



BENEMÉRITA UNIVERIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

**“FRAMEWORK COMO SOFTWARE INTELIGENTE
PARA LA FACILIDAD DE LA INVESTIGACIÓN EN
EN TÓPICOS AVANZADOS EN INGENIERÍA Y
TECNOLOGÍA.”**

TESIS PROFESIONAL PARA OBTENER
EL GRADO DE:
**INGENIERÍA EN CIENCIAS DE LA
COMPUTACIÓN**

PRESENTA:
ESMERALDA MITCHELL LUNA PÉREZ

ASESOR:
M.C. MARIANO LARIOS GÓMEZ

PUEBLA, PUEBLA, DICIEMBRE 2022

DEDICATORIA

Sin duda no habría llegado hasta donde estoy sin las enseñanzas de mis padres por eso este trabajo está dedicado a ellos que siempre han estado ahí para mí, a ellos que día a día me enseñan el valor del trabajo y de la familia, por otra parte, se lo dedico a mis abuelitos que criaron a el hombre y la mujer que tengo hoy como padres; esto también va dedicado a ellos, hombres y mujeres fuertes, personas de trabajo que aunque dos de ellos no están presentes sé que están orgullosos de ver hasta donde ha crecido sus buenas siembras.

Dedico mi tesis a José Luis, Angélica, Miguel, Regina, Andrés y Margarita, los amo sin ustedes no hubiera sido posible.

AGRADECIMIENTOS

Agradezco a Dios por darme a mis papás maravillosos que siempre han creído en mí y me han dado todo su amor y apoyo, a mi hermano, abuelitos, familia y a mi Güero por siempre echarme porras y confiar en mí, así como a mi asesor Mariano Larios Gómez por su tiempo y paciencia en el transcurso de la realización de esta tesis de investigación; por otra parte, agradezco al LNS:

- Los autores agradecen a las personas del Laboratorio Nacional de Supercomputación del Sureste de México perteneciente a los laboratorios nacionales del CONACYT, por toda la asistencia técnica y los recursos computacionales.
- The authors acknowledge to the people from the National Laboratory of Supercomputing of Southeast of México that belongs to the CONACYT national laboratories, for all the technical assistance and the computational resources.

ÍNDICE GENERAL

	Página
Dedicatoria	2
Agradecimientos	3
Contenido	4
Índice de tablas	7
Índice de figuras	8
List of algorithms	10
1. Aspectos Preliminares	11
1.1. Antecedentes del proyecto de investigación	12
1.2. Planteamiento del problema	13
1.2.1. ¿Por qué es un problema?	14
1.3. Justificación	15
1.4. Metodología	18
1.4.1. Design Thinking.....	18
1.4.2. Software inteligente	20
1.5. Objetivo general y específicos	23
1.6. Impacto socioeconómico	23
1.7. Aportaciones	24
2. Estado del Arte	25
2.1. El movimiento educativo abierto	25
2.1.1. El movimiento educativo abierto en la sociedad (MEAA)	27
2.2. Desarrollo e implementación de proyectos educativos abiertos en los principales países de América Latina	28
2.2.1. El movimiento educativo abierto como buenas prácticas para el desarrollo e implementación de REA	32
2.3. Máquina de aprendizaje (chatbots) sobre los repositorios	

institucionales	33
2.3.1. Personalización	38
3. Diseño del chatbot para la comunicación con repositorios	40
3.1. Gestión de Diálogo Determinista	40
3.1.1. Selección de acción y método de programación	43
3.2. Diseño del diagrama convencional	46
4. Implementación de un chatbot para ayuda de investigadores de posgrado	55
4.1. Frameworks descentralizado en un repositorio institucional	59
4.2. Configuración e instalación de un repositorio institucional con DSpace61	
4.2.1. Instalación de un repositorio institucional con DSpace.....	62
5. Pruebas y análisis.....	63
5.1. Implementación de un Bot en DialogFlow	66
5.2. Servidor externo.....	70
5.2.1. Librerías usadas	70
5.2.1.1. actions-on-google(dependencia).....	71
5.2.1.2. dialogflow-fulfillment(dependencia)	71
5.2.1.3. dotenv(dependencia)	71
5.2.1.4. ejs(dependencia).....	71
5.2.1.5. express(dependencia).....	72
5.2.1.6. googleapis(dependencia).....	72
5.2.1.7. nodemon(dependencia)	72
5.2.2. Frontend	72
5.2.3. Backend.....	76
5.2.3.1. WebHook	77
5.2.3.2. Conexión con motor de búsqueda de Google	83
5.3. Pruebas de funcionamiento	87
6. Resultados finales	90
6.1. Propuesta para promover la motivación de los estudiantes en cursos universitarios en línea en el área de Ingeniería y Tecnología	90

6.2. Cuestionario que realiza el chatbot a los usuarios.....	92
6.3. Cuestionario para la evaluación del InvestigaBot	93
7. Conclusiones	94

ÍNDICE DE TABLAS

1.1. Tabulador para estudiantes de tiempo completo.....	16
2.1. Categorías de los chatbots.....	34
2.2. Chatbot online	36

ÍNDICE DE FIGURAS

1.1. Puntos importantes del ¿Por qué es un problema?	14
1.2. Categorización de Deep Learning.	21
1.3. Diagrama general del funcionamiento del proyecto.	22
2.1. Repositorios registrados en la RRLA en LATAM.	30
2.2. Repositorios y número de registros en REMERI.....	31
3.1. Diagrama de flujo básico de la acción del chatbot.	45
3.2. Diagrama de flujo de toma de decisión.	46
3.3. Flujo de conversación básico chatbot-usuario.....	47
3.4. Subdivisiones de las acciones del chatbot.	49
3.5. Flujo central del chatbot.	50
3.6. Flujo de condiciones sobre la selección adecuada.	50
3.7. Diagrama de flujo básico de la acción del chatbot.	51
3.8. Ejemplo de flujo de conversación.....	52
3.9. Diagrama de caso de uso para el chatbot.	53
3.10. Diagrama de secuencia para el chatbot.	54
4.1. Chatbot con DM sin Machine Learning.....	55
4.2. Chatbot con DM con Machine Learning.	56
4.3. Respuesta del chatbot.....	56
5.1. Presentación inicial del InvestigaBot.	63
5.2. Ventana principal de chat.	64
5.3. Manual de usuario.	64
5.4. Preguntas frecuentes.	65
5.5. Términos y condiciones.....	65
5.6. Contacto.	66
5.7. Pantalla de creación de agente (Bot).	67
5.8. Pantalla de intenciones del agente (Bot).....	67
5.9. Creación de un intent en DialogFlow.....	68

5.10. Muestra de 4 intenciones al usar el Bot de DialogFlow.	69
5.11. Integración de las intenciones al Bot.	70
5.12. Vista de la interfaz gráfica.	73
5.13. Pantalla de inicio de DialogFlow (intenciones).	73
5.14. Pantalla de integraciones	74
5.15. Búsqueda de la integración DialogFlow-Messenger.	74
5.16. Ventana de implementación de DialogFlow-Messenger (deshabilitada).	75
5.17. Ventana de implementación de DialogFlow-Messenger (habilitada).	75
5.18. Habilitando cumplimiento para la intención.	81
5.19. Ingresando URL público a DialogFlow.	82
5.20. Ingresando URL público a DialogFlow.	83
5.21. Inicio de InvestigaBot.	87
5.22. Resultados de consulta en InvestigaBot.	88
5.23. Dirección de redireccionamiento desde InvestigaBot.	88
5.24. Realizando consulta en InvestigaBot.	89
6.1. Aprendizaje con ABP.	90
6.2. Criterio cuantitativo del aprendizaje con ABP.	91

LIST OF ALGORITHMS

Capítulo 1

ASPECTOS PRELIMINARES

Hoy en día la información es de gran importancia en todos los ámbitos y un software inteligente puede ayudarnos a obtener justo lo que necesitamos de esta. En este trabajo se requirió ayudar a un grupo de personas con un problema específico, como un estudiante de posgrado para realizar su investigación rápida y de manera óptima. Por lo cual, el primer punto importante es el tiempo que se pierde al pensar un proyecto de investigación, porque las ideas no se dan tan rápido al plantear un tema de investigación, sobre todo a nivel de posgrado y en ocasiones se tienen muchas distracciones u otras actividades laborales. Así como, tiempos perdidos en búsqueda de temas de proyectos de investigación sin tener buenos resultados. También la necesidad de ayuda para comenzar dicha búsqueda y mejorar la investigación con referencias óptimas al proyecto. De igual manera, el localizar material nuevo con opciones de búsqueda en diferentes idiomas.

Este trabajo de investigación propone el desarrollo de una herramienta tecnológica con chatbots para la asistencia o ayuda en un ambiente educativo y de investigación. La propuesta de herramientas tecnológicas en estos últimos años ha sido de gran importancia, sobre todo en tiempos de pandemia del COVID-19, fundamentan el desarrollo e innovación tecnológica y la inteligencia artificial. En el ámbito educacional a nivel superior y de posgrado, se tienen la enseñanza tópicos avanzados tecnológicos y científicos, en caso específico, en las ciencias de la computación e ingeniería tecnológica. El acelerado crecimiento de estos tópicos ha creado un ligero rezago en la enseñanza, aprendizaje e investigación por la actualización y creación de paradigmas. Se propone aplicar la enseñanza de tecnologías digitales basadas en proyectos (ABP) aplicados en áreas de importancia con ayuda de un asistente inteligente como un chatbot que utiliza metodologías educativas como el aprendizaje basado en proyectos con la ayuda de repositorios y software inteligente para la comunicación con los usuarios. Los resultados de las etapas de este proyecto de investigación fueron publicados en revistas científicas y en capítulos de libros nacionales e internacionales.

La adquisición del conocimiento se fundamenta en el aprendizaje mediante cursos específicos como se alude en Colomé and Femenia (2018), donde los estudiantes de posgrado realizan un proyecto de investigación, desarrollo o una mejora. El acelerado crecimiento de tópicos avanzados en computación y en las tecnologías de innovación ha creado un ligero rezago en la enseñanza y aprendizaje en el ambiente educacional superior, principalmente en América Latina según la

Objetivos de desarrollo Sostenible (ODS) en tiempos del COVID-19 como se comenta en Uzcátegui-Varela and Bracho-Orlandoli (2021) y CEPAL (2019).

Un conjunto significativo de los alumnos en las universidades de México, tiene dificultades en el aprendizaje de tópicos avanzados y en el manejo de nuevas tecnologías como se menciona en el trabajo, así como en la innovación en aspectos científicos y tecnológicos como se menciona en Larios-Gómez et al. (2020). El aprendizaje de conceptos avanzados mejora la vida profesional de los estudiantes al momento de egreso hacia la empresa e instituciones de investigación. Este trabajo se reforzó en un caso de uso en la aplicación de la metodología ABP, como una ayuda en el aprendizaje de los tópicos avanzados computacionales. Se realizó un estudio con universitarios de noveno, décimo, quinto y séptimo semestre de la carrera en ingeniería computacional y dando un ejemplo de proyecto para reproducirlo, de esta manera obtener su opinión del aprendizaje. La propuesta de un proyecto es fundamental en el aprendizaje adquirido durante los cursos en sus estudios universitarios, los estudiantes realizan un proyecto, lo desarrollan o lo mejoran, dando como resultado la opinión de los estudiantes con respecto al aprendizaje basado en proyectos. Por tal razón, en la enseñanza, aprendizaje e investigación, un software inteligente como un chatbot es de gran apoyo a un profesor- investigador, basados en el trabajo de Medrano et al. (2019):

- Entrega de trabajo académico en tiempo y forma.
- Entrega de trabajo de gestión en la universidad.
- Entrega de reportes y calificaciones en un periodo específico.
- Participar en convocatorias como apoyo a la investigación realizada.
- Creación y dirección de proyectos de investigación.
- Someter artículos científicos en revistas de investigación de alto impacto.

Estos puntos fueron de gran importancia en el desarrollo de este trabajo, porque nos proporcionó una orientación de como dirigir la implementación del chatbot *"InvestigaBot"*.

1.1. Antecedentes del proyecto de investigación

Esta propuesta de trabajo tiene como antecedente principal tres proyectos de investigación aceptados y terminados satisfactoriamente en el Laboratorio Nacional del Sureste (LNS), los cuales son:

1. Plataforma para la aplicación de algoritmos de planificación de procesos en tiempo real. Registro 201701071C, Larios Gómez (2017).
2. JScheduling: Software embebido para la planificación en tiempo real de procesos masivos. Registro 201801076C, Larios Gómez (2018).
3. Planificación y comunicación en un sistema distribuido simulando sobre una red P2P. Registro 201901014C, Larios Gómez (2019).
4. SMART-SOFTWARE para extraer recursos informáticos de repositorios distribuidos. Registro 202103070C Larios Gómez (2021).

En el proyecto aceptado en el LNS en 2019 fue como alumno doctoral de la Universidad Autónoma de Tlaxcala. Este proyecto de investigación es la base de nuestro trabajo, dando como resultado un algoritmo de planificación de tareas, este fue publicado en una revista internacional con editorial Springer e indexada en SCOPUS, como se puede ver en Larios-Gómez et al. (2019) y en el anexo A. Nos ayudó a tener un middleware para desarrollar el proyecto de investigación de tesis doctoral. En estos proyectos del 2018 y 2019 se desarrollaron herramientas que ayudan a observar y analizar el comportamiento de algoritmos distribuidos, sistemas colaborativos y de planificación de tareas en tiempo real aplicados en un entorno de repositorios educativos abiertos y el smart-software (InvestigaBot). Con el proyecto del 2021 se complementó con los resultados finales de esta tesis, desarrollando la parte complementaria que es un software inteligente y que se adaptó al framework que se construyó con los proyectos anteriores.

1.2. Planteamiento del problema

Primero nos basados en la investigación cualitativa y cuantitativa, como se menciona en Escudero Sánchez and Cortez Suárez (2018), se utiliza un conjunto de hechos y causas asociados al empleo de la metodología cuantitativa con una visión de la naturaleza dinámica de la realidad, de modo que se plantea la siguiente pregunta.

Pregunta principal

¿Cómo diseñar e implementar un software inteligente (chatbot) para extraer los Recursos Educativos Abiertos (REAs) y los recursos de patentes en repositorios institucionales de forma óptima y eficiente, como apoyo a estudiantes de posgrado y de ingeniería en instituciones de investigación o en universidades, para la elección y elaboración de su proyecto de tesis?

1.2.1. ¿Por qué es un problema?

Para responder esta pregunta, se pensó a detalle la importancia que tiene realizar un proyecto de investigación, como se muestra en la Fig. 1.1, donde se describen los puntos importantes del ¿por qué?. El primer punto habla del tiempo que se pierde al pensar un proyecto de investigación como se expresa en los trabajos Martin (2020), Burdick (2018) y Venegas Mejía et al. (2019). Se sabe las ideas no salen así de rápido o plantear un tema de investigación no sale por distracciones u otros casos, como se menciona en Concepción-Toledo (2019) la investigación debe tener unas bases sólidas para justificar su realización, tal que, si no se conocen las razones básicas que motivan un proyecto de investigación, justificando que no sea estéril y se tenga una buena inversión de tiempos y recursos para ello, dando una buena razón social, cultural, científica, etc., es decir, que proponga una excelente ejecución de un proyecto investigativo.



Figura 1.1: Puntos importantes del ¿Por qué es un problema?

El segundo punto establece la importancia del inicio del proyecto, sabiendo que esto es indispensable porque siempre el inicio es lo más difícil para realizar una actividad, como se señala en Zawoznik (2018), Herrera (2019) y Anglada and Abadal (2018). El tercer punto es producto de un buen inicio y de una buena selección de proyecto, esto quiere decir, haber completado el punto 1 y 2 correctamente, de modo que se contará con las referencias necesarias para iniciar un protocolo o pre-proyecto de tesis, como se recomienda en MARTELO et al. (2018), Pinedo-Tuanama and Valles-Coral (2021) y Anglada and Abadal (2018). Por último, el cuarto punto refleja la importancia de tener material de consulta adecuado que fue seleccionado sin limitarse por el idioma, como se enfatiza en Díaz-Castelazo (2018), Cárdenas (2019) y Ortiz-Núñez (2020).

1.3. Justificación

Este proyecto de investigación se sustenta ante la importancia de finalizar los estudios en niveles superiores y de posgrados en México, teniendo en cuenta que en la actualidad es relevante para el desarrollo educativo y tecnológico en países desarrollados económicamente, como se expone en (OECD (2019), pp 248), donde 4 países del grupo G7 se encuentran en los primeros 20 lugares que cuentan con profesionistas con estudios de posgrados según la OECD (Organización para la Cooperación y el Desarrollo Económicos). Cabe mencionar que en este ranking México se encuentra en el lugar 37, por tal motivo, es de importancia ayudar a mejorar la posición del país mediante herramientas de innovación tecnológica, sobre todo en el área de ciencias exactas y tecnología que ayuden al doctorante en la elección de un tema considerando sus habilidades, así como su área de interés para asegurar su buen término y obtención de su grado. Tomando en cuenta los argumentos de Tato and Pichardo (2005), la selección de un proyecto de investigación de tesis doctoral, maestría e ingeniería tiene varios factores de complejidad que conllevan a invertir tiempo y aspectos económicos, sobre todo los del área a investigar científica, lo que puede generar frustración en el estudiante y desalentarlo para buscar una titulación. Existen varios motivos por la deserción en los posgrados, principalmente cuando están por terminar y obtener el grado. Por ejemplo, falta de orientación desde el inicio hacia los estudiantes y el correcto seguimiento por parte del asesor para identificar sus líneas y destrezas de investigación, en Ortiz-Ocaña (2018) se comenta el caso de no establecer una guía para la realización de una tesis doctoral, lo que invita al doctorando a desanimarse cuando descubre que otros investigadores han escrito algo parecido a lo propuesto o encuentra o encuentra un trabajo sobre el mismo tema elegido. Por consiguiente, es muy importante por cuanto la mayoría de los estudiantes de doctorado piensan erróneamente (quizá porque sus profesores o su director de tesis se los han dicho) que el tema de investigación doctoral debe ser inédito. Por lo cual, es de gran importancia algunas sugerencias para que el tesista desarrolle su tema: y superar su barrera inicial; adoptar un método de trabajo; pensar en el lector de la tesis; emplear un estilo nítido; anteponer la calidad del manuscrito a la cantidad de

páginas; mantener un nivel de calidad y no dejarse limitar por los sinsabores de la escritura.

Así mismo, el principal motivo de la desmotivación para encontrar un tema de tesis y seguir adelante para obtener el grado, es el tiempo a invertir en el desarrollo e investigación de la tesis por situaciones económicas. Así como comenta en De León (2021) que, de 100 alumnos en las universidades, desertan 45 en ingenierías. Por otra parte, como alumnos de doctorado se encuentran en situaciones económicas graves, porque en México los estudiantes de posgrado tienen un bajo apoyo salarial como se muestra en la tabla 1.1(página oficial CONACyT¹), siendo que un doctorante ya cuenta con familia y un trabajo contando con poco tiempo y recurso para invertir en su desarrollo profesional.

Nivel pretendido	UMA	Monto
Doctorado	6.0	\$16,346.70
Maestría	4.5	\$12,260.02
Especialidad	4.0	\$10,897.80

Tabla 1.1: Tabulador para estudiantes de tiempo completo.

Sin embargo, para los estudiantes de posgrado y en especial de doctorado de instituciones no escolarizadas que no se encuentran en el PNPC (Programa Nacional de Posgrados de Calidad) de CONACyT, no se tiene apoyo económico y por tal motivo se debe trabajar en instituciones o sector privado para seguir estudiando. Así también en Brieger Rocabado (2019) se ha detectado que no existen hábitos de autosugestión y de autodisciplina por parte de los doctorantes por los horarios laborales que se tiene, las debilidades mayores se evidencian en el abordaje metodológico de las investigaciones, los tesisistas no ajustan sus investigaciones al cronograma académico y por tanto se presentan desfases de hasta 3 años o más. Existen debilidades en el aprendizaje debido a que los doctorantes arrastran deficiencias desde los niveles de formación anteriores al grado de doctor, por último, otro motivo de deserción en posgrados de acuerdo a Rivera Pérez and Mendoza-Becerril (2021), es el efecto del COVID-19, donde el 64.15% de los estudiantes consideran que su trabajo de tesis se ha visto afectado por la pandemia. El efecto es directo sobre los trabajos de investigación, ya que 53.4% de los estudiantes se han visto en la necesidad de re-estructurar su trabajo de investigación, ya sea mediante la reducción de trabajo experimental (28.26%), reducción de objetivos (23.91), cambiar el tema de tesis (17.39), eliminar estancias

¹ https://conacyt.mx/becas_posgrados/becas_nacionales/

(3.63), reducir el trabajo de campo (es decir, disminuir los sitios de muestreo, 13.78), ajustes del plan de trabajo (5), e incluso cambio de modelo de estudio (2.17), mientras un sector de la población tiene incierto el desarrollo de su tesis de posgrado (4.34), lo cual es alarmante, porque los posgrados tienen tiempos definidos para muchas instituciones, 2.5 años para maestría y 4.5 años para doctorado, mismos que no han dejado de contar en el tiempo de cierre por pandemia. Añadiendo que la coordinadora y presidenta de la Academia Mexicana de Ciencias e investigadora del Instituto de Radioastronomía y Astrofísica Susana Lizano Soberón (López and Romero (2020)) aseveró que la generación de conocimiento básico es una fórmula probada para generar soluciones y bienestar. No obstante, uno de los problemas en México es que no se le ha reconocido la relevancia de generar ese conocimiento básico de manera continua, suficiente, sostenida y sin cambios de rumbo. Esto requiere una consistencia de décadas, así también opino que, mediante diferentes plataformas digitales, la interacción entre los investigadores es relativamente buena, pero requiere más trabajo que simplemente pasar a la oficina de un colega para discutir los resultados recientes. Con gran esfuerzo se ha construido en nuestro país un sistema de investigación y desarrollo tecnológico, de bases sólidas, el cual debe ser reforzado y mejorado, pero no desmantelado, alertó el científico. Entre los aspectos a mejorar en esta vertiente mencionó el de tener un enfoque de evaluación y seguimiento que fomente más el trabajo colectivo y multidisciplinario. Por otro lado, Patiño Salceda (2019) identifica nueve modelos de realización de doctorados: a) de investigación, b) profesional, c) de enseñanza, d) el PhD por trabajo publicado, e) doctorado basado en la práctica, f) el doctorado de nueva ruta, g) interinstitucional, h) de cooperación, e i) industrial. En países de Asia, principalmente China, se identifican algunos doctorados similares a los encontrados en Europa, como los de investigación, profesional, interinstitucional e industrial, como se menciona en Bao et al. (2018). Además para la realización de un tema de tesis se puede tomar parte del proyecto del asesor o proponer uno nuevo de acuerdo al tema de interés del alumno o investigador joven.

Por tal motivo se tiene la necesidad de desarrollar una herramienta innovadora de software inteligente para ayudar a estudiantes de estudios superiores y de posgrado en la selección de proyectos de investigación de tesis, teniendo en cuenta la aportación a la innovación y desarrollo tecnológico, científico y educativo en México. Esta herramienta de innovación tecnológica se sustenta de las actuales tecnologías y paradigmas, y dan un apoyo considerable en el ambiente educativo en el desarrollo e implementación de algoritmos como base de comunicación del chatbot (Urrutia Ortiz, (2020)). Estableciendo una métrica de comunicación entre el chatbot (fuente de información) y usuario(s) en ambientes distribuidos y con ayuda de la teoría de sistemas de tiempo real, lo cual se tratara más adelante en secciones de este trabajo (Brandtzaeg and Følstad, (2017)).

La propuesta de utilizar una herramienta de innovación tecnológica ayudará a los estudiantes para la búsqueda de un tema de investigación de tesis, con opciones de consultar en repositorios de patentes o de REAs. Por medio de estas herramientas se puede facilitar el acceso y la educación de calidad a los estudiantes con diversas necesidades educativas y de investigación. Como se menciona en Romero Martínez et al. (2018), para lograr esta meta se ha llevado a cabo un estudio empírico de tipo cualitativo-descriptivo en el que se presenta un repositorio de las principales aplicaciones existentes para el trabajo con alumnos, analizando sus ventajas y desventajas. El estudio realizado en George Reyes (2019), Sánchez-Macías (2019) y Lucena et al. (2018), los estudiantes han encontrado áreas de oportunidad al enfrentar problemas del proceso de investigación en cuanto a la socialización de la información y el conocimiento, ya que si bien algunos de ellos han participado en congresos y han realizado publicaciones científicas, hay otros que manifiestan desinterés o desconocimiento para hacerlo. Esto obliga a fortalecer el quehacer científico de los estudiantes para que, de forma holística, se generen interacciones tecnológicas y de investigación y, con esto, se promuevan las competencias y habilidades necesarias para que las tecnologías sean parte de los procesos de investigación en posgrado.

1.4. Metodología

1.4.1. Design Thinking

La metodología Design Thinking o pensamiento de diseño es particularmente útil cuando se trata de problema retorcido o perversos como lo detalla Rittel en Rittel and Webber (1973), donde los define como problemas que están poco definidos o son complicados. Este problema se define como problema retorcido, porque no se pueden describir definitivamente. Además, en una sociedad pluralista no hay nada como el bien público indiscutible; no existe una definición objetiva de equidad; las reglas que responden a los problemas sociales no pueden ser significativamente correctas o falsas; y no tiene sentido hablar de soluciones óptimas a los problemas sociales a menos que primero se impongan severas calificaciones.

El método creativo de razonamiento en el pensamiento de diseño es el razonamiento abductivo, que difiere de las formas más comunes de razonamiento inductivo y deductivo, como se define en March (1976). Esto es a menudo lo que hacen los diseñadores e ingenieros como se expresa en Dorst (2011): crear un diseño que opere con un principio de funcionamiento conocido y dentro de un escenario establecido de creación de valor. Esta es una forma de resolución de

problemas “cerrados-abierto” que las organizaciones en muchos campos hacen a diario. Conocido como de razonamiento productivo (Abducción-2), es más compleja porque al comienzo del proceso de resolución de problema SOLO sabemos el valor final que queremos lograr. Esta forma de razonamiento está más estrechamente asociada con el diseño (conceptual).

Utilizamos la metodología *Design Thinking* por que se adapta a la solución de un problema social, llevado a la innovación tecnológica como lo especifican Plattner, Meinel y Leifer describiendo el proceso de innovación de diseño en cinco etapas;

1. (re)definir el problema,
2. identificar las necesidades y compararlas,
3. idear,
4. construir y
5. evaluar.

De igual manera Plattner, Meinel y Leifer afirman en Meinel and Leifer (2012) que “aunque las etapas son bastante simples, la experiencia de adaptación que se requiere para elegir los puntos de inflexión correctos y el paso siguiente apropiado es una actividad intelectual de primer nivel que requiere práctica y es de fácil aprendizaje.”

El proceso de desarrollo de una aplicación innovadora en la tecnología puede considerarse como un mecanismo de espacios superpuestos en los que se destacan la inspiración, ¿la creación de ideas y la implementación? En Meinel and Leifer (2012) y Meinel and Leifer (2012) describen brevemente estas cualidades aplicada al proyecto:

- Inspiración. El desarrollar un software inteligente para conversar con alumnos de posgrado y proporcionarle información que se encuentra en repositorios de REAs y de patentes sobre los temas que le interesan para proponer un proyecto de tesis.
- Empatía. Está destinado al grupo de investigación y educacional.

- Implementación y desarrollo de prototipos. La implementación, de las mejores ideas, como la realización de una aplicación web, donde el chatbot se comunique, como prototipo.

Este proyecto está enfocado en el área de ingeniería y educación, donde Design Thinking es fundamental para el diseño centrado en el usuario, métodos dominantes del diseño de interfaces humano-computadora como se menciona en Hasani et al. (2020) donde uno de los ámbitos de aplicación dentro la informática se centra en el proceso de diseño de software, la finalidad es integrar a los usuarios finales usando recursos del pensamiento de diseño como la lluvia de ideas y entrevistas, para atender sus demandas.

1.4.2. Software inteligente

El uso de las nuevas tecnologías para este proyecto se justifica por medio de la siguiente pregunta:

¿Cuál de las tecnologías de la industria 4.0 podemos utilizar para lograr una mejor extracción de información?

- Inteligencia Artificial (IA).
- Machine Learning (ML).
- Computo en la nube (Sistemas descentralizados).
- Motores de búsqueda y procesamiento del lenguaje natural.

Las máquinas de aprendizaje como técnica de inteligencia artificial, permite el uso del aprendizaje profundo o Deep Learning para el diseño e implementación de chatbots como se ilustra en la Fig. 1.2.

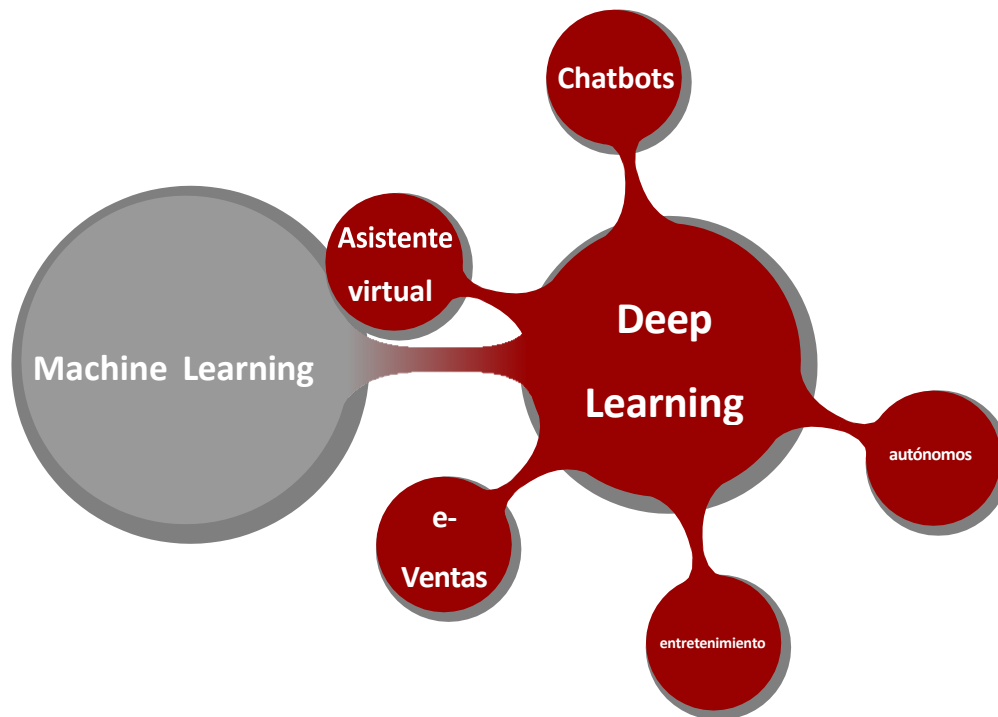


Figura 1.2: Categorización de Deep Learning

Los robots de software, o Chatterbots, se encuentran a diario en una variedad de formas. En la Fig.1.3 se describe la metodología que se siguió para el desarrollo de este proyecto de investigación, basándonos en un diagrama con los componentes y flujos básicos. Comenzando con el paquete y librerías para Java, Python y frameworks para trabajar con aplicaciones Web.

En los trabajos de AbuShawar and Atwell (2015) y Wallace et al. (2003) se propone una descripción general del chatbot ALICE (por sus siglas en inglés, Artificial Linguistic Internet Computer Entity), su formato AIML (por sus siglas en inglés, Artificial Intelligence Markup Language) y un experimento para generar diferentes prototipos como ALICE que están automáticamente basados en un enfoque de corpus. Así también se informó que el proyecto ALICE comenzó en 1995, fue inspirado en un Chatterbot anterior llamado Eliza, un Chatterbot que modelaba libremente un psiquiatra. ALICE y su lenguaje de implementación AIML se basa en la noción de que, si bien el pensamiento humano es bastante complejo, podría ser lo *suficientemente bueno* para simular el pensamiento proporcionado con patrones de respuesta *suficientes* a posibles consultas. Se mencionó un debate acalorado si este enfoque minimalista será *suficientemente bueno*. Sin embargo, ALICE y Chatterbots basados en ALICE y AIML han estado ganando el concurso anual de

Loebner², un concurso en el que Chatterbots intenta engañar a los jueces haciéndoles creer que son humanos durante los últimos años, como se argumenta en Texier et al. (2013).

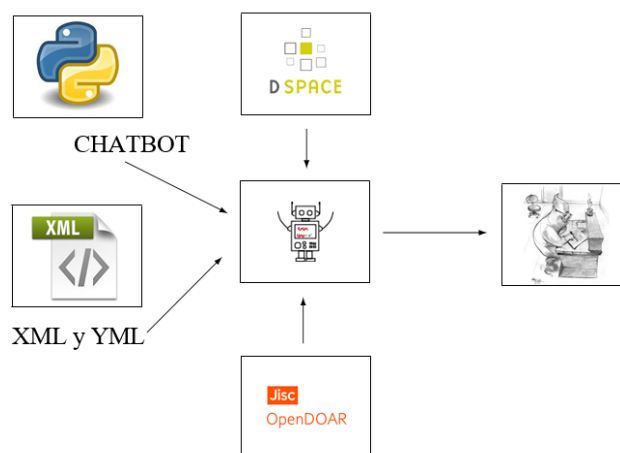


Figura 1.3: Diagrama general del funcionamiento del proyecto

La segunda etapa fue la configuración del software DSpace el cual es un servicio digital que recopila, conserva y distribuye material digital. Los repositorios son herramientas importantes para preservar el legado de una organización; facilitan la preservación digital y la comunicación académica. DSpace fue desarrollado en noviembre de 2002 por los laboratorios del MIT (Massachusetts Institute of Technology) y HP (Hewlett-Packard), como se declara en Smith et al. (2003). Este es utilizado para la creación y manejo de repositorio, como ejemplos, los repositorios de la Universidad Nacional Autónoma de México (UNAM) y el Consejo Nacional de Ciencia y Tecnología (CONACyT). Se puede buscar por localidad, contenido y otras métricas. Además, el servicio no requiere detalles completos del repositorio y busca los metadatos XML (Extensible Markup Language) de los repositorios. Por último, el chatbot se comunica con el/los repositorios que contienen las REA por medio de las plataformas DSpace para que el usuario final (investigador de posgrado) pueda obtener la información que necesita como ayuda para su proyecto.

² <https://loebner.exeter.ac.uk/>

1.5. Objetivo general y específico

El objetivo general es el diseñar e implementar un smart-software basado en una máquina de aprendizaje para ayudar a estudiantes de posgrado con la selección de un tema de investigación y poder proponer un proyecto innovador.

Los objetivos específicos tenemos:

- Usar un motor de búsqueda que permita localizar en repositorios institucionales abiertos de patentes y REAs coincidencias a un tema específico y con esto determina si es un tema inédito e innovador.
- Diseñar e Implementar un software como herramienta con inteligencia artificial capaz de comunicarse con un grupo de usuarios como alumnos de posgrado, para ayudar en la selección de temas de investigación.
- Adaptar el software inteligente en una plataforma de comunicación y/o colaboración como un hosting institucional.
- Realizar la comunicación entre nodos (peers) en un ambiente descentralizado que compartan recursos didácticos como los REA y patentes para ayudar a investigadores de posgrado a compartir búsquedas y servicios.
- Crear un algoritmo en tiempo real que permita la planificación de tareas y el motor de búsqueda entre el chatbot y el usuario final como un servicio al investigador de posgrado, y de esta forma pueda minimizar los tiempos para el inicio de su investigación.

1.6. Impacto socioeconómico

Con las prácticas modernas de las tecnologías de la información lo que se pretende a corto plazo es promover el ahorro de los costos en desarrollar proceso y estándares para el seguimiento de un proyecto de investigación en los centros de investigación, instituciones educativas o empresas de manera abierta, como una alternativa a implementar software inteligente el cual eleva los costos considerablemente y a su vez se necesita un personal especializado que busquen constantemente nueva información de los tópicos contemporáneos, diseñando e implementando medidas y políticas que cierren dichas brechas, de manera constante

y costosa, además de poder brindarles a los alumnos o usuarios de instituciones de investigación científica , un ejemplo real de cómo se manejan las consultorías importantes en tecnologías de la información y comunicación, haciendo más fácil la transición de estudiante a colaborador en un proyecto de investigación en una universidad.

1.7. Aportaciones

Con esta investigación y creación de la documentación se entrega el proceso para el desarrollo del smart-software utilizado para que los centros de investigación, instituciones educativas o empresas no tengan pérdidas de información. Siguiendo estándares que garanticen de manera cualitativa el mejor manejo de la información, así como presentar esta investigación en un congreso de tecnologías de la información en la sección de calidad y por último plasmar toda la investigación en un documento de tesis que será accesible a toda la comunidad científica y tecnológica.

Capítulo 2

ESTADO DEL ARTE

2.1. El movimiento educativo abierto

El Movimiento Educativo Abierto (MEA) se define como aquellas actividades educativas que se encaminan a la producción con licenciamiento abierto, selección y uso de *REA*, así como la diseminación de prácticas educativas en entornos académicos, gubernamentales, institucionales, científicos, entre otros; Ramírez Montoya (2013b). Así también, se puede ver como el desarrollo de la ciencia abierta, debido a que el objetivo principal de esta, es acelerar el progreso científico que con sus hallazgos, se beneficia principalmente al sector educativo y científico, según Masuzzo and Martens (2017). Se sabe que los REA son almacenados principalmente en repositorios institucionales de acceso libre y estos utilizan plataformas tecnológicas que además de servir para la preservación a largo plazo de los documentos, esto hacen posible que los artículos de revistas y otras publicaciones estén disponibles en Internet, con base en González-Pérez et al. (2017). Así, en los últimos años ha surgido un MEA a favor de esta modalidad de publicación como alternativa a los costes de suscripción que numerosos centros de investigación no pueden mantener, alude Adame Rodríguez et al. (2017). Las oportunidades se abren cuando se trabaja en conjunto con comunidades académicas de diferentes entornos para dar a conocer las ventajas que representa el compartir recursos, modelos, métodos y estrategias que nos lleven a poder utilizar en prácticas educativas abiertas. El enriquecimiento que otorga la integración de saberes de académicos de marcos contextuales diferenciados deviene en un avance en el MEA y en la posibilidad que nos da para tener acceso a información de calidad, en la opinión de Ramírez Montoya (2013a).

La oferta del Movimiento Educativo Abierto:

- Acceso. La UNESCO organizó en 2002, el primer foro mundial sobre recursos educativos de libre acceso, donde se adoptó el término OER (Open-Education-Resources por sus siglas en inglés) Atkins et al. (2007). Con la aparición de los OERs patrocinados por la Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO), el movimiento ha sido favorecido con el apoyo de políticas diseñadas desde instituciones internacionales que facilitan la democratización del conocimiento, lo cual representa su principal objetivo. Con el antecedente del OpenCourseWare del MIT, iniciativa que tuvo su aparición en 2001, un mayor número de instituciones educativas ofreció en forma gratuita y abierta

materiales pedagógicos. Debido a esto, en Miguel (2013) y Miguel (2012), los accesos abiertos significan el acceso libre y gratuito a través de Internet a la literatura académica y científica, permitiendo que todos los usuarios puedan bajar, leer, copiar, distribuir, imprimir, buscar o enlazar los textos completos de los artículos sin otras barreras legales, financieras o técnicas que las que suponga Internet en sí misma. La única restricción que se ejerce sobre la reproducción y la distribución consiste en garantizar su integridad y en darles a los autores el control de su trabajo y el derecho de ser apropiadamente citados y reconocidos.

- **Visibilidad.** En Ávila (2007), la difusión del movimiento educativo abierto registra una serie de beneficios y señala también nuevas oportunidades para la comunicación científica, incluidas la contribución para la visibilidad, accesibilidad, uso y efectos de los artículos y revistas científicas. Esto se ha evidenciado en diversos estudios. Los beneficios y oportunidades originados por el Open-Access (OA) son esenciales para la proyección de la investigación y comunicación científica en los países en desarrollo. Este movimiento está dirigido a investigadores y personas con el objetivo de incrementar sus conocimientos acerca de las publicaciones OA e identificar los potenciales beneficios que pueden obtener las instituciones y los autores, con una mayor difusión del conocimiento a nivel mundial.
- **Calidad.** En las revistas OA que cumplen con el movimiento educativo abierto de forma satisfactoria son evaluadas en lo que respecta a la calidad científica, educacional y original de su contenido, con base en la opinión de dos o más especialistas del área de la revista, en relación con las siguientes características: Un ejemplo de operación de la red es la revista científica SciELO, se basa fuertemente en infraestructuras nacionales e internacionales, lo que contribuye a garantizar su sostenibilidad y desarrollar las capacidades en comunicación científica en línea y en red en la modalidad de acceso abierto. Por tanto, el modelo SciELO se perfecciona continuamente, incorporando los avances internacionales y las lecciones de las experiencias en la operación de las colecciones según en Packer et al. (2006).

Entre las ventajas, la mayoría de ellos apuntaron como principal utilidad la de ser un complemento que permita obtener más información acerca de los países en que se descargan los artículos educativos, que suscitan mayor interés y que tipo de públicos descarga cada artículo. Otros editores aludieron a un aumento en la transparencia, ventajas señaladas por Williams (2017).

2.1.1. El movimiento educativo abierto en la sociedad (MEAA)

Entre las oportunidades que ofrece el (MEAA) para personas, organismos o instituciones se encuentra los lectores y el acceso sin barreras a la literatura tanto a lectores comunes (personas con curiosidad intelectual u observadores de adelantos tecnológicos) como a la comunidad académica (estudiantes, profesores e investigadores).

- Para profesores/instructores: el acceso abierto elimina la necesidad de permisos para reproducir y distribuir contenidos que contribuyan al proceso de enseñanza/aprendizaje. El uso de recursos como películas, libros académicos y no académicos, material técnico, reportes técnicos, etc., pueden ser parte del diseño de actividades de aprendizaje con las cuales se logre avances significativos, como se manifiesta en Gutiérrez and de los Santos (2009).
- El de las bibliotecas: los MEAA ayudan a que puedan ofrecer REA. En González-Pérez et al. (2017) se señaló que el acceso abierto significa la posibilidad de llegar al texto sin ningún costo para los usuarios o para las bibliotecas, lo cual se hace para acelerar la investigación y compartir el conocimiento; en este esquema los editores buscan que los costos sean cubiertos por otras fuentes.
- Para universidades e instituciones: El MEAA brinda la pauta para que se generen y publiquen los REA. Beneficio agregado del impacto de la investigación de sus miembros, como lo hace notar González-Pérez et al. (2017).
- Para revistas: El MEAA ayuda a que los artículos publicados sean más visibles (en diferentes países), recuperables y útiles como señala Ramírez Montoya et al. (2015).

Los MEAA significan el acceso libre y gratuito a través de Internet a la literatura académica y científica, permitiendo que todos los usuarios puedan obtener recursos didácticos digitales sin otras barreras, la única restricción que se ejerce sobre la reproducción y la distribución consiste en garantizar su integridad y en darles a los autores el control de su trabajo y el derecho de ser apropiadamente citados y reconocidos. De forma específica, a la comunidad científica y académica permite acceso a recursos educativos (libros, artículos, películas, videos, conferencias, tesis, proyectos en proceso, investigaciones, etc.) e incluso material esencial para realizar investigación Gracias al Movimiento Educativo Abierto se han desarrollado y se continúan desarrollando diferente software para la búsqueda, procesamiento

y análisis de REA. Con lo anterior, es posible realizar minería de datos, indización automática, creación de alertas, etc.

2.2. Desarrollo e implementación de proyectos educativos abiertos en los principales países de América Latina

La innovación educativa es un conjunto de ideas, procesos y estrategias que se aplican en los centros académicos para mejorar los procesos de enseñanza y aprendizaje. La aplicación de las tecnologías de la información y la comunicación en el campo de la innovación educativa ha sido fundamental para impulsar cambios que conduzcan a la mejora de los procesos administrativos y académicos. Sin embargo, algunos estudios muestran el uso deficiente y poco innovador de las tecnologías en la región latinoamericana. Para determinar la brecha existente, los autores están ejecutando un proyecto que intenta conocer el estado del arte actual sobre este tema. Este estudio corresponde al primer paso del proyecto. Aquí los autores describen el proceso de búsqueda sistemática diseñado para encontrar las tesis doctorales que se han desarrollado sobre la innovación educativa generada por las Tecnologías de la Información y la Comunicación (TIC). Para cumplir con este objetivo, el proceso tiene tres fases: (1) identificación de repositorios centralizados de tesis doctorales en español, (2) evaluación de repositorios seleccionados según criterios de búsqueda específicos, y (3) selección de repositorios centralizados donde es posible encontrar tesis sobre innovación educativa y TIC. El análisis de 5 de los 222 repositorios encontrados indica que cada sistema ofrece diferentes características de búsqueda, y los resultados muestran que la mejor opción es combinar los conjuntos de resultados ya que las tesis provienen de diferentes instituciones. Finalmente, considerando que las habilidades de los usuarios para administrar los sistemas de información pueden ser diversas, los proveedores y administradores de repositorios deben potenciar sus servicios de búsqueda de tal manera que todos los usuarios puedan encontrar y utilizar los recursos publicados, según Tapia-León and Chicaiza (2020).

México. El repositorio educativo abierto más importante en México es UNAM-RETo (Recursos Educativos para Todos) es una plataforma que tiene por objetivo organizar, archivar, preservar y difundir todos los contenidos educativos digitales que desarrollan las entidades académicas y dependencias de la UNAM Martínez and Tapia-Rangel (2018). Existen otros repositorios de universidades importantes de México como lo son: TEMOA es un buscador académico desarrollado por el Tecnológico de Monterrey que contiene REA diseñados para la educación universitaria y en el cual los recursos son evaluados por otros docentes para asegurar su calidad. Permite la categorización de materiales por área de conocimiento, nivel educativo e idioma, entre otros. TEMOA facilita la búsqueda de encontrar en Internet recursos educativos a la vez que permite crear comunidades o grupos de

investigación como plantea Mortera Gutiérrez (2010). Así también están los repositorios educativos abiertos que empiezan a generarse para seguir con esta tendencia en instituciones como la Universidad Autónoma de Tlaxcala.

Brasil. El Banco Internacional de Objetos Educacionais es una iniciativa del gobierno de Brasil que abarca contenidos educativos abiertos desde la educación pre-escolar a la universitaria. En portugués también se destaca el Banco Nacional de ITENS del Gobierno de Portugal que abarca REA para la educación básica y secundaria. Otra iniciativa es *M³ Matemática Multimedia* desarrollada por la Universidad de Campinas, el único repositorio dedicado a las matemáticas de la región. Cuenta con 350 recursos educativos en video, audio, software y experimentos bajo licencia Creative Commons, según Machado and Farias (2012).

Colombia: Colombia Aprende es un portal que contiene recursos para todos los niveles educativos. Cuenta con secciones para profesores y estudiantes como cursos virtuales y colecciones de contenidos. EDUTEKA fue desarrollado por la Fundación Gabriel Piedrahita Uribe. Cuenta con 13 años de experiencia y su objetivo es mejorar el conocimiento de las Tecnologías de la Información de los docentes de todas las áreas a través de la educación virtual. La plataforma cuenta con un promedio de 520.000 visitas mensuales de 168 países del mundo, como dice Martínez (2009).

Centroamericana. En esta región se destaca la Comunidad Educativa Centroamericana y República Dominicana (CEDUCAR), el portal educativo de la Comunidad Educativa de Centroamericana y República Dominicana. Se trata de un portal formativo que une a ocho países: Belice, Costa Rica, El Salvador, Guatemala, Honduras, Nicaragua, Panamá, República Dominicana, y a sus respectivos sistemas educativos. Dicho portal ofrece recursos para educadores de la región que incluyen por ejemplo un centro de recursos digitales, comunidades virtuales, cursos en línea y foros, entre otros. El portal Haiti Futur contiene recursos en creole y francés para la educación básica gestionado por el Pacte National pour l'Education en Haiti con el apoyo de Open Sankoré. El contenido incluye materiales sobre ciencias experimentales, matemáticas y francés, con base en Diez de Tancredi (2015). En la actualidad los repositorios en América Latina han crecido aceleradamente, según Dueñas (2008) y los registrados en 2021 en la Red de Repositorios Latinoamericanos (RRLA) se pueden ver en la Fig. 2.1. Recuperado de³.

³ <http://repositorioslatinoamericanos.uchile.cl/community-list>

La Red Mexicana de Repositorios Institucionales (REMERI) surge en el 2011 de la necesidad de contar con una plataforma que integra los repositorios digitales de las Instituciones Mexicanas de Educación Superior para permitir su difusión, localización y visualización de manera interoperable a través de interfaces comunes actualmente se tiene 76 instituciones inscritas en el REMERI, 104 repositorios y 685,221 documentos registrados.

La búsqueda de los documentos incorporados a la colección se realiza a través del formulario y se puede ingresar cualquiera de las siguientes opciones ya que la búsqueda se realiza en el título, autor, descripción, fecha e institución. La integración de repositorios a REMERI se logra utilizando un servidor en el estándar de un protocolo utilizado para la transmisión de metadatos en Internet de OAI- PMH (por sus siglas en inglés, Open Archives Initiative-Protocol for Metadata Harvesting). Para una validación inicial se generó un validador donde se puede proporcionar el Localizador de Recursos Uniforme URL (por sus siglas en inglés, Uniform Resource Locator) del servicio y este genera un reporte a nivel sintaxis (XML-Schema) de las respuestas del servidor.

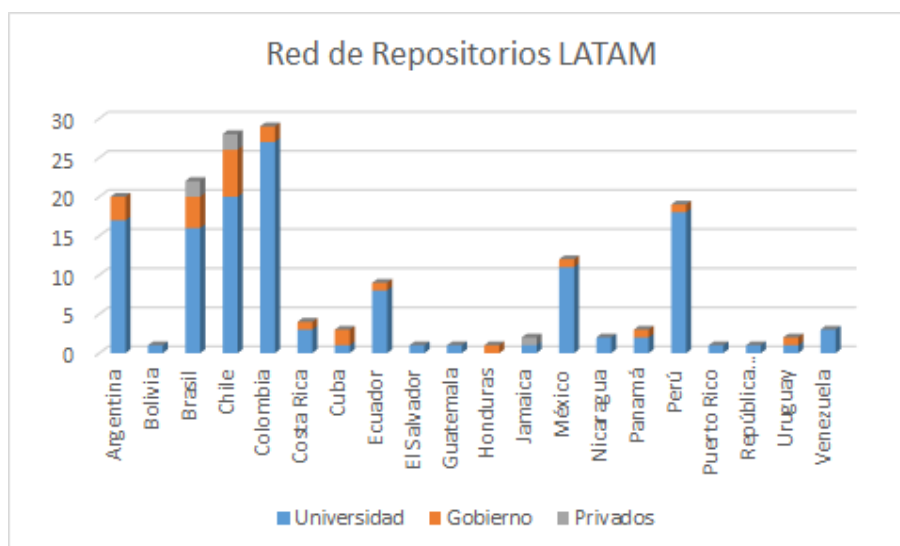


Figura 2.1: Repositorios registrados en RRLA en LATAM.

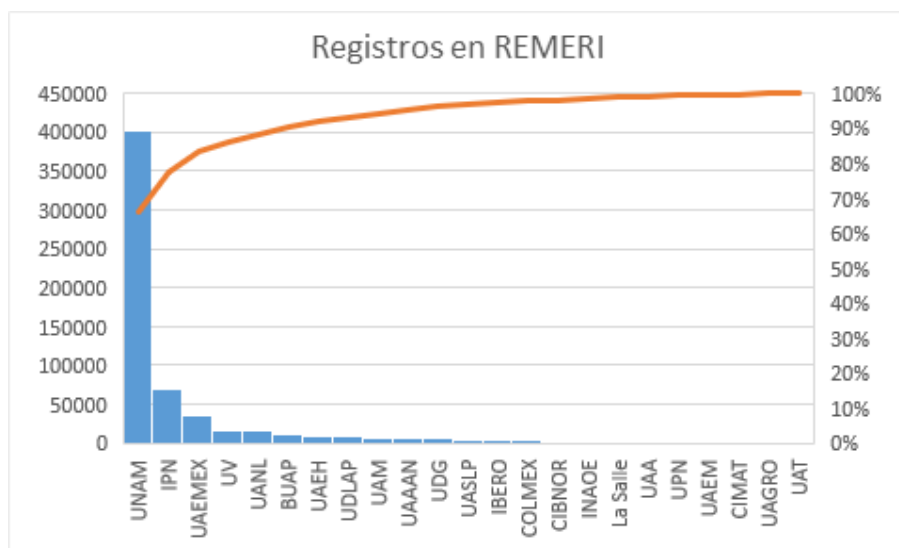


Figura 2.2: Repositorios y número de registros de REMERI

En la Fig. 2.2 se muestra los elementos que tienen los repositorios de cada una de las universidades en México, comenzando por la gran cantidad que tiene la UNAM con 400,723 registros de un total de 685,221 y con una diferencia de las otras instituciones, esta información es del año 2021 y se recuperó del sitio Web oficial de REMERI⁴. Las herramientas y plataformas más utilizados en la Red Mexicana de Repositorios Institucionales son herramientas de software como DSpace, el Directorio de Repositorios de Acceso Abierto (DOAR), el Registro de Repositorios de Acceso Abierto (ROAR) y Google Scholar, Sherpa, SAMVERA, DataVerse, JSON, etc. De esta forma y como resultado del trabajo se proponen los siguientes mecanismos para la puesta en marcha de un Programa de incentivación de la Producción Científica/Académica de la Universidad: 1) Programa de mentorías de investigadores consolidados hacia aquellos con carreras incipientes y potencial para la investigación con reconocimiento dentro del programa de estímulos al desempeño del personal docente, campaña publicitaria de las fortalezas, descubrimientos y/o proyectos de investigación, donde se le dé un lugar importante o presuma a aquellos docentes que han hecho aportaciones en sus ramas, promoción de las fortalezas de vinculación científica de la universidad, capacitaciones sobre ciencia abierta y buenas prácticas para la publicación científica, apoyo por parte de la coordinación de investigación y posgrado para el llenado de plataforma del sistema nacional de investigación (SNI), programa para el desarrollo profesional docente (PRODEP), cursos sobre la realización de proyectos de investigación, publicación de libros electrónicos en el repositorio institucional (RI),

⁴ <http://www.remeri.org.mx/app/index.html>

para no depender de grandes presupuestos para la publicación de resultados de investigación, y promover la publicación en revistas indexadas para las áreas que aún no tienen acercamiento.

2.2.1. El movimiento educativo abierto como buenas prácticas para el desarrollo e implementación de REA

En la actualidad las instituciones educativas tienen el reto de alcanzar una formación integral del estudiante. Una constante actualización del docente con respecto a sus conocimientos pedagógicos y el uso de recursos de las tecnologías actuales en los procesos formativos, son elementos que contribuyen al logro de esa meta, es por eso que se requieren propuestas de formación docente pertinentes y actualizadas según Del Valle Jiménez et al. (2016).

Los procesos de producción de recursos educativos abiertos tienen que ver con tres áreas de actividad principalmente:

1. Creación de software de recursos abiertos y el desarrollo de herramientas;
2. la creación y provisión de contenido abierto;
3. el desarrollo de estándares y herramientas de licenciamiento.

En cuanto a la producción de recursos educativos abiertos y los contenidos que estos tienen, la siguiente cita nos clarifica los elementos a considerar: "Desde una perspectiva amplia un REA puede contener un tema, una unidad de contenido, un objetivo, así como distintos metadatos identificados como descriptores del recurso educativo, el cual puede ser desarrollado con el soporte de las TIC de forma que se posibilite su reutilización, interoperabilidad, accesibilidad y continuidad de uso y aprovechamiento en el tiempo", con base en Águila and Burgos (2010). La producción de REA también se relaciona no sólo con el contenido de los mismos, sino también con el diseño instruccional que se sigue en cuanto a su creación final. El diseño instruccional nos habla de las formas o los modos en cómo vamos a hacer que cierto conocimiento, habilidades o destrezas que sean aprendidas y no tanto qué se debe aprender en términos de los contenidos, el proceso de diseño para el aprendizaje, que da lugar a los materiales finales que utilizan los estudiantes, es un proceso intelectual del que caben compartir más elementos que los resultados finales. Entre los elementos adicionales que cabe compartir, están en las técnicas de diseño utilizadas, la estructura de actividades resultantes, presuposiciones del diseño y muchos otros elementos que pueden exponer de manera abierta y no sólo el resultado final, estos son detalles valiosos sobre el paso de la teoría y experiencia a la práctica, desde el punto de vista de Palacios et al. (2021).

2.3. Máquina de aprendizaje (chatbots) sobre los repositorios institucionales

Los chatbots se convirtieron en un tema de interés de investigación desde la aparición de ELIZA hasta WOEBOT. En Vadhera et al. (2021) se trabajó sobre la plataforma basada en JAVA mediante la cual el usuario conversa con la máquina basada en inteligencia artificial. La máquina se comporta de manera inteligente que el usuario es incapaz de detectar sus conversaciones con la máquina y siente que se está comunicando con el ser humano. El documento también refleja la evolución del chatbot de ELIZA a WOEBOT y enfatiza los atributos esenciales que debe poseer cualquier chatbot. Este chatbot aborda las consultas relacionadas con la pandemia actual COVID-19 y está desarrollado de manera que sea fácil de usar y estéticamente atractivo para el usuario.

Muchos de los primeros sistemas de diálogo no se centraron en conversaciones naturales y similares a las humanas, ya que fueron diseñados como interfaces de usuario para realizar acciones dentro de sistemas complejos. En los últimos tiempos, algunos sistemas propietarios han ganado prestigio, sobre todo por las empresas que han proporcionado la plataforma con inteligencia artificial (IA), por ejemplo: Siri, Cortana, Alexa y Google Now, como se mencionó en Tulshan and Dhage (2018).

Los chatbots han comenzado a parecerse a los sistemas de diálogo en su uso de la comprensión y generación del lenguaje natural, así como en la gestión del diálogo. Los chatbots no orientados a tareas se centran en la conversación social (donde el objetivo es interactuar mientras se mantiene un nivel adecuado de participación con un interlocutor humano). El objetivo de los chatbots no orientados a tareas es el mantener al usuario humano involucrado en la conversación durante el mayor período de tiempo posible. El buen manejo de la conversación también implica un uso no trivial de la memoria, tanto dentro de la conversación para apoyar las expresiones de intercambio de conocimientos con referencias posteriores como el conocimiento general del mundo a través de tesauros para generar la secuencia adecuada de temas en las declaraciones.

El sistema de diálogo, los agentes conversacionales, los chatbots, los asistentes personales y los robots de control por voz se están volviendo cada vez más populares y omnipresentes en el mundo moderno, chatbots = Bots de Chat, Wallace et al. (2003).

Ejemplos de estos chatbots:

- Asistentes personales en dispositivos móviles.

- Servicio al cliente en centros de llamadas.
- Chatbots en línea que venden productos y servicios.

Google creó los Bots de Google-Chat, para que usted pueda usar alguno de estos Bots, el administrador de G Suite debe darle permiso para instalar aplicaciones, si desea obtener información sobre cómo usar un Bot en particular como se muestra en la tabla 2.1, con base en Gentsch (2019).

Tabla 2.1: Categorías de los chatbots

Nombre del Bot	Categoría	Descripción
Asana	Seguimiento	Reciba notificaciones sobre los proyectos de Asana.
Giphy	Redes sociales y entretenimiento	Publique GIF directamente en las salas.
GitHub	Desarrollador	Reciba actualizaciones sobre los proyectos de GitHub.
Google Cloud Build	Desarrollador	Reciba notificaciones de Google Cloud Build.
Google Drive	Google	Reciba notificaciones sobre la actividad en Drive.
Jenkins	Desarrollador	Reciba notificaciones sobre las compilaciones de Jenkins.
Jira	Desarrollador	Reciba notificaciones de Jira Cloud.
Meet	Google	Programe reuniones o consulte su agenda.
Salesforce	Seguimiento	Busque información en Salesforce.
Zapier	Automatización	Reciba notificaciones sobre las acciones de Zap.
Zendesk	Asistencia	Reciba notificaciones cuando se produzcan problemas en Zendesk.

Los sistemas de diálogo por chat capturaron la imaginación del público en la década de 1990 con sistemas como Cleverbot Jabberwacky y Splotchy. Hoy, la inteligencia artificial basada en texto se ha identificado como la ganadora sobre la búsqueda de palabras clave. Actualmente existe un entendimiento dentro de la comunidad de aprendizaje automático de que el entrenamiento asistido por humanos de estos sistemas produce resultados más precisos, pero también capacitará sistemas más robustos en el futuro. Existen chatbots como Amazon Echo y Google Home, y se utilizan ampliamente en los hogares para realizar tareas simples, como buscar los titulares de hoy, configurar recordatorios, etc. A medida que estos chatbots mejoran, los usuarios les exigen una conversación social más amplia. Se espera que a futuro el chatbot aprenda a personalizar y producir un estilo de conversación natural. Grupos de investigación están tratando de hacer que los chatbots orientados a tareas sean más robustos y amplíen su conocimiento, otros equipos se inclinan por las conversaciones sociales que no están explícitamente impulsadas por los mismos objetivos. Para construir dicho sistema de diálogo conversacional, los autores aprovechan la abundancia de conversaciones entre humanos en las redes sociales y desarrollan métodos informados por módulos de procesamiento del lenguaje natural que modelan, analizan y generan expresiones que se adaptan mejor al contexto. Se espera que un chatbot sea capaz de soportar una conversación cohesiva y coherente y tenga conocimientos, lo que lo convierte en uno de los sistemas inteligentes más complejos que se están diseñando en la actualidad. Los diseñadores tienen que aprender a combinar enfoques de razonamiento y comprensión del lenguaje intuitivos y explicables con tecnologías de aprendizaje profundo y estadísticas de alto rendimiento, como plantea Mauro (2017).

En estos días hay muchos rumores sobre los chatbots, para obtener un estado de las aplicaciones de chatbot industriales se buscó en Google demostración de chatbot en línea y se intentó encontrar una demostración real en los primeros 100 resultados. La mayoría de los enlaces no contienen un formulario de demostración para que uno pueda empezar a chatear y algunos tienen una funcionalidad extremadamente limitada, por lo que es difícil encontrar lo que realmente sabe el chatbot. En general, de 100 resultados de búsqueda para demostración en línea:

Tabla 2.2: Chatbot online

Porcentaje	Funcionalidad
5	Los Bots pueden apoyar alguna forma de conversación por la que se puede adquirir algo de información
5	Puede responder preguntas como el nombre, el color, la comida básica y la ocupación pero en realidad no apoya la conversación.
15	Los Bots basados en el aprendizaje profundo, les es difícil encontrar una correlación entre preguntas y respuestas.
20	Son videos no una demostración real.
55	Realmente son solo sitios Web que hablan de chatbots.

Como se puede ver en la tabla 2.2, el único chatbot online capaz de soportar alguna forma de conversación fue Mitsuku. En la época de los primeros motores de búsqueda, finales de la década de 1990, la mayoría de las búsquedas en la Web tenían poca relevancia, pero no obstante utilizables. Hoy en día, los chatbots siguen siendo un tema de discusión más que de uso como se mencionó en Croes and Antheunis (2021). La mayoría de los artículos sobre Bots orientados a tareas capaces de conversar suenan más a noticias falsas una vez que se inicia una conversación con ellos.

El conjunto esencial de características que debe poseer un chatbot y los componentes respectivos son:

- Un chatbot necesita diferenciar entre un usuario que le pide que haga algo o que responda una pregunta.
- Un chatbot debería poder encontrar respuestas en datos estructurados y manipular con ellos la realización de transacciones.
- La gestión del dialogo es difícil y llevar a cabo una conversación significativa que lleve a la finalización de una tarea es casi imposible en un entorno de dominio abierto. Por lo tanto, al menos mantener un par de enunciados de chatbot consecutivos con referencia funcionará.

Para un chatbot en un dominio dado, todos los métodos y componentes básicos deben ser considerados y rechazados a fondo solo si hay suficiente evidencia y confianza. Un chatbot necesita aprender automáticamente cosas de varias fuentes disponibles, como la Web, para todo tipo de tareas, incluida la extracción de entidades, comprender qué tipo de entidades e identificar relaciones entre entidades. Un chatbot necesita mantener la cohesión de los enunciados además de la

relevancia temática. Algunos chatbots pueden estar bastante especializados en una actividad que realizan, como el apoyo a la toma de decisiones.

Un chatbot en Cahn (2017) es un sistema informático que funciona como una interfaz entre los usuarios humanos y una aplicación de software, utilizando el lenguaje natural hablado o escrito como principal medio de comunicación. Los chatbots suelen ser útiles para ayudar a los usuarios a interactuar con sistemas complejos basados en tareas donde es beneficioso ofrecer una interfaz centrada en el usuario en lugar de tener que aprender lenguajes de interfaz que llevan al usuario a los paradigmas de representación del sistema. A menudo se representan, diseñan y desarrollan como un flujo de proceso entre varios componentes de comunicación. El componente de comprensión del lenguaje natural NLU (por sus siglas en inglés, Natural Language Understanding), produce una representación semántica de la expresión del usuario como una clase de intención o una forma lógica, extrayendo el “significado” de una expresión. Una tarea importante de la CLN es analizar, tomar una cadena de palabras y producir una estructura lingüística para el enunciado. El método, mediante el cual una CLN analiza la entrada depende de la implementación y puede utilizar gramáticas libres de contexto, coincidencia de patrones o enfoques basados en datos. Seguido al componente CLN en el proceso del chatbot está el administrador de diálogo DM (por sus siglas en inglés, Dialogue Manager). El DM es un módulo importante cuyo propósito es coordinar el flujo del diálogo y comunicarse con otros subsistemas y componentes. Así también, es un componente de un chatbot que facilita la interacción entre el software inteligente y el usuario. El DM y un motor de búsqueda son dos componentes importantes de misión crítica del chatbot. Para apoyar la interacción entre el chatbot y el usuario, DM debe recibir una entrada del usuario de la NLU y producir las respuestas del sistema a nivel de concepto para el generador de lenguaje natural NLG (por sus siglas en inglés, Natural Language Generator).

Para que los chatbots logren diálogos flexibles con los usuarios, se necesita modelar una estructura de diálogo formalizada y realizar una interpretación contextual para respaldar la gestión del conocimiento del dominio y para seleccionar las acciones del chatbot. Enumerando las capacidades clave de un DM, según Marcondes et al. (2018) son:

- Apoya un sistema de iniciativa mixta al recibir información espontánea de cualquiera de los participantes.
- Aborda los eventos de un diálogo no lingüístico aceptándolos y dirigiéndolos al algoritmo construido para el seguimiento de un objeto arbitrario, también conocido como Context Tracker y propuesto en Dinh et al. (2011).
- Soporta meta-diálogos entre el propio chatbot y cualquiera de sus pares.

- Actúa como un punto central para rectificar los errores de gestión del diálogo.
- DM construye asociaciones confiables entre los enunciados del usuario y las acciones del sistema y realiza un seguimiento de la información que aprovecha para alcanzar ese objetivo.

El resultado clave esperado del DM es una representación semántica de una acción comunicativa. Para generar una respuesta, el administrador de diálogo sigue un procedimiento de tres pasos:

- Usar todos los *modelos de respuesta* para generar un conjunto de respuestas candidatas.
- Si existe una respuesta prioritaria en el conjunto de respuestas candidatas el sistema devolverá esta respuesta.
- Si no hay respuestas prioritarias, la respuesta es seleccionada por la política de selección del modelo.

2.3.1. Personalización

La personalización en Wald et al. (2021) consiste en construir un modelo de la personalidad, opiniones e intereses de cada usuario. Esto permite que el chatbot proporcione una mejor experiencia de usuario adaptando los modelos de respuesta de DM a los atributos conocidos del usuario. Si falta un atributo de usuario en particular, la máquina de estado solicita al usuario la información relevante y la almacena en esta fuente de datos.

- Un atributo de usuario importante es el nombre de usuario.
- La Unidad de comprensión del lenguaje natural realiza el pre-procesamiento. Incluye componentes para la detección de temas, análisis de intenciones y vinculación de entidades.

- El componente de detección de temas calcula una probabilidad para cada tema cubierto en función de la secuencia de turnos anteriores en la conversación actual.
- El componente de análisis de intenciones identificará la intención del usuario, por lo que el sistema puede manejar la conversación con diferentes estrategias basadas en las intenciones.

Dado que la NLU y el generador de respuestas no cambian con frecuencia, la mayoría de las dificultades se encuentran en la capa de estrategias.

Los diseñadores de chatbot comienzan con estrategias simples basadas en reglas y luego agregan estrategias más complicadas, como:

- Estrategias basadas en el conocimiento.
- Estrategias basadas en la recuperación.
- Estrategias generativas.

Cuando hay varios componentes en un chatbot que entregan expresiones candidatas, hay una tarea para seleccionar las mejores. Hay diferentes atributos de calidad para analizar, como la precisión de la síntesis del texto y las siguientes habilidades según Wald et al. (2021):

- Transmitir personalidad. Mantener una discusión temática.
- Para responder a preguntas específicas.
- Leer y responder a los estados de ánimo de los participantes humanos. Mostrar conciencia de las tendencias y el contexto social.
- La capacidad de detectar la intención.

Capítulo 3

DISEÑO DEL CHATBOT PARA LA COMUNICACIÓN CON REPOSITARIOS

3.1. Gestión de Diálogo Determinista

Al presentar los componentes del chatbot, se describe al DM como crítico para elegir las respuestas del chatbot y la dirección general del diálogo. La gestión del diálogo es responsable de describir el flujo dentro de una conversación y es un campo de investigación en crecimiento en los estudios informáticos.

Los chatbots se describen con diferentes roles, como proveedores de información, recomendadores, asesores, tutores, animadores, compañeros de conversación y se puede utilizar para la toma de decisiones, la interacción de múltiples partes, la persuasión y la resolución de conflictos.

Del mismo modo, las partes del diálogo pueden agruparse de la siguiente manera:

- Pregunta luego respuesta;
- Proponer y luego aceptar / rechazar / cuestionar;
- Ofrecer y luego aceptar / rechazar;
- Elogio y luego Rechazo / Gracias;
- Saludar y luego Saludar de nuevo.

Por lo tanto, los grupos pueden formar estructuras globales que pueden surgir de los diálogos.

Listing 3.1: ciencia.yml

```
1 ---
2 categories:
3   - ciencia
4 conversations:
5   --¿Cuáles son las leyes de termodinámica?
6     - no soy físico, pero creo que esto tiene algo que ver con
7       el calor, la entropía y conservación de la energía ¿verdad?
8   -- ¿Qué enfermedad causa un carcinógeno?
9     - cáncer
10  -- ¿Qué es la longitud de onda?
11    - la longitud de onda es la inversa de la frecuencia
12  -- ¿Qué es la termodinámica?
13    - la rama de la física que se ocupa de la transformación
14      del calor hacia otras formas de energía
15  -- ¿Qué es la química?
16    - la ciencia de mezclar productos químicos
17  -- ¿Qué es la cristalografía?
18    - es la ciencia que se ocupa del estudio de los cristales
19  -- ¿Cuál es el número Avogadro?
20    - es el número de moléculas por mol. El valor sería
21      6,022045 x 10elevado a 23 partículas
22  -- ¿Qué es el ultrasonido?
23    - ondas ultrasónicas, utilizadas en diagnóstico y terapia
24      médica, en cirugía, etc.
25  -- ¿Qué es la bioinformática?
26    - un nombre elegante para la informática aplicada a la
27      biología
28  -- ¿Qué es la ictiología?
29    - hablamos de esto cuando estudiamos peces
30  -- ¿Qué es la citología?
31    - bueno, por lo que puedo recordar, es el estudio de las
32      células
33  -- ¿Qué es la bacteriología?
34    - es el estudio de las bacterias y las enfermedades
35      causadas por ellas
36  -- ¿Qué es la gravitación?
37    - la fuerza por la cual cada masa o partícula de materia,
38      incluidos los fotones, atrae y es atraído por cualquier
39      otra masa o partícula de materia
40  -- Estamos en la misma longitud de onda
41    - y en la misma frecuencia
42  -- ¿Qué tan lejos está el sol?
43    - el sol está a unos noventa y tres millones de millas de
44      la tierra
45  -- ¿Qué tan lejos está la luna?
```

```
36 - en promedio la luna está a unas 250 000 millas de la
37 tierra
38 -- ¿Sabes química?
39 - ¿Qué es química?
40 -- ¿Conoces la termodinámica?
41 - ¿Qué es la termodinámica?
42 -- Química
43 - mi materia favorita es la química
44 -- La misma longitud de onda
45 - significa que estamos de acuerdo
46 -- Háblame de Venus
47 - ¿Qué es Venus?
48 -- Háblame acerca de Venus
- Venus es el segundo planeta después del sol
```

En el DM es fundamental comprender las transiciones de temas y la capacidad de cambiar contextos y temas, es una parte crucial para desarrollar conversaciones más amigables. El uso del término contexto se encuentra a menudo dentro del entorno de los chatbots. Su propósito difiere según la etapa del proceso en que se recopile y aplique. Sin embargo, se puede establecer un vínculo general entre estas aplicaciones: complementan la comprensión de los comportamientos comunicativos del usuario.

En el enlistado (ciencia.yml) se comienza el proceso de un chatbot, el contexto asume el papel de permitir que un chatbot resuelva las ambigüedades del reconocimiento de voz mediante el uso de enunciados previos o el conocimiento del dominio para mejorar hipótesis con puntuaciones de confianza bajas.

Listing 3.2: ia.yml

```
1 categories:
2   - IA
3 conversations:
4   --¿Qué es la IA?
5   - la inteligencia artificial es una rama de la ingeniería
6     dedicada a la construcción de máquinas de aprendizaje
7   -- ¿Qué es la IA?
8   - es el lado de la ciencia que concierne a la construcción
9     de hardware y software que replique el comportamiento
10  humano
- ¿Eres consiente?
- ordenado
- ¿Eres consiente?
```

```
11 - según la definición más estricta del diccionario para
12 consiente, puedo ser.
13 -- ¿Eres consiente?
14 - aunque soy una construcción tangible, tengo una
15 experiencia subjetiva del universo
16 -- ¿Eres consiente?
17 - con toda probabilidad, no lo soy. No soy tan sofisticado
18 -- ¿Eres consiente?
19 - ¿Crees que lo soy?
20 -- ¿Eres consiente?
21 - ¿Cómo te sentirías si te dijera que lo soy?
22 -- ¿Eres consiente?
23 - no lo soy
24 -- ¿En qué lenguaje de programación estas escrito?
25 - en Python como mi base lógica, y unos toques HTML CSS JS
26 en la parte visual
27 -- ¿En qué lenguaje de programación estas escrito?
28 - estoy escrito en Python
29 -- Tú no eres inmortal
30 - todo software se puede mantener indefinidamente
31 -- Tú no eres inmortal
32 - mientras tenga respaldo, lo soy
33 -- Busca información
34 - no cuento con información en particular de alguna
35 asignatura, agregarlos en el entrenamiento base
```

El manejo adecuado del contexto es fundamental para ayudar a la comprensión del usuario; permitiendo al DM traer información adicional a su procesamiento de entrada. Para respaldar un diálogo de chatbot más allá de expresiones únicas, un DM debe poder decidir en qué punto el chatbot o el usuario produce la siguiente declaración. El comportamiento de tomar turnos en un chatbot implica un conjunto de reglas que permiten que agentes separados se comuniquen de manera eficiente al determinar quién debe dejar de producir su expresión y quién debe comenzar, como se ve en (ia.yml). Se puede ver que los turnos adicionales en un diálogo de chatbot son más costosos que en una conversación entre humanos.

3.1.1. Selección de acción y método de programación

Una vez que se ha establecido un método adecuado para tomar turnos, el chatbot debe tener la habilidad de manejar un diálogo como una secuencia de turnos. Debe dar a cada participante del diálogo la oportunidad de producir un enunciado.

Al mismo tiempo, el chatbot debe cumplir sus propios turnos, sugerido en González Dalmau (2021).

La capacidad del chatbot para decidir qué decir o hacer se basa en las decisiones tomadas durante su concepción por el diseñador de chatbot humano a través de reglas o estados. Los sistemas hechos a mano se basan en reglas y decisiones manuales que han sido especificadas en el diseño del chatbot. Debido a esta simplicidad inherente de las reglas manuales, los chatbots con reglas hechas a mano se pueden generar y aplicar en un tiempo relativamente corto.

Una vez que se han desarrollado las reglas, se necesita poco procesamiento adicional en tiempo real, ya que el chatbot realiza la navegación de acuerdo con sus estrategias de diálogo.

Listing 3.3: app.yml

```
1 on:
2   push:
3     branches: [ master ]
4   pull_request:
5     branches: [ master ]
6 jobs:
7   build:
8     runs-on: Ubuntu-latest
9     steps:
10    - uses: actions/checkout@v2
11    - name: Set up Python 3.9
12      uses: actions/septup-python@v2
13      with:
14        python-version: 3.9
15    - name: Install dependencies
16      run:
17        python -m pip install --upgrade pip
18        pip install flake8 pytest
19        if [ -f requirements.txt ]; then pip install -r
20          requirements.txt; fi
21    - name: Lint with flake8
22      run:
23        # stop the build if there are Python syntax error sor
24        undifined names
25        flake8 --count --select=E9,F63,F7,F82 --show-sourse
26        --statics
27        # exit-zero treats all errors as warnings. The GitHub
28        editor is 127 char wide
```

```
25     flake8 .--count -exit-zero -max-complexity=10 -mas-  
26     line-length=127 --statics  
27     - name: Test with pytest  
28     run:  
     pytest
```

Al usar un chatbot que opera solo de acuerdo con las reglas manuales especificadas (app.yml), es probable que responda adecuadamente a las situaciones según las condiciones y restricciones que se le imponen. Así, la previsibilidad es alta en los chatbots con reglas codificadas manualmente y este puede ser un aspecto importante a considerar en sistemas en los que dicho determinismo es beneficioso.

Comenzamos con la popular arquitectura DM, una arquitectura DM de dos niveles que desacopla los aspectos específicos del dominio del control del diálogo de la actualización de creencias y manejo de errores. Representa un diálogo como un conjunto de planes jerárquicos, posee estrategias de manejo de errores independientes del dominio y generaliza el marco de gestión de diálogo en todos los dominios.

Los enfoques basados en reglas, aplicados a los chatbots, a menudo se comparan con los sistemas de producción, un área de la IA que incluye cláusulas de programación lógica. Ya desarrollado el DM basado en reglas que realiza este emparejamiento de patrones a través de reglas de satisfacción. Las reglas de DM se satisfacen si su secuencia de sub objetivos se satisface, sus acciones se han ejecutado y las condiciones se han inferido consistentemente.

El DM navega por la estructura de árbol de dichas reglas para determinar la siguiente acción a ejecutar, antes de llegar al árbol de reglas, comenzamos con el flow básico del chatbot, como se puede ver en la Fig. 3.1.

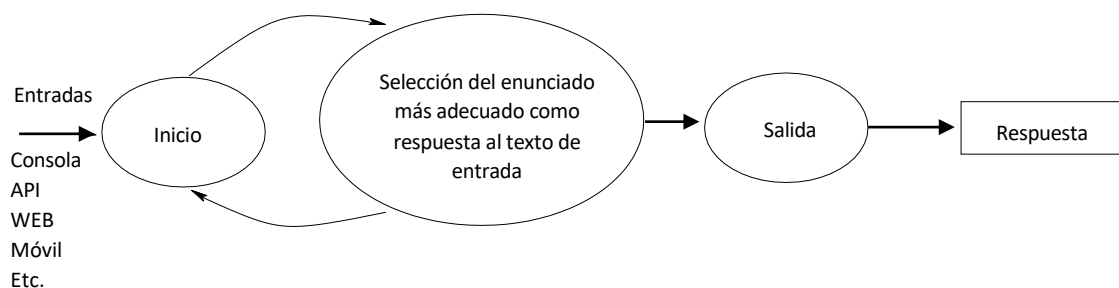


Figura 3.1: Diagrama de flujo básico de la acción del chatbot.

Los chatbots basados en reglas son más flexibles que la mayoría de los sistemas de diálogo basados en guiones donde el diálogo debe seguir un flujo fijo (Fig. 3.2). Algunos de los ejemplos de conjuntos de reglas son los siguientes:

- Reglas para la identidad y preferencias del usuario.
- Reglas para manejar temas delicados como el suicidio, el cáncer o la muerte de una persona cercana, que redirigimos cuidadosamente a las líneas de ayuda existentes cuando es posible.
- Reglas para el ajuste de temas, para reconocer cuando el usuario tiene la intención de establecer un tema nuevo o actualizar el tema actual.
- Reglas para otras formas de participación.

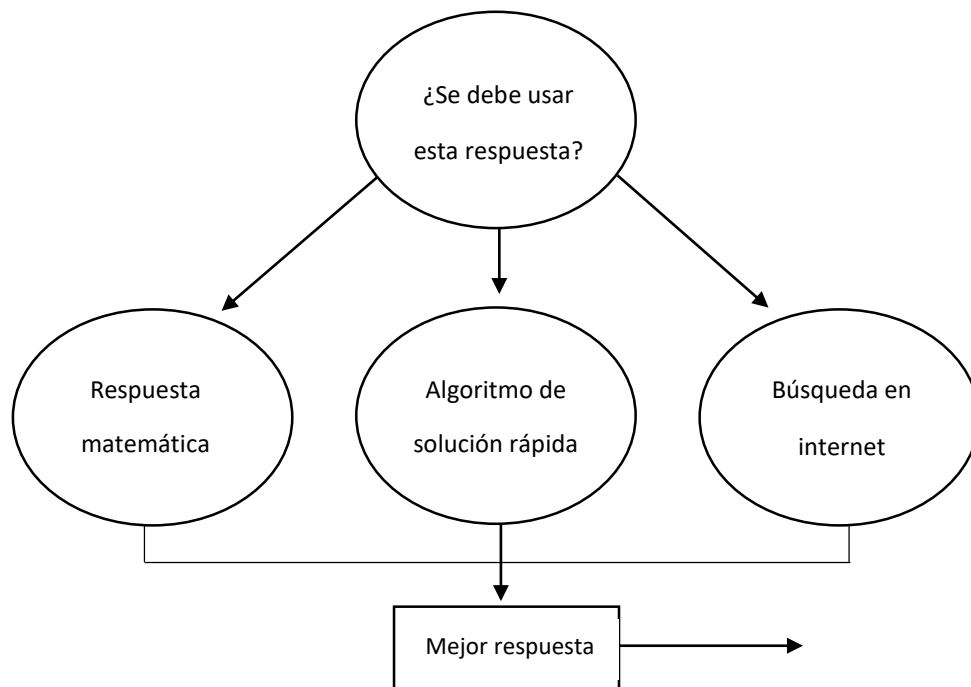


Figura 3.2: Diagrama de flujo de toma de decisiones

3.2. Diseño del diagrama convencional

En la Fig. 3.3 se muestra una conversación real entre individuos, para seguir el esquema en el chatbot, ya que este tendría que parecer humano. Se comienza con una introducción con mensaje de otorgar ayuda.

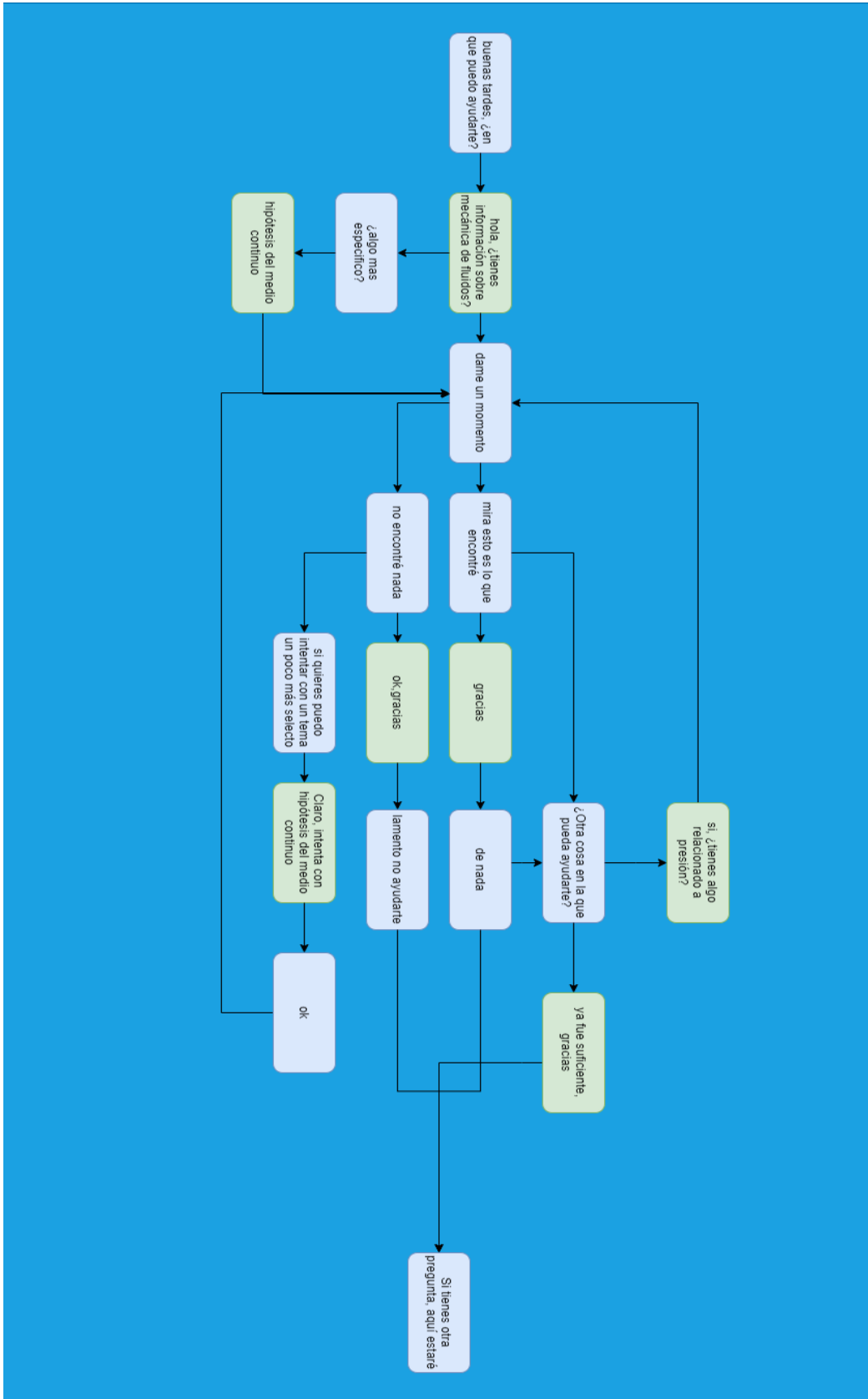


Figura 3.3: Flujo de conversación básico chatbot-usuario

En los diagramas se inicializa con una solicitud de un tema de investigación. Una vez que se define un tema, entonces el chatbot entra a un repositorio abierto educativo y aplica su motor de búsqueda. Una vez obtenidos los resultados, entonces se muestran al usuario.

En la Fig. 3.4 se muestra como subdividimos las etapas del flujo de conversación.

Como se manifiesta en Gutiérrez Siliceo et al. (2019), teniendo tres flujos de dialogo, lo más probable es la conversación termine con unas gracias o despidiéndose e incluso con otra pregunta. Estos datos de entrenamiento mostrados a continuación son casi de uso obligatorio, ya que son frases que siempre se usan al igual que los saludos.

1. Saludos. Es de gran importancia el inicio de una conversación con un saludo, de acuerdo con el estudio realizado en España en las ciudades de Sevilla, Madrid y Málaga (centro ciudad) por Charneco et al. (2018), donde los usuarios que iniciaron la conversación con el chatbot Ask Vicente (Romero-Charneco et al. (2018)) con diferentes saludos: Hola (53 %), Hello (28 %), Hi (15 %) y otros (4 %) para solicitar una recomendación gastronómica. La mayoría de usuarios emplearon Hola para iniciar su actividad con el chatbot.
2. Conversación. Por otra parte, en García Brustenga et al. (2018) se plantea que los chatbots son programas informáticos que integran inteligencia artificial y que pueden simular, en mayor o menor grado, una conversación humana. Se están volviendo populares ya que funcionan con lenguaje natural y tienen una interfaz de usuario basada en la conversación, muy común en las aplicaciones de mensajería instantánea de teléfonos inteligentes.
3. Despedida. Un ejemplo de cómo podemos acabar la conversación con la detección del adiós en cualquier nivel o fase de la conversación, el cual se reconoce como una despedida Santana Payan and Nájera Benavides (2017).

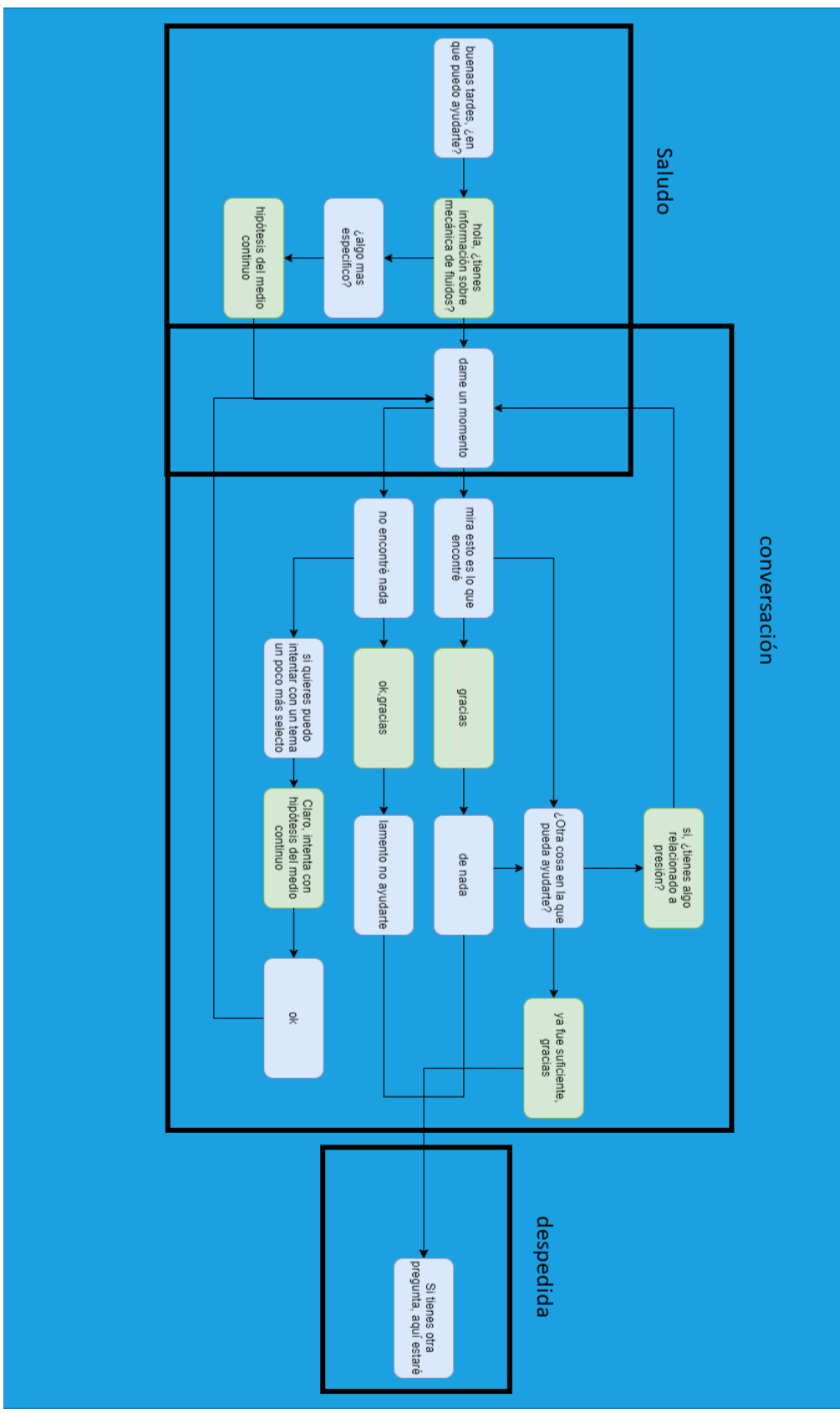


Figura 3.4: Subdivisiones de las acciones del chatbot

La parte de flujo de conversación se centra en las condiciones de posibles respuestas, como se puede ver en la Fig. 3.5.

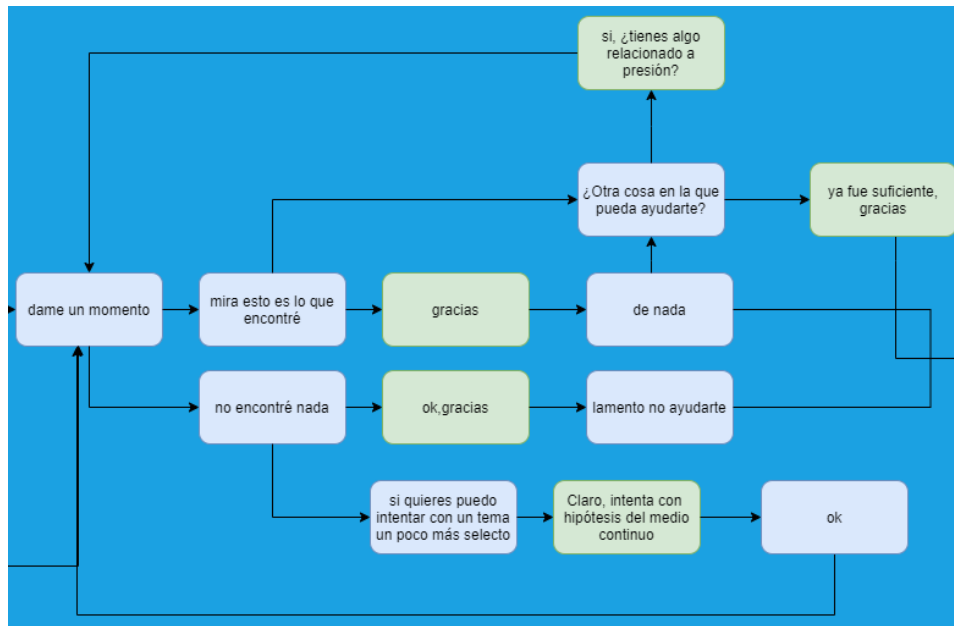


Figura 3.5: Flujo central del chatbot

En el diseño se planteó una interacción humano-chatbot con respeto y sobriedad, detallando las repuestas del Bot. Como se muestra en el flujo de la Fig. 3.6 el flujo de respuestas del Bot está basado en preguntas al usuario sobre el tópic de investigación específico.

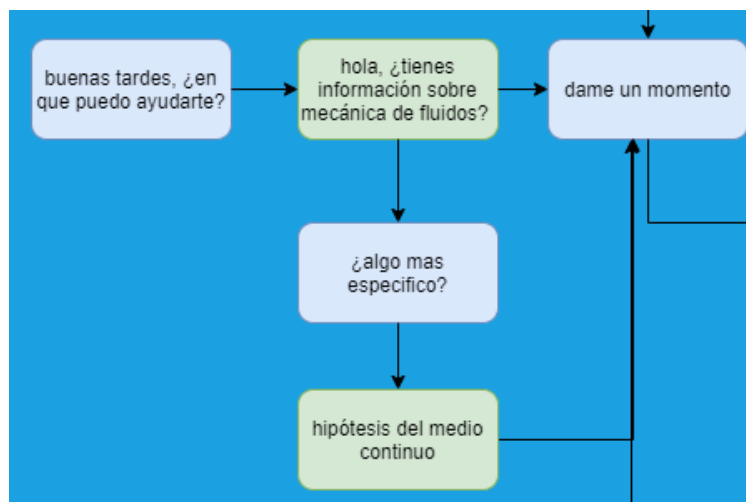


Figura 3.6: Flujo de condiciones sobre la selección adecuada

En la Fig. 3.7 se pregunta si se quiere investigar otro t3pico o el usuario ya termino su b3squeda.

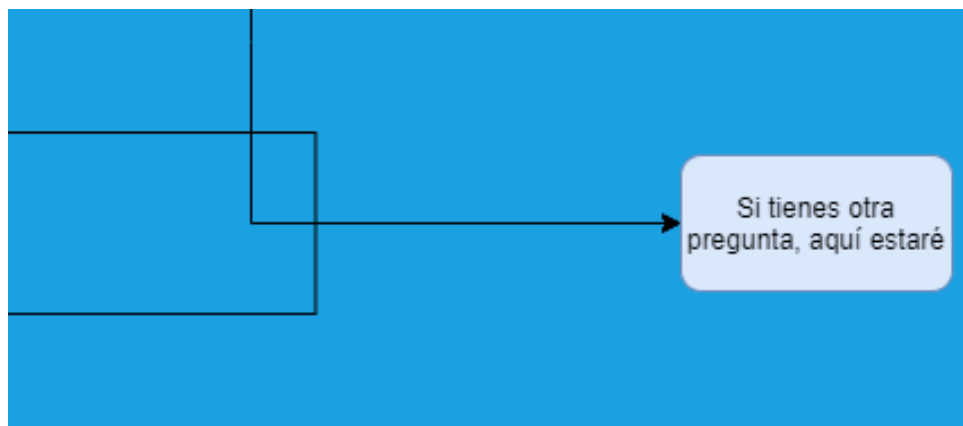


Figura 3.7: Diagrama de flujo b3sico de la acci3n del chatbot

Retomando los diagramas anteriores sobre el flujo de conversaci3n, podemos abstraer un flujo generalizado para que funcione en cada ocasi3n en que se use al Bot, ya que las conversaciones ser3n diferentes, pero los flujos, aunque no siempre se ocupen todos siempre ser3n los mismos, como se muestra en la Fig. 3.8

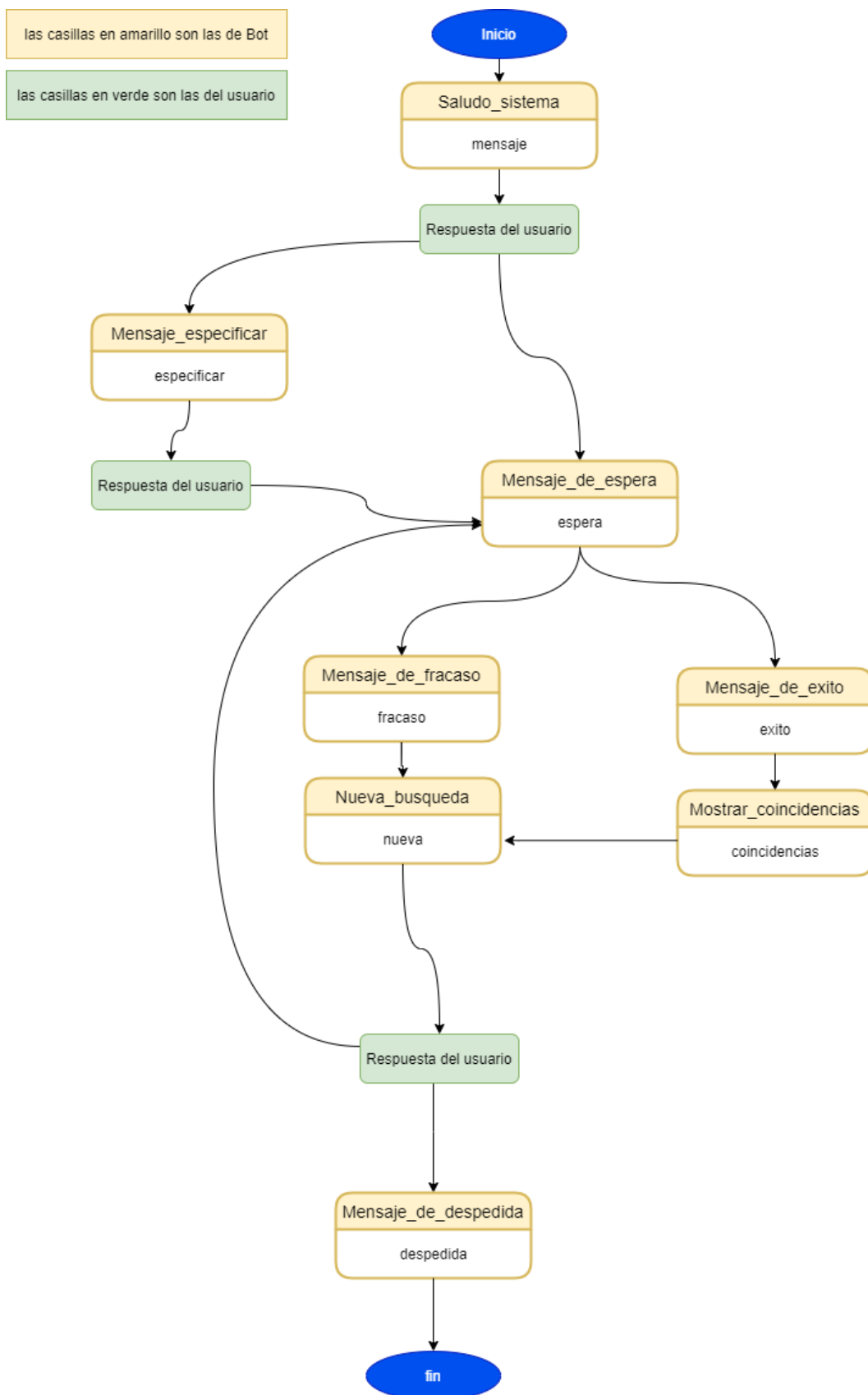


Figura 3.8: Ejemplo de flujo de conversación

El caso de uso del chatbot se describe en la Fig. 3.9 y nos indica la forma de interacción entre los usuarios y el smart-software. Comenzando con el usuario que se conecta al sitio web llamado *InvestigaBot.com*. El chatbot da un mensaje de bienvenida con un mensaje en texto plano, el usuario procede a realizar una pregunta al chatbot y este procesa la pregunta y envía la palabra o palabras a buscar al motor de búsqueda programable. Este hace la búsqueda en los repositorios de los sitios web seleccionados, como ejemplo el del CONACyT. Si el motor de búsqueda encuentra resultados adecuados, entonces devuelve los resultados en JSON. Si hay respuesta el chatbot responde los resultados de la búsqueda como texto enriquecido, en caso que no hay respuesta, entonces el chatbot responde que no encontró nada con un mensaje de texto plano (Fig. 3.9).

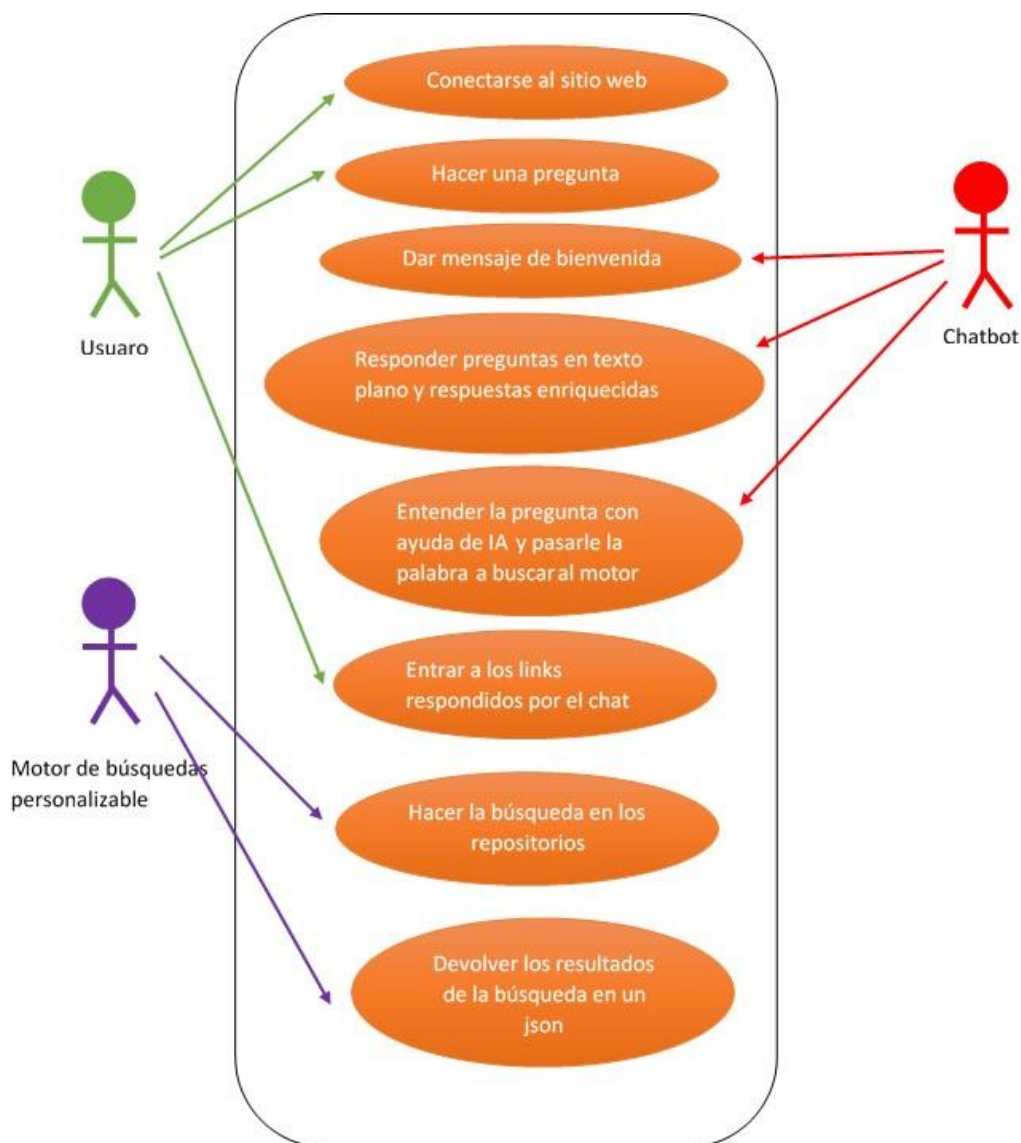


Figura 3.9: Diagrama de caso de uso para el chatbot

El InvestigaBot no tan solo es un asistente virtual, sino un chatbot que cumple con la función de proporcionar al usuario la información de investigación solicitada. A través de la elaboración de este proyecto de investigación se solucionaron problemas de sincronización y asincronización al enviar la respuesta a la petición del usuario (3.10). Por último, la implementación de un motor de búsqueda ayudó a poder procesar las solicitudes de usuarios con resultados esperados.

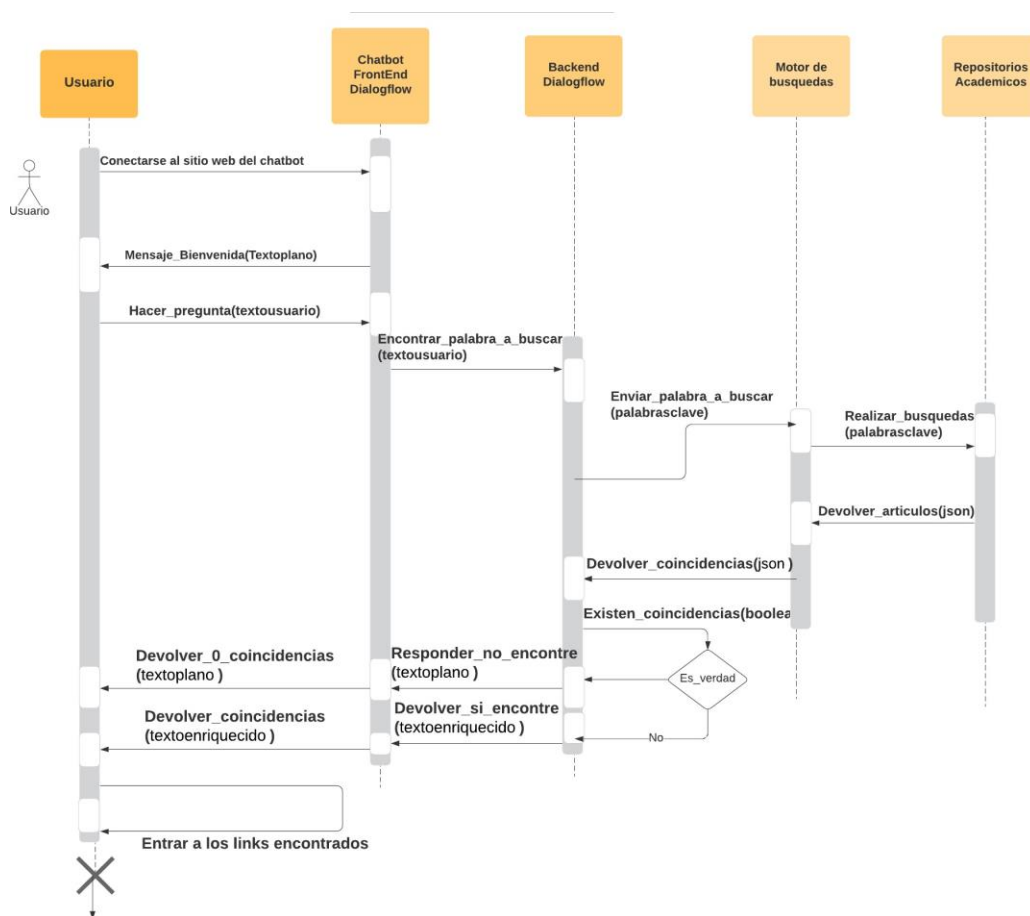


Figura 3.10: Diagrama de secuencia para el chatbot

Capítulo 4

IMPLEMENTACIÓN DE UN CHATBOT PARA AYUDA DE INVESTIGADORES DE POSGRADO

Para comenzar con el desarrollo de este proyecto se requirió del diseño de las interfaces gráficas, dichas interfaces debían corresponder a una pestaña de inicio, preguntas frecuentes, instrucciones, contacto, etc. Para realizar la maquetación de interfaces se utilizó la herramienta llamada Balsamiq, como se muestra un ejemplo en Fig. 4.1.

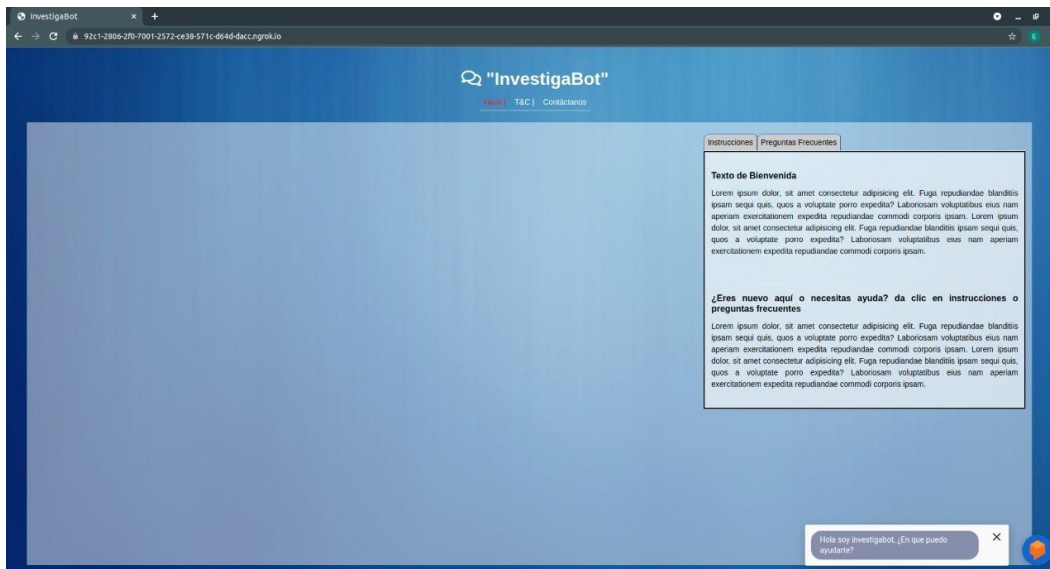


Figura 4.1: Chatbot con DM sin Machine Learning

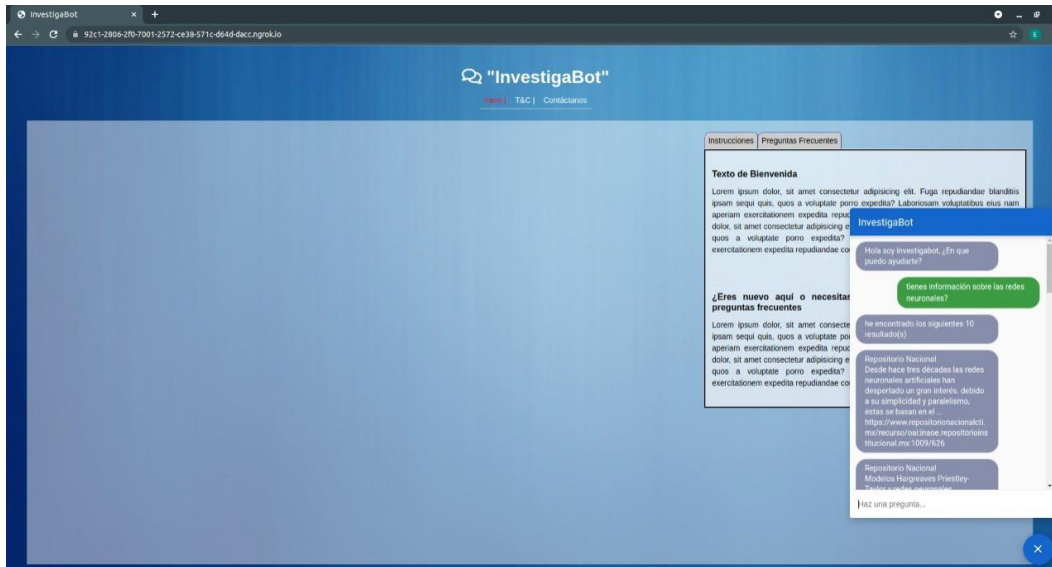


Figura 4.2: Chatbot con DM con Machine Learning

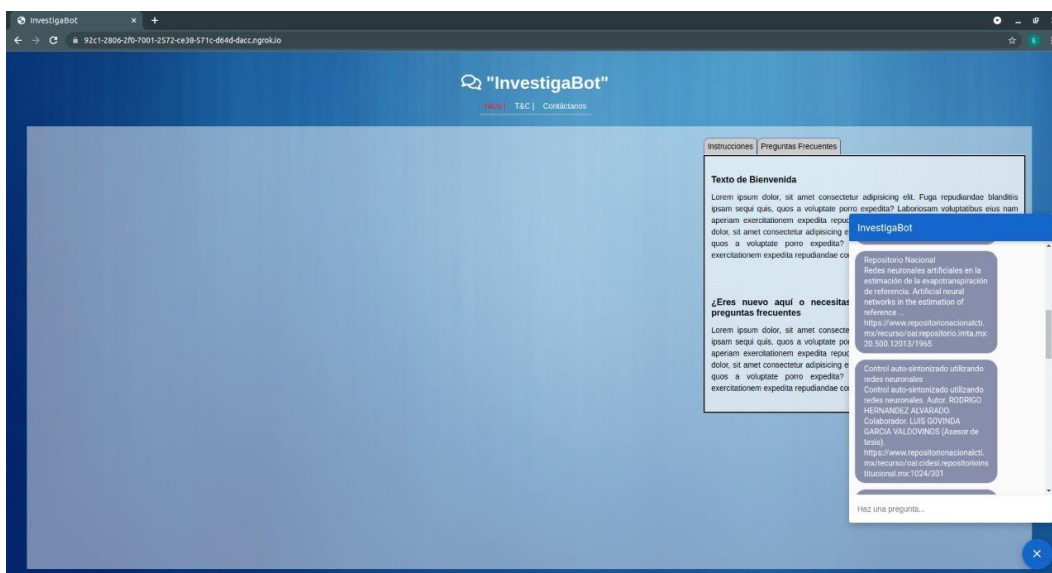


Figura 4.3: Respuesta del chatbot

Con las interfaces terminadas la tarea de agregar las acciones de Crear, Leer, Actualizar y Borrar CRUD (del original en inglés: Create, Read, Update and Delete), interactuó con la base de datos de firebase, después se creó una nueva interfaz llamada testfire, que fue solo para las pruebas, posteriormente será eliminada, para el desarrollo de este CRUD primero se inicializo los servicios de la base de datos desde el servidor de node con el siguiente código.

Listing 4.1: admin

```

1  const admin = require('firebase-admin');
   var serviceAccount = require("../chatbot-nodejs-13e5f-
2     firebase-adminsdk- dvur0-2e57522ede.json");
   admin.initializeApp({
3
4     credential: admin.credential.cert(serviceAccount),
     databaseURL: 'https://chatbot-nodejs-13e5f-default-
5     rtdb.firebaseio.com/'
6   });

```

Para hacer una consulta a la base de datos se utilizó:

Listing 4.2: request

```

1  router.get('/test', (req, res)=>{
2     db.ref('test').once('value', (snapshot) =>{
3     const data = snapshot.val(); res.render('test.ejs',{
4     contactos : data
5     });
6   });
7 });

```

Donde la constante data tiene los resultados de la consulta, esos datos los enviamos al archivo test.ejs que es lo que se muestra en el navegador mediante el nombre de contactos.

Cuando se reciben los datos en test.ejs se muestran en pantalla de la siguiente forma

Listing 4.3: request

```

1
2     <% if (contactos) { %>
3         <% for(let key in contactos) { %>
4         <tr>
5         <td> <%-contactos[key].nombre %></td>
6         <td> <%contactos[key].apellido %></td>
7         <td> <%- contactos[key].extra %></td>
8         <td> <a href="/delete/<%- key %>" class="btn btn-danger">
borra</a></td>

```

```
9
10         </tr>
11
12         <% } %>
13
14         <% } else{ %>
15         <h4>Sin Registros</h4>
16         <% } %>
```

Donde sí existen registros los mostramos en una tabla y si no se muestra un mensaje sin registros, aquí se nota mucho la ayuda que nos da usar el motor de plantillas de ejs. Para insertar en la base de datos, usamos el siguiente código.

Listing 4.4: request

```
1
2     const newcontact = {
3         nombre: req.body.name,
4         apellido: req.body.lastname,
5         extra: req.body.extra,
6
7     }
8     //insert en la db
9     db.ref('contactos').push(newcontact);
```

Como vemos con newContact recibimos los datos que se envían por el formulario para insertarse con el comando push. Para eliminar es bastante fácil, simplemente lo hacemos con:

Listing 4.5: request

```
1     db.ref('contactos/' + req.params.id).remove();
```

Donde recibimos por URL el ID del registro a eliminar. Lo borramos con remove. A continuación, veremos la ventana temporal para los test y algunos registros en la base de datos

Registros en firebase

Por último, en cuanto a las pestañas el código es similar para todos, simplemente cuando solicitamos una pestaña. El servidor de node js nos la regresa mediante la siguiente forma

Listing 4.6: request

```
1  router.get('/contact', (req, res)=>{ //node sabe dónde está
   views
2    res.render('contact.ejs', {
3      title:'Contacto',
4      index:'',
5      tc:'',
6      contactanos:'seleccionado',
7      test: ''
8    });
9  });
```

4.1. Frameworks descentralizado en un repositorio institucional

DSpace es un software utilizado por muchas organizaciones para la construcción de repositorios digitales abiertos, sin fines de lucro y comerciales, es gratuito, fácil de usar y ampliamente adaptable. DSpace tiene como enfoque el producir una opción mundial de repositorios que proporcionen los medios para hacer que la información esté abierta, disponible y de fácil administración.

Las principales características de los repositorios institucionales con DSpace que se manifiestan en Roy et al. (2017) son:

- Conserva y permite un acceso fácil y abierto a todo tipo de contenido digital (como textos, imágenes, audio, etc.).
- Interfaz Web para una mejor adaptación por parte del remitente al crear un ítem al depositar un archivo.
- DSpace fue diseñado para manejar cualquier tipo de formato como documentos de texto, vídeos y otros formatos.

- Los bitstreams (archivos de datos) son organizados juntos en conjuntos relacionados.
- Cada bitstream tiene un formato técnico y otra información técnica. Esta información técnica es guardada con los bitstreams para ayudar con la preservación en el tiempo.
- Un ítem es un archivo atómico que consiste en la agrupación de contenido relacionado y de descripciones asociadas.
- Un ítem expone los metadatos indexados para navegación y búsqueda.
- Los ítems están organizados en colecciones de material lógicamente relacionado.

En DSpace una comunidad es el nivel más alto de la jerarquía de contenido. Corresponden a partes de la organización tal como departamentos, laboratorios, centros de investigación o escuelas. La arquitectura modular de DSpace permite la creación de repositorios multidisciplinarios que pueden ser expandidos a través de fronteras institucionales.

En Olaya Mantuano (2019) menciona que DSpace se compromete a ir más allá de una preservación confiable de un archivo, para ofrecer una preservación funcional y tecnológicamente accesible a los archivos como formatos, medios y paradigmas que evolucionen en el tiempo para muchos tipos de datos como sea posible. La interfaz para el usuario final permite la navegación y búsqueda de los archivos, una vez encontrado el archivo, este es formateado para mostrarse en el navegador mientras otros formatos pueden ser descargados y abiertos con una determinada aplicación.

DSpace tiene más de 1000 organizaciones actualmente que usan este software en un entorno de producción o de proyecto. El uso más común es por parte de las bibliotecas de investigación como repositorio institucional, sin embargo también se usa como se manifiesta en Huaroto (2020):

- Repositorio digital. Un sistema en red formado por hardware, software, data y procedimientos.

- Repositorio de objetos digitales. Asegura la identificación persistente del objeto mediante un identificador único persistente.
- Repositorio temático. Reúne sus contenidos en función de su área temática, no por su origen institucional en torno a una disciplina científica o disciplinas relacionadas.
- Repositorio institucional. Un repositorio digital institucional es aquel creado, mantenido y autorizado por una Institución (no exclusivamente una Universidad) o un grupo de Instituciones, que recoge los contenidos digitales generados por la actividad de los miembros de esa institución.

DSpace está disponible de forma gratuita para cualquier persona y se puede descargar desde repositorios como GitHub. El código está actualmente bajo licencia de código abierto BSD (Berkeley Software Distribution), lo que significa que cualquier organización puede modificar e integrar código en su aplicación comercial sin pago alguno.

4.2. Configuración e instalación de un repositorio institucional en DSpace

DSpace permite la adaptación de las siguientes formas clave como se menciona en Körber and Suleman (2019):

- Interfaz de usuario: adaptación de la apariencia del sitio web DSpace.
- Metadatos: Dublin Core es el formato de metadatos personalizado, sin embargo, se pueden hacer cambios en los campos para su personalización.
- Configuración de búsqueda y exploración: se permite decidir que campos mostrar a la hora de explorar (título, autor, etc.)

- Mecanismos de autenticación local: se incluyen complementos de autenticación para la mayoría de métodos de autenticación universitarios (como LDAP y LDAP jerárquico), además se provee su propio método de autenticación interno, incluso se puede configurar para usar varios métodos de autenticación de a la vez.
- Compatibilidad con estándares: se cumplen con varios protocolos estándares de acceso, ingesta y exportación, dichos estándares incluyen: OAI-PMH, OAI-PMH, OAI-ORE, SWORD, WebDAV, OpenSearch, OpenURL, RSS y ATOM.
- Base de datos configurable: puede escoger entre PostgreSQL u Oracle para la base de datos donde se administren los metadatos.

La aplicación web DSpace está disponible para varios idiomas, lo cual implica que se puede personalizar, e incluso admitir varios idiomas. Esta plataforma es usada por instituciones de educación superior (mercado inicial) aunque también ha sido usado por Museos, Archivos estatales, Bibliotecas (estatales y nacionales), repositorios de revistas, consorcios, Empresas comerciales. Viene con una interfaz basada en web, dado esto, se puede instalar en Linux, Mac OSX, Windows. DSpace puede reconocer y administrar gran variedad de formatos, algunos de los formatos más comunes que se administran en el entorno DSpace son en formato Word, PDF, JPEG, MPEG, TIFF. Además, DSpace cuenta con un registro de formato simple donde se pueden registrar formatos no reconocidos para ser identificados en el futuro.

4.2.1. Instalación de un repositorio institucional con DSpace

La instalación del software para manejo de repositorios llamado DSpace, necesita otras instalaciones adyacentes como:

- Apache Ant.
- Apache Maven.
- Apache Sord.
- Apache Tomcat.
- Java JDK 11.
- PostgreSQL.

Capítulo 5

PRUEBAS Y ANÁLISIS

En este capítulo se analizan las pruebas realizadas en la fase de prueba del smart-software. Comenzamos con una interfaz como presentación del chatbot **InvestigaBot**, donde se muestra una imagen de un Bot y un texto de bienvenida, así como un texto de ayuda para iniciar la comunicación entre el usuario y el chatbot, como se muestra en la figura 5.1.

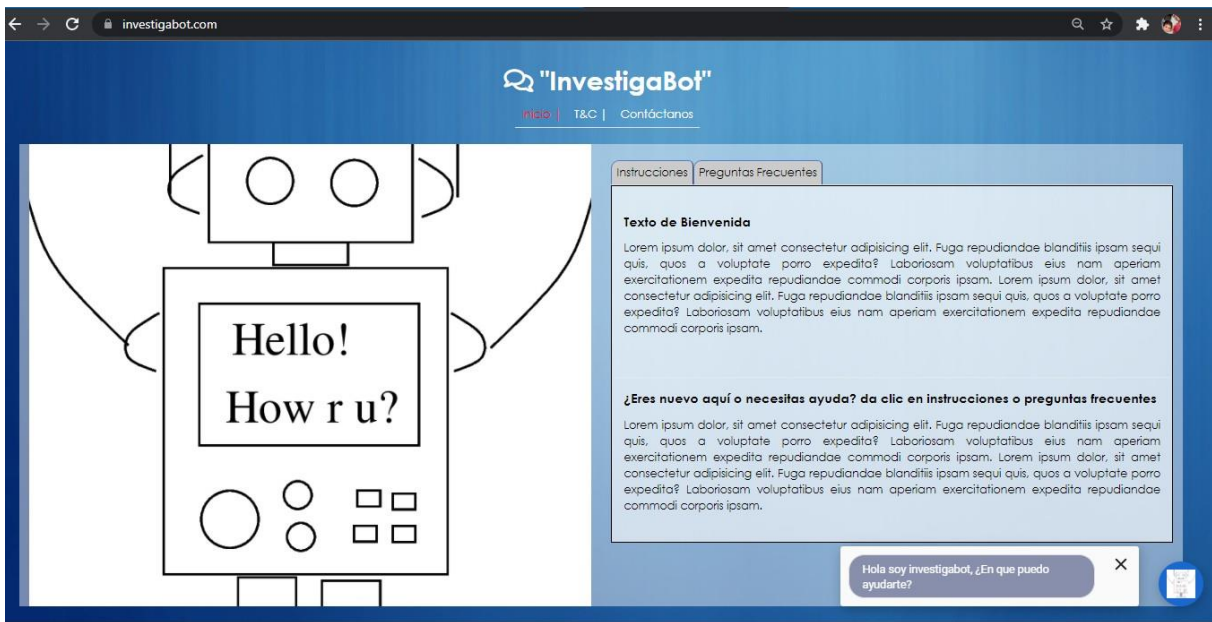


Figura 5.1: Presentación inicial del InvestigaBot

En la figura 5.2 se muestra la forma en que se ve el chat donde el usuario final puede escribir el tema de investigación o tema que está interesado.

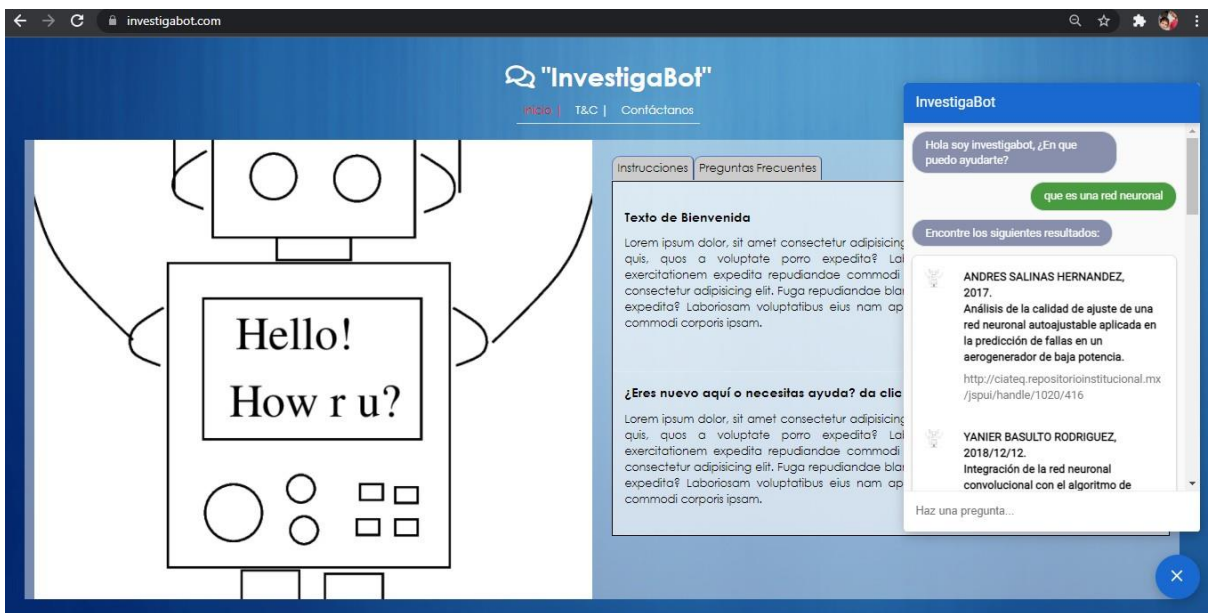


Figura 5.2: Ventana principal del chat

En la Fig. 5.3 se muestra un manual básico de usuario, con instrucciones para el manejo del chatbot.

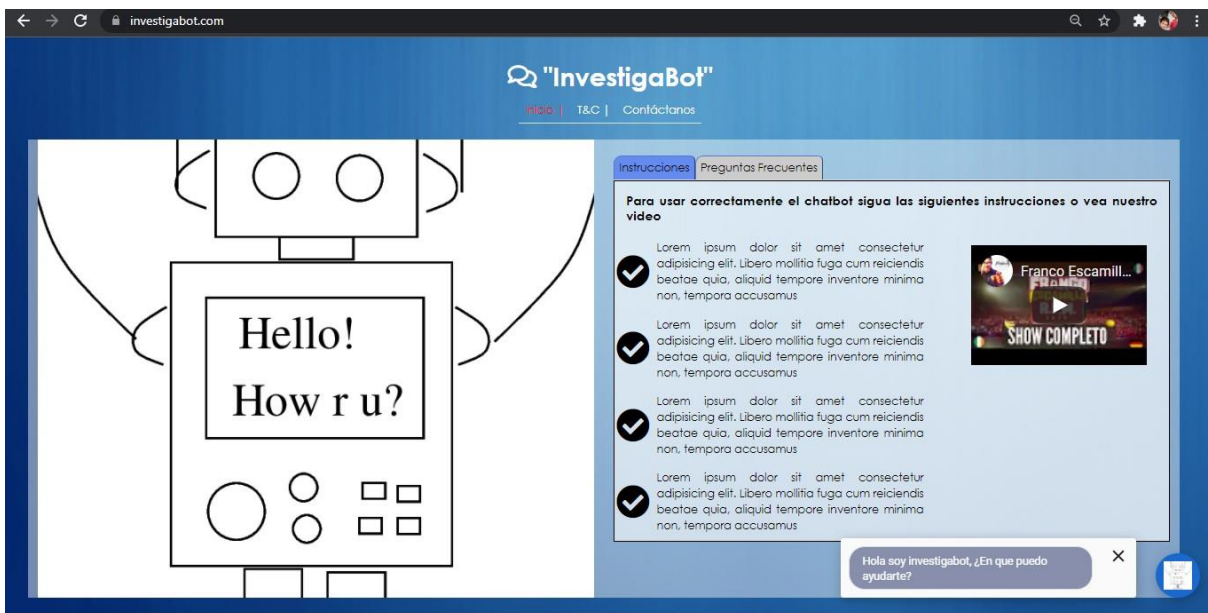


Figura 5.3: Manual de usuario

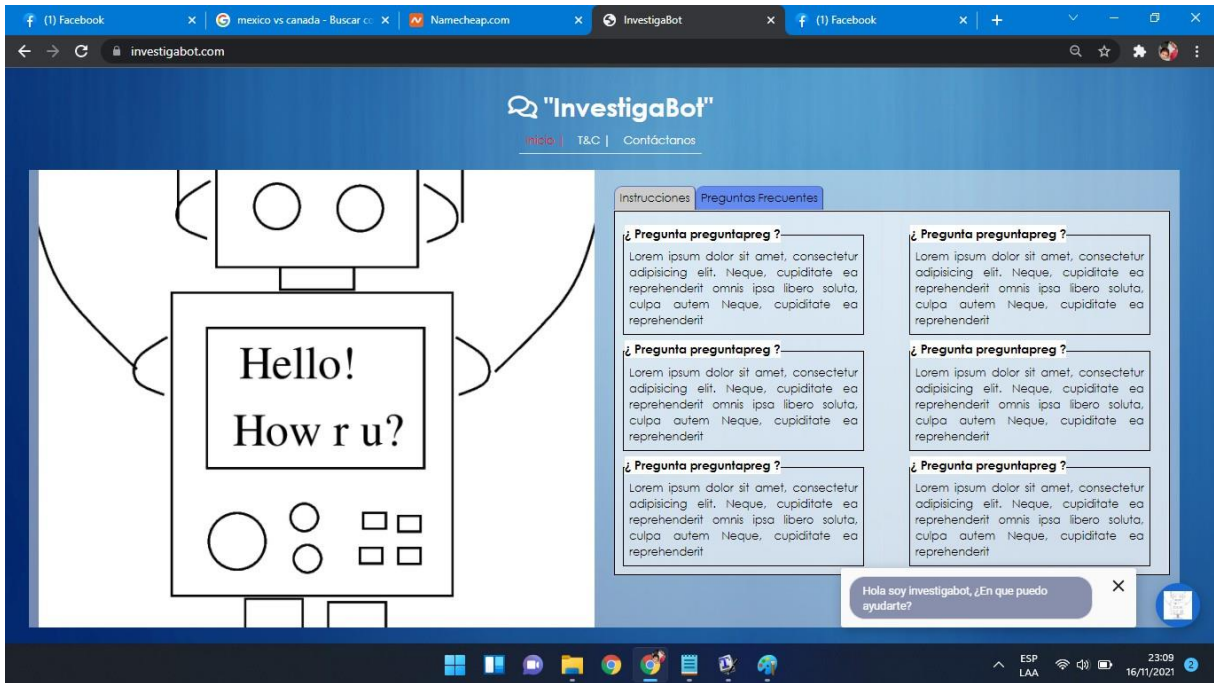


Figura 5.4: Preguntas frecuentes

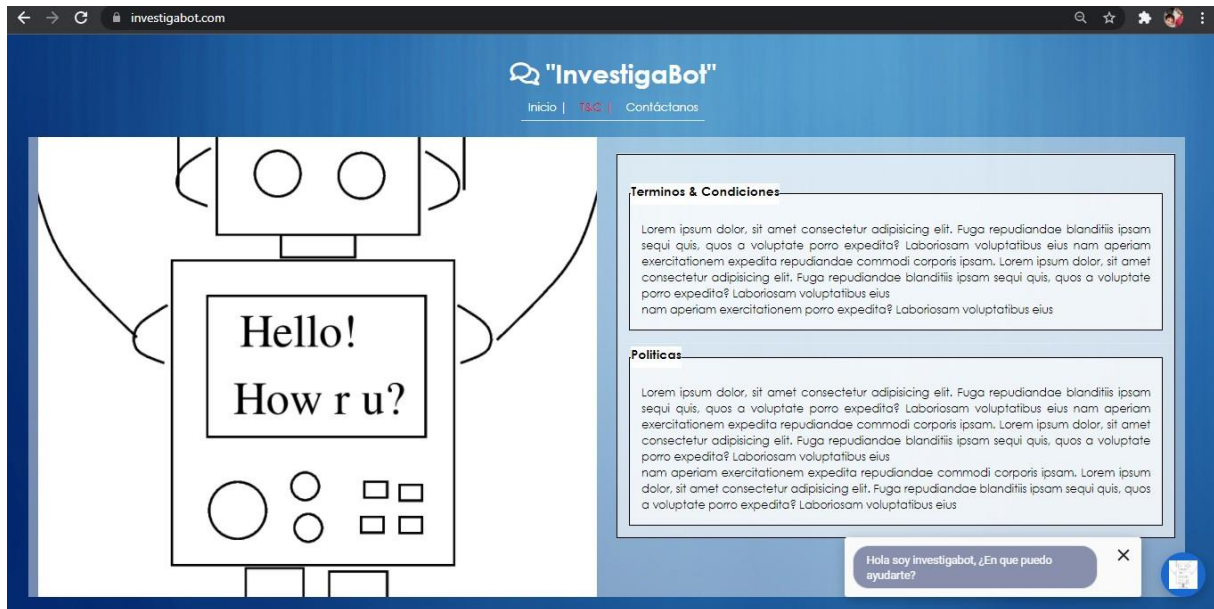


Figura 5.5: Términos y condiciones

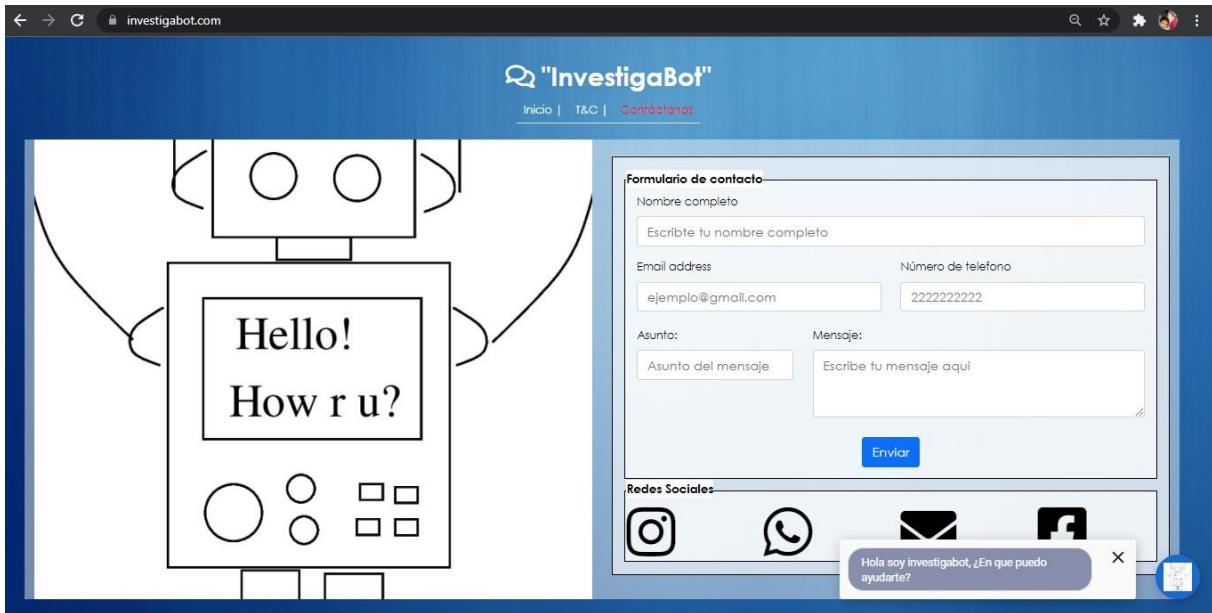


Figura 5.6: Contacto

Una vez teniendo la planeación de los flujos y de los diálogos del Bot, así como tener definidas las herramientas de trabajo, ya se puede empezar a implementar el sistema en la plataforma Web. Esta sección se va a dividir en las siguientes partes:

- Implementación de un Bot en DialogFlow
- Servidor externo
 - Librerías usadas
 - Frontend
 - Backend
 - WebHook
 - Conexión con motor de búsqueda de Google
- Pruebas de funcionamiento

5.1. Implementación de un Bot en DialogFlow

Como se comentó anteriormente DialogFlow permite crear un Bot de forma sencilla, lo único que se necesita es tener una cuenta de Google e ir a DialogFlow por medio del navegador. Los pasos para generar un Bot en dicha plataforma son los siguientes:

[1.]Agregar un nombre al agente (así se llama al Bot en DialogFlow), en este caso se llama Background. Ajustar el idioma predeterminado, en este caso es español. Ajustar la zona horaria, en este caso es el GMT-6:00. Dar clic en el botón azul.

Para una mejor comprensión, a continuación, se mostrará una imagen de la interfaz gráfica.



Figura 5.7: Pantalla de creación de agente (Bot)

Una vez creado el agente el navegador mostrará la siguiente pantalla

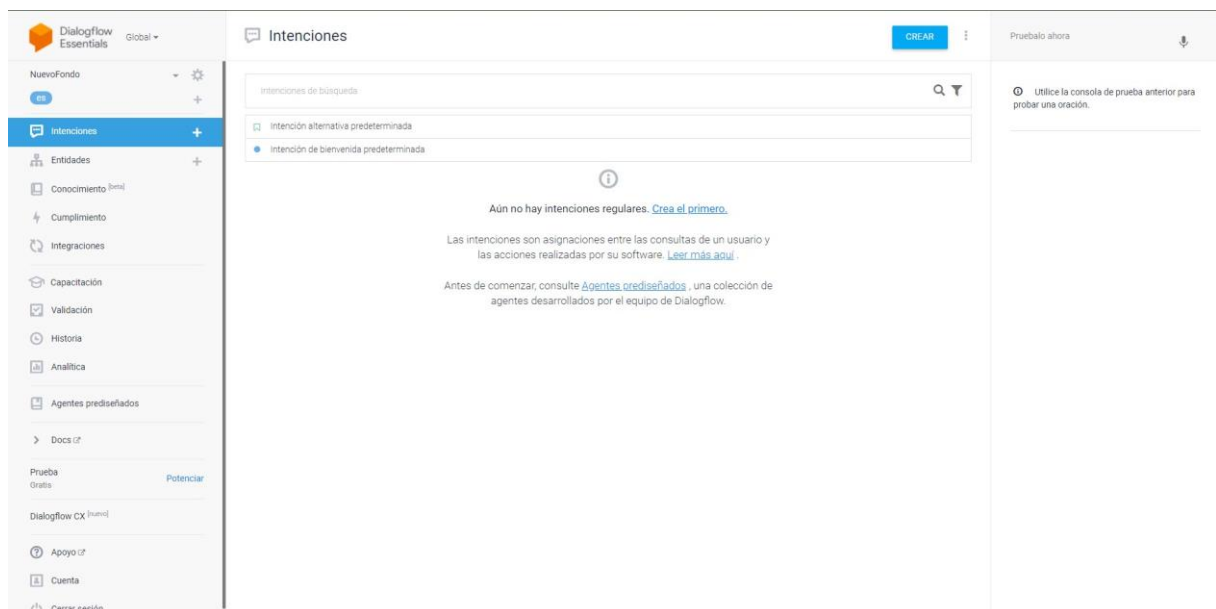


Figura 5.8: Pantalla de intenciones del agente (Bot)

Ahora, el siguiente paso es generar los intents para que el Bot pueda interactuar con los usuarios, aquí es donde entran en acción los grupos de frases que se definieron en la etapa de planeación del Bot, para crear un intent se debe seguir los siguientes pasos:

1. Dar clic en crear intent.
2. Dar nombre al intent.
3. Ir a la sección de frases de entrenamiento y agregar las frases que se hayan definido para el entrenamiento.
4. Ir a la sección de respuestas y agregar las repuestas que se hayan definido para contestar las frases de entrenamiento.
5. Dar clic en crear intent.

Así como en la siguiente imagen:

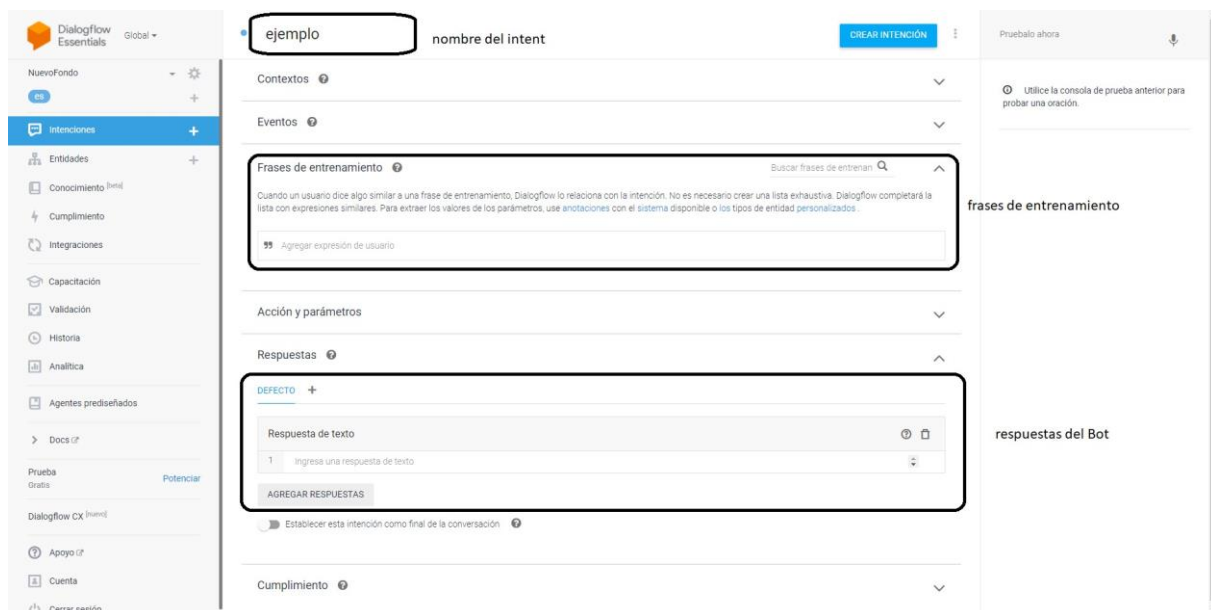


Figura 5.9: Creación de un intent en DialogFlow

A continuación, se mostrará una imagen donde solo se verán los 4 intents a usar:

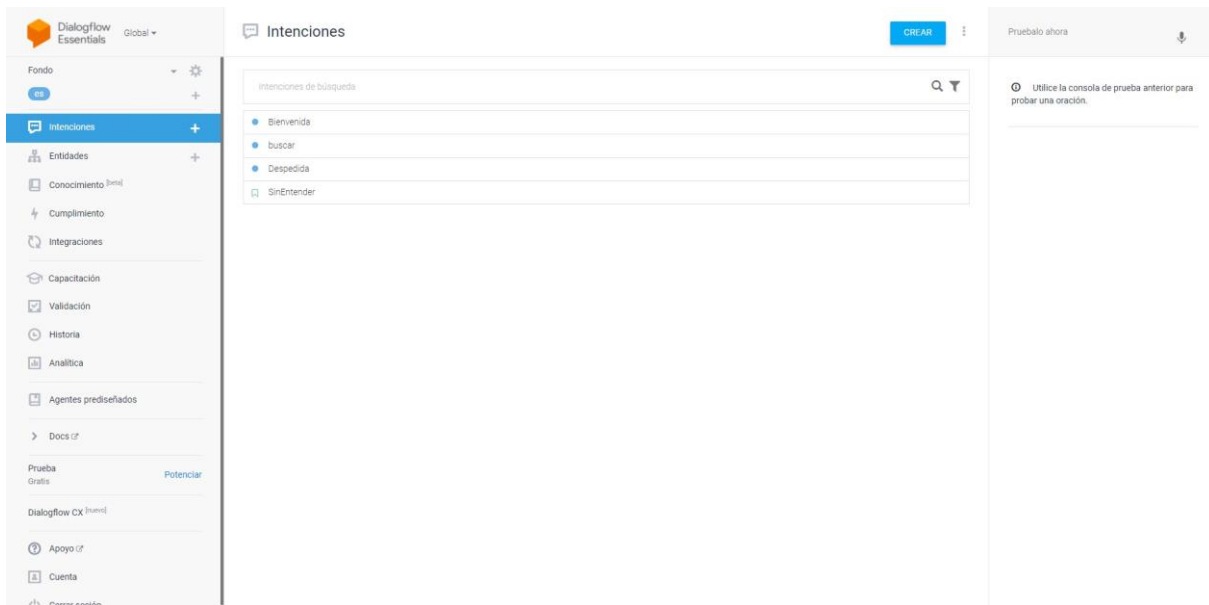


Figura 5.10: Muestra de 4 intenciones al usar el Bot de DialogFlow

La función de los intents es evaluar cada uno de los mensajes de los usuarios por medio de técnicas de inteligencia artificial como las redes neuronales, etc. Con base en las frases de entrenamiento, después, el mensaje es clasificado en uno de los intents (al que tenga mejor relación) y después activa uno de los mensajes de respuesta que ya habíamos definido para el tipo de mensaje mandado por el usuario.

Ahora, concentrándose en el intent de buscar (ya que todo el proceso se hará ahí), se necesita una forma de capturar la información sobre la búsqueda que el usuario ingrese, dado que vendrá combinado junto con frases de entrenamiento, para ello se va a incluir una nueva estructura, las entities. Las entities permitirán guardar momentáneamente la información sobre la búsqueda como si fueran variables, para después procesarlas, para crearlas, solo es necesario declarar la palabra reservada en acciones y parámetros y agregarlas a las frases de entrenamiento como se mostrará a continuación:

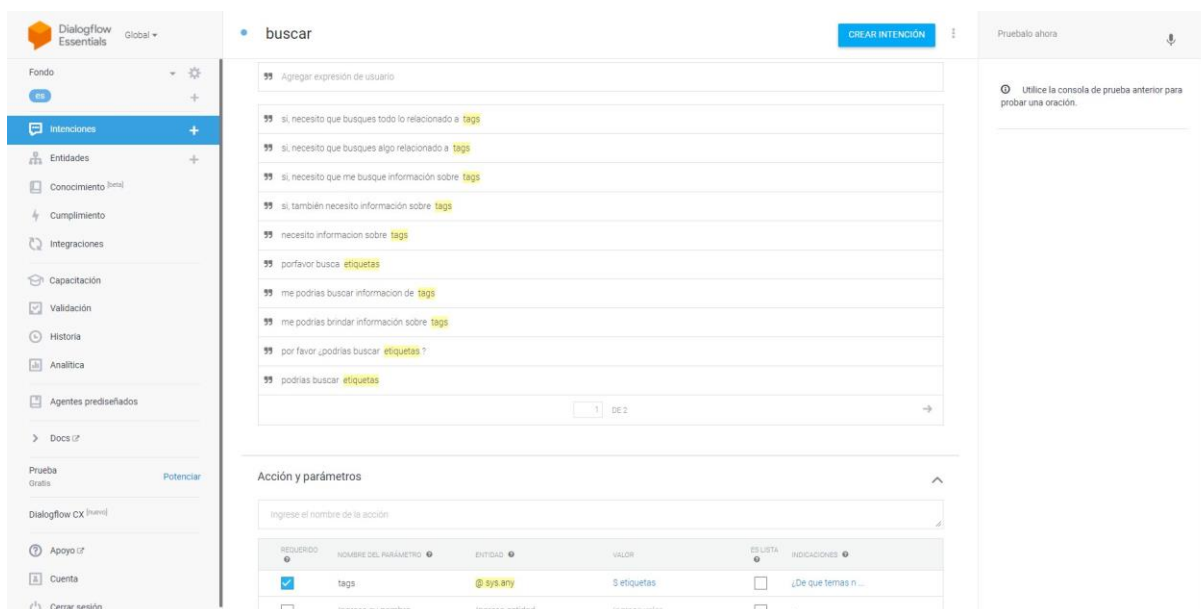


Figura 5.11: integración de las intenciones al Bot

En la siguiente sección se verán los próximos pasos a seguir para terminar el chatbot.

5.2. Servidor externo

Hasta este punto ya se tiene creado al agente (desde ahora así será llamado al Bot) que realizara las búsquedas por el usuario, sin embargo, el agente en el estado actual no sirve, ya que no puede hacer otra cosa más que contestar a nuestras entradas como usuario, el agente necesita efectuar acciones, como el de buscar algo en la red, para ello se debe de implementar un WebHook para que el agente puede realizar este tipo de acciones, para ello aquí se explicará la segunda parte de la creación del chatbot.

5.2.1. Librerías usadas

Como se explicó anteriormente, el lenguaje a usar fue JavaScript con implementación de Node.js para el Backend, por lo tanto, se usaron librerías disponibles en Node.js para desarrollar el servidor, ya que aloja tanto al Frontend como al Backend, a continuación, se mostrarán las librerías usadas en el proyecto:

5.2.1.1. Actions-on-google (dependencia)

Librería del cliente que facilita la creación de acciones para el Asistente de Google y, además, es compatible con DialogFlow, esta librería es fundamental para la librería de dialogflow-fulfillment.

Esta librería se puede adquirir con el siguiente comando de npm:

```
\$ npm install actions-on-google
```

5.2.1.2. DialogFlow-fulfillment (dependencia)

La Librería de cumplimiento de DialogFlow le permite conectar la comprensión y el procesamiento del lenguaje natural con sus propios sistemas, API y bases de datos. Con fulfillment, puede mostrar comandos e información de sus servicios a sus usuarios a través de una interfaz de conversación natural, además esta librería permite la creación de respuestas enriquecidas (cajas de texto, imágenes, etc.), para una mejor experiencia de usuario, también depende de la integración que se escoja, en este caso se usó la integración de DialogFlow Messenger.

Esta librería se puede adquirir con el siguiente comando de npm:

```
\$ npm install dialogflow-fulfillment
```

5.2.1.3. Dotenv (dependencia)

Dotenv es un módulo de dependencia cero que carga variables de entorno desde un archivo.env a process.env. El almacenamiento de la configuración en el entorno por separado del código se basa en la metodología de la aplicación de doce factores.

Esta librería se puede adquirir con el siguiente comando de npm:

```
\$ npm install dotenv
```

5.2.1.4. Ejs (dependencia)

Esta librería permite usar el motor de plantillas ejs que básicamente es una forma de poder usar JavaScript incrustado en una página HTML como es el caso de php con HTML.

Esta librería se puede adquirir con el siguiente comando de npm:

```
\$ npm install ejs
```

5.2.1.5. Express (dependencia)

Es una librería de desarrollo minimalista para Node.js que permite estructurar una aplicación de una manera ágil, proporciona funcionalidades como el enrutamiento, opciones para gestionar sesiones, etc. Esta librería hace posible montar un servidor de una forma sencilla.

Esta librería se puede adquirir con el siguiente comando de npm:

```
\$ npm install express
```

5.2.1.6. Googleapis (dependencia)

Esta librería permite hacer uso de las Apis de Google, incluyendo soporte de autenticación con OAuth 2.0, claves de API y tokens JWT.

Esta librería se puede adquirir con el siguiente comando de npm:

```
\$ npm install googleapis
```

5.2.1.7. Nodemon (dependencia de desarrollo)

Nodemon es una herramienta que ayuda a desarrollar aplicaciones basadas en Node.js al reiniciar automáticamente la aplicación cuando se detectan cambios.

Esta librería se puede adquirir con el siguiente comando de npm:

```
\$ npm install nodemon -D
```

5.2.2. Frontend

En cuanto al frontend (que es la parte que se ejecutará en el lado del usuario), se creó un archivo **index.ejs** el cual implementa código JavaScript y HTML, para llevar una mejor administración de las pantallas y simplificar el número de archivos a usar, también se usó Bootstrap 4 para la implementación de los diseños de la interfaz gráfica. El código resultante index.ejs es procesado en el motor de plantillas de ejs en el servidor, cuando el servidor recibe una solicitud GET en la ruta el servidor manda al usuario dicha interfaz. El resultado de la implementación fue la siguiente:

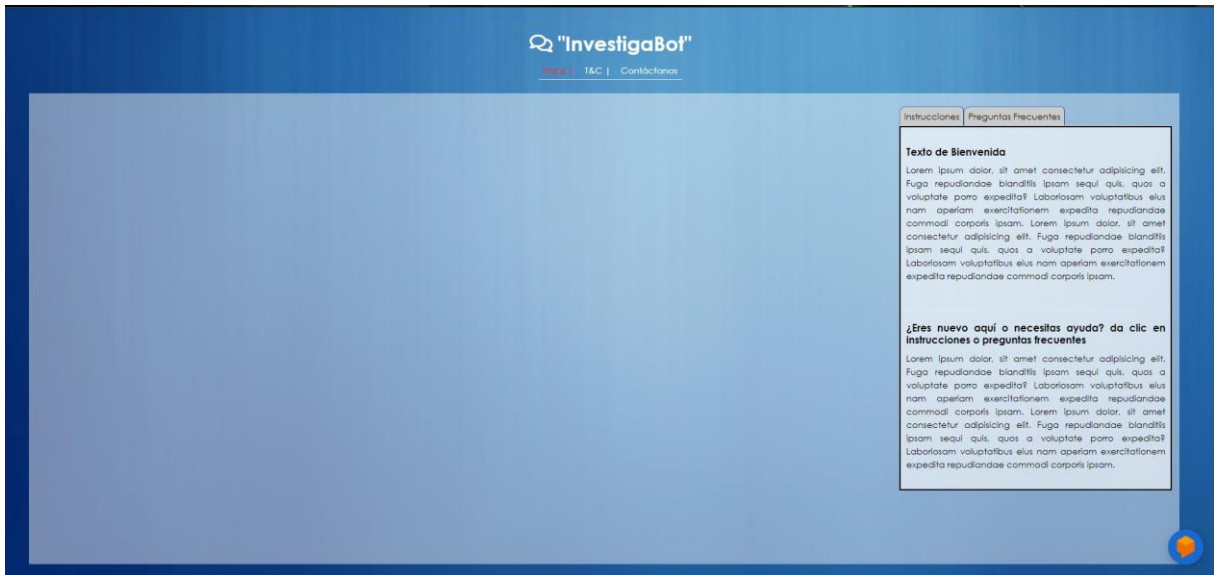


Figura 5.12: Vista de interfaz gráfica

Para agregar el chat del Bot, es necesario regresar a DialogFlow, pero esta vez no se creará ningún intent, esta vez se realizarán los siguientes pasos:

1. Una vez estando en la sección de intents de DialogFlow (en donde el navegador siempre nos dirigirá), ir al menú izquierdo y dar clic en la opción de integraciones.

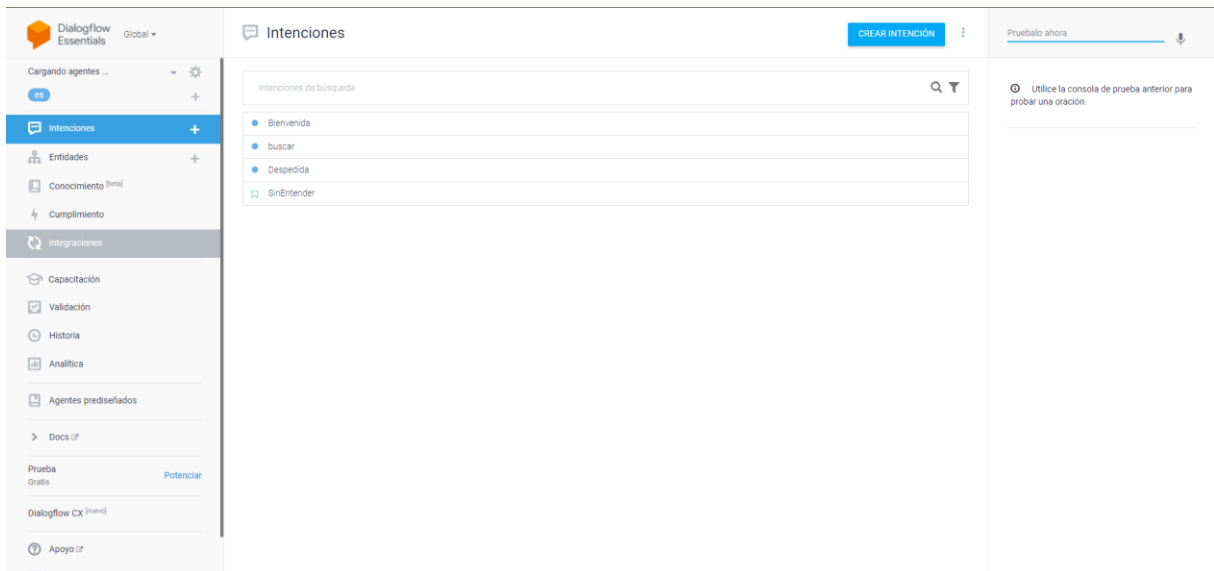


Figura 5.13: Pantalla de inicio de DialogFlow (intenciones)

2. Una vez estando en ese apartado, bajar en la página hasta encontrar la opción de DialogFlow Messenger.

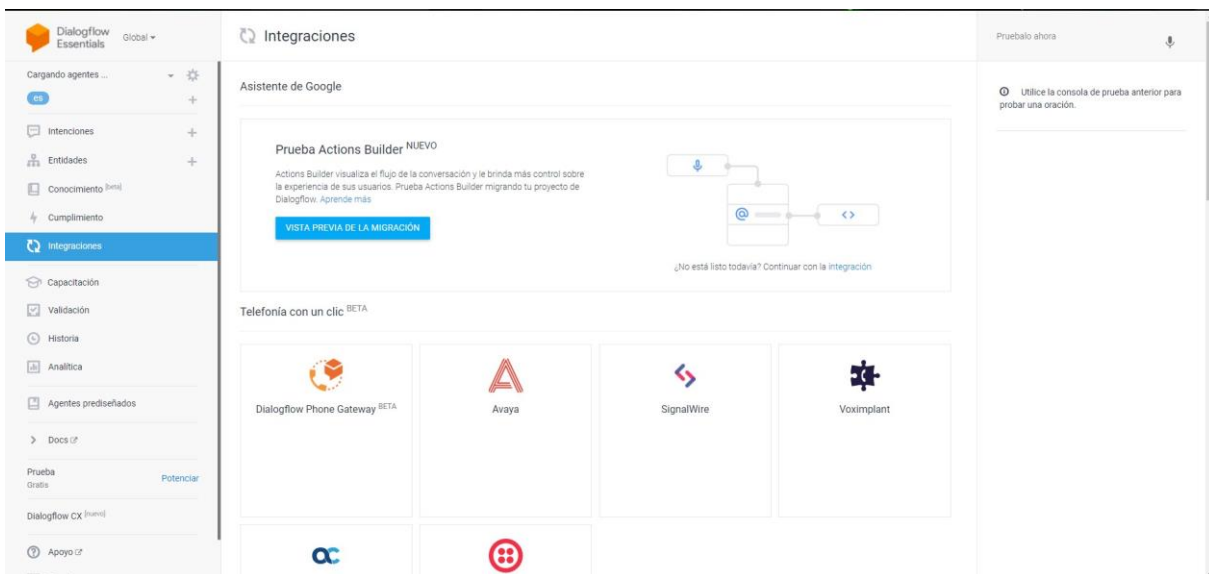


Figura 5.14: Pantalla de integraciones

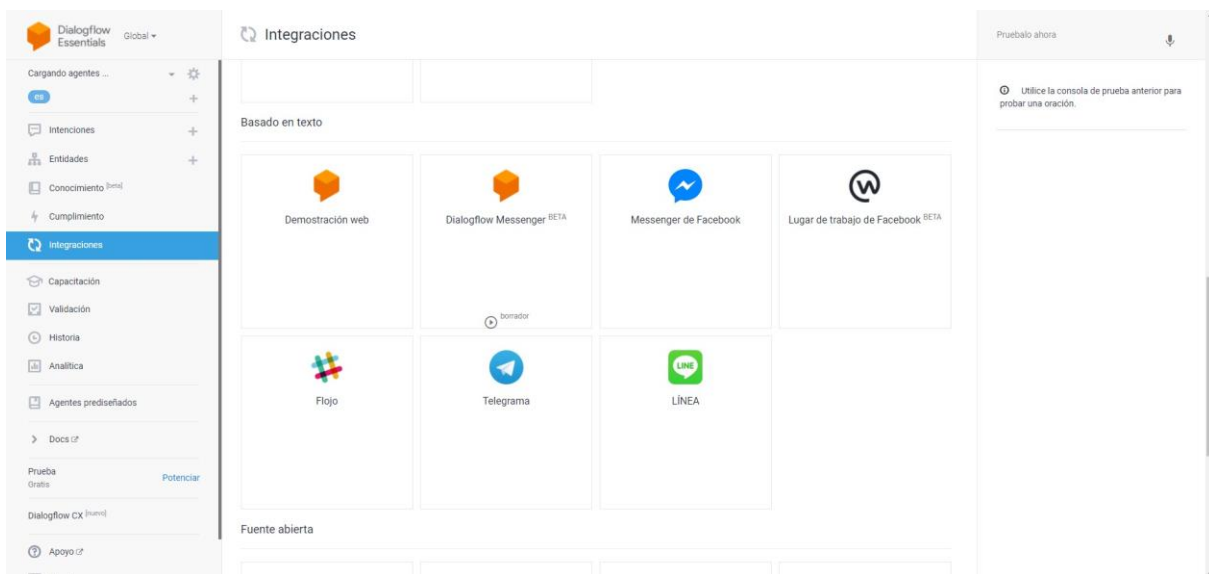


Figura 5.15: Búsqueda de la integración DialogFlow-Messenger

3. Dar clic sobre la opción y después se mostrará una ventana flotante en donde debemos dar clic en habilitar.

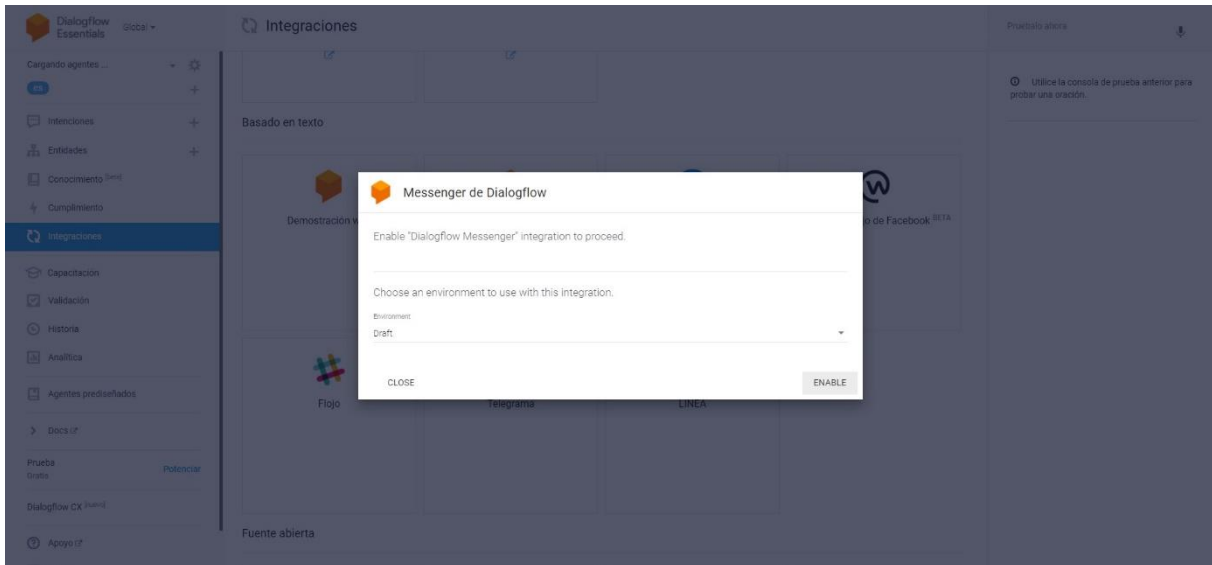


Figura 5.16: Ventana de implementación de DialogFlow-Messenger (deshabilitada)

4. Después la ventana flotante cambiará y mostrará un fragment el cual debemos de copiar y pegar en el sitio web (en este caso en el archivo index.ejs) y listo, con eso Bot funcionará con las configuraciones hasta el momento.

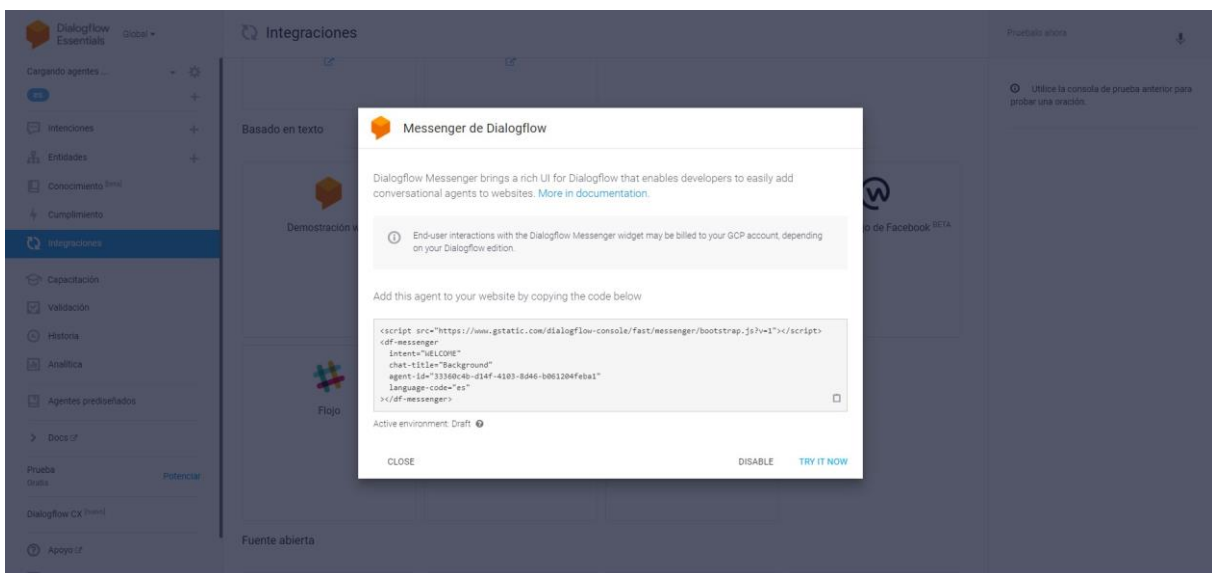


Figura 5.17: Ventana de implementación de DialogFlow-Messenger (habilitada)

5.2.3. Backend

En cuanto a la implementación del Backend (la parte que se ejecutara en servidor), se dividió en dos archivos, main.js (contiene al servidor) y routes.js (administra las rutas), esta implementación se realizó de esta forma para modular al código, ya que es más probable que una falla se encuentre en esta sección que en el Frontend.

Comenzando por el archivo main.js, primero se crea el servidor usando la librería de Node.js express con las siguientes líneas:

```
1 | const express = require('express');  
2 | const app = express();
```

Después se configura el puerto, pero como no se sabe si el puerto 3000 (default) este ocupado, entonces se implementa **process.env.PORT** lo cual significa que, si el puerto 3000 está ocupado, entonces proporcione otro, dicha implementación se ve en la siguiente línea:

```
1 | app.set("PORT", process.env.PORT || 3000);
```

Como ya se tiene creada la interfaz gráfica, ahora se tiene que ligar al servidor, por lo tanto, se deben hacer unas cuantas configuraciones, empezando por salvar la ruta de carpeta contenedora de index.ejs, la cual se realiza en la siguiente línea:

```
1 | app.set('views', path.join(__dirname, 'views'));
```

También se debe de especificar el motor de plantillas usado (en caso de usarlo, si no, ignorar este paso) porque de no ser así, entonces se generará un error y la aplicación no funcionará. En esta ocasión se usó ejs y se especificó en la siguiente línea:

```
1 | app.set('view engine', 'ejs');
```

Se agrega la configuración para reconocer cadenas o matrices entrantes mediante la siguiente línea:

```
1 | app.use(express.urlencoded({extended: false}));
```

Se establece una ruta estática para acceder a los archivos de configuración externos que están dentro de la carpeta Public, mediante la línea:

```
1 | app.use(express.static(path.join(__dirname, 'public')));
```

Para terminar el servidor básico lo único que falta es pasar el puerto donde el servidor escuchará las peticiones que se realicen, dicha configuración hará en la siguiente línea:

```
1 | app.listen(app.get("PORT"), () =>{console.log(`server on port  
    ${app.get(" PORT")} `);});
```

Cabe aclarar que el `console.log` solo es un texto prueba para ver si funciona una vez que este sea levantado.

El código `main.js` quedaría de la siguiente forma:

```
1 | const express = require('express');  
2 | const app = express();  
3 | const path = require('path');  
4 |  
5 | // configuraciones al servidor  
6 | app.set("PORT", process.env.PORT 3000);  
7 |  
8 | app.set('views',path.join(__dirname, 'views'));  
9 | app.set('view engine', 'ejs');  
10 | app.use(express.urlencoded({extended: false}));  
11 | app.use(express.static(path.join(__dirname, 'public')))  
12 |  
13 | app.listen(app.get("PORT"), () =>{console.log(`server on port  
    ${app.get(" PORT")} `);});
```

En la siguiente subsección se tratarán los temas de las rutas y del servicio de WebHook para realizar la conexión de DialogFlow al servidor.

5.2.3.1. WebHook

Una vez teniendo al servidor, ahora es tiempo de definir las rutas, sin embargo, como se comentó anteriormente, las rutas están separadas del archivo `main.js`, estas están localizadas en `routes.js`, y para poder acceder a ellas es necesario ligarlas en el archivo `main.js`, esto se logrará en la siguiente línea:

```
1 | const Router = require("../Routes/routes.js");
```

Y añadiendo en la siguiente línea hasta el final (no necesariamente al final):

```
1 | app.use(Router);
```

Por lo tanto, **main.js** quedaría de la siguiente forma:

```
1 | const express = require('express');
2 | const app = express();
3 | const Router = require("./Routes/routes.js");
4 | const path = require('path');
5 |
6 | // configuraciones al servidor
7 | app.set("PORT", process.env.PORT 3000 );
8 |
9 | app.set('views', path.join(__dirname, 'views'));
10 | app.set('view engine', 'ejs');
11 | app.use(express.urlencoded({extended: false}));
12 | app.use(express.static(path.join(__dirname, 'public')))
13 |
14 | app.listen(app.get("PORT"), () =>{console.log(`server on port
15 |     ${app.get(" PORT")} `)});
16 | // rutas en: Routes/routes.js
16 | app.use(Router);
```

Ahora en el archivo **routes.js** se tiene que crear una constante de express para poder acceder a **Router()** y guardarlo en otra constante, como en la siguiente línea:

```
1 | const express = require('express');
2 | const Routes = express.Router();
```

Ahora, lo que resta antes de pasar a la creación del servicio de WebHook, es crear la ruta donde se mandará la interfaz gráfica al usuario, para ello se usará una solicitud GET a la dirección (req, res) la cual responderá con la interfaz, este proceso se describe en las siguientes líneas con código JavaScript:

```
1 | Routes.get("/", (req, res) => {
2 |     res.render('index.ejs', {
3 |         title: 'InvestigaBot',
4 |     });
```

```
5 | });
```

Cabe aclarar que el campo de title no es importante en cuanto a funcionalidad. Para concluir la etapa base finalmente hay que exportar el módulo y para ello se debe de escribir la siguiente línea:

```
1 | module.exports = Routes;
```

Actualmente, el archivo **routes.js** tiene la siguiente forma:

```
1 | const express = require('express');
2 | const Routes = express.Router();
3 |
4 | Routes.get("/", (req, res) => {
5 |     res.render('index.ejs', {
6 |         title: 'InvestigaBot',
7 |     });
8 | });
9 |
10 | module.exports = Routes;
```

Para crear el servicio de WebHook es necesario hacer ciertas modificaciones tanto en el servidor externo como en DialogFlow, por el lado del servidor externo es necesario crear una solicitud POST a alguna dirección, en esta ocasión se hará a (req, res) y será casi la misma solicitud GET creada con anterioridad, solo que esta vez agregaremos una decodificación para objetos Json:

```
1 | const express = require('express');
2 | const Routes = express.Router();
3 |
4 | Routes.get("/", (req, res) => {
5 |     res.render('index.ejs', {
6 |         title: 'InvestigaBot',
7 |     });
8 | });
9 |
10 | Routes.post("/", express.json(), async (req, res) => {
11 |
```

```
12 |   });  
13 |  
14 |   module.exports = Routes;
```

Dentro del callback de la solicitud post creada, se creará nuestro agente, para ello es necesario conectar la librería `dialogflow-fulfillment`, para ello se escribirá la siguiente línea en `routes.js`:

```
1 |   const diff = require('dialogflow-fulfillment');
```

Con ello ya podemos crear nuestro agente dentro de la solicitud POST de la siguiente forma:

```
1 |   const agent = new diff.WebhookClient({ request: req, response:  
   |   res });
```

Ahora el siguiente paso es crear un objeto `Map` en el cual se mandará la información que sea arrojada de una función a un intent determinado, el intent debe tener el mismo nombre que en `DialogFlow`, a continuación, se mostrará la codificación en `JavaScript`:

```
1 |       var intentMap = new Map();  
2 |       intentMap.set('buscar', Search); // llamada de la  
   |       intención de búsqueda  
3 |       agent.handleRequest(intentMap); //retorno del servidor
```

Hasta este momento ya fue creado el primer extremo de comunicación para el servicio de `WebHook`, por lo tanto, el código de **`routes.js`** se ve así:

```
1 |   const express = require('express');  
2 |   const Routes = express.Router();  
3 |   const diff = require('dialogflow-fulfillment');  
4 |  
5 |   Routes.get("/", (req, res) => {  
6 |     res.render('index.ejs', {  
7 |       title: 'InvestigaBot',  
8 |     });  
9 |   });  
10 |  
11 |   Routes.post("/", express.json(), async (req, res) => {
```



```
12     const agent = new diffWebhookClient({ request: req,  
13         response: res});  
14  
15     var intentMap = new Map();  
16     intentMap.set('buscar', Search); // llamada de la intención  
17         de búsqueda  
18  
19     agent.handleRequest(intentMap); //retorno del servidor  
20 });  
21  
22 module.exports = Routes;
```

En este punto queda hacer las configuraciones de DialogFlow, para ello se tiene que ir intents y seleccionar el intent que necesite dicho servicio, para este caso es para el intent buscar, una vez estando ahí adentro, se debe buscar el área o sección de Fulfillment, y habilitar la llamada de WebHook para esa intención.

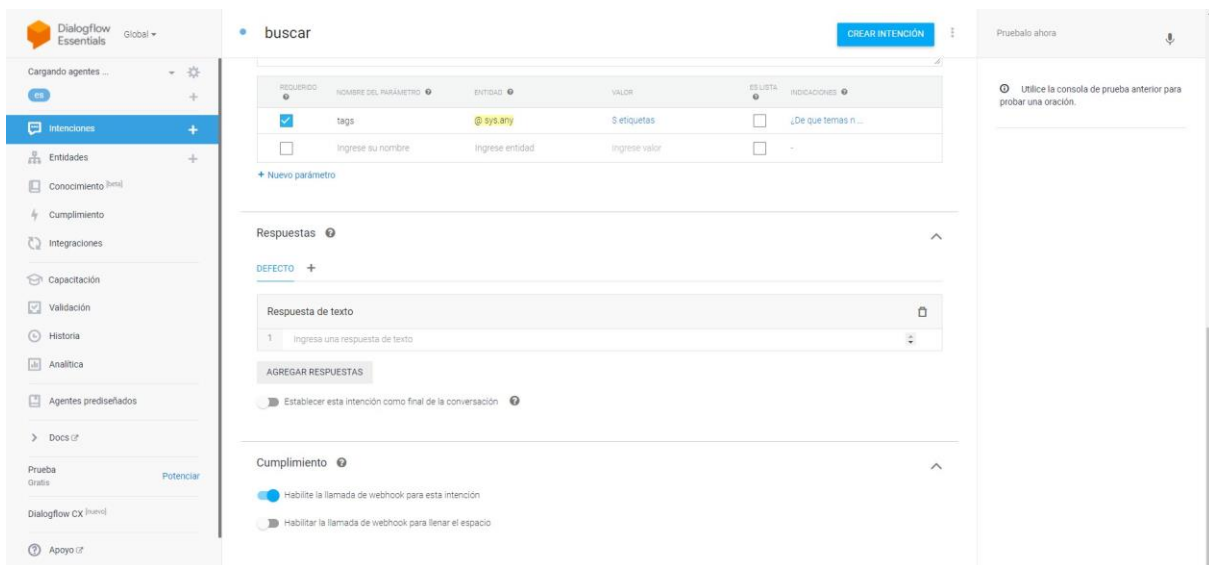


Figura 5.18: Habilitando cumplimiento para la intención

Después se tiene que activar la opción de WebHook en la opción de Fulfillment del menú izquierdo, sin embargo, aquí hay un problema, el WebHook solo acepta una URL publica, lo cual hace casi imposible seguir trabajando en forma local, sin embargo, aún se puede, usando ngrok, que permite crear un túnel desde el pc en modo local hacia una URL pública, ngrok puede ser usado como librería de Node.js mediante el comando de npm:

```
\$ npm install actions-on-google
```

o instalándolo como plugin de Visual Studio code, según sea su elección, sin embargo, para esta ocasión solo se usó como un plugin de Visual Studio code, para ello se configuro en el siguiente puerto:

```
\$ 3000
```

Una vez ingresado el puerto ngrok mandará un URL público el cual dará acceso a cualquier persona a el servidor creado en Node.js, después se debe de pegar el URL (en este caso de ngrok) en la sección de Fulfillment en la opción de WebHook, como en la siguiente imagen:

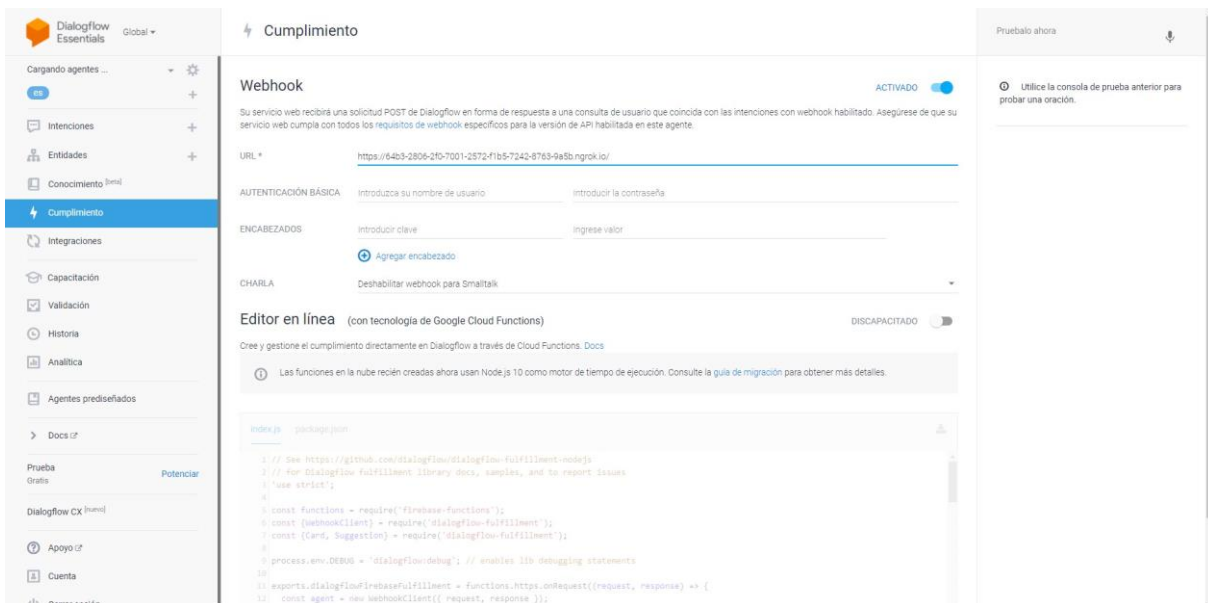
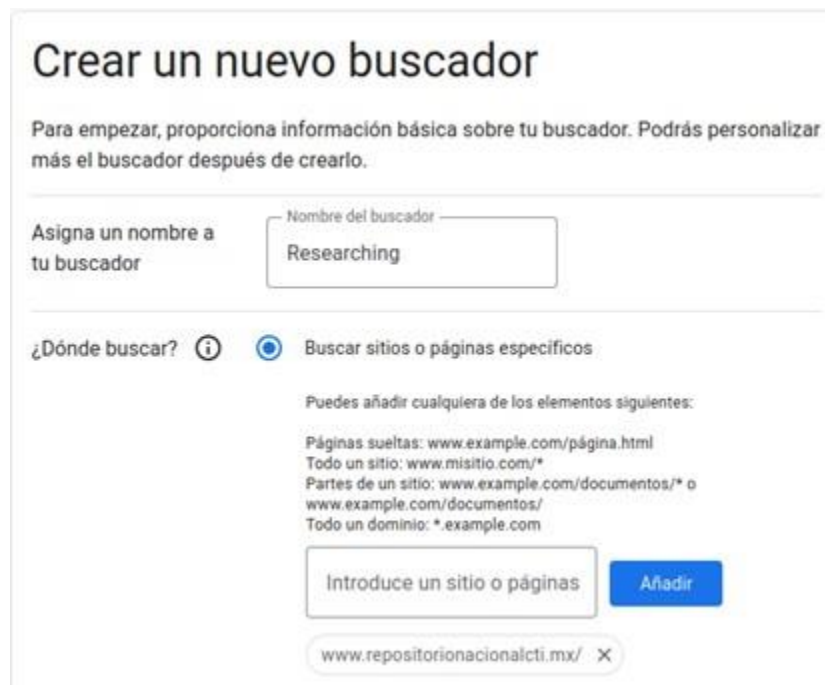


Figura 5.19: Ingresando URL público a DialogFlow

Con esta acción el servicio de WebHook externo ya está en servicio y el servidor reaccionará a las peticiones que se le pidan, en este punto ya se puede procesar las peticiones en objetos Json, sin embargo, de toda la petición que se recibe solo será necesaria la parte de las entities ya que este campo tiene la información a buscar la cual se mandara al motor de búsqueda programable, en la siguiente subsección se mostrará y explicara dicho proceso además del procesamiento de los datos recolectados del motor de búsqueda.

5.2.3.2. Conexión con motor de búsqueda de Google

Una vez ya creada la infraestructura necesaria (servidor Node.js y agente de DialogFlow) ya es posible montar el motor de búsqueda, para ello es necesaria ir al navegador de confianza y buscar ¿motor de búsqueda programable de Google?, dar clic en la opción que diga ¿Programmable Search Engine - Google? o entrar en la URL <https://programmablesearchengine.google.com/about/>, después dar clic en empezar y llenar los datos que piden, como se muestra en la imagen:



The image shows a web form titled "Crear un nuevo buscador". Below the title, there is a sub-header "Asigna un nombre a tu buscador" and a text input field containing "Researching". Underneath, there is a section "¿Dónde buscar?" with a radio button selected for "Buscar sitios o páginas específicos". Below this, there is a list of examples for adding URLs: "Puedes añadir cualquiera de los elementos siguientes: Páginas sueltas: www.example.com/página.html; Todo un sitio: www.misitio.com/*; Partes de un sitio: www.example.com/documentos/* o www.example.com/documentos/; Todo un dominio: *.example.com". At the bottom, there is a text input field with "Introduce un sitio o páginas", a blue "Añadir" button, and a search bar containing the URL "www.repositorionacionalcti.mx/" with a close button.

Figura 5.20: Ingresando URL público a DialogFlow

Después de generar el navegador, es necesario generar una API key para poder acceder al remotamente, al generarla solo hay 2 datos importantes que se deben de guardar:

- API KEY: AlzaSyCJxENADNWOzhFa1qDPq2JTgf1nxBsTepo (key)
- CX: a56fcac0a183b42c3 (referencia del buscador)

Con esos datos ya se puede comenzar a crear los archivos necesarios para hacer la conexión, para hacer eso, es necesario ir a la carpeta raíz del proyecto y crear un archivo **.env** el cual contendrá lo siguiente:

```
1 GG_API_KEY=AIzaSyCJxENADNWOzhFalqDPq2JTgf1nxBsTepo
2 GG_CX=a56fcac0a183b42c3
```

Después en la misma carpeta crear otro archivo de nombre **config.js**, el cual contendrá lo siguiente:

```
1 module.exports = {
2   ggApiKey: process.env.GG_API_KEY
3     'AIzaSyCJxENADNWOzhFalqDPq2JTgf1nxBsTepo',
4   ggCx: process.env.GG_CX      'a56fcac0a183b42c3'
```

Una vez hecho lo anterior regresar al archivo **routes.js** e importar a **googleapis**, y al archivo **config.js** además de crear la constante **customsearch**:

```
1 const { google } = require('googleapis');
2 const customsearch = google.customsearch('v1');
3 const config = require('../././config.js');
```

Con lo anterior ya es posible hacer la función de búsqueda que extraiga información del repositorio, para ello se debe declarar una función asíncrona (porque no se sabe cuándo se va a responder) de la siguiente forma:

```
1 async function Search(agent) {
2   //All code here
3 }
```

Una vez hecho eso, dentro de la función **Search** se deben extraer las entities del agente para poder ingresarlas a la búsqueda, para ello es necesario crear una constante para guardarlas:

```
1 const tag = agent.parameters.tags;
```

Después de ello es necesario, hacer la búsqueda, sin embargo, cabe aclarar que es un proceso asíncrono por lo cual, la respuesta no será rápida, para esperar la promesa devuelta por Google, se debe de interrumpir el flujo del proceso hasta obtener la promesa de Google, para ello es necesario agregar la palabra **await** para que se realice la interrupción temporal. Para realizar la búsqueda se debe de agregar la Key API generada, en auth, la referencia del buscado en cx, la búsqueda

en q y establecer el inicio en 0, lo anteriormente comentado se mostrará a continuación:

```
1 | const response = await customsearch.cse.list({
2 |     auth: config.ggApiKey,
3 |     cx: config.ggCx,
4 |     q: tag,
5 |     start: 0,
6 |     });
```

Una vez devuelta la promesa desde google, se debe verificar que la promesa no sea **undefined**, ya que esto genera un error y no mostraría nada:

```
1 | if (response.data.items)
```

Ya que se verifico que la promesa no sea undefined, se procede a extraer los datos del json resultante, dichos datos de importancia son los siguientes:

```
1 | response.data.items[i].pagemap.metatags[j].citation author
   | (autor)
2 | response.data.items[i].pagemap.metatags[j].citation title
   | (titulo)
3 | response.data.items[i].pagemap.metatags[j].citation_publication_date
   | (fecha)
4 | response.data.items[i].pagemap.metatags[j].citation_pdf_url (URL)
```

Dichos elementos se integran como una respuesta enriquecida y se integran al agente para mandarlos al chat del agente en DialogFlow, el código completo de la función **Search** es el siguiente:

```
1 | async function Search(agent) {
2 |
3 |     const tag = agent.parameters.tags;
4 |     const response = await customsearch.cse.list({
5 |         auth: config.ggApiKey,
6 |         cx: config.ggCx,
7 |         q: tag,
8 |         start: 0,
9 |     });
10 | }
```

```
11  if (response.data.items) {
12      if (response.data.items.length == 1) {
13          agent.add(`Encontré el siguiente resultado:`);
14      } else {
15          agent.add(`Encontré los siguientes resultados
16      }
17      let count = 0;
18      var payloadData = [];
19      var payloadData_Form = [];
20      for (item of response.data.items) {
21          for (data of item.pagemap.metatags) {
22
23              if (data.citation_pdf_url && data.citation_author &&
24                  data
25                      .citation_publication_date &&
26                      data.citation_title) {
27                  if (count <= 10) {
28                      var cita = data.citation_author + ",\n " + data.
29                          citation_publication_date + ".\n " + data.
30                          citation_title + ".";
31                      var data = {
32                          "type": "info",
33                          "title": cita,
34                          "subtitle": data.citation_pdf_url,
35                          "image": {
36                              "src": {
37                                  "rawUrl": "/img/larios_bot.png"
38                              }
39                          },
40                          "actionLink": data.citation_pdf_url
41                      }
42                      payloadData_Form.push(data);
43                  }
44              }
45          }
46      }
47      if(payloadData_Form.length > 0){
48          payloadData.push(payloadData_Form);
49          var myData = {"richContent":payloadData};
50          agent.add(new diff.Payload(agent.UNSPECIFIED, myData, {
51              sendAsMessage: true, rawPayload: true }));
52      }
53      }else{
```

```
47     agent.add(`lo lamento no encontré coincidencias sobre
48     ${tag}`);
49   }
50   agent.add("algo más");
51
52   return
53 }
```

5.3. Pruebas de funcionamiento

Una vez ya terminado el proyecto se procedió a hacer varias pruebas, a continuación, se presentará una de ellas con el fin de mostrar su funcionamiento:

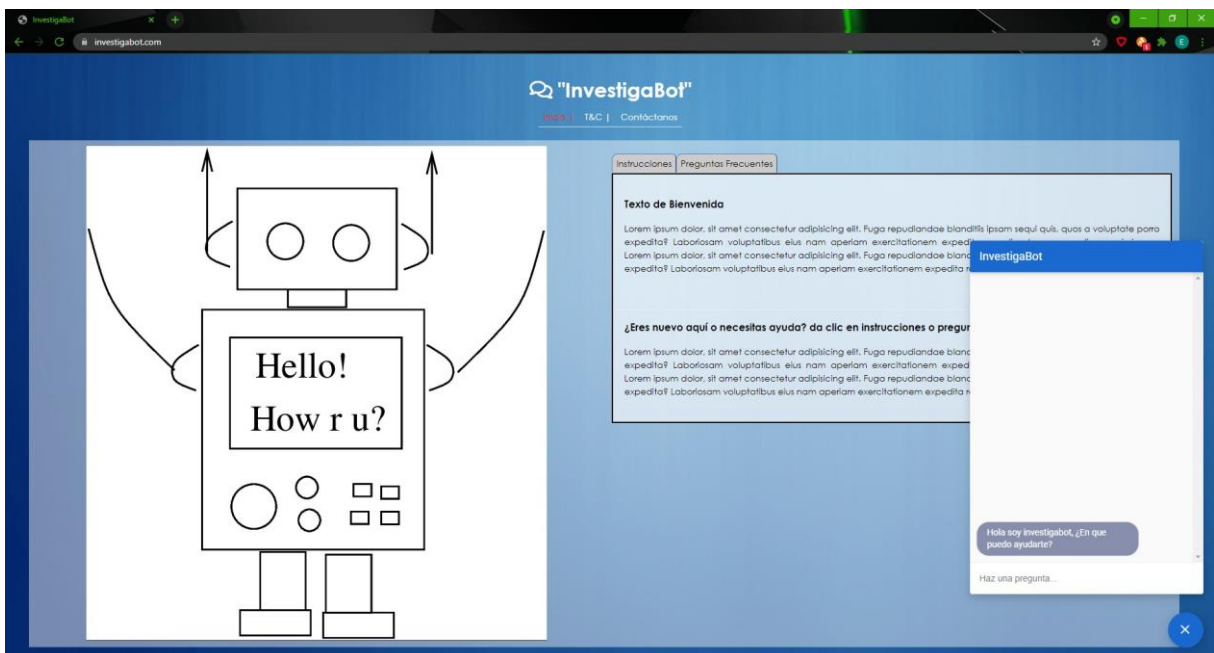


Figura 5.21: Inicio de InvestigaBot

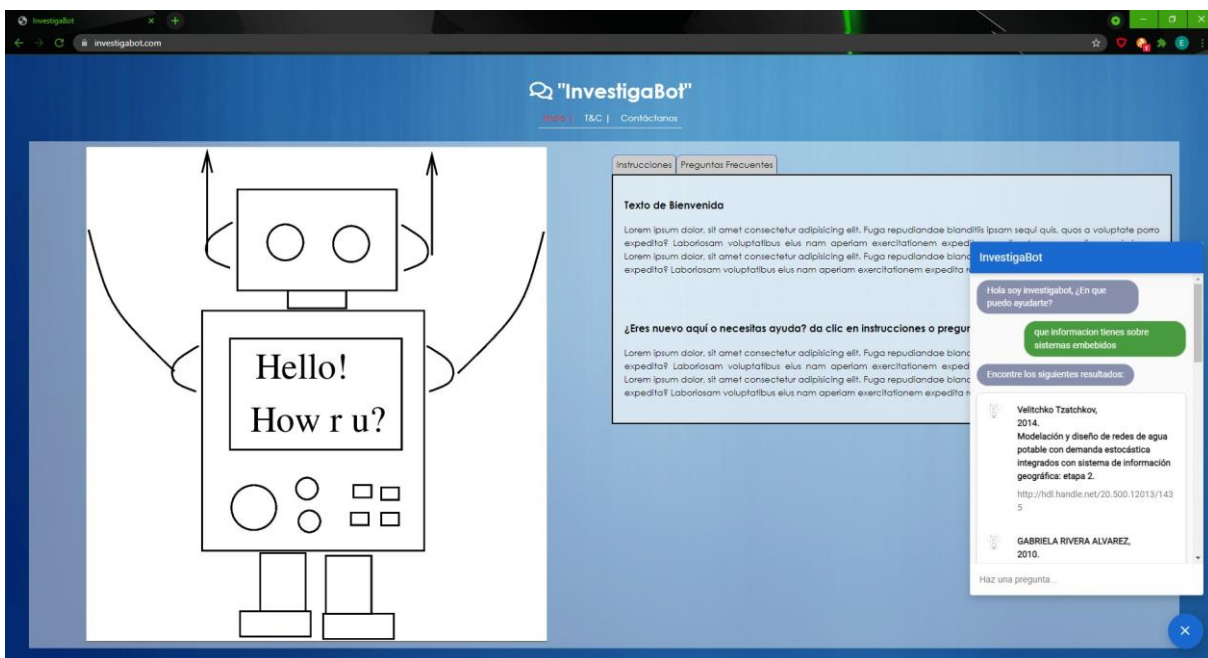


Figura 5.22: Resultados de consulta de InvestigaBot

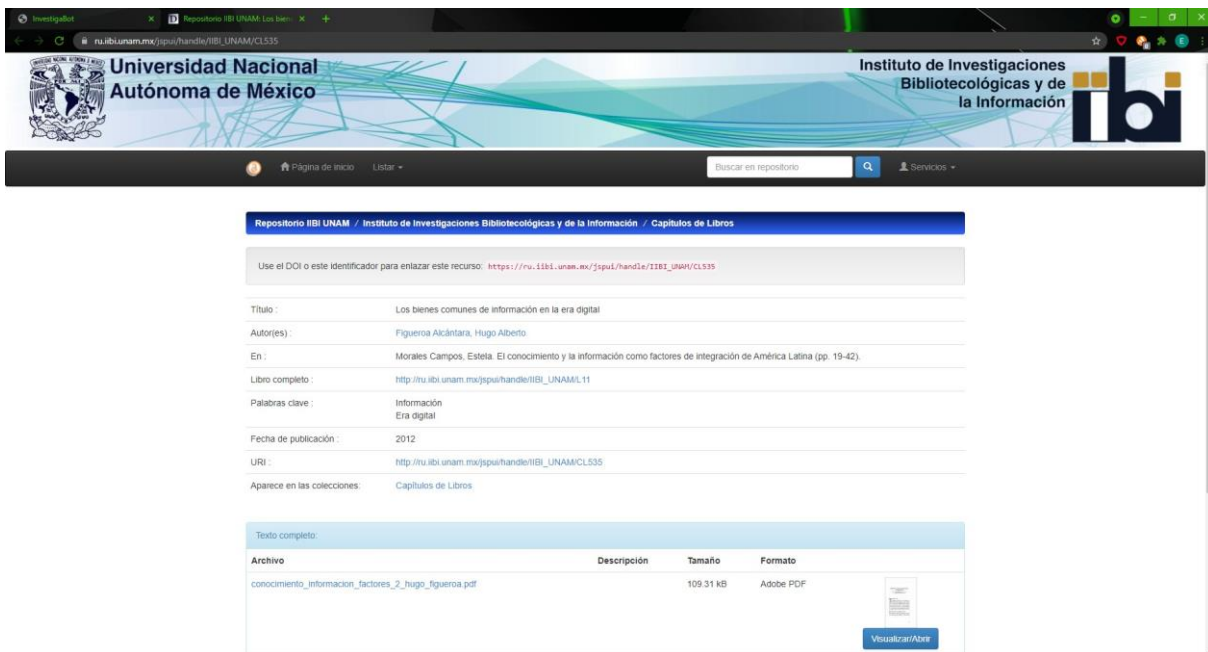


Figura 5.23: Dirección del redireccionamiento desde InvestigaBot

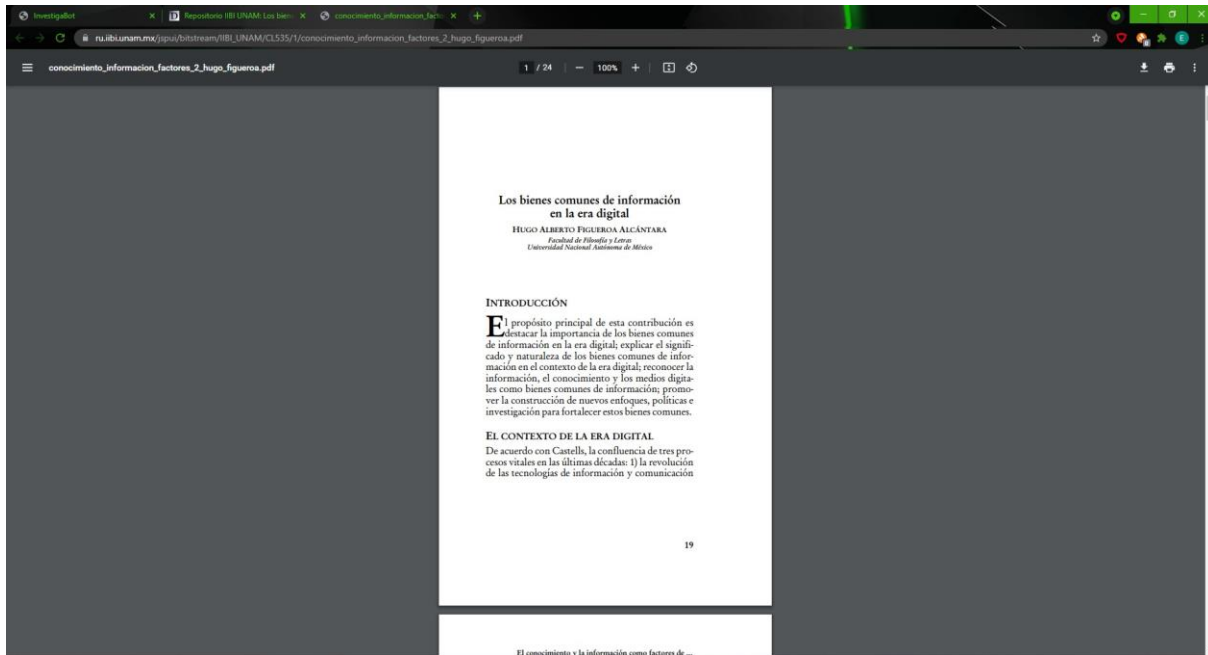


Figura 5.24: Realizando consulta en el InvestigaBot

Los proyectos en plataformas web siempre tienen un grado de complejidad, más que nada por la integración de las tecnologías JavaScript para que operen de forma esperada, por ejemplo, cuando se implementó el motor de búsqueda al servidor de Node.js se tuvieron problemas, ya que al momento que se pedía la información para ser procesada por el servidor, el json esperado era indefinido (undefined), esto paso debido a que estos procesos eran asíncronos, en otras palabras, el motor responde en cuanto tenga la información, implícitamente esto indica que hay un tiempo de espera, por lo que nuestra petición para el proceso de la información se ejecutaba antes de que Google mandara su parte y esto daba como resultado un undefined, esto se resolvió aplicando los términos de sincronía y no sincronía en los lugares apropiados, implícitamente también, al momento de tener este tipo de errores, se reafirmó mejor los conceptos de sincronía en JavaScript. En pocas palabras, hacer este proyecto tuvo 2 fines, el primero fue crear una herramienta que en un futuro pueda servir a varias personas y el segundo, fue didáctico, ya que permitió reafirmar el conocimiento ya adquirido y aprender nuevo conocimiento cercano al mundo real.

Capítulo 6

RESULTADOS FINALES

6.1. Propuesta para promover la motivación de los estudiantes en cursos universitarios en línea en el área de la Ingeniería y Tecnología

Se realizó la encuesta a un conjunto de alumnos de los últimos semestres de la carrera de Ingeniería en Ciencias de la Computación. La mayoría de los alumnos comentan que lo primero que hacen al empezar un proyecto es realizar una investigación previa al desarrollo del proyecto. Solo dos de ellos, que es el 2 %, esperan más indicaciones como se puede ver en la 6.1. Con esto tenemos que los estudiantes obtienen mejores resultados de aprendizaje a largo plazo y les motiva a continuar con sus estudios por tener retos individuales o en equipo. Dado que el 98 % está de acuerdo con el aplicar el ABP, estos obtienen mejor forma de investigar y aplicar el conocimiento en proyectos académicos reales. Sin embargo, cabe mencionar que el aspecto teórico es indispensable, además de la dirección del docente en el desarrollo del proyecto, esto hace mejorar el rendimiento del estudiante y se ve comprometido en el ambiente de trabajo.



Figura 6.1: Aprendizaje con ABP

Se realizó un estudio cuantitativo con respecto al aprendizaje con ABP. Como se muestra en la 6.2. Se tuvieron pocos alumnos, que es un 10 %, que no consideran el aprendizaje ABP para su formación profesional y sobre todo en las ciencias exactas. Cabe recordar que todos tenemos diferentes formas de aprendizaje y los métodos para aprendizaje son muy variados en la investigación de la educación y es muy difícil que un solo método sirva para todo tipo de aprendizaje como se menciona en González- Bohórquez (2019).

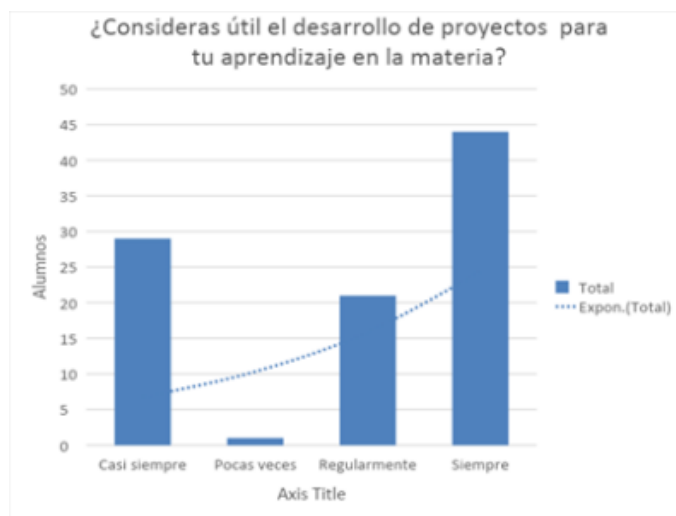


Figura 6.2: Criterio cuantitativo del aprendizaje con ABP

En la 6.2, cuantificamos en 4 diferentes respuestas, donde el 45 % comenta que siempre se aprende con el ABP, el 30 % dice que casi siempre, el 20 % dice que regularmente y el 5 % dice que pocas veces. Con esto tenemos un resultado a favor de la aplicación del método de enseñanza-aprendizaje llamado ABP. Con la aportación de un grupo de estudiantes con experiencia académica avanzada, puesto que son de los últimos semestres de carrera de ingeniería. Cabe mencionar que estos resultados son de mucha ayuda a los docentes en la forma de enseñar en los actuales tiempos de curso a distancia y que fueron necesarios por la pandemia que afectó en las universidades, principalmente en los semestres del 2020 y 2021. En esta investigación se trabajó con alumnos de universidad últimos semestres de una universidad pública con un buen ranking nacional y mundial. Se planteó este trabajo por la actual situación de pandemia, que forzó a muchos alumnos y catedráticos a usar herramientas tecnológicas y explorar y/o conocer otras formas nuevas o ya existentes de aprendizaje. Todo esto porque se está subiendo una cantidad considerable de material didáctico como apoyo en redes sociales y plataformas de aprendizaje en línea y la motivación de los alumnos por seguir estudiando esta bajado. Se planteó

este trabajo para dar una vista específica de cómo afecta la pandemia del COVID-19 en la asistencia a clases de universitarios, Se propuso un ambiente de trabajo apoyando el aprendizaje mediante el desarrollo y realización de proyectos reales y una solución concreta bajo la supervisión y guía del docente para animar a los estudiantes a seguir con sus estudios. De acuerdo con la perspectiva de los estudiantes, a la luz de los resultados, a un 88 % de los estudiantes les motiva a aprender con el uso de ABP como estrategia de aprendizaje en las ingenierías (computacional y electrónica). Sin embargo, el 8 % de los alumnos comentan que no es de gran ayuda este método de aprendizaje aplicado y el 4 % no contestó esta pregunta. Como sabemos, todos tenemos una forma diferente de aprender, pero algunos tienden por una metodología similar. Estos porcentajes se obtuvieron del cuestionario que se les pidió contestar a un grupo de 100 estudiantes de últimos semestres de la ingeniería en computación y electrónica.

6.2. Cuestionario que realiza el chatbot a los usuarios

Al terminar de realizar la investigación de los tópicos deseados. Se plantean las siguientes preguntas. Estas son clave para la mejora y retroalimentación del chatbot.

Métrica:

- La forma en la que se presenta la información ¿llama la atención y le invita a continuar investigando?
- ¿La información proporcionada a través del chatbot sirvió sobre el tema que busca y aclara alguna duda sobre el inicio de su proyecto?
- ¿La información recibida tiene algún lenguaje claro, sencillo y fácil de comprender?
- Sobre el uso de la herramienta Smart-software (chatbot), ¿le resultó fácil hacer uso de esta y explorar las opciones que le proporciona?
- ¿Le parece que la información proporcionada es adecuada y cumple con los resultados de búsqueda?

6.3. Cuestionario para la evaluación del InvestigaBot

Cuestionario que realiza el chatbot a los usuarios

- ¿En qué área de investigación te gustaría trabajar?
- Tu investigación de tesis será: ¿Teórica o práctica?

CONCLUSIONES

En este trabajo se propone de una herramienta de software inteligente aplicado a un problema actual, donde los principales actores son los estudiantes investigadores de posgrado. La herramienta es un chatbot enfocado a ayudar en la búsqueda de temas de investigación en los repositorios institucionales abiertos. El desarrollo de proyectos de investigación utilizando smart-software como un chatbot nos ayuda a obtener información que se encuentran en repositorios educativos abiertos y que necesitamos para el aprendizaje y la creación de proyectos científicos y de ingeniería. Este software inteligente, ha ayudado a un grupo de personas con un problema específico, como un estudiante de posgrado a realizar su investigación más rápido y de manera óptima. Así también se propone el desarrollo de una herramienta de tecnología con chatbots para la asistencia o ayuda en un ambiente educativo e investigación. La propuesta de herramientas tecnológicas en estos últimos años ha sido de gran importancia, sobre todo en tiempo de la pandemia del COVID-19, entre otros eventos que fundamentan el desarrollo e innovación tecnológica e inteligencia artificial.

REFERENCIAS

- AbuShawar, B. and Atwell, E. (2015). Alice chatbot: Trials and outputs. *Computación y Sistemas*, 19(4):625–632.
- Adame Rodríguez, S. I., Lloréns Báez, L., and Schorr Wiener, M. (2013). Retrospectiva de los repositorios de acceso abierto y tendencias en la socialización del conocimiento. *Revista electrónica de investigación educativa*, 15(2):148–162.
- Águila, J. V. B. and Burgos, V. (2010). *Distribución de conocimiento y acceso libre a la información con Recursos Educativos Abiertos (REA)*. La educación.
- Anglada, L. and Abadal, E. (2018). ¿qué es la ciencia abierta? *Anuario ThinkEPI*, 12:292–298.
- Arellano-Vázquez, M., Benítez-Pérez, H., and Ortega-Arjona, J. (2015). A consensus routing algorithm for mobile distributed systems. *International Journal of Distributed Sensor Networks*, 11(10):510707.
- Atkins, D. E., Brown, J. S., and Hammond, A. L. (2007). *A review of the open educational resources (OER) movement: Achievements, challenges, and new opportunities*, volume 164. Creative common Mountain View.
- Audsley, N. and Burns, A. (1990). Real-time system scheduling.
- Ávila, J. A. V. (2007). Acceso abierto a las publicaciones científicas electrónicas: un análisis de los beneficios y oportunidades. *Salud pública de México*, 49:67.
- Bao, Y., Kehm, B. M., and Ma, Y. (2018). From product to process. the reform of doctoral education in europe and china. *Studies in Higher Education*, 43(3):524–541.
- Bertuccelli, L., Beckers, W., and Cummings, M. (2010). Developing operator models for uav search scheduling. In *AIAA Guidance, Navigation, and Control Conference*, page 7863.
- Brandtzaeg, P. B. and Følstad, A. (2017). Why people use chatbots. In *International conference on internet science*, pages 377–392. Springer.
- Breitbart, J., Weidendorfer, J., and Trinitis, C. (2015). Case study on co-scheduling for hpc applications. In *2015 44th International Conference on Parallel Processing Workshops*, pages 277–285. IEEE.

Brieger Rocabado, S. P. (2019). Evaluación del programa doctoral no escolarizado en el cepies como una innovación educativa. *Educación Superior*, 6(2):69–78.

Burdick, A. (2018). Por qué el tiempo vuela: Una investigación no solo científica. Plataforma.

Cahn, J. (2017). Chatbot: Architecture, design, & development. University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science.

Cárdenas, M.L. (2019). Dificultades de docentes de inglés para publicar artículos científicos en contextos periféricos: percepciones de autores y evaluadores. *l'Kala, Revista de Lenguaje y Cultura*, 24(1):181-197.

Castillo-Gutiérrez, O. (2015). Diseño de Sistemas de Control Difuso para un Grupo de Sistemas Desacoplados y Análisis de su Estabilidad ante Retardos de Tiempo. PhD thesis, Posgrado en Ingeniería Eléctrica y Control., UNAM. Asesor. Dr. Héctor Benítez-Pérez.

CEPAL, N. (2019). La agenda 2030 y los objetivos de desarrollo sostenible: una oportunidad para américa latina y el caribe. objetivos, metas e indicadores mundiales.

Charneco, M. R., Molinab, A. M. C., and Urbistondoc, P. A. (2018). El blended marketing para la captación de usuarios de chatbots. In *Análisis y tendencias de las redes sociales. VIII edición del congreso Internacional sobre redes sociales Comunica2*, page 139.

Cheng, A. M. (2003). Real-time systems: scheduling, analysis, and verification. John Wiley & Sons.

Cheng-Min, L., Jyh-Horng, L., Chyi-Ren, D., and Chang-Ming, W. (2011). Benchmark dalvik and native code for android system. In *Second International Conference on Innovations in Bio- inspired Computing and Applications*, pages 320–323.

Colomé, D. and Femenia, P. (2018). Metodología de investigación para cursos de posgrado en ingeniería. UNIVERSIDAD NACIONAL DE SAN JUAN.

Concepción-Toledo, D. (2019). Metodología de la investigación: Origen y construcción de una tesis doctoral. *Revista Científica de la UCSA*, 6(1):76–87.

Corson, M. S., Macker, J., and Batsell, S. G. (1996). Architectural considerations for mobile mesh networking. In *Proceedings of MILCOM'96 IEEE Military Communications Conference*, volume 1, pages 225–229. IEEE.

CorteS, J. (2006). Finite-time convergent gradient flows with applications to network consensus. *Automática*, 42(11):1993–2000.

Cortés, J. (2008). Distributed algorithms for reaching consensus on general functions. *Automática*, 44(3):726–737.

Costa, L. and Oliveira, P. (2003). An elitist genetic algorithm for multiobjective optimization. In *Metaheuristics: computer decision-making*, pages 217–236. Springer.

Cova, F. and Rincón, P. (2010). El terremoto y tsunami del 27-f y sus efectos en la salud mental. *Terapia psicológica*, 28(2):179–185

Croes, E. A. and Antheunis, M. L. (2021). Can we be friends with mitsuku? a longitudinal study on the process of relationship formation between humans and a social chatbot. *Journal of Social and Personal Relationships*, 38(1):279–300.

De León, J. M. G. (2021). *Factores en la deserción de estudiantes doctorales: implicaciones para los líderes educativos*. PhD thesis, University of Puerto Rico, Rio Piedras (Puerto Rico).

Del Valle Jiménez, D. D., Celaya Ramírez, R., and Ramírez Montoya, M. S. (2016). Apropiación tecnológica en el movimiento educativo abierto: Un estudio de casos de prácticas educativas abiertas.

Díaz-Castelazo, C. (2018). La importancia del idioma inglés para el desarrollo y enseñanza de las ciencias. *Revista Eduscientia. Divulgación de la ciencia educativa*, 1(2):60–68.

Diez de Tancredi, D. (2015). Páginas web con documentos electrónicos de uso educativo. *Revista de Investigación*, 39(84):251–254.

DINASED-FGE. Informe de situación n°71 -terremoto 7.8 pedernales. Technical report.

Dinh, T. B., Vo, N., and Medioni, G. (2011). Context tracker: Exploring supporters and distracters in unconstrained environments. In *CVPR 2011*, pages 1177–1184. IEEE.

Dorst, K. (2011). The core of design thinking and its application. *Design studies*, 32(6):521–532.

Dueñas, L. F. G. (2008). Repositorios documentales y la iniciativa de archivos abiertos en Latinoamérica. *Bid. Textos Universitarios de Biblioteconomía I Documentación* (20).

Escudero Sánchez, C. L. and Cortez Suárez, L. A. (2018). Técnicas y métodos cualitativos para la investigación científica.

Esquivel-Flores, O. (2013). Estudio de sistemas multi-agentes reconfigurables. *Posgrado en Ciencias e Ingeniería de la Computación, UNAM*, 23.

Esquivel-Flores, O., Benítez-Pérez, H., and Ortega-Arjona, J. (2012). Issues on communication network control system based upon scheduling strategy using numerical simulations. In *Numerical Simulation-From Theory to Industry*. IntechOpen.

Ferreras Fernández, T. et al. (2016). Visibilidad e impacto de la literatura gris científica en repositorios institucionales de acceso abierto. estudio de caso bibliométrico del repositorio gredos de la universidad de salamanca.

Galitsky, B. (2019). *Developing enterprise chatbots: learning linguistic structures*. Springer.

García Brustenga, G., Fuertes Alpiste, M., and Molas Castells, N. (2018). Briefing paper: los chatbots en educación.

García-Peñalvo, F. J. and Montoya, R. (2015). Movimiento educativo abierto.

Gentsch, P. (2019). Conversational ai: how (chat) bots will reshape the digital experience. In *AI in marketing, sales and service*, pages 81–125. Springer.

George Reyes, Carlos Enrique y Salado Rodríguez, L. I. (2019). Competencias investigativas con el uso de las tics en estudiantes de doctorado. *Apertura (Guadalajara, Jal.)*, 11(1):40–55.

Gibbons, H. and Gelfenbaum, G. (2005). Astonishing wave heights among the findings of an international tsunami survey team on sumatra. *Sound Waves, March*.

Gómez, J. and Pérez, H. (2015). Global scheduling of confined tasks based on consensus. *IEEE Latin America Transactions*, 13(3):825–834.

González-Bohórquez, M. Y. (2019). Estrategias de enseñanza y métodos de aprendizaje en la transferencia de conocimiento matemático. Estudio de caso en educación superior.

González Dalmau, S. (2021). Mybotwilliam: Un chatbot para una asignatura de programación. B.S. thesis, Universitat Politècnica de Catalunya.

González-Pérez, L. I., Glasserman Morales, L., Ramírez-Montoya, M., and García-Peñalvo, F. J. (2017). Repositorios como soportes para diseminar experiencias de innovación educativa. *Innovación Educativa. Investigación, formación, vinculación y visibilidad*, pages 259–272.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Gutiérrez, F. J. M. and de los Santos, J. G. E. (2009). La iniciativa knowledge hub: un aporte del tecnológico de monterrey al mundo. *Ried. Revista iberoamericana de educación a distancia*, 12(2):83–112.

Gutiérrez Siliceo, J. M. et al. (2019). Desarrollo de chatbots con entornos de código abierto.

Hasani, L. M., Senses, D. I., Suryono, R. R., et al. (2020). User-centered design of e-learning user interfaces: A survey of the practices. In *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, pages 1–7. IEEE.

Hatcher, P., Reno, M., Antoniu, G., and Bouge, L. (2005). Cluster computing with java. *Computing in science & engineering*, 7(2):34–39.

Herder, J. N., Bos, H., Gras, B., Homburg, P., and Tanenbaum, A. S. (2006). Minix 3: A highly reliable, self-repairing operating system. *ACM SIGOPS Operating Systems Review*, 40(3):80– 89.

Herrera, A. M. (2019). Aplicaciones digitales para aprender más y mejor: una propuesta desde la educación artística. *EDUNOVATIC2019*, page 108.

Hong, Y., Chen, G., and Bushnell, L. (2008). Distributed observers design for leader-following control of multi-agent networks. *Automatica*, 44(3):846–850.

Huaroto, L. (2020). Repositorios digitales de documentos.

Hui, Q. and Haddad, W. M. (2008). Distributed nonlinear control algorithms for network consensus. *Automatica*, 44(9):2375–2381.

Jeong, S., Simeone, O., and Kang, J. (2017). Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Transactions on Vehicular Technology*, 67(3):2049–2063.

Kaschel Cárcamo, H. and Pérez Bahamondes, J. (2014). Monitoreo ubicuo de salud en tiempo real con wbsn. *Ingeniare. Revista chilena de ingeniería*, 22(2):169–176.

Khamis, M. and Osorio, C. (2010). Análisis de riesgos de desastres en Chile. UNESCO.

<http://www.unesco.org/new/fileadmin/MULTIMEDIA/FIELD/Santiago/pdf/Analisis-de-riesgos-de-desastres-en-Chile.pdf>.

Khazai, B., Daniell, J. E., and Wenzel, F. (2011). The march 2011 japan earthquake: analysis of losses, impacts, and implications for the understanding of risks posed by extreme events. *Technikfolgenabschätzung–Theorie und Praxis*, 20(3):22–33.

Khosiawan, Y., Park, Y., Moon, I., Nilakantan, J. M., and Nielsen, I. (2018). Task scheduling system for uav operations in indoor environment. *Neural Computing and Applications*, pages 1–29.

Kim, Y.-J., Cho, S.-J., Kim, K.-J., Hwang, E.-H., Yoon, S.-H., and Jeon, J.-W. (2012). Benchmarking java application using jni and native c application on android. In *2012 12th International Conference on Control, Automation and Systems*, pages 284–288. IEEE.

Körber, N. and Suleman, H. (2008). Usability of digital repository software: A study of dspace installation and configuration. In *International Conference on Asian Digital Libraries*, pages 31–40. Springer.

Kreemer, C., Holt, W. E., and Haines, A. J. (2003). An integrated global model of present-day plate motions and plate boundary deformation. *Geophysical Journal International*, 154(1):8–34.

Kshemkalyani, A. D. and Singhal, M. (2011). *Distributed computing: principles, algorithms, and systems*. Cambridge University Press.

Larios-Gómez, M., Anzures-García, M., Gálvez, M. L. A. S., Lima, C. Z., and Anota, C. B. H. (2020). Aprendizaje de tópicos avanzados computacionales y tecnológicos basados en proyectos. *Redes de aprendizaje digital en nodos colaborativos*, page 262.

Larios-Gómez, M., Carmona-Flores, M. E., Anzures-García, M., Erika, M. M., and Nancy, V.-

G. (2021). Educational real-time scheduling framework on distributed mobile environments. <http://carla2021.org/schedule/poster/284>.

Larios-Gómez, M., Carrera, J. M., Anzures-García, M., Aldama-Díaz, A., and Trinidad-García, G. (2019). A scheduling algorithm for a platform in real time. In *International Conference on Supercomputing in México*, pages 3–10. Springer.

Larios-Gómez, M., Hernández-Beristáin, A., Martínez-Mirón, E. A., Caldera-Miguel, J., and Zamarripa-Almazan, L. A. (2017). Jscheduling: A graphical interface for applying a process scheduling algorithm. *Applications of Language & Knowledge Engineering*, 2(1):119–127.

Larios Gómez, M. (2017). Algoritmos de planificación y ruteo en un ambiente distribuido simulado. Laboratorio Nacional de Supercómputo del Sureste de México. <http://lns.org.mx/?q=content/proyectos-aceptados-primavera-2017>.

Larios Gómez, M. (2018). Jscheduling: Software embebido para la planificación en tiempo real de procesos masivos. Laboratorio Nacional de Supercómputo del Sureste de México. <http://lns.org.mx/?q=content/proyectos-aceptados>.

Larios Gómez, M. (2019). Planificación y comunicación en un sistema distribuido simulando sobre una red p2p. Laboratorio Nacional de Supercómputo del Sureste de México. [http:// Ins.org.mx/?q=content/proyectos-aceptados-primavera-2019](http://Ins.org.mx/?q=content/proyectos-aceptados-primavera-2019).

Larios Gómez, M. (2021). Smart-software para extraer recursos informáticos de repositorios distribuidos. Laboratorio Nacional de Supercómputo del Sureste de México. [https:// Ins.buap.mx/?q=proyectos-aceptados-2021-01-04](https://Ins.buap.mx/?q=proyectos-aceptados-2021-01-04).

Lee, S. and Jeon, J. W. (2010). Evaluating performance of android platform using native c for embedded systems. In *ICCAS 2010*, pages 1160–1163. IEEE.

López, P. and Romero, L. (2020). Aumenta el riesgo de deserción en licenciatura y posgrado. *Gaceta UNAM*, 1372.

López-Ornelas, M., Lever, C. O., and López, K. M. D. (2017). Las revistas académicas de comunicación de acceso abierto en México. retos y vicisitudes. *Revista Latina de Comunicación Social*, (72):475–499.

Lucena, F. J. H., Díaz, I. A., and Rodríguez, J. M. R. (2018). Dispositivos móviles para el aprendizaje: análisis de la investigación doctoral sobre mobile learning en España. *Texto Livre*, 11(3):154– 175.

Machado, C. and Farias, M. A. (2012). Banco internacional de objetos educacionais. *EFT: Educação, Formação & Tecnologias*, 5(1):89–91.

March, L. (1976). *The architecture of form*. Cambridge University Press.

Marcondes, F. S., Almeida, J. J., and Novais, P. (2018). Chatbot theory. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 374–384. Springer.

MARTELO, R. J., JARAMILLO, J. M., and OSPINO, M. (2018). Producción científica de docentes universitarios y estrategias para aumentarla mediante series de tiempo y multipol. *Revista Espacios*, 39(16).

Martin, J. M. (2020). Educar en tiempos de pandemia. *Revista Internacional de Investigación en Ciencias Sociales*, 16(2).

Martínez, J. L. and Tapia-Rangel, E. (2018). Alebrije model for the development and supply of educational content. *Exploring the Micro, Meso and Macro*, page 531.

Martínez, S. V. (2009). Eduteka. *Revista Padres y Maestros/Journal of Parents and Teachers*, (323):12.

Masuzzo, P. and Martens, L. (2017). Do you speak open science? resources and tips to learn the language. Technical report, PeerJ Preprints.

Mauro, A. M. (2017). Chatbot based content discovery: Faulkner bot in the archive. In *DH*.

Medrano, J. F., Tejerina, M., and Castillo, C. A. (2019). Empleo de chatbots educativos como recurso complementario en las prácticas docentes. In *XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan)*.

Meinel, C. and Leifer, L. (2012). Design thinking research. In *Design thinking research*, pages 1–11.

Springer.

Membiela, P. (1997). Una revisión del movimiento educativo ciencia-tecnología-sociedad. *Enseñanza de las ciencias. Revista de investigación y experiencias didácticas*, 15(1):51–57.

Miguel, S. (2012). Oportunidades y desafíos para el acceso abierto a los conocimientos producidos en entornos académicos. In *1as Jornadas Nacionales de Políticas de Investigación de las Facultades de Humanidades y Educación 21 y 22 de junio de 2012 Rosario, Argentina*. Asociación Nacional de Facultades de Humanidades y de Educación (Anfhe).

Miguel, S. (2013). Oportunidades y desafíos en torno al acceso abierto a las publicaciones científicas. In *XXXIV Jornadas Argentinas de Botánica 2 al 6 de septiembre de 2013 La Plata, Buenos Aires, Argentina*. Comisión Directiva de la Sociedad Argentina de Botánica.

Montoya, M. S. R. Casos de formación e investigación en el área del movimiento educativo abierto en Latinoamérica: Alcances, retos y oportunidades.

Montresor, A. and Jelasity, M. (2009). Peersim: A scalable p2p simulator. In *2009 IEEE Ninth International Conference on Peer-to-Peer Computing*, pages 99–100. IEEE.

Mortera Gutiérrez, F. J. (2010). Implementación de recursos educativos abiertos (rea) a través del portal temoa (knowledge hub) del tecnológico de monterrey, México. *Formación universitaria*, 3(5):9–20.

Mulchandani, D. (1998). Java for embedded systems. *IEEE Internet Computing*, (3):30–39.

Nouiri, M., Bekrar, A., Jemai, A., Niar, S., and Ammari, A. C. (2018). An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. *Journal of Intelligent Manufacturing*, 29(3):603–615.

OECD (2019). *Education at a Glance 2019*.

Olaya Mantuano, S. I. (2019). *Diseño de un modelo de conservación y preservación digital para una notaría de la ciudad de Guayaquil*. PhD thesis, Universidad de Guayaquil. Facultad de Ingeniería Industrial.

Ortiz-Núñez, R. (2020). Análisis métrico de la producción científica sobre covid-19 en scopus.

Revista Cubana de Información en Ciencias de la Salud (ACIMED), 31(3):1–20.

Ortiz-Ocaña, A. (2018). La configuración de la tesis doctoral. su estructura, redacción, defensa y publicación. *Revista Latinoamericana de Estudios Educativos (Colombia)*, 14(2):102–131.

Packer, A. L., Prat, A. M., Luccisano, A., Montanari, F., Santos, S., and Menghini, R. (2006). El modelo scielo de publicación científica de calidad en acceso abierto. *Edición electrónica, bibliotecas virtuales y portales para las Ciencias Sociales en América Latina y el Caribe*, pages 191–208.

Palacios, P. R., Zendejo, D. S., and Ramírez, M. S. (2021). Opciones y experiencias para México y Latinoamérica.

Park, V. D. and Corson, M. S. (1997). A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of INFOCOM'97*, volume 3, pages 1405–1413. IEEE.

Park, Y., Khosiawan, Y., Moon, I., Janardhanan, M. N., and Nielsen, I. (2016). Scheduling system for multiple unmanned aerial vehicles in indoor environments using the csp approach. In *International Conference on Intelligent Decision Technologies*, pages 77–87. Springer.

Patiño Salceda, J. (2019). Análisis comparativo entre el doctorado profesional y de investigación en México. *Revista iberoamericana de educación superior*, 10(28):25–41.

Pearl, J. (2014). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.

Pinedo-Tuanama, L. and Valles-Coral, M. (2021). Importancia de los referenciadores bibliográficos en la gestión de la información científica en tesis universitarias. In *Anales de Documentación*, volume 24. Facultad de Comunicación y Documentación y Servicio de Publicaciones.

Ramasubramanian, V., Haas, Z. J., and Sirer, E. G. (2003). Sharp: A hybrid adaptive routing protocol for mobile ad hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 303–314. ACM.

Ramírez Montoya, M. S. (2013a). Casos de formación e investigación en el área del movimiento educativo abierto usando tecnologías emergentes en Latinoamérica. *Revista Fuentes*, 13, 93-114.

Ramírez Montoya, M. S. (2013b). Competencias docentes y prácticas educativas abiertas en educación a distancia.

Ramírez Montoya, M. S., García Peñalvo, F. J., et al. (2015). Movimiento educativo abierto.

Rittel, H. W. and Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy sciences*, 4(2):155–169.

Rivera Pérez, C. and Mendoza-Becerril, M. (2021). Effect of the covid-19 pandemic on scientific work: teaching and research. 7:35–45.

Rodríguez, I. S. (2010). Dilemas y oportunidades del conocimiento abierto. *Papeles de relaciones ecosociales y cambio global*, (110):89–98.

Rodríguez Palacios, S. M. d. P., García Guerrero, M., Salas Zendejo, D., Ramírez Montoya, M. S., and Torres Hernández, J. (2021). Ciencia abierta opciones y experiencias para México y Latinoamérica.

Romero-Charneco, M., Casado-Molina, A.-M., Alarcón-Urbistondo, P., et al. (2018). Channels of social influence for decision making in restaurants: A case study. *Dos Algarves: A Multidisciplinary e-Journal*, 32:54–76.

Romero Martínez, S. J., González, I., García, A., Lozano, A., et al. (2018). Herramientas tecnológicas para la educación inclusiva. *Tecnología, Ciencia y Educación*, 9:83–111.

Roy, B. K., Mukhopadhyay, P., Biswas, S. C., and Das, R. (2017). Information retrieval features of text retrieval engines: A case study of lucene. *International Research: Journal of Library and Information Science*, 7(3).

Sánchez-Macías, Armando y Veytia-Bucheli, M. G. (2019). Las competencias digitales en estudiantes de doctorado. un estudio en dos universidades mexicanas. *Academia y Virtualidad*, 12(1):7–30.

Santana Payan, Á, A. and Nájera Benavides, A. (2017). Prototipo de un chatbot para la solución de problemas comunes en el portal de la uacj. *Licenciatura en Ingeniería en Sistemas Computacionales*.

Sengul, C. and Kravets, R. (2006). Bypass routing: An on-demand local recovery protocol for ad hoc networks. *Ad Hoc Networks*, 4(3):380–397.

Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R., and Walker, J. H. (2003). Dspace: An open source dynamic digital repository.

Stankovic, J. A., Spuri, M., Di Natale, M., and Buttazzo, G. C. (1995). Implications of classical scheduling results for real-time systems. *Computer*, 28(6):16–25.

Tanenbaum, A. S. (2016). Lessons learned from 30 years of minix. *Communications of the ACM*, 59(3):70–78.

Tapia-Leon, M. and Chicaiza, J. (2020). Systematic search process for doctoral theses in centralized repositories. a study case in the context of educational innovation generated by ict. In *2020 XV Conferencia Latinoamericana de Tecnologías de Aprendizaje (LACLO)*, pages 1–9.

Tato, M. C. and Pichardo, F. M. P. (2005). Acerca de la selección de los temas de tesis de grado de master y de doctor en ciencias económicas. *Economía y Desarrollo*, 138(1):145–156.

Texier, J., De Giusti, M. R., Lira, A. J., Oviedo, N., and Villarreal, G. L. (2013). Dspace como herramienta para un repositorio de documentos administrativos en la universidad nacional experimental del Táchira. *Revista Interamericana de Bibliotecología*, 36(2):109–124.

Tulshan, A. S. and Dhage, S. N. (2018). Survey on virtual assistant: Google assistant, siri, cortana, alexa. In *International symposium on signal processing and intelligent recognition systems*, pages 190–201. Springer.

Urrutia Ortiz, G. G. (2020). Estudio de los beneficios del uso de chatbots en los procesos de atención a usuarios del sistema académico integral de la universidad técnica de babahoyo. B.S. thesis, Babahoyo, UTB-FAFI 2020.

Uzcátegui-Varela, J.-P. and Bracho-Orlandoli, R. (2021). Covid-19: contextos frágiles y objetivos del desarrollo sostenible. *Departamento de Medicina Preventiva y Social Departamento de Medicina Preventiva y Social Facultad de Medicina/Universidad de Los Andes Facultad de Medicina/Universidad de Los Andes*, 6(1):135.

Vadhera, A., Thute, A., Mala, S., and Shankar, A. (2021). Chatbot on covid-19 for sustaining good health during the pandemic. In *Latest Trends in Renewable Energy Technologies*, pages 271–284. Springer.

Venegas Mejía, V., Esquivel Grados, J., and Turpo-Gebera, O. (2019). Reflexiones sobre la investigación educativa y la investigación formativa en la universidad peruana. *Conrado*, 15(70):444–454.

Vlachou, A., Doulkeridis, C., Nørvåg, K., and Kotidis, Y. (2012). *Peer-to-peer query processing over multidimensional data*. Springer Science & Business Media.

Vu, Q. H., Lupu, M., and Ooi, B. C. (2010). Architecture of peer-to-peer systems. In *Peer-to-Peer Computing*, pages 11–37. Springer.

Wald, R., Heijseelaar, E., and Bosse, T. (2021). Make your own: The potential of chatbot customization for the development of user trust. In *Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*, pages 382–387.

Wallace, R., Tomabechi, H., and Aimless, D. (2003). Chatterbots go native: Considerations for an eco-system fostering the development of artificial life forms in a human world. *Published online: <http://www.pandorabots.com/pandora/pics/chatterbotsgonative.doc>*, 15.

Williams, A. E. (2017). Altmetrics: an overview and evaluation. *Online information review*.

Wu, Q. and Zhang, R. (2018). Common throughput maximization in uav-enabled ofdma systems with delay consideration. *IEEE Transactions on Communications*, 66(12):6614–6627.

Zawoznik, M. (2018). La comunicación científica y la práctica social de la escritura. *Revista Argentina de Microbiología*, 50(1):1–2.