



**UNIVERSIDAD  
DE ANTIOQUIA**

**CHATBOT COMO HERRAMIETA DE SOPORTE  
PARA CONSULTAS DE USUARIOS CHOUCAIR.**

Autor(es)  
Gustavo Lozada Fuentes

Universidad de Antioquia  
Facultad de Ingeniería, Programa de Bioingeniería  
Medellín, Colombia  
2020



Chatbot como Herramienta de Soporte para Consultas de Usuarios Choucair

**Gustavo Lozada Fuentes**

Informe de práctica presentado  
como requisito parcial para optar al título de:

**Bioingeniero.**

Asesores (a).

Ángel Fernando Rey Largo, Ingeniero de Sistemas.

Universidad de Antioquia

Facultad de Ingeniería, Programa de Bioingeniería.

Medellín, Colombia

2020.

## Contenido

### Pág.

Resumen .....	5
Introducción .....	6
Objetivos .....	8
General.....	8
Específicos.....	8
Marco Teórico .....	8
¿Que son los Chatbots o Asistentes Virtuales? .....	8
Arquitectura .....	9
Asistentes virtuales. ....	9
Plataforma DialogFlow Essentials.....	10
Agentes: .....	10
Intents.....	10
Frasas de entrenamiento.....	11
Acción:.....	11
Entidades y parámetros .....	11
Respuesta .....	11
Fullfilment y Webhook: .....	12
Funcionamiento. A continuación, se resume el funcionamiento descrito anteriormente: ....	12
Chatbots en la industria médica. ....	13
Metodología .....	15
Proveedores para desarrollo de chatbots. ....	16
Prueba de Chatbot.....	17
Integración en el entorno corporativo.....	17
Flujo de conversación y lógica del negocio.....	17
Resultados y análisis .....	18
Conclusiones .....	21
Referencias Bibliográficas .....	22

## Lista de figuras

### Pág.

Figura 1. Arquitectura de un agente virtual de voz.....	10
Figura 2. Plataforma de desarrollo de bots de Google: DialogFlow.....	13
Figura 3. Arquitectura de un Chatbot en telemedicina.....	14
Figura 4. Metodología sugerida para el desarrollo del chatbot.....	16



## Lista de tablas

Pág.

Tabla 1. Principales características de las plataformas evaluadas. ....	15
Tabla 2. Sitios web corporativos de la empresa Choucair Testing. ....	20



## **CHATBOT COMO HERRAMIENTA DE SOPORTE PARA CONSULTAS DE USUARIOS CHOUCAIR.**

### **Resumen**

El acceso a la información y la consulta adecuada de los servicios y productos ofrecidos por una empresa es un pilar fundamental en la enseñanza de la filosofía corporativa y capacitación continua de sus empleados. Sin embargo, no siempre es fácil centralizar la información y acceder a ella de forma ágil. Durante esta investigación se logró desarrollar un chatbot para apoyar los procesos de consulta de información para empleados y colaboradores de la compañía Choucair Testing S. A. Para desarrollar este sistema, se utilizó la plataforma DialogFlow como capa de inteligencia artificial que ayuda en el entendimiento del lenguaje natural, además se utilizó Node.js como framework de desarrollo para consultar la información vía Apis y en formato JSON. Finalmente, la interfaz conversacional del chatbot se integró a través del canal DialogFlow Messenger que facilita su visibilidad desde un sitio web corporativo.

**Palabras clave:** Chatbot, DialogFlow, Node.js, JSON.

## Introducción

Las habilidades cognitivas basadas en inteligencia artificial (IA) como el procesamiento del lenguaje natural (NLP), el reconocimiento y la síntesis de voz, son elementos constitutivos de un sistema más complejo conocido como chatbot. Este tipo de sistemas, basados en inteligencia artificial representan hoy en día múltiples beneficios para los diversos sectores de la sociedad donde son implementados. Por ejemplo, reducen los tiempos en los procesos de consulta y entrega de información, proporcionan atención personalizada basada en lenguaje natural, reducen los costos en atención al cliente y cuentan con disponibilidad 24/7 (Fernández, 2019).

Los chatbots representan una forma diferenciada a través de la cual los usuarios interactúan con datos y servicios de una compañía de forma rápida (Brandtzaeg & Følstad, 2017). Sin embargo, esta no es una herramienta nueva y desde la década de los 60' son implementados como soluciones tecnológicas. Aunque la idea sigue siendo la misma, la facilidad de implementarlos en nuestros días radica en factores como diversos lenguajes de programación, plataformas de desarrollo ampliamente disponibles y opciones de despliegue a través de plataforma como un servicio (PaaS) (Radziwill & Benton, 2017). Por otro lado, la capacidad de computación, los grandes volúmenes de datos y el mejoramiento de los algoritmos de Machine Learning (ML) han permitido que el entrenamiento de estos sistemas sea más rápido (Fernández, 2019).

En adición a lo anterior, hoy el uso e integración de chatbot en Websites y Apps está estimulado por el impulso reciente en áreas como la inteligencia artificial y el aprendizaje automático (Brandtzaeg & Følstad, 2017). En este sentido, los chatbots son programados para interactuar según cada intención del usuario y adaptados para que adquieran un aprendizaje continuo después de cada predicción. Además de esto, los chatbots son multiplataforma y

pueden ser aplicados en campos como el académico para brindar soporte en una segunda lengua, como el lúdico para (redes sociales, juegos) y como asistentes virtuales basados en FAQ (Frequently Asked Questions) (Gutiérrez, 2019). Otro de los campos donde son ampliamente utilizados es en el sector empresarial, donde se brindan soporte para la atención al cliente y respuestas rápidas a las consultas de los usuarios sobre un determinado servicio, esto conlleva a la automatización de procesos que generan beneficios como: el acceso rápido a la información de la empresa y la reducción de costos operativos del personal (Rísquez-Almodóvar, 2019).

En el sector médico, existen diferentes estudios y aplicaciones que abarcan la importancia de implementar estos sistemas como una herramienta de apoyo para la atención y orientación médica remota (Telemedicina). Un estudio reciente detalla el proceso de construcción de un chatbot llamado MedBot, el cual permite a los pacientes obtener recomendaciones y diagnósticos preliminares en base a los síntomas que el paciente tiene y proporciona al chatbot (Rosruen & Samanchuen, 2018). Así mismo, otro estudio resalta la importancia que tiene la atención médica domiciliaria asistida por comandos de voz, la cual resulta en una alternativa novedosa para monitorizar el estado de salud de pacientes de la tercera edad o en condiciones de discapacidad. Teniendo en cuenta lo anterior, se hace necesario desarrollar un chatbot que permita apoyar los procesos de consulta de información de los empleados y colaboradores de la compañía Choucair Testing S. A. La implementación de este sistema permite optimizar los tiempos de acceso a los grandes volúmenes de información manejados por la empresa. Finalmente, este sistema podría ser replicado para permitir a los usuarios del sector salud acceder más rápidamente a los servicios médicos, apartar citas médicas o consultar información diagnóstica.

## Objetivos

### General

- Diseñar un chatbot que permita atender las solicitudes de consulta de información de los usuarios y/o empleados de la compañía Choucair Testing.

### Específicos

- Seleccionar la plataforma de desarrollo de chatbots que mejor se ajuste a las necesidades de la empresa.
- Desarrollar un prototipo preliminar que cumpla las expectativas iniciales de interactividad y captura de intenciones.
- Integrar el chatbot dentro del ecosistema Choucair.
- Desarrollar un Backend que permita controlar la interacción usuario-Bot, según la información consultada en las Apis de la empresa (Wikichoucair, Moodle y SharePoint).

## Marco Teórico

### ¿Que son los Chatbots o Asistentes Virtuales?

Los chatbots han sido definidos como algoritmos informáticos con capacidad de realizar procesamiento del lenguaje natural (*NPL*) a través de un sistema de preguntas y respuestas (*QA systems*) (Rodríguez et al., 2014). Inicialmente estos eran desarrollados con la finalidad de entretener o brindar un soporte guiado a los usuarios dentro de un juego o programa informático. Sin embargo, su uso fue muy restringido debido a varias razones de ámbito tecnológico, por ejemplo, la ausencia de interfaces gráficas de usuario (websites o apps móviles), donde se pudieran integrar con facilidad y atender las peticiones de un usuario u ofrecerle un determinado servicio. Adicionalmente, tenían un dominio restringido de interactividad con el usuario, ya que se usaban técnicas simples de concordancia de palabras claves ingresadas por el usuario para generar una salida coherente, lo cual limitaba los

escenarios conversacionales o terminaban pronto la comunicación Usuario–Bot. (Shawar & Atwell, 2007).

## **Arquitectura**

A través de los años los usos de chatbots se han diversificado de acuerdo con las necesidades e interés de un usuario. A continuación, se presenta la arquitectura general de un chatbot:

**Asistentes virtuales.** Son implementados desde 1990. Entre sus principales usos están los sistemas de información para viajes en aerolíneas (*ATIS*), por sus siglas en inglés y en tiendas de e-commerce. Estos agentes están diseñados para comprender el lenguaje natural y realizar una serie de tareas específicas para el usuario tales como proveer información de la aerolínea a los turistas, información de vuelo, registrar datos personales y agilizar la venta de productos. En cuanto a su arquitectura, los agentes virtuales contienen los siguientes elementos: Automatic speech recognizer (**ASR**), spoken language understanding module (**SLU**), dialog manager (**DM**), natural language generator (**NLG**), text-to-speech synthesizer (**TTS**) (Figura 1).

El ASR toma la señal de audio sin procesar y las transcribe en hipótesis de palabras, luego son pasadas al SLU, donde se captura la semántica central de dichas palabras las cuales reflejan la intención del usuario. El DM comprueba si se cumplen los requisitos de una intención y decide las acciones a realizar sobre una base de datos, una vez hecho esto el NLG, genera una respuesta en cadenas de texto que son pasadas al sintetizador TTS para entregar el resultado al usuario en forma de voz (Shum et al., 2018). Cabe resaltar que los chatbots pueden permitir diferentes tipos de Input como texto o voz y devolver como output texto o voz, esto dependerá de la finalidad del chatbot.

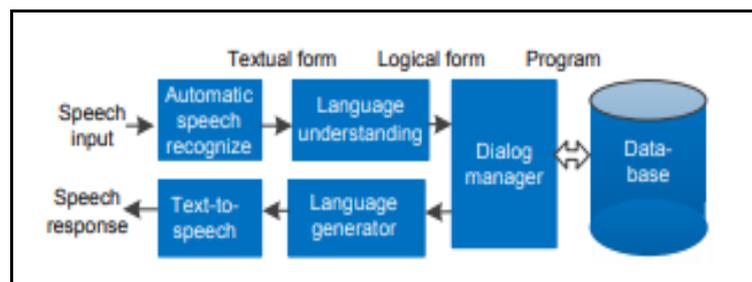


Figura 1. Arquitectura de un agente virtual de voz. Tomado de Shum et al. (2018).

### Plataforma DialogFlow Essentials.

DialogFlow Essentials es un servicio Google diseñado para facilitar la construcción de agentes conversacionales en base a una alta capacidad para procesar el lenguaje natural.

También permite su integración y uso en diferentes canales de comunicación como el asistente de Google de los dispositivos Android, un sitio web particular, dispositivos de IoT, tecnologías de IVR (Interactive Voice Response) y redes sociales como Twitter, Facebook Messenger, entre otras ("Dialogflow Documentation | Google Cloud", s.f.).

**Agentes:** Los agentes de Dialogflow son considerados como sistemas de IA con capacidad de recibir, procesar, interpretar y generar lenguaje natural durante la interacción con los usuarios finales. Durante la conversación el agente se encargará de manejar los requerimientos de cada consulta hecha por el usuario manteniendo una comunicación natural a través de entrada/salida de texto o voz. ("Agents | Dialogflow ES | Google Cloud", s.f.).

**Intents:** Cada agente de DialogFlow consiste en una serie de intenciones que enlazan el objetivo de cada consulta hecha por el usuario mediante el uso de una serie de frases de entrenamiento. Estas frases de entrenamiento determinan la coincidencia o no de un intent con el objetivo inmerso de cada consulta realizada por el usuario ("Intents | Dialogflow ES | Google Cloud", s.f.). La cantidad máxima de intents que se pueden configurar en cada agente es de 2000 intents y 2000 frases de entrenamiento por cada intent y por idioma ("Quotas and limits | Dialogflow | Google Cloud", s. f). En cada intents pueden ser configurados una serie

de elementos como las frases de entrenamiento, la acción y la entidad que ayudan a gestionar la información proporcionada por el usuario

**Frases de entrenamiento:** Son ejemplos de lo que el usuario podría decirle al agente, y por ende, permiten asociar una solicitud del usuario a un intent. Sin embargo, no es necesario definir un gran número de frases de entrenamiento por cada intent ya que el aprendizaje automático que existe en DialogFlow permite extender la lista, es suficiente establecer un número entre 10 a 20 frases ("Training phrases | Dialogflow ES | Google Cloud", s. f.).

**Acción:** Es un campo de texto de uso no obligatorio que se puede configurar en cada intención, se puede utilizar para determinar que fracción de la lógica debe ejecutarse una vez un intents coincida con una frase del usuario ("Actions and parameters | Dialogflow ES | Google Cloud", s. f.).

**Entidades y parámetros:** Las entidades representan el tipo de valor que almacena un parámetro en específico, por ejemplo, en DialogFlow existen algunas entidades predefinidas como sys.geo-city que indica que el parámetro solo deberá almacenar ciudades o sys.given-name para almacenar nombres, etc. Estas entidades también pueden ser creadas desde cero, asociándole una serie de valores comunes entre sí, en todos los casos son usadas para almacenar información relevante del usuario y personalizar las respuestas ("Actions and parameters | Dialogflow ES | Google Cloud", s. f.).

**Respuesta:** En cada intent pueden ser configuradas diferentes tipos de respuestas las cuales son mostradas al usuario una vez el intent sea llamado. Las más comunes de implementar son respuestas tipo texto que pueden ser máximo 30 por intent, devolviendo una de forma aleatoria por cada petición realizada ("Quotas and limits | Dialogflow | Google Cloud", s. f.). Sin embargo, otros tipos de respuesta están disponibles para ser configuradas y dependen principalmente de la plataforma de integración que sea seleccionado en el intent, por ejemplo, el asistente de Google permite generar tarjetas, sugerencias, tablas que permiten

personalizar la información a mostrar. En los casos antes mencionados las respuestas son estáticas ya que no contienen información externa, asociada consultas a bases de datos o Apis, pero pueden personalizarse con nombre, lugares y fechas que el usuario ingresa y que se almacenan mediante las entidades y parámetros del intent. ("Responses | Dialogflow ES | Google Cloud", s.f). Para dinamizar las respuestas es necesario habilitar la entrega o fullfilment en cada intent.

**Fullfilment y Webhook:** Por lo general un agente de DialogFlow responde a una coincidencia de intent con respuestas estáticas. En este sentido, para establecer respuestas dinámicas enriquecidas es necesario habilitar *la entrega* en cada intención. Si el intent no tiene la entrega habilitada responderá de acuerdo con las respuestas configuradas en la misma plataforma. Por el contrario, si la entrega o fullfilment está habilitada el agente responderá a un llamado de intent realizando una petición a un servicio externo vía webhook. ("Fulfillment | Dialogflow ES | Google Cloud", s.f.). Este servicio externo consiste en un Backend habilitado para recibir solicitudes HTTPs Request de tipo Post y enviar respuestas HTTPs Response, ambas en formato JSON. Este sistema de comunicación tiene algunas restricciones en DialogFlow, por ejemplo, la ruta al servicio webhook debe ser publica y debe administrar las solicitudes según el protocolo HTTPS, HTTP no es compatible ("Fulfillment | Dialogflow ES | Google Cloud", s.f.).

**Funcionamiento.** A continuación, se resume el funcionamiento descrito anteriormente:

1. El usuario final escribe o dice una expresión.
2. DialogFlow hace coincidir la expresión del usuario final con un intent y extrae los parámetros.
3. Si la entrega está habilitada en el intent, DialogFlow envía un mensaje de solicitud de webhook a tu servicio de webhook. Este mensaje contiene información sobre el intent

coincidente, la acción, los parámetros y la respuesta definida para ese intent en formato JSON. Si no, el agente responderá de acuerdo con la respuesta predefinida (estática).

4. El servicio realiza peticiones según sea necesario como consultas a las bases de datos o llamadas externas a la API.
5. El servicio envía un mensaje de respuesta de webhook a DialogFlow. Este mensaje contiene la respuesta que se debe enviar al usuario final, pero en formato JSON.
6. DialogFlow transforma la información en lenguaje natural y envía la respuesta al usuario final.
7. El usuario final ve o escucha la respuesta. ("Fulfillment | Dialogflow ES | Google Cloud", s.f.) (Figura 2).

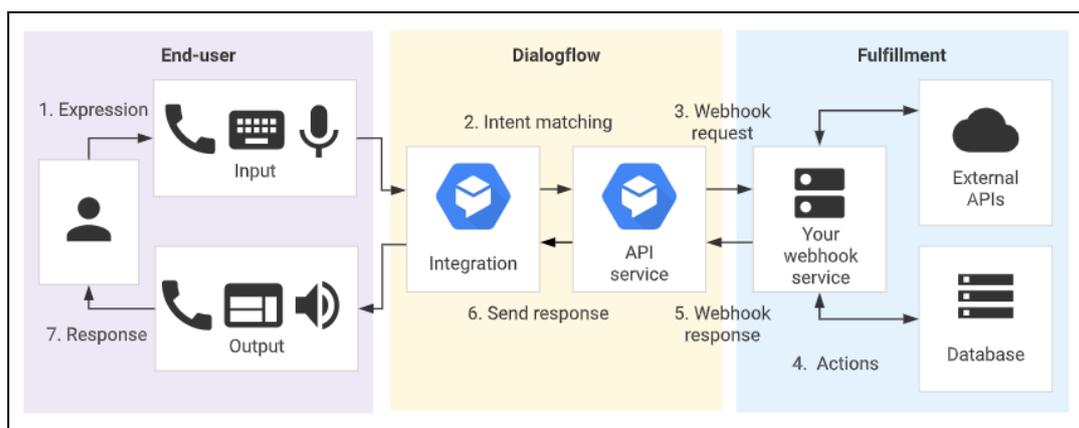


Figura 2. Plataforma de desarrollo de bots de Google: DialogFlow. Tomado de Fulfillment | Dialogflow ES | Google Cloud, s.f.

**Chatbots en la industria médica.** La idea de implementar asistentes virtuales para apoyar los servicios médicos representa una ventaja en algunos casos, por ejemplo, el usuario puede realizar cualquier consulta personal relacionada con la atención médica a través del chatbot sin estar físicamente disponible en el hospital y usando lenguaje natural. A su vez, el usuario pueda recibir un diagnóstico preliminar o recomendación en cuestión de segundos, esto implica una atención oportuna a partir de la cual el usuario puede decidir

qué acciones tomar. Recientemente Fadhil (2018), propone una arquitectura de chatbot en el campo telemedicina. Este sistema actúa como un asistente médico para ayudar a pacientes de edad avanzada que viven en zonas rurales a continuar su proceso de recuperación una vez el médico le ha dado de alta. Esta arquitectura busca integrar fuentes de información como bases de datos y API REST con información médica como ApiMedi, que proporcionan al usuario información útil acorde a sus síntomas. En la Figura 3, se detalla la conexión del sistema completo.

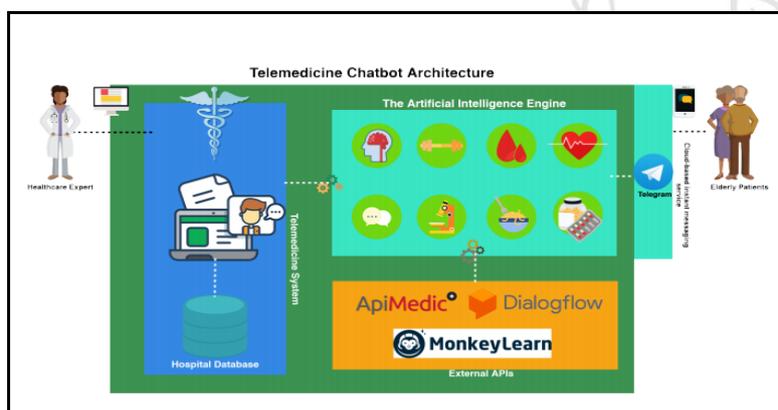


Figura 3. Arquitectura de un Chatbot en telemedicina. Tomado de Fadhil (2018).

En el 2018, Rosruen & Samanchuen realizaron un estudio donde mencionaron que un MedBot es un sistema que le permite al usuario ingresar los síntomas relacionados con su estado actual de salud y proporcionar un diagnóstico preliminar a partir del cual el usuario decide si es necesario consultar un experto o realizar un autocuidado desde casa. Para identificar los síntomas del paciente, el chatbot utiliza la tecnología de desarrollo de chatbots de Google conocida como DialogFlow, la cual se encarga del NLP para capturar los síntomas como objetivo principal (intención) y generar posteriormente una respuesta al paciente o realizar consultas externas a bases de datos mediante un Backend. En este caso, MedaBot es capaz de reconocer 16 síntomas a partir de los cuales el chatbot toma las decisiones acerca de que responderle al usuario. Por su parte Ilievski et al. (2019) plantean que es una prueba de concepto sobre la atención médica desde el hogar asistida por

lenguaje de voz mediante la cual los pacientes de avanzada edad o con discapacidades para escribir en un teclado o manejar abiertamente un smartphone, realizan consultas de información, programan citas y recuerdan terapias a través de simples comandos de voz. Este sistema médico consta de 3 componentes esenciales: la aplicación de voz, el servicio web y la nube médica. La aplicación de voz a través del dispositivo o teléfono inteligente con voz habilitada interactúa con el paciente al recibir comandos de voz, cada comando de voz configurado activa una acción en el servicio web que ejecuta una fracción de código, la cual se encarga de realizar una consulta de información en la nube médica (Ilievski et al., 2019). En particular, El asistente de Google o Alexa de Amazon, son usados como aplicación de voz. A futuro esperan poder verificar la condición cardiaca del paciente y enviar un informe detallado sobre la condición cardiaca a un médico haciendo uso de ellos.

### **Metodología**

Para el desarrollo de esta investigación se implementó el siguiente diagrama de flujo (Figura 4). Este diagrama incluye etapas que van desde la revisión bibliográfica de los principales proveedores de desarrollo hasta el despliegue del chatbot y los ajustes que son necesarios para mejorar la interacción usuario- BOT.

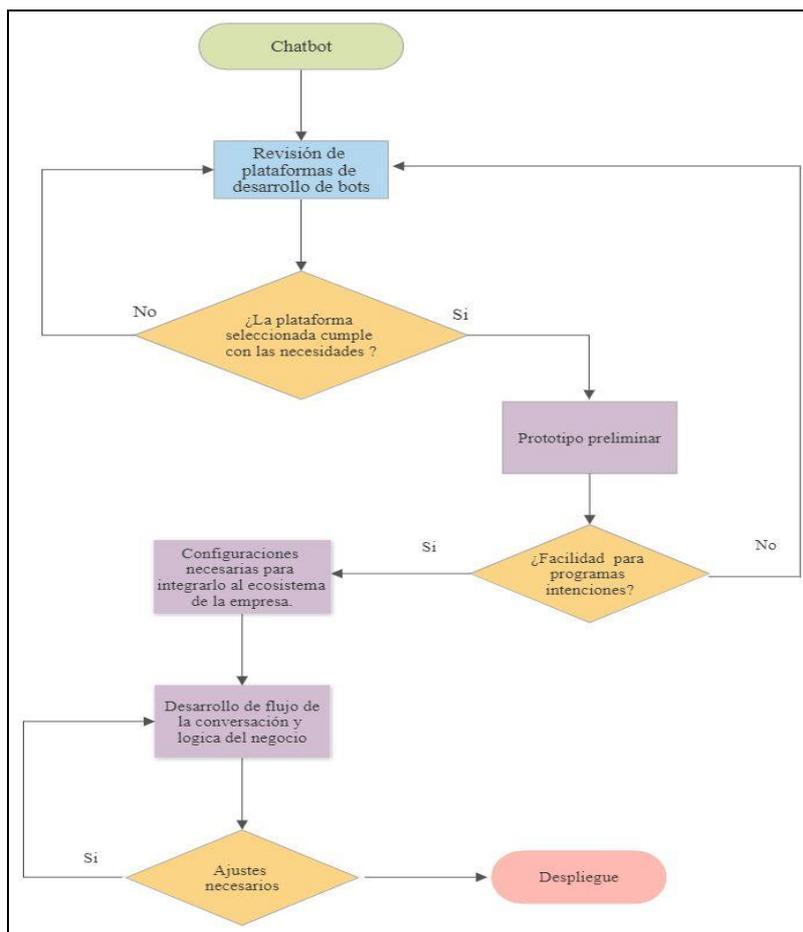


Figura 4. Metodología sugerida para el desarrollo del chatbot. Elaboración propia.

### Proveedores para desarrollo de chatbots.

En esta fase se identificaron las principales ventajas y/o desventajas que presenta cada una de las plataformas de desarrollo de chatbots en cuestión. Para este proyecto, las plataformas y sus respectivos proveedores fueron DialogFlow de Google, BotFramework de Microsoft, Watson Assistant de IBM y Amazon Lex de Amazon. Entre ellas se estableció una matriz de comparación para generar relaciones o encontrar diferencias en base a características que puedan o no estar presente en estas plataformas. Algunas características consideradas fueron el idioma, las versiones (paga o gratuita), los tipos de respuestas, conexión con servicios externos, entre otros.

### **Prueba de Chatbot.**

Se realizó un prototipo preliminar en DialogFlow Essentials que permitió visualizar la facilidad para programar las intenciones y dar respuestas al usuario. De esta forma, se pudo observar las limitación o alcances que se tiene al usar esta tecnología.

### **Integración en el entorno corporativo.**

Durante esta etapa fue necesario conectar la plataforma de chatbots con un sitio web genérico usando el canal DialogFlow Messenger, el cual permitió visualizar la interfaz conversacional dentro de este sitio web y donde se pudo comprobar los diferentes tipos de respuesta que se pueden implementar a la hora de mostrar la información al usuario. Además, se conectó la plataforma de chatbot con un servidor de la empresa, ya que era necesario ejecutar la lógica o Backend que se encontraba alojada para poder extraer información de los sitios web corporativos vía APIs. En este punto fue necesario utilizar una ruta publica de acceso de la forma <https://subdominio.com>, que redireccionó a la IP y puerto del servidor donde se ejecuta el Backend, ya que el webhook solo funciona según el protocolo HTTPS por lo cual no era posible utilizar la IP y el puerto directamente.

### **Flujo de conversación y lógica del negocio.**

Se implementó un pequeño flujo de conversación que permite mantener la coherencia en la interacción Usuario- Bot. De esta forma, la conversación es continua y bajo los criterios que sean establecidos evitando los errores semánticos o de interpretación que pueden generarse; para esto, fueron desarrollados una ruta de menús y sugerencias que son útiles para guiar al usuario.

Por otra parte, se desarrollaron los algoritmos necesarios para darle funcionalidad al chatbot. Para este propósito, se utilizó Node.js el cual es un framework de JavaScript que trabaja del lado del servidor (Backend). Esta lógica permitió que el chatbot pudiera acceder a las fuentes de información corporativas vía API REST, extrayendo la información en

formatos JSON. Una vez la información estuvo representada en el formato deseado fue enviada a la plataforma de chatbots para convertirla en lenguaje natural.

### **Resultados y análisis**

Como resultados de esta investigación se desarrolló un chatbot totalmente funcional e integrado en el ambiente Choucair, que apoya los procesos de consulta de información en diferentes sitios web corporativos. Inicialmente seleccionamos la plataforma que mejor se ajustó a las necesidades de la empresa. Para esto realizamos una revisión detallada de los principales proveedores de desarrollo de chatbots y sus plataformas, teniendo en cuenta aspectos de interés como la facilidad de uso y configuración de la plataforma, el procesamiento del lenguaje natural (NLP) en español, el costo de uso, entre otros. Además, para apoyar esta selección se utilizaron algunos criterios obtenidos de Gartner, empresa para consultoría e investigación sobre soluciones TI, de acceso público y de amplia colaboración a nivel mundial en el sector empresarial ("Gartner: Fueling the Future of Business", s.f.). Las plataformas evaluadas fueron Watson Assistant de IBM, LEX de Amazon, BotFramework de Microsoft y DialogFlow de Google.

Con la información obtenida, se elaboró una matriz de comparación para confrontar estos criterios e identificar ventajas y desventajas del uso de cada una de ellas (Tabla 1). En base a esto, pudimos establecer que la plataformas DialogFlow Essentials era la mejor opción según las necesidades de la empresa, debido a que esta plataforma nos ofrece una versión totalmente gratuita a partir de la cual se pueden explorar sus funcionalidades y establecer limitaciones y alcances ("Ediciones | Documentación de Dialogflow | Google Cloud", s.f.). Sumado a esto, las solicitudes de entrada se pueden realizar en diferentes formatos como texto, audio o llamadas ("Ediciones | Documentación de DialogFlow | Google Cloud", s.f.), mientras que en las otras plataformas es necesario configurar servicios adicionales para procesar audio o llamadas lo cual podría constituir una limitante ya que hacer uso de otros servicios puede

incurrir en mayores costos y esfuerzos de desarrollo. Además, la consola de desarrollo de DialogFlow ofrece una rápida implementación y entrenamiento de agentes, comprensión del lenguaje natural basado en speech to text y text to speech. Así mismo, esta plataforma cuenta con una amplia comunidad de colaboradores que respaldaron esta selección como la mejor solución ("Dialogflow | Google Cloud", s.f.).



Tabla 1

*Principales características de las plataformas evaluadas.*

Características	DialogFlow	BotFramework	IBM watson assistant	Amazon Lex
Solicitud de entrada	Texto, voz y llamadas.	Solo texto a través de L.U.I.S, requiere de un servicio de conversión de voz para procesar audio.	Solo texto (requiere herramienta de transformación text to speech adicional).	Texto y voz.
Respuesta de Salida	Texto, voz, enlaces, botones de sugerencia, tarjetas.	Texto, voz (API diferente a L.U.I.S), enlaces, botones de sugerencia, tarjetas.	Texto simple.	Texto, voz.
Dispositivos y aplicaciones	Sitios web, Smartphones, Smartwatches, Google Home, Dispositivos de IoT.	Dispositivos de IoT, Smartphones, sitios web.	Sitios web, Smartphones, Dispositivos.	Amazon echo, Smartphones, sitios web, Dispositivos IoT.
Idioma	Mas de 20 idiomas.	19 idiomas.	13 idiomas.	Ingles Estadounidense: <b>en-US</b> .
Canales de Integración	Asistente de Google, Slack, Facebook, Microsoft Teams, Messenger, LINE, Telegram, Twitter, Amazon Alexa. Standard: uso limitado de funciones de texto y voz. No incluye análisis de opiniones.	Microsoft Teams, Slack, Skype (100 usuarios), LINE, Facebook Messenger.	Facebook Messenger, Intercom, Slack, Aplicación web.	Facebook Messenger, Slack, Twilio, DialogFlow.
Versiones	Enterprise: Texto: 0.002-0.004\$; Voz: 0.0065-0.0085\$ (15 seg).	Prueba de 12 meses gratis: Para L.U.I.S 10000 solicitudes de texto/mes.  L.U.I.S: 1.50\$ por cada 1000/mes y 5.50\$ por 1000 de voz/mes.	Lite: versión gratis hasta alcanzar limites(10.000msm/mes). Plus (uso gratuito durante 20 días), Premium: 0.0025\$ (más económica) y Deploy Anywhere.	Prueba por 12 meses  0.00075\$ por solicitud de texto; 0.004\$ por solicitud de voz (10 seg).

Características	DialogFlow	BotFramework	IBM watson assistant	Amazon Lex
Lenguaje de programación	DialogFlow client Libraries: Node.js, python, C#, Go, Java, Ruby, PHP.	.NET (C#), Node.js, Python.	Node.js, Python, .NET, Java, Go, Ruby.	Node.js, java, .NET, PHP, Ruby.
Entorno	Google Cloud Plataform.	Azure.	IBM Cloud.	AWS.
Protocolo de acceso a la APIs	REST, gRPC, Bibliotecas del cliente.	REST.	REST.	REST.
Integración con servicios externos	Webhook: Request y Response: API externas, bases de datos.	API externas, bases de datos	API externas, bases de datos: Mongo DB, Postgresql MinIo (Amazon Lex).	AWS Lambda para bases de datos.

La construcción de un prototipo preliminar nos permitió delimitar el alcance de esta plataforma, por ejemplo, se observó la posibilidad de configurar múltiples intenciones de forma rápida, se usaron diferentes entidades para almacenar nombres y parámetros de consulta y se exploraron diferentes canales de integración como son DialogFlow Messenger y Google assistant. El primero permite tener la interfaz conversacional dentro de un sitio web propio, mientras que el segundo invoca al agente desde un dispositivo Android mediante una frase de invocación ("Dialogflow Messenger | Documentación de Dialogflow | Google Cloud", s.f.; Camacho, 2019). Teniendo en cuenta lo anterior, nos inclinamos por tener una ventana de chat conversacional embebida en un sitio corporativo por lo que nuestro canal seleccionado fue DialogFlow Messenger.

Una vez seleccionado el canal, se exploraron diferentes opciones para mostrar las respuestas a los usuarios como botones, tarjetas descriptivas e informativas, imágenes y listas; de esta forma la comunicación entre el usuario y el chatbot puede ser más dinámica y organizada. Seguidamente, se elaboró un menú principal con sugerencias para el usuario que le permiten navegar a través de las diferentes secciones del portal principal. Hasta este punto, se logró tener una comunicación bidireccional entre la interfaz conversacional y DialogFlow. Sin embargo, para que el chatbot pueda entregarle al usuario la información sensible extraída de los sitios web corporativos, es necesario integrarlo en Choucair.

Para realizar la integración con el ecosistema Choucair es muy importante habilitar la comunicación vía webhook en la plataforma DialogFlow, ya que este proceso nos permite conectar DialogFlow con servicios externos a Google, en nuestro caso, el servidor de la empresa donde se tiene almacenada la lógica necesaria para consumir la información vía APIs. Esta comunicación permite intercambiar información entre DialogFlow y el servidor en formato JSON (Java Script Object Notation), según el protocolo HTTPs. De esta manera,

Dialogflow se encarga de transformar la información recibida en formato JSON a lenguaje natural.

Para lograr con éxito esta comunicación en DialogFlow se debió incluir una ruta de acceso publica con dirección al servidor, esta ruta indica hacia donde debe enviar DialogFlow una intención en caso de que un usuario la haya invocado previamente y tenga la opción de entrega habilitada. Debido a que el protocolo de comunicación establecido por DialogFlow es HTTPs no fue posible utilizar la IP y el puerto del servidor según la dirección (<http://IP:puerto>), sino que fue necesario utilizar una opción de tunneling y un subdominio. El tunneling nos permite exponer en la web de forma temporal la ruta de nuestro servidor local donde se implementa la lógica (<http://localhost:8000>), en este sentido, el tunneling es utilizado para efectos del desarrollo del Backend y no para fines de producción. De esta manera, podemos probar la lógica y verificar que la información sea enviada de forma correcta a DialogFlow antes de desplegar el resultado final en el servidor de producción. La opción de tunneling la implementamos mediante Ngrok, este paquete se instala en Node.js y se utiliza para que otros usuarios puedan acceder desde la web a nuestro servidor local (<http://localhost:8000>), de manera que puedan probar algunas funcionalidades durante la etapa del desarrollo de un sitio web ("¿Que es Ngrok?", 2017). Para lograr exponer el servidor local, lo que hace Ngrok es generar una url dinámica de la forma <https://numero.ngrok.co> que redirecciona a nuestro servidor local cuando recibe alguna petición procedente de DialogFlow. Cabe destacar que para fines de producción y despliegue fue necesario usar una solución más estable: un subdominio, esta ruta nos permitió la interacción entre DialogFlow y el servidor de forma permanente.

Una vez integrado el chatbot dentro del ecosistema Choucair, se desarrollaron los algoritmos necesarios para consumir la información almacenada en algunos sitios web corporativos como WikiChoucair, Moodle y Sharepoint. La Tabla 2, resume los sitios

consultado vía APIs y el protocolo de comunicación utilizado para el intercambio de información (REST).



Tabla 2

Sitios web corporativos de la empresa Choucair Testing.

Características	Sitios web corporativos				
	WikiChoucair	Choucair Academy	Intranet	Portal de procesos	Market place
Plataforma de desarrollo	Wikimedia	Moodle	Sharepoint	Sharepoint	Sharepoint
Información general	Filosofía corporativa y terminología.	Cursos formativos para empleados, colaboradores.	Noticias, eventos recientes y accesos a otros aplicativos	Procesos relacionados con pruebas de software.	Portafolio de productos de la empresa.
Protocolo de acceso	REST	REST	REST	REST	REST

Cuando se usa el protocolo REST la información viaja en formato JSON, por este motivo es conveniente utilizar Node.js como framework de desarrollo para construir la lógica. Ya que nos permite manipular este tipo de objetos de forma fácil, así mismo, es adecuado para realizar las diferentes consultas asíncronas que se requieren cuando se consultan la información de los sitios web vía APIs ("Acerca | Node.js", s.f.).

### **Conclusiones**

En este estudio, se encontró que la plataforma que mejor se ajusta a las necesidades de la empresa es Dialogflow Essentials, ya que incluye una serie de factores relevantes como: el diseño simple de agentes, la integración en múltiples canales, el entendimiento del lenguaje español y una versión gratuita que permiten explorar ampliamente sus servicios.

La integración en el ecosistema Choucair fue posible al habitar el canal Dialogflow Messenger, con esto, se pudo visualizar la interfaz conversacional dentro de un sitio web corporativo. Así mismo, se elaboró un subdominio seguro (protocolo HTTPS) que permitiera a Dialogflow intercambiar información con el servidor.

Finalmente, se logró desarrollar el Backend necesario para consultar la información a través de las APIs de algunos sitios corporativos como: WikiChoucair, Academy Choucair (Moodle) y SharePoint, mediante el uso de consultas asíncronas hechas con el framework Node.js

## Referencias Bibliográficas

¿Qué es Ngrok?. *NeverCracker*. (2017). Consultado el 4 de octubre de 2020.

<https://nevercrackerblog.wordpress.com/2017/09/30/que-es-ngrok/>.

*Actions and parameters | Dialogflow ES | Google Cloud*. Google Cloud. (s.f.). Consultado el

28 de septiembre de 2020. [https://cloud.google.com/dialogflow/docs/intents-actions-](https://cloud.google.com/dialogflow/docs/intents-actions-parameters)

[parameters](https://cloud.google.com/dialogflow/docs/intents-actions-parameters)

*Agents | DialogFlow ES | Google Cloud*. Google Cloud. (s.f.). Consultado el 28 de septiembre

de 2020. <https://cloud.google.com/dialogflow/docs/agents-overview>

Brandtzaeg, P., & Følstad, A. (2017). Why People Use Chatbots. *Internet Science*, 377-392.

[https://doi.org/10.1007/978-3-319-70284-1\\_30](https://doi.org/10.1007/978-3-319-70284-1_30)

Camacho, L. (2019). *Paso a paso para crear una acción para Google Assistant en el*

*TEDxPuraVidaJoven 2019*. Medium. Consultado el 3 de octubre de 2020.

<https://planetachatbot.com/paso-a-paso-para-crear-una-acci%C3%B3n-para-google-assistant-en-el-tedx-puravida-joven-2019-174aa40790a0>.

*Dialogflow | Google Cloud*. Google Cloud. (s.f.). Consultado el 4 de octubre de 2020.

[https://cloud.google.com/dialogflow?&utm\\_source=google](https://cloud.google.com/dialogflow?&utm_source=google).

*Dialogflow Documentation | Google Cloud*. Google Cloud. (s.f.). Consultado el 28 de

septiembre de 2020. <https://cloud.google.com/dialogflow/docs>

*Dialogflow Messenger | Documentación de Dialogflow | Google Cloud*. Google Cloud.

Consultado el 3 de octubre de 2020.

<https://cloud.google.com/dialogflow/docs/integrations/dialogflow-messenger?hl=es-419>

*Ediciones | Documentación de DialogFlow | Google Cloud*. Google Cloud. (s.f.) Consultado

el 2 de octubre de 2020. <https://cloud.google.com/dialogflow/docs/editions?hl=es-419>.

- Fadhil, M. (2018). Beyond Patient Monitoring: Conversational Agents Role in Telemedicine & Healthcare Support for Home-Living Elderly Individuals. University of Trento. <https://arxiv.org/ftp/arxiv/papers/1803/1803.06000.pdf#:~:text=Chatbots%20can%20offer%20a%20lot,feel%20more%20human%2Dlike%20characteristics.>
- Fernández, A. (2019). Inteligencia artificial en los servicios financieros. *Boletín económico-Banco de España* 2, 1-8. <https://dialnet.unirioja.es/servlet/articulo?codigo=7040792>
- Fulfillment | Dialogflow ES | Google Cloud*. Google Cloud. (s.f.). Consultado el 28 de septiembre de 2020. <https://cloud.google.com/dialogflow/docs/fulfillment-overview>
- Gutiérrez, J. (2019). Desarrollo de chatbots con entornos de código abierto. [Tesis pregrado, Universidad de Cantabria]. <https://repositorio.unican.es/xmlui/bitstream/handle/10902/16456/418113.pdf?sequence=1&isAllowed=y>
- Ilievski, A., Dojchinovski, D., & Gusev, M. (2019). Interactive Voice Assisted Home Healthcare Systems. *Proceedings Of The 9Th Balkan Conference On Informatics*. <https://doi.org/10.1145/3351556.3351572>
- Intents | Dialogflow ES | Google Cloud*. Google Cloud. (s.f.). Consultado el 28 de septiembre de 2020. <https://cloud.google.com/dialogflow/docs/intents-overview>
- Quotas and limits | Dialogflow | Google Cloud*. Google Cloud. (s.f.). Consultado el 28 de septiembre de 2020. [https://cloud.google.com/dialogflow/quotas#es-agent\\_1](https://cloud.google.com/dialogflow/quotas#es-agent_1)
- Radziwill, N., Benton, M. (2017). Evaluating Quality of Chatbots and Intelligent Conversational Agents. Cornell University. <https://arxiv.org/abs/1704.04579>
- Responses | Dialogflow ES | Google Cloud*. Google Cloud. (s.f.). Consultado el 28 de septiembre de 2020. <https://cloud.google.com/dialogflow/docs/intents-responses>
- Rísquez-Almodóvar, P. (2019). Desarrollo de un chatbot para modelado colaborativo en Slack. [Tesis de Pregrado, Universidad Autónoma de Madrid].

[https://repositorio.uam.es/bitstream/handle/10486/688978/risquez\\_almodovar\\_pablo\\_tfg.pdf?sequence=1&isAllowed=y](https://repositorio.uam.es/bitstream/handle/10486/688978/risquez_almodovar_pablo_tfg.pdf?sequence=1&isAllowed=y)

Rodríguez, J., Merlino, H., & Fernández, E. (2014). Comportamiento Adaptable de Chatbots Dependiente del Contexto. *Revista Latinoamericana De Ingeniería De Software*, 2(2), 115. [doi: 10.18294/relais.2014.115-136](https://doi.org/10.18294/relais.2014.115-136)

Rosruen, N., & Samanchuen, T. (2018). Chatbot Utilization for Medical Consultant System. *2018 3Rd Technology Innovation Management And Engineering Science International Conference (TIMES-Icon)*. <https://doi.org/10.1109/times-icon.2018.8621678>

Shawar, B. A. & Atwell, E. (2007). Chatbots: are they really useful? *Journal for Language Technology and Computational Linguistics*, 22(1): 29-49. [https://www.researchgate.net/publication/220046725\\_Chatbots\\_Are\\_they\\_Really\\_Useful](https://www.researchgate.net/publication/220046725_Chatbots_Are_they_Really_Useful)

Shum, H., He, X., & Li, D. (2018). From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1), 10-26. <https://doi.org/10.1631/fitee.1700826>

*Training phrases | Dialogflow ES | Google Cloud*. Google Cloud. (s.f.). Consultado el 28 de septiembre de 2020. <https://cloud.google.com/dialogflow/docs/intents-training-phrases>

*Acerca / Node.js*. Node.js. Consultado el 4 de octubre de 2020. <https://nodejs.org/es/about/>.