el dominio de preguntas frecuentes, se puede determinar las respuestas que debe entregar el agente conversacional. Cabe mencionar que el ejemplo de respuestas no es único, se puede agregar más respuestas en base a los requerimientos de la

empresa. Para que de esta forma el sistema pueda escoger una respuesta de forma aleatoria del grupo de contestaciones, y así mejorar la experiencia del cliente.

Tabla 8

Clasificación de Intención con sus respuestas.

Intención	Respuestas
Saludar	Hola, ¿En qué le podemos ayudar?.
Despedir	Un placer atenderle.
SolicitarUbicacion	Estamos ubicados en “ <i>Dirección</i> ”.
SolicitarHorario	De lunes a viernes de 9 am a 5 pm.
SolicitarTeléfono	Fijo 02xxxxxxx o Whatsapp 09xxxxxxx.
SolicitarDomicilio	Por el momento no entregamos a domicilio.
VerCatálogo	Puede ver nuestro catálogo en el siguiente link ...
SolicitarFormatoDigital	Si, disponemos libros en formato digital.
SolicitarAudioLibro	Por el momento no disponemos.
VerLibrosUsados	Si disponemos de libros usados.
BuscarLibro	Si disponemos el libro. No disponemos.
SolicitarClub	Si claro contamos con el ello.

Nota. Esta tabla muestra las preguntas del agente conversacional.

Adicionalmente se agregó al corpus de entrenamiento, ciertas peticiones que a priori no pertenecen a las preguntas frecuentes de la tienda de libros. Sin embargo, se considera menester su añadidura, para de esta forma mejorar la experiencia del cliente con el agente conversacional automático. Las peticiones mencionadas se encuentran estructuradas junto a sus respectivas respuestas en la Tabla 9. Las peticiones que se encuentran fuera del dominio del agente conversacional, tienen como respuesta con una disculpa al no poder reconocer lo solicitado.

Tabla 9

Peticiones no comunes

Intención	Peticiones no comunes	Respuesta
SolicitarHumano	Quiero Hablar con una persona real	Por el momento solo atiendo yo.
PreguntarNombre	¿Quién eres?	Mi nombre es Rob.
PreguntarJefe	¿Quién es tu jefe?	Mi jefe es Verónica Ruiz.

Herramientas para el desarrollo

En este apartado se describe las herramientas necesarias para el desarrollo de la app agente conversacional y su integración con la plataforma Facebook, tales como: editor de texto, gestor de base de datos, entre otras. De las herramientas seleccionadas se tiene las siguientes: NodeJS, MongoDB, Visual Studio Code, Ngrok. La función de cada una de estas es detallada a continuación.

Visual Studio Code

Para la edición del código fuente de la aplicación se eligió el software Visual Studio Code (VSC), por tener compatibilidad de lenguaje de programación JavaScript, estar diseñado como aplicación multiplataforma, ser de licencia libre e incluir herramientas como: finalizado inteligente de código, resaltado de sintaxis, entre otras. Una de las ventajas de esta herramienta es la depuración de código desde el editor, para lo cual únicamente se necesita iniciar aplicación, a la cual se puede agregar puntos de interrupción, llamadas a la pila, además de contar con una consola interactiva (Microsoft, 2020).

Figura 11

Logo de Visual Studio Code



Nota. Tomado de (Microsoft, 2020)

Esta aplicación cuenta con extensiones con las cuales se pueden agregar nuevos idiomas, temas, depuradores, e incluso conectar a servicios adicionales. Para garantizar la rápida ejecución de Visual Studio Code, dichas extensiones se ejecutan en procesos separados. Entre las

extensiones más populares se tiene a: depurador para Google, GO de Microsoft, Azure Tools, Phyton, Docker, entre otros (Microsoft, 2020).

Visual Studio Code permite compartir en tiempo real con otros desarrolladores mediante la conexión a la red social de programadores “GitHub”, permitiendo así clonar proyectos, analizar y editar código, todo esto sin la necesidad de instalar herramientas adicionales (Microsoft, 2020).

NodeJS

Es un entorno de tiempo de ejecución en JavaScript para la capa de servidor basado en el motor V8 de Google. Se hace uso de esta herramienta por recomendación de la documentación de Facebook para la conexión de una aplicación de terceros con la plataforma de Messenger, ya que se ha amoldado para funcionar en una amplia variedad de lugares (NodeJs, 2020).

Figura 12

Logo de Nodejs



Nota. Tomado de (NodeJs, 2020)

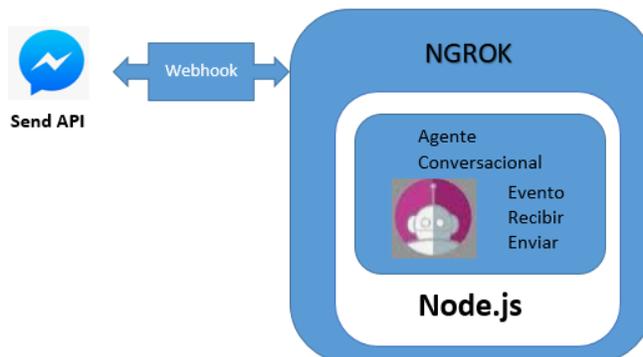
Una aplicación de Node se ejecuta en un solo proceso como realizar operaciones de E/S por ejemplo leer desde la red, acceder a una base de datos o al sistema de archivos, en vez de bloquear el hilo y desperdiciar ciclos de CPU, este reanuda la operación cuando regrese la respuesta, permitiendo manejar miles de conexiones simultaneas (NodeJs, 2020).

Además, se han creado miles de bibliotecas sobre Node.js al ser una plataforma de bajo nivel. NPM es el administrador de paquetes estándar donde dispone más de 350000 en el repositorio de código de un solo idioma y casi para cualquier funcionamiento ya sea instalado de forma local o global (NodeJs, 2020).

En la figura 13 se observa gráficamente el proceso del sistema y la función que cumplirá el Node.js dentro del mismo.

Figura 13

Función Node.js



MongoDB

Es un sistema de bases de datos no relacionales NoSQL, orientado a documentos y de código abierto. El almacenamiento de datos se efectúa en una estructura parecida a un archivo JSON, por esta razón el flujo de datos con una aplicación de NodeJS no tiene cambios en la disposición de datos. Además, cuenta con soporte para búsquedas modernas tales como: búsqueda de gráficos, texto, y basadas en geografía (MongoDB, 2020).

Figura 14

Logo de MongoDB



Nota. Tomado de (MongoDB, 2020)

Esta herramienta cuenta con un servicio de base de datos en la nube, conocido como MongoDB Atlas, el cual es usado en aplicaciones modernas por su fácil implementación, uso y

escalamiento, además está disponible para AWS, Azure, Google Cloud Service. Para los desarrollos que ejecutan la base de datos de MongoDB en un servidor, existe una aplicación la cual permite editar los datos de los documentos a través de una aplicación con interfaz gráfica de nombre MongoDB Compass, misma que permite la toma de decisiones inteligentes sobre indexación, validación de documentos, entre otras (MongoDB, 2020).

MongoDB es la base de datos más utilizada en los desarrollos actuales, algunas de las empresas de gran reconocimiento que trabajan con este servicio son: SEGA®, Adobe®, Google, EA™, ebay®, Verizon. Por esta razón, se eligió esta base de datos para alojar el catálogo de libros de la empresa (MongoDB, 2020).

Ngrok

Ngrok es una aplicación ejecutable que permite exponer nuestro servidor local (localhost) través de un túnel a cualquier persona en Internet, la cual debe disponer una URL que el aplicativo genera de forma dinámica. Este programa es útil para que las aplicaciones en etapa de desarrollo puedan mostrar sus avances sin necesidad de configuraciones adicionales en el router o el firewall. El objetivo del uso de este software es el de generar una dirección externa que es usada como URL de devolución de llamada para la configuración del Webhook y de esta forma comunicar el chatbot con la plataforma de Messenger (Ngrok, 2020).

Figura 15

Logo Ngrok

The logo for Ngrok, featuring the word "ngrok" in a lowercase, bold, orange sans-serif font.

Nota. Tomado de (Ngrok, 2020)

Esta herramienta es comúnmente usada en el desarrollo de aplicaciones para la detección de intrusos en dispositivos móviles Android (LARA & SALINAS, 2018), incluso es utilizada en áreas

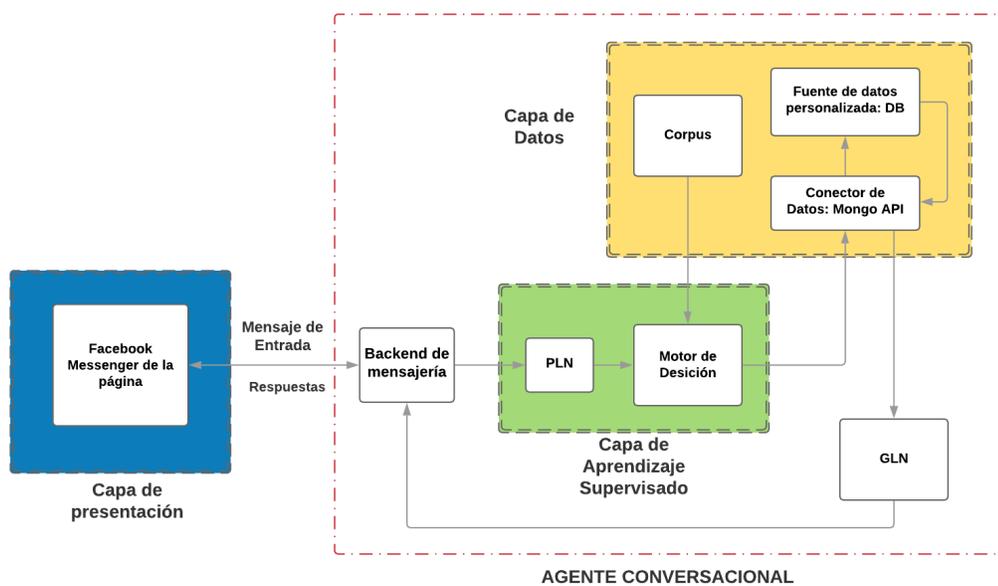
de ingeniería de control en las cuáles se requiere enviar datos a través de puertas de enlace de internet de las cosas (JAQUE, 2016).

Desarrollo del agente conversacional

Este apartado contempla la etapa técnica de desarrollo del agente conversacional, partiendo del diagrama de bloques que se muestra en la Figura 16. En el cual se puede observar los principales elementos del sistema, tales como: capa de presentación, backend de mensajería, capa de aprendizaje supervisado, capa de datos y el generador de lenguaje natural.

Figura 16

Diagrama de Arquitectura



Capa de datos

Como se puede observar en la Figura 16, esta capa es la encargada de albergar los datos del corpus de entrenamiento del agente conversacional, además de la base de datos de libros de la empresa. A continuación, se describe cada uno de estos componentes.

Corpus de entrenamiento.

El corpus de entrenamiento del agente conversacional está formado por la categorización definida en la sección 3.2. Para la estructura de los datos se utiliza el sistema que se puede observar en la figura 17, mismo que se encuentra guardado en un archivo de tipo JSON. Dónde: el subíndice m representa el m -ésimo Intención del corpus, n simboliza la n -ésima PF de su correspondiente Intención, mientras que j indica la j -ésima respuesta a la Intención pertinente.

Figura 17

Estructura del corpus del agente conversacional.

```

1  {
2    "corpus" : [
3      {
4        "intención" : "Intención_1",
5        "PFs" : [ "PF_11", ... , "PF_1n"],
6        "respuestas": [ "R_11", ... , "R_1j" ]
7      },
8      .
9      .
10     .
11     {
12       "intención" : "Intención_m",
13       "PFs" : [ "PF_m1", ... , "PF_mn "],
14       "respuestas": [ "R_m1", ... , "R_mj"]
15     }
16   ]
17 }
```

Base de datos de libros.

Para la recuperación de información de la base de datos de MongoDB desde Nodejs, es necesario utilizar la librería “Mongoose”, la cual, a partir de un esquema y un modelo puede realizar operaciones CRUD con la base de datos. La estructura de la información de cada libro (documentos) está basado en el esquema que se puede observar en la figura 18. Los documentos de la colección de libros pueden estar en constante crecimiento en función del “stock” de la librería.

Figura 18

Estructura de un documento de la colección libros.

```
1  {  
2  |   "autor": String,  
3  |   "nombre": String,  
4  |   "genero": String,  
5  |   "precio": Double  
6  }
```

Capa de aprendizaje supervisado

Esta capa es la encargada del aprendizaje automático del agente conversacional. Como se puede observar en la Figura 16 esta capa está conformada por dos bloques principales: El primero se encarga de realizar el PLN a las preguntas provenientes del backend de mensajería, mientras que el segundo bloque tiene como objetivo determinar a qué Intención del corpus pertenecen las palabras procesadas por su bloque antecesor.

Bloque de PLN.

Para el desarrollo del bloque PLN es necesario la instalación de la librería “Lorca”, que se encuentra en el repositorio de Nodejs. Esta librería es una biblioteca de PLN que ayuda al procesamiento de textos con las técnicas de: separación de oraciones, segmentación por palabras, y Radicación. La radiación está basada en el algoritmo Porter para idioma español.

En el diagrama de flujo de la Figura 19. Se puede observar que al iniciar el proceso este recibe una cadena de texto (enunciado) ejecuta las técnicas de PLN tales como: Segmentación de palabras; normalización, es decir, cambiar todas las palabras a minúsculas; hallar las raíces de cada “token” (radicación); excluir las palabras vacías y eliminar las palabras repetidas. Cabe recalcar que el proceso de radicación se lo realiza con la finalidad de que todas las palabras del corpus y base de datos de libros sean reducidas a su raíz, con lo cual no es necesario escribir las PFs con varias conjugaciones de verbos, tipos de adjetivos, o el número gramatical (singular o

plural). Con esto se consigue que la clasificación de palabras sea más rápida, al tener un número reducido de palabras en el corpus para analizar.

Figura 19

Diagrama de flujo del bloque de procesamiento del lenguaje natural



Al finalizar la función se retorna un arreglo con la lista de palabras totales $palabras_{lista}$, un objeto $PFs_{palabras}$ en el cual tiene como valor todas palabras que lo conforman cada PF . Cabe recalcar que estas palabras son las procesadas con PLN. Adicional a esto se retorna un último objeto $Intenciones_{PFs}$, el cual contiene el conjunto de PFs que conforman cada Intención.

$$palabras_{lista} = \begin{bmatrix} palabra_1, \\ palabra_2, \\ \vdots \\ palabra_n \end{bmatrix} \quad (3.1)$$

$$PFs_{palabras} = \left\{ \begin{array}{l} "PF": [palabra, palabra_{12}, \dots, palabra_{1m}] \\ "PF": [palabra_{21}, palabra_{22}, \dots, palabra_{2m}] \\ \vdots \\ "PF": [palabra_{n1}, palabra_{n2}, \dots, palabra_{nm}] \end{array} \right\} \quad (3.2)$$

$$Intenciones_{PFs} = \left\{ \begin{array}{l} "Intención": [PF, PF_{12}, \dots, PF_{1m}] \\ "Intención": [PF_{21}, PF_{22}, \dots, PF_{2m}] \\ \vdots \\ "Intención": [PF_{n1}, PF_{n2}, \dots, PF_{nm}] \end{array} \right\} \quad (3.3)$$

Motor de decisión.

El motor de decisión usa el clasificador de Naive Bayes descrito en la sección 2.1.2.3, con el cual a partir de un conjunto de palabras de entrada es posible hallar la probabilidad de que dichas palabras se encuentren dentro de una Intención del corpus. Las fórmulas, modelos y diagramas de flujo presentados a continuación están basados en la literatura de (Garcia Serrano, 2017).

La programación del clasificador está dividida en dos funciones principales, mismas que se enlistan a continuación:

1. Entrenamiento
2. Clasificación

La función “Entrenamiento” se encarga de generar un modelo para ser usado por las fórmulas de probabilidad que componen al clasificador. Es importante mencionar que esta función recibe como entrada: el corpus procesado por el PLN y retorna como salida un objeto con cuatro parámetros, que se detallan a continuación:

- a. c_{textos} : Este parámetro registra el número total de PFs del corpus. El valor de PFs para el agente conversacional se muestra a continuación en la ec. 3.4.

$$c_{textos} = 320 \quad (3.4)$$

- b. $c_{total\ palabras}$: Esta variable contiene el número total de palabras de la BdP extraído a partir del corpus. Para este caso de estudio la cantidad de palabras se muestra a continuación en la ec. 3.5.

$$c_{total\ palabras} = 152 \quad (3.5)$$

- c. $c_{palabras}$: Este parámetro figura como la BdP del corpus, la estructura de datos de esta variable se muestra en la ec 3.6. Donde $palabra_m$ representa la última palabra de la lista entregada por el bloque PLN; mientras que $score$ hace referencia a un arreglo que contiene la frecuencia de ocurrencia de una palabra del BdP en cada Intención, por tales motivos fp_{mi} representa el valor de las veces se repite $palabra_m$ en el $Intención_i$.

$$c_{palabras} = \left\{ \begin{array}{l} \{"palabra": palabra_1, "score": [fp_{11} \quad fp_{12} \dots fp_{1i}]\}, \\ \{"palabra": palabra_2, "score": [fp_{21} \quad fp_{22} \dots fp_{2i}]\}, \\ \vdots \\ \{"palabra": palabra_m, "score": [fp_{m1} \quad fp_{m2} \dots fp_{mi}]\} \end{array} \right\} \quad (3.6)$$

- d. $c_{categorias}$: Corresponde a un objeto que contiene todos las Intenciones del corpus junto a la cantidad de PFs que los conforman. Está expresión es definida en la ec. 3.7. Donde: $Intencion_i$ representa la última Intención del corpus, acompañado a este se encuentra el parámetro $NumPFs$, el cual registra el número total de preguntas frecuentes de la $Intención_n$.

$$c_{categorias} = \left\{ \begin{array}{l} \{"Intención": Intención_1, "NumPFs": 15\}, \\ \{"Intención": Intención_2, "NumPFs": 12\}, \\ \vdots \\ \{"Intención": Intención_i, "NumPFs": 20\} \end{array} \right\} \quad (3.7)$$

El objetivo de la función “Clasificación” es encontrar la Intención al cuál pertenecen un grupo palabras de entrada, en base a la aplicación del modelo de entrenamiento en el cálculo de probabilidades de la ec. 3.8, la cual es una forma de reescribir la ec. 2.1, de la sección 2.1.2.3. Donde *PALABRAS* representa un arreglo con las palabras extraídas mediante PLN de la pregunta ingresada por el usuario desde el chat de Messenger tal como se puede observar en la ec 3.9; mientras que *palabra_{ji}* simboliza la palabra *j* analizada durante el análisis del *Intención_i*.

$$P(\text{Intención}_i | \text{PALABRAS}) = P(\text{Intención}_i) \times \prod_{j=0}^n P(\text{palabra}_{ji} | \text{Intención}_i) \quad (3.8)$$

$$\text{PALABRAS} = [\text{palabra}_1, \text{palabra}_2, \text{palabra}_3, \dots, \text{palabra}_n] \quad (3.9)$$

Para encontrar $P(\text{Intención}_i)$, es menester definir el espacio muestral del experimento aleatorio, por tal razón se define la expresión *corpus* de la ec. 3.10, la cual incluye todas las PFs definidas en el archivo corpus.json. La ec. 3.11 representa el conjunto solución del suceso *Intención_i*, este contiene los PFs que componen dicha Intención. Una vez definidas estas expresiones, es posible hallar $P(\text{Intención}_i)$ mediante la regla de Laplace tal como se puede observar en la ec 3.12, Donde *card* representa el cardinal de un conjunto de soluciones, es decir el total de elementos que componen dicho conjunto.

Al aplicar el modelo generado por la función de entrenamiento, se obtiene que: $\text{card}(\text{Intención}_i)$ es igual al parámetro *NumPFs* del objeto *i* de *c_{categorias}*. Mientras que $\text{card}(\text{corpus})$ está definido por la variable *c_{textos}*.

$$\text{corpus} = [PF_1, PF_2, \dots, PF_n] \quad (3.10)$$

$$\text{Intención}_i = [PF_s \text{ del Intención}_i] \quad (3.11)$$

$$P(\text{Intención}_i) = \frac{\text{card}(\text{Intención}_i)}{\text{card}(\text{corpus})} = \frac{\text{NumPFs}_i}{c_{\text{textos}}} \quad (3.12)$$

El espacio muestral necesario para encontrar la probabilidad del suceso $palabra_{ji}$, viene dado por la expresión BdP de la ec. 3.13, el cual contiene la lista de palabras de $c_{palabras}$.

Para definir el conjunto solución del suceso $palabra_{ji}$, en primer lugar, se debe excluir del arreglo PALABRAS, aquellas palabras que no pertenecen a la lista de la BdP, esto con la finalidad de no permitir que $P(palabra_{ji})$ tome el valor de cero, lo que conllevaría a que $P(Intención_i|palabra_{ji})$ tome el mismo valor, al ser esta una expresión que está formada por la multiplicación de factores, por lo que bastaría tener al menos una palabra que no pertenezca a la BdP, para que el sistema crea que no existe ninguna coincidencia de palabras.

Una vez realizada la exclusión se toma la $palabra_j$ para ser comparada con cada una de las PFs del $Intención_i$, del cual se extraen aquellas palabras que tienen coincidencia con $palabra_j$, con las cuales se forma el conjunto que se muestra en la ec 3.14, donde m representa el índice de la BdP de la palabra coincidente, mientras que n indica la n -ésima palabra de coincidencia.

De esta forma $P(palabra_{ji})$ queda definida por la expresión 3.15, donde al aplicar el modelo de entrenamiento se tiene que $card(palabra_{ji})$ es igual al valor fp_{mi} del parámetro score perteneciente al objeto $c_{palabras}$, y $card(BdP) = c_{total\ palabras}$.

$$BdP = [palabra_1, palabra_2, \dots, palabra_n] \quad (3.13)$$

$$palabra_{ji} = [palabra_{m1}, \dots, palabra_{mn}] \quad (3.14)$$

$$P(palabra_{ji}) = \frac{card(palabra_{ji})}{card(BdP)} = \frac{fp_{mi}}{c_{total\ palabras}} \quad (3.15)$$

Para encontrar el productorio de la ec. 3.8, es importante definir la expresión condicional $P(palabra_j|Intención_i)$, misma que se puede observar en la ec 3.16, de la cual solo queda por

definir la expresión $P(Intención_i|palabra_{ji})$, su equivalencia es mostrada a continuación en la expresión de la ec 3.17.

$$P(palabra_j|Intención_i) = \frac{P(Intención_i|palabra_{ji}) \times P(palabra_{ji})}{P(intención_i)} \quad (3.16)$$

$$P(Intención_i|palabra_{ji}) = \frac{P(palabra_{ji})}{P(intención_i)} \quad (3.17)$$

Al finalizar todas las operaciones matemáticas la función clasificar retorna el arreglo clasificador, mismo contiene las probabilidades de cada Intención, su estructura es mostrada a continuación en la ec. 3.18. Este arreglo es interpretado por el bloque GLN para encontrar la respuesta correcta a la solicitud ingresada por el usuario desde Messenger. El diagrama de flujo de la función clasificación se muestra en la Figura 21.

$$clasificador = \begin{bmatrix} P(Intención_1|PALABRAS) , \\ P(Intención_2|PALABRAS) , \\ \vdots \\ P(Intención_i|PALABRAS) \end{bmatrix} \quad (3.18)$$

GLN del agente conversacional

Para el proceso de GLN recibe como entrada el Enunciado de Messenger, el corpus con las PFs y el clasificador de Intenciones. En base al array del clasificador se puede elegir la probabilidad máxima con la cual se puede encontrar el índice de la Intención del Corpus, en el caso de que todas las probabilidades sean cero, esto quiere decir, que el Enunciado ingresado desde Messenger no corresponde a ninguna de las PFs del Corpus. Por lo que, fuera del dominio del agente conversacional.

Cuando la probabilidad más alta corresponde a la Intención “ComprarLibro” se debe hallar la Entidad correspondiente al libro solicitado, esto se realiza a través de un segundo clasificador que como base de entrenamiento contiene todos los libros de la tienda junto a su

correcta escritura. De no encontrarse dicho libro en la base de entrenamiento el agente conversacional responderá: "No disponemos el libro solicitado". Caso contrario, el clasificador de Entidad retorna la correcta escritura del nombre del libro para que la búsqueda coincida con los nombres registrados en la base de datos y de esa forma entregar una respuesta con la disponibilidad y el precio actual del libro registrado en la base de datos. Esta respuesta se retorna como texto plano al backend de Mensajería para que pueda ser enviado al Messenger de Facebook a través de un evento de Webhook. A continuación, en la Figura 20 se detalla en un diagrama de flujo el proceso de GLN.

Figura 20

Diagrama de flujo del generador del lenguaje natural.

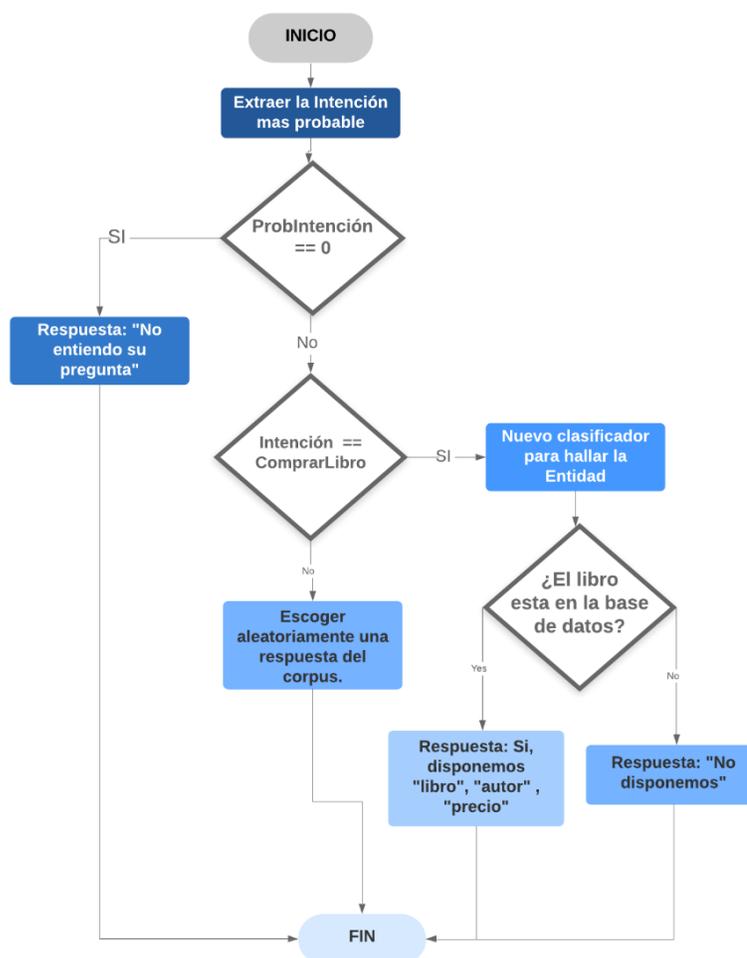
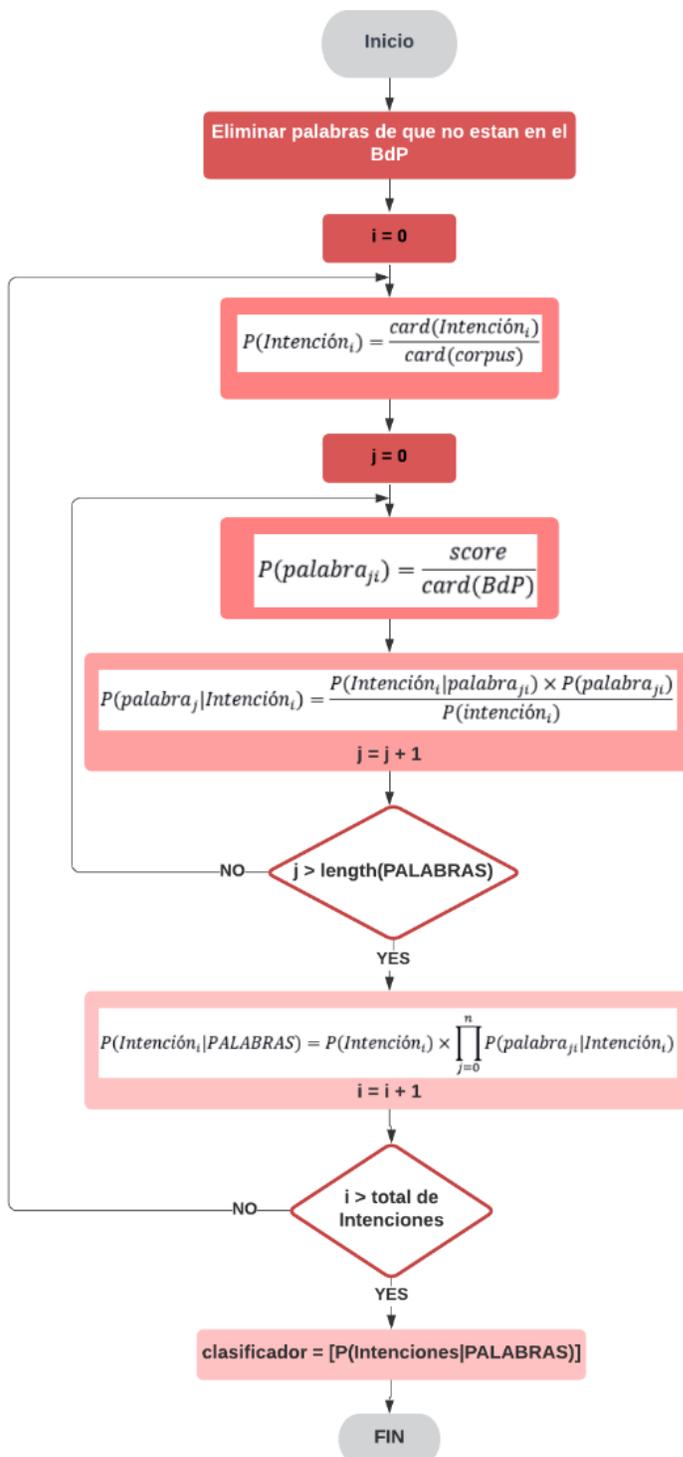


Figura 21

Diagrama de flujo del clasificador de Naive Bayes



Capítulo IV

Integración con Facebook

Facebook

Facebook es una de las redes sociales más grande y popular que existe, cuyo objetivo principal es producir y compartir contenido, es decir, conectar personas. Esta contiene dos tipos de cuenta que son: cuenta de perfil personal que permite la interacción entre personas, publicación de texto, imagen y video totalmente gratis; y la cuenta que puede crear las empresas o también conocida como página para seguidores “Fanpage”, con dos versiones: gratuita o pagada, para ofrecer productos o servicios con el objetivo de tener un mayor alcance de la compañía (Kirkpatrick, 2010).

En febrero del 2004 nace esta plataforma gracias a Mark Zuckerberg con la ayuda de sus compañeros de habitación Andrew McCollum, Eduardo Saverin, Chris Hugues y Dustin Moskovitz, mientras eran estudiantes de la universidad de Harvard inicialmente como un entorno para los alumnos de la misma, pero quería expandirse, en octubre del mismo año, logro conseguir el dinero, la mano de obra y la capacidad institucional respaldada para globalizarse, donde en septiembre del 2006, anuncio que cualquier persona con un correo electrónico valido podía registrar y crear su propio perfil (Crof, 2007).

Actualmente, Facebook maneja a su disposición otras aplicaciones de mercado como Instagram y Whatsapp que son las más conocidas, además de otros servicios de conexión a Internet, las cuales son hasta 80 empresas compradas con la estrategia se seguir promoviendo a Facebook (Llorens Cerdà, 2011).

Facebook Messenger es su principal aplicación de mensajería instantánea de comunicación en tiempo real y es utilizado por más de 900 millones de personas, al cual, en abril 2016, se le agregó Facebook Messenger Bot, que son programas interactivos y automatizados con

diferentes funcionalidades para el usuario, para así llegar a más personas con respecto a un cliente en partículas, para la comunicación uno a uno o para transmitir mensajes a todos los usuarios suscritos. (Hoyle, Das, Kapadia, Lee, & Vaniea, 2017).

Existe dos formas de implementar este agente conversacional, la cual, para el presente proyecto se ha escogido como el más adecuado, el utilizar la guía para desarrolladores de mensajería de Facebook, “Facebook for developers”, ya que se obtiene más libertad para el desarrollo y contiene una guía de inicio rápido, que te ayuda a crear un bot paso a paso, siguiendo instrucciones sencillas. Sin embargo, hay una guía completa con más detalles y opciones sobre la configuración de pagos y análisis (Vukovic & Dujlovic, 2016).

Perfiles públicos de empresas

El perfil de empresa de Facebook representa la identidad de la misma, registra todas las actividades de la página y abre una puerta a varias ventajas en la relación con el cliente. Estos perfiles están diseñados para toda organización y sobre todo pequeñas y medianas empresas (PYMES), tener acceso a estadísticas de la página, crear anuncios y promocionar publicaciones; y junto con “Facebook for Developers” la cual esta explicado con mejor detalle en la siguiente sección, un plus adicional a la página y alcanzar objetivos como es el caso con la Librería, al implementar el agente conversacional (FacebookHelp, 2020).

Facebook for Developers

Facebook para desarrolladores es una plataforma que permite a los programadores crear aplicaciones para la plataforma de Facebook, o que necesiten integrar su trabajo en la misma. Para esto es indispensable disponer de una cuenta válida de Facebook, con la finalidad de tener acceso a la activación de su correspondiente cuenta en (FBFD- Facebook for Developers) la cual proporciona las API y herramientas necesarias para el desarrollador además de su correspondiente documentación (Developers, 2018).

Backend de mensajería del agente conversacional

En esta sección se describe todo lo relacionado al bloque “Backend de Mensajería” de la Figura 16, para lo cual en los siguientes apartados se detalla: el uso de variables de entorno en la aplicación, manejo de la API de mensajería de Facebook Messenger, además de la configuración del webhook de la plataforma.

Variables de entorno

Las aplicaciones en su script de código fuente utilizan valores de configuración tales como: claves de acceso, configuraciones de puertos de comunicación, URLs, entre otras. En una aplicación en desarrollo estos valores de configuración pueden llegar a ser diferentes a los valores utilizados en la etapa de producción, es entonces que las variables del código fuente que utilizan esos valores de configuración deben editarse (Mora, 2001). Para evitar la edición del código fuente, se utilizan el concepto de variables de entorno, que consisten en definir los valores de configuración en el objeto “*process*” que toda app de NodeJS posee, este objeto muestra la información del proceso del script que está en ejecución. Para leer las variables de entorno desde un archivo de extensión ENV se utiliza la librería “*dotenv*” del repositorio de NodeJS, la cual carga las variables desde el archivo (.env) en el objeto *process* al momento de iniciar la ejecución de la aplicación en NodeJS. La estructura de las variables de entorno se muestra a continuación en la Figura 22, cabe mencionar que no deben existir espacios en blanco entre el indicador “=”. Los valores que debe tener cada variable son detallados en las secciones posteriores de este documento (Developers, 2018).

Es de importancia mencionar que los valores de las variables de entorno deben ser compartidos únicamente con los desarrolladores de confianza (Developers, 2018).

Figura 22*Variables de entorno de la aplicación*

```

# Configuración de entorno

#Identificador de la app
PAGE_ID=

#Identificador de la app
APP_ID=

PAGE_ACCESS_TOKEN=

#Clave secreta de la app: Configuración/Básica
APP_SECRET=

#Cualquier cadena de texto
VERIFY_TOKEN=

#URL de la aplicación
APP_URL=https://

#URL de la página de la empresa
SHOP_URL=https://

#Puerto elegido para la comunicación
PORT=

```

SendAPI de Messenger

Facebook para desarrolladores ofrece diversas Interfaces de programación de aplicaciones (API - Application Programming Interface), entre las cuales se tiene a la *SendAPI* de Messenger. Esta es una API pública desarrollada por la red social Facebook, esta es la principal API que se utiliza al momento de enviar mensajes a los usuarios de la plataforma, estos mensajes pueden incluir texto, archivos adjuntos, plantillas de mensajes estructurados, acciones del remitente, entre otras. (Facebook, 2018). El punto final “*endpoint*” de la API se encuentra alojado en el URI: https://graph.facebook.com/v8.0/me/messages?access_token=. Y requiere la autenticación mediante un *ACCES_TOKEN* que es proporcionado desde el panel de aplicación en FBFD. Las aplicaciones en modo de desarrollo pueden enviar mensajes a únicamente personas que tienen un rol en la aplicación, estos pueden ser desarrolladores, administradores o evaluadores. Para lanzar la app a producción es necesario solicitar a FB el permiso de mensajería de páginas *pages_messaging*, para que cualquier usuario pueda interactuar con dicho *endpoint*

(Facebook, 2018). Una solicitud de *sendAPI* exitosa devuelve una cadena JSON que contiene identificadores para el mensaje y su destinatario. Está cadena se muestra a continuación en la Figura 23.

Figura 23

Objeto JSON enviado desde la API de Messenger

```
"messaging_type": "<MESSAGING_TYPE>",
"recipient": {
  "id": "<PSID>"
},
"message": {
  "text": "hello, world!"
}
```

Nota. Tomado de (Facebook, 2018).

Para acceder a los parámetros del objeto JSON que envía la API de Messenger, se debe manipular las propiedades presentadas a continuación en la Tabla 10. De las cuales, las más importantes son: el tipo de mensaje *messaging_type*, además de los objetos *recipient* y *message*, estas propiedades que se detallan en los próximos apartados.

Tabla 10

Propiedades del objeto JSON de mensajería.

Propiedad	Tipo	Descripción
<i>messaging_type</i>	String	El tipo de mensajería que se envía en el mensaje <i>RESPONSE</i> <i>MESSAGE_TAG</i> <i>UPDATE</i>
<i>recipient</i>	Object	Objeto <i>recipient</i>
<i>message</i>	Object	Objeto <i>message</i> no se puede enviar con <i>sender_action</i> . Estado del mensaje para el usuario: <i>typing_on</i> : muestra la burbuja de escritura. <i>typing_off</i> : oculta la burbuja de escritura.
<i>sender_action</i>	String	<i>mark_seen</i> : muestra el icono de confirmación. No se puede enviar con <i>message</i> . Debe enviarse como solicitud por separado. Cuando se usa <i>sender_action</i> , <i>recipient</i> debe ser la única propiedad establecida en la solicitud. Opcional. Tipo de notificación push:
<i>notification_type</i>	String	<i>REGULAR</i> : sonido/vibración <i>SILENT_PUSH</i> : notificación en pantalla <i>NO_PUSH</i> : sin notificaciones Por defecto se tiene <i>REGULAR</i> .

Nota. Tomado de (Facebook, 2018).

El objeto “*Recipient*” contiene la información del usuario que envía el mensaje, todas las solicitudes deben incluir ciertas propiedades que identifican al destinatario o remitente. Estas propiedades se muestran a continuación en la Tabla 11.

Tabla 11

Parámetros del objeto Recipient.

Propiedad	Tipo	Descripción
recipient.id	String	Page Scoped Used ID (PSID) del destinatario del mensaje. El usuario debe haber interactuado con cualquiera de los puntos de entrada de Messenger para poder participar en la mensajería con la página.
recipient.user_ref	String	Usado para el checkbox plugin y customer chat plugin .
recipient.post_id	String	Usado para Private Replies para hacer referencia a la publicación del visitante a la que responder.
recipient.comment_id	String	Usado para Private Replies para hacer referencia al comentario de la publicación para responder.

Nota. Tomado de (Facebook, 2018).

El contenido del mensaje de entrada, se encuentra albergado en el objeto “*Message*”. El acceso al contenido de este objeto se realiza mediante el uso de las propiedades presentadas en la Tabla 12. A manera de ejemplo se muestra el acceso al contenido de un mensaje de texto, el cual debe contener la sintaxis: “*message.text*”; mientras que para acceder a un mensaje con archivo adjunto se debe usar “*message.attachment*”.

Tabla 12

Componentes del objeto Message

Propiedad	Tipo	Descripción
text	String	Mensaje de texto, Debe ser UTF-8 y tiene un límite de 2000 caracteres. En este campo no se mostrarán las vistas previas de URL.
attachment	Object	Objeto attachment . Se utiliza para enviar mensajes con medios o mensajes estructurados. Vista previa de la URL.
quick_replies	Array< quick_reply >	Opcional. Arreglo de quick_reply para ser enviados con mensajes
metadata	String	Opcional. Cadena personalizada que se entrega como <u>message echo</u> . Límite de 1000 caracteres.

Nota. Tomado de (Facebook, 2018).

Para manejar archivos adjuntos se debe manipular las propiedades del objeto message attachment que se muestran a continuación en la Tabla 13.

Tabla 13

Propiedades de mensaje de archivo adjunto

Propiedad	Tipo	Descripción
<i>type</i>	String	Tipo de archivo adjunto, puede ser: <i>image</i> , <i>audio</i> , <i>video</i> , <i>file</i> o <i>template</i> . Con tamaño máximo de 25MB.
<i>payload</i>	Object	Carga útil de un archivo adjunto, puede ser <i>Template Payload</i> o un <i>File Attachment Payload</i>

Nota. Tomado de (Facebook, 2018).

Al conocer todas las propiedades que ofrece la *sendAPI*, es posible enviar correctamente mensajes hacia un chat de Messenger. Para tales fines se utiliza la librería “*request*” la cual permite realizar solicitudes HTTP de manera simplificada. La Figura 24 muestra el código que realiza la solicitud a la *sendAPI*, donde “*requestBody*” contiene la estructura del objeto JSON mostrado en la Figura 23.

Figura 24

Código de envío de mensaje hacia la API Messenger

```
request(
  {
    uri: `https://graph.facebook.com/v8.0/me/messages`,
    qs: {
      access_token: 'Aquí va el access token'
    },
    method: "POST",
    json: requestBody
  },
  error => {
    if (error) {
      console.error("No se pudo enviar el mensaje: ", error);
    }
  }
);
```

Nota. Tomado de (Facebook, 2018).

Webhook de Messenger

Los “webhooks” son llamadas automatizadas que se ejecutan cuando ocurre un evento específico desde una página suscrita o desde la aplicación en desarrollo. En otras palabras, se puede definir como una solicitud POST que se envía a una URL de la página suscrita, misma que tiene una configuración que le permite recibir el cuerpo de la solicitud POST para así poder procesarla (Biehl, 2017).

El webhook de Messenger maneja los eventos de envío y recepción de mensajes. (Facebook, 2018) en su documentación recomienda tener instalado NodeJS, además alojar el código fuente en un servidor HTTPS público, para así poder recibir los eventos desde la plataforma de Messenger. Este servidor debe cumplir con ciertas condiciones, tales como: ser compatible con HTTPS, un certificado SSL válido, y un puerto abierto que acepte solicitudes GET y POST (Facebook, 2018).

Para crear el servidor HTTPS, se utiliza el framework web “express”. El código de la Figura 25 muestra la creación del servidor HTTP, el cual recibe las solicitudes en el puerto especificado, para el caso de la app el puerto seleccionado es el 3000.

Figura 25

Creación del servidor express HTTP.

```
//*****Importa dependencias y configura el servidor http*****
const
  express = require('express'),
  bodyParser = require('body-parser'),
  app = express().use(bodyParser.json()); // Crea el servidor express http

// Establece el puerto del servidor y registra el mensaje en caso de éxito
app.listen(process.env.PORT || 3000, () => console.log('webhook is listening'));
```

Nota. Tomado de (Facebook, 2018).

El código necesario para que la plataforma de Messenger verifique el webhook se muestra a continuación en la Figura 26. Esta verificación es de tipo obligatoria, pues garantiza que el webhook es auténtico y funcional.

En primer lugar es necesario elegir una clave, la cuál debe ser registrada en el archivo (.env) en la variable VERIFY_TOKEN, además debe ser ingresada durante el registro del webhook en FBFD. Una vez elegida la clave, el siguiente punto es el proceso de suscripción con la plataforma, para esto la *sendAPI* envía una solicitud GET al “*endpoint*” de la aplicación, la cual incluye un token en el parámetro “*hub.verify*” de la cadena de consulta. Se debe verificar la coincidencia del token enviado con el VERIFY_TOKEN de las variables de entorno, de ser iguales es menester responder con el parámetro “*hub.challenge*” a la solicitud (Facebook, 2018).

Figura 26

Código fuente para la verificación de webhook.

```
// *****Agrega el soporte para solicitudes GET a nuestro webhook*****
app.get("/webhook", (req, res) => {
  // Analizar gramaticalmente los parámetros de consulta
  let mode = req.query["hub.mode"];
  let token = req.query["hub.verify_token"];
  let challenge = req.query["hub.challenge"];

  // Comprueba si token y modo está en la cadena de consulta de la solicitud
  if (mode && token) {
    // Comprueba que el modo y el token enviado son correctos
    if (mode === "subscribe" && token === config.verifyToken) {
      console.log("WEBHOOK_VERIFIED");// Responde con el token de desafío de la solicitud
      res.status(200).send(challenge);
    } else {
      res.sendStatus(403);// Responde con '403 Forbidden' si los tokens de verificación no coinciden
    }
  }
});
```

Nota. Tomado de (Facebook, 2018).

En la Figura 27, se muestra el código fuente para generar el “*endpoint*” del webhook en la app, este admite solicitudes de tipo POST, y recibe todos los eventos enviados desde la *sendAPI* de Messenger. Al recibir correctamente un evento y analizar el mensaje de llegada, el servidor

responde a la API con el código 200, el cual indica un status correcto ante la petición, de esta forma la plataforma no debe volver a enviar el evento hacia el “*endpoint*” (Facebook, 2018). La configuración para la suscripción del webhook desde FBFD es presentada más adelante en la sección 4.5.6.

Figura 27

Código fuente de la asignación del extremo del webhook.

```
//*****Crea el endpoint de nuestro webhook*****
app.post('/webhook', (req, res) => {
  let body = req.body;
  // Comprueba que este es un evento de una suscripción a una página.
  if (body.object === 'page') {
    // Itera cada entrada del evento, puede haber varias si se agrupan
    body.entry.forEach(function(entry) {
      // Obtener mensaje, entry.messaging es una matriz,
      // el mensaje está en el índice 0
      let webhook_event = entry.messaging[0];
      console.log(webhook_event);
    });
    // Devuelve una respuesta '200 OK' a todas las solicitudes
    res.status(200).send('EVENT_RECEIVED');
  } else {
    // Devuelve un '404 Not Found' si el evento no es de una suscripción de página
    res.sendStatus(404);
  }
});
```

Nota. Tomado de (Facebook, 2018).

Integración del agente conversacional en Facebook

Este apartado describe la forma en la que se integra la aplicación de agente conversacional a la mensajería de una página de Facebook, desde la creación de una cuenta en FBFD, hasta la comunicación bidireccional Messenger ↔ agente conversacional mediante el uso de webhooks.

Crear una cuenta en Facebook for Developers

Para activar un usuario de Facebook como desarrollador, se debe ingresar a la página FBFD mediante el siguiente enlace: https://developers.facebook.com/?no_redirect=1. Una vez

dentro se mostrará una ventana como es visible en la Figura 28. Seleccionar Empezar para iniciar el proceso de registro de cuenta.

Figura 28

Plataforma Facebook for Developers.



Nota. Tomado de (Facebook, 2018).

Iniciar la sesión en su cuenta de Facebook, para esto se debe ingresar su correo electrónico o su número de teléfono en caso de tenerlo registrado, junto a su correspondiente contraseña. Al ingresar se redirecciona a una nueva ventana con los pasos a completar para el correcto registro del usuario desarrollador, lo cual se puede observar en la Figura 29.

Figura 29

Formulario inicio de Sesión y de registro de usuario.

Nota. Tomado de (Facebook, 2018).

La Figura 30, muestra el formulario de verificación de la cuenta, para esto se debe ingresar el país en el cual se encuentra el usuario, junto a un número de celular. Después seleccionar uno de los dos métodos para recibir el código de verificación: Mediante mensaje de texto o llamada.

Una vez recibido el código, ingresarlo en la casilla Código de Confirmación y presionar Verificar. Al completar el proceso de verificación pasará al paso de Información sobre ti, en el cual se debe seleccionar la mejor descripción del usuario de FBFD, para este caso se seleccionó la opción “Estudiante”.

Figura 30

Verificación de Cuenta e Información sobre el usuario.

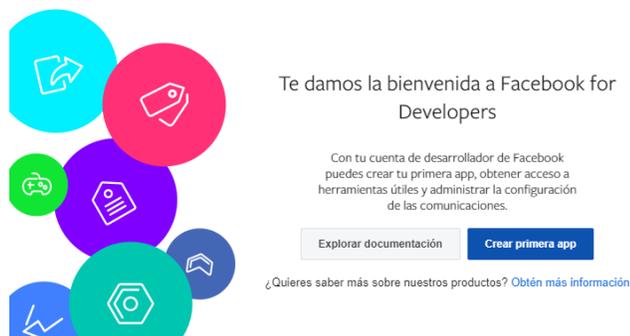
The image displays two sequential steps of the Facebook for Developers (FBFD) registration process. The first step, titled 'Verifica tu cuenta', requires the user to provide their phone number and a confirmation code. The second step, titled '¿Cuál de las siguientes opciones te describe mejor?', allows the user to select a role that best describes them, with 'Estudiante' (Student) being the selected option.

Nota. Tomado de (Facebook, 2018).

Al completar la creación de la cuenta, se mostrará la ventana de bienvenida a FBFD que se puede apreciar en la Figura 31.

Figura 31.

Ventana de Bienvenida a FBFD



Nota. Tomado de (Facebook, 2018).

Crear aplicación en Facebook for Developers

Para crear la primera app en FBFD, se puede presionar Crear primera app en la ventana de bienvenida a FBFD de la Figura 31. O en su defecto, seleccionar la opción Mis Apps en la barra de menú de la página de inicio de FBFD como se puede ver en la Figura 32.

Figura 32

Crear app en FBFD.



Nota. Tomado de (Facebook, 2018).

Como primer punto se debe crear un identificador de la app, el cual es un código único e irreplicable para la aplicación. Seleccionar la opción más adecuada para se va a destinar la aplicación. Para el caso del agente conversacional del presente trabajo se seleccionó la opción: Administrar Integraciones Comerciales, la cual permite administrar cuentas de Messenger. Al dar click en la opción elegida, FBFD redirecciona automáticamente a un nuevo formulario, en el cual se debe ingresar el nombre de la app junto a un correo electrónico de contacto.

Figura 33

Crear identificador de App.

 A screenshot of the 'Crear un nuevo identificador de la app' form. The form is titled 'Crear un nuevo identificador de la app' and has a subtitle 'Empieza a integrar Facebook en tu app o sitio web'. There are three options for app integration:

- Administrar integraciones comerciales** (selected): Crea o administra páginas, grupos, eventos, anuncios, cuentas de Messenger o Instagram u otros tipos de integraciones comerciales.
- Integrar una app de videojuegos de terceros**: Crea una app para que las personas puedan jugar a juegos en el celular fuera de la plataforma de Facebook. Nota: Para crear juegos instantáneos o de página principal, selecciona la opción **Para todo lo demás**.
- Para todo lo demás**: Selecciona esta opción si integrarás el inicio de sesión con Facebook o crearás un juego instantáneo o una app que accederá a los datos de los usuarios en Facebook o Instagram.

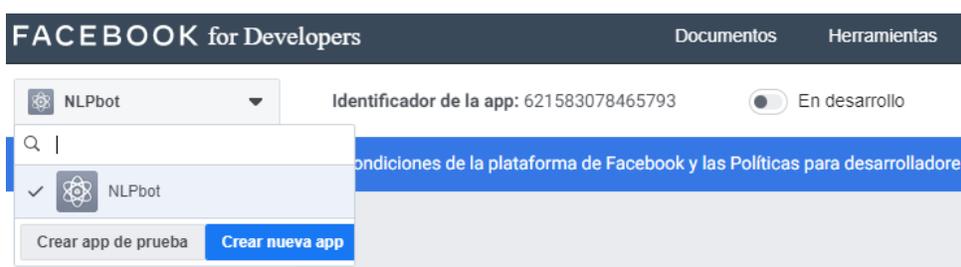
 The 'Nombre para mostrar' field contains 'NLPbot' and the 'Correo electrónico de contacto' field contains 'veronicabelenruiz.23@gmail.com'. At the bottom, there is a 'Cancelar' button and a 'Crear identificador de la app' button. A note at the bottom states: 'Esta dirección de correo electrónico se usa para ponernos en contacto contigo sobre posibles infracciones de la política, restricciones de la app o pasos para recuperarla si se eliminó o está en riesgo.'

Nota. Tomado de (Facebook, 2018).

El proceso finaliza al presionar el botón Crear identificador de la app. La Figura 33, muestra los formularios para la creación del identificador de la nueva aplicación. El identificador de la app creado se puede observar en la parte superior de la ventana de inicio de la aplicación agente conversacional, como se muestra en la Figura 34.

Figura 34

Identificador de la app.



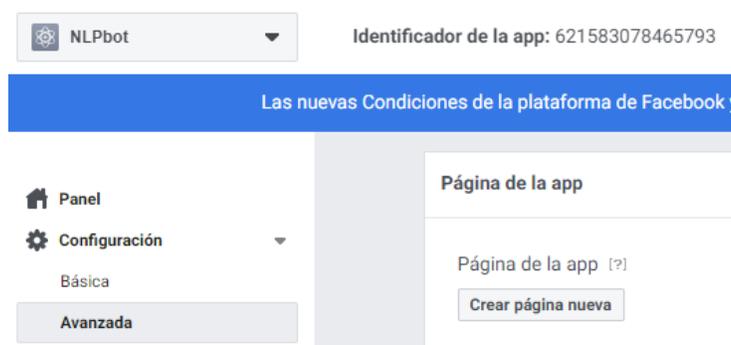
Nota. Tomado de (Facebook, 2018).

Crear página de Facebook

Por motivos de pruebas durante el desarrollo de la app, se crea una nueva página de FB siguiendo la siguiente ruta desde FBFD: NLPbot / Configuración / Avanzada, y click en Crear página nueva, lo cual se puede apreciar en la Figura 35.

Figura 35

Ruta para crear nueva página de FB.



Nota. Tomado de (Facebook, 2018).

En la nueva ventana generada se debe seleccionar Empezar como lo muestra la Figura 36. Después se procede a registrar un nombre y una categoría para la marca. Al pulsar el botón Continuar la nueva página ha sido creada. Se pueden omitir los pasos de añadir fotos e información adicional.

Figura 36

Creación de una nueva página de FB.

Negocio o marca
Conéctate con tus clientes, incrementa tu público y muestra tus productos con una página del negocio gratuita.

Nombre de la página
Bot Alleen

Categoría
Modelo

Negocio o marca
Muestra tus productos y servicios, destaca tu marca y llega a más clientes en Facebook.

Al crear una página en Facebook, se aplican las Políticas de páginas, grupos y eventos.

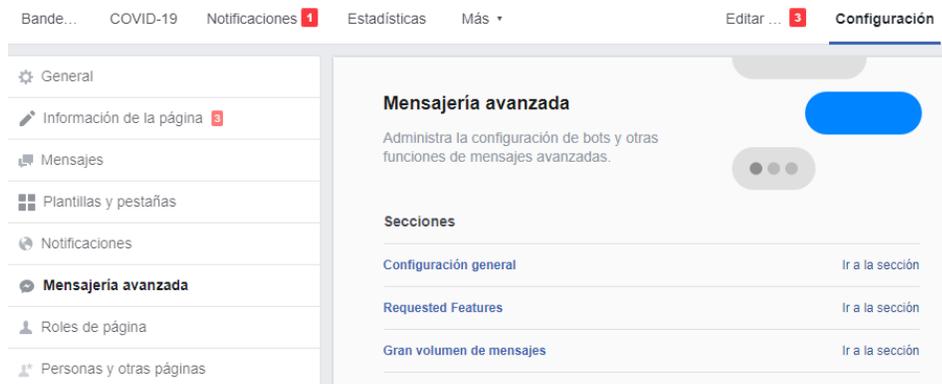
Empezar Continuar

Nota. Tomado de (Facebook, 2018).

La Figura 37 muestra la ruta: Configuración / Mensajería avanzada de la nueva página creada para FB, en la cual se aprecia en que aún no hay aplicaciones conectadas a la página.

Figura 37

Mensajería avanzada sin apps sincronizadas.



Nota. Tomado de (Facebook, 2018).

Agregar productos a la aplicación

Debido a que el agente conversacional necesita integrarse a Messenger de FB es necesario autorizar a Messenger el acceso a la app. Para tales motivos se debe seguir la siguiente ruta: Productos / Messenger y presionar el botón Configurar para finalizar, tal como se muestra en la Figura 38.

Figura 38

Agregar productos a la app.



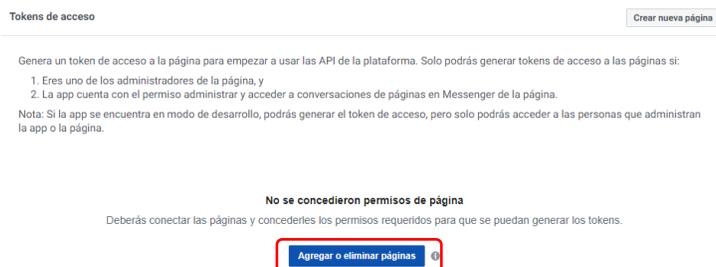
Nota. Tomado de (Facebook, 2018).

Agregar páginas a la aplicación

Aparece la configuración de Messenger de la aplicación, en Tokens de acceso -> Click en Agregar o Eliminar páginas:

Figura 39

Agregar páginas a la aplicación.



Nota. Tomado de (Facebook, 2018).

Marcar la casilla de la página que se ha creado para las pruebas durante el desarrollo, tal como se muestra en la Figura 40.

Figura 40

Agregar página de FB a la aplicación.

¿Qué páginas quieres usar con NLPbot?

En el paso siguiente determinarás lo que NLPbot puede hacer con las páginas que seleccionaste.

Todas las páginas (4) Seleccionar todo

B BOT Alleen	<input checked="" type="checkbox"/>
---------------------	-------------------------------------

[servicio de ayuda](#)

Nota. Tomado de (Facebook, 2018).

Para finalizar se debe permitir la opción “Administrar y acceder a conversaciones de páginas en Messenger”, pulsar siguiente, y presionar Aceptar en la ventana emergente, esto se puede observar en la Figura 41.

Figura 41

Autorizar acceso a Messenger.

¿Qué puede hacer NLPbot?

C Es posible que NLPbot no funcione correctamente si desactivas estas opciones.

Administrar y acceder a conversaciones de páginas en Messenger SI

Vinculaste NLPbot a Facebook

Puedes actualizar lo que NLPbot puede hacer en la [configuración de integraciones comerciales](#). Es posible que NLPbot debata pasos adicionales para finalizar la configuración.

Nota. Tomado de (Facebook, 2018).

Suscripción del Webhook

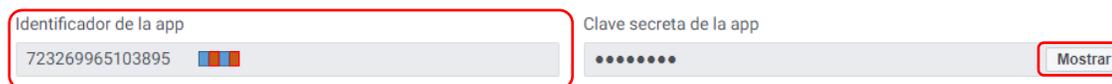
En esta sección se describe la configuración del Webhook tanto del lado del servidor de la app, como del lado de FBFD, para esto es necesario recoger una serie de identificadores y tokens desde la app creada en FBFD.

Tokens e identificadores.

Los primeros parámetros a ser recolectados son los correspondientes a la app creada, entre los cuales se tiene a: el identificador de la app (APP_ID) y la clave secreta generada por FBFD para la app (APP_SECRET), mismos que pueden apreciarse en la Figura 42. Estos datos se encuentran en la siguiente ruta: Configuración / Básica. Para conocer la APP_SECRET se debe presionar el botón Mostrar, el cual solicita el ingreso de la contraseña del usuario de FBFD antes de mostrar la APP_SECRET, misma que tiene la siguiente forma: 95e1d61d0c14f7e4d3172e83a67****.

Figura 42

Identificador y Clave secreta de la app.



The image shows a screenshot of the Facebook developer console. On the left, there is a field labeled 'Identificador de la app' with the value '723269965103895' and a small Facebook logo icon. On the right, there is a field labeled 'Clave secreta de la app' which is masked with dots. A red box highlights the 'Mostrar' button next to the App Secret field.

Nota. Tomado de (Facebook, 2018).

Como segundo punto se recolectan los parámetros correspondientes a la página donde se integrará la app, estos corresponden a Identificador de página (PAGE_ID) y su Token de acceso (PAGE_ACCESS_TOKEN), mismos que pueden ser encontrados en la ruta: Productos / Messenger / Configuración / Tokens de Acceso, tal como lo muestra la Figura 43. De esta sección se puede observar directamente el PAGE_ID el cual tiene la siguiente forma: 11655434346****, este código es único e irreplicable para identificar la página de Facebook creada.

Figura 43

Identificador y token de la página.

Páginas ↑	Tokens
 BOT Aileen 116554343460941	— Generar token

Nota. Tomado de (Facebook, 2018).

Para conocer el PAGE_ACCESS_TOKEN se debe presionar el botón Generar Token mostrado en la Figura 44. Este tiene un aproximado de 190 caracteres alfanuméricos aleatorios. Por seguridad este token debe ser compartido únicamente con desarrolladores de confianza.

Figura 44

Page Token Access generado.

Token generado
×



BOT Aileen
116554343460941

Para mantener tu seguridad, comparte este token SOLO con desarrolladores de apps en los que confías.

Este token solo se mostrará una vez, por lo que deberás guardarlo en un lugar seguro. Si lo pierdes, tendrás que crear otro, ya que cualquiera podría usarlo para hacerse pasar por esta página (según cuál sea la configuración de privacidad de tu app). Si quieres revocar todos los tokens de una página generados anteriormente, elimina la página de la app con el botón que aparece bajo la tabla.

Acepto

EAAI1U5aXIQEBAHciR9peU2GY8lw0ALmnPZAjXlssiXOTF0L9dHN416NIQFMhv...

Copiar

Listo

Nota. Tomado de (Facebook, 2018).

Configuración del webhook en la aplicación.

Para generar una dirección externa conectada al localhost, se ejecuta el aplicativo *ngrok*, se abrirá su terminal en el cual se debe ingresar el comando *ngrok http (# de Puerto)*. Para el

agente conversacional se utilizará el puerto 3000, por lo que la línea de comando a usar es la siguiente: `ngrok http 3000`.

Como se puede observar en la Figura 45, el `ngrok` ha generado la dirección externa enlazada al localhost a través del puerto 3000. De aquí se toma el URL `https`, el cuál será usado como valor para las variables: `APP_URL` y `SHOP_URL`. Cuando la etapa de desarrollo de la app termine, los valores de estas URLs deben ser cambiados por direcciones reales, las cuales serán obtenidas desde Heroku una vez que la app sea subida a este.

Figura 45

Ejecución de Ngrok.

```

ngrok by @inconshreveable (Ctrl+C to quit)
Session Status      online
Session Expires    7 hours, 15 minutes
Version             2.3.35
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding          http://f19b1c93c7ce.ngrok.io -> http://localhost:3000
Forwarding          https://f19b1c93c7ce.ngrok.io -> http://localhost:3000
Connections
  ttl  opn  rt1  rt5  p50  p90
   0    0    0.00 0.00 0.00 0.00

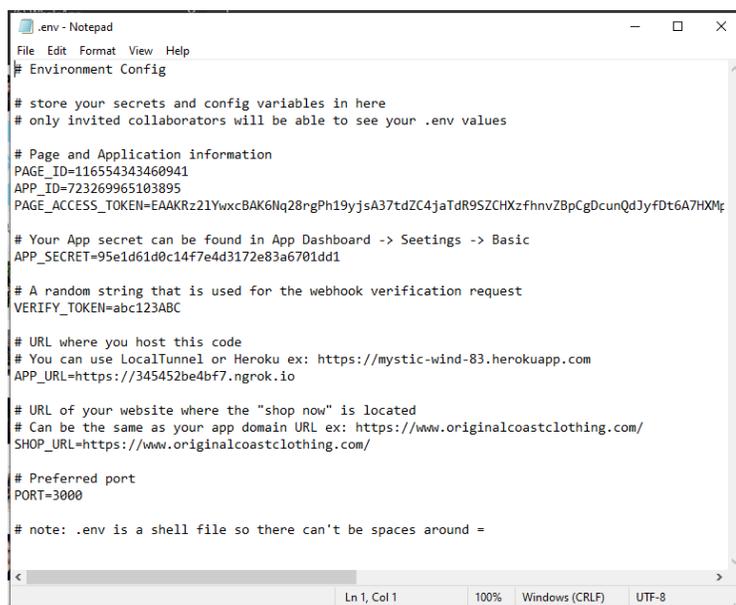
```

Para finalizar la recolección de identificadores y tokens, se debe crear un token de verificación (`VERIFY_TOKEN`). Este es una cadena de caracteres elegida por el usuario. Ejemplo: `abc123ABC`, este token deberá ser ingresado en FBFD para la confirmación del webhook.

Se procede a crear un archivo de variables de entorno (`.env`) dentro del directorio del agente conversacional, el cual debe contener todos los valores recopilados en las secciones 4.4.3, URLs de ngrok, y el token de verificación elegido. El contenido del archivo `.env` se muestra a continuación en la Figura 46, cabe recordar que no deben existir espacios en blanco entre los signos `"="`.

Figura 46

Archivo de variables de entorno.



```

.env - Notepad
File Edit Format View Help
# Environment Config

# store your secrets and config variables in here
# only invited collaborators will be able to see your .env values

# Page and Application information
PAGE_ID=116554343460941
APP_ID=723269965103895
PAGE_ACCESS_TOKEN=EAAKRz21YwxcBAK6Nq28rgPh19yjsA37tdZC4JaTdR95ZCHXzfhnvZBpCgDcunQdJyft6A7HXMf

# Your App secret can be found in App Dashboard -> Seetings -> Basic
APP_SECRET=95e1d61d0c14f7e4d3172e83a6701dd1

# A random string that is used for the webhook verification request
VERIFY_TOKEN=abc123ABC

# URL where you host this code
# You can use LocalTunnel or Heroku ex: https://mystic-wind-83.herokuapp.com
APP_URL=https://345452be4bf7.ngrok.io

# URL of your website where the "shop now" is located
# Can be the same as your app domain URL ex: https://www.originalcoastclothing.com/
SHOP_URL=https://www.originalcoastclothing.com/

# Preferred port
PORT=3000

# note: .env is a shell file so there can't be spaces around =

```

Configuración del webhook en la FBFD.

Para configurar el webhook en el FBFD, ingresar a la app y seguir la ruta: Productos / Messenger / Configuración / Webhooks y seleccionar Agregar “URL de devolución de llamada” como se puede observar en la Figura 47.

Figura 47

Agregar URL de devolución de llamada.



Nota. Tomado de (Facebook, 2018).

El formulario que se debe llenar se puede observar en la Figura 48, donde: “URL de devolución de llamada” corresponde al valor de APP_URL, además de: /webhook al final. Es menester recordar que este URL debe ser cambiado al finalizar la fase de desarrollo. El Token de

verificación concierne al valor de VERIFY_TOKEN, y se finaliza el proceso al seleccionar el botón Verificar y guardar.

Figura 48

URL de devolución de llamada.

Editar URL de devolución de llamada ×

URL de devolución de llamada

Token de verificación

[Más información](#) Cancelar Verificar y guardar

Nota. Tomado de (Facebook, 2018).

Una vez que el URL de devolución de llamada es verificado y guardado, la sección Webhooks debe verse como la mostrada en la Figura 49, donde ya consta la URL, y el webhook de la página refleja cero suscripciones. Para agregar suscripciones se debe presionar el botón Agregar Suscripciones.

Figura 49

Webhook de FBFD configurado.

Webhooks

Para recibir mensajes y otros eventos que envíen los usuarios de Messenger, la app debe tener habilitada la integración de webhooks.

URL de devolución de llamada Token de verificación

Las solicitudes de validación y las notificaciones de webhook de este objeto se enviarán a esta URL. Token que te enviará Facebook como parte de la verificación de la URL de devolución de llamada.

Editar URL de devolución de llamada Mostrar errores recientes

Páginas ↑	Webhooks
<div style="display: flex; align-items: center;"> <div style="background-color: #ccc; border-radius: 50%; width: 20px; height: 20px; display: flex; align-items: center; justify-content: center; margin-right: 5px;">B</div> <div> <p>BOT Aileen</p> <p>116554343460941</p> </div> </div>	<p>0 campos</p> <p style="text-align: right; margin-top: 5px;">Agregar suscripciones</p>

Nota. Tomado de (Facebook, 2018).

Se recomienda seleccionar todas las suscripciones como muestra la Figura 50, pero es totalmente opcional, se pueden seleccionar aquellos que sean más relevantes para la app, Presionar el botón Guardar para almacenar los cambios, y se reflejan los 18 campos seleccionados para poder recibir y enviar mensajes, tal como se observa en la Figura 51.

Figura 50

Suscripciones de webhooks.

Nota. Tomado de (Facebook, 2018).

Figura 51

Campos seleccionados de Webhooks.

Páginas ↑	Webhooks
 BOT Aileen 116554343460941	18 campos messages, messaging_postbacks, messaging_optins, message_deliveries, message_reads, messagin... Editar

Nota. Tomado de (Facebook, 2018).

Verificación del Webhook.

La verificación del webhook se completa, una vez que las configuraciones mostradas en la sección 4.5.6.2 y 4.5.6.3 han sido cumplidas, entonces se inicia el servidor del agente conversacional. Para tales motivos se debe ejecutar el script principal de la aplicación (app.js), esto a través de la terminal de Windows en base al uso de el comando: `node app.js`, tal como lo

muestra la Figura 52. Al ser la primera vez que arranca el servidor, se debe visitar el enlace que la terminal muestra, https://345452BE4BF7.ngrok.io/profile?mode=all&verify_token=abc123ABC, cuando esto sea realizado se muestra el mensaje “WEBHOOK_VERIFIED”, tal como se puede observar en la Figura 52 y el agente conversacional está listo para ser usado.

Figura 52

Verificación de Webhook

```
Your app is listening on port 3000
Is this the first time running?
Make sure to set the both the Messenger profile, persona and webhook by visiting:
https://345452be4bf7.ngrok.io/profile?mode=all&verify_token=abc123ABC
Test your app by messaging:
https://m.me/116554343460941
WEBHOOK_VERIFIED
```

Para verificar que el agente conversacional se conectó correctamente a la página de FB, hay seguir la siguiente ruta: Página / Configuración / Mensajería Avanzada / Apps conectadas. Como se puede observar en la Figura 53, la app ha sido conectada exitosamente, y se puede iniciar las pruebas del agente conversacional.

Figura 53

Apps conectadas a la página.

Apps conectadas
Las siguientes apps están conectadas actualmente a tu página.

	NLPbot 723269965103895 Ver permisos	<input type="button" value="Eliminar"/>
---	---	---

Configuración de apps
Administra las funciones a las que las apps puedes acceder y controlar.

Nota. Tomado de (Facebook, 2018).

Capítulo V

Evaluación de resultados

Este capítulo aborda, la definición de criterios para la evaluación del agente conversacional, la recolección de información durante pruebas del sistema, y un análisis de estos datos para mostrar así la confiabilidad del agente conversacional de librería.

Metodología

Para el desarrollo del proyecto se optó tomar en cuenta la metodología ágil tipo SCRUM por el control, planificación, aclaración y verificación del mismo, para poder obtener resultados cuantitativos explicados posteriormente, los cuales han ido cambiando con el constante entrenamiento del agente conversacional. Ciertamente, se realizaron las entrevistas necesarias con el gerente de la librería y líder del proyecto, para el levantamiento de información, presentación de respectivos avances y cambios que puedan existir en los diferentes "Sprint". A continuación se establece los roles de scrum de la siguiente manera: Dueña del producto o "Product owner" la tienda dedicada a la venta de libros, facilitador o "scrummaster" Ing. Henry Coral encargado de guiar el desarrollo del proyecto, equipo de desarrollo Verónica Ruiz encargado del desarrollo del proyecto y stakeholders los clientes frecuentes y de confianza de la librería para las pruebas con el agente conversacional y encuestas respectivas que están detalladas en el presente capítulo.

Finalmente el periodo de tiempo o "Sprint" que se estableció para el proyecto en jornadas de desarrollo semanal, los martes y jueves como planificación de "Sprint" acordado con el calendario universitario y el facilitador, a continuación por acuerdo de los stakeholders que brindo un periodo de tiempo de 2 semanas : los primeros días para las primeras pruebas con el chatbot, verificar errores, tomar información necesaria para el entrenamiento y cambios a desarrollar en el agente conversacional como el corpus en el transcurso de la misma semana, y a finales de la

segunda semana las últimas pruebas con las actualizaciones respectivas en el agente conversacional, y evaluación con las encuestas para la toma de datos necesarias para las fórmulas respectivas.

Definición de parámetros de evaluación

La definición de parámetros de evaluación, aborda tres puntos principales asociados a ciertas métricas. El primero está enfocado a medir el desempeño teórico del agente conversacional, en base a tres componentes, que son: *exactitud "accuracy"*, *precisión "precision"*, y *sensibilidad "recall"*. Además de la precisión del agente para otorgar respuestas correctas ante las solicitudes de un usuario real. La segunda métrica aborda la satisfacción del usuario, cuya medida es fundamentada con las respuestas de una encuesta realizada al finalizar la experiencia de uso durante el proceso de ejecución de pruebas controladas. Además se incluye el cálculo de la confiabilidad del sistema. A continuación, se detallan las métricas utilizadas para evaluar la confiabilidad del agente conversacional.

Desempeño teórico del agente conversacional

Las métricas de desempeño utilizadas están asociadas al concepto de matriz de confusión, la cual es una herramienta que permite observar los aciertos y errores del modelo de un clasificador de aprendizaje supervisado, además es posible calcular parámetros estadísticos que ofrecen información cuantitativa sobre el desempeño del modelo (Sobrino, 2001).

Para la elaboración de la matriz de confusión es necesario dividir la base de conocimiento supervisado en dos grupos, los cuales son: entrenamiento y prueba. Esta segmentación se realiza definiendo un porcentaje para cada grupo, por lo general suelen tomar los valores de: 80% para el grupo de entrenamiento y 20% para el de prueba. Entonces de manera aleatoria se selecciona el 80% de datos, que servirán para entrenar el modelo, y el 20% restante es usado para realizar la prueba del modelo a cada dato del grupo de prueba (Sobrino, 2001).

Al conocer a que categoría pertenecen los datos de cada grupo, es posible registrar el número de aciertos y errores durante la prueba del modelo. La Tabla 14 muestra el esquema de una matriz de confusión para un clasificador de tres clases. Esta es una matriz cuadrada, cuyas columnas representan las clases de los datos de prueba, mientras que sus filas registran la categoría en la que cada clase de prueba es clasificada durante la prueba. Los elementos m_{ij} representan un valor escalar, la diagonal marcada en color naranja indica el número de aciertos del clasificador, mientras que sus matrices triangular superior e inferior marcadas en color verde contienen los errores registrados para cada clase durante el ensayo del modelo.

Tabla 14

Matriz de confusión de dimensiones 3x3

		Entrenamiento			
		A	B	C	
Prueba	A	m_{11}	m_{12}	m_{13}	$totF_1$
	B	m_{21}	m_{22}	m_{23}	$totF_2$
	C	m_{31}	m_{32}	m_{33}	$totF_3$
		$totC_1$	$totC_2$	$totC_3$	<i>total</i>

Nota. Ejemplo de la matriz de confusión prueba y entrenamiento.

El análisis matemático de los parámetros de evaluación utilizados, está fundamentado en el trabajo de (Jalife, Auza, & Atoany Fierro Radilla, 2017). Como se mencionó en la sección 5.1, estos parámetros son: exactitud, precisión y sensibilidad. Antes de definir las expresiones que rigen a estos parámetros, se puede precisar de ciertos componentes numéricos, los cuales ayudarán al cálculo de los parámetros de evaluación. Dichos componentes se describen a continuación.

El primer componente $totF_i$ se muestra en la ec. 5.1, el cual representa la suma de todos los elementos de la fila i ; de la misma manera $totC_j$ de la ec 5.2 indica la sumatoria de todos los elementos de la columna j , y *total* mostrado en la ec. 5.3 es el número total de observaciones del

ensayo, es decir el número total de datos de prueba analizados; para finalizar, N representa el número de categorías de la base de conocimiento.

$$totF_i = \sum_{j=1}^N m_{ij} \quad (5.1)$$

$$totC_j = \sum_{i=1}^N m_{ij} \quad (5.2)$$

$$total = \sum_{i=1}^N totF_i = \sum_{i=1}^N totC_i \quad (5.3)$$

La exactitud indica una media de los aciertos de cada clase, la expresión que la riges muestra en la ec. 5.4. Donde m_{ii} representa el elemento de la diagonal correspondiente a la clase i analizada, es decir el número de aciertos para dicha clase.

$$exactitud = \frac{\sum_{i=1}^N m_{ii}}{total} \quad (5.4)$$

La precisión individual representa una relación que toma en cuenta de todos los datos clasificados en una categoría i ($totC_i$), y la cantidad de datos que en realidad pertenecían a dicha categoría (m_{ii}). Para determinar el valor de precisión global del modelo se calcula la media de las precisiones individuales de cada clase, esto es mostrado en la ec. 5.5.

$$precisión = \frac{\sum_{i=1}^N \frac{m_{ii}}{totC_i}}{N} \quad (5.5)$$

La sensibilidad individual es una relación que toma en cuenta todos los elementos de prueba que en realidad pertenecen a la categoría i ($totF_j$), y la cantidad de elementos que han sido clasificados correctamente en la clase de entrenamiento i (m_{jj}). La sensibilidad global viene

dada por la media de todas las sensibilidades de cada clase, su expresión viene dada por la ec. 5.6.

$$sensibilidad = \frac{\sum_{j=1}^N \frac{m_{jj}}{totF_j}}{N} \quad (5.6)$$

Una vez determinados los parámetros de evaluación es necesario determinar una función que contenga a cada uno de estos. Para tales fines se aplica el método de la suma ponderada, el cual permite agrupar varias funciones de diversos criterios en una única función de respuesta escalar, donde w_1 , w_2 y w_3 , corresponden a los valores de ponderación y deben cumplir con la condición mostrada en la ec 5.7 para así normalizar la función en un rango de 0 a 1. La función con suma ponderada es mostrada en la ec 5.8.

En un escenario ideal del clasificador se tendría: $exactitud = precisión = sensibilidad = 1$, obteniendo así un desempeño del 100%. Sin embargo, la probabilidad de que esto suceda es prácticamente nula, un ejemplo de esto se muestra más adelante en la sección 5.3 evaluación de resultados.

$$w_1 + w_2 + w_3 = 1 \quad (5.7)$$

$$desempeño = w_1 \times exactitud + w_2 \times precisión + w_3 \times sensibilidad \quad (5.8)$$

Desempeño del agente conversacional con usuarios reales

El desempeño del agente conversacional con usuarios reales considera la precisión a preguntas reales ($prec_PPRR$) del agente, es decir la precisión que el agente tiene al entregar una respuesta correcta ante la petición de un usuario. Este índice viene dado por la relación del total preguntas realizadas por los usuarios de prueba entre el total de respuestas correctas que ha entregado el agente conversacional. Estos valores son determinados por un humano, en base a la inspección minuciosa en cada conversación de los usuarios de prueba, debido a que un humano

puede definir cuales respuestas fueron correctas o incorrectas durante el uso del agente conversacional.

Satisfacción de los usuarios

La métrica de satisfacción toma en cuenta la experiencia de los usuarios de prueba después del uso del agente conversacional. Su evaluación está fundamentada en la Escala de Satisfacción del Cliente (ESC) o CSAT por sus siglas en inglés “Customer Satisfaction Score”. Este método es un indicador de la satisfacción del cliente, y utiliza una encuesta la cual debe ser llenada al finalizar la experiencia con el agente conversacional (Estados Unidos Patente nº US20190347326A1, 2018). El usuario debe calificar asignado un puntaje de 1 a 5 a las preguntas planteadas para medir cada categoría, con la siguiente consideración para la calificación: 1 es mala y 5 excelente. Estas preguntas se muestran a continuación en la Tabla 15.

Tabla 15

Modelo de encuesta de satisfacción de usuario

	Preguntas
Satisfacción	¿Usted siempre recibió una respuesta por parte del agente conversacional?
	¿El agente conversacional le brindo una respuesta coherente a su pregunta ingresada?
	¿Cómo le pareció el trato del agente conversacional con usted?
	¿Recibió usted la respuesta que esperaba a sus preguntas?
	¿Cree que el tiempo de respuesta del agente conversacional fue el prudente o tardo demasiado?
	¿Cree usted que facilito sus dudas en menos tiempo?

La puntuación de cada pregunta se calcula a partir del promedio de las calificaciones recibidas por el usuario. Posterior a eso, dicha calificación debe ser normalizada en un valor de 0 a 1, para así determinar la puntuación al aplicar el método de la suma ponderada. La calificación de la escala de satisfacción del cliente es definida en la expresión de la ec. 5.9.

$$ESC = (w_1 \times P_1) + (w_2 \times P_2) + (w_3 \times P_3) + (w_4 \times P_4) \dots \quad (5.9)$$

Usabilidad del sistema

Para la medición de la usabilidad se usó la escala de usabilidad de sistemas (EUS), fue creada por John Brooke en 1986 y registrada como estándar en la ISO 9241-11, esta contiene una encuesta que incluía 10 ítems que un usuario debe completar después de utilizar una interfaz informática. Se contemplan 5 ítems para evaluar los aspectos positivos del sistema y 5 para los aspectos negativos (Aguilar & Villegas, 2016). Las preguntas usadas en el presente documento se pueden apreciar en la Tabla 16, donde las preguntas impares (1,3,5,7,9) corresponden a los aspectos positivos, mientras que las pares (2,4,6,8,10) representan a los aspectos negativos.

Tabla 16

Modelo de encuesta de satisfacción de usuario

Preguntas	
Usabilidad	1 Creo que usaría este agente conversacional frecuentemente.
	2 Encuentro este agente conversacional innecesariamente complejo.
	3 Creo que el agente conversacional fue fácil de usar.
	4 Creo que necesitaría ayuda de una persona con conocimientos técnicos para usar este agente conversacional.
	5 Las funciones de este agente conversacional están bien integradas.
	6 Creo que el agente conversacional es muy inconsistente.
	7 Imagino que la mayoría de la gente aprendería a usar este agente conversacional en forma muy rápida.
	8 Encuentro que el agente conversacional es muy difícil de usar.
	9 Me siento confiado al usar este agente conversacional.
	10 Necesité aprender muchas cosas antes de ser capaz de usar este agente conversacional.

Nota. Tomado de (Aguilar & Villegas, 2016).

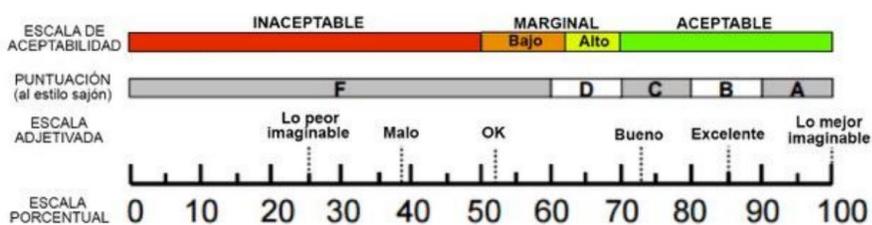
El usuario debe calificar cada pregunta en un rango de 1 a 5, donde 1 es totalmente en desacuerdo, 3 neutral, y 5 totalmente de acuerdo. Se hallan los promedios de las calificaciones de cada pregunta, posterior a eso se encuentra el equivalente positivo de los ítems pares, mientras que al promedio de los ítems impares se les resta el valor de 1, una vez halladas estas equivalencias se escala este valor a un rango de 0 a 100, para esto es necesario multiplicar por una constante de valor 2,5 a la sumatoria de las equivalencias mencionadas anteriormente. Una

expresión para calcular el valor porcentual de la EUS se muestra a continuación en la ec. 5.10; al resultado de esta ecuación también se la conoce como puntuación EUS (Aguilar & Villegas, 2016), y su escala correspondiente se puede visualizar en la Figura 54.

$$EUS = \left(\left(\sum Prom_{impares} - 5 \right) + \left(25 - \sum Prom_{pares} \right) \right) \times 2.5 \quad (5.10)$$

Figura 54

Escala de medidas de la satisfacción. Adjetivos y rangos de aceptabilidad



Nota. Tomado de (GONZALEZ-CAMPOS, BERNABÉ-POVEDA, & PAZMIÑO, 2017)

Confiabilidad del sistema

La confiabilidad es la capacidad de un sistema o componente para desempeñar funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. El criterio para el cálculo de la confiabilidad está incluido en el estándar IEC 60300-3-10 el cuál muestra las normas para la gestión de la confiabilidad. La expresión que rige a la confiabilidad se muestra a continuación en la ec. 5.11, donde: *A* corresponde al tiempo medio entre errores, y *B* el tiempo promedio para solucionar los errores (IEC, 2001).

$$Confiabilidad = \frac{A}{A + B} \quad (5.11)$$

Ejecución de pruebas controladas y recolección de información

Una vez establecidas las métricas de evaluación junto a los parámetros que las componen, es posible realizar pruebas bajo un ambiente controlado. Para esto se seleccionó una población de 21 usuarios, la cual se encuentra en un rango de edad entre 20 a 50 años, la mayoría de ellos

amantes de la lectura. El objetivo de esta población es el interactuar con el agente conversacional de la librería, sin especificarles que este tiene un dominio previamente establecido, y de esta forma poder registrar el comportamiento del BOT ante preguntas establecidas o no establecidas en su corpus. Al ser una aplicación en modo de desarrollo, los usuarios deben ser registrados como “evaluadores” en la configuración de roles de aplicación en Fbfd, para que sus mensajes del chat de Messenger puedan ser enviados a la aplicación. Al terminar la experiencia los usuarios deben completar la encuesta de satisfacción descrita en el apartado 5.1.2, para así realizar un análisis de esos datos en la sección 5.3.

Los primeros datos recolectados se muestran en la Tabla 17, misma que corresponde a la matriz de confusión la cual contiene todas las Intenciones definidas en la sección 3.2. Además de la información de todas las clasificaciones durante la prueba del sistema.

Tabla 17

Matriz de confusión del agente conversacional

		Entrenamiento											Total	
		Saludar	Despedir	Ubicación	Horario	Teléfono	Domicilio	Catalogo	Digital	Audio	Usados	Buscar		Club
Prueba	Saludar	2	0	0	0	0	0	0	0	0	0	0	0	2
	Despedir	0	3	0	0	0	0	0	0	0	0	0	0	3
	SolicitarUbicacion	0	0	3	0	0	0	1	0	0	0	0	0	4
	SolicitarHorario	1	0	0	1	0	0	0	0	0	0	0	0	2
	SolicitarTelefono	1	0	0	0	2	0	0	0	1	0	0	0	4
	SolicitarDomicilio	1	0	0	0	0	3	0	0	0	0	0	0	4
	VerCatalogo	0	0	0	0	0	0	4	0	0	0	0	0	4
	SolicitarFormatoDigital	0	0	0	0	0	0	1	3	0	0	0	0	4
	SolicitarAudioLibro	1	0	0	0	0	0	0	0	3	0	0	0	4
	VerLibrosUsados	0	0	0	0	0	0	0	0	0	3	0	0	3
	BuscarLibro	1	0	0	0	0	0	0	1	0	1	3	0	6
	SolicitarClub	0	0	1	0	0	0	0	0	0	0	0	3	4
		Total	7	3	4	1	2	3	6	4	4	4	3	3

Otro resultado de las pruebas controladas se muestra a continuación en la Tabla 18, en la cual se puede observar el total de preguntas realizadas por la población, el total de preguntas fuera del dominio del agente conversacional, además de las respuestas correctas e incorrectas que este envió a los usuarios.

Tabla 18

Datos de preguntas de usuarios

	Total Preguntas	Preguntas Fuera del dominio	Respuestas correctas	Respuestas incorrectas
Usuario 1	22	3	14	5
Usuario 2	28	5	21	2
Usuario 3	4	0	3	1
Usuario 4	6	0	6	0
Usuario 5	14	0	10	4
Usuario 6	9	0	9	0
Usuario 7	12	0	9	3
Usuario 8	16	0	10	6
Usuario 9	36	3	29	4
Usuario 10	22	4	14	4
Usuario 11	17	0	14	3
Usuario 12	11	0	9	2
Usuario 13	26	3	16	7
Usuario 14	9	0	8	1
Usuario 15	30	3	24	3
Usuario 16	6	0	6	0
Usuario 17	22	1	16	5
Usuario 18	9	0	8	1
Usuario 19	8	0	8	0
Usuario 20	12	2	7	3
Usuario 21	6	0	5	1
Total	325	24	246	55

Los resultados de las encuestas detalladas en la sección 5.1.2 usadas para el cálculo de la métrica de satisfacción se pueden encontrar en el siguiente enlace:
<https://forms.gle/VbamrZivH9QfeWTG6>.

Evaluación de resultados

La evaluación de resultados aborda el análisis de los datos obtenidos para la métrica de desempeño, satisfacción, usabilidad y confiabilidad del sistema.

Análisis del desempeño del sistema

El primer análisis se realiza al desempeño del sistema, al utilizar las expresiones definidas en la sección 5.1.1. De la matriz de confusión presentada en la Tabla 17 se pueden determinar el valor del parámetro *exactitud* el cuál alcanzo un valor de 75%, la *presición* un total de 96.25%, mientras que la *sensibilidad* el 77.08%, el cálculo de estos valores se muestran a continuación en las expresiones ec. 5.12, 5.13 y 5.14 respectivamente.

$$exactitud = \frac{2 + 3 + 3 + 1 + 2 + 3 + 4 + 3 + 3 + 3 + 3 + 3}{44} = \frac{33}{44} = 0.7500 \quad (5.12)$$

$$sensibilidad = \frac{\frac{2}{7} + \frac{3}{3} + \frac{3}{4} + \frac{1}{1} + \frac{2}{2} + \frac{3}{3} + \frac{4}{6} + \frac{3}{4} + \frac{3}{4} + \frac{3}{4} + \frac{3}{3} + \frac{3}{3}}{12} = 0.7708 \quad (5.13)$$

$$presición = \frac{\frac{2}{7} + \frac{3}{3} + \frac{3}{4} + \frac{1}{1} + \frac{2}{2} + \frac{3}{3} + \frac{4}{6} + \frac{3}{4} + \frac{3}{4} + \frac{3}{4} + \frac{3}{3} + \frac{3}{3}}{12} = 0.9625 \quad (5.14)$$

El desempeño teórico del sistema es mostrado en la ec 5.15, la cual muestra un valor aceptable pues este supera el 83%, es decir 8 de cada 10 preguntas de prueba fueron respondidas correctamente.

$$desempeño = 0.4 \times 0.7500 + 0.3 \times 0.9625 + 0.3 \times 0.7708 = 0.8392 = 83.92\% \quad (5.15)$$

Para determinar el índice de precisión en respuestas correctas del sistema, se toman en cuenta los valores presentados en la Tabla 18. De los cuales cabe mencionar que se excluyeron las preguntas fuera de dominio por ser de carácter ambiguo, es decir, no se puede catalogar una respuesta como correcta o incorrecta, si el sistema no tiene la capacidad de determinar lo que el usuario necesita.

Tabla 19

Resultados de la experiencia de usuarios.

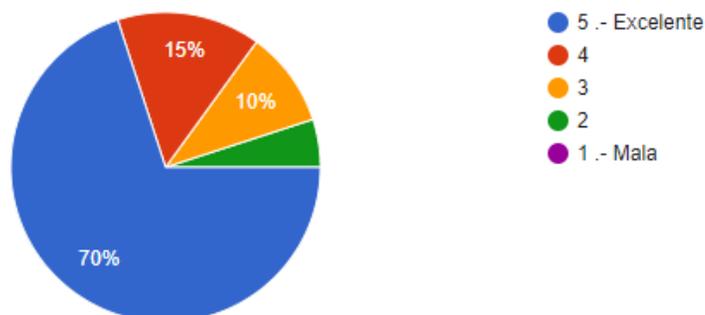
Parámetro	Valor
Total de Preguntas	325
Preguntas fuera de dominio	24
Preguntas Promedio en una conversación	14
Respuestas correctas	81.73%
Respuestas incorrectas	18.27%

En la Tabla 19 se puede observar los resultados del desempeño del agente conversacional con los usuarios reales, además se puede apreciar que el total de respuestas correctas es del 81.73%. Es decir, durante una conversación promedio de 14 preguntas, aproximadamente 11 fueron contestadas correctamente mientras que 3 no responderán correctamente ante la solicitud del usuario de prueba.

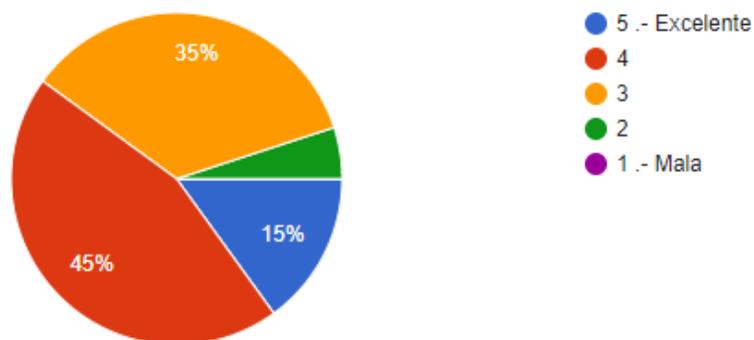
Análisis de la métrica de satisfacción

A continuación, se realiza un análisis de los resultados de cada una de las preguntas de las encuestas mostradas en la sección 5.2, para eso se toma en cuenta las gráficas generadas por la plataforma donde se alberga las encuestas.

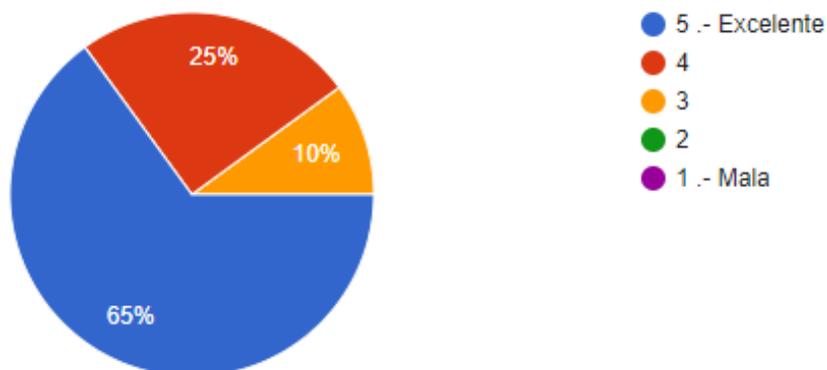
Pregunta 1: Se puede visualizar que 14 personas de 21 calificaron como excelente en recibir una respuesta del agente conversacional, obteniendo como promedio 4.5, normalizado igual a 0.9.

Figura 55*Resultados pregunta 1**Nota.* Tomado de (Google, 2020).

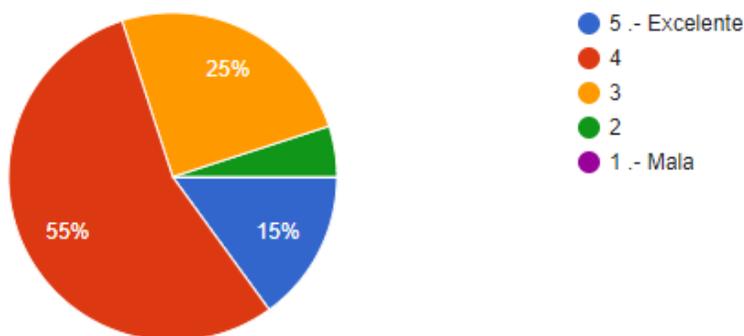
Pregunta 2: Se puede visualizar que 9 usuarios de 21 calificaron con 4 en que la respuesta que recibieron tiene una coherencia, con lo que se obtiene un promedio de 3.7 y normalizado igual a 0.74.

Figura 56*Resultado pregunta 2.**Nota.* Tomado de (Google, 2020).

Pregunta 3: Se puede visualizar que 65% de los encuestados califico como excelente en cuanto al trato que recibió por parte del agente conversacional, con lo que se obtiene un promedio de 4.55 y normalizado igual a 0.91.

Figura 57*Resultados pregunta 3.**Nota.* Tomado de (Google, 2020).

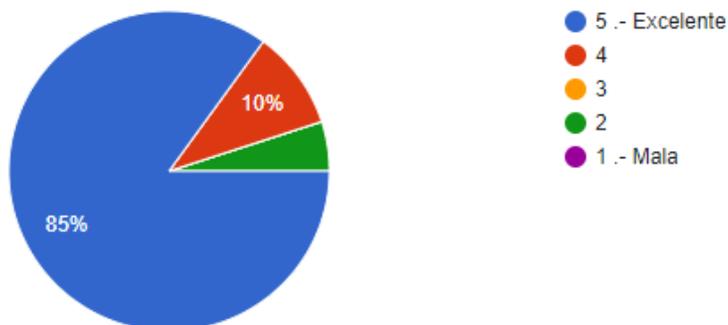
Pregunta 4: Se puede visualizar que el 55% de los encuestados califico con 4 en recibir la respuesta correcta a la pregunta ingresada, con lo que se obtiene un promedio de 3.9 y normalizado igual a 0.78.

Figura 58*Resultados pregunta 4.**Nota.* Tomado de (Google, 2020).

Pregunta 5: Se puede visualizar que 17 usuario de 21, calificador con excelente el tiempo prudente de respuesta del agente conversacional, con lo que se obtiene un promedio de 4.75 y normalizado igual a 0.95.

Figura 59

Resultados pregunta 5.

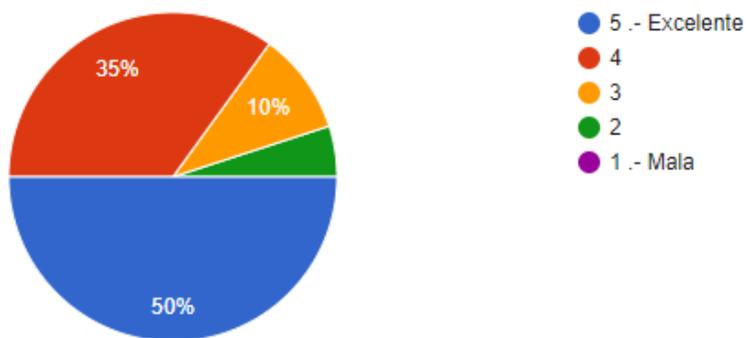


Nota. Tomado de (Google, 2020).

Pregunta 6: Se puede visualizar que el 56.3% de las personas encuestas calificaron como excelente, al poder resolver sus dudas en menor tiempo, sin la necesidad de esperar a un asesor humano, con lo que se obtiene un promedio de 4.3 y normalizado igual a 0.86.

Figura 60

Resultados pregunta 7.



Nota. Tomado de (Google, 2020).

Con estos resultados de las expresiones se procede a calcular la puntuación de la *satisfacción*, tomando en cuenta las ponderaciones: *experiencia del usuario* 40% y *usabilidad* del 60%; esta puntuación se muestra en la ec. 5.16. Los resultados obtenidos son

neutral – positivos pues se puede observar que el 85.69% de los encuestados, una buena satisfacción con el uso del agente conversacional, lo que es bastante favorable para poder seguir con el entrenamiento adecuado para las mejoras del mismo, con el objetivo de lograr una mayor certeza en las respuestas a las preguntas realizadas y menor errores y perdidas del agente conversacional de la librería.

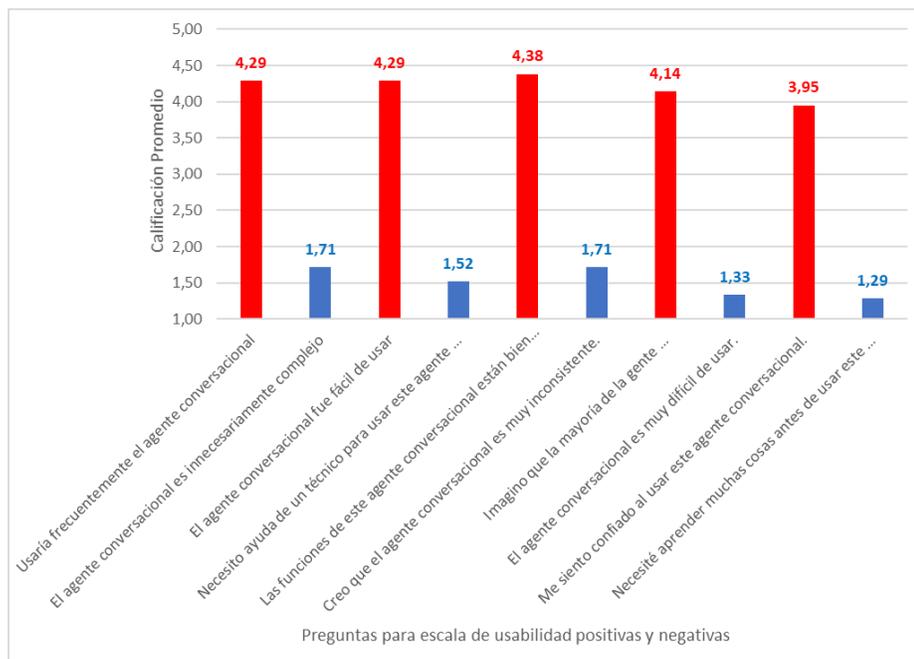
$$\text{satisfacción} = \frac{0.90 + 0.74 + 0.91 + 0.78 + 0.95 + 0.86}{6} * 100\% = 86.68\% \quad (5.16)$$

Análisis de la métrica de usabilidad

Al aplicar el método de escala de usabilidad del sistema, presentado en la sección 5.1.4, en la Figura 61 en la que se puede observar el promedio de calificaciones tanto de los ítems positivos como de los negativos.

Figura 61

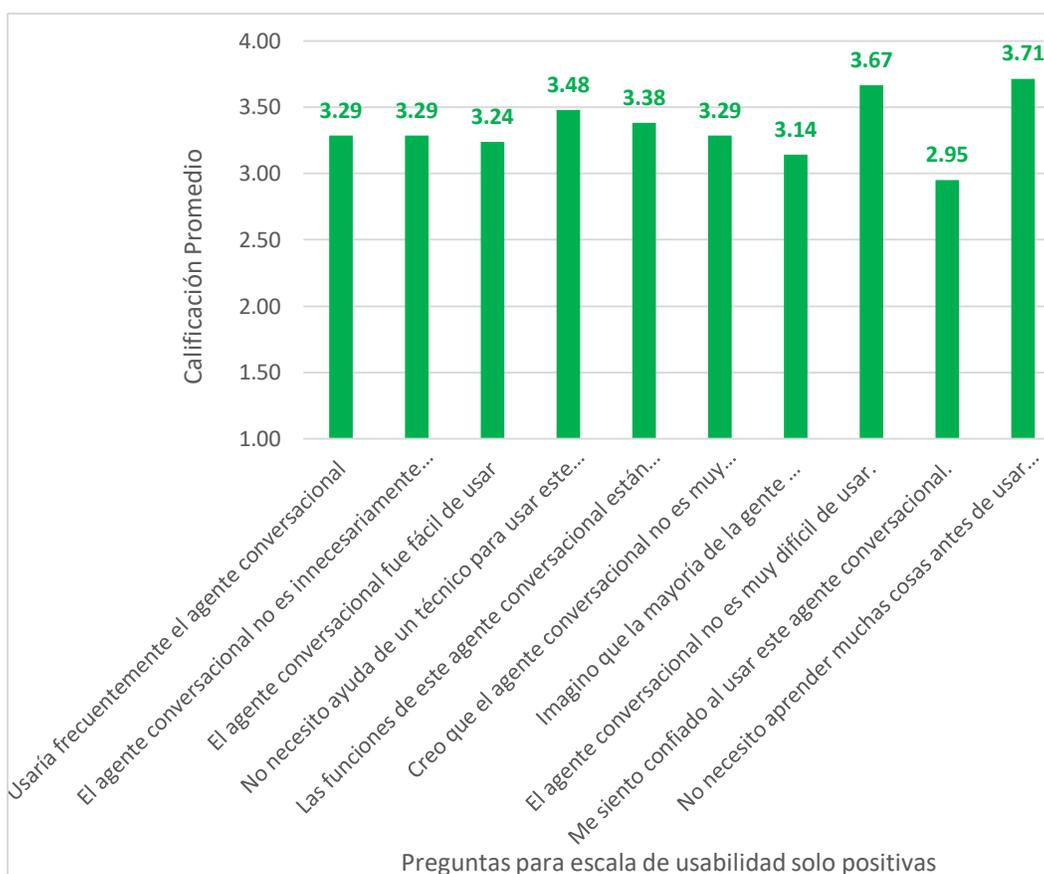
Promedio de calificaciones de preguntas de usabilidad



Al aplicar los cálculos de la escala de usabilidad del sistema, se obtiene en primer lugar, el promedio de calificaciones de los ítems negativos, los cuales son mostrados a continuación la Figura 62.

Figura 62

Promedio de calificaciones de preguntas de usabilidad con ítems negativos revertidos.



Usando los promedios de calificaciones mostrados en la Figura 62, se calcula la puntuación *EUS*, la cual es mostrada a continuación en la ec. 5.17. Según la Figura 54, la puntuación alcanzada corresponde a “aceptable”, de la escala de aceptabilidad, un “B” en la puntuación al estilo sajón, y un “excelente” en la escala adjetivable.

$$EUS = ((21 - 5) + (25 - 7.57)) \times 2.5 = 83.57 \quad (5.17)$$

Confiabilidad del agente conversacional

La confiabilidad del sistema de la ec 5.18 muestra un valor aceptable pues este supera el 98.31%. Asumiendo un uso las 24 horas del año, y con 30 fallas anuales, se puede determinar el tiempo medio entre fallas el cual corresponde a 292 horas, mientras que el tiempo promedio para resolver errores es de 4 horas por la simplicidad del sistema.

$$\text{Confiabilidad} = \frac{292}{292 + 5} * 100\% = 98.31\% \quad (5.18)$$

Resumen de la evaluación de resultados

Como punto final se presenta en la Tabla 20, un resumen de los parámetros de puntuación obtenidos para el sistema de agente conversacional de librería.

Tabla 20

Resumen de parámetros de confiabilidad

Desempeño teórico		Precisión Real	Satisfacción	Usabilidad	Confiabilidad
Exactitud	75%	81.73%	86.68%	83.57%	98.31%
Sensibilidad	77.08%				
Precisión	96.25%				
Total	83.92%				

Capítulo VI

Conclusiones y recomendaciones

Conclusiones

- La integración del procesamiento del lenguaje natural al agente conversacional, reduce el tiempo de la elaboración del corpus de entrenamiento pues no es necesario agregar conjugaciones de verbos, número gramatical en sustantivos, adjetivos, artículos, etc.
- La integración del agente conversacional en Messenger, fue rápida y sencilla, al usar las herramientas y documentación descrita en Facebook for Developers.
- La precisión de respuestas correctas con usuarios reales alcanzó el 81.73%, valor similar al 83,92% del teórico esperado, por lo que se garantiza que 8 de cada 10 preguntas serán contestadas correctamente.
- La confiabilidad del sistema alcanzó un valor aceptable de 98.31%, mientras que la experiencia de los usuarios también es buena pues incluye una satisfacción del usuario de 86.68% y la usabilidad del sistema de 83.57%.

Recomendaciones

- Se recomienda realizar comparaciones de confiabilidad con agentes de auto-respuesta que utilicen otros motores de decisión, tales como: redes neuronales artificiales o árboles de decisión.
- Implementar los algoritmos de agentes conversacionales en otras redes sociales, como Whatsapp, Telegram, Instagram, para así abarcar de mejor forma las necesidades de las PYMES en el país.
- Migrar el agente conversacional a otros lenguajes de programación, que cuenten con librerías para el procesamiento de lenguaje natural en el idioma español, además de herramientas para el desarrollo y evaluación de sistemas de inteligencia artificial.

Trabajo Futuro

Para futuros trabajos es necesario conectar la aplicación de agente de auto-respuesta con la base de datos de una empresa que posea varias sucursales, para así brindar respuestas que contengan la información de la sucursal en la que pueden encontrar los productos solicitados. Además de integrar un método de compra de productos en línea.

Integrar un método de compra de productos en línea, que incluya, método de pago paypal, tarjetas de crédito y débito, tarjetas de regalo y de descuentos. Además integrar un módulo que permita recolectar información para realizar entregas a domicilio. Y finalmente un módulo de aprendizaje automático no supervisado que le permita al agente conversacional aprender sin la necesidad de un desarrollador que edite su corpus.

Bibliografía

- Morales-Rodríguez, M. L., & Domínguez-Martínez, J. R. (2011). *Agentes Conversacionales como un Sistema de Diálogo*. Obtenido de www.academia.edu:
https://www.academia.edu/1187395/Agentes_Conversacionales_como_un_Sistema_de_Di%C3%A1logo
- A. Augello, G. P. (2012). An approach to enhance chatbot semantic power and maintainability: experiences within the FRASI project. *Semantic Computing (ICSC)*, 118-193: IEEE.
- A. Simon, M. S. (2016). An Overview of machine learning and its applications.
- Adiwardana, D., Luong, M.-T., R. So, D., Hall, J., Fiedel, N., Thoppilan, R., . . . Lu, Y. (2020). *Towards a Human-like Open-Domain Chatbot*.
- Aguilar, M. H., & Villegas, A. G. (2016). Análisis comparativo de la Escala de Usabilidad del Sistema. *Análisis comparativo de la Escala de Usabilidad del Sistema*, 5(10).
doi:10.23913/reci.v5i10.48
- Altn. (2019). *Autorespuesta*. Obtenido de
http://help.altn.com/mdaemon/es/ae_auto_responder.htm
- Benjamín, S. (2015). *Ensayo de procesamiento de lenguaje natural*. Los Cabos: Insituto de estudios superiores de los Cabos.
- Biehl, M. (2017). *Webhooks – Events for RESTful APIs*. API-University Press.
- Cahn, J. (2017). *CHATBOT: Architecture, Design, & Development*. Pennsylvania.
- Cahn, J. (2017). *CHATBOT: arquitectura, diseño y desarrollo*. University of pennsylvania school of engineering and applied science department of computer and information science.
- Castillo, E., Gutierrez, J. M., & Hadi, A. S. (1997). *Sistemas Expertos y Modelos de Redes Probabilísticas*. Obtenido de umich.mx:
<http://computo.fismat.umich.mx/~htejeda/gutierjm/BookCGH.pdf>

Chang, A. X., & Manning, C. D. (2014). *Defining cascaded regular expressions over tokens*.

Recuperado el 2020, de <https://nlp.stanford.edu/pubs/tokensregex-tr-2014.pdf>

Cortez, A., Vega, H., & Pariona, J. (2009). *Procesamiento de lenguaje natural*. Obtenido de

Revistas Investigacion:

<http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/download/5923/5121>

Crof, C. (18 de 12 de 2007). *A Brief History of The Facebook*. Obtenido de meerutcollege.org:

http://www.meerutcollege.org/mcm_admin/upload/1587223450.pdf

Developers, F. (2018). *Facebook for Developers*. Obtenido de

<https://developers.facebook.com/docs/>

Escobar Macías, A. (2019). *ANÁLISIS DEL USO DEL PROCESAMIENTO DEL LENGUAJE NATURAL Y*

SU APLICACIÓN EN SISTEMAS CONVERSACIONALES. Obtenido de

<http://repositorio.ug.edu.ec/handle/redug/44837>:

<http://repositorio.ug.edu.ec/bitstream/redug/44837/1/B-CINT-PTG->

[N.448%20Escobar%20Mac%c3%adas%20Ariana%20Dennise.pdf](http://repositorio.ug.edu.ec/bitstream/redug/44837/1/B-CINT-PTG-N.448%20Escobar%20Mac%c3%adas%20Ariana%20Dennise.pdf)

España, R. (23 de 09 de 2019). *TechBusiness*. Obtenido de Que es la inteligencia artificial:

<https://agenciab12.com/noticia/que-es-inteligencia->

[artificial#:~:text=El%20t%C3%A9rmino%20Inteligencia%20Artificial%20lo,procesos%20r](https://agenciab12.com/noticia/que-es-inteligencia-artificial#:~:text=El%20t%C3%A9rmino%20Inteligencia%20Artificial%20lo,procesos%20r)

[ob%C3%B3ticos%20la%20rob%C3%B3tica%20actual](https://agenciab12.com/noticia/que-es-inteligencia-artificial#:~:text=El%20t%C3%A9rmino%20Inteligencia%20Artificial%20lo,procesos%20rob%20la%20rob%C3%B3tica%20actual).

Estevez, P. (24 de 04 de 2020). *Interacciones de chatbots incrementan 87% por las ventas en*

línea. Obtenido de El CEO: [https://elceo.com/tecnologia/interacciones-de-chatbots-](https://elceo.com/tecnologia/interacciones-de-chatbots-incrementan-87-por-las-ventas-en-linea/)

[incrementan-87-por-las-ventas-en-linea/](https://elceo.com/tecnologia/interacciones-de-chatbots-incrementan-87-por-las-ventas-en-linea/)

Eva M Méndez Rodríguez, J. A. (3 de Septiembre de 1999). *Lenguaje natural e Indización*

automatizada. Obtenido de <http://eprints.rclis.org/12685/1/indizacion99.pdf>

- F. J. Martín Mateos, J. L. (2012 - 2013). *Procesamiento del Lenguaje Natural*. Obtenido de <https://www.cs.us.es/cursos/ia2/temas/tema-06.pdf>
- Facebook. (Abril de 2018). *Facebook for Developers*. Obtenido de <https://developers.facebook.com/docs/messenger-platform/reference/send-api>
- FacebookHelp. (2020). *Facebook*. Obtenido de <https://www.facebook.com/help/104002523024878>
- Fernandes, A. (08 de 2018). *Definición de NLP, NLU, NLG y cómo funcionan los Chatbots*. Obtenido de <https://planetachatbot.com/definicion-nlp-nlu-nlg-y-c0mo-funcionan-los-chatbots-fcf4b22e065a>
- Fernandes, A. (2018). *Definición de NLP, NLU, NLG y cómo funcionan los Chatbots*.
- Fernández Torres, A., Sandoya Villafuerte, J., & Crespo Torres, N. (09 de 10 de 2019). *Implementación de los agentes inteligentes en las pymes del Ecuador*. Obtenido de Dialnet: <https://dialnet.unirioja.es/servlet/articulo?codigo=7368588>
- Fuente, I., & Díaz, C. (2006). *Microsoft*. Obtenido de ENSEÑANZA DEL TEOREMA DE BAYES CON APOYO TECNOLÓGICO: <http://www.ugr.es/~batanero/pages/ARTICULOS/Thales2006.pdf>
- Gamallo Otero, P., & García González, M. (2012). *Técnicas de Procesamiento del Lenguaje Natural en la Recuperación de Información*. Santiago de Compostela: Universidade de Santiago de Compostela.
- García Serrano, A. (2017). *Inteligencia Artificial* (Vol. 2). Madrid: Alfaomega.
- GONZALEZ-CAMPOS, M. E., BERNABÉ-POVEDA, M. Á., & PAZMIÑO, M. F. (2017). METODOLOGÍA PARA EVALUAR LA USABILIDAD DEL VISUALIZADOR DE MAPAS DEL GEOPORTAL IDE DE ECUADOR. *Revista Internacional de Ciencia y Tecnología de la Información Geográfica*(19), 109-127.

- Google. (08 de 2020). *Formularios de Google*. Obtenido de <https://docs.google.com/forms/u/0/>
- Grampln. (2012). *Procesamiento de Lenguaje Natural*. Obtenido de https://www.fing.edu.uy/inco/cursos/grampln/presentaciones/GFLN2012_01_intro.pdf
- Gregoire , M., Yann, D., Kaisheng , Y., Yoshua, B., Li , D., Xiaodong, H., & Larry , H. (2015). *Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding*.
- Hoyle, R., Das, S., Kapadia, A., Lee, A., & Vaniea, K. (11 de Mayo de 2017). *Was my message read?: Privacy and Signaling on Facebook Messenger*. Obtenido de Digital Library ACM: <https://dl.acm.org/doi/pdf/10.1145/3025453.3025925>
- Huarote Zegarra, R. E. (13 de 07 de 2013). *Comparativa de sistema experto basado en reglas, redes neuronales y probabilidades*. Obtenido de <http://revistas.ucv.edu.pe/>: <http://revistas.ucv.edu.pe/index.php/RTD/article/view/680/530>
- I. Diaz Torres, E. V. (2020). *Aplicacion movil para el control remoto de dispositivos electronicos de una vivienda basada en un chatbot*. Sangolqui.
- IEC. (2001). *IEC 60300-3-10*. Obtenido de <https://webstore.iec.ch/publication/1295>
- ISO. (2019). *ISO 2500*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/iso-25010/24-fiabilidad>
- Jalife, A. A., Auza, G. C., & Atoany Fierro Radilla, M. N. (2017). Clasificación de Imágenes Urbanas Aéreas: Comparación entre Descriptores de Bajo Nivel y Aprendizaje Profundo. *SciELO Analytics*, 28(3).
- Kirkpatrick, D. (2010). *The facebook effect: The inside Story of the company*. New York: Simon & Schuster paperbacks. Obtenido de https://books.google.com.ec/books?hl=es&lr=&id=PxTvbM-VCPEC&oi=fnd&pg=PA1&dq=facebook+history&ots=DUmcrTSY1i&sig=NhacBcz3px6Er1sELoeTGBEYXI&redir_esc=y#v=onepage&q=facebook%20history&f=false

- Lengua, R. A. (13 de 8 de 2020). *Real Academia de la Lengua*. Obtenido de <https://dle.rae.es/personalidad>
- Libre, W. E. (5 de Abril de 2019). *Procesamiento de Lenguajes Naturales*. Obtenido de https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales#Historia
- Llorens Cerdà, F. (2011). Posibilidades de la plataforma Facebook para el aprendizaje colaborativo en Inea. *Revista de Universidad y Sociedad del conocimiento*. Obtenido de <https://www.learntechlib.org/p/149513/>
- Lovins, J. B. (1968). *Development of a Stemming Algorithm*. Massachusetts.
- Mantilla Sierra, J., & Mauro, A. (20 de Diciembre de 2014). *MiM*. Obtenido de Procesamiento del Lenguaje Natural : <http://www.scian.cl/archivos/uploads/1419931797.0517>
- McTear, M. (2018). Conversation modelling for chatbots: current approaches and future directions. *Konferenz Elektronische Sprachsignalverarbeitung*, (págs. 175-185). Ulm.
- Merchán Flores, J., Galar Idoate, M., & Sesma, M. S. (2020). *Implementación de chatbots con reconocimiento de voz para el control de equipos de consulta en quirófano*. Pamplona.
- Mg. Augusto Cortez, M. H. (2009). *Procesamiento de Lenguaje Natural*. Obtenido de http://200.62.146.19/bibvirtual/Publicaciones/risi/2009_n2/v6n2/a06v6n2.pdf
- Microsoft. (11 de 11 de 2018). *Microsoft* . Obtenido de Bot Framework Available: <https://docs.microsoft.com/en-us/bot-framework/>
- Microsoft. (02 de Octubre de 2018). *Microsoft* . Obtenido de Language Understanding LUIS: <https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/Home>
- Microsoft. (2020). *Visual Studio*. Obtenido de <https://code.visualstudio.com>
- MongoDB. (2020). *MongoDB*. Obtenido de <https://www.mongodb.com>
- Mora, S. L. (2001). *Programación de servidores web con CGI, SSI e IDC*. La Paz: Editorial Club Universitario.

Morales Rodriguez, M. L. (Mayo de 2011). *Diseño de un Asistente Virtual con Diálogo Emocional*.

Obtenido de www.academia.edu:

https://www.academia.edu/1188981/Tesis_Dise%C3%B1o_de_un_Asistente_Virtual_con_Di%C3%A1logo_Emocional

Moreno, J. A., & Antón, A. L. (2018). *Tutor basado en chatbot*. Barcelona.

Moya Rodríguez, J. L., Becerra Ferreiro, A. M., & Chagoyén Méndez, C. A. (Abril de 2012).

Utilización de Sistemas Basados en Reglas y en Casos para diseñar transmisiones por tornillo sinfín. Obtenido de SCielo:

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59442012000100001

Ngrok. (2020). *Ngrok*. Obtenido de <https://ngrok.com>

NodeJs. (2020). *NodeJs*. Obtenido de <https://nodejs.org>

Ono, K., Takeda, R., Nichols, E., Nakano, M., & Komatani, K. (2016). *Toward Lexical Acquisition during Dialogues through Implicit Confirmation for Closed-Domain Chatbots*. Japan.

Passino, K. M., & Yurkovich, S. (1998). *Fuzzy Control*. California: Addison Wesley Longman,.

Pérez, D., Alfonseca, E., Gliozzo, A., & Rodríguez, P. (2005). About the effects of combining Latent Semantic Analysis. *Signos*, 325-343. Obtenido de Universidad Autónoma de Madrid, España: <https://www.redalyc.org/pdf/1570/157013769004.pdf>

R. P. Diez, A. G. (2001). *Introducción a la inteligencia artificial: sistemas expertos, redes neuronales artificiales y computacion evolutiva*. . Universidad de Oviedo .

Ramírez, J. (2013). *Estrategias comerciales en el sector de la telefonía móvil. Análisis de factores condicionantes*. Oviedo: Universidad de Oviedo.

Ramos, F. M., & Velez, J. I. (Mayo de 2016). *Integración de Técnicas de Procesamiento de Lenguaje Natural a través de Servicios Web*. Obtenido de www.ridaa.unicen.edu.ar:

<https://www.ridaa.unicen.edu.ar/xmlui/bitstream/handle/123456789/644/Tesis%20de%20grado%20Velez-Ramos.pdf?sequence=1&isAllowed=y>

Rish, I. (2001). *An empirical study of the naive Bayes classifier*. Obtenido de

<https://www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf>

Rodríguez, S. C., González, E. F., & Hernández, H. C. (12 de 2011). *Case based expert system for hypertension diagnosis*. Obtenido de <http://www.scielo.org.co/>:

http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-62302011000400020

Russel, S., & Norvig, P. (2004). *Inteligencia Artificial un Enfoque Moderno* (Vol. 2). Madrid:

Pearson Prentice Hall.

Serna , A., Acevedo , E., & Serna, E. (2017). *Principios de la Inteligencia Artificial en las Ciencias Computacionales*.

Sobрино, J. A. (2001). *Teledetección*. Valencia, España: Universitat de València.

Suarez, J. A. (2006). *TÉCNICAS DE INTELIGENCIA ARTIFICIAL APLICADAS AL ANALISIS DE LA SOLVENCIA EMPRESARIAL*. Obtenido de

<https://dialnet.unirioja.es/servlet/articulo?codigo=1252835>

Tamayo, S., & Pérez-Marín, D. (2013). ANÁLISIS DE LA EXPERIENCIA DE USO DE UN AGENTE DE COMPRENSIÓN LECTORA CON. *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, 14, págs. 403-409. Salamanca.

Tear, M. M. (2018). *CONVERSATIONAL MODELLING FOR CHATBOTS: CURRENT APPROACHES AND FUTURE DIRECTIONS* . Reino Unido .

Torres, J. C. (2013). *Integracion de un Chatbot como habilidad de un robot social con gestor de dialogos*. España: Facultad de Ingeniería de Sistemas y Automática Universidad Carlos III.

Torres, J. N. (2018). *Estados Unidos Patente nº US20190347326A1*.

- Viloria, S. G. (2011). Sistemas integrados de gestión, un reto para las pequeñas. *Escenarios*, 69-89.
- Vukovic, D., & Dujlovic, I. (23 de 11 de 2016). *Facebook Messenger Bots and Their Application for Business*. Obtenido de IEEE XLORE:
<https://ieeexplore.ieee.org/abstract/document/7818926>
- We Are Social Ltd's. (2018). *GLOBAL DIGITAL REPORT*. Obtenido de We Are Social:
<https://digitalreport.wearesocial.com/>
- Yorita, A., Oakman, J., CHan, C., & Kubota, N. (2019). Self-Adapting Chatbot Personalities for better Peer Support. *IEEE International Conference on Systems, Man and Cybernetics (SMC)* (pág. 1). Italy: IEEE.
- Zarabia Zuñiga, O. (Abril, 2018). *Implementacion de un chatbot con Botframework: caso estudio, servicios a clientes del area de fianzas de seguro equinoccial*. Quito.