



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS
AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

**Arquitectura genérica de chatbot educativo para
asistir el proceso de enseñanza-aprendizaje**

T E S I S

Que presenta

José Fidel Urquiza Yllescas

para obtener el Grado de

Doctor en Ciencias

en Computación

Directores de la Tesis

Dra. Sonia Guadalupe Mendoza Chapa

Dr. J. Guadalupe Rodríguez García

Ciudad de México

Diciembre, 2023

Resumen

En la actualidad, los chatbots han emergido como herramientas poderosas en diversos sectores, desde servicios comerciales, salud, turismo, hasta el ámbito educativo. La pandemia de COVID-19 obligó a muchos estudiantes y profesores a suspender las clases presenciales. Por ello, las escuelas y los gobiernos se vieron en la necesidad de continuar la educación a distancia, utilizando los recursos que proporciona Internet. Este hecho ha creado un mayor interés por los chatbots educativos, por lo que se han propuesto varios proyectos para desarrollar estas herramientas académicas, cada uno siguiendo su forma de implementación y abordando los problemas desde diferentes puntos de vista. En este contexto, en la presente tesis, se propone una arquitectura genérica de chatbot educativo que pueda asistir el proceso de enseñanza-aprendizaje. Para alcanzar este objetivo, se establece una metodología de investigación que comprende una serie de actividades con sus correspondientes productos resultantes. Se inicia con un Estudio de Mapeo Sistemático (EMS), definiendo preguntas de investigación y criterios de selección para obtener una muestra de artículos que serán revisados. Posteriormente, se lleva a cabo una revisión exhaustiva del estado del arte, generando cuadros comparativos de los trabajos revisados. Además, se examinan diferentes métodos para el diseño de la arquitectura de software. También, se propone una definición de lo que es un chatbot educativo y se establece una clasificación de chatbots educativos que es útil para el diseño de la arquitectura. Se realiza el diseño de la arquitectura de chatbots educativos, utilizando un método iterativo que resulta en la obtención de una arquitectura de tres niveles y una arquitectura de capas que delinean la distribución y estructura del chatbot, respectivamente. Finalmente, se implementa un chatbot educativo para el nivel medio superior y se realiza un caso de estudio para evaluar su funcionamiento y efectividad en el proceso de enseñanza-aprendizaje.

Palabras clave: chatbots educativos, proceso de enseñanza/aprendizaje, EMS, clasifi-

cación de chatbots, arquitectura de software

Abstract

Currently, chatbots have emerged as powerful tools in various sectors, from commercial services, health, tourism, to education. The COVID-19 pandemic forced many students and teachers to suspend in-person classes. For this reason, schools and governments saw the need to continue distance education, using the resources provided by the Internet. This fact has created greater interest in educational chatbots, which is why several projects have been proposed to develop these academic tools, each following its own way of implementation and addressing problems from different points of view. In this context, this thesis proposes a generic educational chatbot architecture that can assist the teaching-learning process. To achieve this objective, a research methodology is established that includes a series of activities with their corresponding resulting products. The work begins with a Systematic Mapping Study (SMS), defining research questions and selection criteria to obtain a sample of articles that must be reviewed. Subsequently, an exhaustive review of the state of the art is carried out, generating comparative tables of the studied works. Additionally, different methods for software architecture design are examined. We also propose a definition of what an educational chatbot is and establish a classification of educational chatbots that is useful for the design of the architecture. The research continues with the design of the architecture of educational chatbots, using an iterative method that results in obtaining a three-level architecture and a layered architecture that outline the distribution and structure of the chatbot, respectively. Finally, we proceed to the implementation of an educational chatbot for the high school level and a case study is carried out to evaluate its operation and effectiveness in the teaching-learning process.

Keyword: educational chatbots, teaching/learning process, SMS, chatbot classification, software architecture

***A mi amada hija y
su admirable mamá,
gracias por ser mi familia.
Con ustedes, todo este
esfuerzo cobra sentido.
Cada día a su lado es un
regalo que atesoro, y en
este viaje juntos, les dedico
mi amor infinito.***

Agradecimientos

A mis asesores Sonia Guadalupe Mendoza Chapa y José Guadalupe Rodríguez García.

Mi más sincero agradecimiento por su invaluable apoyo y orientación a lo largo de este proyecto de investigación. Me considero sumamente afortunado por contar con su respaldo emocional a lo largo de este arduo proceso. Su disposición constante para escucharme, brindarme sabios consejos y motivarme fue fundamental. Gracias a ustedes, siempre me sentí respaldado y alentado. Su dedicación a la excelencia académica es inspiradora, y me han transmitido un profundo amor por la investigación. Estoy seguro de que las lecciones aprendidas bajo su guía serán un recurso invaluable a lo largo de mi carrera como investigador.

A mis sinodales Sergio Víctor Chapa Vergara, Eduardo López Domínguez y Enrique Ruíz-Velasco Sánchez.

Por su tiempo y valiosas observaciones durante la revisión de esta tesis. Su experiencia y perspicacia han enriquecido enormemente esta investigación.

A Luis Martín Sánchez Adame.

A quién agradezco su apoyo en momentos cruciales y ahora, más que nunca, valoro el trabajo realizado y el que nos espera en un futuro.

A las y los profesores e investigadores del Departamento de Computación del Cinvestav.

Aprecio profundamente el impacto que han tenido en mi desarrollo como estudiante y futuro profesional.

A José Luis, Erika y Katia.

Por su labor incansable y compromiso con la excelencia en cada tarea que desempeñan. Cada uno de ustedes contribuye de manera invaluable al funcionamiento eficiente y exitoso del Departamento de Computación del Cinvestav.

A papá y mamá.

Gracias por enseñarme el camino correcto y por estar siempre presentes. Este logro también es de ustedes.

A mis hermanas y sobrinos.

Agradezco de corazón todo el amor y apoyo que me han brindado. Ustedes tienen un significado muy especial en mi vida.

A mis amigas y amigos.

Los años de amistad han sido invaluable.

Al Cinvestav.

La institución ideal para alcanzar el más alto grado de estudios.

Al Conahcyt.

Por el apoyo otorgado para los estudios de doctorado.

A los estudiantes del Plantel Vasco de Quiroga del IEMS.

Agradezco sinceramente su tiempo y esfuerzo para colaborar en la realización de las pruebas.

Al IEMS-CDMX.

Por las facilidades brindadas para continuar mis estudios.

Índice general

1. Introducción	1
1.1. Motivación	2
1.2. Planteamiento del problema	4
1.3. Hipótesis de Investigación	7
1.4. Objetivos general y específicos del proyecto	7
1.5. Metodología de investigación	8
1.5.1. Actividad: Estudio de Mapeo Sistemático	8
1.5.2. Actividad: Estado del Arte	10
1.5.3. Actividad: Diseño de la arquitectura genérica para chatbots educativos	11
1.5.4. Actividad: Implementación del chatbot educativo	12
1.5.5. Actividad: Caso de estudio	12
1.6. Contribuciones	13
1.7. Estructura del documento	14
2. Marco Teórico	16
2.1. Estudio de Mapeo Sistemático	16
2.2. Clasificaciones de chatbots	18
2.2.1. Elementos teóricos para desarrollar clasificaciones	20
2.2.2. Método de clasificación iterativo	21
2.3. Arquitectura de software	24
2.3.1. Arquitecturas de chatbots	25
2.3.2. Métodos para el diseño de arquitecturas	27

3. Desarrollo del SMS	30
3.1. Preguntas de investigación	30
3.2. Estrategia de búsqueda	31
3.3. Elaboración de la expresión de búsqueda	32
3.4. Directrices para la selección de estudios	32
3.5. Clasificación de estudios	33
3.6. Extracción de Datos	34
3.7. Respuestas a las preguntas de investigación	34
3.7.1. Pregunta de investigación PI-1	36
3.7.2. Pregunta de investigación PI-2	39
3.7.3. Pregunta de investigación PI-3	40
3.8. Discusión	42
4. Estado del Arte	46
4.1. Clasificación de chatbots educativos	46
4.1.1. Aplicación del método de Nickerson et al.	47
4.1.2. Resultados	49
4.2. Síntesis de los trabajos relacionados	53
4.2.1. Educación en línea	53
4.2.2. Servicios escolares	59
4.2.3. Estudiantes y profesores	62
4.3. Contribuciones y características de los chatbots educativos	69
4.4. Aspectos técnicos de los chatbots educativos	75
4.5. Discusión	79
5. Arquitectura genérica de chatbots educativos	83
5.1. Análisis preliminar	83
5.2. Método ADD 3.0	85
5.3. Entradas	88
5.3.1. Propósito del diseño	90
5.3.2. Modelo de casos de uso	90
5.3.3. Escenarios de atributos de calidad	93
5.3.4. Restricciones	97
5.3.5. Intereses arquitectónicos	98

5.4.	Proceso de diseño	98
5.4.1.	Inicio iteración	99
5.4.1.1.	Paso 1: Revisar las entradas	100
5.4.1.2.	Paso 2: Establecer la meta de iteración seleccionando las directrices	100
5.4.1.3.	Paso 3: Elegir uno o más elementos del sistema a refinar .	100
5.4.1.4.	Paso 4: Elegir uno o más conceptos de diseño que satisfagan las directrices	100
5.4.1.5.	Paso 5: Crear instancias de elementos arquitectónicos, asignar responsabilidades y definir interfaces	102
5.4.1.6.	Paso 6: Vistas de sketch y registro de las decisiones de diseño	103
5.4.1.7.	Paso 7: Realizar análisis del diseño actual y revisar el objetivo de la iteración y el logro del propósito del diseño . .	106
5.5.	Discusión	106
5.5.1.	ADD 3.0	108
5.5.2.	Método	109
5.5.3.	Directrices	110
5.5.4.	Arquitectura, capas y componentes	111
5.5.5.	Usuarios de la arquitectura	114
5.5.6.	Comparación de nuestra propuesta con otras arquitecturas	115
5.5.7.	Un chatbot inteligente	116
5.5.8.	Cruce del diseño instruccional	117
6.	Implementación	119
6.1.	Herramientas	119
6.1.1.	Software de aplicación	119
6.1.1.1.	Diagramas	119
6.1.2.	Software de programación	120
6.1.2.1.	Visual Studio Code	120
6.1.2.2.	Lenguaje de programación y gestor de paquete	120
6.1.2.3.	Servicios en la nube	121
6.1.2.4.	Controlador de versiones	121
6.2.	Cinvesbot Servicio Escolar	122

6.2.1.	Construcción y configuración	123
6.2.1.1.	Construcción del agente	123
6.2.1.2.	Autenticación de usuario	129
6.2.2.	Interacción con otros sistemas	130
6.2.3.	Programación del chatbot	132
6.2.4.	Interfaz de usuario	134
6.3.	Discusión	135
7.	Caso de estudio del chatbot educativo	138
7.1.	Guía de aplicación	138
7.2.	Entradas para la guía de aplicación	140
7.2.1.	Población	140
7.2.2.	Espacio y equipo	141
7.2.3.	Formulario	141
7.2.4.	Prototipo de chatbot educativo	143
7.2.5.	Instrucciones y tareas	144
7.2.6.	Aspectos logísticos a considerar	144
7.3.	Resultados	145
7.3.1.	Proceso del caso de estudio	145
7.3.2.	Cuestionario de AttrakDiff	147
7.4.	Discusión	151
8.	Conclusiones y trabajo futuro	156
8.1.	Contribuciones principales	156
8.2.	Trabajo futuro	158

Índice de figuras

1.1. Intervención de diferentes elementos para construir chatbots educativos.	6
1.2. Actividades y productos de la metodología propuesta.	9
2.1. Método de siete pasos de Nickerson.	23
3.1. Ejecución del SMS.	35
3.2. Estadísticas de propuestas por año.	35
3.3. Estadísticas de los tipos de estudios seleccionados por buscador.	36
4.1. Estructura de la clasificación de chatbots y sus características en el dominio educativo.	49
4.2. Abanico de posibilidades para realizar investigación en chatbots educativos.	71
4.3. Mecanismos de IA utilizados en las propuestas.	76
4.4. Líneas de investigación para futuras propuestas de chatbots educativos.	82
5.1. Primera aproximación de una arquitectura genérica de chatbots educativos.	84
5.2. Modelo de casos de uso del chatbot educativo propuesto.	91
5.3. Flujo del proceso de diseño de ADD 3.0	99
5.4. Arquitectura de tres niveles para el chatbot educativo.	103
5.5. Arquitectura de capas del chatbot educativo.	105
6.1. Consola de Dialogflow.	124
6.2. Estructura de Cinvesbot Servicio Escolar y sus respectivos <i>intentos</i>	127
6.3. Lista de intentos para el chatbot Cinvesbot Servicio Escolar en Dialogflow.	128
6.4. Configuración de la interacción con otros sistemas.	131
6.5. Estructura del chatbot en Angular.	133
6.6. Estructura del chatbot en Nodejs.	134

6.7. Interfaz de Cinvesbot Servicio Escolar.	136
7.1. El modelo de <i>AttrakDiff</i> ilustra cómo las cualidades pragmáticas y hedónicas influyen en la percepción subjetiva del atractivo dando lugar a comportamientos y emociones consecuentes.	142
7.2. Cuestionario <i>AttrakDiff</i> con los diferenciales semánticos.	143
7.3. Estudiantes interactuando con el chatbot.	146
7.4. Portafolio de resultados.	148
7.5. Valores medio de los estudiantes.	149
7.6. Media de la descripción de palabras pares.	150

Índice de cuadros

3.1. Preguntas de investigación.	31
3.2. Elementos en la estructura de las preguntas de investigación.	31
3.3. Términos y sinónimos para las expresiones de búsqueda.	32
3.4. Cadenas de búsqueda para las preguntas de investigación.	32
3.5. Criterios generales de inclusión.	33
3.6. Criterios generales de exclusión.	33
3.7. Criterios generales de selección.	33
3.8. Propiedades a extraer.	34
4.1. Detalle de las iteraciones con respecto a las condiciones finales objetivas y subjetivas.	47
4.2. Resumen de las contribuciones del estado del arte en chatbots educativos. .	70
4.3. Resumen de las características de chatbot educativos.	73
4.4. Aspectos técnicos de los chatbots.	77
5.1. Descripción de los casos de uso del chatbot educativo.	92
5.2. Catálogo de atributos de calidad comúnmente empleados en el proceso de diseño.	94
5.3. Escenarios de atributos de calidad para un chatbot educativo.	96
5.4. Restricciones para el diseño del chatbot educativo.	97
5.5. Interés arquitectónico del chatbot educativo.	98
5.6. Arquitectura de referencia para el chatbot educativo.	101
5.7. Tablero Kanban para la primera iteración.	107
7.1. Tiempo en minutos de las interacciones de los estudiantes con el chatbot. .	147

Capítulo 1

Introducción

Durante décadas, no sólo investigadores sino también autores de libros y películas de ciencia ficción han estado hablando de lo que se conoce como chatbot. Alan Turing mencionó implícitamente lo que podría ser un chatbot [1]. En la actualidad, los chatbots son programas de cómputo que interactúan con los seres humanos en lenguaje natural [2]. No obstante, en 1966, uno de los pioneros en este campo, Weizenbaum [3], presentó su famoso chatbot ELIZA, cuyo objetivo era mantener una conversación con un ser humano. ELIZA se basa en plantillas y técnicas de coincidencia de patrones, pero carece de capacidad de razonamiento. El diseño de este chatbot se limita a la tecnología y al conocimiento de la época. Sin embargo, este trabajo de investigación y los posteriores poco a poco fueron estableciendo las bases para el diseño de los chatbots actuales.

Por otro lado, Dale [4] menciona que los chatbots son comúnmente conocidos como *asistentes digitales e interfaces conversacionales*. Sin embargo, se han propuesto varios nombres para referirse a ellos como: *agentes inteligentes de software conversacional*, *sistemas de software conversacional*, *chatterbots* o simplemente *bots*. La comunidad de Procesamiento de Lenguaje Natural utiliza el término *agente conversacional* para nombrar sistemas conversacionales, mientras que el término *chatbot* se emplea para denotar un subconjunto de sistemas que admiten conversaciones informales [5].

Los chatbots funcionan principalmente buscando palabras clave, frases y ejemplos que se han introducido en una base de conocimiento [6]. Sin embargo, hasta ahora los chatbots no tienen la capacidad de lograr una interacción natural, ya que los usuarios pueden darse cuenta de que están hablando con un *bot* [7].

Inokuchi *et al.* [8], Veglis y Maniou [9], y Matthies *et al.* [10] proponen una arquitectura

de software para un chatbot. La arquitectura propuesta por todos estos autores es cliente-servidor, en donde el cliente realiza una solicitud a otro programa, conocido como servidor, el cual procesa esa solicitud y le da una respuesta al cliente [11]. En el caso de un chatbot, el cliente es la interfaz de usuario, a través de la cual el usuario interactúa con el sistema, y el servidor cuenta con un proceso analizador de reglas lógicas, una base de conocimiento o base de datos y mecanismos de inferencia, entre otros componentes, para procesar las solicitudes.

Según un informe de *Business Insider*, para el año 2020 se esperaba que alrededor del 80% de las empresas en el sector comercial tenga previsto utilizar chatbots [12]. Además, hay un interés creciente en los chatbots [13] y, a medida que aumenta su uso en Internet, las interacciones basadas en chat podrían haber rondado en el 85% tan sólo en el 2020 [14].

Es posible encontrar chatbots en varios dispositivos y servicios, *e.g.*, teléfonos inteligentes, sitios web, redes sociales y mensajeros, que sirven a varios sectores [15]. Algunos chatbots ofrecen servicios al cliente [16] y asesoramiento financiero [17] sobre negocios [18] y mercados de criptomonedas [19]. Otros están destinados a responder Preguntas Frecuentes (*Frequently Asked Questions*, FAQ) [6] y hacer recomendaciones [20] en turismo [21] y servicios de admisión [22]. Apple, Microsoft, Amazon, Google y Facebook también ofrecen chatbots como asistentes personales [23]. Además, existen chatbots especializados en atención a personas mayores [24], servicios médicos [25] y adicciones al tabaco, drogas y alcohol [26, 27, 28]. Incluso hay chatbots que ayudan en actividades agrícolas [29], educación [30] y lucha contra la explotación infantil [31].

Los chatbots modernos son capaces de comprender el contexto de una conversación y aprender de ella, mejorando con el tiempo, gracias al uso de técnicas de aprendizaje automático (*Machine Learning*) [32] y arquitecturas informáticas novedosas [33].

1.1. Motivación

La educación es un sector esencial para el uso de chatbots [34]. Hoy en día, los estudiantes reciben una parte importante de su educación a través de información en línea, *e.g.*, temas de clase, tareas, prácticas, cuestionarios y ensayos. Así, los chatbots pueden brindar una valiosa ayuda en el proceso de enseñanza-aprendizaje. Por un lado, los chatbots podrían atender a miles de estudiantes al mismo tiempo, a través de preguntas y respuestas. Por otro lado, los chatbots podrían contrarrestar la escasez de profesores y cursos

en escuelas y universidades de todo el mundo [35], convirtiéndose en una poderosa herramienta. Aunque el uso de los chatbots todavía es limitado [36], son muy apreciados por los docentes para obtener mejores resultados en el proceso de enseñanza-aprendizaje [37]. Por ello, el desarrollo y uso de chatbots se está convirtiendo en un tema de gran interés para las universidades [38, 39].

Con la aparición de la pandemia del COVID-19, los gobiernos se vieron obligados a implementar mecanismos para que los estudiantes continuaran con sus cursos. Aunque las escuelas cerraron, los cursos continuaron. La preocupación de gobiernos y autoridades encargadas de la educación se centró en no perder el año escolar. En consecuencia, los alumnos tuvieron que adaptarse a las nuevas condiciones de su formación académica y así continuar con sus clases en línea. Este cambio abrupto afectó directamente a los proyectos educativos tradicionales que se centran principalmente en las clases presenciales [40].

Los antiguos chatbots se implementaron de forma sencilla. Tenían un conjunto de preguntas específicas programadas con respuestas predeterminadas. En consecuencia, estos chatbots no siempre fueron útiles y no tenían la capacidad de interactuar con los usuarios [41]. En la actualidad, es posible encontrar varios trabajos que intentan dotar a los chatbots con Inteligencia Artificial, con el fin de promover una interacción fluida con los usuarios [42]. Además, es necesario que el diseño de chatbots con fines educativos tenga en cuenta el nivel de competencia de los estudiantes [43].

Los chatbots para el sector educativo no son nuevos. Antes de la pandemia, ya habían propuestas que involucraban chatbots en dicho dominio. Por mencionar algunas, encontramos: T-Bot/Q-Bot [44], CHARLIE [45], UMT-BOT [46], FIT-EBot [47], LiSA [48], DINA [22], y Ubibot [49]. Sin embargo, los chatbots educativos aún son limitados [36] y la investigación que se ha realizado en esas propuestas no es exhaustiva.

En el Cinvestav-IPN se cuenta con un antecedente de un chatbot educativo. Hernández-León [50] introduce un chatbot diseñado para adaptarse a diferentes estilos de aprendizaje, con el objetivo de optimizar la comunicación entre estudiantes y profesores. Este sistema incorpora un par de personalidades para fomentar una mayor conexión empática con los alumnos. Aunque considera ciertos modelos de aprendizaje, su implementación no se adhiere completamente a un modelo educativo específico. El prototipo, desarrollado en Java, emplea la biblioteca *Stanford NLP* y almacena su base de conocimientos en archivos de texto plano. Para evaluar su prototipo, se utilizó la herramienta *AttrakDiff*, midiendo

la experiencia de usuario en un grupo de estudiantes de secundaria.

Sin embargo, en nuestro planteamiento, buscaremos abordar el diseño de estas herramientas educativas desde una perspectiva estructural genérica revisando el estado actual del conocimiento y pensando en otra población de estudiantes.

1.2. Planteamiento del problema

En la actualidad, existen diversos enfoques educativos adaptados a diferentes niveles escolares, como el método Montessori [51], Waldorf [52], Gamificación [53] o Aula Invertida [54]. La adopción de estos métodos varía según cada país, incluso se observan adaptaciones regionales, por ejemplo, los sistemas educativos nórdicos presentan relaciones con el método Montessori [55]. Así como Finlandia destaca con su enfoque educativo [56], México también cuenta con su proyecto educativo del Instituto de Educación Media Superior (IEMS)¹, un proyecto vanguardista e innovador destinado a la población de la Ciudad de México.

Para el desarrollo de una educación de calidad, el docente tiene un papel fundamental. Éste debe planificar cómo abordar a los estudiantes, definir las actividades del curso, las estrategias de evaluación y la metodología a seguir, entre otros aspectos. Sin embargo, algunas barreras, como la falta de empatía, problemas personales o malas interpretaciones, pueden afectar la comunicación entre docentes y alumnos, alterando la dinámica educativa.

El impacto de las Tecnologías de la Información y la Comunicación (TIC) en el proceso de enseñanza-aprendizaje es cada vez más pronunciado. Los proyectos educativos actuales incorporan estas herramientas como un valioso recurso para estudiantes y profesores por igual. Aunque el aprovechamiento adecuado de las TIC es esencial para el desarrollo estudiantil, su efectividad se encuentra en gran parte guiada por la dirección brindada por los educadores. La integración de las TIC, con especial énfasis en los chatbots, puede ser de gran utilidad en el contexto educativo por su capacidad comunicativa mediante lenguaje natural [57] tanto para alumnos como docentes, abordando cuestiones académicas y administrativas [58]. Por ejemplo, los chatbots pueden brindar tutorías personalizadas, beneficiando a aquellos estudiantes que necesitan asistencia en una asignatura específica o encuentran dificultades en el aprendizaje autónomo. Además, estos chatbots pueden

¹<https://iems.cdmx.gob.mx/>

ofrecer retroalimentación instantánea a los estudiantes sobre su trabajo, contribuyendo a que aprendan de sus errores y profundicen en la comprensión del contenido. Asimismo, la implementación de chatbots puede enriquecer el entorno educativo con elementos multimedia, creando una experiencia de aprendizaje más atractiva e inmersiva que fomente la motivación y el compromiso estudiantil. Estas herramientas, ancladas en el currículo institucional y abarcando diversas áreas de formación, tienen el potencial de atenuar la falta de docentes y cursos en instituciones educativas en todo el mundo [35].

La evolución de los chatbots educativos, a lo largo del tiempo, ha sido influenciada por una variedad de elementos. La Figura 1.1 resume las diversas áreas, herramientas y tecnologías que han contribuido a esta transformación. Por un lado, destaca la relevancia de las Humanidades, la Pedagogía y la Psicología en este tipo de sistemas. Además, en un entorno educativo, es recomendable que un chatbot tenga como base un modelo educativo. En la literatura educativa existen diversos modelos discutidos por la comunidad, como lo pueden ser el conductivismo [59], constructivismo [60], educación tradicional [61], entre otros, que buscan establecer coherencia, estructura, metodología y criterios de evaluación. Por otro lado, desde la perspectiva de las Ciencias de la Computación, que es nuestro enfoque principal, los chatbots educativos se apoyan en gran medida en la Inteligencia Artificial (IA), la Interacción Humano-Computadora (IHC) y la Ingeniería de Software (IS). La colaboración entre las diferentes ramas tiene un impacto directo en las posibles herramientas y tecnologías que pueden ofrecer para la construcción de un chatbot educativo. En consecuencia, es importante destacar que no existe un único tipo o definición de chatbot educativo. Sus características y alcance son definidos por los contextos de uso y objetivos específicos que persiguen.

Debemos de tener en cuenta que cada área tiene sus propias limitaciones que afectan directamente el desarrollo de un chatbot educativo. Por ejemplo, en IA, los chatbots tienen dificultades para entender el lenguaje natural, sin necesidad de utilizar comandos o instrucciones para interactuar con los usuarios, así como generar respuestas coherentes y realistas a las consultas de los usuarios [62].

En cuanto a la IHC, se encuentran obstáculos para establecer interacciones intuitivas y efectivas entre los usuarios y el chatbot educativo. Esto abarca desde la creación de interfaces de usuario fáciles de usar y atractivas, hasta la personalización y empatía del chatbot hacia el usuario [63], lo cual influye en la experiencia general de aprendizaje.

Por último, en la IS, se presentan desafíos particularmente complejos. La programación

y el mantenimiento de un chatbot educativo demandan un enfoque meticuloso [62]. La falta de una estructura estándar para chatbots de propósito general, el diseño requerido para la integración en diversas plataformas y canales de comunicación, la creación de componentes específicos para el Procesamiento del Lenguaje Natural y la implementación de medidas de seguridad para proteger la privacidad de los usuarios son aspectos críticos que requieren atención cuidadosa y soluciones innovadoras.

Como se observa, cada una de esas áreas tiene limitaciones que deben ser solventadas en el desarrollo de un sistema con las características de un chatbot, lo que se refleja en la falta de una arquitectura genérica para el desarrollo de este tipo de sistemas. En consecuencia, la mayoría de las propuestas han seguido esquemas específicos de desarrollo, lo cual ha provocado que se tenga una gran diversidad de arquitecturas y modelos que hace complicado elegir una solución específica. Esto se nota en el gran número de propuestas de chatbots educativos que son iniciados desde cero, como descubriremos en el estado del

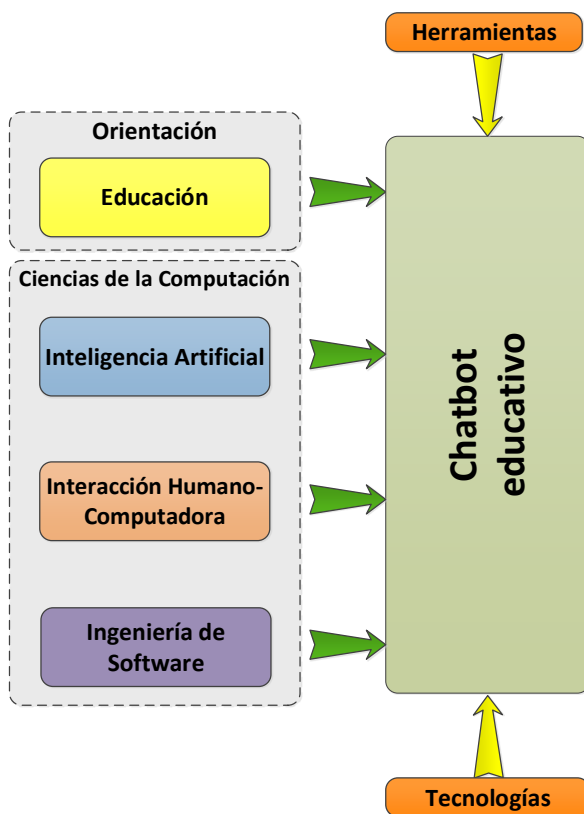


Figura 1.1: Intervención de diferentes elementos para construir chatbots educativos.

arte de esta investigación.

Los chatbots educativos tienen un enorme potencial para revolucionar la forma en que aprendemos. Al combinar las disciplinas de las Humanidades, la Pedagogía, la Psicología, la IA, la IHC y la IS, podemos desarrollar chatbots educativos que sean efectivos, personalizados y atractivos para los estudiantes.

Las contribuciones en el ámbito del conocimiento han dedicado esfuerzos considerables para mitigar las restricciones que aún podemos encontrar en chatbots, aunque este proceso aún presenta un camino significativo por recorrer. A medida que se realicen más aportaciones, muchas de estas dificultades comenzarán a reducirse, aunque también cabe la posibilidad de que puedan surgir nuevas en el proceso.

1.3. Hipótesis de Investigación

La hipótesis de investigación de este proyecto de tesis es la siguiente: Basándose en el análisis de las características de chatbots existentes, es posible desarrollar una arquitectura general para chatbots educativos que, al ser implementada, brinde experiencias de uso positivas.

1.4. Objetivos general y específicos del proyecto

A continuación, se plantean los objetivos de este trabajo:

- Objetivo general:
 - Desarrollar una arquitectura genérica de chatbot educativo, que sirva de intermediario entre el personal educativo y el alumnado, para apoyar los procesos de enseñanza-aprendizaje en los aspectos académicos.
- Objetivos específicos:
 1. Realizar un análisis del estado del arte de chatbots enfocados a la educación para identificar sus fortalezas y debilidades, así como las herramientas y tecnologías utilizadas en su construcción.

2. Proponer una clasificación de chatbots educativos que facilite la identificación de características fundamentales de este tipo de sistemas.
3. Analizar diferentes métodos para el diseño de arquitecturas de software, con el fin identificar los elementos clave de nuestra propuesta.
4. Diseñar una arquitectura genérica de chatbots educativos, tomando como referencia la clasificación propuesta.
5. Con base en la arquitectura propuesta, crear un prototipo de chatbot capaz de comunicarse con profesores y estudiantes, utilizando lenguaje natural.
6. Plantear el caso de estudio, considerando estudiantes y su nivel de estudios, para orientar el desarrollo del proyecto.

1.5. Metodología de investigación

En la metodología presentada en este trabajo de investigación se definen una serie de actividades, tareas y productos a lograr como puede observarse en la Figura 1.2. Las actividades se representan mediante rectángulos de esquinas redondeadas en diversos colores, mientras que los productos resultantes de estas actividades se denotan por medio de rectángulos de esquinas planas. Cabe mencionar que una actividad puede generar como salida uno o varios productos que sirven de entrada a las actividades posteriores.

1.5.1. Actividad: Estudio de Mapeo Sistemático

Esta actividad consiste en realizar un estudio de mapeo sistemático (*Systematic Literature Review*, SMS) [64] mediante las siguientes tareas:

1. Revisar la frontera del conocimiento, desafíos y problemas en el campo de los chatbots.
2. Revisar y ejecutar la metodología SMS para identificar la línea base en el dominio propuesto.
3. Obtener la muestra de artículos (mapa de los resultados) de chatbots educativos que se publicaron en un periodo de dieciséis años (2006-2022).

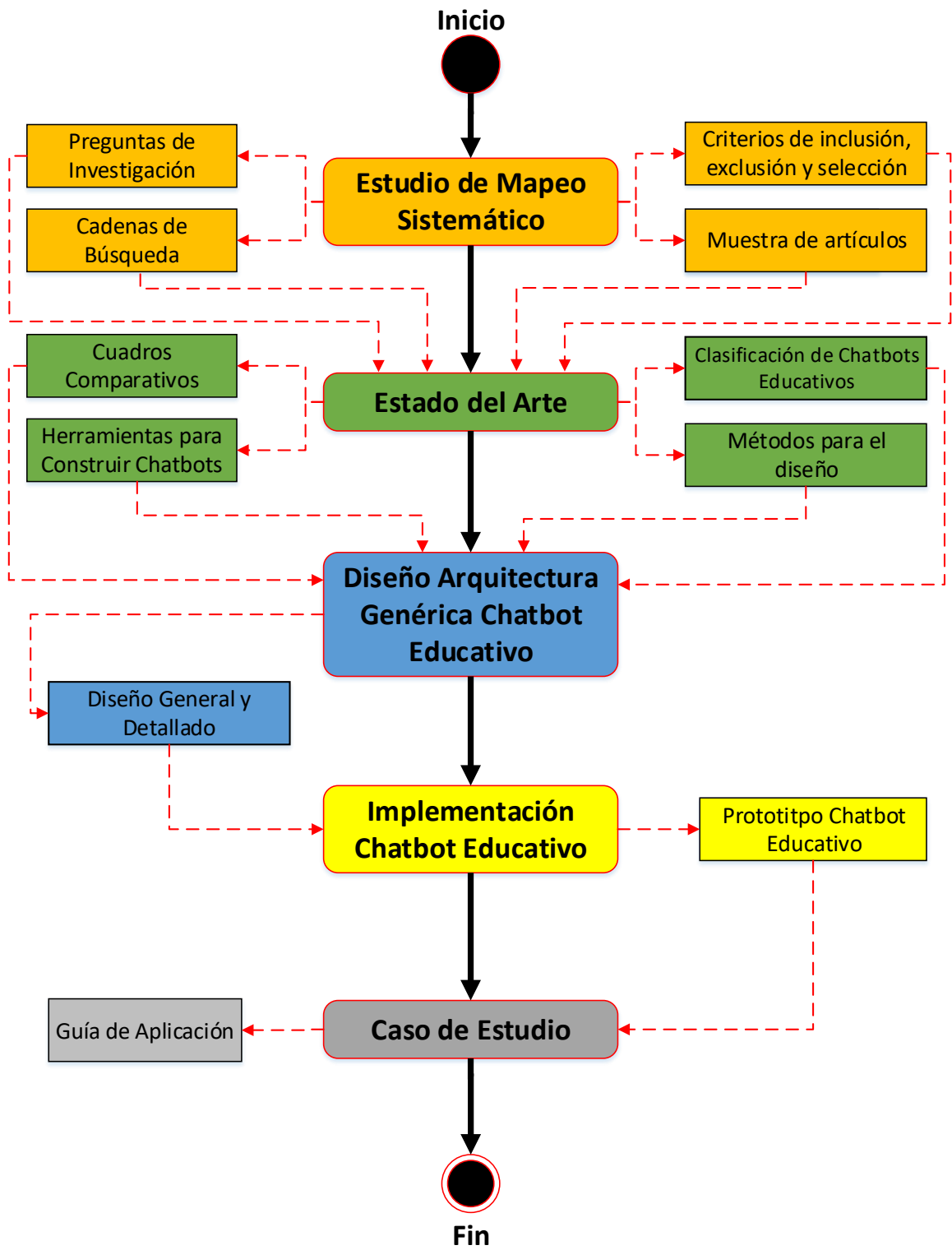


Figura 1.2: Actividades y productos de la metodología propuesta.

Al concluir esta actividad se obtienen los productos:

- Preguntas de investigación.
- Cadena de búsqueda.
- Criterios de inclusión, exclusión y selección.
- Muestra de los artículos filtrados para revisión.

1.5.2. Actividad: Estado del Arte

Esta actividad tiene las siguientes tareas:

1. Analizar la muestra de artículos obtenidos del SMS para llevar a cabo lo siguiente:

a) Elaborar un resumen de cada artículo describiendo:

- Contribución de los autores.
- Área o campo en la que se hace el aporte.
- Grado de reproducibilidad de la propuesta (alto, medio o bajo).

b) Realizar un resumen con las características de los chatbots educativos identificando si:

- El chatbot cuenta con elementos gráficos o animados que permitan un acercamiento, empatía o confianza hacia el usuario.
- El diálogo del chatbot permite la comunicación en diferentes idiomas.
- Existen diferentes mecanismos de interacción (texto, voz o ambos).
- El chatbot fue construido desde cero.
- El chatbot es un parte de un sistema de software más complejo.

c) Describir las herramientas o tecnologías que utilizaron los autores para construir los chatbots:

- Mecanismos de Inteligencia Artificial.
- Arquitectura de Software implementada.
- Tecnologías y herramientas empleadas.
- Tipos de almacenamiento de la información.

2. Catalogar las herramientas para construir chatbots por:
 - Mecanismos de razonamiento.
 - Lenguajes de programación.
 - Manejadores de bases de datos.
 - Herramientas para el procesamiento de lenguaje natural.
 - Interfaz de usuario para la comunicación.
 - Coincidencia de patrones.
 - Intérpretes para la coincidencia de patrones.
 - Herramientas que permiten la interacción en varios idiomas.
3. Desarrollar una clasificación de los chatbots educativos, utilizando un método iterativo.
4. Revisar y comparar diversas propuestas para el diseño de la arquitectura de software.
5. Discutir los resultados obtenidos para definir la contribución de la tesis en línea con los objetivos del proyecto.

Al concluir esta actividad, se obtienen los siguientes productos:

- Cuadros comparativos de los trabajos analizados.
- Herramientas para construir chatbots.
- Clasificación de chatbots educativos.
- Métodos para el diseño de la arquitectura de software.

1.5.3. Actividad: Diseño de la arquitectura genérica para chatbots educativos

Esta actividad involucra las siguientes tareas:

1. Utilizar un método para diseñar la arquitectura de software genérica del chatbot educativo.

2. Construir las entradas que requiera el procedimiento.
3. Seguir los pasos que establece el método.
4. Realizar las vistas y diseños de la arquitectura.

Al concluir esta actividad se obtienen el producto:

- Diseño general y detallado.

1.5.4. Actividad: Implementación del chatbot educativo

Esta actividad requiere de las siguientes tareas:

1. Escoger el nivel educativo para el chatbot.
2. Seleccionar el tipo de chatbot educativo a implementar.
3. Elegir la tecnología y herramientas para la programación.
4. Implementar la arquitectura y componentes establecidos en el diseño general y detallado.
5. Desarrollar el prototipo de chatbot genérico educativo.

Al concluir esta actividad se obtiene el producto:

- Prototipo de chatbot genérico educativo.

1.5.5. Actividad: Caso de estudio

Esta actividad implica la siguiente tarea:

1. Establecer los mecanismos y las estrategias que se utilizarán para realizar el caso de estudio con estudiantes.

Al concluir esta actividad se obtiene el producto:

- Guía de aplicación.

La metodología se desarrolló de manera específica para lograr los objetivos de la investigación. Está planeada y diseñada de tal manera que se pueda replicar y obtener los mismos productos.

En la Figura 1.2 se observa que, desde la primera actividad, se obtienen resultados que serán utilizados en las siguientes actividades. Los beneficios de hacer esto es tener un mejor control y claridad de lo que se va desarrollando de manera puntual. Con las dos primeras actividades se logran obtener más productos, pero las últimas tres se obtienen los resultados más importantes de esta tesis.

1.6. Contribuciones

Las contribuciones en una línea de investigación desempeñan un papel crucial al establecer un camino para el desarrollo del conocimiento en un campo específico, como es el caso de los chatbots y la educación. Estas contribuciones se caracterizan por presentar nuevos descubrimientos, enfoques y soluciones a problemas existentes (como veremos en el resto del documento de tesis). Al compartir los aportes con la comunidad, se fomenta un valioso intercambio de ideas y se impulsa el progreso colectivo en el ámbito académico y científico. Además, es importante destacar que nuestras colaboraciones tienen el potencial de generar un impacto directo en la práctica, influir en aspectos educativos, fortalecer procesos de enseñanza-aprendizaje y abrir nuevas oportunidades de investigación. En definitiva, las contribuciones en una línea de investigación son fundamentales para promover el avance y la innovación en el campo de estudio, beneficiando tanto a la comunidad académica como a la sociedad en general.

A continuación, se esboza de manera general las contribuciones desarrolladas en este proyecto de tesis, las cuales abarcan los siguientes aspectos:

1. **Definición de chatbot educativo:** se proporciona una definición precisa y comprensible de lo que constituye un chatbot educativo, estableciendo una base sólida para comprender su naturaleza y aplicaciones en el ámbito educativo.
2. **Clasificación de chatbots educativos:** se elabora una exhaustiva clasificación de los chatbots educativos, categorizándolos de manera estructurada en función de sus características y funcionalidades específicas. Esta clasificación facilitará el diseño y construcción de un chatbot enfocado a la educación.

3. **Arquitectura genérica de chatbots educativos:** se propone una arquitectura genérica para los chatbots educativos, ofreciendo un marco conceptual que facilita la implementación de funcionalidad en un entorno educativo.
4. **Estudio de mapeo sistemático sobre chatbots educativos:** se realiza un estudio que recopila y analiza de manera exhaustiva la literatura existente relacionada con los chatbots educativos.
5. **Panorama (*survey*) de los chatbots educativos:** se presenta una visión general completa y actualizada del estado del arte en este campo, identificando tendencias, enfoques comunes, brechas en el conocimiento. Este panorama proporciona una visión práctica y contextualizada de la realidad actual de los chatbots educativos.
6. **Experiencias en el desarrollo de chatbots enfocados a la educación:** se muestra el diseño y desarrollo de la implementación de un prototipo de chatbot educativo utilizando la arquitectura propuesta.
7. **Guía de aplicación:** se propone una guía que consta de una serie de entradas para llevar a cabo un caso de estudio de manera efectiva.

En conjunto, estas contribuciones aportan conocimientos significativos y fundamentales para el campo de los chatbots educativos, tanto desde una perspectiva teórica como práctica, y establecen una base sólida para futuras investigaciones y avances.

1.7. Estructura del documento

Este documento de tesis está dividido de la siguiente manera:

- En el Capítulo 2 se presenta el marco teórico. En primer lugar, se describe un método sistemático para obtener una muestra de artículos a analizar en la presente tesis. Después, se exponen diferentes taxonomías que se han propuesto sobre chatbots en general y un método que permite desarrollar clasificaciones. Por último, se expone el concepto e importancia de la arquitectura de software, así como las principales arquitecturas que han sido implementadas en chatbots y una breve descripción y comparación de métodos enfocados al diseño de arquitecturas.

- En el Capítulo 3 se muestra el desarrollo del método para la revisión sistemática, así como los resultados obtenidos después de utilizarlo.
- En el Capítulo 4 se expone el estado del arte. Primero se utiliza un método para establecer una clasificación de chatbots enfocados a la educación y, a continuación, se presentan los trabajos del estado del arte organizados de acuerdo a esa clasificación. Además, se presentan las contribuciones, características y aspectos técnicos de dichos trabajos.
- En el Capítulo 5 se presenta el método a utilizar para el diseño y posteriormente se detalla el proceso para la obtención de la arquitectura genérica de chatbots educativos.
- En el Capítulo 6 se describe la implementación de un prototipo de chatbot educativo, haciendo mención de las herramientas que se utilizaron para ello.
- En el Capítulo 7 muestra el caso de estudio con estudiantes de educación media superior y se analizan los resultados.
- En el Capítulo 8 se presentan las conclusiones y trabajo futuro de esta tesis.

Capítulo 2

Marco Teórico

En este capítulo, se presenta el marco teórico. Primeramente, se explica el método utilizado para realizar la revisión sistemática que permitió obtener el conjunto de artículos analizados en la presente tesis (cf. Sección 2.1). A continuación, se presentan diversas clasificaciones de chatbots en general, así como el método utilizado para desarrollar la clasificación propuesta en esta tesis (cf. Sección 2.2). Finalmente, se introduce el concepto e importancia de la arquitectura de software de chatbots, se exploran, de manera general, las diferentes arquitecturas que se han implementado en el desarrollo de chatbots y se comparan algunos métodos que se pueden utilizar para el diseño de arquitecturas de software (cf. Sección 2.3).

2.1. Estudio de Mapeo Sistemático

El origen de la revisión sistemática de la literatura (*Systematic Literature Review*, SLR) se da en las Ciencias de la Salud, con el objetivo de apoyar a la medicina basada en la evidencia. Por lo que, un SLR es un esfuerzo por recopilar y sintetizar evidencia científica sobre un tema, siguiendo un método sistemático que asegure minimizar el sesgo de dicha evidencia [64].

La comunidad de Ingeniería de Software ha adaptado la SLR, pues la ciencia médica ha generado bastos reportes de estudios empíricos rigurosos, los cuales son menores en la comunidad de Ingeniería de Software [64].

Una SLR se emplea para identificar, evaluar e interpretar investigaciones relevantes y tiene las siguientes fases: formulación de la pregunta de revisión, determinación de las

fuentes y el método de búsqueda, definición y aplicación de los criterios de selección, análisis crítico, obtención y síntesis de los resultados, y extracción de conclusiones e inferencias. De acuerdo a Kitchenham [64], en el contexto de las SLR, un estudio primario es todos los artículos de investigación originales que se encuentran durante la revisión, mientras que los estudios secundarios son aquellos que compilan los resultados de estas revisiones (*e.g.*, *surveys* y las propias SLR).

Por otro lado, un estudio de mapeo sistemático (*Systematic Mapping Study*, SMS) identifica, categoriza y analiza la literatura científica relevante para un tema de investigación específico [64, 65, 66]. Un SMS tiene como objetivo obtener una descripción general completa de un tema de investigación en particular, presentar una evaluación imparcial de la literatura actual, identificar brechas en la investigación y recopilar evidencia para futuras direcciones de investigación [64, 66, 67]. El resultado de un SMS es una muestra clasificada de publicaciones sobre el área de investigación, *i.e.*, un mapa de alto nivel que informa la estructura de los trabajos relacionados con el tema de interés, visualizando el estado de ese campo con respecto a las preguntas de investigación predefinidas y dando un resumen visual de varias categorías de clasificación [65, 66].

Según Kitchenham [64], existen diferencias importantes entre un SMS y una SLR:

- Un SMS plantean preguntas de investigación más amplias que una SLR.
- Los términos de búsqueda de un SMS arrojan una mayor cantidad de estudios.
- El proceso de extracción de datos de un SMS es más amplio.
- El análisis de un SMS resume los datos para responder a las preguntas de investigación propuestas.
- La difusión de los resultados de un SMS está más influenciada por las publicaciones académicas.

Estas diferencias muestran que recurrir a un SMS para iniciar el desarrollo de nuestra investigación es la mejor opción, pero para agregar aún más valor a ello, se utiliza la propuesta de Petersen *et al.* [65] que consta de cinco pasos:

1. Definir preguntas de investigación (alcance de la investigación).
2. Realizar búsquedas de estudios (todos los artículos).

3. Seleccionar trabajos de acuerdo a criterios de inclusión y exclusión (artículos relevantes).
4. Clasificación de estudios (esquema).
5. Extracción de datos y agregación (mapa).

En dicha propuesta, en la salida de cada paso del método, se obtiene un producto y finaliza con el mapeo sistemático como resultado final. En el Capítulo 3, se detalla la aplicación y el desarrollo del método SMS para chatbots educativos.

2.2. Clasificaciones de chatbots

Chen *et al.* [68] clasifican los chatbots en dos grupos principales: orientados a tareas y no orientados a tareas. Aquellos *orientados a tareas* ayudan al usuario a completar una labor, *e.g.*, buscar un producto o tener una conversación breve en un dominio cerrado. Los chatbots *no orientados a tareas* interactúan con el usuario siguiendo un enfoque de pregunta-respuesta con fines lúdicos, generalmente en un dominio abierto.

Nuruzzaman y Hussain [63] dividen las aplicaciones de chatbot en cuatro categorías: basadas en objetivos, basadas en conocimientos, basadas en servicios y basadas en respuestas generadas. Los chatbots *basados en objetivos* están diseñados para una tarea específica y una conversación corta, *e.g.*, enfoque de pregunta-respuesta o resolución de problemas para clientes en un sitio Web. Los chatbots *basados en conocimientos* están destinados a dominios abiertos y cerrados, *e.g.*, responder a temas generales y particulares, respectivamente. Los chatbots *basados en servicios* brindan facilidades a los clientes, *e.g.*, pedidos a tiendas de alimentos. Finalmente, los chatbots *basados en respuestas generadas* se enfocan en cómo responder a las preguntas de los usuarios, *i.e.*, se devuelve una respuesta priorizada que se elige según lo establecido en la política del modelo.

Gnewuch *et al.* [69] propone una clasificación bidimensional: modo primario de comunicación y contexto. La primera dimensión indica la modalidad de interacción con el chatbot, como texto o voz. La segunda dimensión se enfoca en un dominio específico o en un tema de conversación con sus usuarios.

Ramesh *et al.* [70] sugiere una clasificación de seis grupos: basado en recuperación, modelo generativo, conversación de texto corto, conversación de texto largo, dominio abierto

y dominio cerrado. Los chatbots *basados en recuperación* tienen un conjunto de respuestas predeterminadas y, para dar una respuesta adecuada, hacen uso de heurísticas. El *modelo generativo* evita depender de respuestas predefinidas para generar nuevas respuestas. La *conversación de texto corto* se refiere a una respuesta breve cuando el chatbot recibe una pregunta específica. La *conversación de texto largo* significa que el chatbot puede tener una conversación duradera. El *dominio abierto* se refiere a la posibilidad de cambiar entre diferentes dominios en una conversación. El *dominio cerrado* indica un conocimiento específico, por lo tanto, es deseable una respuesta adecuada.

Diederich *et al.* [71] presenta una taxonomía de once dimensiones para chatbots: modo de comunicación, contexto, lenguaje, inteligencia, implementación, alojamiento, modelo de precios, informes, detección de sentimientos, integración empresarial e integración de plataforma. El *modo de comunicación* y el *contexto* son extensiones de la clasificación de Gnewuch *et al.* [69]. La taxonomía *lenguaje* indica si el chatbot puede admitir uno o más idiomas. La *inteligencia* se refiere a si el chatbot se basa en reglas, como la coincidencia de patrones o las habilidades de autoaprendizaje. La *implementación* indica la tecnología utilizada para el desarrollo del chatbot. El *alojamiento* es la plataforma donde se implementa el chatbot. El *modelo de precios* indica el precio que se pagará por usar una plataforma, *e.g.*, Microsoft Azure Bot tiene un costo según la cantidad de interacciones y Dialogflow tiene una versión gratuita pero limitada. La taxonomía *informes* considera si la plataforma cuenta con un monitor para conocer el detalle de interacciones, usuarios y número de conversaciones. La *detección de sentimientos* indica si la plataforma soporta la detección de sentimientos de los usuarios en una conversación. Finalmente, la *integración empresarial* significa que la plataforma ofrece una API o interfaces preconstruidas.

Nimavat y Champaneria [72] clasifican los chatbots en cuatro grupos: dominio del conocimiento, servicio prestado, objetivos y procesamiento de entrada y método de generación de respuestas. El *dominio de conocimiento* se refiere a si el chatbot puede interactuar en un dominio abierto o cerrado. El *servicio prestado* se basa en una subclasificación proxémica: interpersonal, intrapersonal e interagente. Por su parte, el grupo *objetivos* implica una subclasificación basada en un objetivo principal: informativo, basado en chatbot conversacional y basado en tareas. Finalmente, el grupo *procesamiento de entradas y método de generación de respuestas* indica los procedimientos para manejar las entradas y generar las respuestas.

Hussain *et al.* [73] establecen los siguientes grupos para su clasificación: modo de in-

teracción (texto o voz), aplicación chatbot (orientado a tareas o no orientado a tareas), basado en reglas o Inteligencia Artificial (Aprendizaje Máquina, Aprendizaje Profundo o plantillas) y dominio (específico o abierto).

Adamopoulou y Moussiades [74] proponen siete categorías de chatbots. Las primeras cuatro categorías son las mismas presentadas por Nimavat y Champaneria [72]. La categoría *ayuda humana* se refiere a la necesidad de flexibilidad, por lo que las operaciones son realizadas por el chatbot con intervención humana. La categoría *permisos* considera si el chatbot es de código abierto o comercial. Finalmente, la categoría *canal de comunicación* depende de la modalidad de interacción (texto, voz o imagen).

Quiroga Pérez *et al.* [75] mencionan una clasificación de chatbots educativos con dos categorías: orientados al servicio y orientados a la enseñanza. El primero se centra en el soporte de servicios, como las FAQ. El segundo se divide en formal e informal.

Como se puede apreciar, hay relativamente pocas clasificaciones de chatbots y la mayoría de los trabajos se presentan como revisiones de la literatura científica [63, 68, 70, 73, 74, 75]. Ésta es una limitación porque dichas clasificaciones no presentan en detalle el proceso realizado para obtenerlas.

Por otro lado, algunos autores, como Diederich *et al.* [71], Feine *et al.* [76], Janssen *et al.* [77, 78] y Bittner *et al.* [79], utilizan el método propuesto por Nickerson *et al.* [80] para el desarrollo de su clasificación. Sin embargo, éstas no son clasificaciones de chatbots educativos. Es importante mencionar que las clasificaciones de chatbots educativos son casi inexistentes, a excepción del trabajo de Quiroga Pérez *et al.* [75], que clasifica a los chatbots educativos en dos categorías, como se mencionó anteriormente. Sin embargo, no dan detalles de su proceso de clasificación.

En las siguientes subsecciones, se describe el método utilizado en este trabajo de tesis para el desarrollo de una clasificación de chatbots educativos.

2.2.1. Elementos teóricos para desarrollar clasificaciones

En la terminología de Gregor [81] y Nickerson *et al.* [80], los conceptos de tipología, marco de trabajo, taxonomía y clasificación son intercambiables. Para nuestros propósitos, en este trabajo utilizamos el término clasificación.

Una clasificación de objetos es un mecanismo básico para organizar el conocimiento [82]. Algunos beneficios son analizar y comprender dominios complejos [80], proporcionar es-

estructura y organización del conocimiento en un campo [83], implementar un orden que permita establecer relaciones entre conceptos [84] y comprender las diferencias entre trabajos de investigación previos [85].

El estudio y desarrollo de los chatbots es un tema de gran importancia y popularidad en la actualidad y parece poco probable que esta tendencia vaya a cambiar en los próximos años. Por esta razón, es fundamental identificar las características y clasificaciones de las propuestas en el área, especialmente en el ámbito educativo, ya que es un campo fértil en el que la Inteligencia Artificial y la Interacción Humano-Computadora pueden tener un impacto significativo. Sin embargo, para ello es necesario contar con metodologías sólidas que permitan reunir una muestra representativa de las propuestas en cuestión. El método de Nickerson *et al.* [80] y el SMS [65] son los medios que pueden ayudar a lograrla en el presente trabajo de tesis.

2.2.2. Método de clasificación iterativo

En esta sección, se presenta, de manera general, el método de Nickerson *et al.* [80], destacando elementos teóricos importantes.

El método utiliza los términos de dimensiones y características. Las dimensiones pueden verse como variables y las características como instancias de dichas variables. La Figura 2.1 presenta los siete pasos del método de Nickerson *et al.* [80]:

(1) identificar la meta-característica, que se refiere al propósito de la clasificación;

(2) determinar las condiciones finales, que sirven como guía para detener el proceso de clasificación;

(3) seleccionar un enfoque: a) *empírico a conceptual* (se examinan nuevos objetos para determinar si las características son suficientes o si nuevas características y posibles dimensiones son necesarias) o b) *conceptual a empírico* (comienza por conceptualizar dimensiones sin examinar objetos reales); dependiendo de los pasos del enfoque seleccionado los pasos 4-6, pueden cambiar:

(4a) identificar subconjuntos de objetos reconociendo las características de una mues-

tra sistemática;

(5a) identificar características comunes y agrupar objetos;

(6a) agrupar características en dimensiones para crear una taxonomía;

o bien

(4b) conceptualizar las características y dimensiones de los objetos;

(5b) examinar objetos para estas características y dimensiones;

(6b) crear una taxonomía; y

(7) preguntar si se han cumplido las condiciones finales.

Hay ocho posibles condiciones finales objetivas:

1. Se han examinado todos los objetos o, al menos, una muestra representativa.
2. Ningún objeto se fusionó con un objeto similar ni se dividió en varios objetos en la última iteración.
3. Al menos un objeto corresponde a cada característica de cada dimensión.
4. No se agregaron nuevas características o dimensiones en la última iteración.
5. No se fusionaron ni dividieron características ni dimensiones en la última iteración.
6. Cada dimensión es única.
7. Cada característica es única dentro de su dimensión.
8. Cada celda (combinación de características) también es única.

En cuanto a las condiciones finales subjetivas, hay cinco posibilidades:

1. *Conciso*: indica el número de dimensiones que permiten establecer la clasificación.

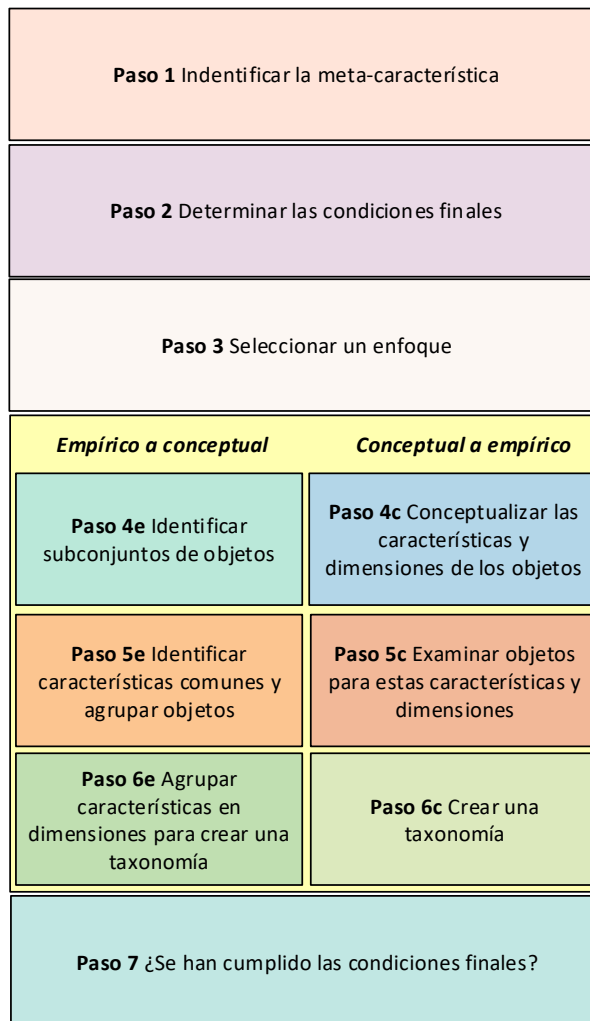


Figura 2.1: Método de siete pasos de Nickerson.

2. *Robusto*: significa que las características y dimensiones identificadas son suficientes para diferenciar los objetos de interés.
3. *Comprehensivo*: se refiere a que todos los objetos analizados pueden ser clasificados dentro del dominio bajo consideración o la clasificación incluye todas las dimensiones de los objetos de interés.
4. *Extendible*: tiene que ver con la posibilidad de añadir nuevas características y nuevas dimensiones.
5. *Explicativo*: proporciona una explicación útil de los objetos de estudio.

Cabe mencionar que el segundo paso está determinado por las condiciones objetivas y subjetivas de finalización; una vez que están satisfechas, el proceso de iteración se detiene. Para llegar al final del método, es necesario verificar que se cumplan las condiciones; de lo contrario, se debe iniciar una nueva iteración desde el tercer paso. El método es flexible, ya que no se deben cumplir todas las condiciones finales objetivas y subjetivas, *i.e.*, es posible elegir entre esas condiciones.

En el Capítulo 4 se explica el desarrollo de este método, paso a paso, para obtener una clasificación de chatbots enfocados a la educación.

2.3. Arquitectura de software

En un sistema de software, la arquitectura es el conjunto de estructuras necesarias para razonar sobre el sistema, éstas comprenden los elementos de software, las relaciones entre ellos y las propiedades de ambos [86].

La arquitectura de software es un aspecto importante en el desarrollo de un sistema, que empieza a realizarse justo antes de iniciar la etapa de diseño. Así, la importancia se encuentra en el hecho de que la arquitectura es una guía para el desarrollo del sistema [87].

Entre los beneficios que se obtienen al crear una arquitectura de software, se encuentran los siguientes: mejor comunicación entre los involucrados, decisiones tempranas de diseño y abstracción transferible de un sistema a otro [86].

Por un lado, debe tomarse en cuenta que el diseño de la arquitectura es una práctica relevante en el desarrollo de software. Tan es así que, para diseñar la arquitectura, se han documentado a lo largo de los años técnicas, métodos, marcos de trabajo y procesos. Por otro lado, el diseño de una arquitectura genérica es una tarea compleja porque cada desarrollo tiene su propósito, requerimientos, restricciones y contexto. Además, los componentes genéricos que se definan deben tener mecanismos significativos de diseño, nodos y sus respectivas configuraciones, subsistemas e interfaces, entre otras cosas. Sin embargo, un aspecto de suma importancia es que «genérico» en el contexto de la Ingeniería de Software puede verse como «reutilizable».

2.3.1. Arquitecturas de chatbots

Los chatbots no son ninguna excepción con respecto a la arquitectura, por lo tanto, deben poseer una. Un número significativo de chatbots han sido implementados siguiendo el modelo cliente-servidor. No obstante, no es la única forma porque existen diferentes arquitecturas de referencia para el desarrollo de este tipo de sistemas [88], como la de componentes, modular, capas, microservicios y modelo vista controlador (MVC), las cuales se describen brevemente a continuación:

- La arquitectura basada en componentes, también conocida como ingeniería de software basada en componentes, es la separación del diseño en funcionalidad o componentes lógicos [89].
- La arquitectura modular es una descomposición del sistema en un subconjunto de elementos [90].
- La arquitectura en capas es la representación abstracta de un sistema de software por niveles, dividiendo al sistema en capas de servicios y funcionalidades [90].
- La arquitectura de microservicios se utiliza para el desarrollo de aplicaciones distribuidas, de manera que una aplicación se basa en componentes más pequeños, los cuales son aplicaciones en sí mismos [91].
- Finalmente, MVC es un patrón de arquitectura de software que divide un sistema en tres componentes interconectados: modelo (representación de información), vista (usualmente interfaz de usuario) y controlador (interfaz entre modelo y vista) [92].

Dado que nuestro objetivo es desarrollar una arquitectura genérica de chatbots educativos, a continuación se presentan diferentes propuestas de arquitecturas genéricas de chatbots.

Inokuchi *et al.* [8] sugieren que un chatbot debe tener dos elementos principales: *servicios de chat* y *bot*. En esta arquitectura, la interacción es de la siguiente manera: un usuario manda un comando como mensaje desde el *servicio de chat* y el *bot* recibe el mensaje y realiza las operaciones indicadas en el mensaje. Finalmente, el *bot* obtiene el resultado y lo manda al usuario.

Veglis y Maniou [9] se decantan por una arquitectura cliente-servidor para el desarrollo de chatbots. El cliente puede ser una página web o aplicación móvil. Del lado del servidor,

se cuenta con los siguientes componentes: 1) el *Analizador de Lógica y Reglas* realiza el reconocimiento del texto de entrada para así obtener una respuesta basada en reglas predefinidas; 2) la *Base de Conocimientos* es la que dota de inteligencia al chatbot; y finalmente, 3) el *Módulo de Procesamiento de Lenguaje Natural* valida el texto de entrada realizando un análisis de opinión e instruye a la *Base de Conocimientos* para que responda de acuerdo con el resultado.

Matthies *et al.* [10] utilizan la propuesta de Inokuchi *et al.* [8] y agregan un elemento extra que consiste en obtener información adicional desde recursos de terceros, *e.g.*, base de conocimientos o recursos de Internet.

Khan [93] analiza las tecnologías y los servicios emergentes para el desarrollo de chatbots, y propone una arquitectura estándar para implementar soluciones de chatbots en general. El autor propone una arquitectura de seis capas: 1) la *Capa de Presentación* contiene componentes que implementan y muestran la interfaz de usuario y gestiona la interacción del usuario; 2) la *Capa de Negocio* tiene componentes que se encargan de procesar, formatear y gestionar los datos; 3) la *Capa de Servicio* proporciona componentes que permiten el acceso a los datos internos y externos, funcionalidad del negocio, conectividad con la lógica de intercambio de información entre aplicaciones y otros servicios; 4) la *Capa de Datos* se encarga de proporcionar un acceso eficiente y seguro a los datos; 5) la *Capa de Utilidad* maneja diferentes parámetros del sistema, como seguridad y configuración; y 6) la *Capa de Servicios Externos* utiliza diferentes servicios externos, dependiendo del tipo de solución del chatbot.

Srivastava y Prabhakar [94] proponen una arquitectura de referencia para chatbots. Esta arquitectura consiste en cinco componentes: 1) las *Utilidades de voz* se encargan de convertir el mensaje de voz del usuario a su representación en texto y viceversa para entregar el mensaje al usuario; 2) los *Intentos* son diferentes clases de entrada que la aplicación espera recibir y los *parámetros* son atributos o detalles de una consulta, que se deben analizar a partir de las entradas para producir una respuesta o realizar una acción; 3) el *Generador de respuestas* tiene la función de generar respuestas apropiadas para el usuario después de cada entrada; 4) el *Gestor de flujo* tiene la tarea de realizar un seguimiento del estado actual de la conversación y decidir cuál debe ser el siguiente paso; y finalmente, 5) las *Acciones y Realizaciones* proporcionan el puente entre las funcionalidades principales de la aplicación y los componentes del chatbot.

A manera de conclusión, se puede decir que la propuesta de Inokuchi *et al.* [8] describen,

de manera sencilla, la interacción del chatbot sin profundizar en ningún tipo de detalle. Veglis y Maniou [9] mencionan aspectos importantes como la arquitectura, el acceso al chatbot, los analizadores, la base de conocimientos y el procesamiento de lenguaje natural. Matthies *et al.* [10] utilizan el trabajo previo de Inokuchi *et al.* [8] y lo mejoran añadiendo recursos de terceros. Khan [93] utiliza una arquitectura de capas, describiendo detalles técnicos importantes y cómo hacer la instancia del modelo. Srivastava y Prabhakar [94] precisan aspectos técnicos significativos de un chatbot con servicios emergentes como los de la nube. Finalmente, cabe mencionar que los aportes de estos autores nos proporcionan elementos claves y fundamentales que serán de mucha ayuda para definir una arquitectura genérica para chatbots educativos.

2.3.2. Métodos para el diseño de arquitecturas

Kruchten [95] es conocido por haber elaborado el Proceso Unificado de Rational (*Rational Unified Process*, RUP). En este proceso se propone un desarrollo iterativo por medio de cuatro fases: 1) inicio, 2) elaboración, 3) construcción y 4) transición. El desarrollo de una arquitectura comienza con actividades en las primeras dos fases, las cuales guían el diseño arquitectónico. RUP es una opción adecuada para grandes proyectos. Para proyectos pequeños se recomienda utilizar OpenUP, pero éste carece de todos los elementos que se manejan en un proyecto.

Lattanze [96] presenta el Método de Desarrollo Centrado en la Arquitectura (*Architecture-Centric Design Method*, ACDM) que consiste de siete etapas, con sus respectivos pasos y actividades, para el desarrollo de una arquitectura de software. Las etapas son: 1) descubrimiento de los controladores arquitectónicos, 2) establecimiento del alcance del proyecto, 3) creación de la arquitectura nocional, 4) revisión de la arquitectura, 5) decisión de seguir o no seguir con la producción, 6) planificación de experimentos o planificación de la producción y 7) ejecución del experimento y refinamiento de la arquitectura o producción. La tercera etapa es la que da inicio al diseño de la arquitectura.

Hofmeister *et al.* [97] comparan y analizan diferentes métodos para el diseño de una arquitectura de software, tales como: Diseño Guiado por Atributos [98] (*Attribute-Driven Design*, ADD), Siemens 4 Vistas [99], RUP 4+1 Vistas [95, 100], Proceso y Organización de la Arquitectura Empresarial [101, 102] y Separación Arquitectónica de Preocupaciones [103]. Una vez finalizado el análisis, se extraen puntos en común para proponer un

Modelo General de Diseño de Arquitectura de Software (GMSAD), el cual hace énfasis en las siguientes actividades: análisis, síntesis y evaluación arquitectónica.

Eeles y Cripps [104] proponen «El Proceso de Arquitectura de Software» (EPAS), el cual consiste en tres actividades principales enfocadas a un diseño arquitectural: definición de requerimientos, creación de la arquitectura lógica y creación de la arquitectura física. Estas actividades van acompañadas con una serie de tareas que deben llevarse a cabo por roles definidos previamente. Por medio de ejemplos detallados, muestran de manera comprensible cómo realizar las tareas del proceso.

Microsoft [105] propone una técnica denominada «Guía de Arquitectura de Aplicaciones» (GAA), que consiste de cinco pasos iterativos para esbozar una arquitectura de software: 1) identificar los objetivos de la arquitectura, 2) identificar los escenarios clave, 3) crear una descripción general de la aplicación, 4) identificar problemas clave y 5) definir soluciones candidatas. Microsoft hace énfasis en mostrar cómo se usa su tecnología con esta técnica, independientemente del proceso de desarrollo que se siga.

Rozanski y Woods [106] proponen un método con dos conceptos importantes: «Puntos de Vistas y Perspectivas Arquitectónicas» (PPA). Los autores inician definiendo las *vistas* como una representación de uno o más aspectos estructurales de una arquitectura, que ilustra cómo aborda una o más preocupaciones de uno o más interesados. Entonces, los *puntos de vista* son colecciones de patrones, plantillas y convenciones para construir un tipo de vista. Por su parte, las *perspectivas arquitectónicas* son colecciones de actividades, tácticas y directrices que se utilizan para garantizar que un sistema exhiba un conjunto de propiedades de calidad, que deben tenerse en cuenta en las vistas arquitectónicas del sistema. Los pasos que proponen para el diseño arquitectónico son: consolidar las entradas, identificar escenarios, identificar estilos arquitectónicos relevantes, producir una primera arquitectura, explorar opciones arquitectónicas, evaluar la arquitectura con los involucrados y rehacer la arquitectura o revisar los requerimientos.

Cervantes y Kazman [107] retoman el método conocido en la comunidad de la arquitectura de software como ADD. Los autores analizan el método y proceden a hacer una serie de mejoras. Al finalizar, renombran al método como ADD 3.0, el cual consta de siete pasos: 1) revisar entradas, 2) establecer la meta de la iteración seleccionando las directrices, 3) elegir uno o más elementos del sistema a refinar, 4) elegir uno o más conceptos de diseño que satisfagan las directrices seleccionadas, 5) crear instancias de elementos arquitectónicos, asignar responsabilidades y definir interfaces, 6) crear vistas de bosquejos

y tomar decisiones de diseño de registros y finalmente 7) realizar un análisis del diseño actual y revisar el objetivo de la iteración, así como el logro del diseño.

Las propuestas mencionadas anteriormente presentan interesantes enfoques. Cada una exhibe sus ventajas y desventajas, y la elección entre ellas no es una tarea que se pueda tomar a la ligera. Sin embargo, para nosotros, ciertos aspectos revisten especial importancia. Por ejemplo, buscamos una propuesta que se centre fuertemente en el diseño arquitectónico, evitando desviarse hacia otras actividades o tareas. Además, el nivel de detalle de la propuesta también es fundamental; un alto grado de abstracción puede llevar a la omisión de elementos relevantes para el diseño. En este sentido, hemos observado que las propuestas iterativas suelen tener ventajas, ya que definen claramente el ciclo de vida del proyecto, un elemento que consideramos esencial. La documentación es otro aspecto importante a considerar, cómo se documenta el diseño y qué tan efectiva es esta documentación. Por último, la presencia de pasos definidos que guíen de manera clara y específica es un criterio importante para nosotros.

En su libro Cervantes y Kazman [107] hacen un análisis bastante minucioso de las propuestas mencionadas anteriormente. En esa comparación, destaca ADD 3.0 sobre las demás, por lo tanto, es la que mejor se podría ajustar a nuestras necesidades en términos de diseño, explicación, aplicación, flexibilidad y adaptabilidad. Sin embargo, un punto débil que podemos apreciar en ADD 3.0, reconocida por los propios autores, radica con respecto a la documentación. Si bien, saber cómo documentar de manera eficiente el diseño, lo cuál es importante para nosotros, los mismos autores ofrecen sugerencias y mecanismos que se pueden integrar bien para fortalecer esa carencia.

Finalmente, la propuesta de ADD 3.0 no solo describe el método en detalle, sino que también proporciona una amplia variedad de ejemplos que ilustran su aplicación. Además, presenta un nutrido catálogo que abarca desde arquitecturas de referencia, patrones de despliegue, patrones de diseño arquitectónico, tácticas, hasta estrategias.

Capítulo 3

Desarrollo del SMS

En este capítulo, se muestra la aplicación y el desarrollo del método SMS, descrito en el Capítulo 2. Primeramente, se definen las preguntas de investigación que sirven de guía para el desarrollo del método (cf. Sección 3.1). Después, se establece la estrategia de búsqueda para conseguir la evidencia (cf. Sección 3.2). A continuación, se crea la expresión de búsqueda para obtener los estudios primarios (cf. Sección 3.3). Posteriormente, se fijan los criterios para la selección de los estudios (cf. Sección 3.4). Luego, se presentan las clasificaciones de los estudios por pregunta de investigación, año y tipo de publicación (cf. Sección 3.5) y se extrae la información relevante de los estudios seleccionados (cf. Sección 3.6). Finalmente, se describen los resultados obtenidos para dar respuesta a las preguntas de investigación (cf. Section 3.7).

3.1. Preguntas de investigación

Las preguntas de investigación (PI) se especifican para refinar el alcance, siguiendo los objetivos del proyecto. El Cuadro 3.1 muestra las preguntas que hemos establecido y sus respectivos objetivos.

Para la construcción de los términos de búsqueda, se toma como guía la propuesta de PICOC (Población, Intervención, Comparación, Resultado y Contexto) [108]. Se omite el elemento de comparación porque este elemento se utiliza más cuando se trata de revisión de literatura en el campo de la medicina, ya que existe la necesidad de comparar entre dos o más tratamientos, por lo tanto, se tienen cuatro elementos en la estructura de las preguntas de investigación, como se muestra en Cuadro 3.2.

Cuadro 3.1: Preguntas de investigación.

ID	Pregunta	Objetivo de la pregunta
PI-1	¿Qué propósito tiene un chatbot?	Conocer las características genéricas o diferenciales de los diversos conceptos, definiciones o significados que se han dado a los chatbots y, con base en estos, proporcionar una definición de chatbot educativo.
PI-2	¿En qué niveles educativos se utilizan los chatbots y cómo es el mecanismo de interacción?	Identificar las diferentes funcionalidades que ofrecen los chatbots en los distintos niveles escolares, así como los mecanismos de interacción para comunicarse con los estudiantes o la comunidad.
PI-3	¿Qué tecnología y arquitectura predominan en la construcción de chatbots educativos?	Identificar las diferentes herramientas que se utilizan en la construcción de este tipo de chatbots, así como las arquitecturas que se emplean para satisfacer las necesidades planteadas.

Cuadro 3.2: Elementos en la estructura de las preguntas de investigación.

ID	Población	Intervención	Resultado	Contexto
PI-1	Chatbot	Artificial Intelligence, HCI	Chatbot concepts, Chatbot definition, Chatbot meaning	Any context
PI-2	Chatbot	Education, HCI	Schools, Teachers, Students, Classes	Any context
PI-3	Chatbot	Education, HCI	Develop, Build, Method, Algorithms, Design, Tool, Architecture	Any context

3.2. Estrategia de búsqueda

Para la obtención de artículos, se utilizan diferentes editoriales digitales. Como es del conocimiento de la comunidad científica, estas editoriales proporcionan motores de búsqueda, pero cada uno tiene su forma de funcionar y su propia sintaxis; también puede haber opciones, operadores o comandos que no estén definidos en otros motores. Ante esta situación, si se define alguna expresión de búsqueda en un motor específico, probablemente se deba adaptar o modificar para otro motor. Dada la naturaleza de las búsquedas, la información proporcionada en los Cuadros 3.2, 3.3 y 3.4 se maneja en inglés.

En nuestro caso, se hace uso de los siguientes buscadores: *ACM Digital Library*, *IEEE Xplore*, *Science Direct*, *Scopus* y *Springer Link*.

3.3. Elaboración de la expresión de búsqueda

El Cuadro 3.3 presenta los términos, sinónimos y palabras alternativas para construir la expresión de búsqueda de las preguntas de investigación. Para lograr esto, se hace uso de la conjunción y disyunción con sus respectivos operadores booleanos OR y AND .

Cuadro 3.3: Términos y sinónimos para las expresiones de búsqueda.

Términos	Sinónimos/Palabras Alternativas
chatbot	bot, chatterbot, conversational agents
education	university, universities institute, institution, college, academy, high school, teach, teacher, teaching, student, class
tool	architecture, algorithm, algorithms, build, develop, development, developer, design, designs, designer, method

Finalmente el Cuadro 3.4 muestra la expresión que será introducida en los motores de búsqueda.

Cuadro 3.4: Cadenas de búsqueda para las preguntas de investigación.

ID	Expresión de búsqueda
PI	(‘‘artificial intelligence’’ OR education OR hci) AND (chatbot OR bot OR chatterbot OR ‘‘conversational agents’’) AND (universit* OR institut* OR college OR academy OR ‘‘high school’’ OR teach* OR student OR class) AND (tool OR architecture OR algorith* OR build OR develop* OR design* OR method)

3.4. Directrices para la selección de estudios

Para la selección de estudios, se establecen criterios de inclusión, exclusión y selección que permitirán obtener mejores resultados.

El Cuadro 3.5 presenta los criterios de inclusión (CI). La columna **ID** muestra el identificador para el criterio y la columna **Criterios** presenta la valoración para incluir un estudio. Es importante mencionar que cada estudio debe cumplir con los criterios de inclusión CI-1 y CI-2. Además, dicho estudio debe satisfacer, al menos, uno de los criterios de inclusión CI-3, CI-4 y CI-5.

Cuadro 3.5: Criterios generales de inclusión.

ID	Criterios
CI-1	El estudio fue publicado a partir del 2006 en adelante.
CI-2	El estudio contiene términos relevantes en el título, resumen o palabras clave.
CI-3	El estudio tiene definiciones de chatbot o similares.
CI-4	El estudio aborda aspectos de chatbots enfocados a la educación.
CI-5	El estudio describe una arquitectura, un método, herramientas y/o un diseño para el desarrollo de chatbots.

El Cuadro 3.6 muestra los criterios de exclusión (CE). En la columna **Acción**, se tiene la operación a realizar sobre el estudio. Es importante mencionar que no todos los criterios de exclusión deben cumplirse, basta con que se satisfaga uno.

Cuadro 3.6: Criterios generales de exclusión.

ID	Criterios	Acción
CE-1	El estudio está duplicado	Eliminar
CE-2	El estudio no hace referencia a la pregunta de investigación.	Eliminar
CE-3	El estudio no es un artículo de revista, de conferencia o de <i>workshop</i> .	Eliminar
CE-4	El estudio no está escrito en inglés o español.	Eliminar
CE-5	El estudio es una revisión de la literatura científica o estudio sistemático.	Eliminar

El Cuadro 3.7 presenta los criterios de selección (CS). La columna **Criterios** muestra el mecanismo definido para seleccionar los estudios.

Cuadro 3.7: Criterios generales de selección.

ID	Criterios
CS-1	Aplicar criterios de inclusión y exclusión en los documentos en el título, resumen y palabras clave.
CS-2	En los trabajos restantes, revisar figuras, metodología y conclusiones para buscar propiedades que se requieran y que no fueron identificadas en el resumen.
CS-3	Obtener una lista refinada de artículos y proceder a leerlos tomando en cuenta los criterios de inclusión y exclusión, pero ahora sobre su contenido.

3.5. Clasificación de estudios

Los trabajos seleccionados se clasifican de acuerdo con los siguientes criterios:

1. Clasificación por pregunta de investigación para conocer cuáles estudios son los que más se producen.
2. Clasificación por año para ver la tendencia de los trabajos publicados a medida que pasan los años.
3. Clasificación por tipo de estudio, *e.g.*, artículo de revista, de conferencia o de *workshop*.

3.6. Extracción de Datos

Para llevar a cabo la extracción de los datos relevantes para la investigación, se revisó el contenido de los estudios seleccionados. La principal función de esta revisión es identificar aquellas propiedades que permitan responder las preguntas de investigación que guían este estudio. En el Cuadro 3.8, se presenta la columna **Propiedades**, que son las características extraídas. La columna **PI** muestra el identificador con la pregunta de investigación que está relacionada.

Cuadro 3.8: Propiedades a extraer.

ID	Propiedades	PI
PE-1	Definición, concepto o significado de un chatbot.	PI-1
PE-2	Tipos de chatbots en el sector educativo y mecanismo de interacción.	PI-2
PE-3	Arquitectura, diseño, métodos y herramientas para construir un chatbot educativo.	PI-3

3.7. Respuestas a las preguntas de investigación

En la Figura 3.1 se muestra el proceso de ejecución del SMS. El método da inicio con las preguntas de investigación PI-1, PI-2 y PI-3. A continuación, se utiliza la expresión de búsqueda (ver Cuadro 3.4) y se introduce a los motores. Una vez que se obtienen los resultados, empieza el proceso de aplicar los criterios de selección, comenzando con el CS-1 hasta terminar con el CS-3. De esta manera, en el último paso se tiene el filtro de los estudios que fueron revisados para la clasificación y la extracción de la información.

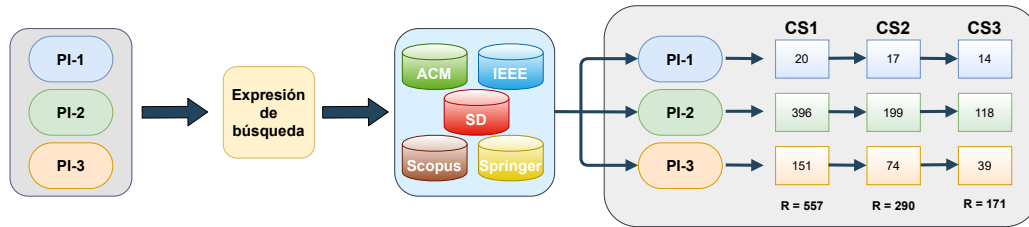


Figura 3.1: Ejecución del SMS.

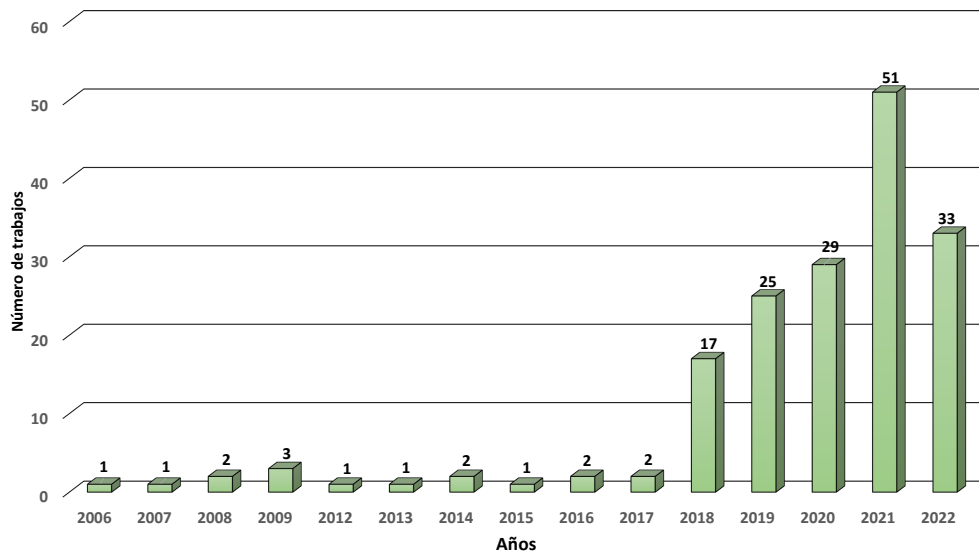


Figura 3.2: Estadísticas de propuestas por año.

Los resultados con respecto a la clasificación por pregunta de investigación (PI-1, PI-2 y PI-3) de los 140 trabajos que se obtuvieron son como sigue: el 9% corresponde a PI-1, el 67% tiene que ver con PI-2 y el 24% se relaciona con PI-3.

Los resultados, de acuerdo a la clasificación por año, revelan qué tanto interés se ha venido dando por los chatbots educativos. La Figura 3.2 muestra que de 2006 a 2017 los estudios fueron escasos. Hasta 2018 se empezó a investigar este tipo de aplicaciones de forma extensiva. También se puede observar un incremento de trabajos sobre chatbots educativos, lo que muestra una tendencia similar a lo que se mencionó en *Business Insider* [12].

La última clasificación se muestra en la Figura 3.3, donde se aprecia que la mayoría de los trabajos que fueron analizados se publicaron como artículos de congresos o *workshops*.

Esta tendencia se repite en la mayoría de los buscadores excepto, en *Scopus* y *Science Direct* (SD), donde hubo más resultados de artículos de revistas.

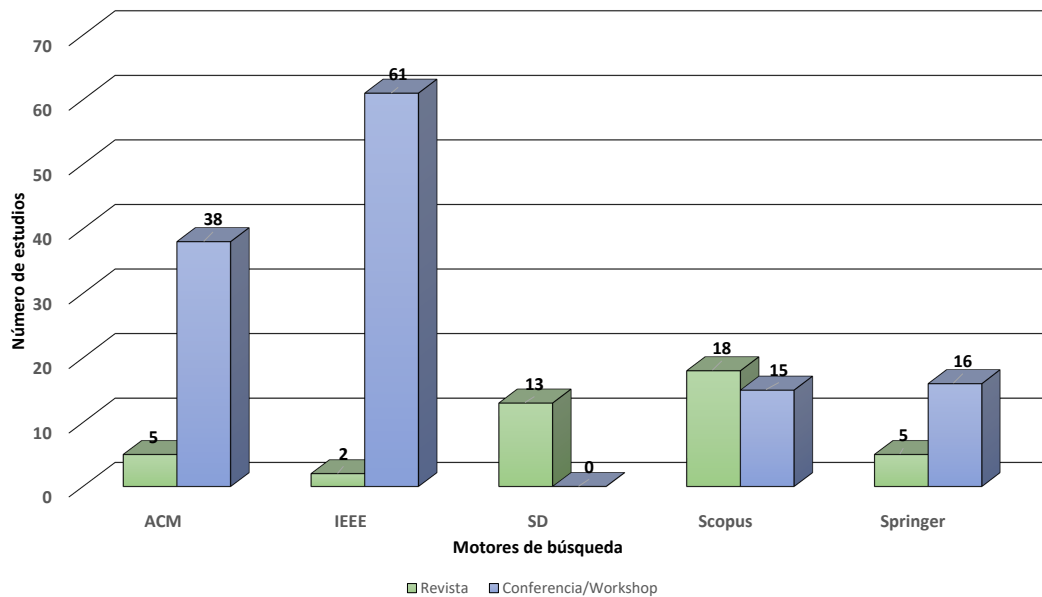


Figura 3.3: Estadísticas de los tipos de estudios seleccionados por buscador.

A continuación, se muestran los resultados obtenidos para las preguntas de investigación PI-1, PI-2 y PI-3.

3.7.1. Pregunta de investigación PI-1

Las respuestas que se obtuvieron para la pregunta de investigación PI-1 «¿Qué propósito tiene un chatbot?» son las siguientes:

- Un chatbot es un programa impulsado por las reglas y habilidades intelectuales de un humano, que se enfoca en estrategias de conversación; los diseñadores de chatbots utilizan, en gran medida, aprendizaje automático y técnicas de Inteligencia Artificial [109].
- Un chatbot simula una conversación verbal similar a la humana en una página Web, lo que permite que un usuario escriba preguntas u oraciones completas en un campo

de entrada de texto y, a cambio, genera una respuesta que se imprime en la pantalla o habla por medio de una síntesis de voz [15].

- Los chatbots o bots conversacionales son software capaces de responder mensajes de texto y de voz [110].
- Las aplicaciones de chatbot son agentes conversacionales automatizados capaces de interactuar con los usuarios de la plataforma [111, 112].
- Los términos «chatbot», «chatterbot» y «bot» se usaron indistintamente para describir la realización de esta visión desde el principio. Pero ahora, se refieren principalmente a una interfaz de usuario de estilo conversacional, un esquema antropomorfizado o un agente que automatiza tareas rutinarias y tediosas [113].
- Un chatbot es un sistema que puede interactuar con humanos y responde preguntas sobre un determinado dominio [114].
- La palabra «chatbots» es un derivado de los robots de chat, entendidos como máquinas que sirven como interfaces de usuario de lenguaje natural para datos y servicios a través de texto o voz [115, 116].
- Un chatbot es un dispositivo destinado a realizar interacciones sociales como lo haría un humano [13, 117].
- Los chatbots son programas desarrollados para entablar conversaciones con humanos [43].
- Un chatbot es un programa de conversación, que puede dar la sensación de estar hablando con personas y se usa activamente en áreas como el servicio al cliente y el servicio de entretenimiento [118].
- Un chatbot es una máquina diseñada para simular y reproducir una conversación inteligente con los usuarios [119, 120].
- Un agente conversacional consiste en un programa de software que interactúa con un usuario, interpretando y respondiendo a las declaraciones del usuario realizadas en lenguaje natural [76, 2, 36, 121].

- Un chatbot es un programa de inteligencia artificial y un modelo de interacción humano-computadora [122, 74].
- Un chatbot es un sistema en línea de diálogo humano-computadora con lenguaje natural [123, 124].
- Un agente conversacional permiten a los seres humanos comunicarse en lenguaje natural para interactuar directamente con agentes informáticos, como asistentes virtuales, agentes de atención al cliente o chatbots individuales-sociales [125].

Como se puede observar existen diferentes definiciones de chatbots, con ligeras variaciones. Nuestro interés principal es en chatbots enfocados a la educación, así que consideramos importante la propuesta de una definición formal:

- *Un chatbot educativo es un software de dominio cerrado que interactúa en tiempo real con estudiantes y profesores usando lenguaje natural con la finalidad de apoyar el proceso de enseñanza-aprendizaje y asistir a la comunidad escolar en diversos temas de interés común.*

Esta definición conlleva, de manera implícita, a que un chatbot educativo puede tener las siguientes funciones:

- Dar información general y específica de temas de interés común en la comunidad, *i.e.*, dominio cerrado.
- Proporcionar indicaciones o procedimientos para algún trámite general o concreto.
- Responder FAQ.
- Realizar evaluaciones y dar retroalimentación a los estudiantes, de tal manera que puedan apoyar en la formación y el progreso de los estudiantes en el periodo de estancia en la institución educativa.
- Dar seguimiento y acompañamiento entre el estudiante y el profesor.
- Responder preguntas específicas de temas en alguna asignatura en curso.
- Proporcionar algún tipo de canalización para temas de salud física y mental.

3.7.2. Pregunta de investigación PI-2

Antes de dar respuesta a la pregunta de investigación PI-2 «¿En qué niveles educativos se utilizan los chatbots y cómo es el mecanismo de interacción?» cabe mencionar lo siguiente. A medida que se analizaban los estudios, se notaron tendencias de chatbots educativos con respecto a su funcionalidad. Para profundizar en ese aspecto, se optó por trabajar, de manera paralela, con un método de clasificación iterativo, el método de Nickerson *et al.* [80] (descrito en la Sección 2.2.2), para empatar ambos resultados. Por simplicidad, a continuación se presentan, de manera somera, los resultados de la pregunta de investigación PI-2 y en el Capítulo 4 se detallará la respuesta a esta pregunta de investigación.

Se identificó que los chatbots se utilizan principalmente desde los niveles medio superior y superior, pero las tres categorías de chatbots educativos observadas en los estudios son:

- **Orientados a servicios escolares.** Son chatbots de propósito general para brindar información sobre trámites administrativos o FAQ. Son útiles tanto para las comunidades internas como externas de una escuela. Aquí se pueden mencionar los siguientes trabajos: CiSA [126], DINA [22], Daswani *et al.* [127], E-orientation [128], EASElective [129], FIT-EBot [47], KEMTbot [130], Khin y Soe [131], Lee *et al.* [132], LiSA [48], Mekni *et al.* [133], Meshram *et al.* [134], Nguyen *et al.* [135], SIAAA-C [136], Sai Vikas *et al.* [137], Sebastian y Nugraha [138], TutorDocente [139], UMT-BOT [46], Universitybot [131], S.A.N.D.R.A. [140] y Coronabot *et al.* [141].
- **Orientados a educación en línea.** Son aquellos chatbots diseñados para acoplarse en sistemas de aprendizaje en línea mucho más complejos (*e.g.*, Moodle) y aplicaciones independientes que, a menudo, se usan fuera de un entorno académico formal (*e.g.*, aprender un nuevo idioma). En esta categoría se incluyen los siguientes trabajos: Chatbot [142], Clarizia *et al.* [143], Damasceno *et al.* [144], Du *et al.* [145], EduBot [146], English Master AMMU [147], English Practice [148], EnglishBot [149], Gaglo *et al.* [150], Goschlberger y Brandstetter [151], Han y Lee [152], INES [153], Jeon [154], MALL [155], Muhammad *et al.* [156], Nguyen *et al.* [157], Oliveira *et al.* [158], Pereira *et al.* [159], Phaokla y Netinant [160], QLearn [161], Sarosa *et al.* [162], Song *et al.* [163], STUART [144], Blingbang [164], TPBOT [165], VirtualOrtografic [166] y Ye *et al.* [167].
- **Orientados a estudiante/profesor.** Son chatbots que comprenden aplicaciones

utilizadas en el día a día de las aulas. Estas herramientas permiten practicar ejercicios sobre los temas tratados en clase o realizar autoevaluaciones. Al igual que los chatbots orientados a servicios escolares, también pueden ayudar a brindar información, pero a una comunidad concreta. Se incluyen dentro de este grupo los siguientes trabajos: AI-IESLS [168], Atmosukarto *et al.* [169], Bala *et al.* [170], Bhargava y Maheshwari [171], Bigham *et al.* [172], BoC [173], CASEExplorer [174], CHARLIE [45], CareerBot [175], ChatBot [120], Chatbot [176], CikguAIBot [177], CultureBot [178], Doly [179], Dutta [180], EDUBOT [181], Edubot [182], El Hefny *et al.* [183], FeedBot&LitBot [184], GanttBot [185], Gómez Róspide y Puente [186], Hirose *et al.* [187], Infobot [188], Ismail *et al.* [189], KEMTbot [130], Kasthuri y Balaji [190], LTKA-Bot [191], Lecturer's Apprentice [192], MALL [155], Mikic-Fonte *et al.* [193], NLAST [194], Nguyen *et al.* [195], Nguyen *et al.* [135], Niranjan *et al.* [196], Oralbayeva *et al.* [197], Pereira y Barcina [198], Professor Atom [199], Python-Bot [200], QuizBot [201], SEG-COVID [202], Sreelakshmi *et al.* [203], Tian *et al.* [204], T-Bot/Q-Bot [44], TecCoBot [205], Tribubot [206], TutorDocente [139], UniBud [207], Ubibot [49], Waizmann *et al.* [208], Abot [209] y Your Physics Trainer [210].

Adicionalmente, se encontró otro tipo de chatbot educativo que surgió a partir de la pandemia de COVID-19. En un principio, se pensó como una posible categoría más pero, después de un análisis, se incorporó como una característica de los chatbots educativos **orientados a estudiante/profesor**. Dicha característica es **salud bienestar**, la cual se refiere a aquellos chatbots que pueden incorporar nuevos desarrollos dirigidos a los problemas de salud física y mental que surgieron a raíz del confinamiento y de la educación a distancia.

Con respecto al mecanismo de interacción, punto crucial para que el usuario pueda tener un acercamiento con el chatbot, se tienen los siguientes resultados: 72 trabajos que se basan en interacción por texto, 4 por voz, y 20 por texto y voz. Por lo tanto, el mecanismo que predomina es la interacción por texto.

3.7.3. Pregunta de investigación PI-3

Las respuestas para la pregunta de investigación PI-3 «¿Qué tecnología y arquitectura predominan en la construcción de chatbots que se utilizan en el sector educativo?» se

presentan a continuación:

- **Lenguajes de programación:** C#, C++, Kotlin, Java, JavaScript, PHP, Python y TypeScript.
- **Interfaz de usuario:** FB Messenger, KakaoTalk, Line, Microsoft Teams, Telegram y WeChat.
- **Técnicas y algoritmos de IA:** AIML (Artificial Intelligence Markup Language), Forward Chaining, GloVe (Global Vectors for Word Representation), Heuristics K-Means, K-Nearest Neighbor, LDA (Latent Dirichlet Allocation), Levenshtein's Distance LSA (Semantic Analysis), LSTM (Long Short Term Memory), n-gram, Naive Bayes, NER (Named Entity Recognition), Ontologies, Pattern Matching Petri Nets, PSA (Porter Stemmer Algorithm), Q-Learning, SIF (Smooth Inverse Frequency), SQS (Spreading Question Similarity) y word2vec.
- **Plataformas y herramientas de desarrollo de IA:** Chatfuel, Chatlayer, CoreNLP, Dialogflow, Google Speech, Microsoft Bot framework, NLTK, QnA Maker, Rasa, spaCy, TensorFlow, Vision AI, Watson y Wit.ai.
- **Base de Datos:** Datastore, Firebase, MariaDB, MongoDB, MySQL, PostgreSQL y SQLite.
- **Desarrollo web:** AJAX, CSS, HTML, JSON y XML.
- **Plataformas y marcos de trabajo:** Angular, Bootstrap, Django, Express, Flask, Node.js, ReactJS y SocketIO.
- **Arquitectura de software:**
 - **Cliente-servidor:** el cliente realiza una solicitud a otro programa, conocido como servidor, que procesa esa solicitud y da una respuesta al cliente [11].
 - **Capas:** es la representación abstracta de un sistema de software por capas (niveles); esta arquitectura divide el sistema en capas de servicios y funcionalidades [90].
 - **Microservicios:** se utiliza para el desarrollo de aplicaciones distribuidas, donde la aplicación se basa en componentes más pequeños, los cuales son aplicaciones en sí mismos [91].

- **Componentes:** es la separación del diseño en funcionalidad o componentes lógicos [89].
- **Modular:** es una descomposición del sistema en un subconjunto de elementos.

Se puede observar que las tecnologías y herramientas, que emplearon los trabajos encontrados, son diversas. Esto nos sugiere que los desarrollos dependen, en gran medida, de las necesidades específicas en cada contexto. Es importante señalar que no todos los estudios tienen los detalles de las arquitecturas, herramientas o tecnologías que utilizaron, a pesar de que el desarrollo de chatbots es una parte esencial de su investigación.

En el Capítulo 4 se complementará la respuesta a la pregunta de investigación PI-3.

3.8. Discusión

Los resultados obtenidos muestran un creciente interés en esta área. Se han desarrollado propuestas novedosas que pueden tener un profundo impacto en la educación. Sin embargo, este campo también está lleno de dificultades y áreas de oportunidad.

Las expresiones de búsqueda son un aspecto importante en este trabajo, ya que permiten automatizar la obtención de los estudios revisados para la presente investigación. No obstante, es posible encontrar obstáculos en este proceso.

Por un lado, las expresiones de búsqueda deben adaptarse a los motores de búsqueda, pues *Science Direct* y *Scopus* no autorizan el uso de comodines, como el asterisco ‘*’. Además, no permiten el uso de un número mayor a ocho operadores booleanos. Como se observa en el Cuadro 3.4, nuestras expresiones tienen un número superior al permitido. Si bien es importante el uso de sinónimos para abarcar un mayor rango de casos, también es cierto que puede ser contraproducente tener muchos de ellos para ciertos motores de búsqueda. En estos casos, lo mejor es revisar las opciones que permiten dichos motores. En caso de verse limitados en algún aspecto, como el que mencionamos anteriormente, se debe elegir aquellos sinónimos que puedan ser más relevantes o importantes. Por otro lado, los motores de *ACM*, *IEEE* y *Springer* no tienen las restricciones de *Science Direct* y *Scopus*, permitiendo más flexibilidad para recibir las cadenas de búsqueda del Cuadro 3.4.

Otro inconveniente que se puede encontrar, relativo a los motores de búsqueda, se debe a que, en algunos casos, son adquiridos por otras compañías y cambian la interfaz gráfica (búsqueda avanzada) o eliminan las funcionalidades que ya tenían (búsqueda de coman-

dos), lo cual puede dificultar la introducción de expresiones. *Elsevier* colabora con *Science Direct* y *Scopus*, así que no es de extrañar que utilicen funciones y restricciones similares. A propósito, los filtros en *IEEE* puede mostrar resultados combinados de artículos y conferencias, facilitando el trabajo de revisión. En cambio, otros motores, únicamente permiten elegir sólo un tipo de documento, por lo que se debe volver a realizar la búsqueda para otro tipo de documento. Esto limita en gran medida los resultados e incrementa el tiempo de revisión y trabajo.

Como ya se mencionó, la definición de chatbot ha cambiado con el tiempo, involucrando a la tecnología disponible. Es a partir de este hecho que se crean las expectativas de lo que se supone que deben hacer estas aplicaciones. Si bien es cierto que no existe una definición ampliamente aceptada, todas las que se pudieron encontrar tienen la misma esencia: un chatbot tiene que ser un constructo de Inteligencia Artificial capaz de interactuar a través de lenguaje natural con un humano, ya sea por texto, voz o ambos. El problema de la definición lleva a la segunda limitación general: el diseño centrado en el usuario. Es claro que el campo de la educación involucra a diversos actores, por lo que si se le preguntara a cada uno qué es un chatbot educativo, probablemente darían respuestas diversas que se corresponderían con contextos y necesidades específicas. Esto es de suma importancia, ya que los usuarios finales y las partes interesadas deben participar en el desarrollo de la herramienta [49]. Por supuesto, esto también conlleva dificultades.

Por un lado, si se habla de evaluaciones de experiencia de usuario y de usabilidad, no existen muchos métodos de evaluación específicamente orientados a chatbots y lo mismo sucede con las evaluaciones de impacto educativo [211, 212]. Por otro lado, algunos artículos tienen poblaciones muy escasas para probar [188]. Además, esto se vuelve complejo, debido a la naturaleza híbrida de este campo, ya que existen pruebas de rendimiento educativo o experiencia de usuario, pero sólo en muy pocos casos se evalúan ambos aspectos [163].

El avance de la tecnología ha permitido que los chatbots evolucionen de «contestadores automáticos» a verdaderos agentes conversacionales especializados, con los que uno puede interactuar con diversos grados de libertad. Esta tendencia se muestra en la Figura 3.2, ya que las propuestas de los últimos años se basan en lenguajes de programación, marcos de trabajo y arquitecturas que permiten un desarrollo mucho más rápido y económico. En contraste, estas tecnologías de desarrollo también revelan la primera limitación general: la falta de capacitación. Dado que los marcos de trabajo están diseñados para aplicaciones de

propósito general, muchos artículos tienen como trabajo futuro mejorar el entrenamiento de sus chatbots. En algunos casos, los usuarios agotan todas las rutas de conversación y el chatbot no les permite realizar su tarea objetivo [178]. En otros trabajos, se está optando por desarrollar su tecnología en el futuro, utilizando ontologías, redes neuronales u otros modelos de conocimiento [47].

En términos generales, se puede decir que la relativa juventud de los chatbots también conlleva una cierta inmadurez epistémica, *i.e.*, no se pudo observar un patrón o modelo general de cómo se deben presentar este tipo de desarrollos, cómo se deben realizar las pruebas o cómo explicar los resultados. Esto se debe a varios factores, especialmente a la naturaleza híbrida de estos sistemas, ya que hay propuestas que se enfocan más en el campo educativo [192] y otras que enfatizan algún aspecto de la Inteligencia Artificial [191]. Esto también puede contribuir a la tendencia que se muestra en la Figura 3.3, ya que las publicaciones en conferencias tienden a ser foros con un alcance más amplio que las revistas.

Un aspecto importante que se puede destacar es lo que se observó al responder las preguntas de investigación. La mayoría de los artículos examinados presentan chatbots destinados a resolver un problema de una comunidad local y las pruebas se realizan con miembros de dicha comunidad [179]. Estos trabajos tienen el objetivo futuro de implementar el chatbot con más grupos e incluso más escuelas. Sin embargo, como se mencionó anteriormente, existen pruebas con muestras no representativas y si a esto le sumamos el hecho de que muchos de estos desarrollos no presentan los detalles técnicos ni las herramientas necesarias para su construcción, esto nos lleva a la tercera limitante general: poca reproducibilidad.

La reproducibilidad es uno de los pilares fundamentales de la ciencia y, aunque podría discutirse extensamente, se puede resumir como resultados similares obtenidos en contextos similares [213, 214]. Por lo tanto, cuando un artículo no explica las tecnologías utilizadas para crear un chatbot y los datos obtenidos de sus experimentos no son claros, es difícil que otros investigadores en contextos similares se beneficien de estos estudios.

Finalmente, un aspecto que debe explorarse en el futuro son las repercusiones socio-políticas del uso de chatbots en la educación. Aspectos como la privacidad [215], la seguridad [216], el sesgo de la Inteligencia Artificial [217], la salud mental [218] y la asequibilidad tecnológica [219], por mencionar algunos pocos, necesitan ser estudiados a medida que estos agentes conversacionales se vuelven más avanzados y automatizan aspectos que, hasta

ahora, requieren interacción humana [220, 221, 222].

Capítulo 4

Estado del Arte

El Capítulo 3 respondió las tres preguntas de investigación planteadas, no obstante, consideramos que sólo la pregunta PI-1 fue respondida satisfactoriamente con la propuesta de definición de chatbot educativo. En consecuencia, en el presente capítulo, se profundiza en las preguntas restantes, PI-2 y PI-3. Para responder detalladamente la pregunta PI-2, primeramente se desarrolla una clasificación de chatbots educativos, aplicando el método iterativo de Nickerson *et al.* [80] (cf. Sección 4.1). A continuación se presenta una síntesis de los trabajos relacionados, organizada con base en la clasificación resultante (cf. Sección 8.2). A partir de dicha síntesis, se extraen las principales contribuciones y características de los chatbots educativos (cf. Sección 4.3). Por otra parte, para responder la pregunta de investigación PI-3, se detallan aspectos técnicos importantes de este tipo de chatbots (cf. Sección 4.4).

4.1. Clasificación de chatbots educativos

Para dar respuesta a la pregunta de investigación PI-2, planteada en el Capítulo 3 («¿En qué niveles educativos se utilizan los chatbots y cómo es el mecanismo de interacción?») se desarrolló una clasificación de chatbots educativos. Uno de los puntos más importantes fue la elección del método Nickerson *et al.* [80]. Este método es adecuado para formalizar el proceso, porque es flexible y tiene una base sólida que se establece mediante condiciones finales, pasos e iteraciones objetivas y subjetivas para obtener un resultado concreto. Sin embargo, puede ser confuso cuando se usa por primera vez porque, según Nickerson *et al.* [80], las condiciones subjetivas son difíciles de identificar y aplicar, y evaluarlas requiere

conocimiento, experiencia y habilidades de los investigadores.

4.1.1. Aplicación del método de Nickerson et al.

Una clasificación puede no ser perfecta, pero es útil si se quiere explicar la naturaleza de los objetos en un estudio [80]. El Cuadro 4.1 presenta el número de iteraciones que se obtuvieron y cómo se cumplieron las condiciones objetivas y subjetivas. Primeramente se muestra el desarrollo de la primera iteración:

Paso 1: el interés de la presente tesis se centra en las características de los chatbots educativos con respecto a la funcionalidad; esta es la meta-característica. Por lo tanto, cada característica obtenida en los pasos posteriores debe ser lógicamente consistente con dicha meta-característica.

Paso 2: las condiciones finales objetivas que se adoptaron son: 1 y de 3 a 6 (ver Cuadro 4.1). Además, se aceptaron las cinco condiciones finales subjetivas.

Cuadro 4.1: Detalle de las iteraciones con respecto a las condiciones finales objetivas y subjetivas.

Condiciones finales objetivas	Iteración 1	Iteración 2	Iteración 3
1) Se examinaron todos los objetos o una muestra representativa.	✓	✓	✓
3) Al menos un objeto corresponde a cada característica de cada dimensión.		✓	✓
4) No se agregaron nuevas características o dimensiones en la última iteración.			✓
5) No se fusionaron ni dividieron características ni dimensiones en la última iteración.			✓
6) Cada dimensión es única y no se repite.	✓	✓	✓
Condiciones finales subjetivas			
1) Conciso		✓	✓
2) Robusto		✓	✓
3) Comprensivo		✓	✓
4) Extendible	✓	✓	✓
5) Explicativo		✓	✓

Paso 3: en este paso comienza la primera iteración, pero es necesario seleccionar un enfoque. Se decidió seguir el enfoque *empírico a conceptual* porque los chatbots serán clasificados según sus características, *i.e.*, a partir de trabajos de chatbots educativos obtenidos de la literatura científica, se identificaron sus semejanzas y se analizaron si existen o no relaciones entre ellas.

Paso 4: para obtener la muestra, se usa el SMS [65] desarrollado en el Capítulo 3, ya que éste permite desarrollar una revisión amplia de estudios primarios en un dominio específico e identificar la evidencia disponible sobre el tema [64]. En particular, se retoma la expresión de búsqueda de la pregunta de investigación PI-2 (ver Cuadro 3.4), así como los criterios de inclusión, exclusión y selección (ver Cuadros 3.5, 3.6 y 3.7). Después, se revisa la muestra y se realizan los siguientes pasos.

Paso 5: se identifican las siguientes dimensiones: *Orientados a Servicios Escolares*, *Orientados a Educación en Línea* y *Orientados a Estudiante*. Asimismo, se identifican las siguientes características: *Evaluación*, *Retroalimentación*, *Información*, *FAQ*, *Procedimientos*, *Cursos*, *LCMS (Learning Content Management Systems)*, *Software*, *Materias*, *Preguntas y Respuestas (P&R)*, *Tutoría*, *Escuelas*, *Aprendizaje en Línea*, *Usuarios y Universidades*.

Paso 6: la agrupación de características en dimensiones se define de la siguiente manera. La dimensión *Orientados a Servicios Escolares* tiene las características *Información*, *FAQ* y *Horario*. Por otro lado, la dimensión *Orientados a Educación en Línea* está asociada a las características *Cursos*, *LCMS* y *Software*. Finalmente, la dimensión *Orientados a Estudiante* posee las características *Evaluación*, *Retroalimentación*, *P&R*, *Materias*, *Horario* y *Tutorías*.

Paso 7: aún no se han cumplido tres condiciones finales objetivas y cuatro subjetivas. Por lo tanto, es necesaria una nueva iteración, comenzando desde el paso 3.

En la segunda iteración (ver Cuadro 4.1), también se selecciona el enfoque *empírico a conceptual* por la misma razón que en la primera iteración. Continuando con los pasos del 4 al 6, se identifica la dimensión *Profesor*. Para que el proceso sea conciso y completo, se procede a agrupar y renombrar la dimensión *Orientados a Estudiante* como *Orientados a Estudiante/Profesor* y tiene las siguientes nuevas características: *Reportes*, *Soporte* y *Salud Bienestar*. Además, se identifican nuevas características, *Procedimientos* y *Calendario*, para la dimensión *Orientados a Servicios Escolares*. Dada la similitud de la nueva característica *Calendario* con la característica anterior, *Horario*, se procede a agruparlas y renombrarlas como *Calendario y Horario (C&H)*. Las condiciones finales no se han cumplido porque se creó una nueva dimensión en esta iteración. Por lo tanto, es necesaria una nueva iteración.

En la tercera iteración (ver Cuadro 4.1), también se selecciona el enfoque *empírico a conceptual* y no se han encontrado nuevas dimensiones y características en los pasos

posteriores. Por lo tanto, el método de clasificación no necesita una nueva iteración, ya que se cumplieron las condiciones objetivas y subjetivas de finalización.

4.1.2. Resultados

Después de aplicar el método de Nickerson *et al.* [80], se encontraron tres dimensiones y quince características, las cuales se presentan en la Figura 4.1.

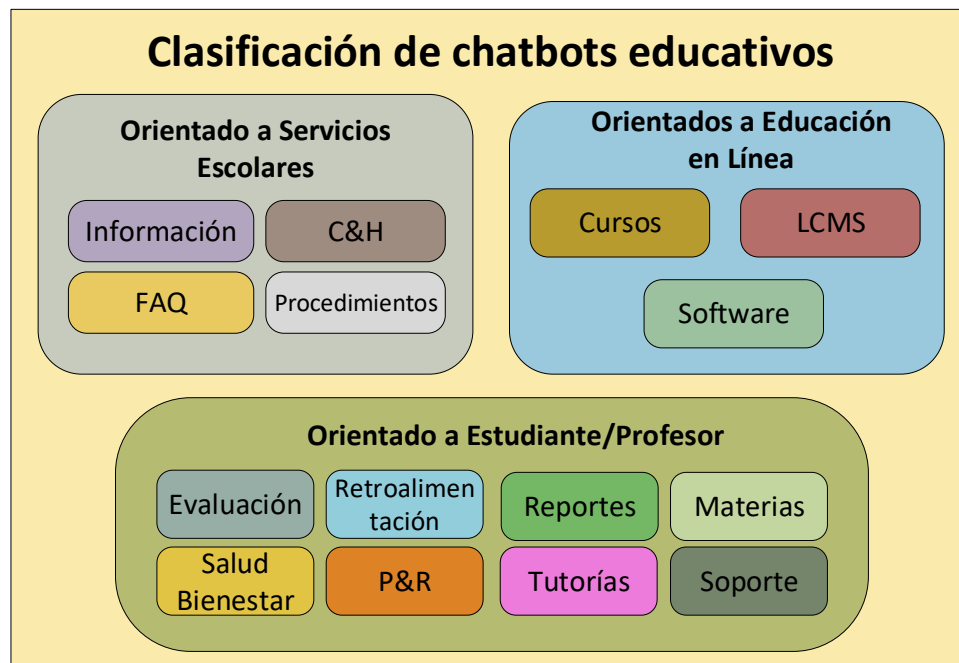


Figura 4.1: Estructura de la clasificación de chatbots y sus características en el dominio educativo.

La dimensión *Orientados a Servicios Escolares* agrupa a los chatbots cuya función principal es brindar información general, FAQ, horarios o procedimientos. Este tipo de chatbots es útil para las instituciones educativas, ya que pueden brindar un servicio completo tanto a su comunidad como a usuarios externos que necesitan más información sobre tarifas u oferta educativa. Las principales ventajas de este tipo de chatbots son la disponibilidad 24/7, la reducción de la carga de trabajo para el personal, el manejo de muchos estudiantes al mismo tiempo y la accesibilidad desde un dispositivo móvil. Las características de esta dimensión son:

- *Información*: el chatbot proporciona información a los usuarios, *e.g.*, oferta educativa, directorio y planes de estudio.
- *FAQ*: preguntas y respuestas que son comunes para los usuarios.
- *Procedimientos*: sirve de guía para que los estudiantes realicen un trámite específico, *e.g.*, cómo inscribirse en una clase o qué requisitos debe cumplir para obtener su certificado.
- *C&H*: información específica sobre actividades, *e.g.*, eventos académicos, horarios laborales del personal e instalaciones, periodos vacacionales y fechas de evaluaciones.

La dimensión *Orientados a Educación en Línea* cubre los chatbots diseñados para MOOC, LCMS y software orientado a la educación, *e.g.*, el sistema Moodle utilizado por estudiantes en modo remoto. Este tipo de chatbots son un complemento de cursos masivos que no necesariamente forman parte de un ambiente académico formal, *e.g.*, clases de inglés como lengua extranjera. Las características de esta dimensión son:

- *Cursos*: en este caso, los chatbots son implementaciones *ad hoc*, cuyos usuarios pueden inscribirse de forma remota, *e.g.*, cursos masivos sobre el aprendizaje de un idioma extranjero.
- *LCMS*: son chatbots que están diseñados para integrarse en una plataforma MOOC existente.
- *Software*: se trata de chatbots que son una herramienta más en un sistema especializado, generalmente accesible a través de la Web.

La dimensión *Orientado a Estudiante/Profesor* agrupa a los chatbots que no sólo interactúan con los estudiantes, sino también con los profesores. Se encontraron varios elementos como: evaluación, FAQ, comentarios, preguntas y respuestas, informes, cronogramas, temas, apoyos y tutorías. Las características de esta dimensión son:

- *Evaluación*: se trata de herramientas de evaluación para estudiantes, *e.g.*, exámenes, tareas, cuestionarios, prácticas y ensayos.
- *Retroalimentación*: el chatbot proporciona comentarios a los estudiantes, de acuerdo con su progreso en la clase.

- *P&R*: maneja preguntas específicas y da respuestas concretas al estudiante sobre un tema específico de una materia.
- *Reportes*: detalles proporcionados por el chatbot al profesor sobre el progreso de sus alumnos.
- *Materias*: el chatbot interactúa con el alumno sobre las clases que tiene matriculadas.
- *Soporte*: ofrece algún tipo de apoyo al estudiante, *e.g.*, cómo conectar un dispositivo electrónico a la red del laboratorio.
- *Tutorías*: ofrece a los estudiantes algún tipo de orientación educativa o personal.
- *Salud Bienestar*: canaliza al estudiante para atención por problemas de salud física y mental a raíz del confinamiento, la educación a distancia por COVID-19 o incluso acoso escolar.

Estas dimensiones pueden ser de valor para los especialistas en el área, ya que ofrecen una visión general del trabajo realizado en la literatura científica y pueden servir de guía para mejorar las implementaciones de los chatbots educativos.

Como se puede ver en la Sección 2.2 del Capítulo 2, la mayoría de los trabajos siguen una metodología *ad hoc*, lo que impide su uso en otros dominios (*e.g.*, chatbots educativos) y dificulta la replicación de los resultados obtenidos. En cambio, el método Nickerson *et al.* [80] establece elementos generales que facilitan su uso para diferentes dominios. Por lo tanto, la clasificación propuesta en esta tesis tiene la ventaja de la reproducibilidad, ya que la pregunta de investigación, la cadena de búsqueda y los criterios de inclusión, exclusión y selección están bien establecidos y sentados sobre bases sólidas. Además, la posibilidad de elegir entre enfoques empíricos/inductivos y conceptuales/inductivos ofrece flexibilidad y versatilidad en el desarrollo de la clasificación.

De la clasificación de chatbots educativos presentada anteriormente surgen tres grupos, que clasifican a los chatbots según sus características. No sólo se muestra cómo se llegó a estos grupos, también las particularidades de cada uno. Esto es un avance con respecto a otros trabajos de clasificaciones menos detallados, como el propuesto por Quiroga Pérez *et al.* [75]. A diferencia de aquellas propuestas, que siguen métodos arbitrarios, nuestra clasificación permitió representar más chatbots en un grupo caracterizado, sin necesidad

de crear pequeños clusters que pueden ser triviales. Esto se debe al uso de un proceso sistemático [80].

Respecto a las tres clasificaciones que resultaron de las iteraciones (ver Figura 4.1), se puede mencionar que están relacionadas de acuerdo a la granularidad de la información que el chatbot puede enviar y recibir. Los chatbots que están orientados a tareas más generales y que, por lo tanto, manejan información básica se encuentra en el grupo *Orientados a Servicios Escolares* ya que los usuarios objetivo suelen ser público en general que busca información rudimentaria. Por el contrario, los chatbots más especializados se clasifican como *Orientados a Estudiantes/Profesores*. Estos chatbots cubren la necesidad de interacciones más personalizadas y realizan tareas mucho más precisas y complejas, ya que están dirigidos a individuos. Los chatbots *Orientados a Educación en Línea* generalmente están en algún punto intermedio pues, aunque también atienden a una población amplia, su contexto de uso es más limitado.

Según Nimavat y Champaneria [72] y Adamopoulou y Moussiades [74], es posible tener varias categorías para un chatbot, *e.g.*, KEMTbot [130] y TutorDocente [139] están dentro de *Orientados a Servicios Escolares* y *Orientados a Estudiante/Profesor*. Sin embargo, estos casos son raros porque los chatbots, en general, se limitan a dar respuestas predefinidas de una base de datos o responder preguntas en dominios cerrados [63].

Se pueden analizar las implicaciones de la clasificación propuesta desde dos puntos de vista. El primero beneficia a los desarrolladores, ya que se identificaron los mecanismos básicos de interacción que se deben implementar de acuerdo a su contexto de desarrollo. Por ejemplo, si se desarrolló un chatbot para una clase de secundaria, que será utilizado tanto por estudiantes como por profesores, entonces el grupo *Orientado a Estudiante/Profesor* es relevante porque les dice que tienen que crear mecanismos que satisfagan las características de esa categoría.

Por otro lado, esta propuesta de clasificación también impacta a los grupos de investigación, ya que se creó una base para más trabajos sobre la clasificación de chatbots orientados a la educación. A partir de la expresión de búsqueda, los criterios de inclusión, exclusión y selección, así como el enfoque y las condiciones finales, los investigadores pueden encontrar más grupos de características en los próximos años.

Una limitación del método de Nickerson *et al.* [80] es su carácter cualitativo, por lo que no es posible realizar un análisis formal cuantitativo. Sin embargo, se puede decir que las características más frecuentes de la muestra de trabajos obtenida fueron las de

Información, Evaluación, Retroalimentación, Materias y P&R. Esto es interesante porque representa una mezcla de funciones orientadas a la automatización con otras, que son más avanzadas y requieren que el chatbot participe activamente en el proceso de enseñanza/aprendizaje.

Por otro lado, parece que hay una tendencia a la desaparición de la investigación en las siguientes funciones rudimentarias: *H&C, Procedimientos, Reportes, FAQ y Soporte.* Esto puede deberse a que las nuevas tecnologías han resuelto todos los problemas de estas funcionalidades. Otras características menos comunes fueron *Cursos, LCMS, Software y Tutorías*, lo cual puede atribuirse al costo y a la complejidad de estos sistemas.

Si bien este análisis no sustituye a un método formal, lo que representa la principal limitación de esta propuesta, se podría complementar con herramientas cuantitativas como las que ofrecen las escalas de Likert [223], el marco de Szopinski *et al.* [224], la Evaluación Integral Fuzzy [225] y el proceso de jerarquía analítica [226].

4.2. Síntesis de los trabajos relacionados

Esta sección se describen los trabajos relacionados con los chatbots enfocados a la educación. Primero, se presentan chatbots que fueron pensados para cursos masivos, componentes de sistemas de software o complementos para algún manejador de contenido (ver Sección 4.2.1). Después, se describen aquellos que tienen como prioridad proporcionar información referente a la misión y visión de la institución educativa, el calendario de actividades a lo largo del ciclo escolar, procedimientos o FAQs (ver Sección 4.2.2). Finalmente, se presentan chatbots que se enfocan en el aprendizaje de los alumnos (ver Sección 4.2.3).

Cabe mencionar que dada la cantidad de artículos recopilados en el Capítulo 3, se optó por mostrar un resumen de los trabajos que tuvieron relevancia para lograr establecer la clasificación de chatbot educativos presentada en la Subsección 4.1.2.

4.2.1. Educación en línea

Mikic-Fonte *et al.* [153] describió el prototipo funcional de una plataforma de Educación en Línea llamada INES. El sistema combina tres capacidades esenciales: *LMS (Learning Management Systems)* [227], *LCMS (Learning Content Management Systems)* [228] e *ITS*

(*Intelligent Tutoring Systems*) [229]. Los autores describen el uso de varias ontologías [230] interconectadas entre sí. La ontología *Ontology of Users* identifica los elementos para describir las características de cada usuario del sistema y se basa en *LIP (Learning Information Package)*. La ontología *Ontology of Contents* obtiene esquemas de metadatos que permiten describir y caracterizar los contenidos educativos y se basa en el modelo *LOM (Learning Object Metadata)* [231]. La ontología *Ontology of Traces* tiene en cuenta la información sobre la actividad de cada alumno que utiliza la plataforma. Cabe señalar que INES no es un chatbot, sino un sistema conformado por varios componentes, uno de los cuales es el chatbot [45].

Benotti *et al.* [142] presentó el software llamado **Chatbot** que está basado en el chatbot **ALICE (Artificial Linguistic Internet Computer Entity)**. **Chatbot** es una herramienta de software educativo, diseñada para alentar y motivar la participación de los estudiantes en Ciencias de la Computación en el concurso en línea «Dar aceptar», organizado por la Fundación Sadosky en Argentina. Los concursantes usan **Chatbot** para codificar sus propios bots. **Chatbot** es de código abierto y se basa en coincidencia de patrones (expresiones regulares). Así, **Chatbot** analiza las entradas y si encuentra una coincidencia, obtiene una respuesta para ese patrón.

Troussa *et al.* [155] tomaron el chatbot **ALICE** como referencia para crear un chatbot que se integró en un sistema llamado **MALL (Mobile-Assisted Language Learning)**. Su objetivo es promover el idioma inglés, a través de mecanismos para dar apoyo a los estudiantes con el vocabulario, conceptos básicos y herramientas que sean significativas para aprender más sobre el idioma y la cultura detrás de él. **ALICE** se implementó con el lenguaje de programación *Java*, para facilitar su uso en dispositivos móviles. Entre otras tecnologías, se encuentran *XML (Extensible Markup Language)*, *PHP*, *JSON (JavaScript Object Notation)* y *AIML (Artificial Intelligence Markup Language)*. Las principales contribuciones son la creación de un diccionario, que permite a los usuarios pedir la definición de una palabra al chatbot.

Song *et al.* [163] desarrollaron un chatbot para promover el aprendizaje en cursos en línea de educación superior. El flujo de la interacción es el siguiente: el usuario accede al sistema a través de un navegador Web, el sistema genera los saludos y pregunta el nombre del usuario, luego el sistema hace preguntas al usuario. El sistema está basado en una arquitectura cliente-servidor. Del lado del cliente están los *sockets* de servicios y los módulos para la interfaz de usuario, que controlan la apariencia del texto en la

ventana de chat. Del lado del servidor se incluyen *Node.js* y *MongoDB*. Además, tiene cuatro módulos externos: *Type*, *Message*, *DB* y *Socket*. El módulo *Type* tiene cuatro submódulos: *user response preprocessing*, *user response categorizing*, *handler processing* y *response generating*. Cuando el usuario responde a una pregunta, la respuesta se envía al servidor a través de la *Web Socket*. Luego, el módulo *Message* envía el texto al módulo *Type*, que procesa el texto y categoriza las respuestas. A continuación, el módulo *Handler Processing* busca un controlador adecuado para la base de datos con información de los posibles tipos de respuesta. El módulo *response generating* recupera una serie de oraciones del documento de respuestas en *MongoDB*. Las respuestas se envían al módulo *Message* para agregar información adicional: código de tiempo, categoría, ID de la sesión de usuario, etc. Finalmente, el mensaje se envía al cliente. La conexión entre el cliente y el servidor se realiza mediante la biblioteca JavaScript *SocketIO*. Las herramientas técnicas son *Angular*, *MongoDB*, *Node.js* y *Typescript*.

Verleger y Pembridge [146] desarrollaron una interfaz de chatbot inteligente, llamada *EduBot*, para un curso de introducción a la programación de sistemas. Para el *backend*, esta propuesta utiliza sistemas de educación en línea como *IMS-Global LTI Integration Standard* para interactuar con *Canvas LMS*¹ y *Microsoft QnA Maker*² para el procesamiento de Inteligencia Artificial. *Microsoft QnA Maker* permite la creación simple de bots, utilizando datos y sin experiencia previa en codificación. Por otro lado, las interacciones adicionales se manejan mediante una interfaz *PHP* personalizada. *EduBot* fue desarrollado como un complemento de *Canvas LMS*. Sin embargo, si los estudiantes necesitan hacer preguntas específicas, *EduBot* no es una opción.

Clarizia *et al.* [143] desarrollaron un prototipo de chatbot que hace uso de lenguaje natural para servir de apoyo en cursos de informática y redes. El chatbot se basa en una arquitectura cliente-servidor y se compone de los siguientes módulos: 1) el módulo *Front-end*, que consiste de una interfaz de usuario accesible a través de tabletas, teléfonos inteligentes o PC; 2) el módulo *Back-Office* para la gestión de operaciones que se ejecutan del lado del servidor y trabaja en colaboración con la base de datos; 3) el módulo *Knowledge Base* donde se almacenan los datos de las FAQ, objetos de aprendizaje y usuarios; y 4) el módulo *E-learning BOT* que es el motor de inferencia del chatbot. Aunque éstos son los módulos más importantes en la arquitectura del sistema, no se descarta el

¹<https://www.instructure.com/canvas>

²<https://www.qnamaker.ai/>

uso de servicios web externos e incluso recursos de redes sociales. Entre las herramientas utilizadas, se encuentran modelos de *Machine Learning* como LDA (*Latent Dirichlet Allocation*) [232, 233], Sistemas Colaborativos, Ontologías y Redes de Petri [234].

Pham *et al.* [148] crearon un chatbot integrado a un sistema llamado **English Practice**. Se instala en dispositivos móviles y la interacción con los usuarios es a través de una ventana de chat. **English Practice** es una aplicación de Android y está disponible para descargar desde Google Play. Ofrece funcionalidades a los usuarios como cuestionarios, lecciones, vocabulario y salas de chat. Las características de este chatbot incluyen recordar automáticamente a los estudiantes que estudien y sugerir algunas respuestas a preguntas de opción múltiple. **English Practice** se basa en una arquitectura cliente-servidor. Del lado del cliente, hay tres módulos de aprendizaje que son *Practice*, *Review/Test* y *Chatbot*. Los dos primeros permiten a los estudiantes practicar inglés y el tercero es la interfaz de usuario que permite a los usuarios consultar el contenido de aprendizaje mediante comandos. Del lado del servidor, se almacena el material de aprendizaje y el progreso del usuario. El núcleo de la aplicación es el módulo *Chatbot*, el cual está basado en Dialogflow³.

Goschlberger y Brandstetter [151] diseñaron un prototipo de un sistema corporativo de educación en línea, que ofrece recursos de aprendizaje a través de una interfaz de chat. El prototipo se basa en una arquitectura de microservicios. Cuando se utiliza este tipo de arquitectura, la aplicación debe construirse como un conjunto de pequeños servicios que interactúan entre ellos mediante una *API* (*Application Programming Interface*). La arquitectura se compone de los siguientes microservicios: 1) *Admin Interface* permite a los administradores cargar documentos en la base de datos, luego estos documentos se analizan para extraer la información relevante que utilizará la interfaz de chat; 2) *Backend* tiene un conjunto de servicios que se utilizan para almacenar documentos y leer y escribir información en la base de datos *Postgres*, así como un motor de búsqueda llamado *Elasticsearch*; 3) *Postgres* almacena todos los datos relevantes que se cargan como documentos; 4) *Elasticsearch* indexa información sobre documentos cargados que son relevantes para las consultas; 5) *Conversational AI Platform* es un motor de inferencia basado en Dialogflow; y 6) *Chat UI* presenta la interfaz de usuario y está conectado a *Conversational AI Platform*.

Oliveira *et al.* [158] desarrollaron un chatbot que interactúa con los estudiantes, a través de mensajes de texto, en cursos de educación superior. La arquitectura se basa en

³<https://dialogflow.com/>

los servicios que proporciona la API de IBM Watson y la aplicación está alojada en IBM Cloud. La interfaz de usuario del chatbot está integrada en una plataforma Moodle VLE (*Virtual Learning Environment*) y se puede acceder a través de *Facebook Messenger* o la Web. La metodología propuesta para esta tarea consta de tres pasos: 1) se crea la API para el chatbot en Facebook y se obtienen los permisos de escritura, lectura o modificación de los datos, los cuales son otorgados por la API de IBM Watson; 2) se conecta el chatbot con *Facebook Messenger* usando los permisos de acceso del primer paso; y 3) se inserta el chatbot en Moodle. El chatbot funciona de la siguiente manera: el estudiante accede a la interfaz de usuario a través de una PC o un dispositivo móvil, luego los alumnos hacen una pregunta y finalmente se utiliza Watson para dar una respuesta rápida y asertiva.

Pereira y Barcina [198] proponen solucionar uno de los problemas a los que se enfrentan los estudiantes a la hora de redactar los FDP (*Final Degree Projects*). Entre las principales dificultades que tienen los alumnos se encuentra la falta de experiencia y habilidades necesarias para redactar correctamente los informes. La investigación muestra, entre otras cosas, que en varias ocasiones los estudiantes copian y pegan textos o imágenes sin poner la referencia de la información, lo que puede interpretarse como posible plagio, entre otras cosas. Para hacer frente a este problema, se construyó **Ikastenbot**, que actúa como una interfaz donde los estudiantes pueden cargar sus FDP mientras los redactan para así detectar posibles errores de ortografía y gramática, tamaño inadecuado de texto e imágenes y falta de referencias. **Ikastenbot** se programó como una interfaz para la aplicación *Telegram*. Además, no sólo recibe texto como entrada, sino también archivos pdf, audio y video. De esta forma, los estudiantes escriben sus informes FDP y generan un archivo PDF para cargarlo en **Ikastenbot** con un simple comando. El chatbot analiza el informe y pone a disposición algunos comandos. Por ejemplo, se puede usar un comando para verificar errores de ortografía y gramática; otro comando permite saber si las imágenes en el PDF están referenciadas correctamente; incluso existe un comando para detectar si el texto de los párrafos está mal referenciado. **Ikastenbot** se basa en una arquitectura cliente-servidor y define cinco módulos principales: 1) un módulo para detectar posibles plagios que se basa en la implementación de la herramienta *text-matcher*⁴; 2) un módulo para detectar malas referencias de imágenes, que utiliza el servicio Google Cloud Vision y una librería ML con TensorFlow de la misma empresa; 3) un módulo para detectar la falta de bibliografía básica; 4) un módulo para revisar ortografía y gramática, que se basa

⁴<https://github.com/JonathanReeve/text-matcher>

en la implementación de *LanguageTool*⁵ y 5) una interfaz del chatbot en *Telegram*. Cabe mencionar que el chatbot tiene licencia de código abierto.

Ruán *et al.* [201] crearon **QuizBot**, el cual ayuda a los estudiantes a aprender sobre ciencia, seguridad y vocabulario en inglés. Fue diseñado como una herramienta de aprendizaje para el público en general y es accesible a través de dispositivos móviles. **QuizBot** tiene dos modos de ejecución: 1) en el modo *quiz*, **QuizBot** le hace al usuario una pregunta seleccionada por un algoritmo y 2) en el modo *chat*, los usuarios pueden interactuar con **QuizBot**. El usuario tiene tres opciones para responder la pregunta: 1) escribir la respuesta si el usuario la conoce; 2) tocar el botón «Pista» en la pantalla; y 3) tocar el botón «No sé». Si el usuario envía su respuesta, **QuizBot** la revisará y evaluará, utilizando un algoritmo de similitud. Con el botón «Pista», el chatbot presentará la respuesta correcta junto con otras opciones incorrectas y el usuario elegirá una de las opciones presentadas. Después de ver la respuesta correcta, el usuario puede tocar el botón «Por qué» para obtener una breve explicación. **Quizbot** utiliza el algoritmo SIF (*Smooth Inverse Frequency*) [235] en combinación con GloVe (*Global Vectors for Word Representation*) [236] para proporcionar una respuesta automática y el modelo DASH (*Difficulty, Ability, and Study History*) [237] para la secuenciación de preguntas. Además de emplear *Python Flask*⁶ en el lado del servidor, se usa *Facebook Messenger* como interfaz de chat en el lado del cliente.

Damasceno *et al.* [144] proponen el chatbot **STUART**. Este chatbot interactúa con los estudiantes en lenguaje natural, brindando automatización para apoyar el proceso de enseñanza y aprendizaje. **STUART** implementa una arquitectura por capas. En el *Front-end* se encuentra la capa *UI (Vista)*, donde reside la interfaz de usuario la cual es accesada por medio de un navegador Web del lado del cliente; en esa misma capa se encuentran los objetos *Chat*, *Ejercicio* y *Evaluación*. En el lado del *Back-end* existen dos capas. La capa de *Inteligencia* que contiene el componente principal *Chatbot (STUART)* que utiliza *Dialogflow* y otro componente *MRC* para producir una respuesta apropiada en caso de no contar con una. En la capa *Servicio* tiene los componentes para la comunicación con el *VLE Aprendizaje Accesible de Dell*.

⁵<https://languagetool.org>

⁶<https://flask.palletsprojects.com/>

4.2.2. Servicios escolares

Valle-Rosado *et al.* [46] desarrollaron el chatbot UMT-BOT, el cual resuelve las dudas que tienen los estudiantes en el proceso de graduación. El objetivo de UMT-BOT es intentar minimizar el tiempo que los alumnos pasan con el personal administrativo. El chatbot está disponible las 24 horas del día, los 7 días de la semana, por lo que la información está disponible en cualquier momento y en cualquier lugar, siempre que los usuarios tengan una conexión a Internet. De esta forma, el estudiante ahorra tiempo al no tener que establecer citas con el personal a cargo. Se utilizó *Program-O*⁷, que es un motor AIML escrito en *PHP* para la creación de chatbots.

Agus Santoso *et al.* [22] propusieron el chatbot DINA (*Dinus Intelligent Assistance*), que ofrece el servicio de información sobre el proceso de admisión a los estudiantes candidatos de una universidad. DINA usa una base de conocimientos como núcleo para seguir un enfoque de *Machine Learning*. Al mismo tiempo, extrae un patrón de la base de conocimientos que puede ser utilizado como respuesta al usuario. Para la base de conocimientos, se utilizó el libro de visitas de la Universidad Dian Nuswantoro como fuente. Este libro contiene preguntas y respuestas sobre los servicios de admisión. DINA toma plataformas como *Telegram*, *Line* o *Websites* como entradas. El análisis del texto se realiza mediante Procesamiento de Lenguaje Natural para encontrar lo que el usuario está preguntando y responder en consecuencia. DINA se basa en una arquitectura modular, donde cada módulo contiene una base de conocimientos, mecanismos de inicialización y manejo lógico de las solicitudes de los usuarios. Los datos son administrados por ontologías y luego pasan a un formato de Dialogflow.

Chun Ho *et al.* [129] describieron el desarrollo del chatbot EASElective. Este chatbot tiene la función de complementar los servicios de asesoría académica en la selección de cursos, a través de lenguaje natural y una interfaz de usuario interactiva. Las características de EASElective son: acceso permanente, información oficial, opiniones de compañeros, análisis de posibles opciones de cursos y recomendaciones. El chatbot se basa en una arquitectura cliente-servidor. La interfaz de usuario es una página web o una aplicación de mensajería instantánea como *Telegram*. El almacenamiento de la información sobre los cursos y las opiniones de los alumnos se realiza mediante *MongoDB* y *MySQL*. Las tecnologías utilizadas son *Node.js*, *PHP*, *JSON* y *API*. Además, la propuesta emplea dos

⁷<https://program-o.com/>

servicios en la nube, el primero es para la detección y gestión de la conversación con Watson⁸ y el segundo es para el análisis de texto.

Hien *et al.* [47] se enfocan en entornos de aprendizaje inteligentes y presentan el chatbot FIT-EBot que responde a los estudiantes de la Facultad de Tecnología de la Información de la Universidad de Ho Chi Min en Vietnam. La arquitectura de FIT-EBot consiste de una interfaz de usuario a la que se accede a través de *Facebook Messenger*, ya sea desde el sitio Web de Facebook o desde la aplicación para dispositivos móviles. Otro elemento arquitectural importante es el modelo de conocimiento, el cual se compone de una base de datos relacional en *MySQL* y del servicio *Dialogflow* de Google.

Dibitonto *et al.* [48] presentan los resultados del desarrollo de un chatbot llamado LiSA (*Link Student Assistant*), el cual pretende informar a los estudiantes de una universidad sobre lecciones y cambios en los calendarios de clase. La metodología detrás de LiSA consiste en el desarrollo de un sitio Web en Facebook, la aplicación de encuestas a los estudiantes y el uso de la herramienta *Chatfuel*⁹ que facilita la integración con *Facebook Messenger*. La información de los estudiantes sirve para alimentar, de cierta manera, al chatbot porque sus respuestas ayudaron a definir la base de datos.

Heo y Lee [126] desarrollaron CiSA (*Chatbot for International Students and Academics*) para la Universidad de Kookmin. CiSA permite a estudiantes y profesores extranjeros adquirir información esencial, tanto académica como de la vida universitaria. El chatbot fue diseñado con un enfoque incluyente, *e.g.*, identidad, edad, género, fluidez del idioma, discapacidad, antecedentes culturales, nacionalidad y nivel socioeconómico. Los pasos para el diseño de CiSA son: definición del concepto, definición del bot, análisis de características, diseño de conversación, creación de estrategia de contenido y creación de prototipos y pruebas. CiSA ofrece dos opciones de diálogo: 1) de dominio abierto, donde el usuario puede escribir sobre cualquier tema y 2) de dominio cerrado, donde se proporcionan opciones predefinidas al usuario. Sin embargo, sólo se comunica en inglés y es necesario agregar más funciones para cubrir otros temas. CiSA está basado en *Dialogflow* e integrado en *Facebook Messenger*.

Ondáš *et al.* [130] se centraron en la integración de un conjunto de tres chatbots individuales como herramientas para compartir información y apoyar el aprendizaje de los estudiantes. El primero se llama KEMTbot y su función es brindar información sobre el per-

⁸<https://www.ibm.com/cloud/ai>

⁹<https://chatfuel.com/>

sonal y sus oficinas en el Departamento de Electrónica y Telecomunicaciones Multimedia. El segundo chatbot también brinda información del departamento a través de preguntas y respuestas, empleando voz en idioma inglés mediante dispositivos *Amazon Echo*. El tercer chatbot fue diseñado para apoyar a los estudiantes en los cursos de sistemas de bases de datos. Dialogflow y *Firebase* se usaron en el primer y tercer chatbot, y *AWS (Amazon Web Services)* se empleó en el segundo.

Lee *et al.* [132] desarrollaron un chatbot interactivo para FAQ en la Universidad Internacional de Corea. Su objetivo es ayudar al personal administrativo a procesar el alto número de solicitudes, muchas de las cuales son preguntas sencillas que pueden ser respondidas, a través de una página web. El chatbot se basa en la aplicación de mensajería instantánea *KakaoTalk*, que es la más popular en Corea. El chatbot utiliza Dialogflow.

Zahour *et al.* [128] presentan un prototipo de chatbot llamado **E-orientation** para apoyar a estudiantes de pregrado y posgrado. Su objetivo es ofrecer a los estudiantes orientación profesional, utilizando el modelo de Holland, para reconocer sus características dominantes, personalidad y aptitudes. **E-orientation** usa Dialogflow y Facebook.

Mekni *et al.* [133] presentan un chatbot que brinda apoyo a las comunidades de estudiantes, personal y facultad del estado de Minnesota en los EEUU. El sistema está disponible las 24 horas del día, es compatible con aplicaciones de mensajería instantánea como *Facebook Messenger*, *Slack* o *Microsoft Skype* y brinda información sobre cursos, aulas, número de créditos, descripción, entre otras funciones. La interacción es por texto. La propuesta tiene una arquitectura de tres capas: 1) la capa *Channel* es el medio para comunicarse con los usuarios; 2) la capa *Bot Platform* tiene los componentes para establecer la conexión con la capa *Channel* y realizar el almacenamiento de control y la gestión de *Machine Learning* mediante Dialogflow; y 3) la capa *Dashboard* tiene los componentes para la administración de usuarios, FAQ y programación de cursos. Además, se utilizaron *Node.js*, *MySQL* y *EJS* (plantillas de JavaScript integradas).

Cordero *et al.* [139] exponen los resultados de usar tres chatbots: 1) **TutorDocente** responde preguntas sobre temas específicos de una materia; 2) **eBooks** guía al usuario para descargar material bibliográfico; y 3) **BecasUTPL** muestra información sobre becas. Cabe mencionar que se utilizaron metodologías ágiles como Scrum y Programación Extrema. Se propuso una arquitectura de tres capas, presentación, negocio y datos, las cuales se comunican a través del protocolo HTTP. La interacción entre el usuario y los chatbots es textual vía una interfaz gráfica. Las herramientas que se utilizaron son *Watson*, *Node.js*,

Postgres e IBM Cloud.

4.2.3. Estudiantes y profesores

Mikic-Fonte *et al.* [44] construyeron dos chatbots, T-Bot y Q-Bot, que se basan en ALICE [238]. T-Bot es capaz de analizar las solicitudes de los estudiantes escritas en lenguaje natural, de tal forma que pueda dar una respuesta adecuada para orientarlos. Q-Bot está enfocado en monitorear y supervisar el progreso del estudiante a través de cuestionarios personalizados. Tanto T-Bot como Q-Bot pueden evaluar a los estudiantes usando lenguaje natural; además, los chatbots son compatibles con los idiomas inglés, español y gallego. T-Bot y Q-Bot se basan en una arquitectura cliente-servidor. Esta propuesta ofrece un mecanismo para la integración de chatbots en plataformas de aprendizaje en línea, como *Claroline* y *Moodle*. Las herramientas utilizadas son AIML [239], *PHP* (acrónimo recursivo de PHP: preprocesador de hipertexto), *Program E* y *AJAX* (*JavaScript* asíncrono y *XML*).

Bigham *et al.* [172] realizaron un taller en el que 15 estudiantes ciegos de nivel secundaria crearon y personalizaron chatbots. Los estudiantes usaron lectores de pantalla como *JAWS*, *Window-Eyes* y *ZoomText*. Estos programas convierten las interfaces visuales y la información en un formato de audio que los estudiantes pueden escuchar o simplemente escalan los elementos visuales de acuerdo con sus componentes y relaciones. La interacción es mediante atajos de teclado y software para codificar como *.NET C#*. Finalmente, se utilizó *Windows Live Messenger* como interfaz de chatbot.

Mikic-Fonte *et al.* [45] presentan el desarrollo del chatbot CHARLIE (*CHAtteR Learning Interface Entity*), el cual se utiliza en el prototipo de una plataforma llamada INES. Las herramientas para la construcción de CHARLIE son AIML y *AJAX*. La comunicación se realiza usando lenguaje natural y la interacción se realiza mediante texto procesado en una base de conocimientos.

Bhargava y Maheshwari [171] proponen un sistema educativo de preguntas y respuestas para personas con problemas visuales y parálisis de manos, haciendo uso de lenguaje natural en idioma Inglés. El sistema fue desarrollado con el lenguaje de programación *C#* y utiliza una base de conocimiento en AIML con un vocabulario limitado. La interacción se implementa usando el software *WSR* (*Windows Speech Recognition*). La voz se procesa con métodos y técnicas como la extracción de características, cuantificación vectorial y

cadenas de Markov. Además, se utiliza la conversión de voz a texto de *.NET SL (SpeechLib Library)*, las correcciones de texto y la eliminación de palabras irrelevantes. También se usa dicha biblioteca para transformar texto en voz, que se entrega a través de una interfaz de usuario auditiva.

Gómez Róspide y Puente Águeda [186] presentan un chatbot cuyo dominio de conocimiento es el temario de una asignatura académica, con la finalidad de resolver las dudas de los estudiantes, suponiendo así un refuerzo en su proceso de aprendizaje. El chatbot tiene una base de conocimientos en inglés y no permite idiomas adicionales. Se utilizó el modelo cliente-servidor y MVC (Modelo Vista Controlador). Cuenta con una base de datos con información relacionada con los usuarios del sistema, dominio de conocimiento del chatbot y procedimientos para reconocer los patrones ingresados por el usuario. En la interfaz de usuario, se hace uso de tecnologías Web como *HTML5 (HyperText Markup Language, versión 5)*, *CSS (Cascading Style Sheets)* y *jQuery*. Para la validación de la entrada del usuario, se utiliza *AJAX, jQuery* y *JavaScript*. El servidor Web está basado en *PHP* y en un analizador morfológico conocido como *Flex*.

Niranjan *et al.* [196] propusieron un chatbot que responde a las preguntas de los estudiantes. El chatbot analiza esas preguntas y extrae palabras clave como sustantivos y verbos. Luego, estos términos se buscan en una base de datos y después se elige la categoría con mayor probabilidad de contener la respuesta más adecuada. Finalmente, se selecciona la respuesta y el texto se convierte en voz antes de ser entregado al estudiante. El chatbot está compuesto de los siguientes módulos: 1) *Chat Agent* analiza preguntas y extrae palabras clave como sustantivos y verbos; 2) *Tokenizer* divide la consulta del estudiante en tokens; 3) *Stop words eliminator* es el filtro para palabras comunes y no significativas; 4) *Bayesian* implementa el algoritmo Naive Bayes [240], el cual se utiliza para buscar la respuesta de una pregunta; 5) *Databases* tiene tres bases de datos utilizadas en el sistema, como *Category database, Question database* y *Answer database*; 6) *Text to Speech Converter* convierte texto en voz mediante el motor de voz *Java*; 7) *Tutor* permite el acceso al administrador para modificar las preguntas y respuestas en las bases de datos; y 8) *Student* permite a los estudiantes enviar consultas al sistema y así obtener respuestas.

Mikic-Fonte *et al.* [194] presentan un sistema que funciona como asistente en el proceso de aprendizaje de los estudiantes. El sistema tiene dos componentes principales: una aplicación de Android (el chatbot) y un servidor que tiene los repositorios y el componente

que funciona como recomendador. El chatbot permite al usuario preguntar mediante voz y texto. El objetivo del sistema es capacitar al estudiante para realizar diversas actividades relacionadas con sus procesos de aprendizaje y evaluación, *e.g.*, consultar las preguntas del examen en un repositorio, recibir recomendaciones sobre el material de aprendizaje, hacer preguntas sobre un curso y verificar sus propios exámenes calificados. El desarrollo se basa en AIML para la base de conocimientos y en un componente llamado *Chatterbean*¹⁰ en *Java*.

Ranoliya *et al.* [6] desarrollaron un chatbot usando AIML. El chatbot responde preguntas FAQ. La interacción es textual. Los estudiantes no sólo pueden preguntar sobre la admisión a una institución educativa, también pueden solicitar información sobre la universidad y temas académicos relacionados. Para ello, se creó una plantilla base con saludos, seguido de preguntas generales que el usuario pueda realizar en cualquier momento. El diseño propuesto puede ser replicado por cualquier universidad para responder a las FAQ y así promover un acercamiento con los usuarios. Como trabajo futuro está mejorar el chatbot combinando AIML y *LSA (Latent Semantic Analysis)* [241], de esta forma la puede ser más natural y acorde al lenguaje de los jóvenes.

Dutta [180] propuso un chatbot inteligente para ayudar a los estudiantes de nivel superior a aprender conocimientos generales de las materias. El chatbot está en línea y puede ser accedido por una gran cantidad de usuarios. Para el desarrollo de esta propuesta, se utiliza Dialogflow y técnicas de procesamiento de lenguaje natural para responder a las preguntas de los estudiantes.

Bala *et al.* [170] propusieron un sistema dividido en tres componentes. El primero realiza el registro, tanto de usuarios como de administradores del chatbot. El segundo es el sistema de respuesta de chatbot, donde se utiliza procesamiento de lenguaje natural y se encarga de buscar en la base de conocimientos. El tercer componente es el encargado de dar las respuestas. También se utiliza PSA (*Porter Stemmer Algorithm*) [242] para eliminar los sufijos de las palabras en inglés y después verificar el orden de las palabras en las oraciones. Asimismo se emplea una base de datos léxica en inglés conocida como Wordnet¹¹. Cabe mencionar que se utilizaron *Java*, *Tomcat* y *MySQL* para el desarrollo de esta propuesta.

Paschoal *et al.* [49] presentan el chatbot *Ubibot*. Este chatbot ofrece soporte para

¹⁰<http://www.geocities.ws/phelio/chatterbean/>

¹¹<https://wordnet.princeton.edu/>

cursos de Ingeniería de Software, enfocándose principalmente en la obtención de requisitos, donde asume el rol de elemento interesado en el proyecto de software, más conocido como *stakeholder*. La base de conocimientos de *Ubibot* tiene preguntas que prevalecen entre los estudiantes de la materia y fue implementada con *XML*. *Ubibot* utiliza técnicas de coincidencia de patrones, así como un motor de inferencia que ejecuta *Program-O*.

Mikic-Fonte *et al.* [193] presentan un chatbot como asistente de FAQ para los alumnos que cursan la asignatura de Arquitectura de Computadores en la carrera de Ingeniería en Tecnologías de Telecomunicación de la Universidad de Vigo, España. El chatbot usa AIML y la interacción es por texto. Las respuestas pueden incluir varios elementos, como videos, audios, animaciones y enlaces incrustados en HTML. Además, los profesores pueden administrar el contenido relacionado con las FAQ creando, editando y eliminando elementos de la base de conocimientos.

Mulyana *et al.* [191] presentan el desarrollo de un chatbot llamado *LTKA-Bot*, cuya función principal es facilitar las actividades relacionadas con los cursos, tanto para alumnos como para profesores. *LTKA-Bot* se basa en una arquitectura cliente-servidor y *Hubot* (disponible en la plataforma GitHub). *LTKA-Bot* está configurado para usar los servicios de *Telegram* y aprovechar su interfaz de usuario. Cada componente de la arquitectura está diseñado para utilizar servicios web, ya sea propios o externos.

Benotti *et al.* [176] presentan la plataforma llamada *Chatbot*, que fue diseñada para enseñar a los estudiantes los conceptos de Ciencias de la Computación. Esta propuesta se basa en un trabajo previo de los mismos autores [142]. *Chatbot* se mejoró con técnicas de procesamiento de lenguaje natural, haciendo uso de una biblioteca en el lenguaje de programación *C++* llamada *FreeLing*¹² para proporcionar capacidades de análisis de lenguaje natural, *e.g.*, análisis morfológico, detección de entidades nombradas, etiquetado, análisis, desambiguación del sentido de palabra y etiquetado de roles semánticos. Cabe señalar que *FreeLing* funciona para una variedad de idiomas como inglés, español, portugués, italiano, francés, alemán, ruso, entre otros.

Nguyen *et al.* [195] presentaron una metodología para diseñar un chatbot que resuelve problemas de matemáticas en una escuela de nivel superior. El chatbot actúa como un instructor que brinda consejos para guiar al estudiante sobre cómo resolver algunos problemas matemáticos, específicamente problemas sobre la topografía de una función. La forma en que el chatbot guía al usuario es similar a la forma en que lo haría un profesor.

¹²<http://nlp.lsi.upc.edu/freeling/>

La propuesta se basa en una arquitectura cliente-servidor. Los datos pueden provenir de una computadora o dispositivos móviles conectados a Internet. La información se envía al servidor, luego se reenvía al motor de tutoría y finalmente se presenta el resultado al usuario.

Sreelakshmi *et al.* [203] propusieron un chatbot para responder preguntas y generar cuestionarios. Está dirigido a estudiantes de primaria y secundaria. El funcionamiento del chatbot es el siguiente. El usuario sube un archivo al sistema en formato PDF. Mediante la herramienta *Apache PDFBox*¹³, el chatbot convierte el PDF en texto y limpia los datos usando CoreNLP [243] para alimentar la base de conocimiento. A continuación, se ejecutan los módulos *Respuesta a Pregunta* y *Generación de Cuestionarios*. El primer módulo toma la pregunta del usuario como entrada y utiliza la base de conocimiento para extraer las respuestas más adecuadas y devuelve una respuesta a través del chatbot. El segundo módulo selecciona un conjunto de preguntas disponibles, tomando como referencia los documentos recibidos como insumo.

Kowsher *et al.* [179] presentan a *Doly*, el chatbot bengalí. *Doly* responde a las preguntas de los usuarios en idioma bengalí y se creó con *Python*. También está construido con NLTK (*Natural Language Toolkit*) y emplea técnicas de NLP y ML. El aporte de este estudio es el manejo del BNL (*Bengali Natural Language Processing*) a través de la construcción de herramientas y la aplicación e implementación de algoritmos de clasificación como Naives Bayes.

Ismail y Ade-Ibijola [192] desarrollan un chatbot, llamado *Lecturer's Apprentice*, que puede interactuar con los estudiantes de nivel superior sobre las dificultades en los cursos de programación. Los estudiantes pueden hacer preguntas sobre algoritmos y programación lógica y pueden usar el chatbot para pedir consejos personales que no sean de temas académicos. La interacción es a través de texto y en idioma inglés. El flujo de interacción es el siguiente: el estudiante tiene un problema, le pregunta al chatbot y éste le da consejos relevantes. El chatbot es una aplicación móvil desarrollada con *Android Studio*. Se utiliza la API de IBM Watson, en particular la opción de coincidencia aproximada, la cual permite reconocer errores ortográficos y dar una respuesta correcta. La coincidencia aproximada funciona con un algoritmo de distancia de Levenshtein [244].

Rafael *et al.* [206] implementaron el chatbot *Tribubot*, el cual apoya a docentes y estudiantes para reforzar contenidos académicos de nivel superior, interactuando en lenguaje

¹³<https://pdfbox.apache.org/>

natural. Tribubot utiliza *Facebook Messenger* como interfaz de usuario y una base de conocimiento la cuál fue generada los profesores del curso.

Pereira *et al.* [159] proponen mejorar el aprendizaje y el proceso de evaluación par a par en los MOOC. La interacción es a través de texto y voz. Además, un avatar llamado «Miguel» es el anfitrión, lo cual permite elementos gráficos en la interacción del chatbot con el usuario. La arquitectura se divide en tres bloques principales: 1) la interacción del usuario con el chatbot a través de un teléfono inteligente, 2) la aplicación *Telegram* y 3) el servidor MOOC. La interacción es como sigue: el chatbot hace una pregunta, el usuario proporciona una respuesta y el chatbot continúa la conversación haciendo nuevas preguntas. El usuario puede interactuar con el chatbot mediante comandos. El chatbot utiliza el API de *Telegram* y *MySQL* para almacenar los metadatos asociados con cada grabación de voz. El desarrollo se realizó con tecnologías de código abierto¹⁴. Así, los scripts de notificación se desarrollaron con *PHP* y para la comunicación entre el chatbot y *Telegram*, se utilizó *PHP Telegram Bot*.

Reyes *et al.* [35] desarrollaron un asistente virtual, que permite a los estudiantes pregraduados acceder a contenido interactivo, adaptado para un curso de introducción a la Inteligencia Artificial. El chatbot muestra las FAQ de forma jerárquica, guiando al alumno. La interacción es por texto, voz o entradas táctiles para responder a las preguntas. El chatbot está basado en Dialogflow, pero también se hizo uso de *Google Sheets*, *Node.js* y *JavaScript*. El punto fuerte de esta propuesta es una metodología de dos fases. La primera, llamada *Abstracción de Conocimiento*, se subdivide en recopilación, manipulación y aumento de datos. La recopilación proporciona los datos sin procesar. La manipulación gestiona y clasifica los datos con fines de diseño. La argumentación es un paso adicional para aumentar la cantidad de ejemplos disponibles para el proceso de entrenamiento del modelo de *Machine Learning*. La segunda fase, llamada *Generación de Respuestas*, consiste en definir el procesamiento de las entidades y los intentos para mostrar contenido útil, según el tipo de interacción entre el usuario y el chatbot.

Nias y Ruffin [178] desarrollaron el chatbot **CultureBot** y se propone un robot social humanoide, llamado **Pepper**, en donde se utiliza el chatbot. El objetivo principal de **CultureBot** es ayudar a los estudiantes a aprender tecnología de una manera cómoda. **CultureBot** es capaz de interactuar en inglés estándar y en el idioma AAVE (*African American Vernacular English*), que es una variación del idioma inglés. Para la implemen-

¹⁴<https://bitbucket.org/juananpe/retosmoocsbot/src>

tación de esta propuesta, se utilizó un marco de trabajo llamado *Softbank NAO* junto con la plataforma *Google Cloud Speech*.

Bathija *et al.* [245] propusieron un sistema como herramienta genérica capaz de resumir y analizar múltiples párrafos de contenido educativo y presentarlo al usuario como una conversación interactiva e inteligente a través de una interfaz de chatbot. Los autores propusieron una arquitectura de microservicios. El sistema se divide en tres subsistemas. 1) *User interface-Chatbot*, 2) *Backend* presentado a través de una *API* y 3) *Database*. Los componentes principales son: 1) *Topic Modelling* que recibe múltiples documentos de texto como entrada, los cuales tienen un número fijo de temas y se clasifican para generar las preguntas que se harán al usuario; 2) *Semantic Relationship Modeling*, el cual se utiliza para crear y almacenar un mapa que describa la relación entre oraciones que influyen entre sí para cada tema, por lo que este mapa representa un camino junto con las diversas oraciones y sugiere cómo se debe cubrir el tema cuando se presenta el contenido al usuario a través del chatbot; 3) *Question Generation*, que analiza el contenido y genera las preguntas; 4) *Conversational Flow*, el cual determina la forma en que se presenta la información al usuario y lo ayuda a aprender contenido nuevo; y 5) *Answer Analysis*, que analiza las respuestas almacenadas de cada usuario para realizar un seguimiento de su aprendizaje. Además, se utiliza el modelo LDA para identificar un curso en particular.

Lee *et al.* [188] presentan el chatbot *Infobot*. El chatbot es capaz de responder preguntas sobre los materiales y la logística del curso. También puede fungir como tutor para informar del horario de las clases que llevan los estudiantes, así como difundir sus calificaciones. En esta propuesta destacan las herramientas utilizadas para la implementación: *Dialogflow*, *JavaScript*, *JSON* y *Firebase*. Las interfaces gráficas del chatbot son las plataformas de mensajería: *Telegram*, *Facebook Messenger* y *Line*. El chatbot puede interactuar con el usuario por texto en lenguaje natural y comandos.

Sia *et al.* [209] proponen el chatbot *Abot*. Dado los problemas de salud que pueden presentar los estudiantes por el estrés y presión que llegan a padecer tanto en la escuela como en la casa es necesario que los estudiantes puedan tener un poco de ánimo para mejorar sus hábitos de vida y bienestar, lo cual se reflejará en su salud mental y como consecuencia tendrán un mejor aprovechamiento escolar. Las herramientas que utilizan para desarrollar su chatbot son: *Dialogflow* y *Facebook Messenger*. Para validar *Abot* utilizan una escala Likert con 25 estudiantes que estuvieron utilizando durante una semana el chatbot.

4.3. Contribuciones y características de los chatbots educativos

El Cuadro 4.2 presenta qué autores han abordado el problema de los chatbots en el dominio de la educación y el alcance de sus contribuciones. Además, se exhibe si es posible repetir la propuesta para tener resultados similares. Las columnas del Cuadro 4.2 son:

- **Autores** presenta los nombres de los autores a cargo de la investigación.
- **Contribución** indica el principal aporte de la propuesta.
- **Área** muestra el terreno en el cuál la propuesta tiene relevancia: Ingeniería de Software (IS), Inteligencia Artificial (IA), Educación (EDU) e Interacción Humano-Computadora (IHC).
- **Reproducibilidad** tiene los siguientes criterios: Bajo (B), Medio (M) y Alto (A). Entendiendo la reproducibilidad como la capacidad de un ensayo o experimento de ser replicado por otros, especialmente dentro de la comunidad científica [246].

La columna de autores está ordenada en orden cronológico para mostrar la continuidad de las investigaciones, si es que ésta existe. Además, en el SMS descrito en el Capítulo 3, se estableció como punto de inicio el año 2006 porque a partir de esa fecha empezó a hacerse común el uso de herramientas de chatbots en la Web. Las contribuciones de las propuestas no son exclusivamente para mejorar los chatbots. Si bien, los chatbots son importantes en esta revisión, también lo son los mecanismos, métodos y pasos que se llevaron a cabo, teniendo como dominio la educación. El Cuadro 4.2 muestra las principales áreas involucradas en las investigaciones, *e.g.*, aquellos trabajos que presentan en detalle la arquitectura de los chatbots están contribuyendo a la Ingeniería de Software. La Figura 4.2 muestra la amplia gama de aplicaciones posibles de los chatbots, sólo en el área de la educación, *e.g.*, desde chatbots para concursos escolares, escuelas públicas y privadas, cursos de un tema específico, aprendizaje en línea, FAQ para fines generales, plataformas como Moodle, prototipos, software, estudiantes y docentes.

Para que un trabajo sea reproducible, se verificó que las tecnologías y herramientas utilizadas estén actualizadas, documentadas y respaldadas, *e.g.*, Gómez Róspide y Puente [186] usan *PHP*, *AJAX* y *JavaScript* en su propuesta. Desde hace varios años, estas

Cuadro 4.2: Resumen de las contribuciones del estado del arte en chatbots educativos.

Autores	Contribución	Área	Reproducibilidad
T-Bot & Q-Bot [44]	Construcción de chatbots para LMS y mecanismo para incorporarlos.	IA, EDU	A
Bigham <i>et al.</i> [172]	Soporte de chatbot para estudiantes con discapacidades.	IA, EDU	B
CHARLIE [45]	Prototipo funcional de chatbot para aprendizaje en línea.	IA, EDU, IS	B - M
INES [153]	Elementos semánticos que pueden ser utilizados en aplicaciones Web y agentes de software inteligentes.	IA, EDU	M
Bhargava & Maheshwari [171]	Reconocimiento de voz, recuperación de conocimientos y respuesta de interés para personas con discapacidad.	EDU	B
Gómez Róspide & Puente [186]	Incorporación de elementos gráficos que cambian según el contenido de la conversación (avatar).	IHC	A
Niranjan <i>et al.</i> [196]	Integración de algoritmos de probabilidad bayesianos	IA, EDU	B
UMT-BOT [46]	Herramienta de apoyo a estudiantes universitarios destacando disponibilidad y usabilidad.	IA, IS	B
Chatbot [142]	Construcción de software educativo.	EDU, IHC, IS	B
NLAST [194]	Objetos de aprendizaje.	EDU	M
Ranoliya <i>et al.</i> [6]	Diseño de chatbot eficiente.	IA	M
Dutta [180]	Comparación de diferentes plataformas Web para crear chatbots.	IA	M
Bala <i>et al.</i> [170]	Herramienta de software para apoyar la prestación de servicios de calidad y la satisfacción del cliente.	IA, IS	M - A
MALL [155]	Mejora e integración de ALICE en un sistema software.	EDU, IS	B - M
Song <i>et al.</i> [163]	Diseño y desarrollo de chatbots de apoyo a la educación.	EDU, IS	A
FIT-EBot [47]	Clasificación de textos y extracción de entidades.	IA, EDU, IS	M
LiSA [48]	Estrategias de diseño de chatbots.	IS	M
DINA [22]	Pasos para aplicar Dialogflow en combinación con Ontologías.	IA, IS	M - A
Ubibot [49]	Características técnicas y funcionamiento para la construcción de un chatbot.	IA, EDU	A
Mikić-Fonte <i>et al.</i> [193]	Manejo mejorado del lenguaje natural.	IA	B
English Practice [148]	Construcción e integración de un chatbot en un sistema de software educativo.	EDU, IS	B
LTKA-Bot [191]	Diseño e implementación de un chatbot para soporte de cursos.	IS	B
EASelective [129]	Diseño e implementación de un chatbot para la resolución de problemas de asesoramiento académico y selección de cursos.	IA, EDU	A
Chatbot [176]	Mejora de la construcción de software educativo, trabajo previo [142].	IA, EDU	B
EduBot [146]	Respuestas contextuales relevantes y construcción de una base de conocimiento.	IA, EDU	B
Clarizia <i>et al.</i> [143]	Mejora de las plataformas de aprendizaje en línea.	IA, EDU, IHC	M
Nguyen <i>et al.</i> [195]	Resolución de problemas matemáticos y construcción de una base de conocimiento.	IA, EDU	B
Ikastenbot [198]	Herramienta para apoyar a los estudiantes en la detección de errores al redactar documentos formales.	IA, EDU, IS	A
Goschlberger & Brandstetter [151]	Diseño de un chatbot para plataformas de aprendizaje en línea.	IA, IS	M
QuizBot [201]	Chatbot para apoyar al estudiante en el aprendizaje.	IA, EDU, IHC	B
Sreelakshmi <i>et al.</i> [203]	Mecanismo para extraer preguntas de archivos en una base de conocimiento para evaluar a los estudiantes.	IA, EDU, IS	A
Lee <i>et al.</i> [132]	Herramienta de software para optimizar las tareas administrativas.	IS	B - M
Doly [179]	Uso de técnicas de Machine Learning para mejorar BLPN.	IA, EDU	M
KEMTbot [130]	Uso de tecnología actual para construir chatbots.	IA, EDU, IS	A
Oliveira <i>et al.</i> [158]	Metodología para integrar chatbots en VLE.	IA, EDU, IS	M
Lecturer's Apprentice [192]	Prototipo de chatbot amigable y fácil de interactuar y uso de algoritmos de Machine Learning.	IA, EDU, IHC, IS	M
Tribubot [206]	Chatbot para mejorar el proceso de aprendizaje del estudiante.	EDU	B
CISA [126]	Diseño de un chatbot inclusivo.	IHC, IS	B
Pereira <i>et al.</i> [159]	Un nuevo método para la evaluación de ensayos MOOC P2P mediante grabación de voz.	EDU, IS	A
Reyes <i>et al.</i> [35]	Metodología genérica para la creación de chatbots educativos.	EDU, IS	A
Mekni <i>et al.</i> [133]	Chatbot con información relevante y pruebas con diferentes tecnologías para su construcción.	IS	M - A
CultureBot [178]	Integración de un chatbot con el robot Pepper.	IS	B
Bathija <i>et al.</i> [245]	Herramienta de software genérica capaz de resumir y analizar múltiples párrafos.	IA, EDU, IHC, IS	B - M
E-orientation [128]	Integración de elementos pedagógicos en chatbots.	EDU	B
TutorDocente [139]	Enfoque para el diseño de chatbots.	EDU, IS	M
Infobot [188]	Estrategias de diseño de chatbots.	EDU, IS	B
STUART [144]	Apoyo al proceso de enseñanza-aprendizaje.	EDU, IHC	B
Abot [209]	Diseño de un chatbot para apoyo en la salud y bienestar de estudiantes.	Salud	B

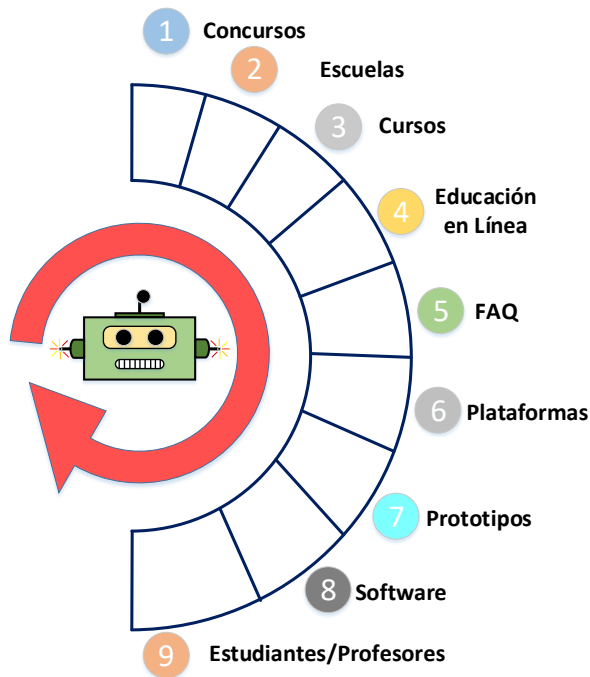


Figura 4.2: Abanico de posibilidades para realizar investigación en chatbots educativos.

herramientas se han posicionado en el mercado con buenos resultados en diferentes sectores. Son populares, simples y fáciles de usar, además de ser flexibles y versátiles. También, existe abundante documentación en la Web, sin olvidar el gran número de comunidades para desarrolladores. Por otro lado, los autores mencionan la arquitectura utilizada. Por lo tanto, existe una alta probabilidad de que el trabajo pueda ser reproducible. Otro ejemplo es la propuesta de Pereira *et al.* [159], quienes no sólo usaron *MySQL*, *PHP* y *Telegram Bot*, sino también compartieron el código fuente y permitieron su distribución y uso. En consecuencia, ésto aumenta en gran medida la reproducibilidad de su obra. Por el contrario, un trabajo que no es reproducible en la actualidad es el de Bhargava y Maheshwari [171], quienes utilizaron *Windows Vista/7*, pero estos sistemas operativos ya están discontinuados. Si se quiere replicar el estudio, no sería factible lograrlo en las mismas condiciones porque no se tendría acceso a estas herramientas, aún cuando se puedan obtener por otros medios. Por tanto, los resultados obtenidos no serían adecuados ni correctos.

Es importante mencionar que la columna de reproducibilidad puede tomar un valor

combinado. Un caso interesante es MALL [155], en el que se emplea el chatbot ALICE, pero se tiene un elemento incompleto para poder reproducirlo. El sitio Web oficial del chatbot ALICE no tiene ningún menú, liga o información para descargar el chatbot. Si se busca por otros medios, como en la plataforma Github, se pueden encontrar varios proyectos basados en el chatbot ALICE, en los que dicho chatbot ha sido modificado para propósitos específicos. Todas estas cuestiones no permiten que la propuesta sea completamente reproducible. Como resultado, MALL [155] tiene una reproducibilidad baja-media.

El Cuadro 4.3 presenta un resumen de las características de cada chatbot, destacando la interacción del avatar, las capacidades multilingües, el mecanismo de interacción, el desarrollo y las propuestas del sistema. Este análisis es importante porque permite identificar específicamente cuáles son los puntos que se pueden mejorar en las propuestas. Las columnas del Cuadro 4.3 son:

- **Avatar** muestra las propuestas que han utilizado elementos gráficos, con el fin de fomentar el acercamiento, la empatía o la confianza en el usuario.
- **Multilinguaje** exhibe si los trabajos permiten la interacción usando diferentes lenguajes.
- **Mecanismo de interacción** presenta la vía de interacción de los chatbots: texto, voz o ambos. Para indicar que se trata de entrada, se usa la letra ‘E’ y para la salida, se usa ‘S’.
- **Desarrollo** indica si el chatbot se implementó desde cero o si el chatbot puede ser una continuación del trabajo previo de los mismos autores o de otros.
- **Sistema** exhibe si la propuesta es un sistema de software y uno de sus componentes es el chatbot.

Como se puede observar en el Cuadro 4.3, existen pocos trabajos en los que se incluyen avatares. De acuerdo con Hernández-León [50], es importante tener un avatar o imagen animada que refleje, de cierta manera, el estado emocional del chatbot ya que esto podría tener un impacto positivo en el experiencia de usuario; además, los avatares permiten un mejor acercamiento con el usuario y debería ser un elemento fundamental para todo chatbot. Compartimos este punto de vista porque si introducen elementos gráficos que permitan algún tipo de identificación con el usuario, esto ayudará a conseguir una mayor

Cuadro 4.3: Resumen de las características de chatbot educativos.

Autores	Avatar	Multilinguaje	Mecanismo de interacción		Desarrollo		Sistema
			Texto	Voz	Nuevo	Legado	
T-Bot & Q-Bot [44]	×	✓	E/S	×	✓	×	×
Bigham <i>et al.</i> [172]	×	×	E/S	×	✓	×	×
CHARLIE [45]	×	×	E/S	×	✓	×	×
INES [153]	×	×	E/S	×	✓	×	✓
Bhargava & Maheshwari [171]	×	×	×	E/S	✓	×	✓
Gómez Róspide & Puente [186]	✓	×	E/S	×	✓	×	×
Niranjan <i>et al.</i> [196]	×	×	E/S	S	✓	×	×
UMT-BOT [46]	×	×	E/S	×	✓	×	×
Chatbot [142]	✓	×	E/S	E/S	×	✓	×
NLAST [194]	×	×	E/S	E/S	✓	×	×
Ranoliya <i>et al.</i> [6]	×	×	E/S	×	✓	×	×
Dutta [180]	×	×	E/S	×	✓	×	×
Bala <i>et al.</i> [170]	×	×	E/S	×	✓	×	×
MALL [155]	×	×	E/S	×	×	✓	×
Song <i>et al.</i> [163]	×	×	E/S	×	✓	×	×
FIT-EBot [47]	×	×	E/S	×	✓	×	×
LiSA [48]	✓	×	E/S	×	✓	×	×
DINA [22]	×	×	E/S	×	✓	×	×
Ubibot [49]	✓	×	E/S	×	✓	×	×
Mikic-Fonte <i>et al.</i> [193]	×	×	E/S	×	✓	×	×
English Practice [148]	×	×	E/S	E/S	✓	×	×
LTKA-Bot [191]	×	×	E/S	×	✓	×	×
EASElective [129]	×	×	E/S	×	✓	×	×
Chatbot [176]	✓	✓	E/S	×	×	✓	×
EduBot [146]	×	×	E/S	×	✓	×	×
Clarizia <i>et al.</i> [143]	×	×	E/S	×	✓	×	×
Nguyen <i>et al.</i> [195]	×	×	E/S	×	✓	×	×
Ikastenbot [198]	×	×	E/S	×	✓	×	×
Goschlberger & Brandstetter [151]	×	×	E/S	×	✓	×	×
QuizBot [201]	×	×	E/S	×	✓	×	×
Sreelakshmi <i>et al.</i> [203]	×	×	E/S	×	✓	×	×
Lee <i>et al.</i> [132]	×	×	E/S	×	✓	×	×
Doly [179]	×	×	E/S	×	✓	×	×
KEMTbot [130]	×	✓	E/S	E/S	✓	×	×
Oliveira <i>et al.</i> [158]	×	×	E/S	×	✓	×	×
Lecturer's Apprentice [192]	×	×	E/S	×	✓	×	×
Tribubot [206]	×	×	E/S	×	✓	×	×
CiSA [126]	×	×	E/S	×	✓	×	×
Pereira <i>et al.</i> [159]	✓	×	E/S	E/S	✓	×	×
Reyes <i>et al.</i> [35]	×	×	E/S	E/S	✓	×	×
Mekni <i>et al.</i> [133]	×	×	E/S	×	✓	×	×
CultureBot [178]	✓	×	×	E/S	✓	×	×
Bathija <i>et al.</i> [245]	×	×	E/S	×	✓	×	✓
E-orientation [128]	×	×	E/S	×	✓	×	×
TutorDocente [139]	×	×	E/S	S	✓	×	×
Infobot [188]	×	×	E/S	×	✓	×	×
STUART [144]	✓	×	E/S	E/S	✓	×	✓
Abot [209]	×	×	E/S	×	✓	×	×

empatía con el chatbot. Sin embargo, Gómez Róspide y Puente [186] exponen que incluso cuando se utilizó un avatar, el chatbot no logró generar empatía. En consecuencia, la introducción de un avatar sin un diseño adecuado puede generar una experiencia de usuario negativa. Por lo tanto, debe haber una investigación previa sobre el tipo de población a la que se dirige el chatbot. Esto permitiría, entre otras cosas, que el chatbot pudiera tener una comunicación más fluida y natural, *e.g.*, **CultureBot** [178] propone realizar una comunicación más coloquial para tener un mejor acercamiento con la población objetivo.

Según Gelbukh y Bolshakov [247] el procesamiento de lenguaje natural no es una tarea fácil, porque al momento de procesar los textos una computadora no los entiende. La computadora sólo lee cadenas de texto que no tienen ningún sentido. Además, el procesamiento de lenguaje natural tiene que resolver algunos problemas como: ambigüedad, complejidad del conocimiento lingüístico o conocimiento extralingüístico.

Si se puede acceder a un chatbot desde la Web, podría ser significativo interactuar en varios idiomas. A lo largo de los años, las traducciones de un idioma a otro han tenido grandes avances, pero para entender el texto a traducir es necesario tener una representación formal y luego generar el texto en otro idioma [247].

Sólo dos estudios consideraron el uso de chatbots multilingües, *i.e.*, que pueden interactuar en diferentes idiomas, además del idioma nativo de la propuesta. Ambos T-Bot & Q-Bot [44] y Chatbot [176] usaron *FreeLing*, sin embargo las propuestas no muestran muchos progresos. Aquí, se encuentra una oportunidad para que los chatbots puedan interactuar con los usuarios en diferentes idiomas. Además, una considerable cantidad de estudios revisados se basan en Dialogflow¹⁵. La documentación de Dialogflow permite la comunicación por texto y voz. Por lo tanto, existe la posibilidad de que esta propuesta también ofrezca interacción de voz, aunque algunos autores que utilizaron Dialogflow no especificaron si ésta era una alternativa.

El mecanismo de interacción principal es a través de texto, tanto de entrada como de salida. Sin embargo, en tiempos en los que la multimodalidad en los dispositivos es común, la interacción multimodal debería ser una tendencia en trabajos futuros. Cabe señalar que no se consideran mecanismos de interacción como vídeo y gestos.

El desarrollo de chatbots nuevos es la opción que tomaron la mayoría de los autores, dejando de lado trabajos previos de chatbots. Sin embargo, si el chatbot es nuevo, no significa que sea original. Es importante mencionar que hay pocas propuestas en las que

¹⁵<https://dialogflow.com/>

existe una relación directa entre los chatbots existentes. Por ejemplo, **Chatbot** [142], **MALL** [155] y **Chatbot** [176] toman trabajos anteriores, pero como puede apreciarse en el Cuadro 4.3, la tendencia es empezar el desarrollo desde cero.

Pocas propuestas fueron pensadas como un sistema de software grande y complejo. Este tipo de proyectos, como puede verse, no fue demasiado favorecido. No hay una razón específica para esto, pero se puede asumir lo siguiente. Al principio, los requerimientos eran más complejos que los mencionados en las propuestas. Posteriormente, los autores intentaron abarcar muchos aspectos para anticiparse a los problemas que se presenten a largo plazo. Además, el sistema de software requiere pruebas específicas para cada componente.

Cabe señalar que el chatbot **ALICE** fue planeado como un proyecto de código abierto, cuyo propósito era el desarrollo de capacidades de los chatbots, utilizando el lenguaje de programación Java. Dada su importancia y carácter reutilizable, las propuestas **T-Bot & Q-Bot** [44], **CHARLIE** [45], **Chatbot** [142] y **MALL** [155] tomaron el código fuente de **ALICE** y lo adaptaron para crear sus propios chatbots o para integrarlo en una plataforma web.

Finalmente, es notable encontrar que no se considera la interacción de vídeo y gestos. Yang *et al.* [248] argumenta que los gestos son métodos importantes de comunicación y pueden ayudar a expresar, de manera efectiva, las demandas de los usuarios. Sin embargo, las ventajas de una interacción por texto es el anonimato del usuario. De hecho, éste es uno de los beneficios de un chatbot es el anonimato [111]. Con el uso de vídeo, esto no es posible y puede darse el caso de que el usuario hable de cosas sensibles, delicadas o críticas. Este aspecto sería una interesante investigación futura.

4.4. Aspectos técnicos de los chatbots educativos

El Cuadro 4.4 muestra un resumen de los aspectos técnicos de los chatbots, destacando técnicas de Inteligencia Artificial (IA), arquitecturas, almacenamiento y herramientas. De esta forma, se pueden ver en orden cronológico las herramientas que más se utilizaron y cuál podría ser la tendencia. Las columnas del Cuadro 4.4 son:

- **Mecanismo de IA** exhibe el enfoque utilizado para proporcionar capacidades de razonamiento e inteligencia a los chatbots.
- **Arquitectura** presenta la arquitectura empleada para diseñar los chatbots.

- **Almacenamiento** muestra si se utilizó una Base de Conocimiento (BC), una Base de Datos (BD) o ambas.
- **Herramientas** presenta los instrumentos seleccionados para la implementación de los chatbots.

En el Cuadro 4.4 se exponen los diferentes mecanismos de IA que se utilizaron para la construcción de chatbots educativos. En la Figura 4.3, se puede apreciar el número de veces que fueron empleadas, aunque se omitieron las que se utilizaron sólo una vez. Se tomaron aquellas que aparecen en dos o más trabajos porque se considera que es donde más se puede avanzar para lograr, de manera rápida, la incorporación de mecanismos inteligentes en un chatbot.

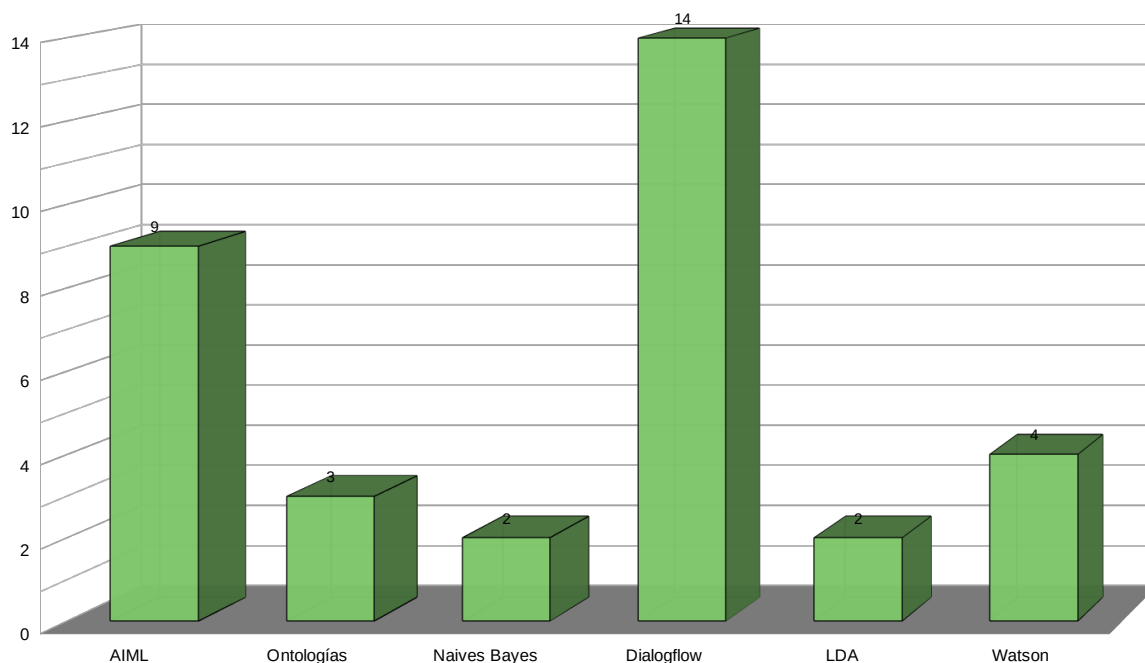


Figura 4.3: Mecanismos de IA utilizados en las propuestas.

Los mecanismos de IA son diversos, sin embargo predominan dos: AIML y Dialogflow. Para una introducción a AIML, Sukarsa *et al.* [249] explican en qué consiste. Un *script* AIML se compone de diferentes etiquetas: 1) *AIML* indica el principio y el final del documento; y 2) *category* siempre tiene dos etiquetas, *pattern* y *template*, donde la primera

Cuadro 4.4: Aspectos técnicos de los chatbots.

Autores	Mecanismo de IA	Arquitectura	Almacenamiento		Herramientas	
			BC	BD	Implementación	Web/Móvil
T-Bot & Q-Bot [44]	AIML		✓	✓	ALICE, Program E, Freeling1.5, AJAX, Caroline, Moodle, PHP MySQL	
Bigham <i>et al.</i> [172]					C#, JAWS, Window-Eyes, Windows Live Messenger ZoomText	
CHARLIE [45]	AIML		✓		ALICE, Program D	AJAX
INES [153]	Ontologías	Componentes	✓		XML	
Bhargava & Maheshwari [171]	AIML				C#, .NET SL, WSR	
Gómez Róspide & Puente [186]		Cliente-Servidor, MVC	✓	✓	Flex	AJAX, CSS, HTML5, jQuery, JavaScript, PHP
Niranjan <i>et al.</i> [196]	Naives Bayes			✓	Java	
UMT-BOT [46]	AIML		✓		Appservy 2013, Program-O	PHP
Chatbot [142]					ALICE	
NLAST [194]	AIML	Cliente-Servidor	✓	✓	Chatterbean, Java	Android
Ranoliya <i>et al.</i> [6]	AIML, LSA					
Dutta [180]	Dialogflow		✓			Android, BotUI, HTML5, JavaScript, Node.js
Bala <i>et al.</i> [170]	PSA	Cliente-Servidor	✓	✓	Java, Tomcat, Wordnet, MySQL	
MALL [155]	AIML	Cliente-Servidor			Java, XML	PHP, JSON
Song <i>et al.</i> [163]		Cliente-Servidor		✓	MongoDB, Typescript	Angular, Node.js, Socket.IO
FIT-EBot [47]	Dialogflow, Named Entity Recognition (NER)	Componentes	✓	✓	MySQL	Facebook Messenger
LiSA [48]	Chatfuel			✓		Facebook Messenger
DINA [22]	Dialogflow, Ontologías	Modular	✓	✓		
Ubibot [49]	AIML		✓		Program-O, XML	
Mikic-Fonte <i>et al.</i> [193]	AIML		✓			HTML
English Practice [148]	Dialogflow			✓		Android, Datastore, Firebase, Google Cloud
LTKA-Bot [191]		Cliente-Servidor, Microservicios			Hubot	Google Docs, Telegram
EASElective [129]	Watson	Cliente-Servidor		✓	MongoDB, MySQL	JSON, Node.js, PHP, RESTful API
Chatbot [176]					C++, FreeLing	
EduBot [146]	QnA Maker		✓	✓	MATLAB	Canvas LMS, IMS-Global, LTI, PHP
Clarizia <i>et al.</i> [143]	LDA, Ontologías, Petri Nets			✓		
Nguyen <i>et al.</i> [195]	Heuristics, Forward Chaining	Cliente-Servidor	✓			
Ikastenbot [198]	Vision AI (Cloud Vision), TensorFlow	Cliente-Servidor			CSV, LanguageTool, Regex/Grep, Text-Matcher	Re- Google Cloud Vision, JSON, Telegram, TensorFlow
Goschlberger & Brandstetter [151]	Dialogflow	Microservicios	✓	✓	Elasticsearch, Postgres	RESTful API
QuizBot [201]	GloVe, SIF					Facebook Messenger, Flask
Sreelakshmi <i>et al.</i> [203]	CoreNLP		✓		Apache PDFBox, Java	
Lee <i>et al.</i> [132]	Dialogflow				Python	KakaoTalk
Doly [179]	Naive Bayes				Python, NLTK	
KEMTbot [130]	Dialogflow			✓		AWS, Firebase, JavaScript, JSON
Oliveira <i>et al.</i> [158]	Watson	Microservicios		✓		CSS, Facebook Messenger, HTML, JavaScript, Moodle, PHP
Lecturer's Apprentice [192]	Levenshtein's Distance, Watson	Cliente-Servidor			XML	Android
Tribubot [206]		Capas	✓			Facebook Messenger
CISA [126]	Dialogflow			✓		Facebook Messenger
Pereira <i>et al.</i> [159]		Componentes		✓	MySQL	PHP, Telegram Bot
Reyes <i>et al.</i> [35]	Dialogflow		✓	✓		Google Sheets, JavaScript, Node.js
Mekni <i>et al.</i> [133]	Dialogflow	Capas		✓	MySQL, EJS	Facebook Messenger, Node.js
CultureBot [178]					Pepper Robot, Softbank NAO	
Bathija <i>et al.</i> [245]	LDA	Microservicios				
E-orientation [128]	Dialogflow					Facebook
TutorDocente [139]	Watson	Capas	✓		Postgres	IBM Cloud, Node.js
Infobot [188]	Dialogflow		✓	✓		Firebase, Line, Node.js, Facebook Messenger, Telegram
STUART [144]	Dialogflow	Capas		✓	Python	Flask, Google Translator
Abot [209]	Dialogflow					Facebook Messenger

es una cadena de texto que debe coincidir con los tipos de usuario y la segunda especifica la respuesta que coincide con *pattern*. Por otro lado, Dialogflow funciona con agentes virtuales. Un agente maneja la conversación con los usuarios finales, utilizando los siguientes componentes [250]: 1) *intent* clasifica la intención del usuario final para un turno de conversación; 2) *entity* tiene una clave y valores que indican cómo extraer los datos; 3) *actions and parameters* es para filtrar intentos y generar un formato de consulta; y 4) *fulfillment* es una pieza de código con propósitos específicos. AIML no proporciona ningún mecanismo de aprendizaje ni un motor de inferencia; Gómez Róspide y Puente [186] deciden no utilizarlo porque no cumplía con los requisitos que los autores esperaban. No obstante, existe una competencia anual de chatbots llamada Premio Loebner, en la que a los chatbots participantes se les aplica la Prueba de Turing. El chatbot *Mitsuku*, creado por Steve Worswick utilizando AIML, ha ganado cinco veces (en 2013, 2016, 2017, 2018, 2019). Por otro lado, Nuruzzaman y Hussain [63] declaran que Dialogflow tiene limitaciones.

Todos los chatbots deben tener una arquitectura. Mediante una buena arquitectura, se logran atributos de calidad. Algunas propuestas mencionan qué arquitecturas implementaron y si fue la adecuada para sus necesidades. De esta manera, la arquitectura brinda beneficios en el mantenimiento de chatbots, *e.g.*, nuevos mecanismos de inteligencia e interacción. En el Cuadro 4.4 se pueden observar las diferentes arquitecturas que los autores utilizaron para la implementación de sus propuestas. Si bien cada modelo arquitectónico permite abstraer un concepto de diseño que puede ser utilizado para el desarrollo, eso no implica que la arquitectura establecida en esos trabajos sea de carácter universal, debido a los diferentes requerimientos que manejan cada uno de ellos. Las arquitecturas mencionadas en las propuestas son: cliente-servidor, componentes, modular, capas, microservicios y modelo vista controlador (MVC).

El almacenamiento permite conservar información relevante para los chatbots. El diseño que elijan los autores para los mecanismos de inferencia será crucial y vital para decidir si utilizar bases de conocimiento, bases de datos o ambas. Las bases de conocimiento van más de la mano con la IA y contienen información sobre pregunta-respuesta o soluciones a problemas. En cambio, las bases de datos son estructuras grandes y complejas que permiten obtener y almacenar datos de manera eficiente [251]. Estos aspectos son extremadamente importantes en las propuestas por las siguientes razones. En primer lugar, tener una base de conocimiento requiere un análisis de la información que se va a guardar y luego procesar a través de un conjunto de reglas. En segundo lugar, el conocimiento

estará sujeto al mantenimiento de la base de conocimiento. En tercer lugar, la base de datos se utiliza principalmente para almacenar datos que no representan conocimiento, *e.g.*, alias, contraseña y correo electrónico del usuario.

Las herramientas son elementos importantes y se puede considerar que las habilidades de los desarrolladores que manejan estos elementos pueden ser un factor de gran impacto para el desarrollo de chatbots.

Las propuestas que se revisaron tienen diferentes enfoques. Como se puede observar en el Cuadro 4.4, éstas no necesariamente cubren toda la información de las columnas. Algunas de ellas se centran principalmente en describir el proceso de desarrollo del chatbot. Otros, como el trabajo de Ranoliya *et al.* [6] y DINA [22] muestran más información sobre los mecanismos para dotar de inteligencia a los chatbots y la arquitectura que se utilizó, pero no especifican las herramientas empleadas para la implementación. Algo similar se observa en los trabajos de Bala *et al.* [170] y Tribubot [206], los cuales no describen el mecanismo de inferencia, en cambio se enfocan en explicar las herramientas de implementación.

También, la arquitectura que se utilizó más para la implementación de chatbots es: cliente-servidor. Se encontró que, en cada implementación, los componentes principales son similares. Lo único que cambia es cómo se implementa la interacción (texto o voz) y el enfoque o dominio del chatbot.

4.5. Discusión

En esta revisión se pudo conocer, de manera general, cómo funcionan los chatbots, las posibles clasificaciones e incluso las diferentes arquitecturas de software. Una pregunta que se puede plantear es: ¿Cómo se espera que sea el chatbot educativo ideal? Sin embargo, esta es una pregunta difícil de responder. Primero, se debe ver cómo sería el chatbot ideal, en general, en vez de centrarse primeramente en aquellos chatbots que caen en un dominio particular, *e.g.*, en la educación. Como primera aproximación a una respuesta, se tiene que Baby [32] espera que los chatbots ideales, en términos generales, sean aquellos que tengan la habilidad de comprender el contexto de una conversación, aprender y mejorar con el tiempo.

Teniendo en mente esta idea, primero es necesario adaptarla al contexto educativo. Así, un chatbot educativo ideal depende de las necesidades de cada comunidad porque, aunque sea para el ámbito académico, puede haber necesidades diferentes. No obstante,

en esta revisión, no se encontró un chatbot con todas estas capacidades ideales, pero los avances en tecnología y computación están alimentando cada vez más investigaciones para lograrlas.

Los chatbots en general han sido clasificados de diferentes maneras: alto nivel [73] y bajo nivel [63, 74]. Sin embargo, cada grupo de clasificación responde a necesidades específicas. La clasificación de alto nivel tiene en cuenta los chatbots orientados a tareas y no orientados a tareas. La clasificación de bajo nivel considera el dominio del conocimiento, el servicio prestado, los objetivos, el método de generación de respuestas, la ayuda humana, los permisos y el canal de comunicación.

Una arquitectura general para un chatbot educativo debe ser una arquitectura de alto nivel, donde los elementos generales están definidos, pero los detalles específicos de la arquitectura no están completamente especificados, pues su implementación depende de las funcionalidades requeridas por el chatbot.

El futuro puede ser incierto, pero lo que se podría esperar es que los chatbots evolucionen para implementar mecanismos que automáticamente enriquezcan el contenido o incluso produzcan la respuesta del docente a los ejercicios de los estudiantes [159].

Los desafíos de las propuestas revisadas han sido diversos, empezando principalmente por la tecnología y las herramientas que permiten que los chatbots sean inteligentes. Lograr una interacción entre el chatbot y el usuario, utilizando lenguaje natural, es un trabajo en desarrollo que ha durado décadas. Sin embargo, se observa que, en las propuestas, los enfoques pueden volverse específicos, *e.g.*, mejora de los mecanismos para incorporar chatbots en plataformas de aprendizaje en línea, desarrollo de estrategias de diseño de chatbots, adaptación de técnicas de ML para chatbots, así como el desarrollo de chatbots para servicios de apoyo a los estudiantes en sus cursos escolares.

Por otro lado, las posibilidades de explorar nuevas líneas de investigación en chatbots educativos pueden ampliarse aún más, en comparación con lo presentado en la Figura 4.2. Sin embargo, aunque el interés personal puede ser una fuente de motivación importante, existen otros factores que también deben considerarse para las nuevas líneas de investigación: relevancia, visto desde la importancia y utilidad para la comunidad escolar en general; viabilidad, en cuanto a su factibilidad para llevar a cabo el proyecto de investigación, incluyendo el acceso a los recursos necesarios (financieros, humanos o tecnológicos) y la disponibilidad de datos y fuentes de información; originalidad, en el ámbito de ser una propuesta única y aportar algo nuevo al campo de estudio correspondiente; y con-

tribución, referente a la capacidad de colaborar al desarrollo y avance de la disciplina o campo de estudio correspondiente.

Finalmente, la Figura 4.4 presenta posibles oportunidades de líneas de investigación las cuales se explican a continuación:

1. Trámites de servicios escolares, *e.g.*, credencial de estudiante, inscripción, constancia de estudios o tira de materias.
2. Inducción para estudiantes y profesores de nuevo ingreso con el propósito de brindarles la información necesaria para integrarse exitosamente a la institución educativa.
3. Propedéutico para estudiantes que requieran prepararse para un nivel de estudios más avanzado.
4. Programa de atención tutorial para combatir el rezago académico y deserción escolar.
5. Seguimiento para la elaboración de documentos formales, con el fin de cumplir con los requisitos de un plan de estudios, *e.g.*, problema eje, informe académico o de trabajo, proyecto terminal, reporte de investigación o tesis.
6. Seguimiento al trabajo académico de los estudiantes por parte de los padres de familia.
7. Apoyo a sistemas semiescolarizados en los cuales los estudiantes tienen una sesión presencial semanal y el resto del tiempo trabajan de manera independiente.
8. Seguimiento a los trámites de certificación o finalización de estudios para asegurarse de que se completen correctamente y en tiempo y forma.
9. Apoyo a comisiones evaluadoras de una institución para asegurar un proceso de evaluación efectivo y eficiente.
10. Asistir eventos de puertas abiertas con la finalidad de dar a conocer el trabajo que se realiza.
11. Implementaciones basadas en enfoques pedagógicos como el constructivismo, el conductivismo, el método tradicional, el Montessori, el Waldorf, entre otros.

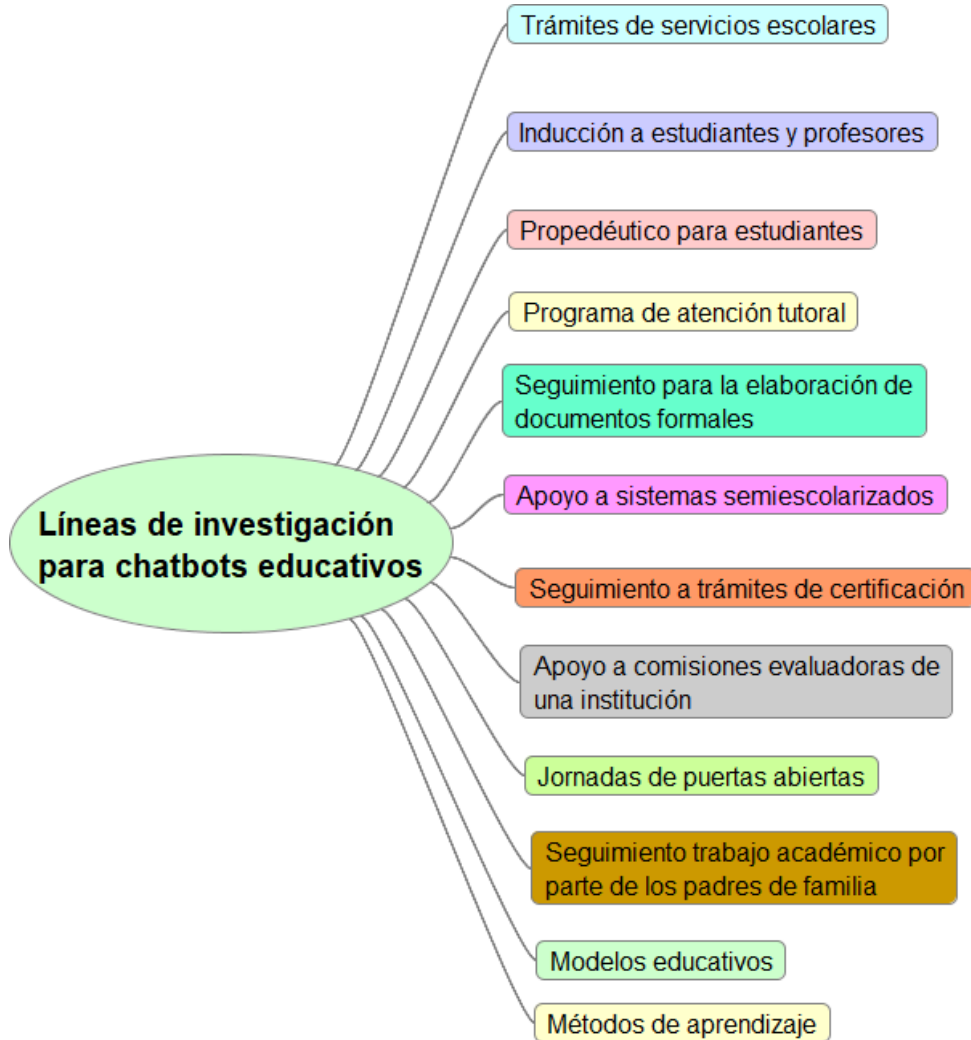


Figura 4.4: Líneas de investigación para futuras propuestas de chatbots educativos.

12. Integración de métodos de aprendizaje para crear una experiencia de aprendizaje más completa y efectiva.

Capítulo 5

Arquitectura genérica de chatbots educativos

En este capítulo se presenta la aplicación y desarrollo del método denominado Diseño Guiado por Atributos (ADD 3.0), descrito en el Capítulo 2, para definir la primera versión de la arquitectura genérica de chatbots educativos. Primeramente, se recapitula y se esboza cómo podría ser la arquitectura de un chatbot educativo, de acuerdo al estudio del estado del arte, sin utilizar ningún mecanismo para el diseño de la arquitectura (cf. Sección 5.1). Después, se describen los términos, pasos y elementos generales de ADD 3.0 (cf. Sección 5.2), así como las entradas que requiere el método para iniciar (cf. Sección 5.3). Finalmente, se presenta el proceso de diseño y diagramas, así como los detalles de cada uno de los pasos para obtener la arquitectura genérica de chatbots educativos (cf. Sección 5.4.1).

5.1. Análisis preliminar

De acuerdo a lo que se ha revisado en el estado del arte, los componentes principales de cualquier chatbot son el mecanismo de interacción, el análisis de entrada-salida, el motor de razonamiento y la base de conocimientos. Se considera que un acercamiento de arquitectura de software apropiada para chatbots educativos podría ser la de cliente-servidor, aunque con algunas mejoras. Cabe destacar que, en esta propuesta, uno de los retos principales es el diseño de la arquitectura genérica de chatbots educativos. Por lo tanto, se hace una propuesta inicial de cómo podría ser dicha arquitectura genérica, sin

entrar en detalles específicos. La Figura 5.1 presenta una aproximación de arquitectura cliente-servidor para chatbot educativo, la cual es el resultado del análisis que se ha hecho del estado del arte, sin la necesidad de utilizar algún proceso de diseño.

Se tiene un usuario que usa una PC o dispositivo móvil, en el lado del cliente, para enviar un mensaje de texto o de voz por medio de un navegador web. En el lado del servidor hay un *Componente Chatbot* que tiene dos módulos, por los cuales pasa el mensaje del usuario: *Analizador E/S* y *Motor de Razonamiento*. El módulo *Analizador E/S* envía el mensaje analizado al módulo *Motor de Razonamiento*, el cual tiene las reglas para procesar la información y realizar las operaciones necesarias con la información que tiene la *Base de Conocimientos* o la *Base de Datos*. La primera alimenta con inteligencia al chatbot y la segunda tiene los datos relacionados con los usuarios. Del análisis de los aspectos técnicos de los chatbots educativos (cf. Sección 4.4) presentados capítulo precedente, se observa que el uso de servicios como Amazon Lex, Chatfuel, Dialogflow y Watson, está en aumento y probablemente estos servicios serán una tendencia (sino es que ya lo son) para la construcción de chatbots. Por lo tanto, se requiere una capa más en la arquitectura para tener en cuenta estos mecanismos de razonamiento.

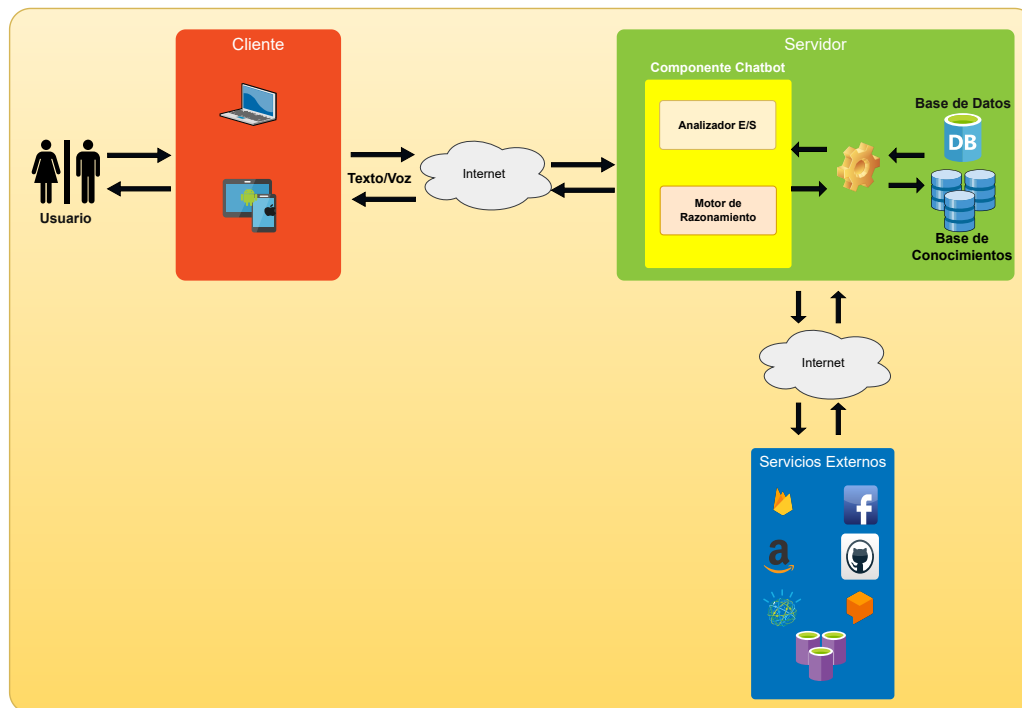


Figura 5.1: Primera aproximación de una arquitectura genérica de chatbots educativos.

5.2. Método ADD 3.0

Para el desarrollo de la propuesta de arquitectura genérica de chatbots educativos, se eligió ADD 3.0 por las siguientes razones. ADD 3.0 es un método enfocado totalmente en el diseño de la arquitectura de software. Además, no omite detalles en pasos, actividades e iteraciones, como puede suceder con las otras propuestas que no guían el diseño de la manera adecuada. También, ADD 3.0 se puede utilizar como complemento para fortalecer el diseño de otras propuestas similares. Asimismo, es más ligero que un proceso o marco de trabajo. Finalmente, promueve de manera explícita, la reutilización de arquitecturas de referencia y se combina con un catálogo de conceptos de diseño, que incluye una amplia selección de tácticas, patrones, marcos, arquitecturas de referencia y tecnologías.

Antes de comenzar, es necesario conocer algunos conceptos importantes que se manejan en ADD 3.0 [107]:

- **Propósito de diseño:** se refiere a lo que se pretende lograr con respecto al diseño de la arquitectura para dominios nuevos o existentes.
- **Atributos de calidad:** se definen como propiedades medibles o comprobables de un sistema software, que se utilizan para indicar qué tan bien el sistema satisface las necesidades de las partes interesadas.
- **Funcionalidad primaria:** es la capacidad del sistema para hacer el trabajo para el que fue diseñado.
- **Intereses arquitectónicos:** abarcan aspectos adicionales que deben considerarse como parte del diseño arquitectónico, pero que no son expresados como requerimientos tradicionales.
- **Restricciones:** son decisiones sobre las que se tiene poco o ningún control.
- **Arquitecturas de referencia:** son planos que proporcionan una estructura lógica general para tipos específicos de aplicaciones.
- **Patrones de diseño arquitectónico:** son soluciones conceptuales a problemas de diseño recurrentes que existen en un contexto definido.
- **Patrones de despliegue:** proporcionan modelos sobre cómo estructurar físicamente el sistema para implementarlo.

ADD 3.0 es un método enfocado totalmente en el diseño de la arquitectura de software y consta de siete pasos:

1. **Revisar las entradas.** Antes de iniciar este paso, es necesario contar con una serie de directrices¹ que sirven de entrada y que son el propósito del diseño, los requisitos funcionales primarios, los escenarios de atributos de calidad, las restricciones y los intereses arquitectónicos.
2. **Establecer la meta de iteración² seleccionando las directrices.** Se formaliza el objetivo de la iteración antes de iniciar una iteración de diseño particular.
3. **Elegir uno o más elementos del sistema a refinar.** Se seleccionan los elementos que contribuyen en la satisfacción de las directrices específicas.
4. **Elegir uno o más conceptos de diseño que satisfagan las directrices seleccionadas.** Requiere identificar alternativas entre los conceptos de diseño que puedan ser utilizados para lograr la meta de la iteración; posteriormente se hace una selección de las alternativas.
5. **Crear instancias de elementos arquitectónicos, asignar responsabilidades y definir interfaces.** Los elementos a instanciar necesitan estar conectados permitiendo colaboración uno con el otro. También, se necesita asignar las responsabilidades que tiene cada uno de ellos. Además, se requiere definir interfaces³ para el intercambio de información entre ellos.
6. **Vistas de sketch y registro de las decisiones de diseño.** Se genera documentación preliminar de las vistas y registro de las decisiones importantes en la iteración, lo que facilitará el análisis y entendimiento.
7. **Realizar análisis del diseño actual y revisar el objetivo de la iteración y el logro del propósito del diseño.** Una vez analizado el diseño se revisa el estado de la arquitectura en términos del cumplimiento del propósito de diseño.

¹Son un conjunto de requerimientos que se consideran para el desarrollo de un sistema de software y toman relevancia en el diseño de la arquitectura de software. En inglés se utiliza el término *drivers* [252].

²Una iteración se refiere a seguir los pasos del método hasta lograr el propósito establecido.

³La interfaz es una especificación contractual de cómo la información debería fluir entre los elementos.

ADD 3.0 es un método iterativo, por lo tanto, una vez finalizado el paso 7 se debe iterar, si es necesario, repitiendo del paso 2 al 7 para cada directriz que fue considerada como parte de la entrada.

Cabe mencionar que, de acuerdo a Cervantes y Kazman [107], puede haber ocasiones en que se deba invertir el orden de los pasos 2 y 3, obligando a iniciar primero con el paso 3 y luego a continuar con el paso 2. Esto puede ocurrir porque hay casos en los que al diseñar un sistema *greenfield*⁴ o al desarrollar ciertos tipos de arquitecturas de referencia; al menos en las primeras etapas de diseño, el diseñador se centrará en los elementos del sistema y comenzará la iteración, seleccionando un elemento en particular y luego se considerarán los controladores que desean abordar.

Por otro lado, en ADD 3.0 se menciona que el diseño de un sistema de software tiene tres tipos de propósitos:

- Diseño de sistemas *greenfield* para un dominio maduro, *i.e.*, la arquitectura se construye desde cero y el dominio, la infraestructura, herramientas, tecnologías y el conocimiento base son elementos conocidos.
- Diseño de sistemas *greenfield* para nuevos dominios, *i.e.*, estos son sistemas desafiantes y complicados, a diferencia de un dominio maduro. Por lo tanto, puede ser que las referencias arquitectónicas sean pocas o no existan, teniendo como consecuencia que en el dominio se cuente con una infraestructura menos establecida, así como con poco conocimiento base.
- Diseño para un sistema existente *brownfield*⁵, *i.e.*, ya existe una arquitectura para el sistema. Entonces, se pretende dar mantenimiento y, también hacer cambios en él.

El beneficio de utilizar ADD 3.0 [107] para el desarrollo de la presente propuesta es la guía paso a paso, de manera sistemática, con detalles para lograr una arquitectura genérica de chatbots educativos. Además, ADD 3.0 provee un catálogo de conceptos de diseño, los cuales abarcan referencias arquitectónicas, patrones de despliegue, patrones de diseño y tácticas. No obstante, un inconveniente de ADD 3.0 es cómo dar un seguimiento

⁴Se refiere cuando se trabaja desde cero, *i.e.*, un campo sin construcciones ni restricciones previas.

⁵Indica que se diseña a partir de un trabajo previo, por lo que habrá restricciones debido a las dependencias con otros sistemas de software.

adecuado al progreso del diseño. Para esto, Cervantes y Kazman [107] sugieren que se utilice un tablero Kanban⁶ como herramienta para solventar esa debilidad.

5.3. Entradas

Antes de iniciar ADD 3.0, es esencial disponer de un conjunto de directrices que sirvan como punto de partida para el método. Algunas de estas directrices se derivan de los resultados obtenidos en la Sección 4.1.2. Sin embargo, en ese momento, estos resultados no estaban expresados de manera ideal para su inclusión en el método. En esta sección, se realizó un análisis para adaptarlos adecuadamente como las entradas requeridas por ADD 3.0.

Las directrices que se utilizaron son las siguientes:

1. Propósito de diseño:

- El objetivo principal es establecer con claridad el propósito de este diseño, el cuál se pretende lograr siguiendo uno de estos tres enfoques: sistemas *greenfield* para un dominio maduro, sistemas *greenfield* para un nuevo dominio o sistemas existentes conocidos como *brownfield*.

2. Modelo de casos de uso para describir las interacciones entre el chatbot educativo y los usuarios:

- Para desarrollar estos casos de uso, hemos tenido en cuenta la propuesta de clasificación de chatbots educativos que distingue entre aquellos orientados a servicios escolares y orientados a estudiantes/profesores, como se muestra en la Figura 4.1. Dado que esta clasificación es de suma importancia, resulta esencial que se refleje claramente como una directriz en la etapa inicial del método, especialmente en la definición de los casos de uso.

3. Escenarios de atributos de calidad priorizados que describen la respuesta del sistema ante estímulos específicos:

- Estos escenarios se derivan tomando en cuenta atributos de calidad y es fundamental asegurarse de que los atributos propuestos no entren en conflicto entre

⁶Los tableros Kanban son herramientas que se utilizan en las gestiones de desarrollos ágiles.

sí o, en la medida de lo posible, no afecten el diseño genérico. Es primordial garantizar que estos atributos de calidad no introduzcan complejidad innecesaria y que sean fáciles de comprender, hay que destacar que cumplir en su totalidad con todos los atributos de calidad es poco probable. En su lugar, es esencial encontrar un equilibrio que refleje las necesidades reales de los usuarios de la mejor manera posible. El objetivo no es alcanzar una calidad perfecta, sino una calidad necesaria y suficiente para que se pueda establecer una arquitectura de chatbot para un dominio educativo. En general, puede ser más efectivo concentrarse en un número reducido de atributos de calidad clave al diseñar una arquitectura para esta primera etapa. Esto facilitará la comprensión del sistema y garantizará que se satisfagan los requisitos más importantes. Para lograrlo, se presentará un catálogo de atributos de calidad, y a partir de un análisis correspondiente, se priorizarán los atributos de calidad más esenciales con sus respectivos escenarios.

4. **Restricciones** presentes para el desarrollo:

- Las restricciones son aspectos sobre los cuales el arquitecto tiene poco o ningún control. Como parte del proceso de diseño arquitectónico, es esencial catalogar estas restricciones de desarrollo, que pueden manifestarse en forma de tecnologías necesarias, requisitos de interoperabilidad con otros sistemas, cumplimiento de leyes, normas y regulaciones, disponibilidad de recursos humanos, entre otros aspectos. Para este proyecto, se han establecido restricciones para asegurar que la arquitectura resultante esté adecuadamente ordenada con el contexto educativo. Para lograrlo, es fundamental considerar desde el principio tanto la definición de un chatbot educativo como su alineación o aproximación a un modelo educativo.

5. **Intereses arquitectónicos:**

- Se deben considerar como parte del diseño arquitectónico pero no son como los requisitos tradicionales. Para esto hay varios tipos de intereses: generales, específicos, internos e incidencias (*issues*). Nosotros nos enfocamos en el interés arquitectónico del tipo general, porque el propósito principal de este tipo es establecer la estructura del chatbot educativo, entre otras cosas.

5.3.1. Propósito del diseño

El propósito de este diseño en esta primera etapa consiste en un tipo de sistema *green-field* relativamente para nuevos dominios:

- Producir un diseño esencial que sea simple y contenga detalles generales sobre los componentes necesarios para un chatbot educativo, con el propósito de servir como guía para la construcción de un prototipo.

5.3.2. Modelo de casos de uso

Un caso de uso es una forma de describir actividades y acciones que se llevarán a cabo en un sistema de software. Se pueden representar con un diagrama que tiene como propósito mostrar de forma visual las finalidades esenciales [253].

Para esta directriz logramos identificar un conjunto de doce casos de uso destinados a los usuarios *estudiante*, que desempeña, el rol de alumno en la institución educativa, y *profesor*, quien representa al personal docente de la institución. Estos casos de uso incluyen funciones como consultar horarios, obtener información general, responder a preguntas frecuentes, mostrar procedimientos generales, llevar a cabo evaluaciones, proporcionar detalles sobre asignaturas, responder preguntas y ofrecer respuestas, brindar retroalimentación, abordar cuestiones de salud física y mental, prestar soporte técnico para conectar dispositivos a la red inalámbrica del laboratorio, mostrar el progreso del estudiante y proporcionar apoyo en tutorías.

La Figura 5.2 presenta estos casos de uso. Tenemos como primer caso de uso CU-0 el cuál es necesario para iniciar sesión. Los casos de uso CU-1 hasta CU-10 y CU-12 están destinados a la interacción con el usuario *estudiante*, mientras que los casos de uso CU-1 hasta CU-4 y CU-11 se relacionan con el usuario *profesor*. La distinción radica en que los casos de uso CU-1 a CU-4 son de interés común tanto para el estudiante como para el profesor, en contraste con los casos de uso CU-5 a CU-10 y CU-12, que están específicamente diseñados para el estudiante. Por último, el caso de uso CU-11, es exclusivo para el profesor.

Para asegurar que estos casos de uso son apropiados para iniciar el proceso de diseño y crear la arquitectura genérica, es posible verificarlos fácilmente en función de la clasificación de chatbots educativos que se presenta en la Figura 4.1:

- Orientados a Servicios Escolares:

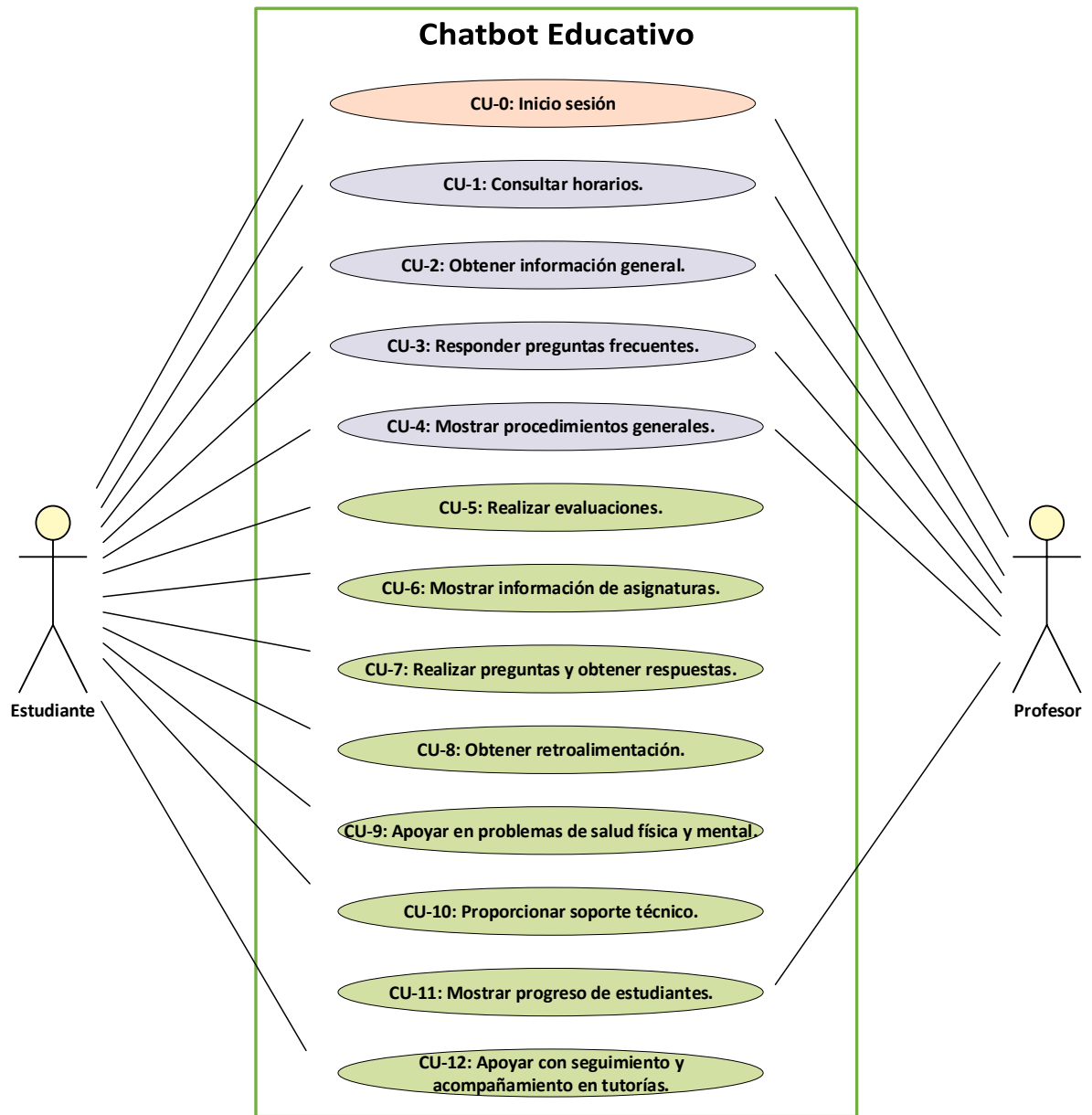


Figura 5.2: Modelo de casos de uso del chatbot educativo propuesto.

Cuadro 5.1: Descripción de los casos de uso del chatbot educativo.

ID	Caso de Uso	Descripción
CU-0	Inicio de sesión	Los usuarios ingresan al chatbot mediante un usuario y contraseña.
CU-1	Consultar horarios	Un usuario <i>estudiante</i> o <i>profesor</i> pueden consultar los horarios disponibles del periodo escolar en curso, eventos académicos, periodo de exámenes o vacaciones.
CU-2	Obtener información general	Un usuario <i>estudiante</i> o <i>profesor</i> obtiene información general sobre la historia de la institución, la oferta educativa, el directorio y los planes de estudio.
CU-3	Responder FAQ	Un usuario <i>estudiante</i> o <i>profesor</i> pueden hacer preguntas y obtener respuestas. El tipo de preguntas y respuestas de este caso de uso es normalmente conocido como FAQ, <i>e.g.</i> , horarios de laboratorios, biblioteca o servicio médico, entre otros.
CU-4	Mostrar procedimientos generales	Un usuario <i>estudiante</i> o <i>profesor</i> requiere una guía con información específica para realizar trámites, <i>e.g.</i> , inscribirse en un periodo escolar, grupo, clase, idioma o actividades extracurriculares, tramitar la credencial de estudiante, solicitar préstamos de libros o material didáctico u obtener el certificado escolar.
CU-5	Realizar evaluaciones	Un usuario <i>estudiante</i> puede realizar exámenes o entregar prácticas, tareas, cuestionarios y ensayos.
CU-6	Mostrar información de asignaturas	Un usuario <i>estudiante</i> puede obtener información sobre las materias que debe llevar durante el semestre, <i>e.g.</i> , temario, objetivos y evaluaciones.
CU-7	Realizar preguntas y obtener respuestas	Un usuario <i>estudiante</i> puede realizar una pregunta concreta y obtener una respuesta específica.
CU-8	Obtener retroalimentación	Un usuario <i>estudiante</i> puede obtener retroalimentación de acuerdo a su progreso.
CU-9	Apoyar en problemas de salud física y mental	Un usuario <i>estudiante</i> puede solicitar la canalización con un psicólogo para tratar algún problema.
CU-10	Proporcionar soporte técnico	Un usuario <i>estudiante</i> puede recibir soporte, <i>e.g.</i> , cómo conectar un dispositivo a la red inalámbrica del laboratorio.
CU-11	Mostrar el progreso de un estudiante	Un usuario <i>profesor</i> puede solicitar información sobre el progreso de un estudiante.
CU-12	Apoyar con seguimiento y acompañamiento en tutorías	Un usuario <i>estudiante</i> puede recibir orientación educativa o personal.

- C&H → CU-1
 - Información → CU-2
 - FAQ → CU-3
 - Procedimientos → CU-4
- Orientados a Estudiante/Profesor:
 - Evaluación → CU-5
 - Materias → CU-6
 - P&R → CU-7
 - Retroalimentación → CU-8
 - Salud Bienestar → CU-9
 - Soporte → CU-10
 - Reportes → CU-11
 - Tutorías → CU-12

El Cuadro 5.1 proporciona un desglose completo de estos casos de uso. La columna **ID** muestra el identificador del caso de uso, la columna **Caso de Uso** describe el escenario de uso, y la columna **Descripción** brinda una explicación detallada del caso de uso correspondiente.

5.3.3. Escenarios de atributos de calidad

Los atributos de calidad son elementos fundamentales en el diseño de la arquitectura. Como hemos mencionado previamente, estos atributos se definen como propiedades medibles o verificables de un sistema de software, que indican qué tan bien el sistema cumple con las necesidades de las partes interesadas. Además, es relevante destacar la existencia de los escenarios de atributos de calidad, que son breves descripciones de cómo el sistema debe responder ante estímulos específicos [107].

Debemos señalar que existen diversas listas de atributos de calidad, ya que su relevancia depende del dominio de la aplicación y de los objetivos de negocio específicos. Sin embargo, podemos identificar aquellos que suelen ser ampliamente utilizados. El Cuadro 5.2 presenta un catálogo de estos atributos que suelen emplearse en el diseño de sistemas [86].

Cuadro 5.2: Catálogo de atributos de calidad comúnmente empleados en el proceso de diseño.

Atributo de Calidad	Descripción
Disponibilidad	Es la propiedad del sistema de software de estar apto y listo para llevar a cabo su tarea en el momento que se solicite.
Interoperabilidad	Se refiere al grado en que dos o más sistemas pueden intercambiar información útil a través de interfaces en un contexto determinado.
Modificabilidad	Tiene que ver con el cambio, y se centra en el costo y el riesgo de realizar cambios.
Desempeño	Capacidad del sistema software para cumplir los requisitos de tiempo.
Seguridad	Es una manera de medir la capacidad del sistema para proteger los datos y la información de acceso no autorizado, teniendo en cuenta el acceso a la información de los usuarios y a los sistemas que están autorizados
Capacidad de prueba	Se refiere a la capacidad del sistema para revelar sus fallos de manera efectiva cuando se somete a un conjunto de pruebas.
Usabilidad	Se refiere a la facilidad con la que los usuarios pueden realizar operaciones dentro del sistema.
Variabilidad	Es una forma especial de modificabilidad y se refiere a la capacidad de un sistema y sus artefactos de apoyo, como los requisitos, los planes de pruebas y las especificaciones de configuración, para soportar la producción de un conjunto de variantes que difieren entre sí de forma planificada previamente.
Portabilidad	Es una forma especial de modificabilidad y se refiere a la facilidad con la que un software que fue creado para funcionar en una plataforma puede modificarse para funcionar en otra diferente.
Escalabilidad	La horizontal consiste en añadir más recursos a unidades lógicas, como añadir otro servidor a un clúster de servidores. La vertical se refiere a añadir más recursos a una unidad física, como añadir más memoria a una sola computadora.
Mantenibilidad	Es el grado de eficacia y eficiencia con que un producto o sistema puede ser modificado por los encargados de su mantenimiento

Por otro lado, es importante revisar que no exista un conflicto potencial entre los atributos de calidad, ya que esto puede tener un impacto en el diseño de la arquitectura [254]. Por ejemplo, el rendimiento y la mantenibilidad pueden entrar en conflicto, ya que los componentes grandes pueden mejorar el rendimiento, pero los componentes de grano fino pueden mejorar la mantenibilidad [253].

Entre otros conflictos, podemos mencionar aquellos que involucran a la modificabilidad, interoperabilidad, usabilidad y desempeño en relación con la seguridad. La modificabilidad puede aumentar la complejidad del sistema, lo que a su vez puede dificultar la seguridad. La interoperabilidad requiere compartir información con otros sistemas, lo que puede aumentar el riesgo de seguridad. La usabilidad implica hacer que el sistema sea más accesible, lo que también puede complicar la seguridad. Por último, la búsqueda de un

alto desempeño puede llevar a reducir las capas de seguridad, lo que potencialmente compromete la seguridad del sistema. Por lo tanto, lo mejor es priorizar los atributos de calidad [255] para esta primera etapa de diseño.

A su vez, un aspecto que está ligado con la elaboración de los escenarios de atributos de calidad, son los involucrados, mejor conocidos como *stakeholders*. Un *stakeholder* pueden ser cualquier persona, *e.g.*, individuos, grupos, organizaciones o entidades que tienen un interés en el éxito o el fracaso de un proyecto o en la operación de una empresa u organización, pero a su vez pueden incluir a clientes, empleados, proveedores, accionistas, socios comerciales, reguladores gubernamentales y otros [86, 107].

Para la elaboración los escenarios de atributos de calidad, resulta fundamental contar con la participación de los *stakeholders*. No obstante, en esta primera etapa del desarrollo, no están disponibles. En tales circunstancias, se vuelve imperativo tomar decisiones acerca de cómo abordar y priorizar los numerosos desafíos que el sistema enfrenta. Siguiendo las recomendaciones de Cervantes y Kazman [107], una técnica útil en estas situaciones es la creación de un Árbol de Utilidad. Esta técnica funciona de la siguiente manera: en una hoja de papel, se escribe la palabra *utilidad* como nodo principal. Luego, se ponen los diferentes atributos de calidad que componen la utilidad para el sistema. A continuación, se describe el aspecto del atributo de calidad que más preocupa en relación con ese sistema. Finalmente, en las hojas del árbol se expresan los escenarios que ilustran las preocupaciones específicas de cada atributo de calidad.

Para mayor practicidad, los escenarios de calidad del Árbol de Utilidad se presentan en el Cuadro 5.3. La columna *ID* corresponde al identificador del atributo de calidad, *Atributo de Calidad* presenta el atributo elegido, *Escenario* describe la respuesta del sistema ante un estímulo, *Caso de Uso Asociado* indica a qué caso de uso alcanza el atributo de calidad y *Prioridad* es una clasificación en dos dimensiones de acuerdo a la importancia de negocio/riesgo técnico con los valores de ‘Alto’, ‘Medio’ y ‘Bajo’.

Como se puede observar, se han planteado un total de 17 escenarios de atributos de calidad. Estos escenarios abarcan aspectos generales y en algunos casos un poco técnicos. La tarea de elaborarlos y priorizarlos no es una labor sencilla, especialmente cuando no se dispone de la participación de los *stakeholders*. Sin embargo, la priorización para esta primera etapa debe centrarse en aquellos escenarios que puedan ayudar a generar una estructura con los componentes básicos para un chatbot educativo.

Para esta primera etapa seleccionaremos cuatro atributos de calidad como base: dis-

Cuadro 5.3: Escenarios de atributos de calidad para un chatbot educativo.

ID	Atributo de Calidad	Escenario	Caso de Uso Asociado	Prioridad
AC-1	Desempeño	Un estudiante utiliza el chatbot educativo para obtener ayuda con un problema de una materia en específico. El chatbot debe proporcionar respuestas rápidas y precisas a las preguntas en un tiempo de respuesta inferior a 3 segundos para facilitar el aprendizaje sin interrupciones.	CU-7	(M, B)
AC-2	Desempeño	Varios estudiantes realizan una evaluación en línea a través del chatbot educativo. El chatbot debe procesar las respuestas de los estudiantes de manera eficiente y mostrar los resultados en tiempo real, asegurando una experiencia de evaluación sin demoras significativas.	CU-5	(M, M)
AC-3	Disponibilidad	Los estudiantes que estudian fuera de las instalaciones educativas dependen del chatbot educativo para obtener ayuda en sus deberes escolares. El chatbot debe estar disponible las 24 horas del día, los 7 días de la semana, para brindar asistencia continua y apoyar la educación.	Todos	(A, A)
AC-4	Disponibilidad	Los estudiantes utilizan el chatbot educativo para estudiar para sus exámenes. El chatbot debe mantenerse operativo en todo momento, permitiendo a los estudiantes estudiar y practicar en cualquier momento que les resulte conveniente.	CU-7, 8	(A, M)
AC-5	Disponibilidad	En el transcurso de una interacción normal con el chatbot, se presenta una pérdida de conexión a Internet. El chatbot notifica al usuario sobre la falta de conectividad.	Todos	(M, M)
AC-6	Interoperabilidad	El chatbot educativo se utiliza en un entorno de educación en línea que incluye diversas plataformas. El chatbot debe ser compatible e interoperable con estas plataformas para permitir una experiencia de aprendizaje fluida y la transferencia de datos entre sistemas.	Todos	(B, A)
AC-7	Portabilidad	Los estudiantes pueden acceder al chatbot desde una variedad de dispositivos, como computadoras de escritorio, tabletas y teléfonos inteligentes. El chatbot debe ser interoperable en diferentes plataformas y dispositivos, garantizando una experiencia uniforme sin importar cómo los estudiantes elijan acceder a él.	Todos	(A, A)
AC-8	Mantenibilidad	Ante los problemas técnicos o errores en el chatbot, el equipo de soporte técnico debe poder realizar reparaciones y actualizaciones de manera rápida y efectiva. El chatbot debe ser mantenible para facilitar la resolución de problemas técnicos de manera eficiente.	Todos	(M, M)
AC-9	Modificabilidad	Personal educativo desea personalizar el contenido de aprendizaje proporcionado por el chatbot. El chatbot debe ser fácilmente modificable, permitiendo agregar, editar o eliminar contenido de manera sencilla y rápida.	CU-5, 6, 7	(M, M)
AC-10	Portabilidad	El chatbot educativo se implementa en varias instituciones educativas con diferentes sistemas y configuraciones tecnológicas. Debe ser fácilmente portable, lo que permite su despliegue sin problemas en diferentes entornos educativos sin requerir modificaciones significativas.	Todos	(B, A)
AC-11	Portabilidad	La plataforma tecnológica en la que se ejecuta el chatbot educativo se actualiza periódicamente. El chatbot debe ser portátil en estas nuevas versiones de plataforma, lo que facilita las actualizaciones sin interrupciones en el servicio.	Todas	(B, M)
AC-12	Seguridad	Los estudiantes proporcionan información personal al chatbot educativo. El chatbot debe garantizar la seguridad de estos datos y proteger la información de los estudiantes contra accesos no autorizados.	CU-5, 8, 9, 10	(A, A)
AC-13	Seguridad	Un usuario desconocido sin registro trata de ingresar en la pantalla de inicio de sesión en un entorno normal de operación. El sistema debe impedir el acceso y mostrar un mensaje que solo estudiantes y profesores registrados pueden ingresar.	CU-0	(A, M)
AC-14	Seguridad	Un estudiante solicita información referente a su progreso en una asignatura. El chatbot debe mostrar la información referente únicamente a ese usuario de forma resumida.	CU-8	(M, M)
AC-15	Seguridad	Un profesor requiere un reporte del progreso de uno o varios estudiantes. El chatbot debe presentar únicamente la información relevante de cada estudiante al profesor.	CU-11	(M, M)
AC-16	Seguridad	Un estudiante recurre al chatbot para solicitar orientación, ya sea de naturaleza educativa, personal o sobre temas sensibles. El chatbot debe proporcionar una canalización precisa para asegurar que el estudiante reciba la atención adecuada.	CU-12	(A, M)
AC-17	Usabilidad	Un estudiante utiliza el chatbot para reforzar su aprendizaje en temas educativos o informativos. El chatbot debe ofrecer una navegación intuitiva y permitir al estudiante encontrar fácilmente las respuestas a sus preguntas.	CU-8	(A, A)

ponibilidad, interoperabilidad, seguridad y usabilidad. Por lo tanto, los escenarios que se eligen son aquellos que tienen como prioridad una combinación de (A, A), (A, M) y (M, A): AC-3, AC-4, AC-7, AC-12, AC-13, AC-16 y AC-17.

Cuadro 5.4: Restricciones para el diseño del chatbot educativo.

ID	Restricción
RES-1	El chatbot puede ser accedido a través un navegador web en diferentes sistemas operativos.
RES-2	Se deben emplear servicios de la nube, que faciliten el procesamiento del lenguaje natural y que estén enfocados al desarrollo de chatbots.
RES-3	Se deben utilizar servicios de almacenamiento y base de datos (no relacional) desde la nube.
RES-4	Para el desarrollo se deberá tener un conocimiento básico del uso de los servicios que se ofrecen en la nube.
RES-5	Los usuarios del chatbot deberán registrar una cuenta para autenticarse al iniciar sesión en el chatbot educativo.
RES-6	El chatbot debe tener soporte para un mínimo de 15 usuarios simultáneos.
RES-7	El chatbot debe hacer un análisis del lenguaje, pues los mensajes del usuario pueden contener palabras que generen violencia, odio u acoso.
RES-8	El chatbot debe tener algún aspecto pedagógico. Por ejemplo, se puede adoptar un enfoque estímulo-respuesta o de refuerzo para facilitar el proceso de aprendizaje.
RES-9	Considerar la definición chatbot educativo propuesta en la Subsubsección 3.7.1.

5.3.4. Restricciones

Las restricciones en este desarrollo de arquitectura son limitaciones que deben considerarse, ya que pueden tanto facilitar como dificultar el éxito del proyecto. El Cuadro 5.4, presenta una serie de restricciones que deben ser consideradas al seguir los pasos de ADD 3.0:

- RES-1 se deriva de la naturaleza común de los chatbots, ya que éstos suelen estar disponibles en línea y accesibles a través de navegadores web en diversos sistemas operativos.
- RES-2 se origina por las ventajas que pueden proporcionar los servicios de procesamiento de lenguaje natural diseñados para el desarrollo de chatbots. Estos servicios ofrecen numerosos beneficios cuando se acceden a través de la nube, aunque es importante señalar que la gran mayoría suelen ser de pago y pueden tener limitaciones en cuanto a funcionalidad o acceso temporal, *e.g.*, Dialogflow o Watson.
- RES-3 surge como una extensión de la restricción RES-2, ya que al utilizar servicios

en la nube, se abre la posibilidad de una integración más fluida con servicios de almacenamiento y bases de datos.

- RES-4 es inmediata por las restricciones RES-2 y RES-3.
- RES-5 se plantea como medida de control de usuarios.
- RES-6 expresado para ser utilizado en las primeras pruebas.
- RES-7 se deriva para prevenir el uso inapropiado del lenguaje en un inicio y así evitar problemas mayores en un futuro.
- RES-8 denota que un modelo educativo da pauta a los posibles mecanismos de interacción que se deben incluir en el chatbot.
- RES-9 es una restricción esencial para el diseño.

5.3.5. Intereses arquitectónicos

Las preocupaciones o intereses arquitectónicos son aquellos factores a considerar en el diseño, ya que actúan como guías. El interés arquitectónico inicial que se ha tenido en cuenta para esta etapa se presenta en el Cuadro 5.5.

Cuadro 5.5: Interés arquitectónico del chatbot educativo.

ID	Interés
IA-1	Establecer una estructura general inicial de alto nivel.

5.4. Proceso de diseño

Si exploramos a profundidad en el diseño, podríamos descubrir debates sobre si el diseño de la arquitectura de software puede ser considerado un arte o simplemente una disciplina técnica. Aunque el diseño implica la aplicación de principios y conceptos técnicos, también involucra elementos de creatividad, intuición y toma de decisiones. En nuestro enfoque hacia la creación de una arquitectura genérica, consideramos que ambas dimensiones son fundamentales y se complementan mutuamente, siempre y cuando se siga un enfoque metódico.

En esta sección, presentamos el proceso de diseño para desarrollar una arquitectura genérica destinada a un chatbot educativo. El diagrama de flujo que ilustra este proceso que seguimos se presenta en la Figura 5.3 el cuál abordamos en la siguientes subsecciones.

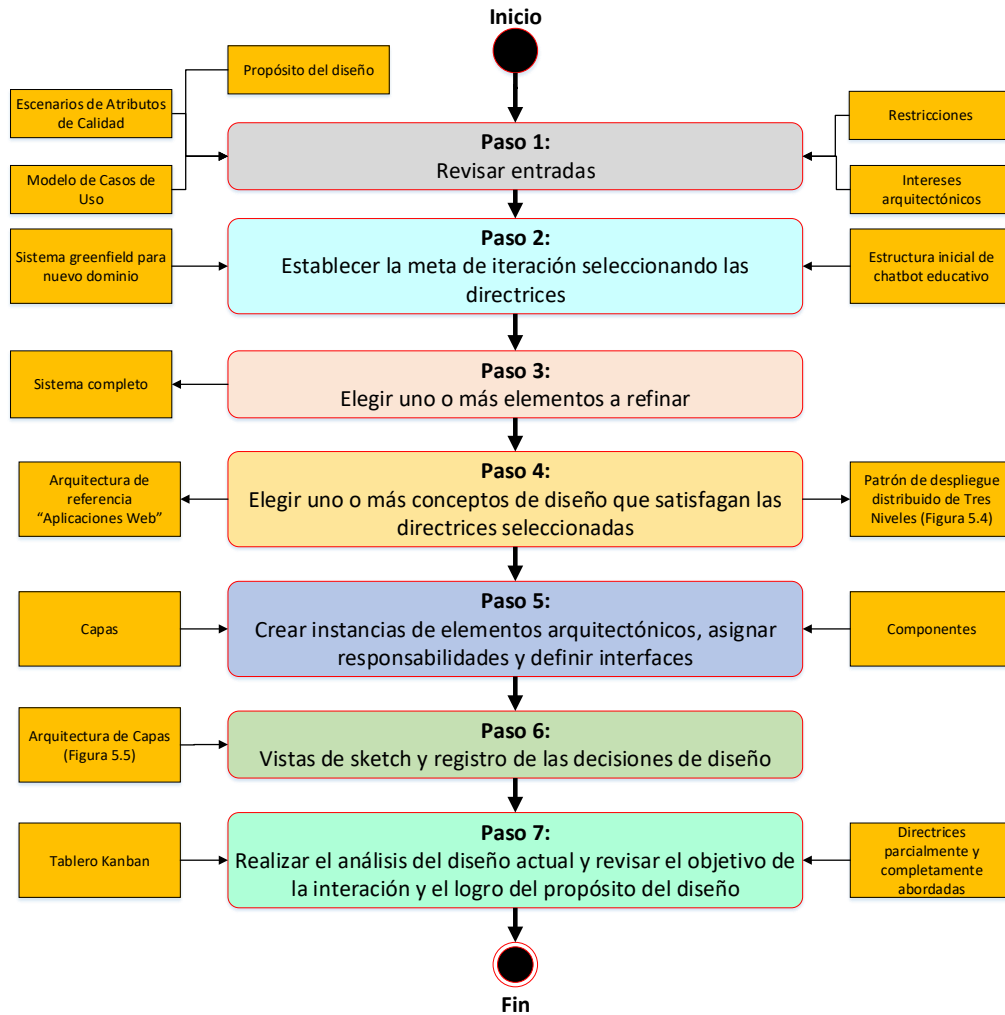


Figura 5.3: Flujo del proceso de diseño de ADD 3.0

5.4.1. Inicio iteración

Anteriormente, se establecieron y documentaron las entradas necesarias para iniciar ADD 3.0. A partir de este punto, presentamos el paso 1 y, a continuación, del paso 2 al 7 que marcan el comienzo de la primera iteración del proceso de diseño.

5.4.1.1. Paso 1: Revisar las entradas

Las entradas a que se ocuparán son:

1. **Propósito de diseño** será el que fue delineado en la Subsección 5.3.1.
2. **Requerimientos funcionales primarios** derivan de los casos del Cuadro 5.1, desde el CU-0 hasta el CU-12.
3. **Escenarios de atributos de calidad** del Cuadro 5.3, se utilizarán AC-3, AC-4, AC-7, AC-12, AC-13, AC-16 y AC-17.
4. **Restricciones** que se incluyen son las mencionadas en RES-1, RES-2, RES-3, RES-5, RES-6, RES-7, RES-8 y RES-9 del Cuadro 5.4.
5. **Intereses arquitectónicos** únicamente se toma IA-1 del Cuadro 5.5.

5.4.1.2. Paso 2: Establecer la meta de iteración seleccionando las directrices

Esta es la primera iteración en el diseño de un sistema *greenfield* para nuevo dominio, por lo que la meta se enfoca en establecer una estructura general inicial del chatbot (IA-1). Consideramos que, por el momento, con una primera iteración será suficiente para definir la estructural general inicial y estará dirigida por las directrices mencionadas en el paso 1.

5.4.1.3. Paso 3: Elegir uno o más elementos del sistema a refinar

Como se está iniciando la primera iteración y es un desarrollo *greenfield*, el elemento a refinar es en sí el sistema completo.

5.4.1.4. Paso 4: Elegir uno o más conceptos de diseño que satisfagan las directrices

La meta es la estructura general del chatbot. Para ello, se debe elegir un concepto de diseño que satisfaga las directrices.

Este paso es importante porque es el punto de partida del diseño general que va a establecerse en la arquitectura. Lo conveniente en este caso es presentar las opciones que

Cuadro 5.6: Arquitectura de referencia para el chatbot educativo.

Arquitectura de referencia	Descripción y justificación	Acción
Aplicaciones de Cliente Enriquecido (<i>Rich Client Applications</i>)	Son aplicaciones que se instalan y ejecutan en la máquina del usuario. Sus interfaces de usuario pueden proveer un alto desempeño, interactividad y experiencia de usuario enriquecida. Esta arquitectura es ideal si queremos realizar la puesta de operación en las máquinas de los usuarios. Por lo tanto, para nuestro caso no es la indicada.	Descartar
Aplicaciones de Internet Enriquecida (<i>Rich Internet Application</i>)	Son aplicaciones desarrolladas usando código que se ejecuta dentro de un navegador web. Acorde a nuestras necesidades, no se requiere desempeñar ningún tipo de procesamiento del lado del cliente. Por lo tanto, esta referencia no es la indicada.	Descartar
Aplicaciones Móviles (<i>Mobile Applications</i>)	Son aplicaciones que se ejecutan en un dispositivo móvil y pueden trabajar en colaboración con un soporte de infraestructura de manera remota. En nuestro caso, no se requiere en esta primera versión ningún desarrollo móvil.	Descartar
Aplicaciones de Servicio (<i>Service Applications</i>)	Son aplicaciones que se caracterizan por no tener ningún tipo de interacción directa con el usuario.	Descartar
Aplicaciones Web (<i>Web Applications</i>)	Es la aplicación que se accede por medio de un navegador web y que se comunica con un servidor mediante el protocolo HTTP. No se requiere instalar nada del lado del cliente. La aplicación reside en el servidor y normalmente emplea una arquitectura por capas. Esta arquitectura es la que más se aproxima a lo establecido en nuestros requerimientos.	Utilizar

hay disponibles para analizar y explicar brevemente el motivo por el cuál se podrían utilizar o no. Cervantes y Kazman [107] facilitan una lista de arquitecturas de referencia, las cuales se exponen en el Cuadro 5.6 para analizar y elegir la más adecuada:

- **Arquitectura de referencia:** indica una lista con las arquitecturas de referencia.
- **Descripción y justificación:** explica brevemente en qué consiste la arquitectura y el criterio de selección.
- **Acción:** muestra la operación descartar u ocupar.

Los casos de uso, que fueron analizados para el sistema, requieren ser accedidos por medio de un navegador web (RES-1). Por lo tanto, la mejor opción es utilizar la arquitectura de referencia Aplicaciones Web (cf. Cuadro 5.6).

Por otro lado, para describir la estructura física, están los patrones de despliegue: No Distribuidos y Distribuidos. En los No Distribuidos, todos los componentes de los

módulos en las diferentes capas residen en un solo servidor, excepto la funcionalidad del almacenamiento de datos. Pero para los Distribuidos, los componentes de la aplicación residen en niveles físicos separados.

En nuestro caso, se utilizará el patrón de despliegue distribuido de tres niveles (*Three-Tier Deployment*). Este patrón establece una arquitectura de software preestablecida para aplicaciones web en tres niveles lógicos:

- Nivel cliente o presentación.
- Nivel aplicación donde los datos son procesados.
- Nivel datos donde los datos asociados con la aplicación son almacenados y gestionados.

De esta manera, se está respetando la restricción de que el chatbot pueda ser accedido desde un navegador web (RES-1) y puede utilizar servicios de la nube tanto para el procesamiento de lenguaje natural (RES-2) como el del almacenaje y base de datos (RES-3).

5.4.1.5. Paso 5: Crear instancias de elementos arquitectónicos, asignar responsabilidades y definir interfaces

En este paso, se muestran las decisiones de diseño consideradas:

- Crear el componente para el inicio de sesión en la capa de presentación (CU-0).
- Crear el componente para el acceso del usuario en la capa de negocio (AC-12, AC-13, RES-5).
- Configurar el acceso del usuario por medio de servicios *Auth*⁷ (CU-0) para que la información correspondiente de estudiante esté ligada, únicamente, a su cuenta (AC-12). Por lo que si un estudiante quiere conocer información de otros estudiantes, el chatbot no deberá permitirlo.
- Crear los componentes para la funcionalidad que manejan los usuarios estudiante y profesor en la capa de orientación (CU-5, CU-6, CU-7, CU-8, CU-9, CU-10, CU-11, CU-12).

⁷Es un método de autenticación de acceso básico disponible para aplicaciones web.

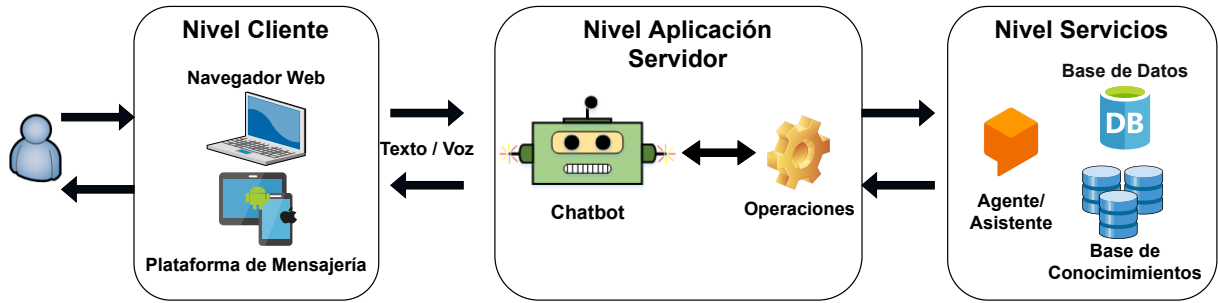


Figura 5.4: Arquitectura de tres niveles para el chatbot educativo.

- Crear el componente servicio de chatbot para manejar los servicios de éste en la capa de servicios (RES-2).
- Crear los componentes para el acceso de datos y el acceso a la base de conocimientos en la capa de datos (RES-3).

Para esta iteración, no se requiere definir interfaces porque estamos centrados en establecer la estructura general del sistema.

5.4.1.6. Paso 6: Vistas de sketch y registro de las decisiones de diseño

La Figura 5.4 muestra la estructura física y la distribución del chatbot educativo. En el lado del cliente, un usuario envía un mensaje de texto o voz a través de un navegador web o una plataforma de mensajería. Este mensaje es recibido por el chatbot, que se encuentra en un servidor, y procede a realizar las operaciones necesarias para proporcionar una respuesta. Estas operaciones pueden incluir el análisis del mensaje o su envío a un servicio de chatbot en la nube a un agente/asistente, y a partir de allí, realizar solicitudes para extraer información de la base de datos o la base de conocimientos, que se encuentran como servicios en la nube. Posteriormente, el chatbot obtiene una respuesta y la entrega al usuario.

La Figura 5.5 muestra un sketch de arquitectura por capas, acorde a las decisiones de diseño, tomando como punto de partida las arquitecturas de referencia y de despliegue que se eligieron. Esta arquitectura reside del lado del servidor, por lo que los elementos y responsabilidades son:

1. **Capa de presentación:** contiene los componentes que implementan y muestran la interfaz de usuario:
 - *Web* pensado para mostrar al chatbot por medio de un navegador web.
 - *Plataformas de Mensajería* presenta al chatbot a través de aplicaciones como Facebook Messenger, Telegram y Whatsapp.
 - *Interfaz de Usuario* implementa mecanismos para una fácil interacción del usuario con el chatbot, independientemente si fue por medio de un navegador web o una plataforma de mensajería.

2. **Capa de negocio:** comprende los componentes que reciben las peticiones del usuario:
 - *Acceso de Usuario* se encarga de manejar el acceso del usuario dependiendo de su rol.
 - *Estudiante* tiene la funcionalidad para el rol *estudiante*.
 - *Profesor* implementa la funcionalidad para el rol *profesor*.
 - *Procesamiento y Formato de Mensajes* analiza los mensajes del usuario para evitar palabras violentas, de odio o acoso.

3. **Capa de orientación:** incluye los componentes *Servicios Escolares* y *Estudiante/Profesor* donde se implementan las características de la clasificación presentada en la Sección 4.1.2 con sus correspondientes casos de uso del Cuadro 5.1. Tiene los módulos: *Calendario y Horario (C&H)*, *Información*, *FAQ*, *Procedimientos*, *Evaluación*, *Materias*, *Retroalimentación*, *Reportes*, *Preguntas y Respuestas (P&R)*, *Salud Bienestar*, *Tutorías* y *Soporte*.

4. **Capa de servicio:** abarca los componentes que brindan acceso a datos internos y externos, funcionalidad de negocio y otros servicios:
 - *Servicio de Chatbot* se encarga de proporcionar la funcionalidad para manejar los servicios del chatbot en la nube.
 - *Servicio de Acceso de Datos* tiene la funcionalidad para el caso en que un mensaje deba transformarse a un formato que deba ser comprensible. Dado

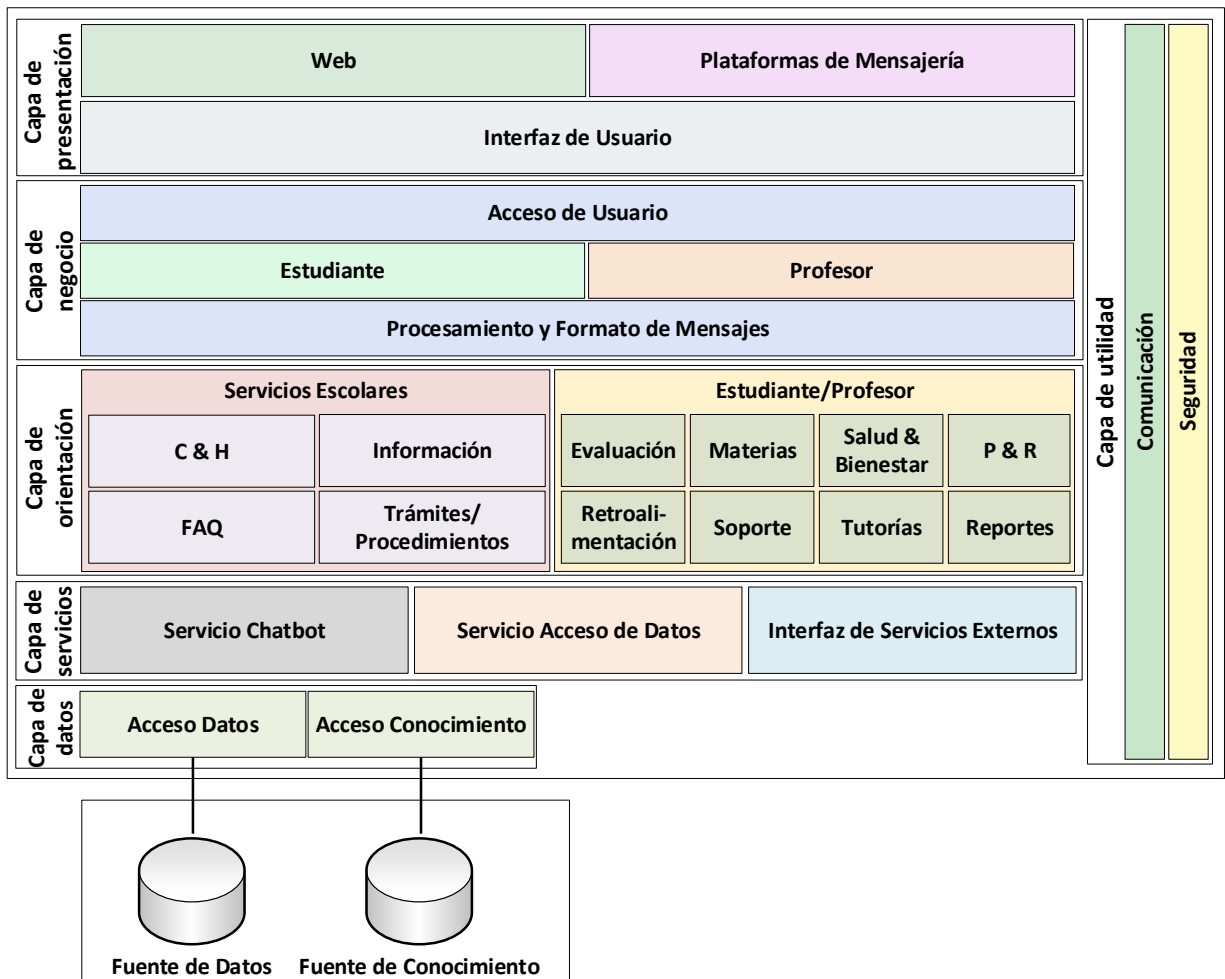


Figura 5.5: Arquitectura de capas del chatbot educativo.

que el chatbot se integra con un servicio de la nube, puede que se necesite implementar algún adaptador que convierta los datos de los servicios a un formato que los otros componentes también entiendan.

- *Interfaces de Servicios Externos* se contempla para complementar la funcionalidad que proporciona el chatbot, lo que podría requerir de otros servicios ajenos a los que se tenía en mente en un principio. Cabe recordar que la tecnología avanza rápidamente y muchas veces la forma de acceder a los servicios cambia.

5. **Capa de datos:** envuelve la funcionalidad para la comunicación con el componente donde residen los datos y/o conocimientos permitiendo el acceso a los mismos.

6. **Capa de utilidad:** se aplica de manera transversal y tiene componentes que manejan aspectos de la comunicación y la seguridad entre las capas: *Comunicación y Seguridad*.

5.4.1.7. Paso 7: Realizar análisis del diseño actual y revisar el objetivo de la iteración y el logro del propósito del diseño

En la descripción de ADD 3.0 se recomienda utilizar un tablero *Kanban* para resumir el estado de las directrices arquitectónicas y las decisiones tomadas durante la iteración. El Cuadro 5.7 contiene columnas en las que se muestra si las directrices han sido parcialmente, completamente o no abordadas en la iteración.

Nuestra meta para esta iteración era establecer una estructura general inicial del sistema, identificando sus componentes básicos, sin profundizar en detalles de implementación lo cuál dio como resultado las Figuras 5.4 y 5.5. Por lo tanto, no se requiere una segunda iteración y concluimos el proceso en esta primera fase.

5.5. Discusión

Dado la importancia y alcance de este Capítulo en el trabajo de investigación la discusión se describirá por secciones.

Cuadro 5.7: Tablero Kanban para la primera iteración.

No abordado	Parcialmente abordado	Completamente abordado	Decisiones de diseño realizadas durante la iteración
	AC-3, AC-4	RES-1	Se seleccionó la arquitectura de referencia Aplicaciones Web. Además, el componente <i>Web</i> ha sido identificado y establecido en la capa de presentación.
	AC-7		El componente <i>Web</i> en la capa de presentación ha sido identificado y establecido para que se pueda operar independientemente si se usa una PC, tableta o teléfono inteligente. Adicionalmente, se añadió el componente <i>Plataformas de Mensajería</i> en la capa de presentación para que en caso de utilizar aplicaciones como Telegram o Facebook Messenger en una tableta o teléfono inteligente pueda operar de igual manera que si fuera en una PC.
	AC-17		El componente <i>Interfaz de Usuario</i> ha sido identificado y establecido en la capa de presentación.
		CU-0	El componente <i>Acceso de Usuario</i> ha sido identificado y establecido en la capa de negocio (ver Figura 5.5). Además, se utilizará <i>Auth</i> para esta funcionalidad.
	AC-3, AC-4, AC-12, AC-13, AC-16	RES-5	Los componentes <i>Acceso de Usuario</i> , <i>Estudiante</i> , <i>Profesor</i> han sido identificados y establecidos en la capa de negocio. También, el módulo <i>Tutorías</i> para la canalización. Adicionalmente, el componente <i>Seguridad</i> en la capa de utilidad (ver Figura 5.5).
	RES-7		El componente <i>Procesamiento y Formato de Mensajes</i> en la capa de negocio ha sido identificado y establecido (ver Figura 5.5).
	CU-1, CU-2, CU-3, CU-4		El componente <i>Servicios Escolares</i> y los módulos preliminares <i>C&H</i> , <i>FAQ</i> , <i>Información</i> y <i>Trámites/Procedimientos</i> han sido identificados y establecidos en la capa de orientación (ver Figura 5.5).
	CU-5, CU-6, CU-7, CU-8, CU-9, CU-10, CU-11, CU-12		El componente <i>Estudiante/Profesor</i> y los módulos preliminares <i>Evaluación</i> , <i>Materias</i> , <i>Salud & Bienestar</i> , <i>P & R</i> , <i>Retroalimentación</i> , <i>Soporte</i> , <i>Tutorías</i> y <i>Reportes</i> han sido identificados y establecidos en la capa de orientación (ver Figura 5.5).
	RES-2		Los componentes <i>Servicio Chatbot</i> e <i>Interfaz de Servicios Externos</i> en la capa de servicios han sido identificados y establecidos (ver Figura 5.5).
		RES-6	Estructurando el chatbot en una arquitectura de tres niveles, permitirá soportar múltiples clientes para conectarse al servidor de la aplicación (ver Figura 5.4).
	RES-3		Se seleccionó la estructura física del chatbot usando el patrón de arquitectura de despliegue de tres niveles (ver Figura 5.4). También, se identificó y estableció la capa de datos (ver Figura 5.5).
		IA-1	Selección de la arquitectura de referencia Aplicación Web, patrón de despliegue de tres niveles (ver Figura 5.4) y Arquitectura de Capas (ver Figura 5.5).
		RES-8	El componente <i>Estudiante/Profesor</i> y los módulos preliminares <i>Evaluación</i> , <i>P & R</i> , <i>Retroalimentación</i> y <i>Tutorías</i> han sido identificados y establecidos para manejar estímulo-respuesta y reforzar el proceso de aprendizaje (ver Figura 5.5), promoviendo una participación activa.
		RES-9	Las capas orientación, servicios y datos junto a los componentes <i>Estudiante</i> y <i>Profesor</i> , son suficientes para abarcar la definición de chatbot educativo (ver Figura 5.5).

5.5.1. ADD 3.0

El conocimiento científico se basa en la acumulación de evidencia empírica que se obtiene cuando se realiza una observación sistemática y detallada del fenómeno de interés. Los métodos son los enfoques que ayudan a obtener las formas (datos) de lo que se observa. De tal manera que seguir un método nos obliga a considerar, en múltiples niveles, una forma de pensar y resolver uno o varios problemas, junto a prácticas concretas que se requieran para realizar una investigación [256].

En un principio, se consideró la posibilidad de diseñar la arquitectura del chatbot sin seguir un método específico. La experiencia en proyectos de diseño de arquitectura y desarrollo de software puede brindar ventajas significativas, como la capacidad para trabajar en equipo, conocimiento de herramientas, tecnologías y tendencias, así como el desarrollo de habilidades. Sin embargo, es importante destacar que la experiencia por sí sola no garantiza la obtención de los datos más adecuados, dado que hay múltiples variables que se ven implicadas. La combinación de teoría y experiencia puede resultar en la obtención de los mejores datos posibles en los resultados del proyecto.

Para nuestro caso consideramos importante aplicar un método que proporcione una guía concreta, para minimizar los posibles errores que se puedan tener al introducir u omitir aspectos importantes, para lograr nuestro objetivo de crear una arquitectura genérica de chatbots educativos. Entre los diversos métodos disponibles que podemos encontrar para el diseño, como se exploró en la Sección 2.3.2, hemos seleccionado el método ADD 3.0 por varias razones fundamentales.

ADD 3.0 se enfoca fuertemente en el diseño y ofrece una orientación detallada para cada fase del proceso de desarrollo. Esto es particularmente valioso, especialmente cuando no se cuenta con una experiencia extensa en el campo. El método también proporciona listas de arquitecturas de referencia, patrones de diseño, estrategias y tácticas que pueden ser de gran utilidad. Adicionalmente, se destaca por su enfoque didáctico, lo que facilita la comprensión de cómo aplicar el método en situaciones concretas. No solo establece un marco de trabajo claro, sino que también brinda ejemplos concretos que ilustran la aplicación práctica del método. Por ejemplo, gracias a la adopción de ADD 3.0, hemos podido establecer directrices fundamentales, como propósito de diseño, casos de uso, escenarios de atributos de calidad, restricciones e intereses arquitectónicos. Estas directrices fueron cruciales y necesarias, ya que proporcionan una base sólida para nuestro resultado. Sin su implementación, los planteamiento hechos podrían carecer de la profundidad y el nivel de

detalle necesarios para crear una arquitectura efectiva. A continuación, analicemos más detenidamente cada paso de este método.

5.5.2. Método

El inicio del paso 1 es sencillo, pues simplemente se requiere revisar que se disponga de todas las entradas necesarias para la iteración. Esto formaliza, en cierto sentido, la generación de documentación requerida y permite familiarizar, a los posibles involucrados en el diseño, con los aspectos que conlleva el proyecto. Para nuestro caso, si fue importante utilizar las directrices que se presentaron en la Sección 5.3 para esta primera iteración, aunque en un principio no se tenían.

La elección de la meta de iteración en el paso 2 y los elementos a seleccionar para el refinamiento en el paso 3 también resultaron ser simples. Concretamente se tenía que establecer entre tres tipos de diseño *greenfield* para sistemas maduros, *greenfield* para nuevos dominios o *brownfield* para un sistema existente, y a partir de esa elección, construir la estructura general con los componentes básicos de un chatbot educativo. Sin embargo, existiría el riesgo que nuestro proyecto caiga relativamente en un nuevo dominio y no en uno maduro, lo que significa que las listas de referencias, que facilita el método, son pocas o inexistentes. A grandes rasgos, la diferencia entre un sistema de nuevo dominio y uno de dominio maduro reside en las herramientas, tecnologías y conocimiento base que se tiene para el proyecto. Si bien los chatbots no son una novedad, las herramientas y tecnologías de desarrollo de los últimos años sí lo son, ya que son relativamente recientes y están en constante evolución. Lo que se utilizó hace diez años en este campo podría considerarse obsoleto en la actualidad.

Los pasos 4, 5 y 6 son cruciales y críticos en el proceso de diseño, y requieren una revisión minuciosa. Es en esta fase donde el diseño comienza a adquirir relevancia. En primera instancia, se definió la estructura lógica, y para este propósito, se determinó que la arquitectura de referencia de Aplicaciones Web (consultar Cuadro 5.6) era la mejor elección para nuestras necesidades específicas. Después, se estableció la estructura física utilizando el patrón de despliegue de tres niveles (ver Figura 5.4). Esta elección facilitó la comprensión de cómo se distribuiría el chatbot. Como resultado, algunas de las directrices mencionadas en las fases anteriores se satisfacían con estos dos elementos de manera casi inmediata. Las directrices restantes se abordaron mediante la implementación de una

arquitectura de capas y sus respectivos componentes (ver Figura 5.5).

En el paso 7, se realiza un análisis del diseño actual para evaluar si se ha alcanzado el objetivo de la iteración y el logro del propósito del diseño. En este punto, resulta crucial poder visualizar de alguna manera (por ejemplo, mediante el uso de tableros Kanban) las directrices que no se han abordado, las que se han abordado parcialmente y las que se han completado en el diseño.

5.5.3. Directrices

Como podemos observar en el Cuadro 5.7, el número de directrices parcialmente abordadas supera al de las completamente abordadas. Esta disparidad se debe a la naturaleza de desarrollar una arquitectura genérica para un chatbot educativo. En este contexto, es necesario definir ciertos elementos y características de manera general, sin entrar en detalles técnicos específicos. Esto se hace para no limitar prematuramente las posibles implementaciones futuras. Si los desarrolladores deciden utilizar estrategias o tácticas, tendrán la flexibilidad de hacerlo a su manera con las herramientas y tecnologías que más les acomoden, pero tiene que ser sobre la base de los componentes previamente analizados y establecidos que proponemos. En cambio con la directrices que logramos abordar completamente en su mayoría se debió a que éstas fueron completamente satisfechas por utilizar una arquitectura de referencia, patrón de despliegue y arquitectura de capas.

Una de las fortalezas de este trabajo de diseño fue el análisis y la elaboración de las entradas requeridas por el método. Al principio, algunas de estas entradas, como el propósito de diseño, el modelo de casos de uso y el interés arquitectónico, se manifestaron de manera natural debido a los elementos como el objetivo de este proyecto de investigación y la clasificación de los chatbots educativos, lo cuál facilitó en gran medida la elaboración de las entradas. Sin estos elementos, el proceso de diseño habría sido más complejo y posiblemente habría requerido más iteraciones.

Sin embargo, con respecto a los escenarios de atributos de calidad y, quizá en menor medida, a las restricciones, la tarea de análisis y elaboración era más compleja. Idealmente, para los escenarios de casos de uso, lo recomendable es contar con la participación activa de los *stakeholders*, los cuales en este proceso de diseño no estuvieron disponibles. Su presencia habría facilitado el diseño. No obstante, existen técnicas como los Árboles de Utilidad los cuáles podemos utilizar en caso de contar con el *stakeholder*.

5.5.4. Arquitectura, capas y componentes

En relación a las dos arquitecturas presentadas: arquitectura de tres niveles y arquitectura de capas, podemos decir lo siguiente. La arquitectura de tres niveles (ver Figura 5.4), muestra claramente cómo está distribuido el chatbot educativo. Se destaca que los usuarios de lado del cliente no necesitan contar con alguna instalación adicional proveniente del chatbot, ya que solo requieren un navegador web o una plataforma de mensajería para interactuar con él. Esta elección se tomó dada la revisión de las tendencias actuales del uso de chatbots (ver Capítulo 4), donde es común que los usuarios accedan a través de navegadores web o aplicaciones de mensajería instantánea en tabletas y teléfonos inteligentes. Plataformas populares como Facebook Messenger, Line, Telegram, WeChat y WhatsApp han estado promoviendo la integración de interfaces conversacionales para los usuarios, como lo pueden ser los chatbots, lo que amplía significativamente su alcance entre ellos. A pesar de que inicialmente no se había considerado la inclusión de una plataforma de mensajería en la arquitectura, se volvió evidente su importancia como un componente esencial. Por otro lado, en el servidor es donde reside el chatbot educativo y desde allí se realizan las operaciones necesarias para dar respuesta a las peticiones de los usuarios. En caso de ser necesario, el chatbot también puede acceder a servicios en la nube para llevar a cabo estas operaciones. En última instancia, se decidió que los servicios de chatbot sean accedidos desde la nube, el motivo principal es que la tendencia es utilizar marcos de trabajo que ofrezcan soluciones bastante avanzadas para el desarrollo de chatbots lo cuál facilita el desarrollo, en especial cuando los posibles mensajes que un usuario manda son por voz.

En la arquitectura de capas (ver Figura 5.5), tenemos varios elementos que examinaremos a continuación. La capa de presentación incluye tres componentes esenciales. Hasta cierto punto son componentes básicos que podemos encontrar en muchos tipos de desarrollos web, en el cuál hay un componente web para que se pueda acceder mediante una página web, pero también, un componente destinado para plataformas de mensajería, en caso de que el usuario requiera acceder por medio de una aplicación de mensajes como Telegram, entre otros. Por último, se encuentra el componente para la interfaz de usuario. Por lo tanto, la capa de presentación es una capa sencilla sin ningún tipo de complejidad relevante, pero con la aclaración de que dada la arquitectura de referencia utilizada (Aplicaciones Web) esta capa reside del lado del servidor y no se requiere instalar nada del lado del cliente.

Avanzando hacia la capa de negocio encontramos un componente crucial que permite el acceso al chatbot. No obstante, este acceso puede variar según el rol del usuario, que puede ser estudiante o profesor. Debemos destacar que en esta capa incluimos un componente para filtrar mensajes en busca de contenido violento, discriminatorio y ofensivo, lo que empieza a introducir un acercamiento hacia el procesamiento del lenguaje natural.

Dentro de la arquitectura, consideramos que la capa de orientación, con sus componentes y módulos, es la más significativa. Esta capa es la encargada de dar un enfoque educativo al chatbot de manera predeterminada, y su inclusión se alineó naturalmente con la clasificación de chatbots que presentamos en la Subsubsección 4.1.2.

Al incorporar esta capa, estamos abordando dos aspectos esenciales en las restricciones de las entradas: la definición de un chatbot educativo y un acercamiento a elementos pedagógicos. La inclusión de esta capa simplificó significativamente el proceso de diseño, ya que sin ella habría sido necesario realizar un análisis mucho más complejo. Además, vale la pena señalar que a medida que refinemos la clasificación de chatbots educativos de la Subsubsección 4.1.2, los componentes y módulos de esta capa podrán adaptarse y evolucionar sin afectar de manera directa al resto de las capas de la arquitectura. Esto es un beneficio valioso para futuros desarrollos y ajustes en la orientación educativa del chatbot ante otras arquitecturas propuestas.

La capa de servicios se ha definido para alojar componentes que permitan el acceso a datos tanto internos como externos, facilitando así la utilización de servicios de chatbot. Durante nuestra investigación sobre cómo dotar de inteligencia a un chatbot, evaluamos diversas herramientas (ver Figura 4.3), destacando claramente entre ellas, Dialogflow, seguida de AIML.

Como mencionamos anteriormente, Dialogflow se presenta como un marco de trabajo versátil y completo para la creación de chatbots, proporcionando una amplia variedad de herramientas avanzadas para el desarrollo e integración, junto con una impresionante gama de servicios. Esto coloca a Dialogflow en una posición altamente ventajosa en comparación con AIML. En contraste, AIML se basa en una plantilla que debe coincidir con precisión con el patrón de texto definido en un etiqueta para dar una respuesta. Si no se logra una coincidencia de la manera más exactamente posible, la respuesta al usuario no será satisfactoria.

Bajo este escenario, hemos planeado que la capa de servicios facilite la integración de marcos de trabajo para el desarrollo de chatbots desde la nube. Aunque es posible utilizar

AIML en esta capa, lo cual sería más sencillo y eficiente en términos de codificación, pero debemos tener en cuenta que esta elección podría tener repercusiones negativas en el rendimiento del chatbot. La carga de trabajo generada en el servidor podría aumentar significativamente, lo que podría afectar la experiencia de los usuarios al disminuir la capacidad de respuesta del chatbot. Además, AIML no tiene mecanismos para interactuar con voz lo cuál lo pone en una desventaja con otras herramientas.

La capa de datos en esta arquitectura no difiere significativamente de otros proyectos web en términos generales. En esta capa, encontramos los componentes para manejar el acceso tanto a datos como al conocimiento. A grandes rasgos, una base de datos tiene como propósito almacenar datos de manera estructurada para su posterior procesamiento. En cambio, una base de conocimientos almacena información que debe estar organizada y se utiliza para la resolución de problemas.

La base de datos se emplea para almacenar los datos de los usuarios, mientras que la base de conocimientos se centra en enriquecer, en este caso, al chatbot educativo con ese conocimiento (inteligencia).

¿Cómo se puede organizar la información de una base de conocimientos? Primero hay que revisar qué herramienta se utilizará para la base de conocimientos. Por ejemplo, hay quienes proceden a definir un conjunto de reglas o un conjunto de casos, pero hay otros mecanismos que permiten poner el conocimiento en forma de preguntas y respuestas definidas por una serie de reglas.

Por otro lado, una base de conocimientos no contiene información de usuarios, sino que se enfoca en temas de interés, en nuestro caso el educativo. Organizar la información en una base de conocimientos implica seleccionar la herramienta adecuada para el propósito. Por ejemplo, algunas personas optan por definir un conjunto de reglas o casos, mientras que otros utilizan mecanismos que permiten representar el conocimiento en forma de preguntas y respuestas definidas por instrucciones para que la herramienta pueda extraer el conocimiento. Dialogflow en su versión en inglés, tiene una gran variedad de modos para hacer una base de conocimientos. La elección dependerá en gran medida de las necesidades y preferencias del proyecto.

Finalmente, llegamos a la capa de utilidad. Aunque esta capa no se considera una parte funcional directa del chatbot educativo, desempeña un papel en las operaciones generales de esta arquitectura. Dado que existen varias capas con sus respectivos componentes, es esencial contar con un medio de comunicación entre ellas y garantizar al mismo tiempo

seguridad.

Las capas que proponemos se ligan de la siguiente manera. Primeramente, el usuario es el que va a interactuar con la capa de presentación, la cuál encierra los diferentes componentes que tienen que ver con la interfaz de usuario para la comunicación con el chatbot. Luego, la capa de presentación se comunica con la capa de negocio para facilitar el acceso del usuario según su rol.

La capa de negocio establece las reglas y validaciones para el usuario y consume funcionalidades correspondientes de la capa de orientación, por lo tanto, la capa de orientación debe ofrecer la funcionalidad educativa esencial del chatbot. La capa de servicio es responsable de integrarse con los servicios del chatbot y consume servicios de la capa de datos que gestiona el acceso a los datos que a su vez proporciona a las capas superiores. Finalmente, la capa utilidad está ligada de manera transversal al resto de las capas con funcionalidad ajena al chatbot pero de carácter común para la comunicación y seguridad entre las capas.

Este diseño busca asegurar una separación de responsabilidades y una comunicación eficiente entre las diversas capas de tal manera que pueda ser una arquitectura flexible.

5.5.5. Usuarios de la arquitectura

Es claro que los usuarios finales del chatbot educativo son principalmente los estudiantes. Sin embargo, al establecer nuestra arquitectura, el tipo de usuario ya no es directamente el estudiante. Por lo tanto, surge la pertinente pregunta: ¿Quiénes serán los usuarios de esta arquitectura?

Los posibles usuarios finales de la arquitectura podrían estar divididos de la siguiente manera. En primer lugar en un entorno práctico, tenemos al equipo de desarrollo, encargado de la implementación. Además, está el arquitecto de software, responsable de diseñar la estructura y la organización del sistema basándose en nuestra propuesta. Continuamos con posibles diseñadores, quienes se ocuparán de la interfaz de usuario, garantizando la experiencia del usuario con respecto al chatbot.

Por otro lado, se encuentra el contexto educativo, que abarca las escuelas interesadas en integrar chatbots, aprovechando su creciente popularidad en el último año. Proponemos que los usuarios potenciales podrían ser profesores y educadores, que interactuarán con el chatbot para crear contenido educativo o supervisar el progreso del estudiante. Además,

un equipo de desarrollo de contenido, responsable de crear y actualizar el contenido proporcionado por el chatbot. Por último, en la actualidad es común que las asociaciones de padres de familia participen activamente en la educación, especialmente cuando los estudiantes son menores de edad, por lo tanto, no se descarta esta posibilidad como usuarios.

5.5.6. Comparación de nuestra propuesta con otras arquitecturas

Para realizar una comparación de nuestra propuesta arquitectónica con las arquitecturas identificadas en el estado del arte, consideremos lo siguiente. Los trabajos revisados representan publicaciones, particularmente, que abordan problemas específicos de chatbots educativos mediante arquitecturas *ad hoc*. En contraste, nuestra propuesta sigue un enfoque genérico, aplicable a diversas situaciones o propósitos educativos siempre se encuentren dentro de la clasificación de chatbots educativos (ver Subsección 4.1.2). En comparación con las arquitecturas revisadas en el estado del arte, nuestra propuesta exhibe una flexibilidad alta a largo plazo. Además, favorece la estandarización para este tipo de desarrollos, a diferencia de las soluciones revisadas, que fueron principalmente adaptaciones inmediatas. Por lo tanto, la adaptabilidad se convierte en otro factor destacado en nuestra propuesta, algo que las arquitecturas *ad hoc* tienden a carecer. Esta diferencia podría explicar por qué la mayoría de los desarrollos de chatbots educativos suelen iniciarse desde cero.

Con respecto a los trabajos que utilizaron arquitecturas específicas como Cliente-Servidor, Modelo Vista Controlador, Capas, Componentes y Microservicios, no es sencillo establecer una comparación homogénea. Nosotros tomamos como entradas una serie de elementos, que nos requirió el método de diseño utilizado, y que no vienen documentados en estos trabajos. Además, los diagramas presentados suelen omitir muchos de los detalles de sus arquitecturas. Esto puede tener la siguiente razón. Las publicaciones tienen un cierto límite de caracteres o páginas. Exponer muchos diagramas dificulta las posibles publicaciones. Además, entrar en muchos detalles en una sección puede perjudicar otras. Por lo tanto, no hay una manera concreta de hacer estas comparaciones con las arquitecturas mencionadas.

Sin embargo, al comparar nuestro trabajo con los Cuadros 4.3 y 4.3 presentados en el estado del arte, observamos columnas que describen las características y aspectos técni-

cos de los trabajos revisados. En este contexto, podemos caracterizar nuestro trabajo. Por ejemplo, en nuestra arquitectura no se han definido componentes específicos para gestionar avatares ni para el manejo de multilinguaje. Tampoco se ha establecido de manera explícita algún componente para la gestión de texto o voz. Sin embargo, dado que definimos componentes que hacen uso de mecanismos avanzados mediante herramientas de procesamiento de lenguaje natural, tanto para texto como para voz, no consideramos necesario definir componentes específicos para estos propósitos, no lo cual no descartamos para un trabajo futuro. De esta manera, la interacción por texto y voz está respaldada en la arquitectura de manera implícita.

Además, nuestra arquitectura ofrece soporte para nuevos desarrollos y la continuidad de dicho trabajo como legado. También incorporamos componentes para el mecanismo de IA, aunque de manera implícita en el componente del chatbot. La elección de una arquitectura de capas resulta ser la más adecuada para nuestra propuesta genérica, y por último, contemplamos el manejo de bases de datos como bases de conocimiento.

La elección entre una arquitectura genérica, una específica y una *ad hoc* dependerá en gran medida de los requisitos únicos del proyecto, las metas a largo plazo y las restricciones en los recursos.

Finalmente, somos conscientes que podría existir la posibilidad de que nuestra arquitectura genérica no sea una buena opción y se requiera de una solución personalizada que se adapte mejor a las necesidades de un proyecto.

5.5.7. Un chatbot inteligente

Nuestra propuesta incorpora dos elementos de Inteligencia Artificial para el desarrollo de chatbots educativos inteligentes. En primer lugar, tenemos el componente de *Procesamiento y Formato de Mensajes* en la capa de negocio, que se encarga del procesamiento del lenguaje natural para analizar si el mensaje del usuario incluye contenido violento u ofensivo. Por otro lado, aprovechamos los servicios de la nube para el desarrollo de chatbots, una tendencia que ha ido en aumento en los últimos años. Estos servicios ofrecen capacidades de aprendizaje automático, eliminando la necesidad de que los desarrolladores posean conocimientos avanzados en la materia para lograr que sus chatbots sean inteligentes.

La vieja generación de chatbots, desarrollada entre las décadas de los sesenta y noventa,

tenía requisitos bastante estrictos para los usuarios. Se esperaba que los usuarios escribieran sus frases con precisión, evitando errores de puntuación, ortografía y gramática, ya que cualquier error podía confundir al chatbot y llevar a respuestas incoherentes. Además, era esencial que las frases coincidieran exactamente con lo que se había programado en las plantillas de preguntas o en la base de conocimientos del chatbot.

En la actualidad, con la introducción de la comprensión del lenguaje natural, esta rigidez ha disminuido significativamente. Los chatbots modernos están diseñados para comprender el lenguaje humano de manera más flexible y pueden interpretar y responder a preguntas o mensajes con ciertos errores gramaticales o de ortografía. Ésto ha llevado a considerar que los chatbots, que utilizan la comprensión del lenguaje natural, poseen una forma intrínseca de inteligencia [257].

Finalmente, es probable que con el paso del tiempo la palabra «inteligente» se deje de utilizar de manera explícita para describir a los chatbots, ya que se asumirá que la inteligencia es una característica inherente en estos sistemas.

5.5.8. Cruce del diseño instruccional

Entre las diferentes actividades que realizan los docentes y pedagogos para la creación de cursos, encontramos lo denominado como: diseño instruccional. Este proceso implica la planificación, la preparación y el diseño de los recursos y entornos necesarios para facilitar el aprendizaje. Para esto, se pueden encontrar modelos que sirven de guía para sistematizar este proceso, *e.g.*, los modelos de Gagne, ASSURE, Dick y Carey, Jonassen y ADDIE [258].

La integración del diseño instruccional en nuestra arquitectura de chatbot educativo es un desafío complejo. Se requiere de un cuidadoso análisis de aspectos pedagógicos y metodológicos para asegurar que el chatbot contribuya de manera efectiva al proceso de enseñanza-aprendizaje. En una primera instancia, reconocemos que no podemos asegurar una alineación con todos los elementos que abarcaría el diseño instruccional, sin la asesoría de expertos en esa área. No obstante, nuestra propuesta de arquitectura lograr incorporar elementos básicos, incluso sin recurrir a modelos específicos como los mencionados anteriormente.

Un claro ejemplo de esta alineación la podemos observar en la capa de orientación (ver Figura 5.5), que está diseñada para centrarse principalmente en el estudiante, pensada

de la siguiente manera. Con retroalimentación inmediata se podrá dar respuestas y acciones para monitorear el progreso del estudiante, lo cual es esencial para la corrección de errores y el aprendizaje efectivo. La evaluación de tareas, prácticas o exámenes, también se integran en esta capa. La posibilidad de recibir seguimiento por parte de la tutoría está presente. El acceso a realizar una pregunta y obtener una respuesta está abordado. La actualización continua cognitiva mediante la fuente de conocimiento se puede apreciar en la capa. Y si bien, la inclusión de elementos multimedia y enlaces a recursos adicionales se puede dar sin problemas en cualquiera de los componentes de la capa, lo que enriquece la experiencia de aprendizaje y facilita la explicación efectiva de conceptos.

Por otro lado, reconocemos puntos para mejorar en nuestra propuesta con respecto al diseño instruccional. Por mencionar algunos que son evidentes, tenemos la accesibilidad desde la perspectiva de estudiantes con diferentes habilidades y necesidades, esto es un desafío que requiere una mayor atención para garantizar la inclusividad en el diseño. Asimismo, la falta de estrategias específicas y claras para motivar a los estudiantes, como recompensas virtuales o elementos lúdicos, es un punto débil identificado.

Los modelos educativos comparten el objetivo fundamental de verificar que los alumnos hayan internalizado los contenidos establecidos en un programa educativo. Aunque existen varios enfoques, como el constructivista, tradicional, conductivista, entre otros, todos tienden a converger en esta meta general. Desde nuestra perspectiva, la arquitectura que proponemos no se adscribe directamente a ninguno de estos modelos, pero se alinea más con un enfoque constructivista. Esta afinidad se refleja en la participación activa entre el estudiante y el chatbot, tal como se define en los componentes de la capa de orientación.

Finalmente, un análisis a mayor profundidad tomando como referencia el diseño instruccional en nuestra propuesta no solo beneficiaría la efectividad del aprendizaje, también ofrecería a los estudiantes una experiencia educativa más significativa y atractiva. Sin embargo, reconocemos que la asesoría de expertos en la materia sería invaluable para lograr una alineación más precisa.

Capítulo 6

Implementación

En este capítulo, se presenta la implementación del chatbot educativo tomando como referencia la arquitectura que se desarrolló en el Capítulo 5. Primeramente, se presentan las herramientas utilizadas para el desarrollo (cf. Sección 6.1). Después, el tipo de chatbot que se implementó (cf. Sección 6.2). Finalmente, se presenta la discusión de este capítulo (cf. Sección 6.3).

6.1. Herramientas

6.1.1. Software de aplicación

El software de aplicación son aquellos programas informáticos que permiten realizar una o varias tareas específicas. En esta sección se presenta el software de aplicación que se utilizó para realizar los diagramas en el Lenguaje Unificado de Modelado (UML).

6.1.1.1. Diagramas

Diagrams.net¹ es un software libre que permite la elaboración de diagramas de flujo, organigramas, diagramas de UML, entre otras cosas. Tiene una basta galería para incorporar iconos de alta calidad.

Microsoft Visio 2016 es una herramienta privada con una basta funcionalidad para elaborar todo tipo de diagramas. Entre ellos destacan aquellos diagramas para UML.

¹<https://app.diagrams.net/>

Ambos programas se utilizaron para elaborar los diagramas que se muestran en este capítulo.

6.1.2. Software de programación

El software de programación es un conjunto de herramientas que nos permite construir programas informáticos utilizando un lenguaje de programación, redes, bases de datos o marcos de trabajo. En esta sección se presenta el software de programación que se utilizó para la programación del chatbot educativo.

6.1.2.1. Visual Studio Code

Visual Studio Code (VSC)² es un editor de código fuente desarrollado por Microsoft y se ofrece de forma gratuita. Se puede instalar en sistemas operativos GNU/Linux, macOS y Windows. Este editor es bastante utilizado por la comunidad debido a su integración con una gran cantidad de lenguajes de programación.

VSC se utilizó para la programación del *front-end* y *back-end* del chatbot educativo. Los complementos que están disponibles de forma gratuita hacen primordial esta herramienta dado que acelera el desarrollo. Nosotros recomendamos utilizar los siguientes: Angular Language Service, Angular Snippets, Auto Rename Tag, CSS Peek, Image Preview, Ident-Rainbow, Material Icon Theme y Prettier.

6.1.2.2. Lenguaje de programación y gestor de paquete

Angular³ es un macro de trabajo para el desarrollo de aplicaciones web de una sola página. Fue creado en el lenguaje de programación TypeScript. Algo que caracteriza a Angular es que implementa el patrón de diseño Modelo Vista Controlador, permitiendo que el desarrollo y las pruebas sean fáciles de llevar a cabo. Además, permite utilizar una interfaz de línea de comandos para añadir componentes, clases y servicios al proyecto, ahorrando tiempo para el programador. También, facilita la integración de HTML, CSS y TypeScript.

Npm⁴ es un sistema para la gestión de paquetes para NodeJs⁵. Fue desarrollado en

²<https://code.visualstudio.com/>

³<https://angular.io/>

⁴<https://www.npmjs.com/>

⁵<https://nodejs.org/en/>

el lenguaje de programación Javascript. Npm, se utiliza principalmente para instalar bibliotecas de manera sencilla de tal manera que permite administrar módulos, distribuir paquetes y agregar dependencias a los proyectos. Cabe señalar que npm va en combinación con Node.js.

NodeJs es un entorno de ejecución para el lenguaje de programación JavaScript. Permite crear aplicaciones web escalables de una manera sencilla.

6.1.2.3. Servicios en la nube

Dialogflow es una plataforma de comprensión del lenguaje natural que se utiliza para diseñar e integrar una interfaz de usuario conversacional en aplicaciones de tipo chatbot. Se ha posicionado como uno de los más utilizados en el desarrollo de *bots* y proporciona dos tipos de servicios virtuales.

Actualmente, Dialogflow cuenta con dos productos Dialogflow ES y CX. Ambos permiten desarrollar chatbot pero hay diferencias notables en ellos, ya que el producto CX es un poco más avanzado proporcionando más características a diferencia que su versión ES.

Firebase⁶ es una plataforma de desarrollo que permite crear aplicaciones. Entre los servicios que ofrece encontramos los siguientes: autenticación de usuarios, bases de datos en tiempo real y almacenamiento de datos.

Uno de los beneficios de emplear Dialogflow y Firebase en nuestro proyecto es que ambos pertenecen a la compañía de Google y facilita la interoperabilidad entre los dos.

Ngrok⁷ es un programa que agiliza la creación de un servidor local y generar dinámicamente una URL pública.

Esta herramienta nos facilita la creación de un sistema de comunicación automático (conocido como *webhook*), por el protocolo HTTP, entre aplicaciones.

6.1.2.4. Controlador de versiones

Git⁸ es un sistema distribuido, libre y de código abierto, para el control de versiones. Está diseñado para manejar desde proyectos pequeños hasta proyectos grandes de manera rápida y eficiente. Es bastante utilizado cuando se desea trabajar desde locaciones

⁶<https://firebase.google.com>

⁷<https://ngrok.com>

⁸<https://git-scm.com/>

geográficas diferentes. Permite tener proyectos públicos y privados. Cabe resaltar que fue desarrollado por Linux Torvalds.

La implementación del proyecto de chatbot estará en el siguiente repositorio:

- <https://github.com/cinvesbot/chatterbot>

Actualmente, contamos con una versión preliminar del proyecto, por lo que se encuentra en modo privado. Una vez que se complete la documentación y se tenga una versión estable, el repositorio será de dominio público, pero eso quedará como trabajo futuro.

6.2. Cinvesbot Servicio Escolar

Una de las categorías (ver Cuadro 4.1) que tenemos de nuestra clasificación de chatbot educativos es: Orientado a Servicios Escolares (OSE). En esta categoría resalta que la mayor parte de la funcionalidad se enfoca en proporcionar información.

Para familiarizarnos con las herramientas, empezaremos implementando el chatbot educativo OSE. Esto nos permitirá ver cómo se comporta el chatbot con las herramientas que elijamos. Además, existe la posibilidad de generar componentes que después podremos ocupar en caso de querer implementar un chatbot Orientado a Estudiante/Profesor.

Para OSE debemos tener en cuenta lo siguiente:

1. El proyecto educativo es el Instituto de Educación Media Superior (IEMS), esto se debe a que tenemos facilidad para obtener la información que se requiere así como el asesoramiento de personal administrativo y docente en caso de dudas.
2. El nombre asignado al chatbot será *Cinvesbot - Servicio Escolar*.
3. La herramienta a utilizar es Dialogflow ES, porque nos proporciona los elementos básicos para la creación de chatbots sin hacer registros de tarjetas de crédito u otro medio de pago.
4. La manera de manejar la autenticación de los usuarios será con Firebase.
5. La interacción será mediante texto en el idioma español.
6. La integración se hará para una página web.

6.2.1. Construcción y configuración

6.2.1.1. Construcción del agente

Existen varias herramientas que se pueden emplear para la construcción de chatbots. Anteriormente, en la Figura 4.3 mostramos los mecanismos de Inteligencia Artificial que se emplearon para el desarrollo de chatbots educativos. Entre ellos destacaron Dialogflow y AIML. A grandes rasgos, Dialogflow es un marco de trabajo con infraestructura en la nube y ofrece opciones que facilitan la integración del chatbots en plataformas web o de mensajería, a diferencia de AIML que son plantillas y se requiere de un interprete.

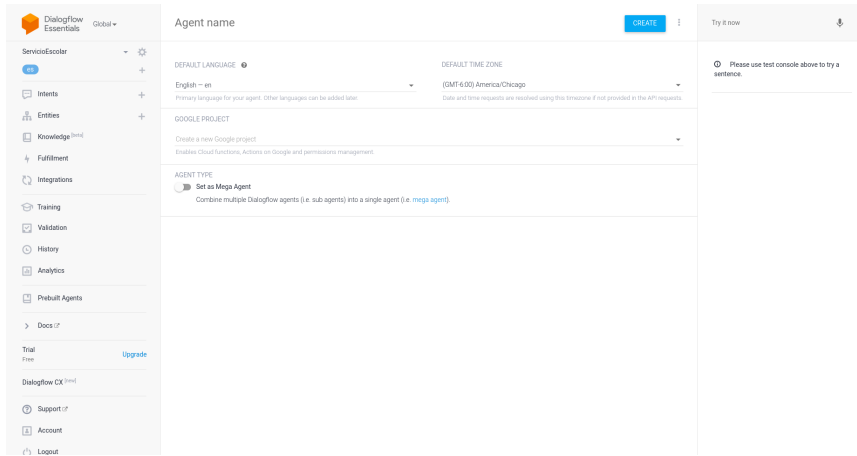
Recordando lo mencionado en el Capítulo 4, Dialogflow trabaja con elementos que se llaman *agentes*, quienes se encargan de llevar la conversación con el usuario. Estos *agentes* utilizan: *intentos* (*intents*) y *entidades* (*entities*). Los *intentos* clasifican la intención del usuario en una conversación, por este motivo un *agente* debe estar compuesto por varios *intentos*. Las *entidades* determinan cómo se van a extraer los datos de una expresión del usuario en un *intento*.

Existen dos tipos de servicios para la creación de *agentes*:

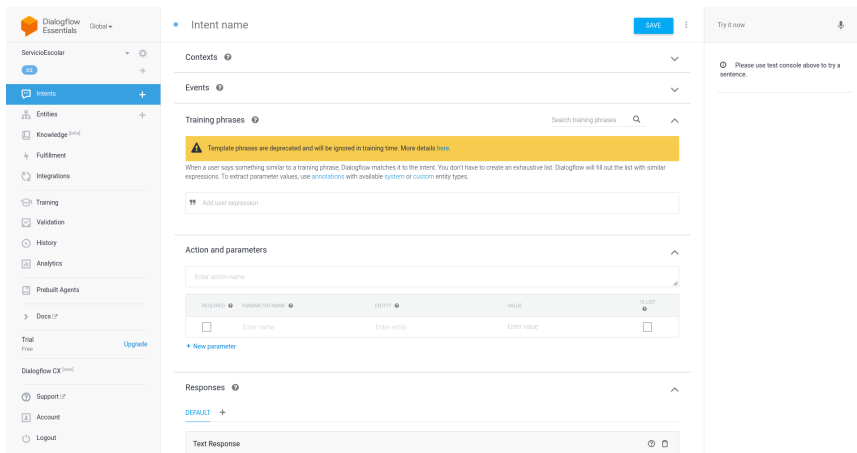
- **Dialogflow ES**, es una versión estándar para la creación de *agentes* pequeños y simples. No requiere de registro de tarjeta para facturación por lo que está limitado al número de solicitudes y transacciones mensuales.
- **Dialogflow CX**, es una versión avanzada para crear *agentes* grandes o complejos. Requiere del registro de tarjeta para facturar desde su inicio.

Para el desarrollo del chatbot elegimos Dialogflow ES. Esto nos permitirá crear los *agentes* e *intentos* necesarios para cubrir las funcionalidades básicas.

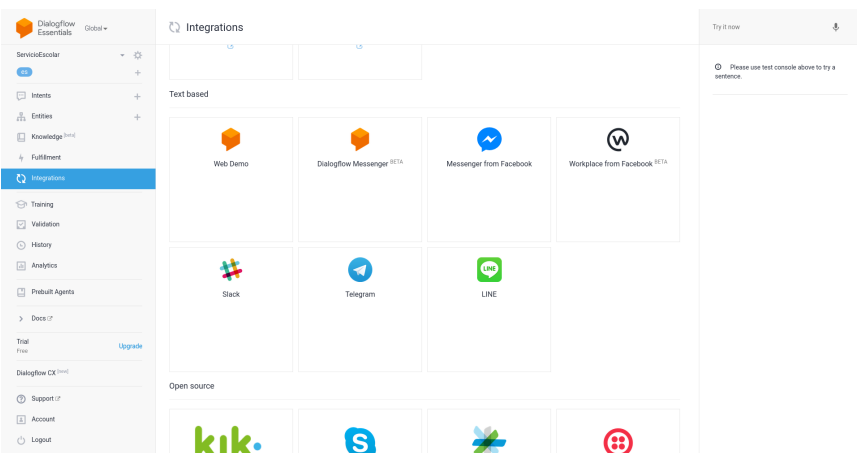
La Figura 6.1 muestra los elementos de la consola de Dialogflow: *agente*, *intentos* e *integración*. Para crear un *agente* es necesario proporcionar un nombre y elegir el idioma con el que se requiere la interacción (ver Figura 6.1(a)). Una vez creado el *agente* se deben de establecer los *intentos*. Para crear un *intento*, al menos es imprescindible, asignar un nombre, definir frases de entrenamiento y poner las respuestas que dará (ver Figura 6.1(b)). Existen más opciones como contextos, eventos, acciones y parámetros, e interacción con otros sistemas (*fulfillment*) pero estas no se abordarán de momento. La *integración* consiste en elegir la plataforma para desplegar el chatbot creado en redes sociales, telefonías, página web o plataformas de mensajería.



(a)



(b)



(c)

Figura 6.1: Consola de Dialogflow.

Para construir el chatbot es necesario definir y configurar un *agente* en Dialogflow, que en esencia es una colección de conversaciones. En este proceso, debemos asignar un nombre al *agente* y posteriormente crear varios *intentos*, que representan los objetivos que el chatbot buscará alcanzar.

Antes de definir los intentos, es recomendable hacer un diagrama para visualizar cómo estarán compuestos los *intentos*. En nuestro caso es necesario para ver si los que establecemos concuerda con la clasificación (ver Figura 5.5) y arquitectura (ver Figura 5.5). También es importante tener obtener la información que el chatbot va responder.

Antes de definir los *intentos*, se recomienda crear un diagrama para visualizar la composición de los mismos. En nuestro caso, es necesario para verificar si los *intentos* que establecemos concuerdan con la clasificación (ver Figura 4.1) y la arquitectura (ver Figura 5.5). También es importante contar con la información que el chatbot responderá a medida que le hagan preguntas.

La Figura 6.2 muestra la estructura del Agente Servicios Escolares. Este *agente* contiene los *intentos* que van a representar en gran medida a las respectivas características de un OSE. A continuación explicamos en qué consisten los intentos:

- **Bienvenida:** se encarga de dar el saludo principal y solicita el nombre del usuario. Este *intento* es de suma importancia porque es la primera interacción que tiene el usuario con el chatbot. A partir de ahí es cuando el chatbot podrá recordar el nombre y utilizarlo para cuando de las respuestas de tal manera que empezará a generar un acercamiento más personalizado con el usuario.
- **Información:**
 - **Ayuda:** muestra los temas, en forma de menú, de los cuáles podrá dar respuesta el chatbot.
 - **Plan de Estudios:** presenta la currícula de la institución.
 - **Historia:** explica la historia de la institución educativa.
 - **Planteles:** muestra todos los planteles que tiene la institución.
- **FAQ:**
 - **Beca:** muestra los requisitos para tramitar una beca de estudiante.

- **Cancelar Beca:** indica en qué condiciones se puede cancelar la beca del estudiante.
 - **Turnos:** muestra los turnos que tiene la institución educativa.
- **Procedimientos:**
 - **Certificado:** señala los requisitos para obtener el certificado del estudiante.
 - **Constancia:** explica qué es una constancia y dónde solicitarla.
 - **Credencial:** muestra cómo tramitar la credencial
 - **Bajas:** presenta los supuestos para dar de baja a un estudiante.
 - **Reinscripción:** indica las fechas y requisitos para poder reinscribirse.
 - **Seguro Médico:** proporciona información de cómo tramitar el seguro médico de un estudiante.
 - **C&H:**
 - **Calendario:** proporciona una liga con el calendario escolar.
 - **Agradecimiento:** contiene frases de gratitud hacia los usuarios.
 - **Despedida:** incluye frases para finalizar la conversación.
 - **Malas Palabras:** tiene una lista de vocabulario que el usuario debería evitar.

Los *intentos* Bienvenida, Agradecimiento, Despedida y Malas Palabras no pertenecen a ninguna características de OSE. No obstante, estos *intentos* son útiles para llevar un flujo ameno en la conversación.

La Figura 6.3, muestra los *intentos* creados en Dialogflow. Estos *intentos* son los que se encargarán de dar las respuestas acorde a lo que pregunta el usuario.

Para finalizar la creación del agente con sus respectivos intentos será necesario elegir la forma en cómo se va a presentar el chatbot, esto se le conoce como la integración.

Dialogflow ofrece varios mecanismos para hacer la integración. Estos pueden ir para redes sociales aunque en su mayoría están para plataformas de mensajería. La Figura 6.1(c) muestra algunas opciones. Nosotros elegiremos la integración con Dialogflow Messenger. El motivo principal se debe a que nosotros lo vamos a utilizar en una página web. No obstante, hay ser conscientes de que puede no funcionar correctamente ya que esta opción es

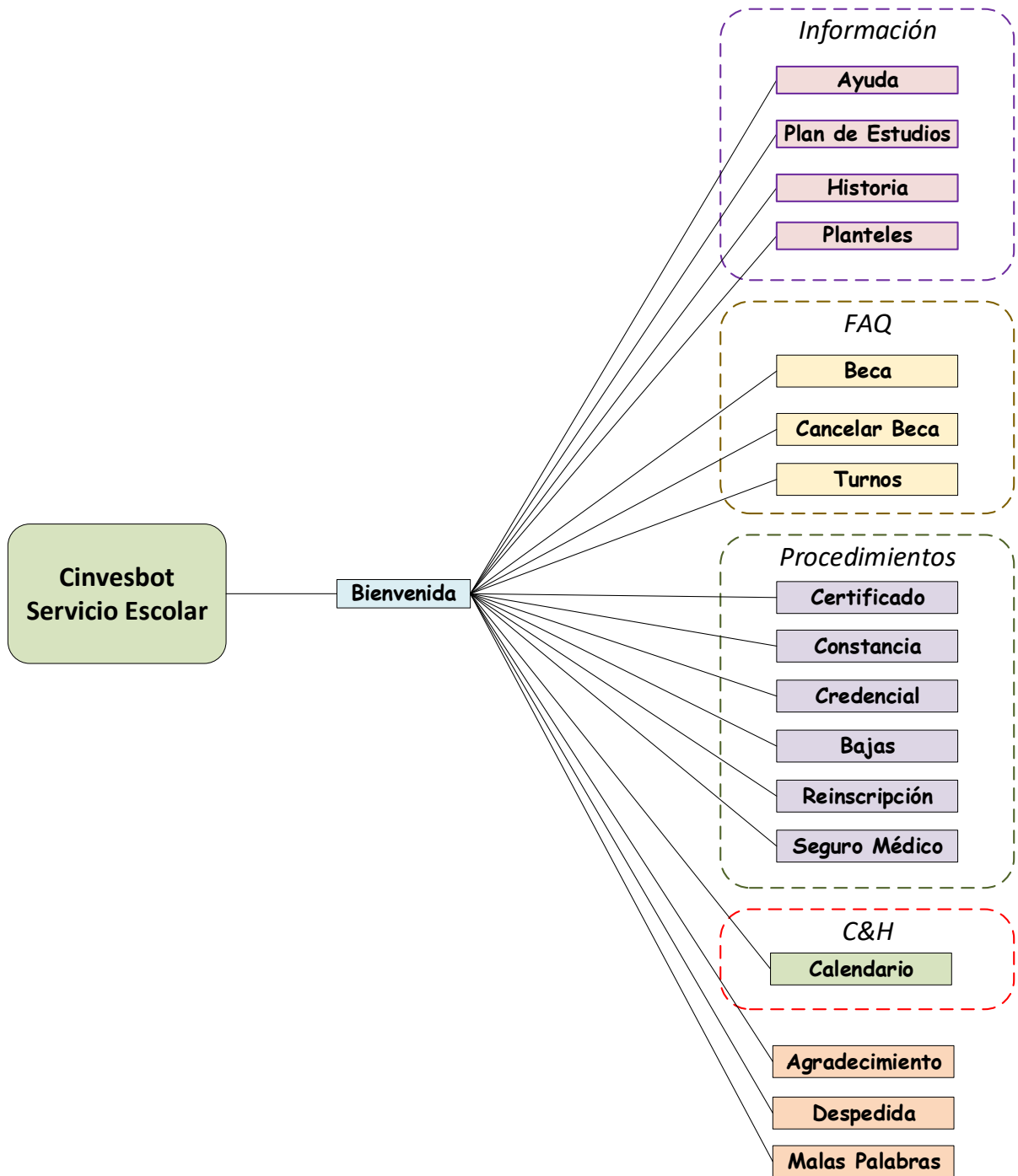


Figura 6.2: Estructura de Cinvesbot Servicio Escolar y sus respectivos *intentos*.

☰ Intents CREATE INTENT ⋮

Search intents 🔍 ⌵

OF 2 →

- 1.MensajeBienvenida ⌵
- 2.InformacionAyuda
- 2.InformacionHistoria ⌵
- 2.InformacionPlanDeEstudios
- 2.InformacionPlanteles
- 3.FAQBeca
- 3.FAQCancelarBeca
- 3.FAQTurnos
- 4.ProcedimientosBajas
- 4.ProcedimientosCertificado
- 4.ProcedimientosConstancia
- 4.ProcedimientosCredencial
- 4.ProcedimientosReinscripcion
- 4.ProcedimientosSeguroMedico
- 5.CyHCalendario ⌵
- EstructuraProblemaEje
- MensajeAgradecimiento
- MensajeDespedida
- MensajesMalasPalabras
- 🔖 MensajesNoEntiendo

OF 2 →

Figura 6.3: Lista de intentos para el chatbot Cinvesbot Servicio Escolar en Dialogflow.

aún una versión inestable (beta). El código que nos proporciona la herramienta al escoger Dialogflow Messenger es el siguiente:

```
<script src="https://www.gstatic.com/dialogflow-console/fast/messenger/  
bootstrap.js?v=1"></script>  
<df-messenger  
  intent="WELCOME"  
  chat-title="ServicioEscolar"  
  agent-id="e186c919-113e-4950-b9b2-fb23c4"  
  language-code="es">  
</df-messenger>
```

Este código lo debemos introducir en nuestra página web para que pueda aparecer. No obstante, viene con las configuraciones por defecto y, si se desea personalizar el diseño, es necesario configurarlo manualmente, lo cuál mostraremos más adelante.

6.2.1.2. Autenticación de usuario

Para implementar la autenticación de usuarios, se puede utilizar *Firebase Authentication*. Este es un servicio que combina tanto la programación como la configuración de los servicios, facilitando y agilizando el proceso. Para utilizarlo es necesario contar con una cuenta de Google.

Los pasos para lograr la configuración se esboza de la siguiente manera: primeramente, se debe crear un proyecto a través de la consola Firebase (<https://console.firebase.google.com>). A continuación, se procede a habilitar la autenticación, lo cual se realiza yendo a la consola de Firebase, seleccionando la opción *Authentication* y haciendo clic en *Habilitar*. Después, se elige el método de autenticación, eligiendo correo electrónico entre las opciones disponibles, y se finaliza completando la información solicitada.

Una vez habilitado y configurado la autenticación, se pueden empezar a crear los usuarios directamente desde la consola de Firebase. Sin embargo, la herramienta proporciona un SDK (Software Development Kit) que se puede utilizar más adelante para que mediante programación creamos, validemos y gestionemos a los usuarios, lo cuál haremos en las próximas secciones de este capítulo.

Por último, es necesario dirigirse a la configuración del proyecto y, en las opciones generales, añadir una nueva aplicación web, poner el nombre del proyecto y registrar la

aplicación. Después, se generará un código que se debe guardar para ocuparlo posteriormente. El código tendrá la siguiente apariencia:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries
// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzfsd3UKw7Cv8sdfLczHe2C3qP7ndff2F",
  authDomain: "fir-cinvesbotestprof.firebaseio.com",
  projectId: "fir-cinvesbotestprof",
  storageBucket: "fir-cinvesbotestprof.appspot.com",
  messagingSenderId: "208752313966706",
  appId: "1:39024566706:web:d097c20149d71124e7cceb"
};
// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

6.2.2. Interacción con otros sistemas

Dialogflow permite la interacción con otros sistemas o servicios, esto se le conoce como *fulfillment*, con la finalidad de realizar cosas más complejas. Sin embargo, la configuración puede resultar un tanto más complicada y confusa, ya que la documentación oficial y la información disponible en la comunidad a veces omiten algunos pasos importantes o se actualizan sin proporcionar una documentación completa.

Dialogflow ofrece una interacción con otros sistemas por medio de un cobro, y esto proporciona herramientas bastante avanzadas para hacer el *fulfillment*. Nosotros decidimos ocupar la versión estándar de Dialogflow y hacer la configuración de forma manual, por lo que es necesario utilizar otras herramientas para lograr ese fin.

Como mencionamos anteriormente, **ngrok** es una herramienta que nos permite exponer servicios locales a Internet. A esto nos referimos a que al utilizar ngrok se crea una conexión (túnel) entre la máquina local y la nube. Este túnel enruta el tráfico de Internet a través de los servidores de ngrok, lo que hace que los servicios de la aplicación local sean

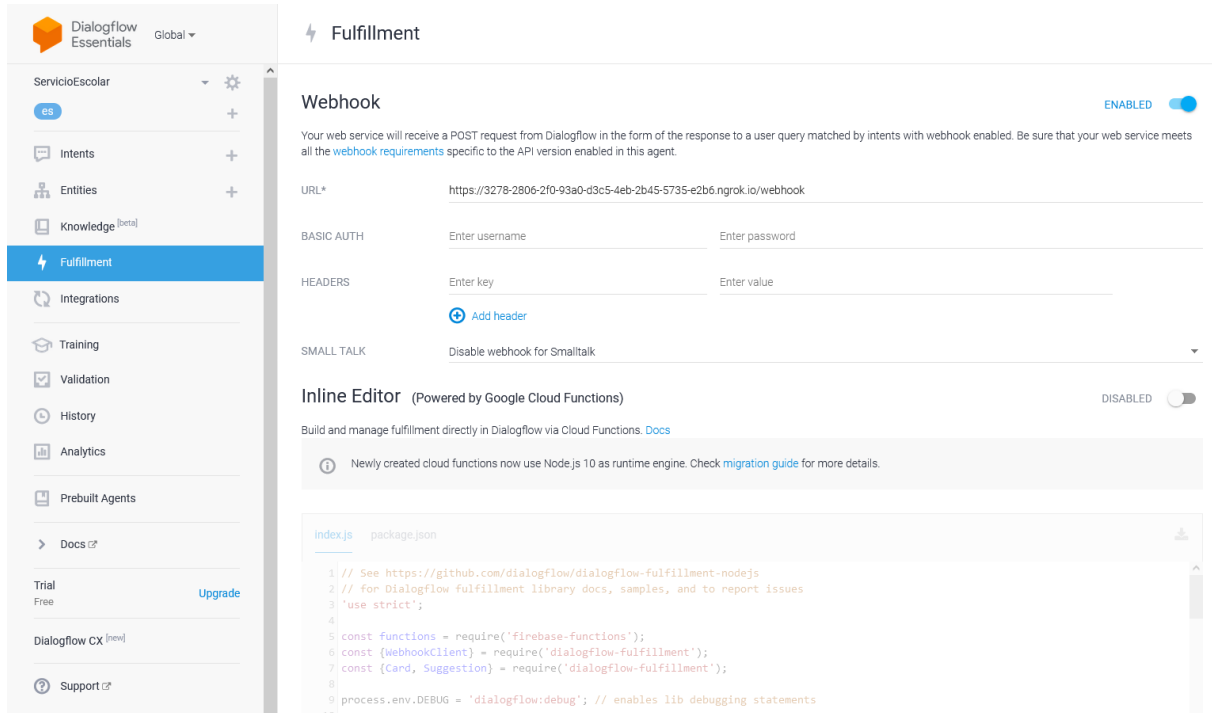


Figura 6.4: Configuración de la interacción con otros sistemas.

accesibles desde cualquier lugar. Para lograr esto se debe de instalar ngrok en la máquina local y utilizar el comando `ngrok http 8080` para crear una URL pública por donde se accederá al servicio de la aplicación local.

Para establecer el túnel con Dialogflow se debe hacer la siguiente configuración. Ir a Dialogflow. Después, elegir el agente que creamos. Nos movemos hasta la opción de *Fulfillment*. Por último, debemos activar la opción que dice *Webhook*⁹ En el campo URL del *Webhook* se debe poner la URL pública que se generó cuando se utilizó ngrok. La Figura 6.4 ilustra ese proceso.

De esta manera habremos logrado establecer el mecanismo para la interacción con otros sistemas.

⁹Es un mecanismo que se utiliza para integrar aplicaciones o servicios entre sí.

6.2.3. Programación del chatbot

Para la programación del chatbot se utilizó la arquitectura propuesta en el Capítulo 5. Para esto es importante elegir las herramientas adecuadas para la codificación. Como mencionamos anteriormente, la implementación de un chatbot OSE no requiere de funcionalidades compleja.

Con respecto al *front-end* se utilizó el lenguaje de programación Angular. Para el *back-end* se empleó Nodejs. A continuación detallamos la organización del proyecto y comentaremos sobre los componentes más representativos de este proyecto. Cabe mencionar que este proyecto fue desarrollado utilizando el IDE de Visual Studio Code.

En la Figura 6.5 podemos ver la estructura del proyecto con respecto a la parte que se desarrolló en Angular. A continuación mencionamos los aspectos más relevantes:

- El directorio *presentación* tiene el componente *iu* (interfaz de usuario). Dentro de *iu*, se encuentra un módulo denominado *login* el cuál incorpora la funcionalidad esencial de inicio de sesión.
- El directorio *negocio* cuenta con el componente *au* (acceso usuario). A su vez, podemos observar que, existen módulos como *dashboard*, *registrar-usuario*, *recuperar-password* y *verificar-correo* con su respectiva funcionalidad.
- El directorio *servicios* tiene el componente *ise* (interfaz de servicios externos), el cual presenta definiciones de los posibles errores que podrían surgir al utilizar Firebase.
- El archivo `enviroment.ts` es el espacio para la configuración integral del proyecto. Este contexto específico, contiene el código mencionado previamente (ver Subsección 6.2.1.2), el cual sirve para el correcto funcionamiento del proceso de autenticación.

Por otro lado tenemos la estructura que se desarrolló con NodeJs. A continuación describimos los aspectos importantes:

- El directorio de *servicios* tiene el componente *servicio-chatbot*. Con el objetivo de mejorar la organización, se han separado varios elementos. Primero encontramos el archivo `app.js`, que se encarga del arranque del servidor. Después, dentro de las carpetas *controllers* y *routes*, encontramos la funcionalidad asociada a las rutas, recordando que tenemos un *webhook* hacia el agente que creamos en Dialogflow.

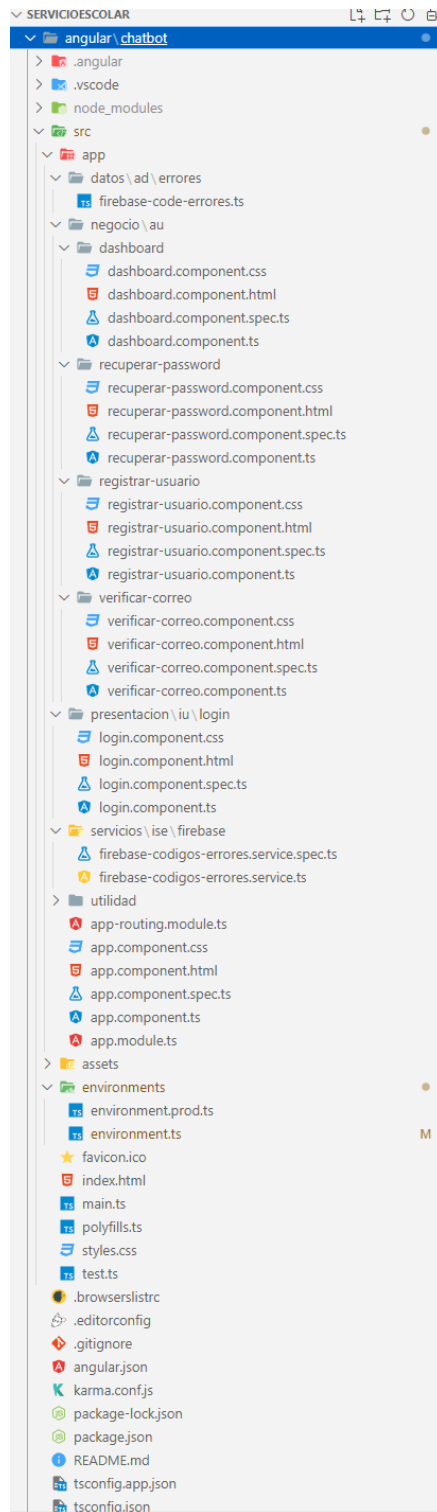


Figura 6.5: Estructura del chatbot en Angular.

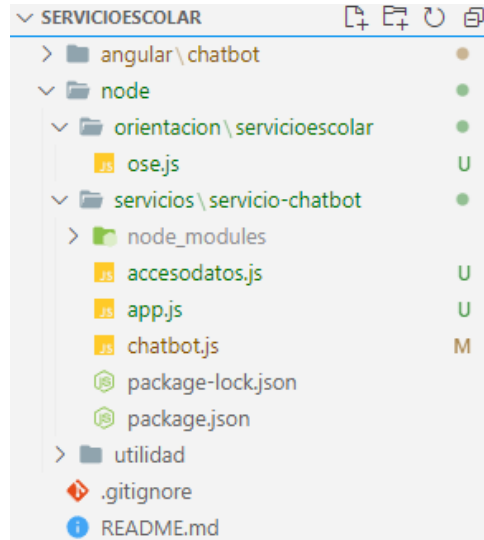


Figura 6.6: Estructura del chatbot en Nodejs.

- El directorio *orientacion* tiene el componente con las funciones correspondientes para los *intents* que definimos en Dialogflow.

6.2.4. Interfaz de usuario

Dialogflow proporciona una interfaz predeterminada para el chatbot. Para mejorar su presentación, personalizamos esta interfaz incluyendo un logotipo en el botón que abre la ventana del chatbot. Además, ajustamos los colores del título y botón del chatbot, así como el fondo de la conversación, el fondo del texto y los mensajes tanto del chatbot como del usuario. También, personalizamos el color de la fuente de texto para el chatbot y el usuario, y el ícono utilizado para enviar mensajes de texto. A continuación, se presenta un fragmento de esta personalización:

```
<style>
df-messenger {
  /* Color barra de título y botón para iniciar el chatbot*/
  --df-messenger-button-titlebar-color: #009383;
  /* Color de fondo de la conversación */
  --df-messenger-chat-background-color: #e6e6e6;
```

```

/* Color de fondo del texto de los mensajes del chatbot */
--df-messenger-bot-message: #c8e6e2;
/* Color de fondo del texto de los mensajes del usuario */
--df-messenger-user-message: #50c2b8;

/* Color fuente del texto del chatbot y el usuario */
--df-messenger-font-color: #gray;
/* Color del icono para mandar el mensaje de texto */
--df-messenger-send-icon: #878fac;
}
</style>

```

Finalmente, la Figura 6.7 se muestra cómo queda la interfaz de usuario del chatbot Cinvesbot Servicio Escolar. En un inicio el chatbot pregunta el nombre del usuario, esto con la finalidad de utiliza su nombre en las respuestas para ir generando cierta empatía con el usuario. Una vez que el usuario introduce su nombre el chatbot le indica cómo puede empezar con el flujo de la conversación. El usuario puede escribir en lenguaje natural las preguntas que requiere siempre y cuando sea acorde con la información que tiene Cinvesbot.

6.3. Discusión

La implementación puede ser una experiencia enriquecedora cuando se dispone del conocimiento y manejo de herramientas adecuadas.

En nuestro caso, la elección de herramientas no representó un obstáculo, ya que durante la revisión del estado del arte (ver Sección 4.4), identificamos algunas que podría ser de utilidad. Sin embargo, fue necesario investigar otras herramientas, como en el caso de ngrok, y cómo realizar un *webhook*.

Un aspecto que pudo obstaculizar la implementación son las herramientas que requieren un proceso de facturación. Esto es muy común cuando se utilizan plataformas con infraestructura en la nube. Sin embargo, nuestros propósitos no son comerciales y las versiones estándar que utilizamos eran suficientes para un primer prototipo.

La elección del tipo de chatbot, como se plantea en nuestra clasificación, no debe tomarse a la ligera, ya que dependiendo del enfoque, la naturaleza del proyecto puede tener

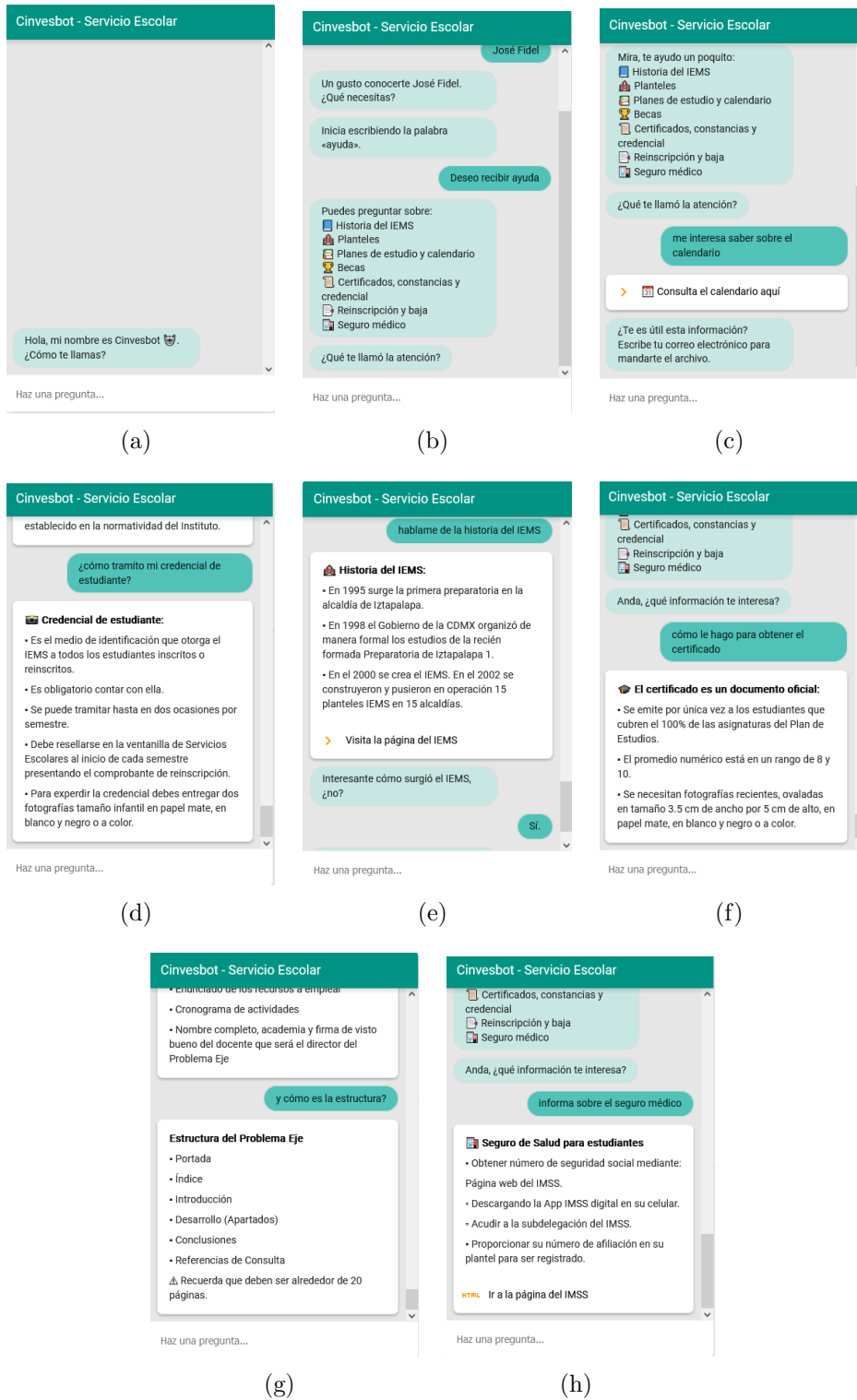


Figura 6.7: Interfaz de Cinvesbot Servicio Escolar.

implicaciones a largo plazo en el desarrollo. Para fines prácticos, la opción más ágil fue la implementación de un chatbot orientado a servicios escolares. Sin embargo, surgieron dificultades a raíz de la elección del nivel educativo en el cual se basaría el desarrollo.

Afortunadamente, contamos con acceso a información y conocimiento relacionado con el IEMS, lo que facilitó la obtención de información para el usuario. Por eso se eligió esa institución. De no haber tenido esa facilidad, el desarrollo del prototipo habría sido más prolongado. No obstante, fue necesario analizar y organizar la información para que se ajustara a la clasificación de chatbots del tipo OSE.

Por otro lado, los desarrollos que utilizan infraestructura en la nube requieren implementar tanto la funcionalidad como la configuración de servicios. Estos desarrollos agilizan las implementaciones pero introducen complejidad en la abstracción del proyecto. Aquí es recomendable que los desarrolladores cuenten con habilidades específicas en el manejo de herramientas y experiencia en proyectos similares para comprender mejor cómo se va a llevar a cabo la implementación.

La programación del chatbot se llevó a cabo utilizando diversas herramientas, ya que la arquitectura seguida no estaba específicamente vinculada a un conjunto particular de herramientas. Por un lado, el *front-end* se programó en un lenguaje, mientras que el *back-end* se desarrolló en otro.

Finalmente, la elección de la herramienta Firebase simplificó la implementación de la autenticación del usuario, eliminando la necesidad de crear una base de datos. Además, dado el enfoque y las características del chatbot OSE, tampoco fue necesario contar con una base de conocimientos.

Capítulo 7

Caso de estudio del chatbot educativo

En este capítulo, se expone el caso de estudio con el propósito de hacer pruebas con usuarios finales con respecto a la implementación del prototipo presentado en el Capítulo 6.

Aquí, buscamos someter a prueba el prototipo, para asegurarnos de que los requisitos de los usuarios hayan sido abordados de manera adecuada. Además, probamos como el diseño de la interfaz gráfica impacta en la realización de su trabajo, analizando si contribuye a mejorar la productividad y proporciona una experiencia cómoda.

Primeramente, se presenta la guía de aplicación que se planificó para este caso de estudio (cf. Sección 7.1). A continuación, se presentan las entradas que se establecieron para la guía misma (cf. Sección 7.1). Después, se muestran los resultados del caso de estudio (cf. Sección 7.3). Finalmente, se discuten los resultados (cf. Sección 7.4).

7.1. Guía de aplicación

La presente guía de aplicación tiene como finalidad ofrecer una estructura consistente y detallada para la realización de un caso de estudio destinado a evaluar un chatbot educativo. Los casos de estudio representan una herramienta valiosa y una experiencia enriquecedora para comprender la efectividad y la experiencia del usuario, en este caso un chatbot enfocado a la educación.

Siguiendo la misma dinámica del método ADD 3.0 (ver Sección 5.3), esta guía que

planteamos propone una serie de entradas esenciales que se deben seguir para asegurar una ejecución satisfactoria y exitosa para este caso de estudio. Esto comprende desde la selección de la población objetivo hasta la captura de datos relevantes y el respeto al anonimato de los participantes. Las entradas de la guía de aplicación están planteadas para ofrecer orientación clara y así, asegurar que las pruebas se realicen de manera efectiva y ética.

Las entradas propuestas son las siguientes:

1. **Población a la que va dirigido las pruebas.** Antes de comenzar, es importante identificar el público objetivo del chatbot para poder seleccionar a los participantes que sean más representativos. Los criterios para la selección de la población pueden incluir el nivel educativo, la edad, el género, los intereses y el uso de tecnologías.
2. **Espacio adecuado y equipo necesario.** Establecer si las pruebas del chatbot se pueden realizarse de manera presencial o en línea. En el caso presencial, es necesario disponer del equipo necesario, como computadoras, tabletas o teléfonos inteligentes, así como un espacio adecuado para que los participantes puedan interactuar con el chatbot sin interrupciones y que se propicie la concentración.
3. **Formularios para los participantes.** Para recopilar datos durante las pruebas, es necesario contar con un formulario preparado antes de iniciar el proceso. Este formulario puede presentarse en diversas formas, ya sea impreso o en formato digital, y debe ir acompañado de herramientas básicas como bolígrafos, lápices o dispositivos como una computadora, según la preferencia. También, se debe asegurar de que el formulario y los materiales estén listos con anticipación para garantizar una captura efectiva de la información esencial durante las pruebas.
4. **Prototipo o demo que se va a utilizar.** Antes de iniciar las pruebas, el chatbot que se va a probar debe estar en un estado de desarrollo lo suficientemente avanzado funcionalmente y accesible para que los participantes puedan interactuar con él de manera significativa. Cualquier fallo técnico debe ser resuelto antes de comenzar.
5. **Instrucciones y tareas que deberán realizar los usuarios.** Es importante explicar a los participantes las instrucciones y tareas que deberán realizar durante las pruebas. Esto ayudará a garantizar que los datos recopilados sean los más precisos posibles y fiables.

6. **Aspectos logísticos a considerar.** Es posible que la población estudiantil con la que realizaremos las pruebas tenga compromisos académicos y actividades recreativas durante la jornada escolar. Entonces, es esencial establecer un horario que no interfiera con sus responsabilidades escolares. Además, es de utilidad contar con el equipo necesario para capturar evidencia de las pruebas, *e.g.*, el tiempo, las fotos o los vídeos de las interacciones de los estudiantes con el chatbot. Esto nos permitirá analizar los resultados de manera efectiva. No obstante, es de suma importancia garantizar el anonimato de los participantes y evitar recopilar información personal innecesaria que pueda poner en riesgo su privacidad o comprometerlos de alguna manera. Este enfoque no solo protegerá sus derechos y privacidad, sino que también respaldará la integridad del estudio, especialmente teniendo en cuenta que algunos de los participantes pueden ser menores de edad.

7.2. Entradas para la guía de aplicación

7.2.1. Población

Para llevar a cabo el caso de estudio, se seleccionó el Instituto de Educación Media Superior de Ciudad de México (IEMS) como la institución para probar el chatbot educativo.

El IEMS opera en dos modalidades, la escolarizada y semiescolarizada, y consta de 28 planteles distribuidos en diversas alcaldías de la Ciudad de México: Álvaro Obregón, Azcapotzalco, Coyoacán, Cuajimalpa de Morelos, Cuauhtémoc, Gustavo A. Madero, Iztacalco, Iztapalapa, La Magdalena Contreras, Miguel Hidalgo, Milpa Alta, Tláhuac, Tlalpan, Venustiano Carranza y Xochimilco, siendo Benito Juárez la única alcaldía que no alberga un plantel. En conjunto, la población estudiantil asciende aproximadamente alrededor de 30,000 estudiantes inscritos en ambas modalidades: escolarizada y semiescolarizada.

Para este caso de estudio, se eligió el plantel “Álvaro Obregón 2 - Vasco de Quiroga”, ubicado en la Avenida Río de Guadalupe S/N, Col. El Mirador, Delegación Álvaro Obregón, C.P. 00167. Esta elección se basó en las facilidades que se tienen para seleccionar un grupo de estudiantes dispuestos a participar en las pruebas del chatbot.

Los criterios para seleccionar a los estudiantes fueron:

1. Veinte jóvenes, compuestos por igual número de mujeres y hombres.

2. Estudiantes matriculados en cualquiera de los semestres, tanto en el turno matutino como vespertino, con un enfoque prioritario en aquellos de primer semestre, ya que éstos requieren acceso a la mayor cantidad de información posible.
3. Conocimientos básicos de computación, como la capacidad de utilizar dispositivos de entrada, *e.g.*, el ratón y el teclado.
4. La edad mínima generalmente es de 14 años para los estudiantes de primer ingreso, aunque también puede haber personas adultas. En términos generales, los participantes deberán tener edades comprendidas entre los 14 y 25 años.

7.2.2. Espacio y equipo

En el plantel Vasco de Quiroga, se dispone de varios espacios para llevar a cabo las pruebas. Estas instalaciones incluyen un laboratorio de cómputo equipado con 30 máquinas que utilizan el sistema operativo GNU/Linux con una distribución de Ubuntu. Además, cuentan con aulas digitales con equipos Chromebooks, computadoras disponibles en la biblioteca y cubículos de estudio y de profesores.

Dado que se debe de minimizar las posibles interrupciones al momento de las pruebas, se tomó la decisión de llevarlas a cabo en un cubículo de profesor. Este entorno ofrece una atmósfera más tranquila y propicia para la interacción con los estudiantes.

En términos del equipo de cómputo a utilizar es una laptop con el sistema operativo GNU/Linux, distribución Ubuntu. Además, se dispuso de un monitor adicional y el aseguramiento de Internet para garantizar un entorno de pruebas adecuado sin fallas de conectividad.

7.2.3. Formulario

Para el formulario se utilizó *AttrakDiff* [259], el cuál es una herramienta comúnmente presentada en forma de cuestionario. Esta herramienta se emplea en investigaciones relacionadas con la experiencia del usuario y la usabilidad de productos, en particular en el diseño de interfaces y productos digitales. Los cuestionarios de *AttrakDiff* están diseñados para evaluar tanto la atracción como la satisfacción del usuario con un producto o sistema. En este contexto, «atracción» se refiere a la impresión general positiva que un

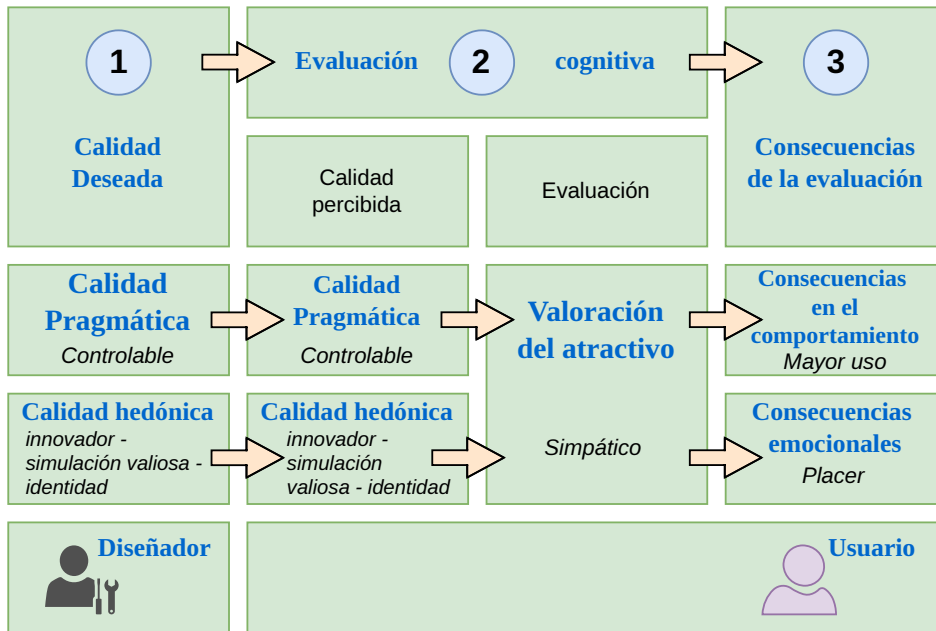


Figura 7.1: El modelo de *AttrakDiff* ilustra cómo las cualidades pragmáticas y hedónicas influyen en la percepción subjetiva del atractivo dando lugar a comportamientos y emociones consecuentes.

usuario experimenta con respecto al producto, mientras que «satisfacción» se refiere a la evaluación de qué tan bien se cumplen las expectativas del usuario.

El modelo de trabajo de *AttrakDiff*¹ se puede observar en la Figura 7.1. Este modelo consiste en cuatro componentes esenciales: (1) la calidad del producto prevista por el diseñador; (2) la percepción subjetiva de la calidad y la evaluación subjetiva de la misma; (3) las cualidades pragmáticas y hedónicas, consideradas como independientes entre sí; y (4) las consecuencias conductuales y emocionales resultantes de las interacciones anteriores.

En este modelo se proporciona una representación clara de cómo las cualidades pragmáticas y hedónicas impactan en la percepción subjetiva del atractivo, lo que a su vez desencadena comportamientos y emociones específicas.

Por otro lado, la Figura 7.2 muestra el cuestionario *AttrakDiff*, el cual utiliza escalas y preguntas específicas para medir la percepción de los usuarios en aspectos como la estética, la identidad, la originalidad y la utilidad de un producto. Este cuestionario se estructura con pares de palabras que aparentemente carecen un orden coherente en las filas, acompañados de una numeración que va del 1 al 7. La disposición de esta forma se

¹<https://www.attrakdiff.de/science-en.html>

elige para evitar sesgos en las respuestas. La representación final se puede visualizar en las figuras presentadas en la sección de resultados.

Los resultados de este cuestionario ofrecen información significativa para diseñadores y desarrolladores, ya que les ayudan a comprender la percepción de los usuarios con respecto a su producto e identificar posibles áreas de mejora. Cabe mencionar que la herramienta de *AttrakDiff* puede ser accedida a través de su página web y no tiene costo alguno por el momento.

	1	2	3	4	5	6	7	
humano								máquina
aislante								conectivo
agradable								desagradable
original								convencional
simple								complicado
profesional								aficionado
repulsivo								atractivo
práctico								impráctico
simpático								antipático
confuso								directo
elegante								desaliñado
predecible								impredecible
común								especial
entorpecedor								integrador
me acerca a la gente								me separa de la gente
impresentable								presentable
repelente								incitante
ordinario								creativo
bueno								malo
desordenado								estructurado
detestable								cautivante
audaz								cauteloso
innovador								conservador
aburrido								fascinante
sencillo								desafiante
motivante								desalentador
insólito								cotidiano
turbulento								manejable

Figura 7.2: Cuestionario *AttrakDiff* con los diferenciales semánticos.

7.2.4. Prototipo de chatbot educativo

Para llevar a cabo el caso de estudio, es esencial contar con un prototipo funcional del chatbot educativo. Como se describe en el Capítulo 6, presentamos una implementación

de un chatbot Orientado a Servicios Escolares, el cual proporciona información relevante y necesaria para los estudiantes. La razón de comenzar con un prototipo es validar, en una primera instancia, la arquitectura propuesta en el Sección 5.4 y explorar diversas herramientas que faciliten su implementación. También, otra finalidad es la de establecer una guía de aplicación para las pruebas.

7.2.5. Instrucciones y tareas

Las instrucciones para el caso de estudio deben ser sencillas y claras, especialmente considerando el tipo de población seleccionada. En este caso de estudio, al participante se le proporciona una hoja con la siguiente tarea:

- *¡Bienvenido al IEMS! Como estudiante, es fundamental que te familiarices con nuestra Institución preguntando por nuestra historia, los planteles disponibles, el plan de estudios, el seguro médico y las becas que puedes solicitar. No dudes en hacer preguntas. Recuerda que el chatbot está ahí para brindarte respuestas y orientarte en todo momento.*

Después de leer la tarea, el estudiante debe utilizar el chatbot y completarla.

Es importante aclarar que en ningún momento previo a la tarea se le instruyó o capacitó para saber cómo utilizar el chatbot.

7.2.6. Aspectos logísticos a considerar

Se estableció un calendario que abarcó del 23 de octubre al 31 de noviembre, en un horario de 11:00 a 18:00 horas, para que los estudiantes pudieran acudir al cubículo de profesor y participar en las pruebas del chatbot. Este horario se diseñó de manera que ofreciera flexibilidad a los estudiantes, permitiéndoles participar en las pruebas sin que éstas afectaran sus actividades académicas regulares. Para llevar un registro preciso del tiempo empleado en las interacciones y recopilar evidencia, se recomienda un equipo para cronometrar y una cámara o dispositivo para tomar fotografías. Un teléfono inteligente o tableta es más que suficiente.

7.3. Resultados

7.3.1. Proceso del caso de estudio

A continuación, describiremos el proceso que se siguió para llevar a cabo las pruebas. Antes de comenzar, se verificó que se contara con todas las entradas: una población de estudiantes dispuestos a participar, un espacio adecuado, el equipo necesario, el formulario, el prototipo del chatbot, instrucciones y tareas claras, así como otros detalles logísticos a considerar.

Cuando un estudiante acudía al cubículo de profesor, se le proporcionaba una breve explicación de la finalidad de su participación en las pruebas. También se le notificaba que se tomarían fotos sin exponer sus rostros para preservar su privacidad y que una persona estaría presente para asistirlos en caso de que surgieran situaciones inesperadas.

Posteriormente, se procedía a dar las instrucciones y la tarea que el estudiante debía realizar con el chatbot. El tiempo transcurrido se medía de manera discreta para que los estudiantes no se sintieran presionados durante la interacción.

Una vez completada la tarea, se les explicaba el cuestionario *AttrakDiff* que debían contestar. Además, se les informaba que en el reverso de la hoja tenían la opción de agregar comentarios o sugerencias basados en su experiencia utilizando el chatbot.

La Figura 7.3, muestra un grupo de estudiantes interactuando con el chatbot para realizar la tarea propuesta.

El tiempo registrado desde el inicio hasta la finalización de la interacción de cada estudiante con el chatbot se puede observar en el Cuadro 7.1. El periodo más breve se situó en 2 minutos y 31 segundos, mientras que el más extenso alcanzó los 11 minutos y 30 segundos. En total, las veinte interacciones sumaron 125 minutos y 15 segundos.

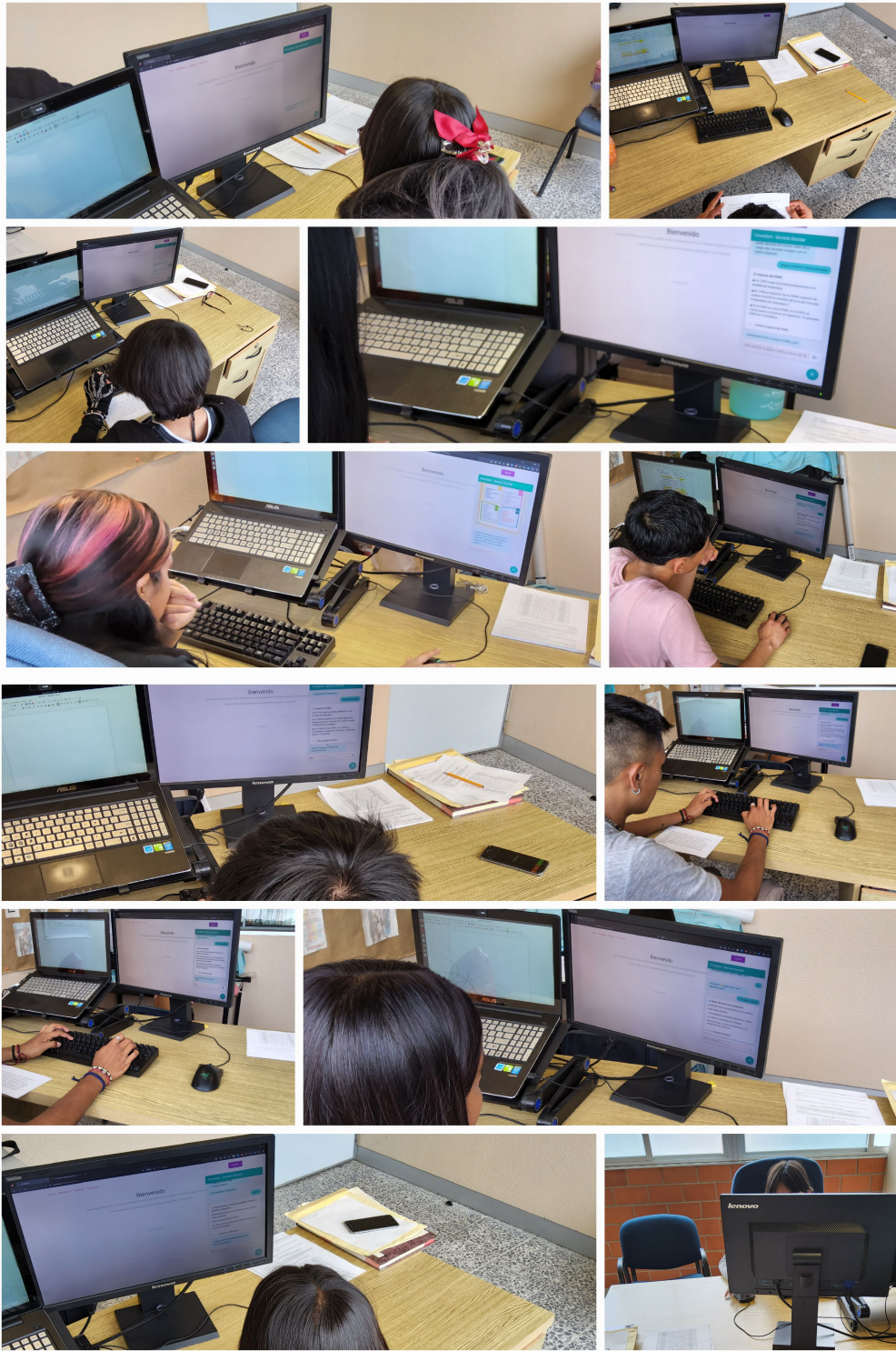


Figura 7.3: Estudiantes interactuando con el chatbot.

Cuadro 7.1: Tiempo en minutos de las interacciones de los estudiantes con el chatbot.

Estudiante	Tiempo	Estudiante	Tiempo
1	2:31	11	5:43
2	3:04	12	6:39
3	3:08	13	7:02
4	3:09	14	8:10
5	3:34	15	8:31
6	3:36	16	9:37
7	3:56	17	9:45
8	4:32	18	10:22
9	5:11	19	11:24
10	5:31	20	11:30

7.3.2. Cuestionario de AttrakDiff

La Figura 7.4 muestra el portafolio de resultados. Este portafolio está compuesto por un plano cartesiano. El eje vertical representa la Calidad Hedónica (*Hedonic Quality*, HQ), y el eje horizontal representa la Calidad Pragmática (*Pragmatic Quality*, PQ). En función de los valores de estas dimensiones, el producto se puede situar en una o varias regiones, como demasiado orientado a sí mismo, orientado a sí mismo, deseado, neutro, orientado a las tareas, superfluo o demasiado orientado a las tareas.

Se pueden identificar dos tipos de figuras. Un cuadrado azul, situado en el centro de un rectángulo translúcido, ese es el producto. También, un rectángulo más grande en color azul pero translúcido, ese se le conoce como el rectángulo de confianza y es el que nos interesa explicar con más detalle.

El rectángulo de confianza, muestra el grado de coincidencia en la evaluación del producto por parte de los usuarios. Si el rectángulo es grande, es menos probable que los usuarios coincidan en la región a la que pertenece el producto. Por el contrario, si el rectángulo de confianza es pequeño, es más probable que los usuarios coincidan en la región. Finalmente, con esto podemos ver la variabilidad de las valoraciones de los usuarios con respecto al producto.

La Figura 7.5 muestra el diagrama de valores medios. En esta presentación, hay que aclarar que la calidad hedónica se divide en dos aspectos: estimulación e identidad. Por lo



Figura 7.4: Portafolio de resultados.

que, *AttrakDiff* se compone de cuatro dimensiones en este caso:

1. Calidad Pragmática (PQ): Mide la facilidad de uso y la eficacia del producto para cumplir los objetivos del usuario.
2. Calidad Hedónica-Identidad (HQ-I): Mide la capacidad del producto para hacer que el usuario se sienta identificado con él.
3. Calidad Hedónica-Estimulación (HQ-S): Mide la capacidad del producto para generar placer o satisfacción en el usuario.
4. Atractivo (ATT): Mide la percepción general de la calidad del producto.

La Figura 7.6 exhibe los valores medios de los pares de palabras (ver Figura 7.2) . Es importante destacar que los valores extremos indican las características que son particularmente críticas o que han sido especialmente bien abordadas.

Con respecto a los comentarios que dejaron los participantes tenemos los siguientes:

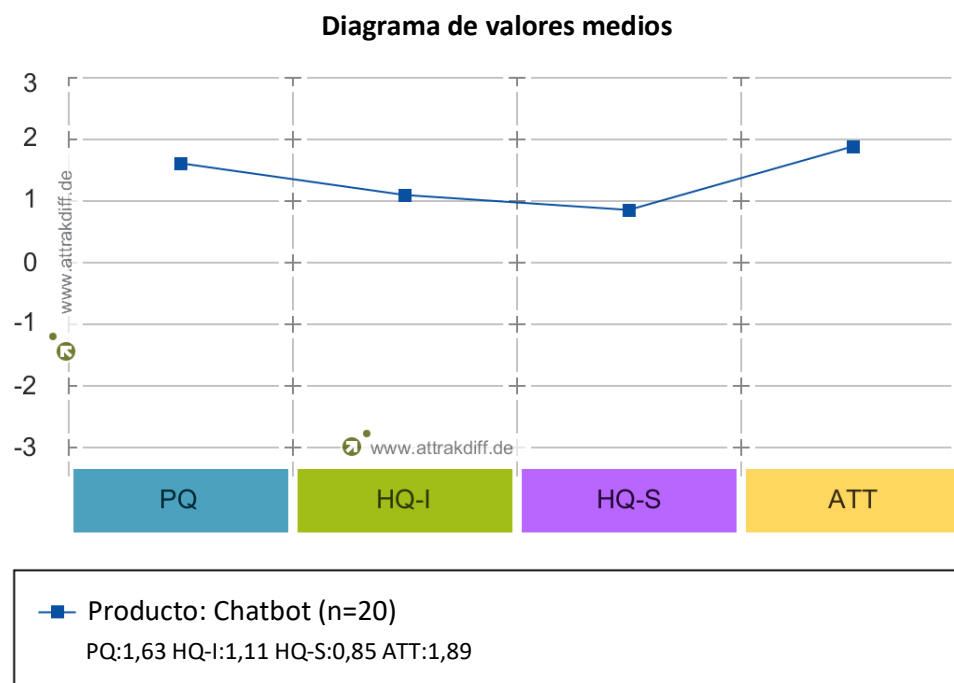


Figura 7.5: Valores medio de los estudiantes.

Descripción de palabras - pares

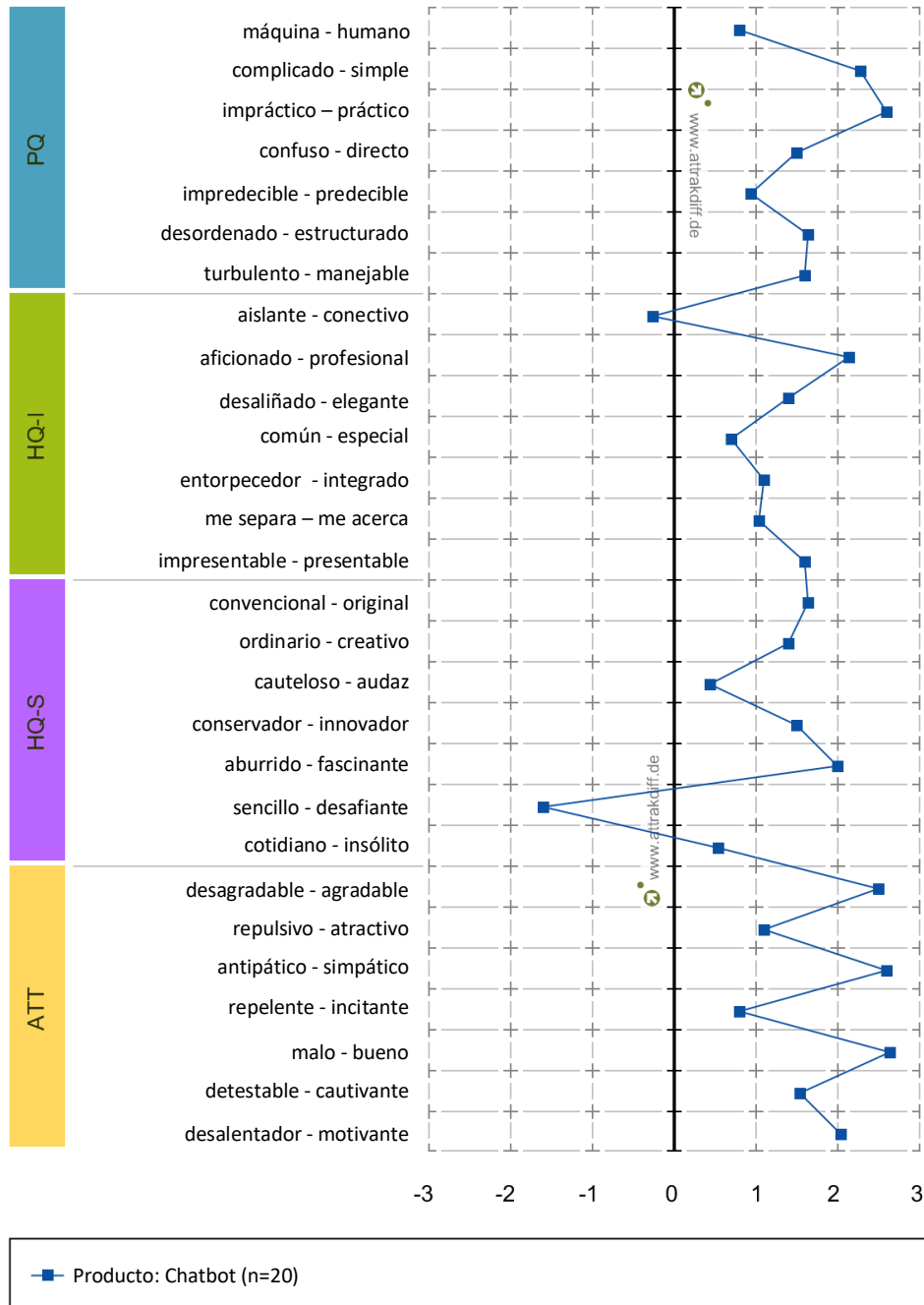


Figura 7.6: Media de la descripción de palabras pares.

1. “Me gustó, porque te saca muchas respuestas buenas, y te ayuda a resolver preguntas.”
2. “Al momento de hacer una pregunta exacta no la contesta directamente sino que te da opciones para buscarlo en otro lado.”
3. “Me parece algo muy innovador y creativo para irse acomodando y acoplando a las nuevas tecnologías.”
4. “En mi opinión, me gustó porque no me hace bolas, me dice las cosas bien y rápido, me explica, me ayuda a entender más y era algo que no había visto antes. Me gustó para que fuera la primera vez que lo uso.”
5. “A mi me parece que sí me sirve como para saber más. Hasta eso sí te da la información que quieres y muy completa.”
6. “En general es muy bueno, y tiene una manera muy rápida al contestar, sería muy recomendable para que los estudiantes la utilicen.”
7. “Me pareció increíble porque así puedo saber información sobre la historia de algo que quiero aprender. Sin embargo, me confundió un poco pero eso no importa para mí, si lo utilizaría diario sería muy útil.”
8. “Me gustó, ya que como muchos estudiantes, no sabemos cómo llegar a otra página, por ejemplo, al calendario escolar, la página del IMSS. Nos lleva directamente desde el chat, no da un manual y me parece sencillo y concreto. Me gustaría saber sobre cuándo empieza a correr o más bien se activa el seguro a los estudiantes.”
9. “Me pareció muy creativo, ya que ayuda a los estudiantes en su proceso académico resolviendo sus dudas.”
10. “Es una herramienta que acerca más al conocimiento y te motiva.”

7.4. Discusión

La etapa final de la metodología propuesta en la Sección 1.5.5 se completa con la guía de aplicación. Esta contribución es sencilla, pero resulta fundamental para estructurar de

manera sistemática las pruebas, proporcionando una herramienta para minimizar posibles problemas y errores que puedan suscitarse. En nuestro caso de estudio, la implementación de la guía se llevó a cabo de manera ordenada, facilitando significativamente el flujo de trabajo entre cada estudiante, considerando que eran 20 participantes, muchos de ellos en la adolescencia y con desafíos personales. Por lo tanto, esta guía sirve de orientación para ayudar y facilitar a investigadores para llevar a cabo un caso de estudio relacionado con chatbots y educación.

Las entradas fue un punto clave y de mucha utilidad. La incorporación de elementos de entrada para la guía de aplicación surgió a partir del método ADD 3.0. Al reconocer la eficacia de este tipo de elementos, decidimos retomar la idea e integrarlos en nuestra guía, lo cual consideramos un acierto. Esto nos permitió preparar y organizar todos los elementos necesarios con anticipación al inicio de las pruebas.

En relación con el cuestionario *AttrakDiff*, comencemos analizando la Figura 7.4. El rectángulo de confianza muestra poca dispersión, exhibiendo una uniformidad que se asemeja a un cuadrado. La explicación principal de este resultado se debe a que las calificaciones de los usuarios fueron similares. Además, podemos afirmar que el chatbot se sitúa entre un producto deseado y un producto orientado a las tareas. Esta ubicación no representa una experiencia negativa, ya que proporciona a los participantes la impresión de que el chatbot está ligeramente orientado a la resolución de tareas, pero acercándose al cuadrante deseable.

En relación con las escalas de las figuras siguientes, estas se encuentran en un rango de -3 a 3, lo cual implica lo siguiente. Desde -3 a 0, se refiere a valores negativos, indicando que a los usuarios el producto les podría parecer poco atractivo o desagradable. El valor 0 señala una posición neutral o indiferente. Los valores positivos, de 0 a 3, indican que los usuarios podrían encontrar atractivo el producto.

La Figura 7.5 ilustra un análisis general de las dimensiones evaluadas. Se destaca que ninguna dimensión registra valores negativos, lo que refleja un nivel generalmente aceptable en las cualidades del chatbot. Al profundizar en el análisis, observamos que la dimensión PQ sobresale por sus valores elevados, indicando una alta eficacia y facilidad de uso del chatbot, lo que facilita que los usuarios alcancen sus objetivos. En cuanto a HQ-I, se percibe una identificación positiva del usuario con el chatbot. Sin embargo, HQ-S muestra los valores más bajos, sugiriendo que, aunque no se logra una satisfacción plena, se mantiene en un nivel aceptable al ubicarse en el rango positivo. Es importante destacar

que valores cercanos al +3 son ideales, pues sugieren un buen desempeño con margen para mejora. Por último, la dimensión ATT registra los valores más altos, lo que implica que el chatbot resultó atractivo para los participantes.

Finalmente, examinamos la Figura 7.6. En ella, se destacan datos interesantes. Solamente 2 diferenciales semánticos presentan valores desfavorables, oscilando entre 0 y -2, los cuales son: aislante y sencillo. La percepción de los estudiantes sugiere que el producto tiende a ser aislante con respecto a otras cosas, aunque lo consideran sencillo. Tal vez esperaban algo más sofisticado y desafiante. Aunque este último diferencial semántico podría interpretarse como algo negativo, consideramos que un chatbot no debe ser complejo; debe ser algo simple de usar y que se pueda utilizar desde el inicio sin requerir conocimientos avanzados de computación. A pesar de que *AttrakDiff* no ofrece definiciones específicas para cada par semántico, su propósito principal es medir cómo el cumplimiento de necesidades básicas influye en las experiencias positivas de los usuarios con productos y tecnologías interactivas [260].

Por otro lado, encontramos valores positivos entre 0 y 1 en los diferenciales semánticos: humano, predecible, especial, audaz, insólito e incitante. Continuando, observamos el mayor número de diferenciales semánticos está con valores del 1 al 2: directo, estructurado, manejable, elegante, integrado, me acerca a la gente, presentable, original, innovador, atractivo y cautivante. Por último, están aquellos que tienen valores entre 2 y 3: simple, práctico, profesional, fascinante, agradable, simpático, bueno y motivante.

La gran mayoría de los diferenciales semánticos son positivos, con respecto a los negativos, los cuales son casi mínimos. Esto indica que el producto es considerablemente aceptable, tomando en cuenta que es el primer prototipo de chatbot educativo. No obstante, lo ideal sería que todos los diferenciales semánticos estuvieran lo más cercano posible al valor de 3. Por lo tanto, se debe trabajar, en primera instancia, en aquellos que resultaron negativos y posteriormente en aquellos que se encuentran en un rango de 0 a 1.

El tiempo de duración no fue un factor de mucha relevancia en nuestro caso de estudio, aunque observamos variaciones entre los estudiantes. Algunos completaron la tarea rápidamente, mostrando destrezas avanzadas en el uso de la computadora y la paquetería. Estos participantes emplearon estrategias eficientes, como formular preguntas más extensas al principio y luego optar por palabras clave para agilizar el proceso. En casos muy aislados, algunos estudiantes no leyeron toda la información presentada, cumpliendo su tarea al formular preguntas y obtener las respuestas correspondientes.

Por otro lado, hubo estudiantes que se tomaron el tiempo necesario para leer detenidamente toda la información. Este grupo, en general, siguió reglas ortográficas al hacer preguntas, utilizando correctamente los signos de interrogación y la acentuación. También, les fue de interés saber más acerca de cierta información ya que la desconocían, *e.g.*, el estudiante que quería saber más sobre su seguro médico.

Una gran mayoría de estudiantes se sorprendieron al descubrir que la interacción se realizaba en lenguaje natural, sin requerir comandos o instrucciones específicas. Al principio, algunos titubearon al recibir la bienvenida del chatbot y que este se presentara y les preguntara su nombre. Hubo cierta indecisión sobre si debían proporcionar sus nombres completos incluyendo apellidos, ya que al estar en un ambiente escolar es normal poner nombre y apellido, o comenzar directamente con las tareas. Por otro lado, es notable que no se registró ningún caso en el cuál los estudiantes hicieran un uso indebido del lenguaje, a pesar de que en algunas ocasiones las respuestas no fueran favorables.

Cabe mencionar, que resulta interesante destacar que alrededor de tres o cuatro estudiantes expresaron su agrado por el chatbot mediante palabras de agradecimiento, y el chatbot respondió a la cortesía de manera correspondiente, incluso mencionando sus nombres. Este aspecto específico, se previó durante la implementación del chatbot, donde los usuarios expresaban gratitud por las respuestas, y el chatbot respondía de manera positiva, personalizando la interacción.

La velocidad de respuesta del chatbot no generó ningún inconveniente, ya que responde de manera instantánea, lo cual cubre el escenario de que puede ofrecer respuestas en un tiempo inferior a 3 segundos. En un momento se planteó la posibilidad de que esta rapidez pudiera dar la impresión de que el producto se asemejaba más a una máquina, sin embargo, al observar la Figura 7.6, en el par de palabras *máquina - humano* notamos que no presenta valores negativos y se percibe más como humano.

Los escenarios prioritarios que se abordan de manera natural son aquellos relacionados con AC-12, AC-13, AC-16 y AC-17 (ver Cuadro 5.3). Los escenarios AC-12, AC-13 y AC-16 se gestionan mediante el control de acceso. Por otro lado, el escenario AC-17 se aborda a través de la interfaz gráfica de usuario proporcionada. Sin embargo, aún quedan pendientes algunos escenarios por evaluar, como AC-3, AC-4 y AC-7. Los escenarios AC-3 y AC-4 podrían evaluarse cuando se implemente un chatbot orientado a estudiante/profesor y se levante en un servidor, lo cual queda planteado como trabajo futuro. Respecto a AC-7, su evaluación se podría realizar al agregar el componente de platafor-

mas de mensajería e incorporarlo directamente con la plataforma Telegram mediante las opciones de integración de Dialogflow.

Respecto a los problemas identificados durante la ejecución del chatbot, se destacan los siguientes: cuando las preguntas eran extensas, es decir, con aproximadamente ocho palabras o más, el chatbot proporcionaba respuestas incorrectas. Asimismo, al realizar preguntas y obtener respuestas, en algunos casos, cuando se buscaba información más específica de la respuesta dada, el chatbot proporcionaba información incorrecta. Por último, se observó que al formular preguntas compuestas, es decir, con dos o más preguntas en una, el chatbot solo respondía a una de ellas.

En el contexto de nuestro estudio, la Escala de Usabilidad de Sistemas (*System Usability Scale*, SUS) emerge como otra herramienta potencialmente útil. Centrada en la usabilidad, SUS evalúa las dimensiones clásicas de eficacia, eficiencia y satisfacción. SUS es reconocida por su facilidad de administración, validez en resultados con muestras pequeñas y su habilidad para diferenciar entre sistemas usables y no usables. Sin embargo, su sistema de puntuación puede ser complejo y no es diagnóstico, enfocándose en clasificar la facilidad de uso de un sitio, aplicación o entorno [261].

Adicionalmente, el Cuestionario de Satisfacción de la Interacción del Usuario (*Questionnaire For User Interaction Satisfaction*, QUIS 7.0), está especialmente diseñado para medir la satisfacción del usuario en interacciones con sistemas. QUIS presenta dos grandes retos: primero, la necesidad de obtener una licencia para su uso [262], y segundo, su capacidad de ser personalizado para adaptarse a contextos específicos. Aunque esta personalización ofrece flexibilidad, también puede incrementar la complejidad del estudio y potencialmente impactar en la validez de los resultados [263].

De esta manera, para una evaluación exhaustiva que abarque no sólo la usabilidad sino también la experiencia del usuario en su conjunto, se requiere una herramienta más holística. En este sentido, *AttrakDiff* se posiciona como una opción superior para nuestro estudio. *AttrakDiff* permite evaluar simultáneamente aspectos pragmáticos y hedónicos, proporcionando una comprensión más integral de la interacción del usuario con el sistema [260], siendo una de las herramientas más utilizadas para la evaluación de la experiencia del usuario [264]

Capítulo 8

Conclusiones y trabajo futuro

8.1. Contribuciones principales

La contribución principal de este trabajo de tesis es la definición de una arquitectura de software para chatbots enfocados en la educación. A lo largo de esta investigación, se descubrieron otras contribuciones significativas que enriquecieron considerablemente este trabajo:

1. Esta tesis muestra el desarrollo de un Estudio de Mapeo Sistemático (SMS) con respecto a los chatbots educativos (ver Capítulo 3). Para ello se formulan tres preguntas de investigación que sirven como guía para el SMS. Asimismo, se crea una cadena de búsqueda para obtener los resultados seguido de un criterios de inclusión, exclusión y selección que funcionan como un filtro para obtener una muestra de artículos.
2. Se presenta un estado del arte (ver Capítulo 4) completo identificando contribuciones, características, aspectos técnicos y líneas de investigación para chatbots educativos. Este análisis ofrece una visión integral y actualizada de la realidad de estas herramientas.
3. Las definiciones son fundamentales para brindar claridad y evitar malos entendidos de manera precisa del significado de las cosas. En este contexto, esta tesis ofrece una definición específica de lo que es un chatbot educativo (ver Subsección 3.7.1).

Este enfoque es importante para comprender las características y los alcances que podrían tener este tipo de herramientas educativas.

4. Esta tesis presenta una propuesta de clasificación para los chatbots en un dominio educativo (ver Subsubsección 4.1.2), simplificando la categorización de estas herramientas según sus características y funciones particulares. Esta clasificación abarca chatbots que son orientados a servicios escolares, educación en línea y estudiante/profesor. Los Orientados a Servicios Escolares proporcionan *información, calendario y horarios, FAQ y procedimientos*. Los Orientados a Educación en Línea incluyen *cursos, sistemas de gestión de contenidos de aprendizaje* y software específico. Por último, están los Orientados a Estudiante/Profesor que comprenden evaluación, retroalimentación, reportes, materias, salud bienestar, preguntas y respuestas, tutorías y soporte. Además, se detalla el proceso para la obtención de la clasificación mediante el método específico de Nickerson *et al.*
5. En este trabajo, se establece una arquitectura genérica de los chatbots enfocados a la educación (ver Capítulo 5), proporcionando un marco conceptual que facilita su implementación. Esta arquitectura se presenta de dos formas: primero, como una arquitectura de tres niveles, que incluye el Nivel Cliente, el Servidor a Nivel de Aplicación y el Nivel de Servicios. Segundo, como una arquitectura con capas de: Presentación, Negocio, Orientación, Servicios, Datos y Utilidad. Además, se detallan una serie de entradas que son un requisito para la aplicación del método de diseño ADD 3.0.
6. Esta tesis presenta la implementación de un prototipo funcional de un chatbot educativo utilizando la arquitectura propuesta, específicamente diseñado para estudiantes de educación media superior en Ciudad de México (ver Capítulo 6). Se detallan las herramientas utilizadas para la programación, los servicios de la nube y sus configuraciones correspondientes. Además, se proporciona una visión de la interfaz de usuario del chatbot.
7. Esta tesis presenta una guía sencilla compuesta por una serie de entradas para llevar a cabo un caso de estudio de manera efectiva.

8.2. Trabajo futuro

Dado el alcance y el tiempo de desarrollo de esta tesis, se identifican diversas áreas que requieren atención para trabajo futuro.

Con respecto al SMS, aplicar nuevamente el SMS para obtener las propuestas chatbots educativos que han sido propuestos a partir del año 2023. Además, se puede considerar la inclusión de nuevas preguntas de investigación que permitan identificar aquellos chatbots que implementaron algún modelo educativo y métodos de aprendizaje.

Actualizar la clasificación de chatbots educativos para agregar nuevas categorías y características. Este proceso puede realizarse nuevamente utilizando el método propuesto por Nickerson *et al.*

Mejorar la arquitectura del chatbot educativo mediante una nueva iteración con el método ADD 3.0. Además, considerar agregar componentes para la gestión de usuarios, así como el contenido en la base de datos y la base de conocimientos. También, se puede explorar añadir un componente que permita al chatbot aprender con las interacciones de los usuarios, así como la incorporación explícita de un componente para la interacción por voz. De manera similar, un análisis para nuevos atributos de calidad que mejoren la arquitectura. Paralelamente, agregar portabilidad para otros posibles uso, no limitándose solo al ámbito educativo. También, evaluar la posible integración de un avatar e interacción multilenguaje. Por último, se sugieren pruebas de la arquitectura con desarrolladores.

Diversificación de tecnologías y herramientas para la implementación del chatbot. En el caso de usar *agentes e intentos*, proponer un estándar para definir nombres que faciliten su identificación y organización. También, generar los manuales correspondientes con información para configurar los servicios que se utilizan y documentar la primera versión del chatbot, publicándola en su respectivo repositorio.

Realizar implementaciones de chatbots educativos orientados a estudiante/profesor, abarcando diversas materias dentro de una institución educativa.

En el caso de estudio, se puede explorar otras herramientas para la evaluación que permitan ver otros aspectos adicionales del chatbot, *e.g.*, el Sistema de Escalas de Usabilidad o el Cuestionario de Satisfacción de la Interacción del Usuario.

Con el aspecto pedagógico, se sugiere desarrollar una herramienta para medir el impacto del chatbot en la educación. También, se plantea identificar nuevas líneas de investigación para los chatbots educativos, además de las que ya mencionamos anteriormente.

Los servicios de chatbots en la nube ofrecen una tarifa dependiendo el número de usuarios, las sesiones, las interacciones o el número de palabras del usuario con respecto al chatbot. Se sugiere realizar un estudio para conocer los costos reales de una implementación de chatbot educativo utilizando estos servicios.

Finalmente, se espera la publicación de los siguientes artículos derivados de este trabajo de tesis: *Towards an Architecture for Educational Chatbots*, *Educational-focused Chatbots: Systematic Mapping Study*, *Experiencias en el Desarrollo de Chatbots Educativos* y *Explorando Efectividad en la Educación: Un Caso de Estudio de un Chatbot en el Instituto de Educación Media Superior CDMX*.

Publicaciones del autor

1. **José Fidel Urquiza Yllescas**, Sonia Mendoza, José Rodríguez, and Luis Martín Sánchez-Adame “*Chatbots: llegaron para quedarse*”, *Revisa Avance y Perspectiva*, Volumen 8 - Número 4, Enero 2023,
<https://avanceyperspectiva.cinvestav.mx/chatbots-llegaron-para-quedarse/>
2. **José Fidel Urquiza-Yllescas**, Sonia Mendoza, José Rodríguez, and Luis Martín Sánchez-Adame, “*An approach to the classification of educational chatbots*”, In *Journal of Intelligent & Fuzzy Systems*, 43(4):5095-5107, IOS Press, August 2022,
DOI:10.3233/JIFS-213275.
3. Luis Martín Sánchez-Adame, Sonia Mendoza, **José Fidel Urquiza-Yllescas**, and José Rodríguez, Almicar Meneses Viveros, “*Towards a Set Heuristics for Evaluating Chatbots*”, In *IEEE Latin America Transactions* vol. 19, no 12, pp. 2037-2045, December 2021, *DOI:10.1109/TLA.2021.9480145*.
4. Luis Martín Sánchez-Adame, **José Fidel Urquiza-Yllescas**, and Sonia Mendoza, “*Measuring Anticipated and Episodic UX of Tasks in Social Networks*”, In *Applied Sciences-Basel*, Special Issue User Experience for Advanced Human-Computer Interaction, 10(22):8199, MDPI, November 2020, *DOI: 10.3390/app10228199*.

Bibliografía

- [1] Turing, “I.—COMPUTING MACHINERY AND INTELLIGENCE,” *Mind*, vol. LIX, no. 236, pp. 433–460, 10 1950.
- [2] B. Shawar and E. Atwell, “Chatbots: are they really useful?” *Journal for Language Technology and Computational Linguistics*, vol. 22, no. 1, pp. 29 – 49, 2007, <http://eprints.whiterose.ac.uk/81655/>. Accessed 15 August 2020.
- [3] J. Weizenbaum, “Eliza—a computer program for the study of natural language communication between man and machine,” *Commun. ACM*, vol. 9, no. 1, p. 36–45, Jan. 1966.
- [4] R. Dale, “The return of the chatbots,” *Natural Language Engineering*, vol. 22, no. 5, p. 811–817, 2016.
- [5] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing*, C. Linguistics and S. R. (1st ed.), Eds. Prentice Hall PTR, 2000.
- [6] B. Ranoliya, N. Raghuwanshi, and S. Singh, “Chatbot for university related faqs,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India, Sep 2017, pp. 1525–1530.
- [7] S. Reshmi and K. Balakrishnan, “Implementation of an inquisitive chatbot for database supported knowledge bases,” *Journal Sādhanā*, vol. 41, no. 10, pp. 1173–1178, Oct 2016.
- [8] A. Inokuchi, H. Tamada, H. Hata, and M. Tsunoda, “Toward obliging bots for supporting next actions,” in *2016 4th Intl Conf on Applied Computing and Information*

Technology/3rd Intl Conf on Computational Science/Intelligence and Applied Informatics/1st Intl Conf on Big Data, Cloud Computing, Data Science Engineering (ACIT-CSII-BCD), 2016, pp. 183–188.

- [9] A. Veglis and T. A. Maniou, “Embedding a chatbot in a news article: Design and implementation,” in *Proceedings of the 23rd Pan-Hellenic Conference on Informatics*, ser. PCI ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 169–172.
- [10] C. Matthies, F. Dobrigkeit, and G. Hesse, “An additional set of (automated) eyes: Chatbots for agile retrospectives,” in *Proceedings of the 1st International Workshop on Bots in Software Engineering*, ser. BotSE ’19. IEEE Press, 2019, p. 34–37.
- [11] S. Sulyman, “Client-server model,” *IOSR Journal of Computer Engineering*, vol. 16, pp. 57–71, 01 2014.
- [12] R. Kar and R. Halidar, “Applying chatbots to the internet of things: Opportunities and architectural elements,” *International Journal of Advanced Computer Science and Applications*, vol. 7, 11 2016.
- [13] F. Marcondes, J. Almeida, and P. Novais, “Chatbot theory,” in *Intelligent Data Engineering and Automated Learning – IDEAL 2018*, H. Yin, D. Camacho, P. Novais, and A. J. Tallón-Ballesteros, Eds. Cham: Springer International Publishing, 2018, pp. 374–384.
- [14] H. Levy. (2016, Oct.) Gartner predicts a virtual world of exponential change. <https://www.gartner.com/smarterwithgartner/gartner-predicts-a-virtual-world-of-exponential-change>. Accessed 1 May 2020.
- [15] C. Boden, J. Fischer, K. Herbig, and U. Spierling, “Citizentalk: Application of chatbot infotainment to e-democracy,” in *Technologies for Interactive Digital Storytelling and Entertainment*, S. Göbel, R. Malkewitz, and I. Iurgel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 370–381.
- [16] K. Fitzpatrick, A. Darcy, and M. Vierhile, “Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated

- conversational agent (woebot): A randomized controlled trial,” *JMIR Mental Health*, vol. 4, p. e19, 06 2017.
- [17] L. Fryer, M. Ainley, A. Thompson, A. Gibson, and Z. Sherlock, “Stimulating and sustaining interest in a language course: An experimental comparison of chatbot and human task partners,” *Computers in Human Behavior*, vol. 75, pp. 461 – 468, 2017.
- [18] R. Ravi, “Intelligent chatbot for easy web-analytics insights,” in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. Bangalore, India: IEEE, Sep 2018, pp. 2193–2195.
- [19] Q. Xie, D. Tan, T. Zhu, Q. Zhang, S. Xiao, J. Wang, B. Li, L. Sun, and P. Yi, “Chatbot application on cryptocurrency,” in *2019 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*, 2019, pp. 1–8.
- [20] A. Argal, S. Gupta, A. Modi, P. Pandey, S. Shim, and C. Choo, “Intelligent travel chatbot for predictive recommendation in echo platform,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*. Las Vegas, NV, USA: IEEE, Jan 2018, pp. 176–183.
- [21] A. Dian Sano, T. Daud Imanuel, M. Intanadias Calista, H. Nindito, and A. Raharto Condrobimo, “The application of agnes algorithm to optimize knowledge base for tourism chatbot,” in *2018 International Conference on Information Management and Technology (ICIMTech)*. Jakarta, Indonesia: IEEE, Sep 2018, pp. 65–68.
- [22] H. Agus Santoso, N. Anisa Sri Winarsih, E. Mulyanto, G. Wilujeng saraswati, S. Enggar Sukmana, S. Rustad, M. Syaifur Rohman, A. Nugraha, and F. Firdausillah, “Dinus intelligent assistance (dina) chatbot for university admission services,” in *2018 International Seminar on Application for Technology of Information and Communication*. Semarang, Indonesia: IEEE, Sep 2018, pp. 417–423.
- [23] E. Luger and A. Sellen, “Like having a really bad pa: The gulf between user expectation and experience of conversational agents,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 5286–5297.

- [24] M. Su, C. Wu, K. Huang, Q. Hong, and H. Wang, “A chatbot using lstm-based multi-layer embedding for elderly care,” in *2017 International Conference on Orange Technologies (ICOT)*, 2017, pp. 70–74.
- [25] N. Rosruen and T. Samanchuen, “Chatbot utilization for medical consultant system,” in *2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, 2018, pp. 1–5.
- [26] F. Dubosson, R. Schaer, R. Savioz, and M. Schumacher, “Going beyond the relapse peak on social network smoking cessation programmes: Chatbot opportunities,” *Swiss Medical Informatics*, 09 2017.
- [27] C. Lisetti, R. Amini, and U. Yasavur, “Now all together: Overview of virtual health assistants emulating face-to-face health interview experience,” *KI - Künstliche Intelligenz*, vol. 29, 06 2015.
- [28] D. Elmasri and A. Maeder, “A conversational agent for an online mental health intervention,” in *Brain Informatics and Health*, G. A. Ascoli, M. Hawrylycz, H. Ali, D. Khazanchi, and Y. Shi, Eds. Cham: Springer International Publishing, 2016, pp. 243–251.
- [29] D. Sawant, A. Jaiswal, J. Singh, and P. Shah, “Agribot - an intelligent interactive interface to assist farmers in agricultural activities,” in *2019 IEEE Bombay Section Signature Conference (IBSSC)*, 2019, pp. 1–6.
- [30] A. Shaw, “Using chatbots to teach socially intelligent computing principles in introductory computer science courses,” in *2012 Ninth International Conference on Information Technology - New Generations*. Las Vegas, NV, USA: IEEE, 04 2012.
- [31] J. Murcia Triviño, S. Moreno Rodríguez, D. O. Díaz López, and F. Gómez Mármol, “C3-sex: a chatbot to chase cyber perverts,” in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCCom/CyberSciTech)*, 2019, pp. 50–57.

- [32] C. Baby, F. Khan, and J. Swathi, “Home automation using iot and a chatbot using natural language processing,” in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vellore, India, Apr 2017, pp. 1–6.
- [33] V. Hristidis, “Chatbot technologies and challenges,” in *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, 2018, pp. 126–126.
- [34] G. Molnar and Z. Szuts, “The role of chatbots in formal education,” in *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*. Subotica, Serbia: IEEE, Sep 2018, pp. 197–202.
- [35] R. Reyes, D. Garza, L. Garrido, V. De la Cueva, and J. Ramirez, “Methodology for the implementation of virtual assistants for education using google dialogflow,” in *Advances in Soft Computing*, L. Martínez-Villaseñor, I. Batyrshin, and A. Marín-Hernández, Eds. Cham: Springer International Publishing, 2019, pp. 440–451.
- [36] S. Yang and C. Evans, “Opportunities and challenges in using ai chatbots in higher education,” in *Proceedings of the 2019 3rd International Conference on Education and E-Learning*, ser. ICEEL 2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 79–83.
- [37] P. Bii, J. Too, and C. Mukwa, “Teacher attitude towards use of chatbots in routine teaching.” *Universal Journal of Educational Research*, vol. 6, no. 7, pp. 1586–1597, 2018.
- [38] Admithub, “Harvard innovation lab, admithub,” <https://www.admithub.com>. Accessed 15 March 2019, 2018.
- [39] J. Maderer, “Jill watson, round three, georgia tech,” <https://www.news.gatech.edu/2017/01/09/jill-watson-round-three>. Accessed 15 March 2019, 2017.
- [40] G. Silva de Souza, Y. Bento Marques, W. Siqueira Jardim, N. Cesar Lima, G. Lopes Junior, and R. Silveira Ramos, “Brazilian students expectations regarding distance learning and remote classes during the covid-19 pandemic.” *Educational Sciences: Theory & Practice*, vol. 20, no. 4, pp. 66–80, 2020.

- [41] E. Gogh and A. Kovari, “Examining the relationship between lifelong learning and language learning in a vocational training institution,” *Journal of Applied Technical and Educational Sciences*, vol. 8 No. 1, no. 1, pp. 52–69, Apr 2018.
- [42] A. Rahman, A. Mamun, and A. Islam, “Programming challenges of chatbot: Current and future prospective,” in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. Dhaka, Bangladesh: IEEE, Dec 2017, pp. 75–78.
- [43] L. Fryer, K. Nakao, and A. Thompson, “Chatbot learning partners: Connecting learning experiences, interest and competence,” *Computers in Human Behavior*, vol. 93, pp. 279 – 289, 2019.
- [44] F. Mikic-Fonte, J. Burguillo, D. Rodriguez, E. Rodriguez, and M. Llamas, “T-bot and q-bot: A couple of aiml-based bots for tutoring courses and evaluating students,” in *2008 38th Annual Frontiers in Education Conference*, 2008, pp. S3A–7–S3A–12.
- [45] F. Mikic-Fonte, J. Burguillo, M. Llamas, D. Rodriguez, and E. Rodriguez, “Charlie: An aiml-based chatterbot which works as an interface among ines and humans,” in *2009 EAEEIE Annual Conference*, June 2009, pp. 1–6.
- [46] L. Valle-Rosado, M. García-García, and J. López-Martínez, “Desarrollo e implementación de un bot conversacional como apoyo a los estudiantes en su proceso de titulación,” in *International Conference on Robotics and Computing*, May 2013.
- [47] H. Hien, P.-N. Cuong, L. Nam, H. Nhung, and L. Thang, “Intelligent assistants in higher-education environments: The fit-ebot, a chatbot for administrative and learning support,” in *Proceedings of the Ninth International Symposium on Information and Communication Technology*, ser. SoICT 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 69–76.
- [48] M. Dibitonto, K. Leszczynska, F. Tazzi, and C. M. Medaglia, “Chatbot in a campus environment: Design of lisa, a virtual assistant to help students in their university life,” in *Human-Computer Interaction. Interaction Technologies*, M. Kurosu, Ed. Cham: Springer International Publishing, 2018, pp. 103–116.

- [49] L. Paschoal, M. de Oliveira, and P. Chicon, “A chatterbot sensitive to student’s context to help on software engineering education,” in *2018 XLIV Latin American Computer Conference (CLEI)*, Oct 2018, pp. 839–848.
- [50] M. Hernández-León, “Desarrollo de un chatbot inteligente, caso de estudio: alumnos y profesores de secundaria,” Master’s thesis, Departamento de Computación - Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Feb 2019.
- [51] M. Montessori, *The Montessori method: the origins of an educational innovation: including an abridged and annotated edition of Maria Montessori’s The Montessori method*. Rowman & Littlefield, 2004.
- [52] M. Moreno, “Pedagogía waldorf,” *Arteterapia. Papeles de arteterapia y educación artística para la inclusión social*, vol. 5, pp. 203–209, 2010.
- [53] G. Kiryakova, N. Angelova, and L. Yordanova, “Gamification in education,” in *Proceedings of 9th international Balkan education and science conference*, vol. 1, 2014, pp. 679–684.
- [54] J. Bergmann and A. Sams, *Flip your classroom: Reach every student in every class every day*. International society for technology in education, 2012.
- [55] P. Sandgren and E.-M. T. Ahlquist, “Montessori education in nordic countries,” *The Bloomsbury Handbook of Montessori Education*, p. 315, 2023.
- [56] I. Enkvist, “El éxito educativo finlandés,” *Bordón: Revista de pedagogía*, vol. 62, no. 3, pp. 49–67, 2010.
- [57] G. Garcia Brustenga, M. Fuertes Alpiste, and N. Molas Castells, “Briefing paper: los chatbots en educación,” 2018.
- [58] S. Cunningham-Nelson, W. Boles, L. Trouton, and E. Margerison, “A review of chatbots in education: practical steps forward,” in *30th Annual Conference for the Australasian Association for Engineering Education (AAEE 2019): Educators Becoming Agents of Change: Innovate, Integrate, Motivate*. Engineers Australia, 2019, pp. 299–306.

- [59] B. Skinner, *Science and human behavior*. Simon and Schuster, 1965, no. 92904.
- [60] M. Carretero, “¿ qué es el constructivismo,” *Progreso*. Recuperado de: [http://www.educando.edu.do/Userfiles P](http://www.educando.edu.do/Userfiles/P), vol. 1, pp. 39–71, 1997.
- [61] K. He, M. Jiao, and Z. Liu, “The promotion of maker education to traditional education,” in *2021 Tenth International Conference of Educational Innovation through Technology (EITT)*, 2021, pp. 95–99.
- [62] Okonkwo and A. Ade-Ibijola, “Chatbots applications in education: A systematic review,” *Computers and Education: Artificial Intelligence*, vol. 2, p. 100033, 2021.
- [63] M. Nuruzzaman and O. K. Hussain, “A survey on chatbot implementation in customer service industry through deep neural networks,” in *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, 2018, pp. 54–61.
- [64] B. Kitchenham, “Guidelines for performing systematic literature reviews in software engineering. technical report,” Keele University and Durham University Joint Report, Tech. Rep. EBSE 2007-001, 01 2007.
- [65] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering,” in *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE’08. Swindon, GBR: BCS Learning and Development Ltd., 2008, p. 68–77.
- [66] M. Salama, R. Bahsoon, and N. Bencomo, “Chapter 11 - managing trade-offs in self-adaptive software architectures: A systematic mapping study,” in *Managing Trade-Offs in Adaptable Software Architectures*, I. Mistrik, N. Ali, R. Kazman, J. Grundy, and B. Schmerl, Eds. Boston: Morgan Kaufmann, 2017, pp. 249–297.
- [67] D. Budgen, M. Turner, P. Brereton, and B. Kitchenham, in *PPIG*, vol. 8, 2008, pp. 195–204.
- [68] H. Chen, X. Liu, D. Yin, and J. Tang, “A survey on dialogue systems: Recent advances and new frontiers,” *SIGKDD Explor. Newsl.*, vol. 19, no. 2, p. 25–35, Nov. 2017. [Online]. Available: <https://doi.org/10.1145/3166054.3166058>

- [69] U. Gnewuch, S. Morana, and A. Maedche, “Towards designing cooperative and social conversational agents for customer service,” in *Proceedings of the 38th International Conference on Information Systems (ICIS), Seoul, ROK, December 10-13, 2017. Research-in-Progress Papers*. AIS eLibrary (AISEL), 2017.
- [70] K. Ramesh, S. Ravishankaran, A. Joshi, and K. Chandrasekaran, “A survey of design techniques for conversational agents,” in *Information, Communication and Computing Technology*, S. Kaushik, D. Gupta, L. Kharb, and D. Chahal, Eds. Singapore: Springer Singapore, 2017, pp. 336–350.
- [71] S. Diederich, A. Brendel, and L. Kolbe, “Towards a taxonomy of platforms for conversational agent design,” in *Wirtschaftsinformatik*, 2019.
- [72] K. Nimavat and T. Champaneria, “Chatbots: An overview. types, architecture, tools and future possibilities,” vol. 5. *International Journal for Scientific Research & Development*, Oct. 2017, pp. 1019–1024.
- [73] S. Hussain, O. Ameri Sianaki, and N. Ababneh, “A survey on conversational agents/chatbots classification and design techniques,” in *Web, Artificial Intelligence and Network Applications*, L. Barolli, M. Takizawa, F. Xhafa, and T. Enokido, Eds. Cham: Springer International Publishing, 2019, pp. 946–956.
- [74] E. Adamopoulou and L. Moussiades, “Chatbots: History, technology, and applications,” *Machine Learning with Applications*, vol. 2, p. 100006, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S26668270-20300062>
- [75] J. Quiroga Pérez, T. Daradoumis, and J. M. M. Puig, “Rediscovering the use of chatbots in education: A systematic literature review,” *Computer Applications in Engineering Education*, vol. 28, no. 6, pp. 1549–1565, 2020.
- [76] J. Feine, S. Morana, and A. Maedche, “Leveraging machine-executable descriptive knowledge in design science research – the case of designing socially-adaptive chatbots,” in *Extending the Boundaries of Design Science Theory and Practice*, B. Tulu, S. Djamasbi, and G. Leroy, Eds. Cham: Springer International Publishing, 2019, pp. 76–91.

- [77] A. Janssen, D. Rodríguez Cardona, and M. Breitner, “More than faq! chatbot taxonomy for business-to-business customer services,” in *Chatbot Research and Design*, A. Følstad, T. Araujo, S. Papadopoulos, E. L.-C. Law, E. Luger, M. Goodwin, and P. B. Brandtzaeg, Eds. Cham: Springer International Publishing, 2021, pp. 175–189.
- [78] A. Janssen, J. Passlick, D. Cardona, and M. Breitner, “Virtual assistance in any context - a taxonomy of design elements for domain-specific chatbots,” *Business & Information Systems Engineering*, vol. 62, pp. 211–225, 04 2020.
- [79] E. Bittner, S. Oeste-Reiß, and J. Leimeister, “Where is the bot in our team? toward a taxonomy of design option combinations for conversational agents in collaborative work,” in *HICSS*, 01 2019.
- [80] R. Nickerson, U. Varshney, and J. Muntermann, “A method for taxonomy development and its application in information systems,” *European Journal of Information Systems*, vol. 22, no. 3, pp. 336–359, 2013.
- [81] S. Gregor, “The nature of theory in information systems,” *MIS quarterly*, vol. 30, no. 3, pp. 611–642, 2006.
- [82] Y. Wand, D. Monarchi, J. Parsons, and C. C., “Theoretical foundations for conceptual modelling in information systems development,” *Decision Support Systems*, vol. 15, no. 4, pp. 285–304, 1995. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0167923694000436>
- [83] R. Glass and I. Vessey, “Contemporary application-domain taxonomies,” *IEEE Software*, vol. 12, no. 4, pp. 63–76, 1995.
- [84] D. McKnight and N. Chervany, “What trust means in e-commerce customer relationships: An interdisciplinary conceptual typology,” *International Journal of Electronic Commerce*, vol. 6, no. 2, pp. 35–59, 2001.
- [85] R. Sabherwal and W. R. King, “An empirical taxonomy of the decision-making processes concerning strategic applications of information systems,” *Journal of Management Information Systems*, vol. 11, no. 4, pp. 177–214, 1995. [Online]. Available: <http://www.jstor.org/stable/40398146>

- [86] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 3rd ed. Addison-Wesley Professional, 2012.
- [87] I. Jacobson, G. Booch, and J. Rumbaugh, *The unified software development process*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.
- [88] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *PATTERN-ORIENTED SOFTWARE ARCHITECTURE: A SYSTEM OF PATTERNS, VOLUME 1*. Wiley India Pvt. Limited, 2008, no. v. 1.
- [89] G. Heineman and W. Councill, Eds., *Component-Based Software Engineering: Putting the Pieces Together*. USA: Addison-Wesley Longman Publishing Co., Inc., 2001.
- [90] D. Herzberg and M. Broy, “Modeling layered distributed communication systems,” *Formal Aspects of Computing*, vol. 17, no. 1, pp. 1–18, 2005.
- [91] D. Namiot and M. Sneps-Sneppe, “On micro-services architecture,” *International Journal of Open Information Technologies*, vol. 2, no. 9, pp. 24–27, 2014.
- [92] T. Reenskaug and J. Coplien, “The dci architecture: A new vision of object-oriented programming,” *Artima Developer Best Practices in enterprise Software Development (Artima)*, Mar. 2009.
- [93] R. Khan, “Standardized architecture for conversational agents aka chatbots,” *International Journal of Computer Trends and Technology*, vol. 50, no. 2, pp. 114–121, 2017.
- [94] S. Srivastava and T. Prabhakar, “A reference architecture for applications with conversational components,” in *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, 2019, pp. 1–5.
- [95] P. Kruchten, *The rational unified process: an introduction*. Addison-Wesley Professional, 2004.
- [96] A. Lattanze, *The architecture centric development method*. Carnegie Mellon University, School of Computer Science, 2005.

- [97] C. Hofmeister, P. Kruchten, R. Nord, H. Obbink, A. Ran, and P. America, “A general model of software architecture design derived from five industrial approaches,” *Journal of Systems and Software*, vol. 80, no. 1, pp. 106–126, 2007. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0164121206001634>
- [98] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice, Second Edition*. Addison-Wesley Professional, Apr. 2003.
- [99] C. Hofmeister, R. Nord, and D. Soni, *Applied Software Architecture*. USA: Addison-Wesley Longman Publishing Co., Inc., 1999.
- [100] P. Kruchten, “The 4+1 view model of architecture,” *IEEE Software*, vol. 12, no. 6, pp. 42–50, 1995.
- [101] P. America, E. Rommes, and H. Obbink, “Multi-view variation modeling for scenario analysis,” in *International Workshop on Software Product-Family Engineering*. Springer, 2003, pp. 44–65.
- [102] H. Obbink, J. Muller, P. America, and R. Copa, “A component-oriented platform architecting method for families of software-intensive electronic prod,” 2000.
- [103] Ran, *ARES Conceptual Framework for Software Architecture In: azayeri, Mehdi and Ran, Alexander and Van Der Linden, Frank and Van Der Linden, Philip Software architecture for product families: principles and practice*. Addison-Wesley Reading, 2000.
- [104] P. Eeles and P. Cripps, *The Process of Software Architecting*. Addison-Wesley Professional, 2010.
- [105] M. Patterns and P. Team, *Microsoft® Application Architecture Guide*. Microsoft Press, 2009.
- [106] N. Rozanski and E. Woods, *Software systems architecture: working with stakeholders using viewpoints and perspectives*. Addison-Wesley, 2012.
- [107] H. Cervantes and R. Kazman, *Designing software architectures: a practical approach*. Addison-Wesley Professional, 2016.

- [108] M. Petticrew and H. Roberts, *Systematic Reviews in the Social Sciences: A Practical Guide*, 01 2006, vol. 11.
- [109] B. Shawar and E. Atwell, “Using corpora in machine-learning chatbot systems,” *International Journal of Corpus Linguistics*, vol. 10, pp. 489–516, 12 2005.
- [110] A. Shevat, *Designing bots: Creating conversational experiences*. O’Reilly Media, Inc., 2017.
- [111] L. Klopfenstein, S. Delpriori, S. Malatini, and A. Bogliolo, “The rise of bots: A survey of conversational interfaces, patterns, and paradigms,” in *Proceedings of the 2017 Conference on Designing Interactive Systems*, ser. DIS ’17. New York, NY, USA: Association for Computing Machinery, 2017, p. 555–565.
- [112] G. Daniel, J. Cabot, L. Deruelle, and M. Derras, “Multi-platform chatbot modeling and deployment with the jarvis framework,” in *Advanced Information Systems Engineering*, P. Giorgini and B. Weber, Eds. Cham: Springer International Publishing, 2019, pp. 177–193.
- [113] Lebeuf, Storey, and Zagalsky, “Software bots,” *IEEE Software*, vol. 35, no. 1, pp. 18–23, 2018.
- [114] A. Shah, B. Jain, B. Agrawal, S. Jain, and S. Shim, “Problem solving chatbot for data structures,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018, pp. 184–189.
- [115] P. Brandtzaeg and A. Følstad, “Chatbots: User changing needs and motivations.” *Interactions*, vol. 25, no. 5, pp. 38–43, 2018.
- [116] A. Følstad and P. B. Brandtzaeg, “Users’ experiences with chatbots: findings from a questionnaire study,” *Quality and User Experience*, vol. 5, no. 1, pp. 1–14, 2020.
- [117] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, “The rise of social bots,” *Commun. ACM*, vol. 59, no. 7, p. 96–104, Jun. 2016.
- [118] J. Jang and K. Lee, “Transforming a specialized q&a system to a chatbot system: A case of a simplified taxation in korea,” in *HCI International 2019 – Late Breaking Posters*, C. Stephanidis and M. Antona, Eds. Cham: Springer International Publishing, 2019, pp. 296–308.

- [119] J. Huang, M. Zhou, and D. Yang, “Extracting chatbot knowledge from online discussion forums.” in *IJCAI*, vol. 7, 2007, pp. 423–428.
- [120] Z. Chen, Y. Lu, M. Nieminen, and A. Lucero, *Creating a Chatbot for and with Migrants: Chatbot Personality Drives Co-Design Activities*. New York, NY, USA: Association for Computing Machinery, 2020, p. 219–230.
- [121] P. Weber and T. Ludwig, “(non-)interacting with conversational agents: Perceptions and motivations of using chatbots and voice assistants,” in *Proceedings of the Conference on Mensch Und Computer*, ser. MuC ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 321–331.
- [122] H. Bansal and R. Khan, “A review paper on human computer interaction,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, p. 53, 04 2018.
- [123] J. Cahn, “Chatbot: Architecture, design, & development,” *University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science*, 2017.
- [124] K. Fornalczyk, K. Bortko, and J. Jankowski, “Improving user attention to chatbots through a controlled intensity of changes within the interface,” *Procedia Computer Science*, vol. 192, pp. 5112–5121, 2021, knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021.
- [125] P. Robe and S. Kuttal, “Designing pairbuddy—a conversational agent for pair programming,” *ACM Trans. Comput.-Hum. Interact.*, vol. 29, no. 4, may 2022. [Online]. Available: <https://doi.org/10.1145/3498326>
- [126] J. Heo and J. Lee, “Cisa: An inclusive chatbot service for international students and academics,” in *HCI International 2019 – Late Breaking Papers*, C. Stephanidis, Ed. Cham: Springer International Publishing, 2019, pp. 153–167.
- [127] M. Daswani, K. Desai, M. Patel, R. Vani, and M. Eirinaki, “Collegebot: A conversational ai approach to help students navigate college,” in *International Conference on Human-Computer Interaction*. Springer, 2020, pp. 44–63.

- [128] O. Zahour, E. Benlahmar, A. Eddaoui, H. Ouchra, and O. Hourrane, “A system for educational and vocational guidance in morocco: Chatbot e-orientation,” *Procedia Computer Science*, vol. 175, pp. 554–559, 2020, the 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), The 15th International Conference on Future Networks and Communications (FNC), The 10th International Conference on Sustainable Energy Information Technology.
- [129] C. Chun Ho, H. Lee, W. Lo, and K. Lui, “Developing a chatbot for college student programme advisement,” in *2018 International Symposium on Educational Technology (ISET)*, 2018, pp. 52–56.
- [130] S. Ondáš, M. Pleva, and D. Hládek, “How chatbots can be involved in the education process,” in *2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2019, pp. 575–580.
- [131] N. Khin and K. Soe, “University chatbot using artificial intelligence markup language,” 2020.
- [132] K. Lee, J. Jo, J. Kim, and Y. Kang, “Can chatbots help reduce the workload of administrative officers? - implementing and deploying faq chatbot service in a university,” in *HCI International 2019 - Posters*, C. Stephanidis, Ed. Cham: Springer International Publishing, 2019, pp. 348–354.
- [133] M. Mekni, Z. Baani, and D. Sulieman, “A smart virtual assistant for students,” in *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, ser. APPIS 2020. New York, NY, USA: Association for Computing Machinery, 2020.
- [134] S. Meshram, N. Naik, Megha, T. More, and S. Kharche, “College enquiry chatbot using rasa framework,” in *2021 Asian Conference on Innovation in Technology (ASIANCON)*, 2021, pp. 1–8.
- [135] M.-T. Nguyen, M. Tran-Tien, A. Viet, H.-T. Vu, and V.-H. Nguyen, “Building a chatbot for supporting the admission of universities,” in *2021 13th International Conference on Knowledge and Systems Engineering (KSE)*, 2021, pp. 1–6.

- [136] Sweidan, A. Laban, Alnaimat, and Darabkh, “Siaaa-c: A student interactive assistant android application with chatbot during covid-19 pandemic,” *Computer Applications in Engineering Education*, vol. 29, no. 6, p. 1718 – 1742, 2021.
- [137] G. Sai Vikas, I. Kumar, S. Shareef, B. Roy, and G. Geetha, “Information chatbot for college management system using multinomial naive bayes,” in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 2021, pp. 1149–1153.
- [138] D. Sebastian and K. A. Nugraha, “Academic customer service chatbot development using telegrambot api,” in *2021 2nd International Conference on Innovative and Creative Information Technology (ICITech)*, 2021, pp. 221–225.
- [139] J. Cordero, A. Toledo, F. Guamán, and L. Barba-Guamán, “Use of chatbots for user service in higher education institutions,” in *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 2020, pp. 1–6.
- [140] Santana, Ferreira, Rolim, Miranda, Nascimento, and Mello, “A chatbot to support basic students questions,” vol. 3059, 2021, p. 58 – 67.
- [141] N. Gunson, W. Sieińska, Y. Yu, D. H. Garcia, J. L. Part, C. Dondrup, and O. Lemon, “Coronabot: A conversational ai system for tackling misinformation,” in *Proceedings of the Conference on Information Technology for Social Good*, ser. GoodIT ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 265–270.
- [142] L. Benotti, M. C. Martínez, and F. Schapachnik, “Engaging high school students using chatbots,” in *Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 63–68.
- [143] F. Clarizia, F. Colace, M. Lombardi, F. Pascale, and D. Santaniello, “Chatbot: An education support system for student,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11161 LNCS, pp. 291–302, 2018.
- [144] A. Damasceno, A. R. Martins, M. L. Chagas, E. M. a. Barros, P. Maia, and F. C. M. B. Oliveira, “Stuart: An intelligent tutoring system for increasing scalability of

- distance education courses,” in *Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems*, ser. IHC '20. New York, NY, USA: Association for Computing Machinery, 2020.
- [145] J. Du, W. Huang, and K. F. Hew, “Supporting students goal setting process using chatbot: implementation in a fully online course,” in *2021 IEEE International Conference on Engineering, Technology Education (TALE)*, 2021, pp. 35–41.
- [146] M. Verleger and J. Pembridge, “A pilot study integrating an ai-driven chatbot in an introductory programming course,” in *2018 IEEE Frontiers in Education Conference (FIE)*, Oct 2018, pp. 1–4.
- [147] G. A.N. and V. R. V., “English master ammu: Chatbot for english learning,” 2021, p. 618 – 623.
- [148] X. L. Pham, T. Pham, Q. M. Nguyen, T. H. Nguyen, and T. T. H. Cao, “Chatbot as an intelligent personal assistant for mobile language learning,” in *Proceedings of the 2018 2nd International Conference on Education and E-Learning*, ser. ICEEL 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 16–21.
- [149] S. Ruan, L. Jiang, Q. Xu, Z. Liu, G. M. Davis, E. Brunskill, and J. A. Landay, *EnglishBot: An AI-Powered Conversational System for Second Language Learning*. New York, NY, USA: Association for Computing Machinery, 2021, p. 434–444.
- [150] K. Gaglo, B. Degboe, G. Kossingou, and S. Ouya, “Proposal of conversational chatbots for educational remediation in the context of covid-19,” in *2021 23rd International Conference on Advanced Communication Technology (ICACT)*, 2021, pp. 354–358.
- [151] B. Göschlberger and C. Brandstetter, “Conversational ai for corporate e-learning,” in *Proceedings of the 21st International Conference on Information Integration and Web-Based Applications and Services*, ser. iiWAS2019. New York, NY, USA: Association for Computing Machinery, 2019, p. 674–678.
- [152] S. Han and M. K. Lee, “Faq chatbot and inclusive learning in massive open online courses,” *Computers & Education*, vol. 179, p. 104395, 2022.

- [153] F. Mikic-Fonte, J. Rial, M. Llamas-Nistal, and D. Hermida, “Using semantics in ines, an intelligent educational system,” in *2009 39th IEEE Frontiers in Education Conference*, Oct 2009, pp. 1–6.
- [154] J. Jeon, “Exploring ai chatbot affordances in the efl classroom: young learners’ experiences and perspectives,” *Computer Assisted Language Learning*, 2021.
- [155] C. Troussas, A. Krouska, and M. Virvou, “Integrating an adjusted conversational agent into a mobile-assisted language learning application,” in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2017, pp. 1153–1157.
- [156] A. Muhammad, D. Susanto, A. Alimudin, F. Adila, M. Assidiqi, and S. Nabhan, “Developing english conversation chatbot using dialogflow,” 2020, p. 468 – 475.
- [157] T. My-Thanh Nguyen, T. Hai Diep, B. Bien Ngo, N. Bich Le, and X. Quy Dao, “Design of online learning platform with vietnamese virtual assistant,” in *2021 6th International Conference on Intelligent Information Technology*, ser. ICIIT ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 51–57.
- [158] Oliveira, D. B. Espíndola, R. Barwaldt, L. M. Ribeiro, and M. Pias, “Ibm watson application as faq assistant about moodle,” in *2019 IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1–8.
- [159] J. Pereira, M. Fernández-Raga, S. Osuna-Acedo, M. Roura-Redondo, O. Almazán-López, and A. Buldón-Olalla, “Promoting learners’ voice productions using chatbots as a tool for improving the learning process in a mooc.” *Technology, Knowledge and Learning*, vol. 24, no. 4, pp. 545–565, 2019.
- [160] N. Phaokla and P. Netinant, “Design an environment information chatbots system for a smart school framework,” in *2021 The 4th International Conference on Software Engineering and Information Management*, ser. ICSIM 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 152–157.
- [161] C. Șerban and L. Ioan, “Qlearn: Towards a framework for smart learning environments,” *Procedia Computer Science*, vol. 176, pp. 2812–2821, 2020, knowledge-Based

- and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020.
- [162] M. Sarosa, M. Kusumawardani, A. Suyono, and M. Wijaya, “Developing a social media-based chatbot for english learning,” vol. 732, no. 1, 2020.
- [163] D. Song, E. Y. Oh, and M. Rice, “Interacting with a conversational agent system for educational purposes in online courses,” in *2017 10th International Conference on Human System Interactions (HSI)*, 2017, pp. 78–82.
- [164] Y. Du, X. Su, and Y. Jiang, “Developing an intelligent personal tutor to support efl learning,” in *Proceedings of the 4th International Conference on Computer Science and Application Engineering*, ser. CSAE 2020. New York, NY, USA: Association for Computing Machinery, 2020.
- [165] M.-H. Hsu, P.-S. Chen, and C.-S. Yu, “Proposing a task-oriented chatbot system for efl learners speaking practice,” *Interactive Learning Environments*, 2021.
- [166] E. Vázquez-Cano, S. Mengual-Andres, and E. Meneses, “Chatbot to improve learning punctuation in spanish and to enhance open and flexible learning environments,” *International Journal of Educational Technology in Higher Education*, vol. 18, 12 2021.
- [167] Y. Ye, J. Deng, and X. Liu, “The effect of oral practice via chatbot on students’ oral english accuracy,” in *International Symposium on Emerging Technologies for Education*. Springer, 2021, pp. 344–354.
- [168] L. Li, R. Subbareddy, and C. Raghavendra, “Ai intelligence chatbot to improve students learning in the higher education platform,” *Journal of Interconnection Networks*, 2022.
- [169] I. Atmosukarto, C. W. Sin, P. Iyer, N. H. Tong, and K. W. Peng Yu, “Enhancing adaptive online chemistry course with ai-chatbot,” in *2021 IEEE International Conference on Engineering, Technology Education (TALE)*, 2021, pp. 838–843.
- [170] K. Bala, M. Kumar, S. Hulawale, and S. Pandita, “Chat-bot for college management system using a.i,” *International Rsearch Journal of Engineering and Technology (IRJET)*, pp. 2395–0056, Nov. 2017. [Online]. Available: www.irjet.net

- [171] V. Bhargava and N. Maheshwari, “An intelligent speech recognition system for education system,” 2009.
- [172] J. P. Biggam, M. B. Aller, J. T. Brudvik, J. O. Leung, L. A. Yazzolino, and R. E. Ladner, “Inspiring blind high school students to pursue computer science with instant messaging chatbots,” *SIGCSE Bull.*, vol. 40, no. 1, p. 449–453, Mar. 2008.
- [173] D. Petousi, A. Katifori, S. McKinney, S. Perry, M. Roussou, and Y. Ioannidis, “Social bots of conviction as dialogue facilitators for history education: Promoting historical empathy in teens through dialogue,” in *Interaction Design and Children*, ser. IDC ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 326–337.
- [174] T. Lee, T. Zhu, S. Liu, L. Trac, Z. Huang, and Y. Chen, “Casexplorer: A conversational academic and career advisor for college students,” in *The Ninth International Symposium of Chinese CHI*, ser. Chinese CHI 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 112–116.
- [175] N. Suresh, N. Mukabe, V. Hashiyana, A. Limbo, and A. Hauwanga, “Career counseling chatbot on facebook messenger using ai,” in *Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence*, ser. DSMLAI ’21’. New York, NY, USA: Association for Computing Machinery, 2021, p. 65–73.
- [176] L. Benotti, M. C. Martnez, and F. Schapachnik, “A tool for introducing computer science with automatic formative assessment,” *IEEE Transactions on Learning Technologies*, vol. 11, no. 2, pp. 179–192, 2018.
- [177] N. A. Nasharuddin, N. M. Sharef, E. I. Mansor, N. Samian, M. A. A. Murad, M. K. Omar, N. I. Arshad, F. Shahbodin, and M. H. Marhaban, “Designing an educational chatbot: A case study of cikguaibot,” in *2021 Fifth International Conference on Information Retrieval and Knowledge Management (CAMP)*, 2021, pp. 19–24.
- [178] J. Nias and M. Ruffin, “Culturebot: A culturally relevant humanoid robotic dialogue agent,” in *Proceedings of the 2020 ACM Southeast Conference*, ser. ACM SE ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 280–283.

- [179] M. Kowsher, F. S. Tithi, M. Ashraful Alam, M. N. Huda, M. Md Moheuddin, and M. G. Rosul, “Doly: Bengali chatbot for bengali education,” in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, 2019, pp. 1–6.
- [180] D. Dutta, “Developing an intelligent chat-bot tool to assist high school students for learning general knowledge subjects,” 2017. [Online]. Available: <http://hdl.handle.net/1853/59088>
- [181] J. Sophia and T. Jacob, “Edubot-a chatbot for education in covid-19 pandemic and vqabot comparison,” in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2021, pp. 1707–1714.
- [182] A. Mahroof, V. Gamage, K. Rajendran, S. Rajkumar, S. Rajapaksha, and D. Wijendra, “An ai based chatbot to self-learn and self-assess performance in ordinary level chemistry,” in *2020 2nd International Conference on Advancements in Computing (ICAC)*, vol. 1, 2020, pp. 216–221.
- [183] W. El Hefny, A. El Bolock, C. Herbert, and S. Abdennadher, “Applying the character-based chatbots generation framework in education and healthcare,” in *Proceedings of the 9th International Conference on Human-Agent Interaction*, ser. HAI '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 121–129.
- [184] A. Neumann, T. Arndt, L. Köbis, R. Meissner, A. Martin, P. de Lange, N. Pengel, R. Klamma, and H. Wollersheim, “Chatbots as a tool to scale mentoring processes: Individually supporting self-study in higher education,” *Frontiers in Artificial Intelligence*, vol. 4, 2021.
- [185] J. Pereira and O. Díaz, “Struggling to keep tabs on capstone projects: A chatbot to tackle student procrastination,” *ACM Transactions on Computing Education*, vol. 22, no. 1, oct 2021.
- [186] C. Gómez Róspide and C. Puente Águeda, “Agente virtual inteligente aplicado a un entorno educativo,” *Revista Pensamiento Matemático*, vol. 2, Oct 2012.

- [187] N. Hirose, S. Shiramatsu, and S. Okuhara, “Development of chatbot to support student learning strategies in design education,” in *2021 IEEE International Conference on Agents (ICA)*, 2021, pp. 1–6.
- [188] L. K. Lee, Y. C. Fung, Y. W. Pun, K. K. Wong, M. T. Y. Yu, and N. I. Wu, “Using a multiplatform chatbot as an online tutor in a university course,” in *2020 International Symposium on Educational Technology (ISET)*, 2020, pp. 53–56.
- [189] H. Ismail, N. Hussein, R. Elabyad, and S. Said, “A serverless academic adviser chatbot,” in *The 7th Annual International Conference on Arab Women in Computing in Conjunction with the 2nd Forum of Women in Research*, ser. ArabWIC 2021. New York, NY, USA: Association for Computing Machinery, 2021.
- [190] E. Kasthuri and S. Balaji, “A chatbot for changing lifestyle in education,” in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp. 1317–1322.
- [191] E. Mulyana, R. Hakimi, and Hendrawan, “Bringing automation to the classroom: A chatops-based approach,” in *2018 4th International Conference on Wireless and Telematics (ICWT)*, 2018, pp. 1–6.
- [192] M. Ismail and A. Ade-Ibijola, “Lecturer’s apprentice: A chatbot for assisting novice programmers,” in *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, 2019, pp. 1–8.
- [193] F. Mikic-Fonte, M. Nistal, and Rodriguez, “Using a chatterbot as a faq assistant in a course about computers architecture,” in *2018 IEEE Frontiers in Education Conference (FIE)*, Oct 2018, pp. 1–4.
- [194] F. Mikic-Fonte, M. Nistal, J. Rial, and Rodriguez, “Nlast: A natural language assistant for students,” in *2016 IEEE Global Engineering Education Conference (EDU-CON)*, April 2016, pp. 709–713.
- [195] H. Nguyen, V. Pham, D. Tran, and T. Le, “Intelligent tutoring chatbot for solving mathematical problems in high-school,” in *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, Oct 2019, pp. 1–6.

- [196] M. Niranjana, M. S. Saipreethy, and T. G. Kumar, “An intelligent question answering conversational agent using naïve bayesian classifier,” in *2012 IEEE International Conference on Technology Enhanced Education (ICTEE)*, 2012, pp. 1–5.
- [197] N. Oralbayeva, A. Shakerimov, S. Sarmonov, K. Kantoreyeva, F. Dadebayeva, N. Serkali, and A. Sandygulova, “K-qbot: Language learning chatbot based on reinforcement learning,” in *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '22. IEEE Press, 2022, p. 963–967.
- [198] J. Pereira and M. A. Barcina, “A chatbot assistant for writing good quality technical reports,” in *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, ser. TEEM'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 59–64.
- [199] R. Aguilar-Mejía and S. Tejeda, “Using virtual assistant for learning selected topics of physics,” in *2020 IEEE Integrated STEM Education Conference (ISEC)*, 2020, pp. 1–4.
- [200] Okonkwo and Ade-Ibijola, “Python-bot: A chatbot for teaching python programming,” *Engineering Letters*, vol. 29, no. 1, p. 25 – 34, 2021.
- [201] S. Ruan, L. Jiang, J. Xu, B. J.-K. Tham, Z. Qiu, Y. Zhu, E. L. Murnane, E. Brunskill, and J. A. Landay, “Quizbot: A dialogue-based adaptive learning system for factual knowledge,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–13.
- [202] S. Sweidan, S. Abu Laban, N. Alnaimat, and A. Darabkh, “Seg-covid: A student electronic guide within covid-19 pandemic,” in *2021 9th International Conference on Information and Education Technology (ICIET)*, 2021, pp. 139–144.
- [203] A. Sreelakshmi, S. Abhinaya, A. Nair, and S. Jaya Nirmala, “A question answering and quiz generation chatbot for education,” in *2019 Grace Hopper Celebration India (GHCI)*, 2019, pp. 1–6.

- [204] X. Tian, Z. Risha, I. Ahmed, A. B. Lekshmi Narayanan, and J. Biehl, “Let’s talk it out: A chatbot for effective study habit behavioral change,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. CSCW1, apr 2021.
- [205] Pengel, Martin, Meissner, Arndt, Neumann, Lange, and Wollersheim, “Teccobot: Technology-aided support for self-regulated learning: Automatic feedback on writing tasks via chatbot,” vol. 3046, 2020.
- [206] M. Rafael, T. Maria, F. Antonio, and D. Hanns, “Support to the learning of the chilean tax system using artificial intelligence through a chatbot,” in *2019 38th International Conference of the Chilean Computer Science Society (SCCC)*, 2019, pp. 1–8.
- [207] Alkhoori, Kuhail, and Alkhoori, “Unibud: A virtual academic adviser,” 2020, cited by: 5.
- [208] B. Waizmann, H. Schuhbauer, and P. Brockmann, “Smart technology to improve cultural and gender diversity in engineering education,” in *2020 IEEE Global Engineering Education Conference (EDUCON)*, 2020, pp. 61–65.
- [209] D. Sia, M. Yu, J. Daliva, J. Montenegro, and E. Ong, “Investigating the acceptability and perceived effectiveness of a chatbot in helping students assess their well-being,” in *Asian CHI Symposium 2021*, ser. Asian CHI Symposium 2021. New York, NY, USA: Association for Computing Machinery, 2021, p. 34–40.
- [210] I. Bohomolova, N. Kushnir, and S. Moshkovska, “Using chatbots to approach individual learning trajectories in physics for foreign students,” vol. 3013, 2021, p. 253 – 263.
- [211] S. Mendoza, M. Hernández-León, L. M. Sánchez-Adame, J. Rodríguez, D. Decouchant, and A. Meneses-Viveros, “Supporting student-teacher interaction through a chatbot,” in *Learning and Collaboration Technologies. Human and Technology Ecosystems*, P. Zaphiris and A. Ioannou, Eds. Cham: Springer International Publishing, 2020, pp. 93–107.
- [212] L. M. Sánchez-Adame, S. Mendoza, J. Urquiza, J. Rodríguez, and A. Meneses-Viveros, “Towards a set of heuristics for evaluating chatbots,” *IEEE Latin*

- America Transactions*, vol. 19, no. 12, p. 2037–2045, May 2021. [Online]. Available: <https://latamt.ieee9.org/index.php/transactions/article/view/5052>
- [213] D. Fanelli, “Is science really facing a reproducibility crisis, and do we need it to?” *Proceedings of the National Academy of Sciences*, vol. 115, no. 11, pp. 2628–2631, 2018. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.1708272114>
- [214] E. National Academies of Sciences, Medicine *et al.*, “Reproducibility and replicability in science,” 2019.
- [215] C. Ischen, T. Araujo, H. Voorveld, G. van Noort, and E. Smit, “Privacy concerns in chatbot interactions,” in *Chatbot Research and Design*, A. Følstad, T. Araujo, S. Papadopoulos, E. L.-C. Law, O.-C. Granmo, E. Luger, and P. B. Brandtzaeg, Eds. Cham: Springer International Publishing, 2020, pp. 34–48.
- [216] M. Hasal, J. Nowaková, K. Ahmed Saghair, H. Abdulla, V. Snášel, and L. Ogiela, “Chatbots: Security, privacy, data protection, and social aspects,” *Concurrency and Computation: Practice and Experience*, vol. 33, no. 19, p. e6426, 2021.
- [217] S. Akter, G. McCarthy, S. Sajib, K. Michael, Y. K. Dwivedi, J. D’Ambra, and K. Shen, “Algorithmic bias in data-driven innovation in the age of ai,” *Int. J. Inf. Manag.*, vol. 60, no. C, oct 2021.
- [218] A. A. Abd-alrazaq, M. Alajlani, A. A. Alalwan, B. M. Bewick, P. Gardner, and M. Househ, “An overview of the features of chatbots in mental health: A scoping review,” *International Journal of Medical Informatics*, vol. 132, p. 103978, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1386505619307166>
- [219] A. Følstad, P. B. Brandtzaeg, T. Feltwell, E. L.-C. Law, M. Tscheligi, and E. A. Luger, “Sig: Chatbots for social good,” in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–4.
- [220] M. Song, X. Xing, Y. Duan, J. Cohen, and J. Mou, “Will artificial intelligence replace human customer service? the impact of communication quality and privacy

- risks on adoption intention,” *Journal of Retailing and Consumer Services*, vol. 66, p. 102900, 2022.
- [221] Y. Zhou, Z. Fei, Y. He, and Z. Yang, “How human–chatbot interaction impairs charitable giving: The role of moral judgment,” *Journal of Business Ethics*, Feb. 2022.
- [222] J. E. Brown and J. Halpern, “Ai chatbots cannot replace human interactions in the pursuit of more inclusive mental healthcare,” *SSM - Mental Health*, vol. 1, p. 100017, 2021.
- [223] M. A. Gil and G. González-Rodríguez, *Fuzzy vs. Likert Scale in Statistics*, 01 2012, pp. 407–420.
- [224] D. Szopinski, T. Schoormann, and D. Kundisch, “Because your taxonomy is worth it: Towards a framework for taxonomy evaluation,” in *ECIS*, 06 2019.
- [225] P. Zhang and G. Feng, “Application of fuzzy comprehensive evaluation to evaluate the effect of water flooding development,” *Journal of Petroleum Exploration and Production Technology*, vol. 8, no. 4, pp. 1455–1463, 2018.
- [226] R. Saaty, “The analytic hierarchy process—what it is and how it is used,” *Mathematical Modelling*, vol. 9, no. 3, pp. 161–176, 1987.
- [227] A. Grace and T. Butler, “Beyond knowledge management: Introducing learning management systems,” *Journal of Cases on Information Technology (JCIT)*, vol. 7, no. 1, pp. 53–70, 2005.
- [228] W. Horton, *Designing Web-Based Training: How to Teach Anyone Anything Anywhere Anytime*. Wiley, 2000.
- [229] T. Murray, “Authoring intelligent tutoring systems: An analysis of the state of the art,” *Int J Artif Intelligence Educ*, vol. 10, 01 1999.
- [230] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, “What are ontologies, and why do we need them?” *IEEE Intelligent Systems and their Applications*, vol. 14, no. 1, pp. 20–26, 1999.

- [231] W. Hodgins and E. Duval, “Learning object metadata standard,” *IEEE 1484.12. 1-2002*, 2002.
- [232] F. Colace, M. De Santo, L. Greco, and P. Napoletano, “Improving relevance feedback-based query expansion by the use of a weighted word pairs approach,” *Journal of the Association for Information Science and Technology*, vol. 66, 04 2015.
- [233] F. Colace, M. De Santo, L. Greco, and P. Napoletano, “Weighted word pairs for query expansion,” *Information Processing and Management*, vol. 51, no. 1, pp. 179 – 193, 2015.
- [234] F. Colace., M. D. Santo., F. Pascale., S. Lemma., and M. Lombardi., “Botwheels: a petri net based chatbot for recommending tires,” in *Proceedings of the 6th International Conference on Data Science, Technology and Applications - Volume 1: KomIS*, INSTICC. SciTePress, 2017, pp. 350–358.
- [235] S. Arora, Y. Liang, and T. Ma, “A simple but tough-to-beat baseline for sentence embeddings,” 2017.
- [236] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [237] M. C. Mozer and R. V. Lindsey, “Predicting and improving memory retention,” *Big data in cognitive science*, p. 34, 2016.
- [238] R. S. Wallace, *The Anatomy of A.L.I.C.E.* Dordrecht: Springer Netherlands, 2009, pp. 181–210.
- [239] A. Neves, I. Diniz, and F. Barros, “Natural language communication via aiml plus chatterbots,” in *V Symposium on Human Factors in Computers Systems (IHC 2002)*, vol. 387, 2002.
- [240] B. K. Singh and A. Agarwal, “A novel approach towards naïve bayesian classifier,” in *2011 Nirma University International Conference on Engineering*, 2011, pp. 1–4.
- [241] N. Thomas, “An e-business chatbot using aiml and lsa,” in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016, pp. 2740–2742.

- [242] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.
- [243] M. Recasens, M.-C. de Marneffe, and C. Potts, “The life and death of discourse entities: Identifying singleton mentions,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 627–633. [Online]. Available: <https://www.aclweb.org/anthology/N13-1071>
- [244] W. Heeringa, “Measuring dialect pronunciation differences using levenshtein distance,” Ph.D. dissertation, University of Groningen, 2004.
- [245] R. Bathija, P. Agarwal, R. Somanna, and G. B. Pallavi, “Guided interactive learning through chatbot using bi-directional encoder representations from transformers (bert),” in *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2020, pp. 82–87.
- [246] S. N. Goodman, D. Fanelli, and J. P. A. Ioannidis, “What does research reproducibility mean?” *Science Translational Medicine*, vol. 8, no. 341, pp. 341ps12–341ps12, 2016. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scitranslmed.aaf5027>
- [247] A. Gelbukh and I. Bolshakov, “Avances y perspectivas de procesamiento automático de lenguaje natural. cuento de una máquina parlante,” *IPN ciencia, arte: cultura*, vol. 2, no. 31, pp. 10–18, May 2000.
- [248] L. Yang, J. Huang, T. Feng, W. Hongán, and D. Guozhong, “Gesture interaction in virtual reality,” *Virtual Reality & Intelligent Hardware*, vol. 1, p. 9, 02 2019.
- [249] I. M. Sukarsa, P. W. Buana, and U. Yogantara, “Multi parameter design in aiml framework for balinese calendar knowledge access.” *KSII Transactions on Internet & Information Systems*, vol. 14, no. 1, pp. 114 – 130, 2020.
- [250] S. Salvi, V. Geetha, and S. Sowmya Kamath, “Jamura: A conversational smart home assistant built on telegram and google dialogflow,” in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, 2019, pp. 1564–1571.

- [251] M. L. Brodie and J. Mylopoulos, *Knowledge Bases vs Databases*. New York, NY: Springer New York, 1986, pp. 83–86.
- [252] H. Cervantes. Requerimientos y arquitectura. SG Software Guru. [Online]. Available: <https://sg.com.mx/revista/28/requerimientos-y-arquitectura>
- [253] I. Sommerville, *Ingeniería del software*. Pearson educación, 2011.
- [254] K. Henningsson and C. Wohlin, “Understanding the relations between software quality attributes—a survey approach,” in *Proceedings 12th International Conference for Software Quality*, 2002.
- [255] K. Henningsson, “Trade-offs and conflicts between quality attributes,” 2001.
- [256] A. E. Kazdin, “Methodology: What it is and why it is so important.” 2016.
- [257] L. Boonstra, *The Definitive Guide to Conversational AI with Dialogflow and Google Cloud: Build Advanced Enterprise Chatbots, Voice, and Telephony Agents on Google Cloud*. Apress, 2021.
- [258] C. Belloch, “Diseño instruccional,” 2017.
- [259] M. Hassenzahl, M. Burmester, and F. Koller, “Attrakdiff: Ein fragebogen zur messung wahrgenommener hedonischer und pragmatischer qualität,” *Mensch & Computer 2003: Interaktion in Bewegung*, pp. 187–196, 2003.
- [260] M. Hassenzahl, S. Diefenbach, and A. Göritz, “Needs, affect, and interactive products – facets of user experience,” *Interacting with Computers*, vol. 22, no. 5, pp. 353–362, 2010, modelling user experience - An agenda for research and practice.
- [261] J. R. Lewis, “The system usability scale: Past, present, and future,” *International Journal of Human–Computer Interaction*, vol. 34, no. 7, pp. 577–590, 2018.
- [262] B. H. Kent Norman, Ben Shneiderman and L. Slaughter, “Quis: Questionnaire for user interaction satisfaction 7.0,” 2007. [Online]. Available: <https://isr.umd.edu/news/story/quis-questionnaire-for-user-interaction-satisfaction-70-isr-ip>

- [263] J. P. Chin, V. A. Diehl, and K. L. Norman, “Development of an instrument measuring user satisfaction of the human-computer interface,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '88. New York, NY, USA: Association for Computing Machinery, 1988, p. 213–218.
- [264] L. Takahashi and K. Nebe, “Observed differences between lab and online tests using the attrakdiff semantic differential scale,” *Journal of Usability Studies*, vol. 14, no. 2, pp. 65–75, 2013. [Online]. Available: <https://uxpajournal.org/attrakdiff-semantic-differential-scale/>