Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería de Ciencias y Sistemas

DEFINIR UNA GUÍA TEÓRICA Y PRÁCTICA PARA PODER IMPLEMENTAR ASISTENTES VIRTUALES EN ESPAÑOL

Luis Estuardo Azurdia Cárcamo
Asesorado por el Ing. Sergio Arnaldo Méndez Aguilar

Guatemala, junio de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

DEFINIR UNA GUÍA TEÓRICA Y PRÁCTICA PARA PODER IMPLEMENTAR ASISTENTES VIRTUALES EN ESPAÑOL

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

LUIS ESTUARDO AZURDIA CÁRCAMO
ASESORADO POR EL ING. SERGIO ARNALDO MÉNDEZ AGUILAR

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, JUNIO DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés de la Cruz Leal
VOCAL V	Br. Kevin Armando Cruz Lorente
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Ing. Herman Igor Véliz Linares
EXAMINADOR	Ing. César Rolando Batz Saquimux
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DEFINIR UNA GUÍA TEÓRICA Y PRÁCTICA PARA PODER IMPLEMENTAR ASISTENTES VIRTUALES EN ESPAÑOL

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Ciencias y Sistemas, con fecha 28 de agosto de 2019.

Luis Estuardo Azurdia Cárcamo

Guatemala, 18 de enero del 2020

Ingeniero
Carlos Alfredo Azurdia Morales
Revisor de tesis
Universidad de San Carlos de Guatemala
Presente

Por este medio atentamente me dirijo a usted, para comunicarle que he revisado la tesis del estudiante **Luis Estuardo Azurdia Cárcamo**, con número de carnet **201408606**, con título: **"Definir una guía teórica y práctica para poder implementar asistentes virtuales en español"**, luego de realizadas las revisiones correspondientes he encontrado que ha culminado el 100% de la investigación, en virtud de lo anterior recomiendo su aprobación.

Si otro particular me despido.

Atentamente,

Sergio Arnaldo Méndez Aguilar
Ingeniero en Ciencias y Sistemas
Colegiado No. 10958

Ing. Sergio Arnaldo Méndez Aguilar
Ingeniero en Ciencias y Sistemas
Asesor de trabajo de tesis

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 07 de julio de 2020

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **LUIS ESTUARDO AZURDIA CÁRCAMO** con carné **201408606 y CUI 2494 72732 0101** titulado **"ELABORACIÓN DE UNA GUÍA TEÓRICA Y PRÁCTICA PARA PODER IMPLEMENTAR ASISTENTES VIRTUALES EN ESPAÑOL"** y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÌA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación ***"DEFINIR UNA GUÍA TEÓRICA Y PRÁCTICA PARA PODER IMPLEMENTAR ASISTENTES VIRTUALES EN ESPAÑOL",*** *realizado por el estudiante, LUIS ESTUARDO AZURDIA CÁRCAMO aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

Msc. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 04 de junio de 2021

Decanato
Facultad de Ingeniería
24189101 - 24189102

DTG.251.2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **DEFINIR UNA GUÍA TEÓRICA Y PRÁCTICA PARA PODER IMPLEMENTAR ASISTENTES VIRTUALES EN ESPAÑOL,** presentado por el estudiante universitario: **Luis Estuardo Azurdia Cárcamo,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANA
FACULTAD DE INGENIERÍA

Inga. Anabela Cordova Estrada
Decana

Guatemala, junio de 2021

AACE/asga

ACTO QUE DEDICO A:

Dios	Por darme sabiduría.
Mis padres	Carlos Azurdia y Dora Cárcamo de Azurdia, por su incondicional amor.
Mi abuela	Noemi Argueta, que ha estado presente en todo.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por darme la oportunidad de estudiar, abrirme sus puertas y ser un profesional.
Facultad de Ingeniería	Por recibirme con las puertas abiertas, por todos los conocimientos y experiencias adquiridas durante los años y dejarme ser parte de su escuela.
Mis amigos de la Facultad	Juan Lemus, Alejandro Hernández, Aylin Azulema, Ali Daryousef, Fabio de Paz, Sharolin Lacunza, Andrea Vicente, Mauricio Batres, Alfredo Alvarado, Luis Barrios, Katerin Rodriguez, Oscar Monterroso, Daniel Solá, Miguel Ruano y Rodrigo Chaclan, por el apoyo incondicional.
Ing. Sergio Arnaldo Méndez Aguilar	Por la motivación para que continuara mi carrera universitaria
Mis hermanos	Andrés y Karla Azurdia, por estar siempre atentos.
Mi novia	María José Estrada Bonilla, por estar apoyando en las decisiones que he tomado.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
Σ	Conjunto de símbolos de entrada
Q	Estado
F	Estado final
q0	Estado inicial
δ	Función de transición

GLOSARIO

API	Protocolo de comunicación entre computadoras.
ASCII	Es el nombre que reciben los caracteres de una computadora. Ejemplo: código ASCCI de la letra “a” es 92.
Bot	Un asistente virtual con inteligencia artificial.
Chatbot	Asistente virtual.
Machine Learning	Concepto que se usa para el aprendizaje de las computadoras en la inteligencia artificial.
Python	Lenguaje de programación.
Serverless	Aplicación que no necesita uso de un servidor.
Servidor	Conjunto de computadora que logran realizar algún proceso de un programa.
WLan	Protocolo de comunicación de máquinas en una red de internet.

RESUMEN

La tesis elaborada contiene la información necesaria para la construcción de un asistente virtual con herramientas que se encuentran actualmente disponibles; además, incluye la teoría que se encuentra en la elaboración de un asistente virtual que es mejor conocido en el ambiente informático como *chatbot*. Junto a ello contiene información y consejos de seguridad, y los costos monetarios al crear un asistente virtual. En un apartado de la tesis contiene una guía simple de elaboración de un asistente virtual; por último, una lista de herramientas que son útiles para la elaboración de un asistente virtual.

OBJETIVOS

General

Definir una guía teórica y práctica para implementar asistentes virtuales en español.

Específicos

- Listar las mejores herramientas actuales para crear asistentes virtuales en español como una guía a empresas que necesita implementar ese tipo de soluciones.

- Describir e ilustrar los distintos pasos que deben realizarse para construir un asistente virtual.

- Describir la teoría involucrada en el desarrollo de un asistente virtual.

- Construir un asistente virtual en español para poder definir una guía de implementación y mejores prácticas.

- Enumerar y describir aspectos técnicos y costos para publicar asistentes virtuales en ambientes de producción.

INTRODUCCIÓN

La tecnología es algo que no podemos detener en el desarrollo laboral y no podemos dejar a un lado el beneficio que nos brinda. La tecnología nos ha facilitado la toma de decisiones y procesos, lo cual hace que las empresas, compañías y profesionales sean competitivos. Esto nos ha dado recursos donde las fronteras de un país no son el límite al hacer competencias de primer mundo, y no se debe justificar el crecimiento de una compañía sin tecnología. Esto quiere decir que tanto productos y servicio al cliente deben de seguir una norma de calidad y respuesta para el actual mercado demandante.

El servicio al cliente es parte fundamental de cualquier compañía; es la base para que se sostenga monetariamente y siga sus funciones. Es necesario cuidar a los clientes y obtener más de forma eficiente. Para ello puede ayudarnos la tecnología. Una marca debe estar en todos lados visibles para no pasar por alto algún cliente potencial; esto es difícil para pequeñas y medianas empresas, las cuales posiblemente aún no tengan recursos para competir con alguna megacorporación. La tecnología puede entrar a solucionar ese vacío. Las empresas deben colocarse lo más rápido posible en alguna red social, como Facebook, WhatsApp o Instagram, donde puedan ser vistas. Esta demanda ocupa una gran inversión por la facilidad que brindan las redes sociales de comunicarnos. Los usuarios prefieren comunicarse con una corporación que vende productos por medio de chat y que brindan estas tecnologías.

Los asistentes virtuales son el software que forma día a día los nuevos negocios como una vía súper práctica para tener un centro de atención al

cliente que ayuda rápidamente al prestigio de la empresa en alguna red social, y premia de alguna forma la rápida respuesta al cliente. Esto quiere decir que la compañía no pierde la oportunidad de enganchar a un cliente con sus productos o servicios, para evitar que se vayan con su competidor más cercano.

Esto lleva a la pregunta si realmente toda empresa requiere un asistente virtual para estar al nivel de sus competidores. La respuesta es sí. La empresa debe crecer exponencialmente a la tecnología. Hace algunos años se creía que todas las empresas deben tener un sitio web, ahora resulta que toda empresa debe estar al menos en alguna red social. La tendencia es que toda compañía debe atender a sus clientes las 24 horas del día, los 7 días de la semana.

Esto realmente ha sido un beneficio para las empresas que se dedican en el área de tecnología a crear software bajo demanda; esto implica la elaboración de algún asistente virtual. La industria de la tecnología ha visto a los asistentes virtuales como una necesidad de día a día para las marcas que desean competir, así que elaboran librerías que algún programador pueda utilizar, sean estas liberadas o privadas. Cabe resaltar que las empresas más grandes como Google, Amazon y Microsoft ofrecen estos servicios de una forma posiblemente barata.

La interrogante es que existe infinidad de herramientas que nos permiten crear estos asistentes, pero ¿cuáles son las más recomendables?, ¿por qué lo son?, ¿es fácil realizar una?, ¿es posible implementar alguna de forma más rápida que otra?, ¿la que ofrecen en línea obtiene información?, si soy una entidad bancaria, ¿puedo confiar en cualquier servicio que nos pueda proveer de forma segura por nuestros clientes?, ¿es rentable tener un asistente virtual o un departamento con mano de obra?

Este trabajo de grado está comprometido a responder todas estas interrogantes y para tomar la decisión correcta, y que los servicios más conocidos sean usados en distintos escenarios, y se pondrá a prueba su inteligencia artificial e implementación.

1. MARCO CONCEPTUAL

1.1. Antecedentes del problema

Los desarrolladores de aplicaciones para servicio al cliente quieren implementar un asistente virtual para atender a sus clientes las 24 horas al día y los 7 días de la semana. En los últimos años han surgido tendencias y herramientas, las cuales pueden crear asistentes virtuales, según investigaciones que se pueden consultar al respecto, como *Comportamiento adaptable de asistentes virtuales dependiente del contexto*, *Desarrollo de un asistente virtual utilizando Facebook Messenger para la mejora al cliente para la universidad privada de Tectana* y *Asistente virtual tipo asistente virtual Bogotá Colombia*.

La investigación *Comportamiento adaptable de asistentes virtuales dependiente del contexto,* habla sobre el comportamiento en contextos de los *bots* como lenguaje natural; se evaluó distintas pruebas a la inteligencia que participaron en concursos de inteligencia artificial para responder lo más natural posible. Estas pruebas solo fueron para asistentes diseñados para el idioma inglés.

Las investigaciones *Desarrollo de un asistente virtual utilizando Facebook Messenger para la mejora al cliente para la universidad privada de Tectana* y *Asistente virtual tipo asistente virtual Bogotá Colombia,* tiene en común la herramienta que utilizó API.ia (*Dialog Flow*). Ambas fueron realizadas y aceptadas por los clientes. Son favoritas las arquitecturas hechas por la nube.

1.2. Problema

Al estar creciendo esta rama de la inteligencia artificial, se ha puesto al mercado y a manos de programadores muchos desarrollos, los cuales tienen el mismo objetivo: crear un asistente virtual, pero surge la interrogante: ¿Qué herramientas se pueden utilizar para automatizar el soporte en línea en redes sociales sin intervención humana? Esta pregunta busca en la situación actual qué herramientas son las más útiles, baratas o eficientes según las necesidades. Un asistente virtual presentará el reto de ser un bienestar para cualquiera que desea implementar como tal.

1.3. Justificación

A menudo en distintos foros muchas personas implementan un asistente virtual. Estos tutoriales brindan a las personas cómo implementar un asistente virtual siempre utiliza un servicio de arquitectura basado en la nube de Google. Posiblemente sea el más simple, pero existen infinidad de herramientas que pueden ser igual de buenas o mejores, de las cuales ninguno habla o las compara de una forma adecuada. Además, de ninguna forma indica cómo funcionan o qué está detrás de ellas. Es posiblemente fácil implementar un asistente virtual pero no se comenta el precio de mantener uno o si es mejor tener un asistente virtual desde 0, y si es así qué costo tiene en hardware contar con uno.

Con el crecimiento y demanda para entidades que brindan algún servicio, las empresas dedicadas a la tecnología han propuesto sus herramientas para la creación de estos servicios, lo cual ha dado como resultado (de alguna u otra forma) una competencia poco regulada. Algunos no conocen ni siquiera las políticas de privacidad de ciertos servicios. Estos se pretenden realizar una

comparación que incluye respuesta de entrenamiento e implementación. Cabe resaltar que no se conoce de forma exacta un escenario donde se pueda involucrar precios sobre el uso de cada implementación.

1.4. Límites y alcances

La investigación se obtendrá a partir de la experimentación de distintas herramientas que se encuentran en el mercado. Esta información se buscará en foros y páginas de comunidades donde recomiendan el uso de las herramientas. Además, se implementará de forma casi genérica los pasos que se utilizan para la construcción de tales herramientas, poniendo en práctica y comprobando la teoría que se encuentra detrás de cada tecnología para conocer su trasfondo colectivo. Además, la experimentación de un asistente virtual construido con redes neuronales aplicando la teoría del lenguaje natural programada, involucraría la guía definitiva de la creación de un asistente virtual. Todas estas experimentaciones se lograrán a partir de implementaciones de arquitecturas en un computador.

Estos estudios serán sujetos a recomendaciones. Servirán para la selección, debido a que existen muchas formas de implementación se hará una selección de buenas prácticas.

2. MARCO TEÓRICO

Este capítulo contiene la teoría involucrada sombre el desarrollo de un asistente virtual, de tal manera es un aspecto totalmente teórico que fue tomado como base para la elaboración de distintos asistentes virtuales que tenemos hoy.

2.1. Teoría de autómatas

La teoría de autómatas es el estudio de los estados de una computadora. Esta ciencia está estrechamente relacionada con la teoría del lenguaje formal, ya que los autómatas son clasificados a menudo por la clase de lenguajes formales que son capaces de reconocer.

Un autómata es un modelo matemático para una máquina de estado finito (FSM sus siglas en inglés). Una FSM es una máquina que, al recibir un estado de entrada, el símbolo avanza al siguiente estado de transición.

2.2. Conceptos fundamentales de la teoría de autómatas

La importancia de la compresión de la teoría de autómatas es porque es el puente entre la comunicación de una computadora y el ser humano. A continuación, se presenta la división de los conceptos básicos.

2.2.1. Alfabetos

Los alfabetos son un conjunto de símbolos finitos y no vacío. Se pueden representarse usando el símbolo Σ. Entre los alfabetos más comunes están los siguientes:

- Alfabeto binario
- El conjunto de caracteres *ASCII*

2.2.2. Cadena de caracteres

Una cadena de caracteres también es una secuencia de símbolos finitos seleccionados para un alfabeto.

2.2.3. Lenguaje

El conjunto de cadenas que tienen un fin representativo es denominado lenguaje. Un lenguaje no necesita incluir cadenas con todos los símbolos.

Todos los alfabetos son finitos para todos los lenguajes, aunque parezca que las cadenas sean infinitas están restringidas por los símbolos del lenguaje.

2.2.4. Autómatas finitos

Un autómata finito tiene un conjunto de estados y su 'control' pasa de un estado a otro en respuesta a las entradas externas.

2.2.5. Diagramas de transiciones

Un diagrama de transiciones de un AFD A=(Q, Σ, δ, q0, F) es un grafo definido como sigue:

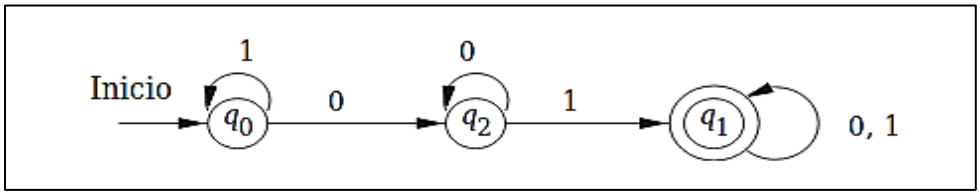
- Para cada estado de Q, existe un nodo.

- Para cada estado q de Q y cada símbolo de entrada 'a' de Σ, sea δ (q, a) = p. Entonces, el diagrama de transiciones tiene un arco desde el nodo q hasta el nodo p, etiquetado como 'a'. Si existen varios símbolos de entrada que dan lugar a transiciones desde q hasta p, entonces el diagrama de transiciones puede tener un único arco etiquetado con la lista de estos símbolos.

- Existe una flecha dirigida al estado inicial q0, etiquetado como Inicio. Esta flecha no tiene origen en ningún nodo.

- Los nodos correspondientes a los estados de aceptación (los que pertenecen a F) están marcados con un doble círculo. Los estados que no pertenecen a F tienen un círculo simple.

Figura 1. **Autómatas**



Fuente: HOPCROFT, John; MOTWANI, Rajeev y ULLMAN, Jeffrey. *Introducción a la teoría de autómatas lenguajes y computación*. p. 3.

El lenguaje $\Sigma = \{0,1\}$ y q_x son los estados.

2.3. Inteligencia artificial

La inteligencia artificial es un conjunto de algoritmos que intentan imitar el comportamiento que caracteriza al ser humano, la inteligencia. El uso de razón de los seres humanos da como resultado la creación de procesos, los cuales el ser humano puede guiar y tomar decisiones. La inteligencia artificial imita ese comportamiento por medio de un aprendizaje desarrollado por componentes computacionales de un ser humano.

2.3.1. Inteligencia artificial débil

Son aquellos sistemas que solamente pueden cumplir un número limitado de tarea. Todas las inteligencias artificiales creadas por el humano actualmente se encuentran en este grupo.

2.3.2. Inteligencia artificial fuerte

Es la inteligencia artificial que puede acoplarse a distintas problemas y dominios diferentes.

2.3.3. *Machine Learning*

Es la rama de la inteligencia artificial que busca dotar a las máquinas la capacidad de aprender. El *Machine Learning* se puede dividir en aprendizaje supervisado y no supervisado.

El aprendizaje supervisado es una clasificación de *Machine Learning* que permite a una computadora o máquina aprender por medio de variables entradas con una relación existente a variables de salida, en la cual los datos se encuentran clasificados. En esta rama debe intervenir un tercer agente, por lo regular es un humano enseñando como a un niño los colores.

2.3.3.1. Aprendizaje no supervisado

El aprendizaje supervisado y el futuro de la inteligencia artificial genera conocimiento a partir de los datos que solo se proporcionan, de entrada, sin alguna necesidad de explicarle al sistema qué resultados debe obtener.

2.3.4. Redes neuronales

Una red neuronal es un conjunto de nodos que están unidas por conexiones. Similares a una neurona real, reciben estímulos internos, los cuales realizarán cálculos internos y generarán valores de salida. Esta es una función matemática ponderada; la ponderación de cada una de ellas está dada por el peso de la entrada.

2.3.4.1. Redes neuronales alimentación progresiva

En esta red las conexiones son unidireccionales y no existen ciclos. La red es una gráfica acíclica dirigida, es un grafo que no tiene ciclos.

2.3.4.2. Redes neuronales recurrentes

En estas, las conexiones pueden formar topologías arbitrarias, más asociadas al cerebro humano. Cuentan con un estado de guardado en los

niveles de activación de las unidades; es decir, podemos crear modelos con estado.

2.4. Árboles de decisión

Es una estructura jerárquica o mapa de posibles caminos que permite comparar entre sí y tomar el mejor camino en función de costos, probabilidad o beneficios.

2.5. Sistemas expertos

Son aquellos sistemas de toma de decisiones con la competencia de un experto a partir de la inferencia; además, razona las operaciones que realiza, de cómo y por qué las ejecuta. También puede ser fácil de modificar sin necesidad de reprogramar y es hecho con un lenguaje natural limitado.

2.6. Teoría de agentes

Un agente es cualquier ente capaz de percibir su medio ambiente con la ayuda de sensores y realizar acciones en su medio. Actúa de forma lógica para realizar una acción.

2.7. Propiedad del entorno del agente

- Determinista: el agente tiene unas acciones limitadas que se asocian a cada transición.

- Casi observable: los entornos no siempre se pueden controlar; simplemente se pueden observar.

- Secuencial: una actividad lleva al a otro en un orden definido.

- Discreto: las acciones son finitas y siempre tendrán un inicio y un final.

- Agentes múltiples: se dice que es un agente múltiple cuando varias partes diferentes actúan entre sí.

2.8. Procesamiento de lenguaje natural (PNL)

El lenguaje natural es una rama de la inteligencia artificial que se encarga de estudiar la comunicación del ser humano. Este procesamiento se ha extendido gracias a la teoría de lingüística de Chomsky, quien relata que la lengua del humano es muy ambigua, empezando por lo léxico de una palabra. Sin embargo, una palabra puede combatir su ambigüedad en función del contexto, obteniendo como resultado la semántica de una oración.

Esta ciencia debe encargarse de muchas interpretaciones de un ser humano, como los pronombres. Maravillosamente, el ser humano es capaz de interpretar y seguir una conversación. Un ejemplo de una conversación es: <<Juan juega todas las mañanas, él también desayuna>>. Somos capaces de interpretar esa oración sin ningún esfuerzo, pero para que una máquina logre interpretarlo se ha creado varios algoritmos, como *Pronominal Anaphora Resolution* [Lappin, Leass 1994].

2.9. Asistentes virtuales (asistentes virtuales)

Un asistente virtual es un sistema informático capaz de mantener una conversación mediante un lenguaje natural humano con una persona o con otro asistente virtual. Este sistema ha sido programado de forma inteligente para

interpretar un contexto de una conversación y así tomar la decisión que lleva a un flujo de la conversación. Estos programas deben ser entrenados para que su nivel de respuesta sea cada vez más aceptado.

2.9.1. Características de un asistente virtual

- Diseño del diálogo: el diálogo es una de las características más importantes de un asistente virtual. Es el diálogo que se usará para el humano. Es importante que el *bot* sea el que aprenda a dialogar con el humano.

- Inteligencia artificial: es la forma en la cual un asistente virtual facilitará la interpretación de la conversación y que tomará decisiones.

- Programación y despliegue: el asistente virtual debe programarse solo una vez y debe ser independiente a cada plataforma (Messenger, Telegram, WhatsApp. El asistente virtual debe adaptarse a cualquier plataforma sin necesidad de crear una nueva inteligencia artificial o reprogramar todo de nuevo.

- Entrenamiento: un asistente virtual nunca sabrá todas las respuestas, es importante conectarse a una inteligencia artificial. El asistente virtual debe aprender nuevas preguntas y respuestas para una mejora continua.

- Marketing y posición: es importante utilizar una plataforma en la cual pueda ser reconocido el asistente virtual.

2.10. Conceptos de un asistente virtual

- Intenciones son todos aquellos métodos que interpretan lo que el usuario ha querido comunicar, por ejemplo: se ha definido un asistente virtual para compra de vehículos, con diferentes intenciones como comprar carro, ver carro, buscar carro.

- Entidades: van asociadas con intenciones. Son aquellas palabras o frases que determinarán la respuesta del asistente virtual, por ejemplo: para nuestra intención, comprar vehículo, podemos tener una entidad 'estado vehículo' que contendrá si el vehículo es nuevo o usado. También una entidad 'lugar', en donde se encuentra el vehículo.

- Diálogos: es la respuesta programada que ofrece un asistente virtual después de escuchar la intención.

3. ASISTENTES VIRTUALES

Este capítulo tiene como objetivo conocer las herramientas actuales para la implementación de un asistente virtual, y las distintas soluciones que pueden elegir las empresas para implementar uno de estos asistentes virtuales.

3.1. *Instant Messenger*

Son aquellos softwares que comúnmente se usan para chatear con otra entidad, tales como WhatsApp, Messenger, Telegram. Otra definición puede ser un software que logra la transmisión de mensajería en tiempo real. Son usuales en las redes sociales. La comunicación entre los dispositivos debe estar conectada a un tipo de red tal como *Wlan*, Internet, entre otros.

3.1.1. WhatsApp

> WhatsApp fue fundada por Jan Koum y Brian Acton quienes conjuntamente trabajaron por 20 años en Yahoo. WhatsApp se unió a Facebook en el 2014, pero continúa operando como una aplicación independiente y enfocada en construir un servicio de mensajería rápido y confiable en cualquier parte del mundo.[1]

3.1.2. Telegram

Telegram es un *instant Messenger* que ha tenido popularidad los últimos años. Fue lanzada en el año 2013. Su origen es ruso. Comparte su API para

1 WhatsApp. *Acerca de WhatsApp*. https://www.WhatsApp.com/about/?lang=es. Consulta: 15 de septiembre de 2019.

cualquier desarrollador, lo que le permite crecer en distintos sistemas operativos y a los negocios o marcas optar por un chat para teléfonos móviles.

3.1.3. Facebook Messenger

Facebook Messenger es otro *instant messenger* propiedad de Facebook. Funciona independientemente de la plataforma, es una opción casi obligada para las empresas, sin contar empresas de países en donde se encuentra restringido Facebook.

3.2. API para asistentes virtuales

La construcción de asistentes virtuales requiere libertad para acceder como un humano a un *instant messenger* para interactuar como humano. Los *instant Messenger* ofrecen sus *API* para que un desarrollador pueda implementar un asistente virtual.

- WhatsApp

Ofrece su API como WhatsApp Business. Para el uso de esta API se debe registrar un negocio con un número de teléfono, el cual identificará a la empresa. Se debe comprobar del número.

- Telegram

Ofrece su API para cualquier usuario, realmente no es nada difícil registrarse más solo ingresar a la cuenta de Telegram y dirigirse a la opción de desarrollador.

- Facebook Messenger

El uso de esta API es más restringido, posiblemente por la cantidad de información, comportamiento, preferencias de una persona. Para optar por esta API es necesario tener una cuenta activa de Facebook. Además, la empresa debe de tener una página (fan page) de Facebook. Una vez se cumplan estos requisitos, en la opción de desarrollo existe otras restricciones, como crear una página web alterna (en otro servidor) con las políticas de privacidad, en las cuales se muestra cuál será el uso de la API. Se debe agregar una empresa encargada que usará la API y Facebook debe aprobar los requisitos para el uso de dicha API.

3.3. Tipos de asistentes virtuales

Para conocer un poco de los asistentes virtuales es necesario saber su subclasificación, para lo cual se elaboró una lista de los tipos con una pequeña descripción.

3.3.1. *Dumb Chatbots*

Estos son los sistemas de asistentes virtuales más comunes y sencillos de realizar. Por lo regular están incluidos en los *instant Messenger* para negocios. Funcionan para respuestas inmediatas a saludos. Un ejemplo de una conversación es:

- Usuario: Hola buenas tardes.

- Asistente virtual: Hola buenas tardes, lo atenderemos lo más pronto posible.

Este tipo de asistentes virtuales también puede almacenar preguntas y respuestas no tan complicada en el sentido semántico.

3.3.2. *Smart chatbots*

Estos tipos de asistentes virtuales son más elaborados, tienen la capacidad de tomar decisiones y llevar una conversación mucho más fluida y natural. Gracias a la integración de la inteligencia artificial, su nivel de integridad y razonamiento es directamente proporcional a la dedicación de la inteligencia artificial que se proporciona. Esta es una de las razones por las que es necesario seleccionar el motor de inteligencia artificial para cada asistente virtual.

3.4. Aplicaciones prácticas de los asistentes virtuales

El mercado de los asistentes virtuales ha evolucionado con el tiempo. Para realizar un tipo de *Smart chatbot* debemos definir alcances. Se podrá hacer énfasis en la necesidad de cada proyecto y asignar un equipo para dicho trabajo.

3.4.1. Atención al cliente

Estos asistentes virtuales son los más comúnmente demandados. Su objetivo es recolectar datos de los clientes. Son entrenados para contestar preguntas frecuentes que asocian con el entrenamiento previo.

3.4.2. Gestión de compras y pagos *online*

Este tipo de uso es mucho más personal y delicado por la razón de datos monetarios como tarjetas de crédito, pero puede tener alguna de estas funcionalidades:

- Ayuda de búsqueda de artículos similares
- Guían en el proceso de compra
- Pagos en línea

3.4.3. Envío de información y novedades sobre la empresa

Este tipo de asistentes virtuales es elemental para una empresa, porque automatiza el envío de mensajes masivos a clientes, clientes potenciales o clientes en seguimiento. Es fundamental segmentar a los clientes porque los mensajes pueden llegar a no ser deseados por el usuario. Los *instant Messenger* tienen la información de la actividad de los clientes con el uso del chat, esto logra ser una herramienta muy útil para mandar un mensaje con una probabilidad mucho mayor de ser leído. Un asistente virtual de este tipo automatiza todo este proceso.

3.4.4. Soporte

Este tipo de asistentes virtuales son muy difíciles de implementar por el uso extenso de inteligencia artificial. Tratan de simular lo inteligente que es un humano para la toma de decisiones, por consiguiente, es el mejor simulador de una conversación de un cliente.

3.5. Lenguajes de programación utilizados para asistentes virtuales

Existe diversidad de lenguajes populares para la creación de asistentes virtuales.

3.5.1. *Python*

Python es un lenguaje interpretado. Se ha hecho muy conocido por el hecho de su sintaxis simple. Existe una gran cantidad de librerías extensas sobre *Machine Learning* que ha facilitado a los desarrolladores trabajar sobre *Python*, entre ellas, NLTK.

3.5.2. Node

Node es un intérprete de código JavaScript muy simple de utilizar, ya que no es tipado sino simple para comenzar y todo desarrollador lo conoce. No es muy popular para la creación de asistentes virtuales formales, pero hay varias librerías que pueden ayudar.

3.5.3. C#

Es un lenguaje orientado a objetos propiedad de Microsoft. Es muy estable y gracias a su IDE (entorno de desarrollo integrado, por sus siglas en inglés), Visual Studio, facilita el trabajo de la integración de un asistente virtual a la nube de Microsoft. Azure es una de las nubes públicas más grandes del mundo. La elaboración de un asistente virtual es muy simple en la inteligencia artificial usando un PNL. Visual Studio trae *Bot Framework* que es muy fácil de empezar. Existen muchos tutoriales en línea para utilizar este lenguaje.

3.6. Herramientas existentes para realizar un asistente virtual

PNL es una forma en que las computadoras analizan, comprenden y derivan el significado del lenguaje humano de una manera inteligente y útil. Al utilizar el PNL, los desarrolladores pueden organizar y estructurar el conocimiento para realizar tareas como el resumen automático, la traducción, el reconocimiento de la entidad nombrada, la extracción de relaciones, el análisis de sentimientos, el reconocimiento de voz y la segmentación de temas.

3.6.1. *Dialogflow* (Google)

Es una plataforma para desarrollar asistentes virtuales. Esta herramienta nos ofrece la creación de agentes con PNL y utiliza la gestión dinámica de subcontexto. Al aceptar el uso de este servicio de Google se aceptan dentro del contrato los términos de servicios de Google *APIs* y los términos de contrato de Fire Base.

Dialogflow ofrece dos planes de uso, *Standard Edition* y *Enterprise Edition*. Un distintivo es el análisis de sentimiento de una conversación, que es lo más relevante de esta diferenciación. El uso de *Enterprise Edition* se rige a otros términos y condiciones basados en Google *Cloud Platform*.

Algunos de los entornos de desarrollo disponibles en *Dialogflow* son: Node.js, .NET, C++, *Python*, Ruby, PHP, Java, Android, Xamarin e iOs.

3.6.2. Microsoft *Bot Framework*

La plataforma de desarrollo de Microsoft consta de tres partes:

- Portal de desarrollo
- *Bot Connector*
- *Bot Directory*

Portal de desarrollo: este *Framework* permite un entorno flexible para los desarrolladores experimentados. Tiene muy buen soporte para usuario de .Net, Node.js y *Python*.

- *Bot Connector*: permite conectar el asistente virtual a distintos entornos como *Cortanar*, *instant Messenger*, correo electrónico y aplicaciones propias.

- *Bot Directory*: es una colección de asistentes virtuales desarrollados con Microsoft *Bot Framework*.

Este *Framework* integra el PNL llamado LUIS (*Language Undestanding Intelligent Service*) de Microsoft.

3.6.3. Amazon Lex

Es un servicio de AWS para crear asistentes virtuales. El sistema PNL está basado en *intents slots*. Los *slots* son los parámetros que puede requerir un *intent*.

Cuenta con los siguientes entornos de programación: Node.js, *Python*, .Net, Android, IOS, Java, PHP y Ruby. La integración de este servicio puede agregarse con otro servicio de AWS.

3.6.4. *TensorFlow*

Es la forma más básica para crear un asistente virtual. Esta librería existe para los lenguajes Swift, Node.js y *Python. TensorFlow* es una herramienta para construcción de redes neuronales. Es de código abierto de Google que se puede implementar en cualquier proyecto que requiera *Machine Learning.*

3.6.5. Rasa

Es una herramienta para elaborar asistentes virtuales que debe entrenarse para realizar las acciones. Es posible construir un asistente virtual con esta herramienta de forma local; es una de las vías más simples para realizar. Se integra muy fácil con *TensorFlow* para crear un lenguaje más complicado de entender.

4. ARQUITECTURA DE UN ASISTENTE VIRTUAL

A continuación, se presentan los aspectos técnicos que enriquecen el entendimiento de las herramientas y algunas buenas prácticas de seguridad. Los siguientes aspectos técnicos son esenciales para todas las herramientas.

4.1. Manipulación de texto

La manipulación de texto es algo vital para la creación de un asistente virtual. Estas herramientas están incluidas en las de asistentes virtuales.

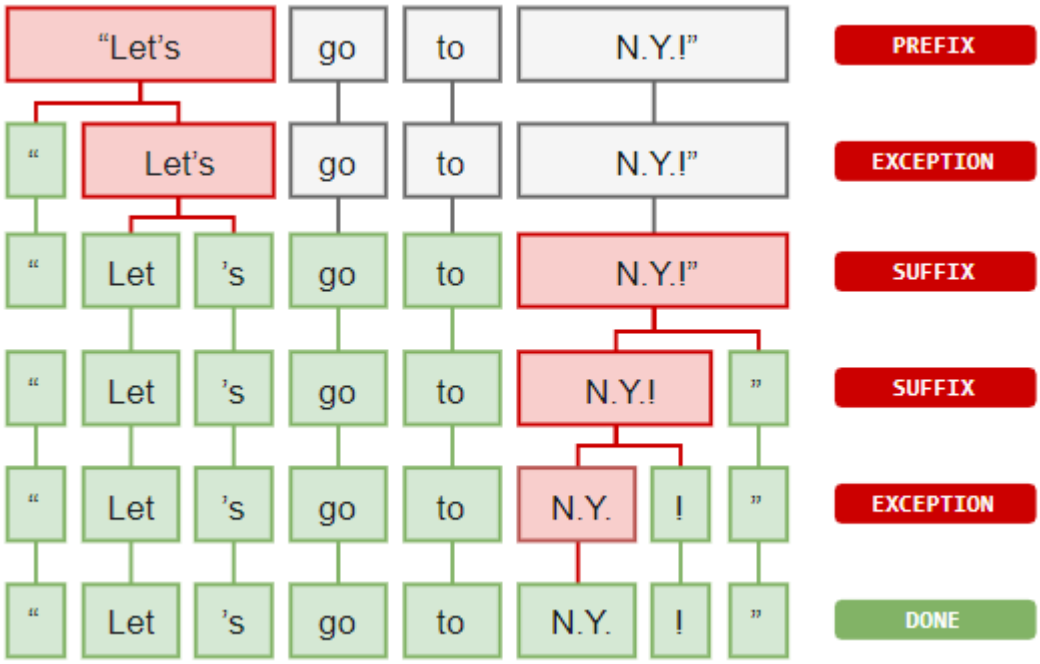
4.1.1. Normalización

El proceso de normalización es convertir todo el texto en mayúsculas o minúsculas, para que el algoritmo no trate las mismas palabras como diferentes.

4.1.2. Formar *tokens* (*tokenization*)

Es la parte inicial del procesamiento del lenguaje. Al generar *tokens* nos aseguramos de que cada palabra pertenece a una gramática. Existen librerías en *TensorFlow* que ayuda a realizar este proceso.

Figura 2. **Proceso de *tokenization***



Fuente: MA, Edward. *PNL Pipeline: Word Tokenization (Part 1),* https://medium.com/@makcedward/PNL-pipeline-word-tokenization-part-1-4b2b547e6a3. Consulta: 18 de noviembre de 2019.

4.1.3. Palabras derivadas (*Stemming*)

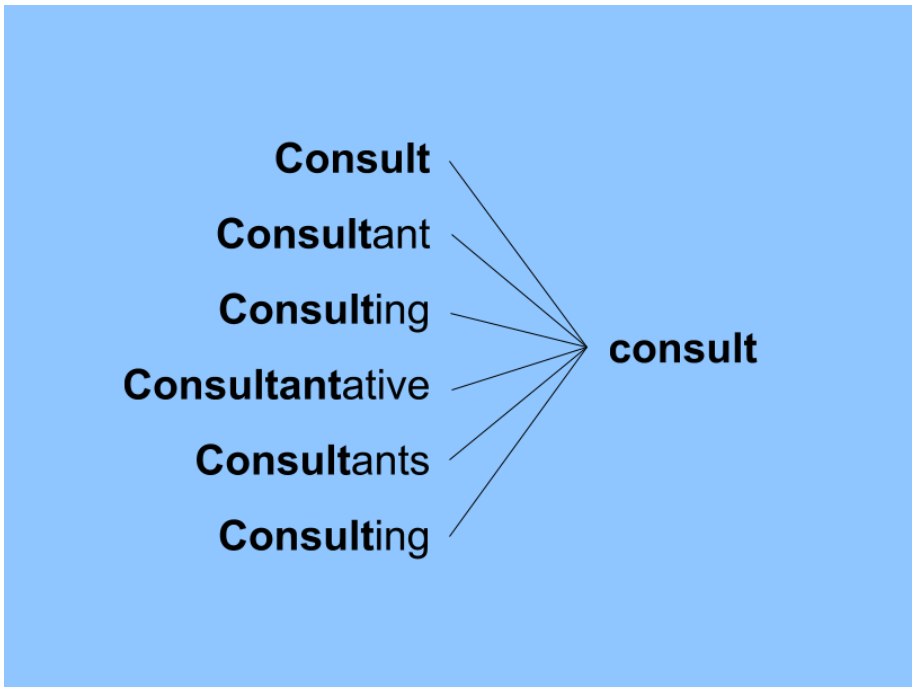
Stemming es el siguiente proceso. Es sobre un conjunto de palabras que debe significar lo mismo. En el idioma español requiere una enorme tarea porque hay muchas variaciones en los verbos que pueden significar ser acciones de lo mismo.

Figura 3. **Variación del verbo hacer**

Fuente: Open English. *¿Cree que es difícil el inglés?*. https://www.facebook.com/openenglish. Consulta: 18 de noviembre de 2019.

El proceso de reducir las palabras con inflexión (o algunas veces derivadas) a su forma base o raíz, generalmente una forma de palabra escrita.

Figura 4. **Proceso de *stemming***



Fuente: Developedia*. Stemming*, https://devopedia.org/stemming. Consulta: 18 de noviembre de 2019.

4.1.4. Lematización

Esto es una derivación de *stemming* que busca en función del lema. Con esto es posible encontrar errores e interpretarlos sin ningún problema.

Tabla I. **Proceso de lematización**

Lexema	**Significado**
ccorriendo	Correr
avlar	hablar

Fuente: elaboración propia.

4.1.5. Categorización y palabras claves (*Tagging words*)

Para procesar el texto debemos indicar de una forma explícita un contexto a la conversación. Si el asistente virtual es para compras, se debe crear un banco de palabras que indiquen 'qué se va a comprar', 'cómo es lo que compra', venta, entre otros. Al ser almacenado se debe colocar un peso a las palabras, refriéndose a una acción o a una característica. Supongamos que tenemos una florería. La conversación seria así:

- Hola, quiero saber información de una rosa.

Tabla II. **Clasificación y palabras clave**

Palabra	**Clasificación**	**Categoría**	**Peso**
Hola	Saludo	Saludo	1
quiero	Verbo	Ninguno	0
saber	Verbo	Información	2
Información	Detalle	Información	2
De	Ninguno	Ninguno	0
Una	Ninguno	Ninguno	0
Rosa	Producto	Rosas	5

Fuente: elaboración propia.

En la tabla II es posible dar jerarquía al interpretar la oración. La forma más básica de interpretar la conversación es 'hola saber rosa'. Esto es entendible para consultar a una base de datos y mostrar el precio de una rosa, De esta manera es posible construir frases de entrada para cambiar parámetros repetitivos como 'rosa', que puede ser otro tipo de planta. Una rosa está clasificada como 'producto' y está en la categoría 'rosas'; la categoría puede cambiar a cactus. Se puede analizar la palabra 'saber' que está en la

clasificación de un verbo y su categoría es información; si se cambia a 'comprar' sigue siendo verbo, pero es una compra.

Esta clasificación no solo debe ser de entrada. Las respuestas de los asistentes deben ser clasificadas de igual manera. Con el ejemplo anterior, el asistente virtual tiene la información de saludo, información y rosa. En la categoría 'saludo', el asistente debe tener un diccionario de saludos, por ejemplo: 'Qué tal'. El aspecto 'información' indica que va a consultar la información de un producto, entonces debe contestar 'el precio es de' y se añade el precio del producto, en este caso la rosa. El asistente virtual generaría una respuesta como esta:

- Qué tal, el precio de la rosa por unidad es de Q 25,00.

Tabla III. **Construcción de respuesta por entrada**

Respuesta	**Categoría y/o clasificación**
Qué tal	Saludo
El precio de	Información + producto
Rosa por unidad es de Q 25,00	Producto + rosa

Fuente: elaboración propia.

En la tabla III se aprecia que la combinación de categorías y clasificación de las palabras. Es posible crear una respuesta para que parezca natural.

4.2. Lectura de parámetros

Es importante conocer para qué se desea implementar un asistente virtual, con esto se puede formar una idea de lo que puede pedir un cliente al asistente virtual. Ejemplo: supongamos que se encuentra en una tienda de flores, la tienda de flores vende diferentes tipos de flores: cactus, rosas, pinos, entre otros.

- Cliente: hola, ¿cuánto cuestan los cactus?
- Asistente virtual: hola, los cactus se encuentran a Q 25,00

Dado este ejemplo, la pregunta de los clientes puede ser similar. Los parámetros que pueden cambiar son los tipos de cactus almacenados en una base de datos. El asistente virtual la puede consultar para definir los parámetros de lectura de una conversación. Al generalizar la pregunta pude quedar de esta manera:

- ¿Cuánto cuestan los {tipo_planta}?
- Donde {tipo_planta} es un parámetro variable dónde es posible un cambio por cliente.

4.3. Entrenar la inteligencia artificial

Es importante entrenar una inteligencia artificial para la implementación de un asistente virtual. Se debe tomar en cuenta la posibilidad de palabras que podrían abarcar una conversación al definir los *intent* (intenciones) correctamente. Esto también podría interpretarse como palabras clave para que el asistente pueda responder. Ejemplo: supongamos que estamos en la tienda

de flores y ya definimos los parámetros, ahora para entrenar la inteligencia artificial.

- ¿Cuánto cuesta los/las {tipo_planta}?
- ¿Cuánto vale los/las {tipo_planta}?

Como se observa, una palabra clave es 'cuánto' que puede interpretarse como la cantidad, mientras 'vale' y 'cuesta' es el valor monetario del producto. También debe tomarse en cuenta todas posibilidades de preguntas que podría hacer un usuario. Por ejemplo:

- ¿Cuánto {tipo_planta} existe?

Se tiene en cuenta ahora la existencia y cantidad del producto. El asistente virtual debe interpretarlo y contestar.

4.4. Agregar a una infraestructura

Se debe contar con una infraestructura IT donde se alojará el *bot* para su futuro uso. En la actualidad existen diferentes proveedores de nubes públicas que es posible utilizar, como Google, Azure y AWS. Es importante tener el nivel de seguridad adecuado, estabilidad y alta disponibilidad, para evitar caídas de los servidores.

4.5. Monitorear

Es la parte más importante que se debe tomar en cuenta. Es la que indicará cómo mejorar todo el proceso de crear un *bot* para una empresa,

identificar patrones de preguntas o quejas muy seguidas para identificar alguna mejora.

4.6. Arquitecturas para asistente virtual

Para una arquitectura de un asistente virtual existen tres formas habituales para su creación.

4.6.1. *Serverless*

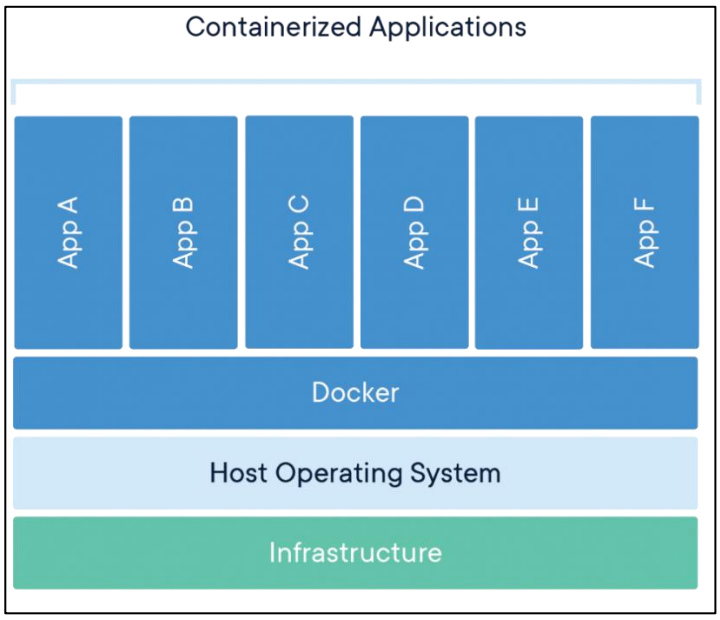
Es una de las tecnologías más nuevas de trabajar en la nube. *Serverless* es un concepto nuevo que al traducirlo significa 'sin servidor'. La arquitectura es consumida por un API, que el proveedor brinda, Amazon Lex y Google *Dialog Flow*. Son asistentes virtuales *serverless.* Estas arquitecturas cobran por consulta a sus servicios, y no por uso de instancias.

Una arquitectura *serverless* es aquella donde los desarrolladores obtienen el servicio que brinda la nube por medio de API y no deben encargarse de actualización de sistema operativo ni realizar algún crecimiento de forma horizontal o vertical al hardware.

4.6.2. *Container*

El contenedor es otra arquitectura muy útil cuando se trata de escalabilidad y mantenimiento. Es muy fácil de compartir y pude ser implementado en una instancia o un entorno local. Esta arquitectura empaca una aplicación de una manera simple. En la figura 5 muestra la arquitectura de un contenedor utilizando *Docker* como administrador de contenedores.

Figura 5. **Arquitectura de contenedores**



Fuente: Docker. *What is a container?,* https://www.docker.com/resources/what-container. Consulta: septiembre de 2019.

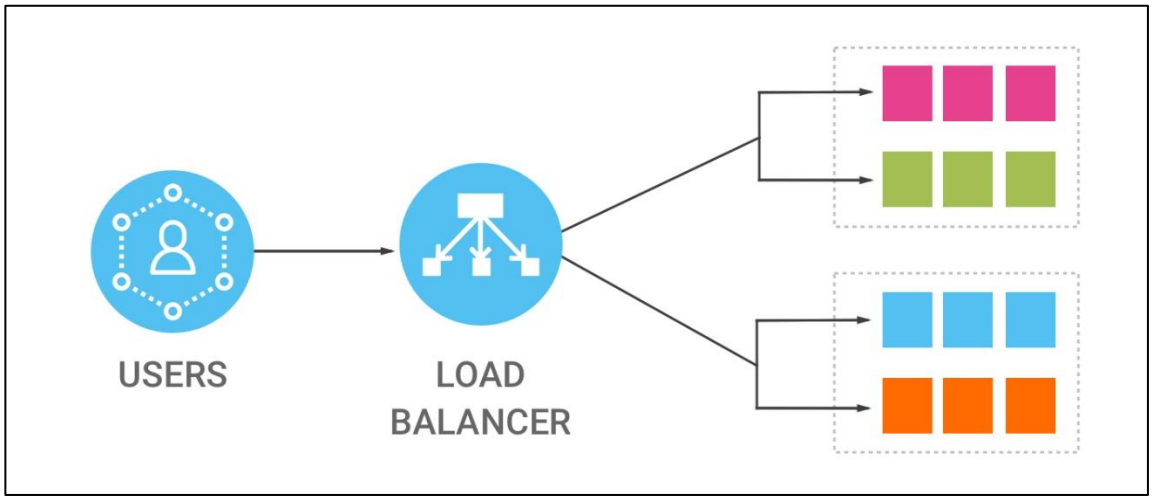
Con la arquitectura de contenedores es muy simple el cambio de ambiente entre computadoras. Es ideal para la creación de un asistente virtual con algún lenguaje de programación. Se debe preparar el ambiente de desarrollo dentro del contenedor agregando librerías, dependencias del lenguaje de programación, entre otros, para luego ser publicado en algún servidor.

4.6.3. Instancias

Una instancia es una máquina virtual que ofrecen los proveedores de nube; es hardware. En una instancia es posible agregar contenedores; además, es posible crear un clúster de instancias que contengan contenedores, las cuales pueden crecer horizontalmente. Esto ayudará a que el aplicativo que

está en las instancias tenga recursos de hardware para su óptimo funcionamiento. En la figura 6 se muestra cómo un usuario hace la petición a un balanceador de carga (*load balancer*), que está conectado a dos *clústers* de instancias.

Figura 6. **Balanceador de carga con dos instancias**



Fuente: SANDBU, Marius. *Outbound traffic with Standard Load Balancer Microsoft Azure*. https://msandbu.org/outbound-traffic-with-standard-load-balancer-microsoft-azure/ Consulta: septiembre de 2019.

4.7. Seguridad en asistentes virtuales

La seguridad en los asistentes virtuales es necesaria. Facebook exige que para usar su API haya un archivo de políticas de seguridad. Algunos asistentes virtuales utilizan información del usuario, como tarjetas de crédito, dirección, cuentas bancarias y/o alguna información confidencial.

4.7.1. Alertas de seguridad

Existen dos tipos de alertas de seguridad que deben tornarse en cuenta y realizar pruebas necesarias para que estas se cumplan.

4.7.1.1. Amenazas

Una amenaza es llamada así para cualquier sistema que comprometa un peligro.

Una amenaza puede costar hasta millones de dólares para una compañía. Puede ser una manipulación de la información indebida, suplantación de identidad.

Tabla IV. **Ataques comunes por hackers**

Ataque	Intención del ataque	Técnicas de reducción
Spoofing	Acceso de información de otro usuario por algo o alguna persona utilizando las credenciales del usuario.	Realizar una apropiada autenticación. Proteger los datos confidenciales. No exponer todo los datos.
Tampering	Cambio malicioso o modificación de data.	Encriptar datos Firmas digitales Habilitar la auditoría de autenticación.
Repudiation	Realizar operaciones indebidas a los sistemas.	Firmas digitales Uso de autenticación por tiempo.
Divulgación de Información	Robo de información.	Encriptación Protocolos de privacidad Excluir datos sensibles en la autenticación.
Denegación de servicios	Denegar el acceso a los usuarios.	Autenticación autorizada Filtro de accesos a los servicios.
Elevar permisos de usuario	Cambio de privilegios a otros usuarios para el uso del sistema.	Manejar jerarquías de usuarios. Autenticación fuerte Principio de menor privilegio

Fuente: elaboración propia.

4.7.1.2. Vulnerabilidad

Un sistema puede volverse vulnerable y sensible a los ataques si no se establece un buen mantenimiento, tiene una codificación deficiente, carece de protección o hay errores humanos. La manera más efectiva es implementar actividades de SDL (ciclo de vida del desarrollo de la seguridad).

Figura 7. **Ciclo de vida del desarrollo de la seguridad**

Fuente: elaboración propia, empleando herramienta SmartArt de Microsoft Word.

4.7.2. Mejores prácticas de seguridad

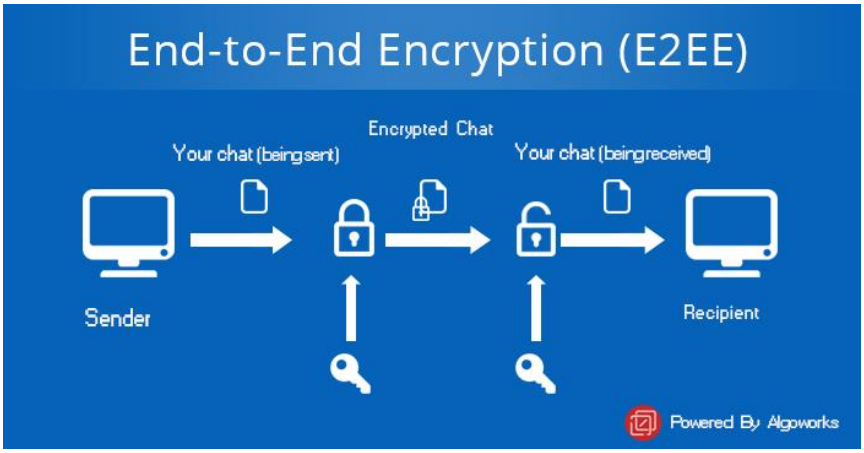
A continuación, se listan las mejores prácticas para implementar una arquitectura segura.

4.7.2.1. Codificación de extremo a extremo (E2EE)

Este protocolo permite encriptar la información mientras se envía o traslada a un servidor destinado. Es una capa de seguridad importante para prevenir la divulgación de información mencionada, brinda una alta seguridad en la protección de datos. Algunas herramientas son:

- *VeraCrypt*
- GNU *Privacy Guard*

Figura 8. **Codificación de extremo a extremo**



Fuente: Algoworks. *End To End Encryption (E2EE) – Secure Chats In Mobile Apps!,* https://www.algoworks.com/blog/end-to-end-encryption-secure-chats-in-mobile-apps/. Consulta: noviembre de 2019.

4.7.2.2. Autenticación y autorización de la identidad del usuario

Esta parte depende de la información confidencial que manejan los asistentes virtuales. Como principio se cuenta con un usuario y una contraseña. Las credenciales deben brindarse por un *token* de sesión.

Regularmente, las sesiones de los usuarios de bancos utilizan *tokens* de autorización por tiempo. Se recomienda usar estos *tokens* en caso de tener alguna limitante, como el tiempo de desarrollo o presupuesto. Existen otras alternativas como:

- Autenticación de dos pasos: el inicio de sesión debe estar asociado a un segundo dispositivo físico como un teléfono. El sistema envía un mensaje y este debe ser ingresado al iniciar la sesión.

- Mensajes por correo electrónico: el sistema envía un mensaje al correo electrónico del usuario y este puede verificar si es esa persona. Google es una empresa que utiliza este tipo de seguridad.

4.7.2.3. Mensajes autodestructivos

Esta parte debe utilizarse solo para los bancos y terceros que puedan tener acceso a la conversación virtual. De ninguna forma el sistema debe guardar la información y mostrarla ni siquiera en los servidores que se aloja el servicio. No se debe guardar la información, solo los *intent* (intenciones) y únicamente debe usarse para fines de auditoría.

4.7.3. Protocolos de seguridad

En la siguiente tabla se presentan los protocolos de red.

Tabla V. **Protocolos de seguridad en la red**

Protocolo	Definición
HTTPS	Garantiza la privacidad e integridad de los datos.
HTTP	Permite la trasferencia de los datos.
SSL o TSL	Permite la comunicación entre los servicios de una forma cifrada por medio de llaves públicas y privadas.

Fuente: elaboración propia.

4.8. Análisis de sentimientos

El análisis de sentimientos es una subrama del PLN (procesamiento del lenguaje natural). Es una herramienta opcional, pero a la vez muy poderosa. El análisis de sentimientos puede ser utilizado para medir la satisfacción del cliente por la compañía. Esto es reflejado ahora en las redes sociales, que son un producto valioso desde la perspectiva comercial.

Existen herramientas en API que ayudan a la compresión de sentimientos. Las más conocidas son:

- Amazon Comprehead
- Google Natural Leanguage

Estas API pueden ser implementadas de forma muy simples al código que se desarrolla.

El sarcasmo es un problema recurrente en el análisis de sentimientos. Existe una inteligencia artificial que se está dedicando a comprender cómo funciona; su nombre es DriverlessAI de H2O.

DriverlessAI viene equipado con guías de procesamiento del lenguaje natural (PLN) para la clasificación de texto y problemas de regresión. La plataforma admite texto independiente y texto con otros valores numéricos como características predictivas; además, cuenta con una biblioteca de entramiento de comentarios sarcásticos.

5. CHATBOT PILOTO

Este capítulo tiene como objetivo implementar un asistente virtual enlazando todos los conceptos e implementado un *Instant Messenger*, además del presupuesto para que el asistente virtual permanezca en producción y una arquitectura de microservicios elaborada en AWS. Esta sección contiene información sobre los precios de mantener una arquitectura con microservicios, y otra arquitectura con *serverless*.

5.1. Guía de implementación de un asistente virtual con una arquitectura de microservicios

Esta sección contiene todo lo relacionado con mantener servidores dedicados a la implementación de un asistente virtual y bots para respuestas complejas, además de un posible presupuesto de mantener una arquitectura como tal. También incluye la implementación de Rasa con Messenger de Facebook.

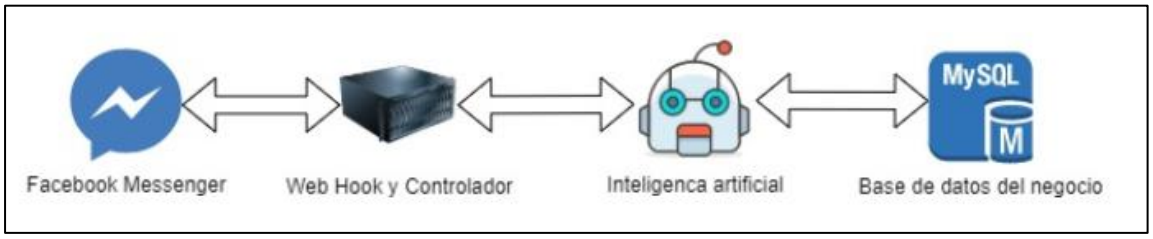
5.1.1. Elaboración de arquitectura para un asistente virtual

Se presenta una propuesta para la elaboración de una arquitectura de microservicios en AWS, enumerando los aspectos técnicos de este. Para la elaboración de la arquitectura se debe tomar en cuenta los niveles de comunicación entre los microservicios los cuales son:

- *Instant Messenger*
- *Webhook* y controlador de *instant Messenger*

- Inteligencia artificial
- Base de datos

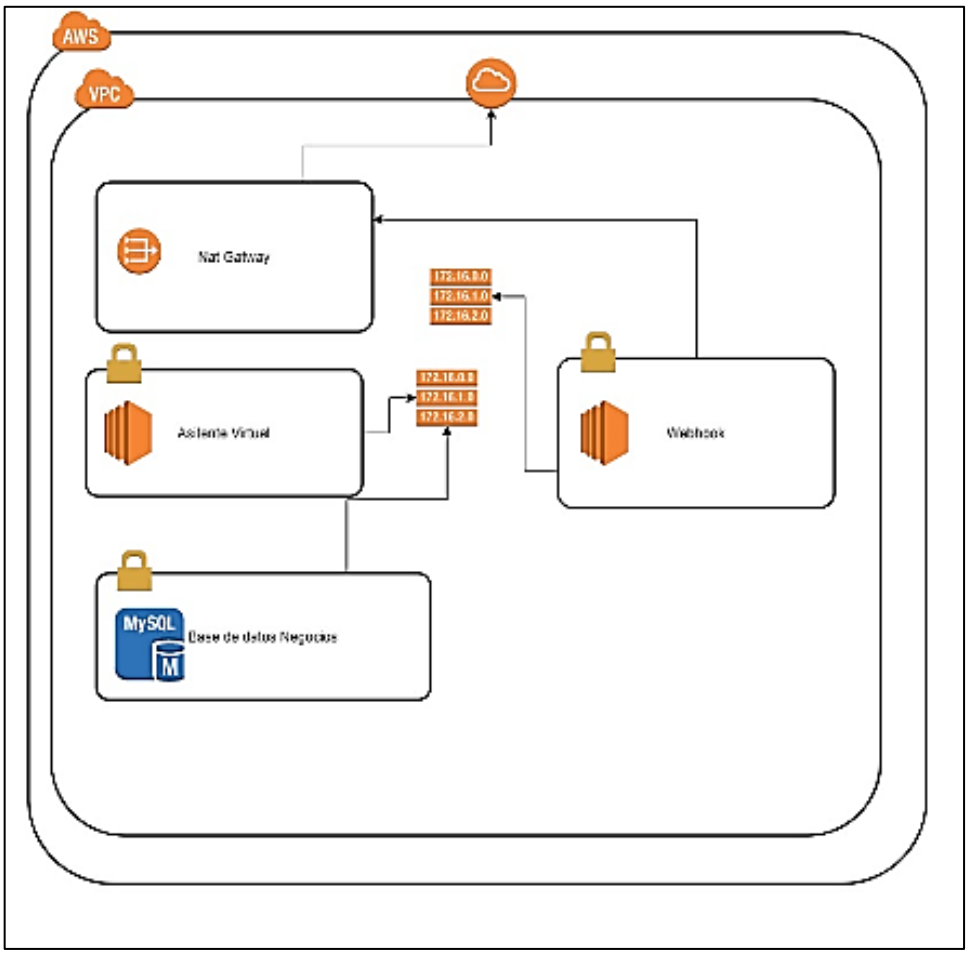
Figura 9. **Diagrama de comunicación de servicios**



Fuente: elaboración propia, empleando herramienta Draw.io.

En el diagrama de la figura 9 se muestra la comunicación de los microservicios. Se visualiza cómo elaborar la arquitectura que se muestra en la figura 10.

Figura 10. **Arquitectura propuesta en AWS**



Fuente: elaboración propia, empleando herramienta de Draw.io.

Debido a que es una buena práctica elaborar una red en la nube, se estableció crear una VPC (nube virtual privada, por sus siglas en ingles), la cual contiene cuatro subredes. En la primera capa de las subredes se encuentra la subred pública, que permite salir. En esta se encuentra el *natgatway* que funciona para ocultar las redes. En las redes privadas se encuentra dos grupos de autoescalado; en uno de ellos está el *webhook*, y en el otro la inteligencia artificial. Estos servicios de autoescalado ayudan a que, si existe alta demanda

de procesador o ram, automáticamente crean nuevas instancias de EC2 (computadoras elásticas en la nube, por sus siglas en inglés). Por último, se encuentra la base de datos del negocio, el cual puede cambiar. En esta es recomendable tener siempre tres bases de datos, una que es la principal, la segunda para una caída repentina y la última para una réplica de lectura, pero esto queda a criterio del desarrollador. Con esta arquitectura simple es posible construir un asistente virtual a prueba de fallos y seguro. Con esto se puede empezar a pedir permisos al *instant messenger*.

5.1.2. Crear un entorno de trabajo con Facebook Messenger

Se decidió utilizar Facebook Messenger debido a que es la red social con más usuarios activos y se utiliza como esencial en campañas de mercadeo. Para crear un entorno de trabajo en Facebook Messenger se debe cumplir varios requisitos previos, los cuales son:

- Contar con una cuenta de Facebook
- Contar con una Fan Page
- Contar con una cuenta de desarrollador de Facebook

Una vez que se cumplan se debe crear un entorno de trabajo.

Después de haber realizado los pasos anteriores se debe crear una aplicación, como se menciona a continuación:

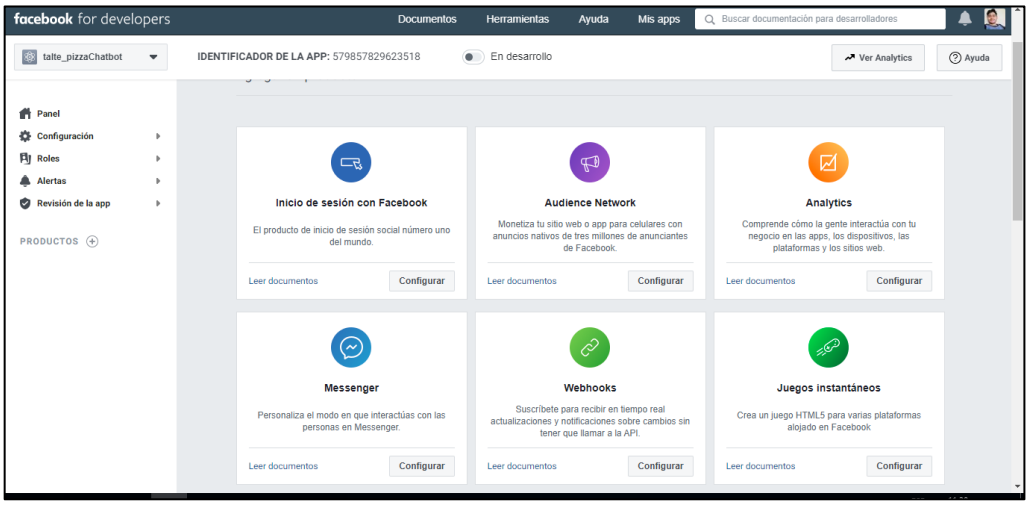
- Dirigir con un navegador a developers.facebook.com.

- Luego en el menú 'Mis apps' dar clic a 'crear nueva app'.

- Como buena práctica es necesario colocar el nombre de la industria en donde se trabaja seguido de la función y el nombre de la compañía donde se presta el servicio, como, por ejemplo, 'Talte_PizzaChatbot'.

- Luego se da clic en 'configurar' en la ficha de Messenger. En la figura 11 se muestra cómo se ve el menú.

Figura 11. **Menú de Facebook for developers**



Fuente: For developers. *Menú de Facebook*. https://developers.facebook.com/. Consulta: 20 de mayo de 2020.

Una vez finalizado el último paso se debe crear un *webhook* para comunicar el servidor *webhook* con *instant Messenger*

5.1.3. Crear un *webhook*

Un *webhook* es un servicio web que ayuda a la comunicación entre el *instant messenger* y el servidor en donde está alojado el asistente virtual.

La creación de un *webhook* es sumamente simple debido a que es un servicio web. En el apéndice 1 se muestra la creación del *webhook* en Nodejs utilizando la librería de *express* como servidor web.

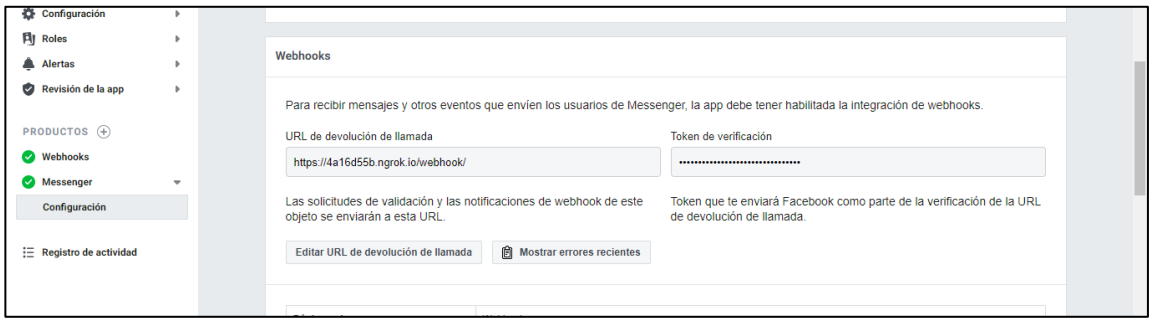
En este apartado tomaremos dos enlaces que permiten la comunicación entre el *instant Messenger* y el asistente virtual.

- Se utiliza una función /*webhook* de tipo GET que ayuda a verificar la seguridad de la comunicación entre el cliente de Facebook Messenger y nuestro servidor.

- Luego se utiliza una función /*webhook* de tipo POST que ayuda a la comunicación bilateral. Todo este código se encuentra en el apéndice 2.

Inmediatamente después de la creación de este servidor es posible realizar un enlace a Facebook Messenger en el apartado de crear un *webhook*. Luego se debe colocar la dirección pública en donde se encuentra nuestro dominio y el *token* que se asignó. En este caso se utiliza la función de GET del *webhook* para realizar esta comunicación. En la figura 12 se encuentra el apartado en donde se ingresa los datos del servidor anteriormente creado.

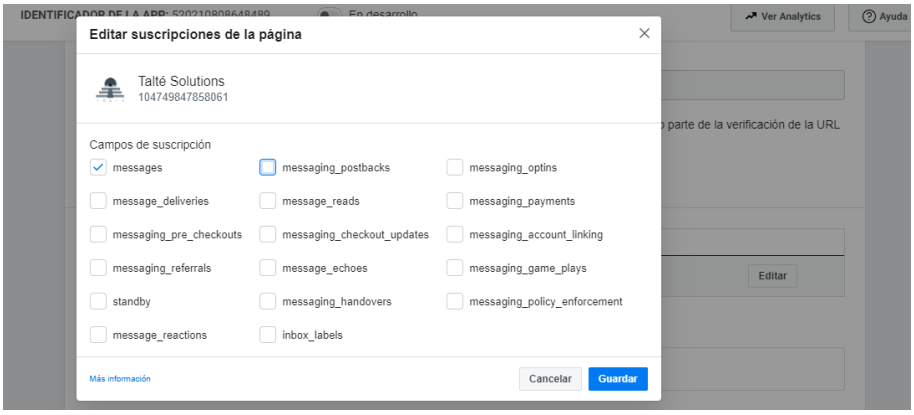
Figura 12. **Campos de suscripción de un servidor *webhook***



Fuente: For developers*. Menú de Facebook*. https://developers.facebook.com/. Consulta: 20 de mayo de 2020.

Una vez registrado el servidor webhook es importante aplicar los permisos necesarios para que pueda comunicarse y manipular la información. La figura 13 muestra las sucripciones de mensajes comunes en estos momentos.

Figura 13. **Campos de relleno de un servidor *webhook***



Fuente: For developers*. Menú de Facebook*. https://developers.facebook.com/. Consulta: 20 de mayo de 2020.

5.1.4. Crear un entorno con Rasa NLU

Para el asistente virtual es simple mantener en un contenedor un modelo entrenado. En el apéndice 3 se encuentra un archivo *Docker* para implementar Rasa en un contenedor. En la siguiente tabla se define los datos esenciales para construir un asistente virtual en Rasa.

Tabla VI. **Estructura de archivos para comprender**

__init.py	Archivo para el uso de las acciones
actions.py	Archivo para personalizar las acciones que hace el asistente virtual
config.yml '*'	Configuración de los modelos
credentials.yml	Conexión a *instant Messenger*
data/nlu.md '*'	Datos de entrenamientos
data/stories.md '*'	contexto y sentido a la conversación
domain.yml '*'	Respuestas del asistente virtual
endpoints.yml	Detalles de *conxion* a los *intant Messenger*
models/<timestamp>.tar.gz	Modelos creados después del entrenamiento

Fuente: Rasa. *Rasa tutorial*. https://rasa.com/. Consulta: 28 de noviembre de 2019

Rasa, por ser uno de los mejores modelos de entrenamiento, cuenta con distintas implementaciones como la creación de un *webhook*; esto puede evitarnos la creación de un paso anterior. En el apéndice 5 se muestra cómo se entrena un modelo con Rasa NLU.

5.1.5. Costo de la infraestructura en la nube

En la elaboración de la infraestructura es necesario establecer tres servicios para una pequeña empresa que tiene hasta 25 000 peticiones por mes.

5.1.5.1. Base de datos

Con el servicio de RDS se puede establecer tres tipos de costos monetarios, en dado caso se desea establecer un sistema unificado con la lógica de negocio. Los valores pueden estar entre los rangos.

Tabla VII. **Costos de una base de datos**

	T2.Small	**T3.Medium**	**M5.xlarge**
StandBy	No	No	Si
CPU	1	2	4
RAM	2	4	16
Disco	20 GB	20 GB	100 GB
Tipo de almacenamiento	Propósito General	Propósito General	*Provisioned* IOPS
Precio/mes	27,12 USD	51,94 USD	362,16 USD

Fuente: elaboración propia.

5.1.5.2. Caché

Para el caché se recomienda Redis. Pude ser opcional.

Tabla VIII. **Costos de una base de datos en caché**

	T3.Small
CPU	2
RAM	1,37
Disco	5 GB
Precio/mes	24,48 USD

Fuente: elaboración propia.

5.1.5.3. Servidor asistente virtual

Este servidor es donde se almacena la lógica de un asistente virtual elaborado con Rasa, parte que puede ser beneficiada con un grupo de autoescalado. Este grupo puede agregar más instancias si se satura el servidor.

Tabla IX. **Servidor de asistente virtual y *webhook***

	T2.Small
CPU	1
RAM	2
Precio/mes	16,56 USD c/u

Fuente: elaboración propia.

5.1.5.4. Balanceador de carga

Se estima que el costo puede ser de 24,00 a 100,00 USD /mes, ya que esta parte es bajo demanda del usuario.

5.1.5.5. Rango de precios

Para elaborar el presupuesto mensual se tomaron 2 T2.Smal de la tabla VIII con el fin de tener dos servidores, una T3.Small de la tabla X y el rango más bajo del balanceador de carga. Para elaborar el presupuesto alto se tomó los 100,00 USD del balanceador de cargas de 6 instancias, de las cuales 3 son el servidor *webhook* y las restantes en servidor principal. Además se tomó en cuenta el caché y la base de datos de precio más alto de la tabla X.

Tabla X. **Rango de precio**

Rango bajo	Rango Alto
60 USD/mes	500 USD/mes

Fuente: elaboración propia.

5.1.5.6. Costo en canal

Esto depende en dónde se desea que interactúe el asistente virtual. Depende de la interacción de los clientes. Un canal de WhatsApp está entre 5 USD a 15 USD. Otros canales son gratuitos.

5.2. Guía de implementación de un asistente virtual con una arquitectura *serverless*

En esta sección muestra cómo es implementada una arquitectura *serverless* con *Dialogflow*. Además, esta sección tiene como objetivo demostrar cuán simple es realizar un asistente virtual con la arquitectura *serverless* y sus ventajas.

5.2.1. Crear una infraestructura *serverless* con *Dialogflow*

Dialogflow es un servicio de Google muy simple de implementar. Lo primero es crear una cuenta de Google y dirigirse a https://dialogflow.com y en seguida iniciar sesión con la cuenta de Google.

Una vez iniciada debemos crear un agente. Se llama agente a todo proyecto creado en *Dialogflow*. En la figura 14 puede observar cómo es.

Figura 14. **Crear un agente en *Dialogflow***

Fuente: Google cloud. *Menú de Dialogflow*, https://cloud.google.com/dialogflow/docs/. Consulta: 20 de mayo de 2020.

Al crear el asistente es necesario configurarlo; para ello pondremos un nombre y el idioma de aprendizaje del asistente virtual.

5.2.2. Entrenar un modelo en *Dialogflow*

En la sección de *intents* (intenciones), es donde se entrena el asistente virtual con las posibles preguntas que puede realizar. En la sección de entidades el administrador coloca las palabras clave y únicas que pertenecen al negocio. Es posible realizar el mismo ejercicio del apéndice 5.

5.2.3. Implementar un canal *serverless* a Facebook Messenger

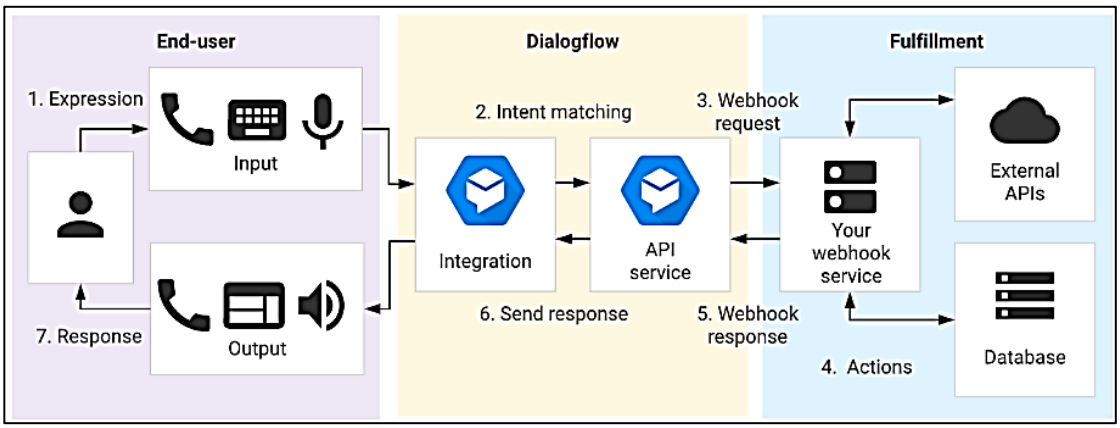
Es muy simple. En la sección de integración es posible elegir los canales de comunicación del asistente virtual. Para integrar el canal se utiliza el mismo procedimiento de la subsección 5.2.3 al crear un entorno de trabajo con Facebook Messenger.

5.2.4. Comunicación con otros servicios

Esta fase es importante para los desarrolladores, debido a que se incorpora la implementación de servicios de terceros en dado caso se requiera consultar en un base de datos.

La sección de *fulfillment* (entrega) tiene dos opciones, la de servicios *webhook* en donde *Dialogflow* pedirá respuesta a un servidor de una compañía con las respuestas más dinámicas y con alguna acción a otro servidor. Si se desea crear un servidor *webhook* puede guiarse en la sección anterior, o la sección de insertar Lindea de código, que es un *serverless* donde se construye la comunicación a partir de un código. Como esta sección tiene el objetivo de construir una arquitectura totalmente *serverless* se utiliza la sección de *Inline Editor*. En la figura 15 se indica cuál es el flujo que conlleva utilizar el servicio de entrega.

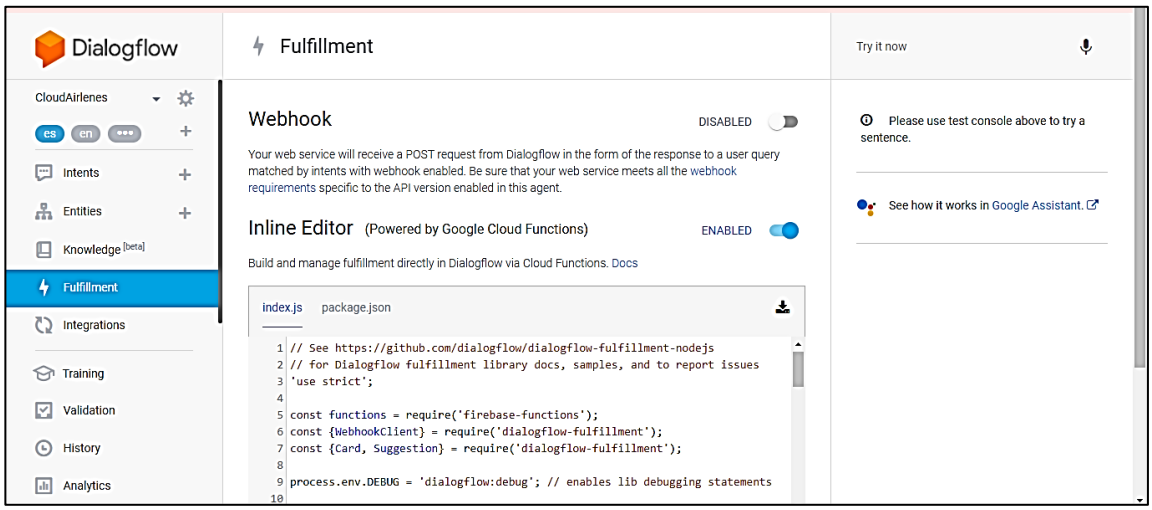
Figura 15. **Diagrama de flujo de entrega**



Fuente: M Cloud fulfillment. *Flujo de comunicación*. https://cloud.google.com/dialogflow/docs/fulfillment-overview, entrega. Consulta: 10 de junio de 2020.

Para habilitar *Inline Editor* (Insertar Linea de Código) se puede dirigir a *fulfillment* (entrega) y se habilita.

Figura 16. **Habilitar *Inline Editor***



Fuente: Google cloud. *Menú de Dialogflow*, https://cloud.google.com/dialogflow/docs/. Consulta: 20 de mayo de 2020.

5.2.5. Costo de infraestructura *serverless*

Una de las ventajas de una arquitectura *serverless* es que puede crecer en función de las necesidades del negocio.

5.2.5.1. Costo de *Dialogflow*

Dialogflow es muy accesible en el precio. Cuenta básicamente con dos paquetes: la edición estándar y la empresarial.

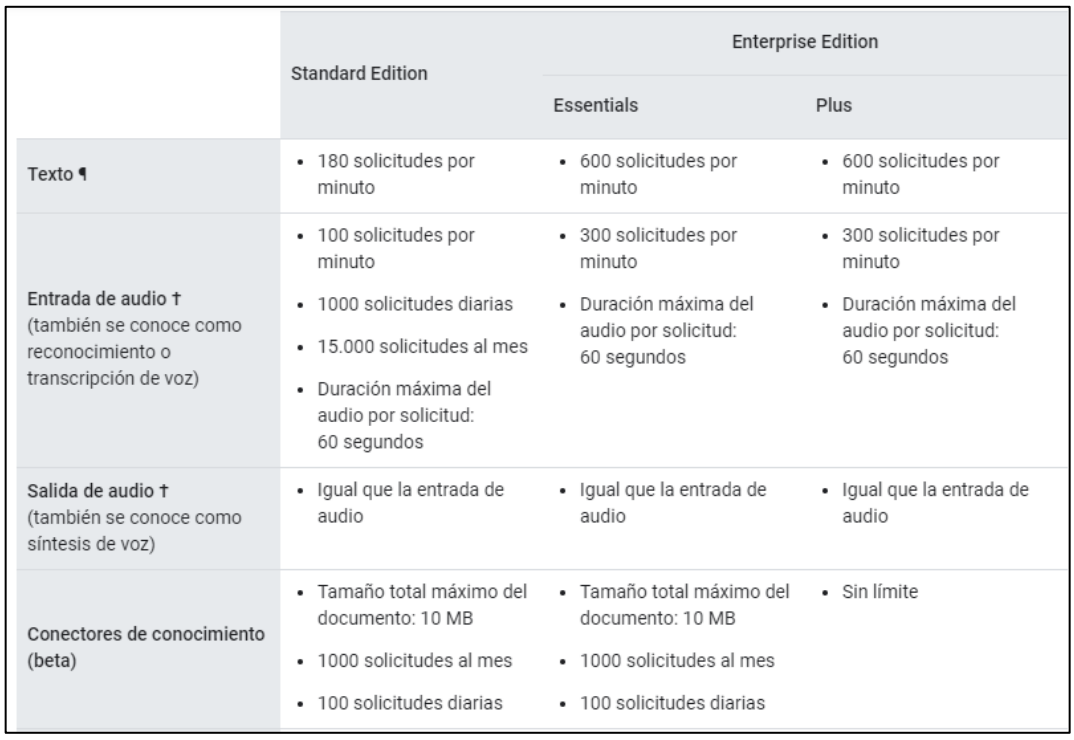
En la figura 17 se presentan los precios importantes para este trabajo. En la figura 18 se presenta las limitantes de los paquetes. En este caso hay 180 solicitudes por minuto en la versión estándar.

Figura 17. **Costos de uso *Dialogflow***

	Standard Edition	Enterprise Edition	
		Essentials	Plus
Texto	• Gratis*	• 0,002 USD por solicitud	• 0,004 USD por solicitud
Entrada de audio Incluye reconocimiento y transcripción de voz.	• Gratis*	• 0,0065 USD por 15 segundos de audio†	• 0,0085 USD por 15 segundos de audio†
Salida de audio Incluye síntesis y transcripción de voz.	• Gratis*	• Voces estándares: 4 USD por cada millón de caracteres • Voces de WaveNet: 16 USD por cada millón de caracteres	• Voces estándares: 4 USD por cada millón de caracteres • Voces de WaveNet: 16 USD por cada millón de caracteres
Conectores de conocimiento (beta)	• Gratis*	• Gratis	• Gratis

Fuente: Google cloud*. documentación de Dialogflow Precios*. https://cloud.google.com/dialogflow/docs/. Consulta: 10 de junio de 2020.

Figura 18. **Limitaciones de precios en *Dialogflow***



	Standard Edition	Enterprise Edition	
		Essentials	Plus
Texto ¶	• 180 solicitudes por minuto	• 600 solicitudes por minuto	• 600 solicitudes por minuto
Entrada de audio † (también se conoce como reconocimiento o transcripción de voz)	• 100 solicitudes por minuto • 1000 solicitudes diarias • 15.000 solicitudes al mes • Duración máxima del audio por solicitud: 60 segundos	• 300 solicitudes por minuto • Duración máxima del audio por solicitud: 60 segundos	• 300 solicitudes por minuto • Duración máxima del audio por solicitud: 60 segundos
Salida de audio † (también se conoce como síntesis de voz)	• Igual que la entrada de audio	• Igual que la entrada de audio	• Igual que la entrada de audio
Conectores de conocimiento (beta)	• Tamaño total máximo del documento: 10 MB • 1000 solicitudes al mes • 100 solicitudes diarias	• Tamaño total máximo del documento: 10 MB • 1000 solicitudes al mes • 100 solicitudes diarias	• Sin límite

Fuente: Google cloud*. documentación de Dialogflow Precios*. https://cloud.google.com/dialogflow/docs/. Consulta: 10 de junio de 2020.

Para esto se puede crear un escenario donde una persona haga un promedio de 50 solicitudes para realizar una compra; es decir, que aproximadamente podría atender a 12 clientes simultáneos por minuto. En un minuto gastaría 1,20 USD. En una hora, suponiendo que este evento ocurra 3 veces por hora, se estaría gastando 3,60 USD, con 36 clientes. En una jornada laboral podría atender a 288 personas diferentes; esto equivale a 28,8 USD. En un mes podríamos estimar que se gastaría 892,80 USD con 8 640 personas. Este cálculo se estimó gracias a la colaboración de una empresa que se dedica a vender utensilios de hogar. Se investigó que, para realizar una venta, un cliente interactúa con un agente en promedio con 50 peticiones antes de

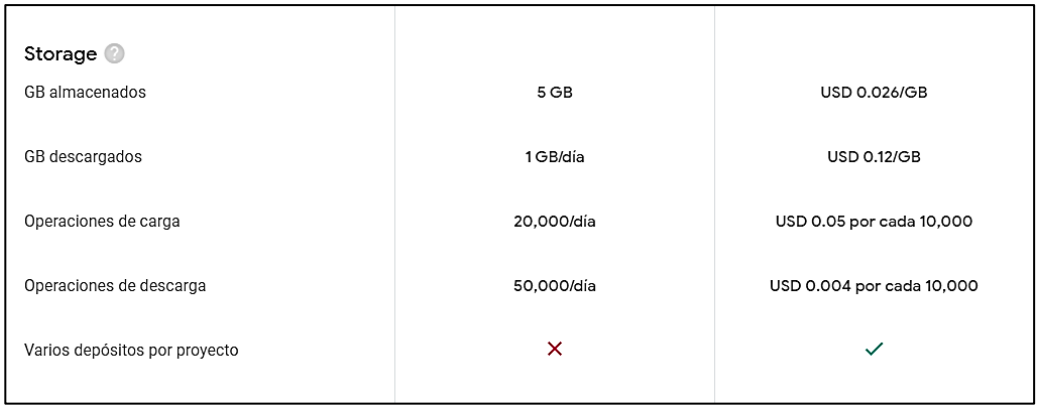
realizar una prueba. Si se desea agregar un poco más de complejidad como guardar datos se usará una base de datos.

5.2.5.2. Costos de *Firebase*

Firebase es una base de datos *serverless* que pertenece a Google. Para este ensayo se guardará la siguiente información de los clientes planteada en la sección anterior: nombre, apellido, número de teléfono, dirección de domicilio, el monto de la compra, junto a una lista de artículos.

Los cálculos para el tamaño de los *Sting* utilizan la codificación UTF-8, suponiendo que se guardaría estos datos para realizar una entrega. Cada encabezado guarda 8 bytes y la dirección, 256 bytes. Esto equivale a 264 bytes, en un mes con 8640 personas se almacenan 2 280.96 kilobyte; es decir, que no gastaríamos casi nada en almacenar información hasta posiblemente en 10 años. En la figura 19 indica los precios por gigabytes.

Figura 19. **Precios de *Firebase***



Storage		
GB almacenados	5 GB	USD 0.026/GB
GB descargados	1 GB/día	USD 0.12/GB
Operaciones de carga	20,000/día	USD 0.05 por cada 10,000
Operaciones de descarga	50,000/día	USD 0.004 por cada 10,000
Varios depósitos por proyecto	✕	✓

Fuente: Documentacion de Firebase*. Precios*. www.firebase.com. Consulta: 10 de junio de 2020.

5.3. Buenas prácticas al elaborar las respuestas de un asistente virtual y aspectos técnicos

Debido a lo simple que es crear un asistente virtual, es importante contar con un diferenciador para ello. Se implementará algunas buenas prácticas para entrenar un asistente virtual.

5.3.1. Conocer la marca

Crear un asistente virtual se ha convertido en una tarea simple, pero crear asistentes virtuales de forma genérica no aporta mucho a las empresas que ahora han cambiado las cuatro 'p' (producto, precio, punto de venta y promoción) por las 4 'c' (cliente, costo, conveniencia, comunicación) que están más enfocadas al cliente. En esta parte se recomienda conocer los siguientes aspectos para elaborar el asistente virtual:

- Conocer la propuesta de valor de la marca
- Qué soluciona el producto
- Características únicas del producto o servicio
- En qué fallan productos similares
- En qué podría fallar
- Conocer el público objetivo

Para conocer mucho más a la marca se sugiere elaborar un corpus con las palabras claves, que el asistente virtual puede responder y utilizar para mostrar la característica única de la marca. Un corpus de palabras es:

- Sustantivos
- Adjetivos

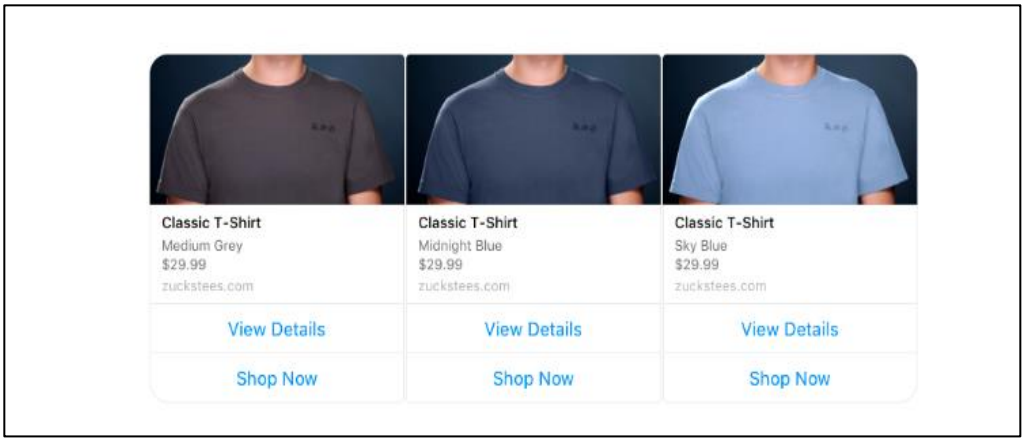
- Verbos
- Frases recurrentes
- Palabras que nunca se usarían
- Tópicos

Una vez conocida la marca hay atajos con los que los clientes pueden interactuar. Podemos ayudar a nuestro asistente virtual a comprender algunas peticiones de los clientes.

5.3.2. Crear atajos para los clientes

De alguna forma es mucho más simple guiar al usuario para encontrar lo que busca por medio de plantillas prediseñadas. Facebook ofrece cómo es posible implementarlas, lo cual es muy fácil.

Figura 20. **Plantilla de productos de Facebook**



Fuente: Documentación plataforma de Messenger. *Plantillas genéricas*. https://developers.facebook.com/docs/. Consulta: 16 de junio de 2020.

5.3.3. Agregar personalidad a las respuestas

Esta sección no es muy técnica, pero puede hacer la diferencia para los usuarios finales que interactúan con un asistente virtual. Para esto debemos mezclar el propósito, el tono y el carácter del asistente virtual y crear un patrón de respuestas. Para empezar, se muestran los propósitos que pueden ser:

- Educar
- Entretener
- Divertir
- Vender
- Comunicar

Con esto es posible empezar construir una personalidad al asistente virtual. Una vez elegida necesitamos un tono para comunicar el propósito. Algunos de estos tonos son:

- Corporativo
- Informal
- Cercano
- Demostrativo
- Testimonial
- Informativo

Junto ello se crea un carácter y alguna de estas características que puede crear es:

- Educativo
- Personal

- Emocional
- Directo
- Explicativo

Entonces alguna de estas combinaciones podemos crear una personalidad al asistente virtual utilizando la información de la marca o empresa. Uno de estos ejemplos puede expresarse como:

- Corporativo y formal, expresa y promueve los valores de la marca utilizando un lenguaje formal.

- Cercano e informal, expresa empatía con la audiencia y utiliza segunda persona.

- Demostrativo y con testimonio, pone como valor la calidad de sus servicios o productos y se apoya en clientes satisfechos.

- Con toque de humor, comunica con humor y gracia a las personas.

5.3.4. Entrenar a un asistente virtual

Para entrenar a un asistente virtual, se debe al menos plantear 10 posibles entradas para que este pueda comprender el contexto. Con el corpus mencionado y palabras clave es posible crear diez posibles entradas que haría un usuario. Además, en las entidades se debe colocar todo lo relacionado al producto. El ejemplo más practico fue un asistente virtual de vuelos, donde la entidad que definía su negocio era el código iata (*international Air Transport Association airport code*) que pertenece al lugar de destino. El ejemplo es de

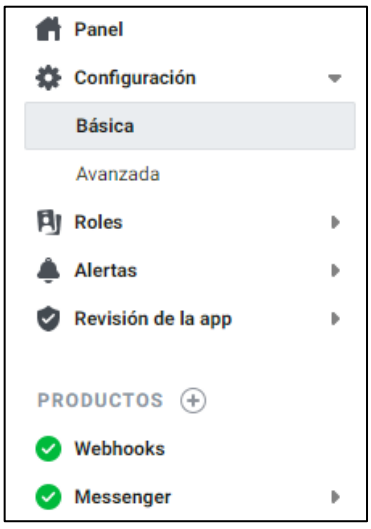
AJI que puede tener sinónimos como Agri o Turquía. Estas entidades pueden plantearlas más al conocer la marca.

5.3.5. Recomendaciones para mandar a producción un asistente virtual

La mayoría de los *instant messenger* requieren algunas revisiones para que un asistente virtual pueda trabajar para todo el público. En esta sección nos abocaremos a la documentación de Facebook Messenger para que pueda estar en producción.

Como primer paso se debe ir a la consola de Facebook y dirigirse a configuración básica. En la figura 21 se muestra en dónde se encuentra el menú.

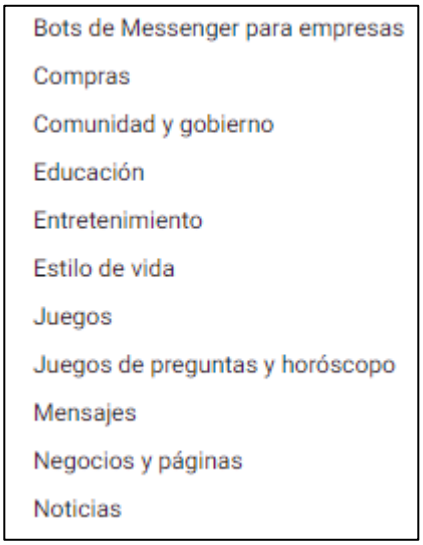
Figura 21. **Menú de configuración básica**



Fuente: Documentación plataforma de Messenger. *Plantillas.* https://developers.facebook.com/docs/. Consulta: 16 de junio de 2020.

Para mandar a producción un asistente virtual con el *instant Messenger* de Facebook debemos tener un icono de 1 025 x 1 025 pixeles. Este puede ser el logo de la empresa o el del asistente virtual; luego, se debe elegir una categoría. En la figura 22 se muestra las categorías de Facebook Messenger.

Figura 22. **Categorías de Facebook Messenger**



Fuente: elaboración propia, empleando captura de pantalla al configurar Facebook Messenger.

Por último, se debe tener en la página oficial de la empresa un documento de políticas de privacidad. Como buena práctica, el url debe estar estructurado de la siguiente manera: 'DOMINIO.com/term.html'. En algunos lados la llaman políticas legales. Las políticas de privacidad son un documento que debe estar público para todos los usuarios que visiten el sitio web; esta sección contiene toda la información sobre la forma como se recabará los datos personales de los usuarios y describe cómo son utilizados. Se debe redactar conforme a los usos reales y concretos de cómo se utilizan los datos personales recolectados.

A continuación, Acua Social Media brida los pasos para elaborar un documento:

- Qué información se va a recopilar (nombres, correos electrónicos, teléfonos, entre otros).

- Cómo se va a utilizar la información (para estadísticas, para mejorar la experiencia de compra o navegación, promociones, email marketing, entre otros).

- Qué se va a hacer con los datos recopilados.

- La posibilidad de modificar la política en un futuro.

- Forma de contrato (para modificaciones, actualizaciones o cancelaciones).

- Política de cookies.

- Ofrece información relevante sobre la forma que vas a proteger los datos.

Uno de los ejemplos de políticas de privacidad más extensas es de la aerolínea Aeroméxico. En el siguiente link es posible encontrar una redacción https://aeromexico.com/es-mx/informacion-legal. Se eligió esta porque Aeroméxico tiene uno de los asistentes virtuales más dinámicos.

Una vez finalizados los requisitos anteriores debemos solicitar la revisión de la aplicación por el equipo de Facebook. Para esto requiere unos pasos simples: primero, aceptar las políticas de Facebook; luego, redactar todos los

pasos que se requieren para el uso adecuado del asistente virtual. Por último, subir un video de toda la funcionalidad del asistente virtual, preferiblemente la pantalla del computador.

CONCLUSIONES

1. Una de las mejores herramientas para crear un asistente virtual es *Dialogflow* debido a su simplicidad de uso. Además, las *start up* pueden implementar un asistente virtual con una arquitectura *serverless*, ya que los costos de utilizar esta arquitectura se deben a la demanda de los servicios; es decir, si no se utiliza, no se cobra.

2. Se implementó una guía para elaborar un asiste virtual con las bases suficientes para que pueda ser desarrollado de la manera más rápida. La guía introduce conceptos teóricos sin muchos tecnicismos para que un desarrollador novato pueda crear un asistente virtual. Junto a ello se agregaron estilos de respuestas para que los asistentes virtuales puedan tener una personalidad interactiva con cualquier usuario, con base en los valores que una marca quiera transmitir.

3. La teoría involucrada en toda la tesis es una introducción a los aspectos técnicos de entrenamiento de cualquier herramienta para elaborar un asistente virtual. En las herramientas utilizadas al entrenar la inteligencia artificial están muy explícitos.

4. Siempre es importante pensar en el usuario. Dado que todas las herramientas se encuentran disponibles para su implementación personal, un desarrollador de asistentes virtuales debe velar por la interacción del usuario y facilitar un viaje de interacciones.

5. La mayoría de los aspectos técnicos se encuentran totalmente desarrollados en blogs. En el aspecto comercial no recomiendo construir un asistente virtual, debido a que muchas herramientas funcionan muy bien. Los costos de los asistentes virtuales pueden variar. En la experiencia de construir asistentes virtuales, se recomienda utilizar una arquitectura *serverless* debido que los costos son extremadamente rentables para cualquier empresa.

6. Elaborar un asistente virtual para los conocedores de la informática es mucho más sencillo gracias a la eficiencia de los computadores actuales que permiten procesar gran cantidad de información.

RECOMENDACIONES

1. Antes de la implementación de un asistente virtual complejo, se debe preguntar si el negocio cuenta con conversaciones suficientes para poder entrenar al asistente virtual.

2. No olvidar, realizar una auditoría de las preguntas y respuestas que interpreta el asistente virtual, por lo regular un humano interpreta de manera distinta algunas respuestas, debe analizar el contexto.

3. No hace falta construir una arquitectura compleja, para un asistente virtual, recuerde que existen los *dump chatbot.*

4. A la comunidad estudiantil, se recomienda el estudio de las siguientes tecnologías: GTP-3, capaz de crear párrafos con coherencia; Blender Bot, Un chatbot publicado por Facebook que es capaz de sostener conversaciones naturales y Open IA, puede ejecutar comando en consola en un lenguaje natural. La repercusión de esta inteligencia artificial podría sustituir, servicio al cliente, periodistas e inclusos programadores, la nueva revolución tecnológica ya se encuentra presente, los artículos fueron publicados en 2020.

BIBLIOGRAFÍA

1. Algoworks*. End To End Encryption (E2EE) – Secure Chats In Mobile Apps!* [en línea]. <https://www.algoworks.com/blog/end-to-end-encryption-secure-chats-in-mobile-apps/>. [Consulta: 15 de noviembre de 2019].

2. DATAHACK. *Desarrollo Avanzado de Chatbots: Bot Frameworks Programables.* [en línea]. <https://www.datahack.es/desarrollo-avanzado-de-chatbots-bot-frameworks-programables>. [Consultado: 21 de septiembre de 2019].

3. Dialogflow. *Terminos y condiciones.* [en línea]. <https://dialogflow.com/terms>. [Consulta: 21 de septiembre de 2019].

4. Emprende rioja. *¿Qué son y para qué sirven? ¿Cómo los están utilizando las empresas?* [en línea]. <http://emprenderioja.es/blog/2017/09/04/chatbots-sirven-los-estan-utilizando-las-empresas/>. [Consulta: 15 de septiembre de 2019].

5. HOPCROFT, John; MOTWANI, Rajeev y ULLMAN, Jeffrey. *Introducción a la teoría de autómatas lenguajes y computación.* 3a ed. Madrid España: Pearson, 2007. 458 p.

6. Planeta CHATBOT. *Conceptos básicos a tener en cuenta antes de construir tu chatbot.* [en línea]. <https://planetachatbot.com/conceptos-clave-para-construir-chatbots-fe67d977dfd>. [Consulta: 15 de septiembre de 2019].

7. _____. *Construyendo un Chatbot simple desde cero en Python (usando NLTK).* [en línea]. <https://planetachatbot.com/construyendo-un-chatbot-simple-desde-cero-en-python-usando-nltk-31b9ae4f71db>. [Consultado: 19 de septiembre de 2019].

8. _____. *Introducción al mundo ChatBot! - Planeta Chatbot: todo sobre los Chatbots, Voicebots e Inteligencia Artificial*. [en línea]. <https://planetachatbot.com/conceptos-clave-para-construir-chatbots-fe67d977dfd>.[Consulta: 15 de septiembre de 2019].

9. _____. *Sarcasmo.* [en línea]. <https://planetachatbot.com/inteligencia-artificial-para-an%C3%A1lisis-sentimientos-en-textos-e52782ad05e9>. [Consulta: 15 de noviembre de 2019].

10. _____. *Seguridad.* [en línea]. <https://planetachatbot.com/medidas-de-seguridad-con-chatbots-que-saber-7d99be58ca2d>. [Consultado: 08 de noviembre de 2019].

11. RUSSELL, Stuart. y NORVIG, Peter. *Inteligencia artificial un enfoque moderno.* 2a ed. Madrid, España: Pearson, 2004. 1170 p.

12. World conference*. Unlocking your serverless functions with OpenFaaS for AI chatbot projects.* [en línea]. <https://conferences.oreilly.com/oscon/oscon-or-2019/public/schedule/detail/76105>. [Consulta: 25 de septiembre de 2019].

APÉNDICES

Apéndice 1. **Crear un servidor web con nodejs y *express***

```
'use strict'
//librerias de para Node
const express = require('express');
const bodyParser =require('body-parser');
const request = require('request');
const access_token = ""

const app = express();
//Puerto en donde navega la informacion
app.set('port',5000);
//Se utiliza para dar formato json a una respuesta
app.use(bodyParser.json());
//bienvendida en la creacion de un servidor web
app.get('/',function(req,response){
  response.send('hola Mundo!');
});

/* codigo de implementacion*/

//Incializar servidor web
app.listen(app.get('port'),function(){
  console.log('Nuestro servidor esta funcionando en el puerto', app.get('p
ort'))
})
```

Fuente: elaboración propia.

Apéndice 2. **Funciones *webhook***

```
//Incializar comunicacion del Intant Messenger
app.get('/webhook',function(req,response){
  if (req.query['hub.verify_token']==='token de respuesta'){
    response.send(req.query['hub.challenge'])
  }else {
    response.send('Usted no tiene permisos para manejar este chatbot');
  }
})
//Comunicacion de usuario y chatbot
app.post('/webhook/', function(req, res){
  const webhook_event = req.body.entry[0];
  if(webhook_event.messaging) {
      webhook_event.messaging.forEach(event => {
          console.log(event);
        });
  }
  res.sendStatus(200);
});
```

Fuente: elaboración propia.

Apéndice 3. **Archivo para instalar Rasa en *Docker* archivo *Docker* file**

Docker brinda facilidad al implementar máquinas virtuales, en las siguientes líneas de código que contiene la estructura del archivo *Docker* file para implementar una máquina virtual con Rasa NLU.

```
FROM python
WORKDIR /App
ADD . /App
RUN git clone https://github.com/RasaHQ/rasa.git
RUN pip install -r requirements.txt
RUN pip install -e .
```

Fuente: elaboración propia

Apéndice 4. **Comandos de consola Linux con *Docker***

En esta sección del apéndice se elaboró una pequeña guía para poder implementar una máquina virtual con *Docker* utilizando como base el archivo del apéndice anterior utilizando la consola de Linux.

1) Debe dirigirse a la carpeta donde está el archivo de *Docker* file elaborado en el apéndice anterior. Luego, ejecutar en la consola para construir la imagen.

   docker build -t rasa .

2) Debe crear una carpeta en su sistema operativo donde se creará el proyecto.

   mkdir ~/Documents/Rasa

3) Construir contenedor dentro de la carpeta anterior e iniciar la Terminal de la maquina creada en *docker*.

   docker run -it -v ~/Documents/Rasa:/App --name rasa rasa /bin/bash

Fuente: elaboración propia.

Apéndice 5. **Crear proyecto con Rasa**

En esta sección del apéndice se elaboró para crear un proyecto con Rasa una vez ingresada a la máquina virtual del apéndice anterior.

1) Para crear un proyecto en Rasa se debe escribir la siguiente línea de código en la terminal de la máquina virtual elaborada anterior mente.

```
PYTHONIOENCODING='utf8' rasa init --no-promp
```

2) En el archivo data/nlu.md debe colocar los datos de entramiento los cuales el asistente virtual podrá comprender, las siguientes líneas de código elaboradas indican una conversación básica de una persona.

```
1) ## intent:greet
2) hola
3) Que tal
4) Buenos días
5) Buenas tardes
6) Buenas noches

7) ## intent:goodbye
8) Adiós
9) Gracias por todo
10) hasta luego
11) en otra ocasión
```

Continuación del apéndice 5.

12)## intent:affirm
13)si
14)por supuesto
15)genial
16)suena bien

17)## intent:deny
18)no
19)nunca
20)no me gusta
21)nel

22)## intent:mood_great
23)perfecto
24)muy bien
25)genial

26)## intent:bot_challenge
27)Eres un robot?
28)eres un humano?
29)estoy hablando con un robot?
30)estoy hablado con un human

3) Archivo domain.yaml, contienen la información de las respuestas del asistente virtual que deberá realizar al escuchar las peticiones del usuario. En las siguientes líneas de código se visualiza un ejemplo.

Continuación del apéndice 5.

1) intents:
2) greet
3) goodbye
4) affirm
5) deny
6) mood_great
7) mood_unhappy
8) bot_challenge

9) actions:
10) utter_greet
11) utter_cheer_up
12) utter_did_that_help
13) utter_happy
14) utter_goodbye
15) utter_iamabot

16) templates:
17) utter_greet:
18) text: "Hola como estas?"

1) utter_did_that_help:
2) text: "puedo ayudarte en algo?"

3) utter_happy:
4) text: "Esta bien"

Continuación del apéndice 5.

5) utter_goodbye:
6) text: "adios"

7) utter_iamabot:
8) text: "si soy un robot"

La estructura del archivo es explicada a continuación, con el fin de comprender de forma más clara la estructura de las líneas de código anteriores.

intents	Lo que se espera que el usuario diga
action	Lo que se espera que el asistente haga
template	Lo que se espera que el asiste responda

4) Para entrenar el asistente debe ejecutar la siguiente línea de código en la Terminal de la máquina virtual.

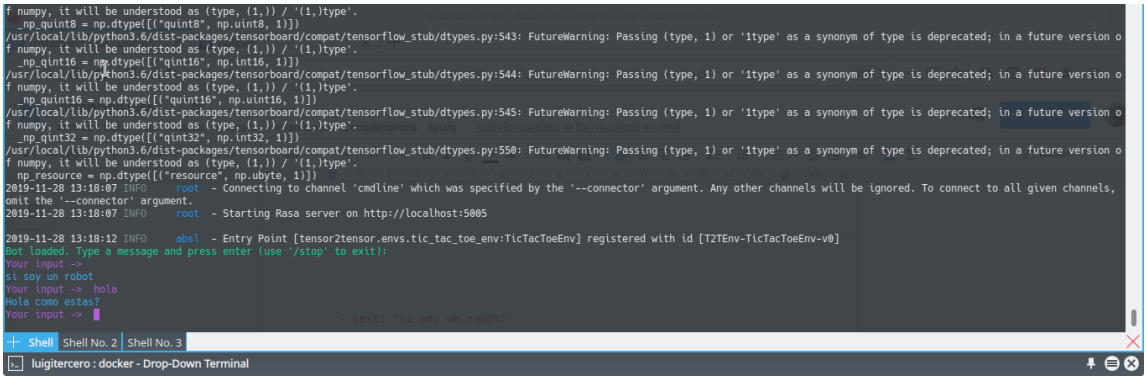
```
rasa train
```

5) Para probar el asistente virtual debe ejecutar la siguiente línea de código

```
rasa shell
```

Continuación del apéndice 5.

6) En la siguiente imagen muestra como el asistente virtual está funcionando.



Fuente: elaboración propia.