



Interfases
ISSN: 1993-4912
Universidad de Lima

Ayala Cadena, Omar; Aguilar Juárez, Irene
La enseñanza de la programación mediante software
educativo especializado y los agentes conversacionales
Interfases, núm. 17, 2023, Enero-Junio, pp. 170-186
Universidad de Lima

DOI: <https://doi.org/10.26439/interfases2023.n017.6337>

Disponible en: <https://www.redalyc.org/articulo.oa?id=730178910009>

- ▶ [Cómo citar el artículo](#)
- ▶ [Número completo](#)
- ▶ [Más información del artículo](#)
- ▶ [Página de la revista en redalyc.org](#)



Sistema de Información Científica Redalyc
Red de Revistas Científicas de América Latina y el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso
abierto

LA ENSEÑANZA DE LA PROGRAMACIÓN MEDIANTE *SOFTWARE* EDUCATIVO ESPECIALIZADO Y LOS AGENTES CONVERSACIONALES

OMAR AYALA CADENA

oayalac426@alumno.uaemex.mx

<https://orcid.org/0009-0009-2942-2470>

Universidad Autónoma del Estado de México

IRENE AGUILAR JUÁREZ

iaguilarj@uaemex.mx

<https://orcid.org/0000-0003-4747-0336>

Universidad Autónoma del Estado de México

RESUMEN

En las actividades cotidianas de aprendizaje se generan intercambios de ideas, dudas y explicaciones entre alumnos y docentes. Estos diálogos se interrumpen por limitaciones de tiempo o espacio, pues las clases terminan y la posibilidad de preguntar a un docente durante el trabajo en casa no es posible. Para facilitar la comunicación, se han desarrollado nuevas herramientas automatizadas que aprovechan los programas de procesamiento de texto. Un ejemplo muy concreto son los chatbots, los cuales han probado su utilidad en el comercio y en la atención al cliente. Actualmente, algunas instituciones educativas exploran los beneficios de estos programas en la atención a los alumnos. En este trabajo, se describe un estudio exploratorio sobre el uso del *software* educativo y de los agentes conversacionales en la enseñanza de la programación; el objetivo es identificar la potencialidad de este tipo de programas que apoyarán a los docentes en la formación de los programadores y en el acompañamiento de los alumnos en la autogestión del aprendizaje. Esta investigación es relevante porque las actividades de enseñanza de la programación son un reto, sobre todo cuando el alumno de nuevo ingreso inicia su contacto con la disciplina y debe aprender definiciones y tecnicismos de los que no tiene una noción previa; además, se enfrenta a desarrollar algoritmos sin nunca haberlos usado, a comprender el uso de lenguajes de programación, a participar en procesos de *software* y conocer por primera vez herramientas tecnológicas, técnicas y metodologías de desarrollo.

PALABRAS CLAVE: minería de textos, inteligencia de procesos, prestación inteligente de servicios

TEACHING PROGRAMMING THROUGH SPECIALIZED EDUCATIONAL SOFTWARE AND CONVERSATIONAL AGENTS

ABSTRACT

Daily learning activities generate exchanges of ideas, doubts, and explanations between students and teachers. These dialogues and the possibility of asking a teacher questions disappear after classes. New automated communication tools developed with word processing programs, such as chatbots, have proven helpful in commerce and customer service. Some educational institutions are currently exploring the benefits of these programs in student care. This paper describes an exploratory study on the use of educational software and conversational agents in the teaching of programming; the objective is to identify the potential of this type of program to support teachers in the training of programmers and the accompaniment of students in their self-managed learning.

KEYWORDS: text mining, process intelligence, intelligent service provisioning

1. INTRODUCCIÓN

Aprender a programar es un reto de alta complejidad para los alumnos del nivel superior en las licenciaturas relacionadas con el procesamiento de la información, como Ciencias de la Computación, Informática Administrativa, Ingeniería de Sistemas o Ingeniería en Computación; sobre todo si los alumnos no han trabajado previamente en sus habilidades de pensamiento algorítmico. En México, la enseñanza de la programación no se incluye en los estudios del nivel medio superior general, ya que estas habilidades se desarrollan solo en los planes de estudio del bachillerato tecnológico, por ejemplo, en carreras técnicas como Programación o Computación e Informática. Además, las estrategias de enseñanza de los profesores no siempre son acertadas; las evidencias muestran que los estudiantes manifiestan inconformidad con la forma de enseñanza de los docentes tradicionales, ya que, entre otros aspectos, los alumnos perciben que tienen poco apoyo de los profesores, los docentes usan ejercicios que no corresponden a contextos reales y la explicación se limita al discurso del docente (Benítez et al., 2009). Otro factor que frustra a los alumnos es que muchas veces los profesores no aprovechan los avances tecnológicos en formato, contenidos, casos de estudio o ejemplos. Se requiere una constante actualización e innovación para la impartición de clases, su evaluación y la gestión del aprendizaje, en donde se pueden adaptar contenidos y herramientas que sean fuentes de motivación para el alumno.

Por eso, la difusión de nuevas estrategias y formas de trabajar la enseñanza de la programación puede ayudar a los docentes a adaptar e innovar su forma de enseñar para lograr mejores resultados. Las aplicaciones del procesamiento de texto, específicamente los chatbots, pueden mejorar la forma en que el alumno haga consultas en los momentos de trabajo académico.

El presente artículo se compone de cinco secciones. La primera es una introducción del problema y su contexto. En la segunda, se detalla la metodología usada en el análisis de las referencias seleccionadas sobre el uso del *software* educativo en el acompañamiento del aprendizaje de la programación y de los agentes conversacionales. En el tercer apartado, se describe el resultado del análisis, la clasificación del *software* educativo y su uso en la enseñanza de la programación. En la cuarta sección, se expone el funcionamiento de los agentes inteligentes y su arquitectura. En la quinta y última sección, se formulan las conclusiones y la definición de nuevas tareas de investigación.

2. METODOLOGÍA

Este trabajo se ha realizado con un enfoque exploratorio y una investigación documental en línea, ya que se recopiló información previamente publicada para conocer e indagar sobre el uso de *software* educativo dedicado a facilitar el aprendizaje y la enseñanza de la programación.

La búsqueda de los trabajos se llevó a cabo en Google y Google Scholar, pues este último es un buscador de carácter académico que facilita el acceso a artículos especializados, libros y tesis. Se especializa en identificar contenido y bibliografía científica, además de organizar la información según la relevancia de las fuentes. Se optó por identificar palabras clave que permitan hacer una búsqueda y el filtrado de artículos.

- Palabras clave: problemas de aprendizaje, programación de computadoras, lenguaje natural, generación de lenguaje natural, enseñanza de programación, chatbot, uso de chatbot, arquitectura de un chatbot.
- Cadenas de búsqueda: se generaron a partir de conectores lógicos como OR y AND. Las cadenas ingresadas fueron estas: (“chatbot educativo” OR “uso de chatbot ámbito educativo” OR “arquitectura de un chatbot”), con 57 resultados; (“generación de lenguaje natural” OR “análisis de procesamiento de texto”), con 625 resultados; y (“problemas de aprendizaje” AND “enseñanza de programación” AND “programación de computadoras”), con 24 resultados.

En cuanto a la selección de los artículos, se utilizaron los siguientes criterios de inclusión y exclusión:

- Criterios de inclusión
 - Artículos científicos cuyo objetivo es el uso de *software* para la enseñanza de programación de computadoras.
 - Artículos científicos que explican los procesos involucrados en el procesamiento de texto.
 - Artículos científicos que exploran las dificultades para desarrollar el pensamiento algorítmico.
 - Artículos científicos que documentan el uso de chatbots en un contexto educativo.
- Criterios de exclusión
 - Artículos científicos que no tienen como objetivo de uso de herramientas tecnológicas para la enseñanza de programación de computadoras.
 - Artículos científicos que no explican los procesos involucrados en el procesamiento de texto.
 - Artículos científicos que no exploran las dificultades para desarrollar el pensamiento algorítmico.
 - Artículos científicos que no exploran el uso de chatbots en el ámbito educativo.

Luego de aplicar los criterios, se analizó un libro, siete páginas web referenciadas, cinco tesis y catorce artículos.

3. RESULTADOS

3.1 El *software* educativo dedicado al aprendizaje de la programación

La programación de computadoras permite resolver problemas de diversa naturaleza usando ordenadores, que funcionan con lenguajes de programación. Estos son conjuntos de símbolos, reglas de sintaxis y semánticas que definen las instrucciones que ejecutarán las computadoras (Jiménez-Toledo et al., 2019). La implementación de los programas resuelve la necesidad de procesar rápidamente grandes cantidades de información; sin embargo, este proceso creativo es realmente muy complejo, ya que es necesario aplicar habilidades ingenieriles, tecnológicas y de los contextos específicos de los problemas. Para los alumnos de nuevo ingreso, esta situación es muy retadora, pues se enfrentan a definiciones técnicas abstractas de las que no tienen una noción empírica previa; por eso, su aprendizaje no es sencillo.

En opinión de Gómez (2020), en la adquisición de un lenguaje se involucran tres propiedades: la expresividad del lenguaje, la interactividad formativa y la fluidez. La expresividad del lenguaje es muy importante para la enseñanza de los lenguajes formales y lógicos, pues este aspecto define la profundidad o control que se tiene al codificar programas. En los programas de bajo nivel, se encuentra una mayor expresividad porque permiten tener mejor control del *hardware*; sin embargo, para desarrollar con estos lenguajes, aunque los programas sean simples, se debe aprender detalles de mayor complejidad. Aquellos lenguajes de menor expresividad permiten la codificación de programas más fácilmente, pero con el inconveniente de que el programador se verá limitado en el control del lenguaje (Gómez, 2020).

Por su parte, la interactividad de un lenguaje de programación no es un aprendizaje espontáneo, sino que se adquiere y evoluciona respecto a su uso. Considerando el tipo de interactividad, estos pueden ser cerrados o abiertos. El entorno cerrado permite tener ejercicios predefinidos en donde el docente puede prever los errores a los que se enfrentarán los estudiantes y actuar en consecuencia, mientras que los entornos abiertos dan paso a ejercicios exploratorios como aquellos que se consideran en el aprendizaje basado en problemas, pero en este caso el docente no podrá prever los errores que existirán y los alumnos necesitarán un acompañamiento y un monitoreo personalizado (Gómez, 2020).

Finalmente, la fluidez es la habilidad de las personas en el dominio de un lenguaje natural; en el caso del aprendizaje de los lenguajes de programación, se refiere a la

adquisición demostrada de un lenguaje en específico. Este aspecto está ligado a los códigos que produzca un estudiante para determinar si tiene el dominio sobre un lenguaje de programación; es decir, se puede saber si fue capaz de resolver algún problema, o si el alumno requiere ayuda del profesor (Gómez, 2020).

Ante la dificultad de los alumnos para adquirir un lenguaje de programación, se obtienen bajos indicadores de aprendizaje en las materias de programación. Por eso, se ha buscado resolver el problema con herramientas tecnológicas, algunas con potencial uso de algoritmos de inteligencia artificial, como la minería de datos educativos.

Mancuzo (2022) clasifica el *software* educativo en cinco categorías:

1. Programas de resolución de problemas
2. Programa educativo de práctica y simulación
3. *Software* educativo de tutorial
4. *Software* de juego educativo
5. *Software* educativo de simulación

Con este criterio y con los documentos revisados, podemos identificar que para la enseñanza de la programación se han usado todos los tipos de *software* educativo. En el siguiente listado, se señalan algunas experiencias documentadas:

1. Programas de resolución de problemas: en esta categoría se encuentra el *software* para diseñar algoritmos, por ejemplo, GCompris, Raptor, Dia, StarUML o DFD (Ballesteros et al., 2020; Shiguay, 2019).
2. Programa educativo de práctica y simulación: las plataformas especializadas en codificación entran en esta categoría, pues entre los servicios que ofrecen se encuentra la posibilidad de practicar en línea la codificación (Franco et al. 2020; Massachusetts Institute of Technology, s. f.; Pérez-Narváez et al., 2020).
3. *Software* educativo de tutorial: en esta categoría están los cursos en línea especializados, pues su principal función es la de ser tutoriales de los temas de interés (Codewars, Codecademy, edX, Khan Academy, Udemy, Coursera, Dash, Bento.io, Code Avengers). Esta categoría tiene un amplio potencial para aplicar técnicas de inteligencia artificial, ya que son usadas por miles de usuarios y se generan almacenes de datos listos para explorar e identificar en ellos patrones de comportamiento durante el aprendizaje.
4. Juegos serios: *software* de juegos educativos (Pilas Bloques, Alice, Greenfoot) (Kuz & Ariste, 2021; Torbado, 2021; Xinogalos & Tryfou, 2021). En esta categoría se pueden aplicar técnicas de gamificación para generar motivación y aumentar el interés de los alumnos.

5. Programas trazadores de código: este tipo de programas usa animaciones para dar seguimiento a las sentencias de los programas (BlueJ, Jeliot, Greenfoot) (Aguilar Juárez et al., 2022).

No obstante los esfuerzos invertidos en el desarrollo de *software* con fines educativos, no basta únicamente con aprovechar estas herramientas si estas no se acompañan de estrategias didácticas diseñadas para facilitar la adquisición del lenguaje de programación por parte de los estudiantes. También es importante evitar la desmotivación y los obstáculos en el aprendizaje de la programación. Algunos de los modelos de aprendizaje que pueden aplicarse son el aprendizaje basado en *software*, el aprendizaje basado en resolución de problemas y el aprendizaje basado en juegos (Ibarra-Zapata et al., 2021).

Asimismo, es necesario formar al alumno para que él mismo conozca y regule sus propios procesos de aprendizaje. Para ello, se debe promover la capacidad de aceptar la frustración que provoca el error para sentirse capaces de superar esas fallas, realizar un trabajo colaborativo y fortalecer la autoestima. De esta manera, se puede tener una autoevaluación y coevaluación de su ritmo de trabajo (García Mauri, 2020).

En consecuencia, además de usar *software* educativo, es necesario proveer al alumno de herramientas que faciliten su autogestión y un acompañamiento didáctico constante para poder aclarar dudas e inquietudes en el proceso de aprendizaje. Así es como surge la propuesta de aprovechar los asistentes conversacionales que aplican la inteligencia artificial en la interacción con los estudiantes de programación.

3.2 Agentes conversacionales

El principal objetivo de los agentes conversacionales es establecer un diálogo con los usuarios para proveer información. La funcionalidad del procesamiento de lenguaje natural (*natural language processing*, NLP) implica la utilización de reglas lingüísticas y algoritmos inteligentes para procesar un texto sin procesar en un lenguaje natural, de manera que se le pueda atribuir un valor significativo mediante técnicas de aprendizaje automático, como el *machine learning* (ML), el cual es un enfoque de análisis de datos que permite automatizar diversas tareas y comprender conceptos para generalizar comportamientos a partir de los datos recopilados (Arredondo Castillo, 2021).

El NLP es una rama multidisciplinaria que se relaciona estrechamente con la lingüística, las ciencias cognitivas, la psicología, la filosofía y las matemáticas, en especial con la lógica (Singh, 2022, p. 17). Su objetivo es analizar el lenguaje humano mediante programas informáticos. Pese a las dificultades no resueltas, se han logrado aplicaciones funcionales con dominios muy acotados, como programas de reconocimiento óptico de caracteres, revisores de ortografía y gramática, identificadores de sentimientos, generadores de documentos, traductores y recuperadores de texto.

En el sector educativo, el texto es el principal medio de intercambio de ideas; por eso, en este documento se aborda el procesamiento informático sobre el texto. El procesamiento de texto identifica y asigna a cada unidad textual sus propiedades lingüísticas, como son la función fonológica, morfológica, sintáctica, semántica y pragmática; posteriormente, se obtiene e infiere información mediante la inteligencia artificial, la cual frecuentemente se entrena en nuevos documentos textuales o se usa para enriquecer los modelos de representación.

Una de las aplicaciones del procesamiento del lenguaje natural de mayor éxito son los generadores de diálogo, es decir, programas usados para lograr interfaces humano-computadora amigables que automatizan la interacción de un usuario común con una empresa, un sistema computacional o la entidad interesada en atender a muchos usuarios en cualquier hora y lugar. Coloquialmente llamados *chatbots*, se caracterizan por establecer un diálogo limitado a un dominio previamente establecido, por ejemplo, la compra de un producto, la respuesta a dudas frecuentes, el cumplimiento de órdenes, etcétera.

Desde un punto de vista técnico, un chatbot se define como un programa inteligente que se comunica con un usuario en lenguaje natural a través de texto (chat) que imita una conversación humana utilizando NLP (Villón Cabrera, 2020). Existen también los agentes conversacionales, que son un sistema de diálogo generado con NLP dirigido a una tarea; es decir que, además, están basados en una máquina finita de estados, que tiene una estructura previamente programada (Alonso Astruga, 2021).

La diferencia entre uno y otro es que en los chatbots se aplican técnicas de inteligencia artificial (AI, por sus siglas en inglés) para responder dentro de una conversación humana con un conjunto de respuestas predefinidas y coherentes, que simulan comprender dicha interacción mediante el análisis del contexto en el discurso. En cambio, los agentes conversacionales se limitan a una sola estructura de diálogo y responden a ella; si dentro de la conversación el usuario sale del contexto, el agente conversacional ignorará la entrada o la malinterpretará respondiendo de forma equivocada.

Los chatbots pueden funcionar de dos formas (Villón Cabrera, 2020): la primera se basa en reglas y la segunda, en máquinas inteligentes. Los chatbots basados en reglas entregan respuestas predefinidas en una base de conocimiento extrayendo las palabras clave emitidas por el usuario. Por su parte, los basados en máquinas utilizan la AI para tener un análisis de lenguaje natural mediante reglas para mejorar la interacción humano-máquina y generarán un diálogo con una finalidad bien definida.

Orozco González (2020) clasifica a los chatbots como agente conversacional o como asistentes virtuales. El primero está diseñado para simular una conversación inteligente mediante textos o audios, mientras que los segundos ayudan a buscar información en los recursos web o responden a comandos ya establecidos. En el ambiente educativo,

los chatbots se pueden utilizar como asistentes virtuales de clase, sistemas de tutorías inteligentes, sistemas de compañeros de aprendizaje u objetos de aprendizaje.

En la actualidad, hay varias estrategias de desarrollo de estos programas. La de mayor aceptación es usar los servicios de plataformas que se encargan de encapsular la complejidad del procesamiento y que ofrecen servicios de diseño y publicación de chatbots. Las principales empresas que ofrecen estos servicios de inteligencia son las siguientes:

- IBM Watson
- Microsoft Azure
- Google: Tensor Flow, Cloud AI (Artificial Intelligence), Cloud Natural Language API
- Amazon Lex
- Facebook: Wit.ai
- Otras: Motion, Smooch, Gupshup, Botkit, Rasa (Martín, 2017), Api.ai, Semantic Machines, Digital Genius, Chatfuel, Pypestream, Pandorabots, AgentBot, ChatterBot o ChatScript (Davydova, 2017)

3.2.1 La arquitectura de los chatbots

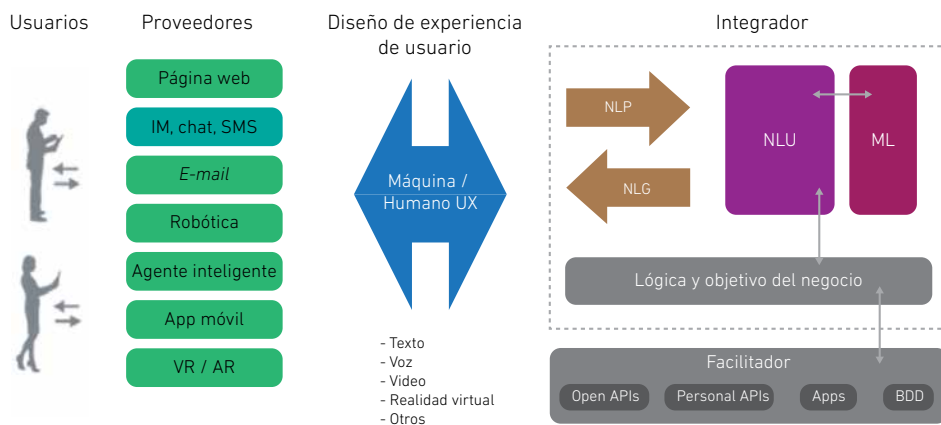
El objetivo de los chatbots es establecer una conversación con el usuario que sea semejante a un diálogo humano-humano, por lo que internamente se forman módulos que construyan dicha conversación. Si bien existen plataformas que encapsulan el proceso, es implícito que el chatbot debe vincularse con módulos que implementan la inteligencia artificial conversacional, en los cuales se produce el procesamiento de lenguaje natural.

Los chatbots más antiguos se enfocaban en la interpretación y reconocimiento de patrones y reglas predefinidas. Actualmente, se utiliza *deep learning* y *machine learning* para generar una respuesta adecuada (Nieves, 2018), que hace posible establecer una conversación más natural y coherente mediante el uso de la interfaz de usuario, donde se puede ver o escuchar las conversaciones con el chatbot.

La Figura 1 representa la arquitectura de un chatbot, donde se muestra el encapsulamiento de las entidades que participan. Estas son el proveedor, el integrador y el facilitador (Cornejo, 2018).

Figura 1

Adaptación de arquitectura de un chatbot



Nota. Reproducido de *Arquitectura de un chatbot*, por P. Cornejo, 2018, Medium (<https://medium.com/@patcornejo/arquitectura-de-un-chatbot-cb2d1c5f86c7>).

El proveedor es la interfaz con la cual el usuario entra en contacto con el chatbot; enseguida se encuentra una capa intermedia de experiencia de usuario (UX, por sus siglas en inglés), en donde se define cómo se mostrará y cómo se comunicará el chatbot con el usuario, puede ser mediante texto, imágenes o voz. El integrador es la pieza fundamental, ya que en él se encuentran las herramientas de procesamiento de lenguaje natural, *machine learning* y *deep learning*, que son la parte de inteligencia del chatbot.

Como parte del procesamiento de lenguaje natural (NLP), se encuentran tanto el módulo de NLU (*natural language understanding*) como el de NLG (*natural language generation*); una vez procesada, la entrada pasa al NLU, en donde se identifica la intención que tiene el usuario, en otras palabras, lo que el usuario quiere (la razón por la que habla o escribe al chatbot). El funcionamiento del NLU va de la mano con *machine learning* no supervisado de clasificación, dado que existe un banco de datos con posibles intenciones del usuario, las cuales son previamente definidas y clasificadas; mediante algoritmos se asigna un puntaje a la intención que el usuario trata de comunicar y, a partir de estos cálculos, se le entrega información al NLG.

La respuesta al usuario se genera con el NLG, en el cual al identificar la intención clasificada se seleccionan las respuestas preprogramadas, que bien pueden ser personalizadas o generales; la finalidad de este módulo es generar la respuesta en lenguaje natural en diferentes formatos, ya sea texto o audio.

El funcionamiento de un chatbot se produce por la integración de reglas preestablecidas y los algoritmos de AI, por lo que debe ser constantemente entrenado para que la generación de respuestas sea más precisa y corresponda a las intenciones del usuario. Si bien se requiere de la optimización de algoritmos, la aplicación de *deep learning* con diferentes tipos de entrenamiento y la implementación de diversas teorías lingüísticas, las plataformas integradoras son las que en la actualidad se encargan de realizar estas tareas complejas. Para los desarrolladores, son transparentes las herramientas que se proporcionan como servicios de creación de los chatbots.

Finalmente, el facilitador no es más que el intermediario entre el integrador y el banco de datos, que puede estar alojado en una base de datos, en la plataforma para crear chatbots, en una API que consulte algún servicio o en alguna aplicación general o personal, en donde se encuentre toda la información precargada de las respuestas con respecto a las intenciones del usuario.

Aunque existen herramientas para que un autor independiente genere su propio chatbot, las respuestas de su programa serían poco competitivas comparadas con los servicios que ofrecen las plataformas, ya que en la actualidad se está invirtiendo muchos recursos humanos y financieros en aumentar el nivel y complejidad del NLP que ofrecen las empresas.

4. APLICACIÓN DE CHATBOTS EN EL ÁMBITO EDUCATIVO

Uno de los primeros chatbots, llamado ELIZA, se desarrolló a partir del año 1964 con capacidad para mantener conversaciones en idioma inglés sobre una amplia variedad de temas, empleaba etiquetas para representar y clasificar los textos; y, además, fue configurado para actuar como un psiquiatra (Thorat & Jadhav, 2020). Otro ejemplo fue ALICE, inspirado en ELIZA, que se basaba en un patrón de selección sin percepción de conversación, pero con la habilidad de obtener información sobre cualquier tema en la web y generar así una discusión con los usuarios (Adamopoulou & Moussiades, 2020). Después de ese chatbot, fueron surgiendo con el paso de los años y de las nuevas tecnologías lo que hoy conocemos como asistentes personales de voz, entre los cuales podemos mencionar a Siri de Apple, a Cortana de Microsoft o a Alexa de Amazon (Adamopoulou & Moussiades, 2020).

El uso de los chatbots como herramienta de gestión educativa y didáctica es cada vez más frecuente. En esta sección, se listan algunos de los trabajos que han explorado los beneficios de usar chatbots en el sector educativo:

- La ProfeBot: es un chatbot con inteligencia artificial que permite al estudiante acceder a los conceptos y preguntas frecuentes relacionados con la materia que se implementó para el primer año de la Carrera de Licenciatura en Protección Civil y Emergencias en Buenos Aires (Lima & Rodríguez, 2021) .

- Ani: es un chatbot diseñado y creado con el propósito de brindar un enfoque de aprendizaje con el objetivo de ofrecer tutoría personalizada; ayuda, motiva y retroalimenta a los estudiantes. Fue aplicado en España (Garcia Brustenga et al., 2018).
- Botter: es un robot físico para ayudar a los estudiantes de la Universitat Oberta de Catalunya; interactúa por medio de sonidos, luces y frases. Su objetivo es el acompañamiento en el progreso del aprendizaje de los estudiantes (Garcia Brustenga et al., 2018).
- Bot de la Universidad CEU Cardenal Herrera: es un chatbot desarrollado en Microsoft Azure para responder consultas del alumno; actúa como asistente personal desde el 2017 (Garcia Brustenga et al., 2018).
- CourseQ: diseñado en la Universidad Cornell (Estados Unidos), es un bot conversacional basado en texto que permite conocer los horarios, eventos y fechas de entregas de trabajos tanto a los estudiantes como a los profesores (Garcia Brustenga et al., 2018).
- Differ: empleado en la BI Norwegian Business School, crea comunidades, envía recordatorios y publica mensajes. Su objetivo es crear un espacio seguro para estudiantes sin que existan prejuicios de por medio (Garcia Brustenga et al., 2018).
- Duolingo: es una aplicación que permite el aprendizaje de distintos idiomas; utiliza técnicas de gamificación para una mejor interacción con el estudiante (Garcia Brustenga et al., 2018).
- Genie: diseñado por la Deakin University de Australia, responde a las preguntas que los estudiantes tienen acerca de la escuela; se desarrolló con IBM Watson (Garcia Brustenga et al., 2018).
- Ivy: fue diseñado para la enseñanza en el nivel superior; su objetivo es responder y apoyar a los estudiantes sobre trámites administrativos y configuración de aplicaciones; además, cuenta con un apartado de preguntas frecuentes (Garcia Brustenga et al., 2018).

Estos chatbots se encuentran disponibles para su uso; sin embargo, el objetivo que se cubre con ellos son trámites administrativos, lo que es un uso de asistente personal. El reto para estas aplicaciones es su uso en la enseñanza-aprendizaje dentro del ámbito educativo. Para ello, en este trabajo se identificó un área de oportunidad para utilizar chatbots en el aprendizaje de programación.

5. CONCLUSIONES

Aprender a programar es un reto para los alumnos de licenciaturas relacionadas con la computación en la educación superior. No obstante, la presencia de la computación como herramienta que potencializa el desarrollo de las disciplinas ha generado la necesidad de aprender a programar en casi todas las áreas de estudio y desde edades tempranas; pero si en la programación los contenidos presentados por los profesores son poco didácticos, mal expuestos y se desaprovecha el *software* educativo existente, se pierden los beneficios de la tecnología en las diversas esferas del desarrollo humano.

Con el surgimiento de las computadoras, se ha tenido la posibilidad de procesar datos textuales de forma automática. Así se originó una rama multidisciplinaria conocida como NLP, cuyo objetivo es analizar y procesar el lenguaje humano mediante programas informáticos para identificar, asignar valor y significado al texto a fin de formar respuestas en lenguaje natural. Los sistemas creados mediante las teorías y técnicas de esta disciplina resultan exitosos en entornos acotados a un dominio específico; aún no se logran aplicaciones de dominio general que procesen eficientemente la ambigüedad del lenguaje natural. Los principales retos por superar son encontrar las mejores técnicas para representar el conocimiento humano con un buen tratamiento de la ambigüedad, integrar eficientemente datos de entrada de diferente naturaleza y generar salidas coherentes y estructuradas comprensibles para el usuario.

Una de las aplicaciones más exitosas del NLP son los chatbots, programas que se comunican con un usuario para imitar una conversación humana. Para ello, existen diversas plataformas que ofrecen servicios de diseño y alojamiento que están propiciando el uso generalizado de estos sistemas en múltiples servicios.

La arquitectura de un chatbot se compone de módulos con tareas específicas: una interfaz de usuario, un procesamiento del lenguaje natural desde la entrada hasta su salida. Su objetivo es identificar la intención del usuario y generar una respuesta coherente con ella; requiere de programas de preprocesamiento de entrada de datos, una base de conocimiento, modelos lingüísticos y algoritmos de aprendizaje automático.

Una de las dificultades del uso de NLP es la optimización de los algoritmos que se requiere para tener un mejor entrenamiento y respuestas más precisas y coherentes cuando se interactúa con un chatbot. Sin embargo, esta optimización actualmente es responsabilidad de las empresas que están ofreciendo estos servicios e invirtiendo en múltiples recursos. El sector académico puede aportar en el desarrollo de teorías lingüísticas, algoritmos de optimización y aprendizaje automático, además de propuestas de diseño instruccional que faciliten, por un lado, la eficiencia de los programas y, por otro, la adopción de estos programas en las actividades de estudio de los estudiantes.

Los docentes sin experiencia en tecnología pueden usar las plataformas para acceder a los servicios inteligentes de vanguardia que las empresas ofrecen, ya sea

cursos MOOC (*Massive Open Online Courses*, por sus siglas en inglés) orientados a la codificación, o juegos serios, trazadores de algoritmos, *software* de modelado o gestores de chatbots, de manera que aprovechen la disponibilidad de múltiples herramientas de *software* para innovar su labor docente.

Es un reto difundir estas alternativas y convencer al sector docente de que estas herramientas son un apoyo más en su vida diaria, semejantes a los procesadores de texto o de imágenes que pueden facilitar y enriquecer su didáctica. La clave radica en incorporar este *software* de manera estructurada y estratégica en el proceso de aprendizaje de los estudiantes, con el fin de obtener los mayores beneficios posibles. La exploración del uso de *software* educativo en el aprendizaje de la programación es necesaria para identificar buenas prácticas, proponer nuevas formas de evaluación del aprendizaje de los alumnos y aumentar la motivación de aprender.

Los siguientes trabajos de investigación se encaminarán a emplear un chatbot especialmente diseñado para atender consultas relacionadas con la programación orientada a objetos, con la finalidad de ofrecerlo a nuestra comunidad y valorar los beneficios que los alumnos identifiquen de este apoyo en sus actividades de aprendizaje.

REFERENCIAS

- Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2, 100006. <https://doi.org/10.1016/J.MLWA.2020.100006>
- Aguilar Juárez, I., Rojas Espinoza, B. A., & Ayala de la Vega, J. (2022). Una experiencia del aprendizaje móvil como apoyo para el estudio de la programación. *Revista Iberoamericana de Producción Académica y Gestión Educativa*, 9(18), 44-63. <https://www.pag.org.mx/index.php/PAG/article/view/899>
- Alonso Astruga, J. (2021). *Propuesta metodológica para el análisis y diseño de chatbots basados en texto* [Tesis de maestría, Universidad de Valladolid]. Repositorio Documental. Universidad de Valladolid. <https://uvadoc.uva.es/handle/10324/50064>
- Arredondo Castillo, C. C. (2021). *Inteligencia artificial en la educación: uso del chatbot en un curso de pregrado sobre Investigación Académica en una universidad privada de Lima* [Tesis de maestría, Pontificia Universidad Católica del Perú]. Repositorio Digital de Tesis y Trabajos de Investigación PUCP. <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/20996>
- Ballesteros, V. A., Rodríguez, O. I., Lozano, S., & Nisperuza, J. L. (2020). El aprendizaje móvil en educación superior: una experiencia desde la formación de ingenieros. *Revista Científica*, 38(2), 243-257. <https://doi.org/10.14483/23448350.15214>

- Benítez, R. P., Torres, V. J., Camacho, F. Y., & Ramírez, V. (2009). *La influencia de las estrategias de instrucción sobre la motivación de los estudiantes en un curso de programación Java: un caso de estudio*. XXII Congreso Nacional y VIII Congreso Internacional de Informática y Computación, Ensenada, Baja California, México.
- Cornejo, P. (2018, 3 de abril). *Arquitectura de un chatbot*. Medium. <https://medium.com/@patcornejo/arquitectura-de-un-chatbot-cb2d1c5f86c7>
- Davydova, O. (2017, 11 de mayo). 25 chatbot platforms: A comparative table. *Chatbots Journal*. <https://chatbotsjournal.com/25-chatbot-platforms-a-comparative-table-aeefc932eaff>
- Franco, D., García, D. G., Guevara, C. F., & Erazo, J. C. (2020). Scratch para la enseñanza de lenguaje de programación en primero de bachillerato. *Revista Arbitrada Interdisciplinaria Koinonía*, 5(5), 398-414. <https://doi.org/10.35381/r.k.v5i5.1050>
- García Brustenga, G., Fuertes-Alpiste, M., & Molas-Castells, N. (2018). *Briefing paper: chatbots in education*. Universitat Oberta de Catalunya. <https://doi.org/10.7238/elc.chatbots.2018>
- García Mauri, D. R. (2020). Aprender a aprender. *Revista Referencia Pedagógica*, 8(2), 203-218. <https://rrp.cujae.edu.cu/index.php/rrp/article/view/212>
- Gómez, M. J. (2020). *Aspectos de adquisición de lenguaje en la enseñanza de programación* [Tesis doctoral, Universidad Nacional de Córdoba]. Repositorio Digital UNC. <https://rdu.unc.edu.ar/handle/11086/16051>
- Ibarra-Zapata, R. E., Castillo-Cornelio, J. O., Trujillo-Natividad, P. C., García-Villegas, C., Yanac-Montesino, R., & Pando, B. (2021). Enseñanza-aprendizaje de programación de computadoras: avances en la última década. *Revista Científica*, 42(3), 290-303. <https://doi.org/10.14483/23448350.18339>
- Jiménez-Toledo, J. A., Collazos, C., & Revelo-Sánchez, O. (2019). Consideraciones en los procesos de enseñanza-aprendizaje para un primer curso de programación de computadores: una revisión sistemática de la literatura. *Tecnológicas*, 22, 83-117. <https://doi.org/10.22430/22565337.1520>
- Kuz, A., & Ariste, M. C. (2021). Un análisis desde la programación estructurada del lenguaje Scratch como entorno lúdico educativo. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa*, 33, 14-21. <https://dialnet.unirioja.es/servlet/articulo?codigo=7985875&>
- Lima, M. R., & Rodríguez, J. M. (2021). La ProfeBot, un chat para la educación. *INNOVA UNTREF. Revista Argentina de Ciencia y Tecnología*, 1(8). <https://www.revistas.untref.edu.ar/index.php/innova/article/view/1240>

- Mancuzo, G. (2022, 7 de enero). Los 5 tipos de *software* educativos más usados. *Compara Software*. <https://blog.comparasoftware.com/los-5-tipos-de-software-educativos-mas-usados/>
- Martín, J. (2017). Estado del arte en el desarrollo de chatbots a nivel mundial. *Futurizable*. <https://futurizable.com/chatbot/>
- Massachusetts Institute of Technology. (s. f.). *Scratch*. <https://scratch.mit.edu/>
- Nieves, B. (2018, 3 de mayo). *IA conversacional: definición y conceptos básicos*. Planeta ChatBot. <https://planetachatbot.com/ia-conversacional-conceptos-basicos-y-definicion/>
- Orozco González, M., Pytel, P., & Pollo-Cattaneo, M. F. (2020). Metodología de implementación de un chatbot como tutor virtual en el ámbito educativo para universidades en Latinoamérica. En E. Serna (Ed.), *Investigación formativa en ingeniería* (4.ª ed., pp. 218-226). Editorial Instituto Antioqueño de Investigación. <https://doi.org/10.5281/zenodo.4031253>
- Pérez-Narváez, H. O., Roig-Vila, R., & Jaramillo-Naranjo, L. (2020). Uso de Scratch en el aprendizaje de programación en educación superior. *Cátedra*, 3(1), 28-45. <https://doi.org/10.29166/10.29166/catedra.v3i1.2006>
- Shiguay, G. A. (2019). *Las tecnologías de la información y comunicación en el aprendizaje de algoritmos y diagramas de flujo en el curso de Informática V en los estudiantes del área de tecnología e informática del quinto grado de educación secundaria de la Institución Educativa Emblemática Ricardo Bentín Sánchez, Rímac, 2018* [Tesis de maestría, Universidad Nacional de Educación Enrique Guzmán y Valle]. Repositorio Institucional. Universidad Nacional de Educación Enrique Guzmán y Valle. <http://repositorio.une.edu.pe/handle/20.500.14039/3465>
- Singh, A. (2022). *Procesamiento de lenguaje natural con Python*. Bebelcube. <https://www.kobo.com/mx/es/ebook/procesamiento-de-lenguaje-natural-con-python>
- Thorat, S. A., & Jadhav, V. D. (2020). A review on implementation issues of rule-based chatbot systems. En *Proceedings of the International Conference on Innovative Computing and Communications* (pp. 1-6). <https://ssrn.com/abstract=3567047>
- Torbado, M. (2021). *Gamificación de fundamentos de la programación: juegos serios para el aprendizaje de estructuras de código iterativas, métodos y funciones* [Tesis de licenciatura, Universidad de Valladolid]. Repositorio Documental. Universidad de Valladolid. <https://uvadoc.uva.es/handle/10324/50440>
- Villón Cabrera, N. (2020). *Inteligencia artificial aplicada al marketing: impacto del uso de chatbots cognitivos en la satisfacción del cliente del sector bancario* [Tesis de

grado, Universidad Peruana de Ciencias Aplicadas]. Repositorio Académico UPC.
<https://repositorioacademico.upc.edu.pe/handle/10757/652700>

Xinogalos, S., & Tryfou, M. M. (2021). Using Greenfoot as a tool for serious games programming education and development. *International Journal of Serious Games*, 8(2), 67-86. <https://doi.org/10.17083/IJSG.V8I2.425>