

OpenChatBotUNL: Propuesta de plataforma de ejecución de Agentes Conversacionales

Jose-Luis Granda¹, Luis Chamba-Eras¹, Milton Labanda-Jaramillo¹, Edison Coronel-Romero¹, Rene Guaman-Quinche², Carlos Maldonado-Ortega²

{jose.l.granda,lachamba,miltonlab,edisoncor,rguaman,carlos.v.maldonado}@unl.edu.ec

¹ Grupo de Investigación en Tecnologías de la Información y Comunicación (GITIC), Carrera de Ingeniería en Sistemas, Facultad de Energía, Universidad Nacional de Loja, Av. Pío Jaramillo Alvarado y Reinaldo Espinosa, EC110110, Loja, Ecuador.

² Carrera de Ingeniería en Sistemas, Facultad de Energía, Universidad Nacional de Loja, Av. Pío Jaramillo Alvarado y Reinaldo Espinosa, EC110110, Loja, Ecuador.

Pages: 1-14

Resumen: En este trabajo se propone una arquitectura de ejecución de Agentes Conversacionales (ChatBots) basada en herramientas Open Source, denominado OpenChatBotUNL. La propuesta integra en una arquitectura algunos paquetes de software como el protocolo XMPP, Openfire, martinbigio/chatterbot, Python ChatterBot, como soporte para la infraestructura de comunicación, gestión, monitoreo e inteligencia artificial. OpenChatBotUNL se plantea como una alternativa a las plataformas de implementación y ejecución (SDKs), disponibles bajo la modalidad de Software como Servicio, representando una oportunidad hacia la innovación en la línea de investigación de Procesamiento de Lenguaje Natural en las universidades en Ecuador.

Palabras-clave: Procesamiento de Lenguaje Natural; Inteligencia Artificial; Agente Conversacional; ChatBot; Open Source

OpenChatBotUNL: Propuesta de plataforma de ejecución de Agentes Conversacionales

Abstract: *In this work we propose an architecture of execution of Conversational Agents (ChatBots) based on Open Source tools, called OpenChatBotUNL. The proposal integrates various software package on an architecture, such as the XMPP protocol, Openfire, martinbigio/chatterbot, Python ChatterBot, as support for the communication, management, monitoring and artificial intelligence infrastructure. OpenChatBotUNL is proposed as an alternative to implementation and execution platforms*

(SDKs), available under the Software as a Service modality, representing an opportunity for innovation in the Natural Language Processing research line at universities in Ecuador.

Keywords: Natural Language Processing; Artificial Intelligence; Conversational Agent; ChatBot; Open Source

1. Introducción

Para crear un ChatBot, actualmente es necesario disponer de una cuenta en alguna red social y habilitar la aplicación correspondiente. Si se requiere algo más avanzado, basta un par de horas, en algunos de los SDK (Software Development Kit) disponibles en las plataformas de Amazon, Google, IBM, Facebook, Slack, Twitter, Whatsapp, WeChat (William Vorhies, 2017).

Fabricantes de SDK como IBM o Google, ofrecen versiones gratuitas, basadas en la nube, destinadas a reducidas cargas de trabajo. Una vez superados ciertos límites, se debe migrar a versiones empresariales (IBM, 2017). Adicional a la limitante de carga, los SDK ocultan la complejidad del software para materializar un agente conversacional (ChatBot). Así, los SDKs representan una caja negra, desde el punto de vista del investigador que pretenda agregar valor o innovar en la línea del procesamiento del lenguaje natural (PLN).

En países, en vías de desarrollo, como el Ecuador, la inversión en innovación, la adquisición, absorción, modificación y creación de conocimiento tecnológico y no tecnológico, son actividades indispensables para el desarrollo económico (Schwartz & Guaipatín, 2014), en esta misma línea, el impulso a la ciencia abierta trae consigo prácticas de innovación que pueden detonar en el desarrollo de los países, especialmente si se destierran los mitos que por diversos motivos se vienen asociando al movimiento del conocimiento en abierto (Ramírez-Montoya & García-Peñalvo, 2018).

En la industria del software, la filosofía de software Open Source, presenta una oportunidad para adquirir, absorber, modificar y construir tecnología. En este trabajo, se propone una plataforma de ejecución de agentes conversacionales, *OpenChatBotUNL* construida mediante la integración de software Open Source. La plataforma tiene por objetivo transparentar el acceso a la complejidad, de la caja negra, de los SDK de los grandes fabricantes de software, al mismo tiempo de empoderar al investigador académico en estas tecnologías con miras a futuras innovaciones.

La organización del trabajo es la siguiente: la sección 2, hace una revisión de literatura respecto a los agentes conversacionales y sus ambientes de ejecución. Así

también, se describe software Open Source de soporte a funcionalidades de mensajería instantánea como candidatos a integrar la plataforma propuesta. La sección 3, plantea el diseño de arquitectura de la plataforma *OpenChatBootUNL* y ejemplos de algunas funcionalidades esperadas. Finalmente, en la sección 4, se mencionan las conclusiones y posibles líneas futuras.

2. Revisión de literatura

Un agente conversacional, a simple vista, se puede percibir como un conjunto de interfaces de usuario que nos permite enviar y/o recibir un mensaje, a través de un mecanismo de entrada/salida de cómputo, con cierta “inteligencia”.

Mas ¿qué hay debajo de las interfaces? ¿Es tan sencillo, implementar un agente conversacional, conforme a la promesa de los gigantes del software? Para responder estas preguntas y plantear una alternativa a la ejecución de los agentes conversacionales, es importante compendiar el estado actual de los conceptos y desarrollos de software en esta línea.

2.1. Agentes conversacionales (ChatBots).

Los agentes conversacionales (ChatBots), son tecnologías que se usan para conseguir una experiencia de usuario más amigable (Kerlyl, Hall, & Bull, 2006). Desde la acuñación del término Inteligencia Artificial (IA), en los años 50, por Jhon Macarthy; y el planteamiento del test para etiquetar como “inteligencia” a una máquina, propuesto por Alan Turing (Russell & Norvig, 2004), una de las primeras aplicaciones, en la línea de conversaciones inteligentes, capaz de entablar una conversación básica, fue ELIZA en 1965 (Russell & Norvig, 1995). Posteriormente ALICE, extendió la base de conocimiento de ELIZA e introdujo el Lenguaje de Marcas de IA, AIML (AI Markup Language). Actualmente, extensiones de este lenguaje soportan consideraciones de contexto, ramificaciones condicionales y aprendizaje supervisado para producir nuevas respuestas (Kerlyl et al., 2006).

Otros ejemplos de agentes conversacionales son Microsoft's Clippy de MS Office97; algunos más recientes y populares como Alexa, Cortana, Siri, and Google Assistant (William Vorhies, 2017). Estos últimos emplean técnicas de aprendizaje automático, para monitorear y cambiar dinámicamente los patrones (pregunta, respuesta) y así mejorar la experiencia del usuario.

En (William Vorhies, 2017) se hace referencia al estudio de 2017 de la empresa Gartner, donde se concluye “que todavía estamos en una curva ascendente para los ChatBots. Sin embargo, todos los principales proveedores de redes o redes sociales, incluidos Amazon, Google, IBM, Facebook, Microsoft, Slack, Twitter, Whatsapp, WeChat y una gran cantidad de independientes, ofrecen SDK fáciles de usar para

desarrollarlos. El campo está evolucionando rápidamente tanto en términos de tecnología como de adopción¹. También se mencionan los campos de aplicación, en el contexto empresarial: servicio al cliente, ventas/mercadotecnia, procesamiento de pedidos, redes sociales, pagos y contratación. En el ámbito educativo: tutoría, preguntas y respuestas, prácticas de conversación en estudio de idiomas, agentes pedagógicos y acompañamiento de aprendizaje, diálogos para fortalecer las habilidades cognitivas y de reflexión, entre otros. En general, estos agentes son capaces de ofrecer apoyo personalizado para cada individuo, reconocer y construir sobre las fortalezas, los intereses y habilidades de las personas con el fin de fomentar la participación y aprendizajes independientes (Kerry, Ellis, & Bull, 2009).

2.2. Software Open Source para Mensajería Instantánea.

La mensajería instantánea representa un conjunto de interfaces para interactuar con los agentes conversacionales. Aplicaciones como WhatsApp, Facebook Messenger, Hangouts son parte de la vida cotidiana de los “conectados”¹. Usar estas aplicaciones resulta trivial, gracias al fuerte soporte de infraestructura software, como: protocolos de comunicación, servidores de colaboración en tiempo real, panel de administración, motores de PLN, entre otros. A continuación, la descripción de algunas opciones de software de soporte, disponible en modalidad Open Source:

XMPP

Son las siglas para el protocolo extensible de mensajería y presencia. El protocolo está basado en XML y permite el intercambio casi en tiempo real de datos estructurados pero extensibles entre dos o más entidades de red (Saint-Andre, 2011).

El protocolo central de XMPP soporta: instalación y desmontaje de flujos XML, cifrado de canal, autenticación, manejo de errores y comunicación primitivas para mensajería, disponibilidad de red (“presencia”), interacciones solicitud-respuesta (Saint-Andre, 2011). Mayor detalle de la especificación de XMPP, se desarrolla en (Xuefu & Ming, 2012).

¹ Los usuarios de Internet y dispositivos móviles

Openfire

Es un servidor de colaboración en tiempo real (RTC), con licencia de Open Source Apache License, que utiliza como protocolo de mensajería XMPP (Realtime, Ignite, 2008).

Openfire representa una alternativa al dominio de las grandes compañías de software en el área de Mensajería Instantánea Empresarial (MIE). En (Xuefu & Ming, 2012) se hace notar la diferencia entre Mensajería Instantánea (MI) y la Mensajería instantánea Empresarial, estos últimos implican mayores requerimientos en cuanto a funcionalidades extras como envió/recepción de archivos, vídeo conferencia, entre otros. En la presente propuesta nos limitaremos a la MI como interface del agente conversacional.

Aplicación martinbigio/chatbot²

Es un agente inteligente simple capaz de chatear con los clientes de un centro de llamadas y cuyo comportamiento se puede monitorear y personalizar a través de la interfaz Web. El servicio de mensajería instantánea esta implementada sobre OpenFire.

Para administrar el sistema, se cuenta con una aplicación Web como panel de control. En ella se pueden administrar los usuarios del sistema, monitorear las conversaciones de los agentes de chat (chatbots) en tiempo real, consultar los registros y estadísticas del sistema y configurar la semántica del lenguaje que deben interpretar los chatbots (Bigio, 2012/2018).

Python ChatterBot

Es una biblioteca de Python que facilita el generar respuestas automáticas a la entrada de un usuario. Python ChatterBot utiliza una selección de algoritmos de aprendizaje automático para producir diferentes tipos de respuestas. Permite crear agentes conversacionales (ChatBots) y automatizar las conversaciones con los usuarios.

Una instancia no entrenada de agente conversacional, comienza sin conocimiento de cómo comunicarse. Cada vez que un usuario ingresa una declaración, la biblioteca guarda el texto ingresado y el texto al que la declaración respondió. A medida que ChatterBot recibe más información, la cantidad de respuestas que

² Se usa el nombre de usuario GitHub y el repositorio de la aplicación, para diferenciarla de Python ChatterBot también descrito y usado posteriormente

puede responder y la precisión de cada respuesta en relación con la declaración de entrada aumentan³. El diagrama de flujo de procesos se detalla en la figura 1.

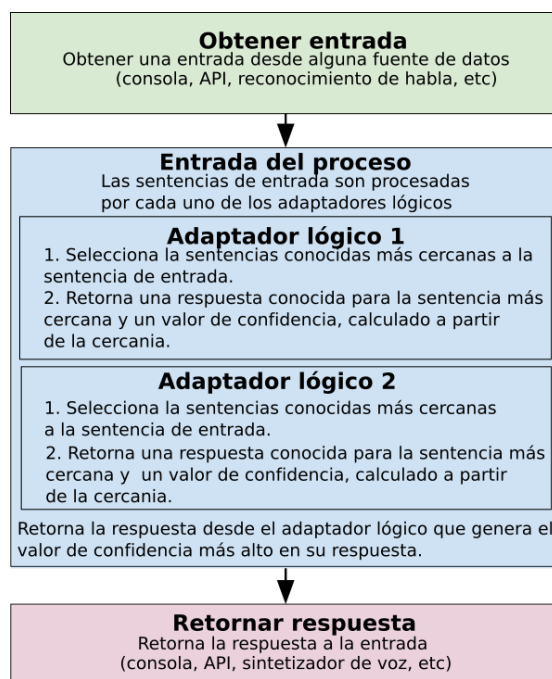


Figura 1 – Diagrama de flujo de procesos Python ChatterBot

Plataformas de ejecución de agentes conversacionales

Como se mencionó, los principales proveedores de redes o redes sociales y servicios de software, proveen plataformas sobre las cuales se puede implementar agentes conversacionales. El denominador común de estas plataformas es la modalidad de negocio, software como servicio o, en este contexto, ChatBot como servicio. Cabe

³ About ChatterBot — ChatterBot 0.8.7 documentation. (s. f.). Recuperado 26 de julio de 2018, de <https://chatterbot.readthedocs.io/en/stable/index.html>

recaltar que en su mayoría disponen de una versión de libre acceso, limitada en la cantidad de mensajes en las conversaciones.

En (Rahman, Al Mamun, & Islam, 2017) se hace una clasificación de las plataformas antes mencionadas, así: chats sin programación, chats orientados a conversaciones y plataformas de chat proveídas por los “gigantes”⁴.

3. Especificación de la arquitectura

3.1. Arquitectura

OpenChatBotUNL plataforma de ejecución de agentes conversacionales, se concibe como el resultado de la integración de herramientas de software disponibles bajo modalidad Open Source. La plataforma se basa en la arquitectura para MIE propuesta en (Xuefu & Ming, 2012). Esta arquitectura integra servicios basados en XMPP, clientes para protocolo XMPP e integración con otros servicios XMPP. La figura 2 muestra la propuesta de la arquitectura *OpenChatBotUNL*.

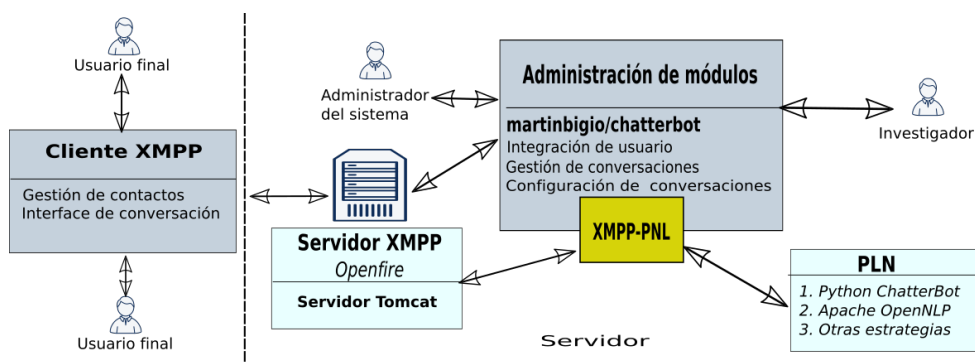


Figura 2 – Arquitectura de *OpenChatBotUNL*.

⁴ Se refiere a los gigantes de la industria del software, como: Google, Facebook, IBM, entre otros

El componente **XMPP-PLN** permitirá la conversación entre XMPP y algún modelo de PLN, en la primera implementación de la plataforma se propone *Python Chatterbot* como motor de PLN, adicionalmente *martinbigio/chatterbot* implementa PLN con la librería Java *Apache OpenNLP*⁵, así se amplían las posibilidades de la plataforma. El componente XMPP-PLN ha de implementarse empleando tecnología “*enchufar-usar*” (plugin), así será posible usar otras tecnologías como R, Octave u otras.

Se consideran 3 tipos de usuarios: el usuario final quien usa el chat, el administrador quien gestiona las conversaciones, la información y datos resultantes; y finalmente el investigador quien crea el o los agentes conversacionales y analiza los resultados. La identificación de estos roles de usuario, también determina las herramientas a proveer a cada uno de ellos. El usuario final, en producción, puede elegir entre la gran variedad de clientes XMPP⁶ (Ej.: pidgin, converse, entre otros) y adicionalmente, la plataforma debe implementar un cliente basado en Web para efectos de pruebas. El administrador cumple un rol de orquestador y sus herramientas son de naturaleza más técnica a fin de dar soporte tanto a los usuarios y/o investigadores. El investigador por su parte debe poder desplegar sus agentes conversacionales y/o ajustarlos en tiempo de ejecución, debe poder acceder a datos estadísticos de ejecución y al contenido de las conversaciones, a fin de poder hacer nuevas experimentaciones sobre sus modelos de agentes conversacionales u otros en la línea de PLN.

3.2. Vistas previas

En las figuras 3, 4 y 5, se observa los primeros prototipos basados en la plataforma *OpenChatBotUNL*.

5 Apache OpenNLP. Recuperado 08/11/2018. <https://opennlp.apache.org/>

6 XMPP Clientes. Recuperado 08/11/2018. <https://xmpp.org/software/clients.html>

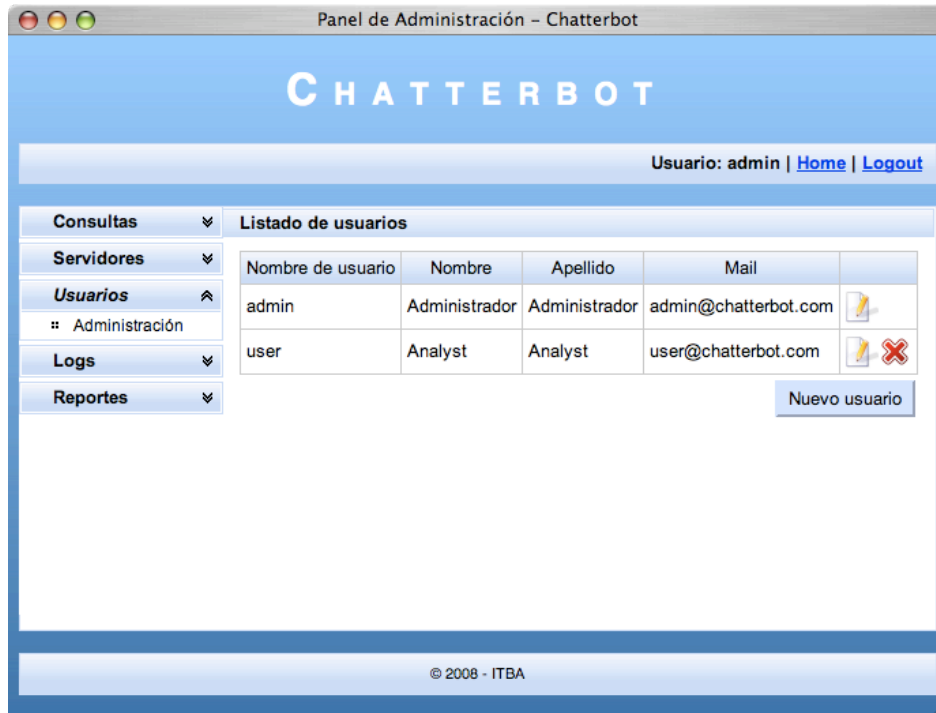


Figura 3 – Aplicación martinbigio/chatterbot: Panel de Administración

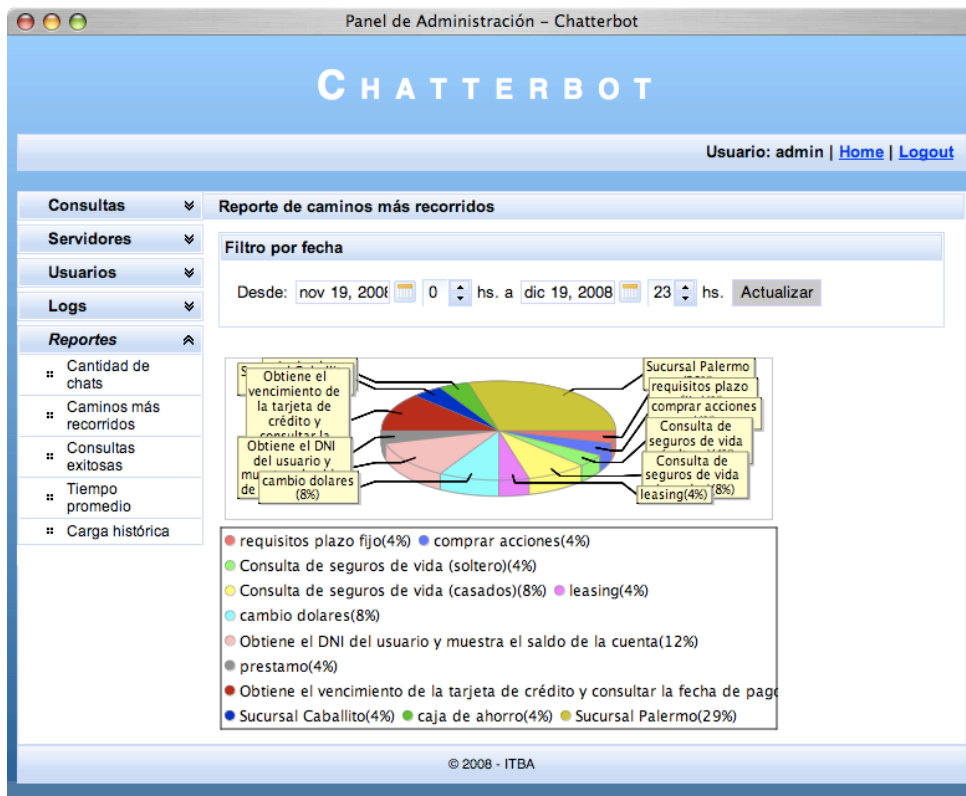


Figura 4- Aplicación martinbigio/chatterbot: Reportes y estadística

Detalle del chat

Servidor: server-test1

Cliente: jose@mail.com

Duración: 3 minutos (desde 16/12/2008 14:30 hasta 16/12/2008 14:33)

16/12/2008 14:30 Hola, como estas?

16/12/2008 14:30 Hola, bien.

16/12/2008 14:31 Esta abierta la sucursal?

16/12/2008 14:31 El horario de atención es de 8:00 a 16:00 hs.

16/12/2008 14:33 Realizan cambio de dólares?

16/12/2008 14:33 Se realizacn cambios de dólares. El cambio actual es de 3.14 U\$S.

Volver

Figura 5 – Aplicación *martinbigio/chatterbot*: Una conversación de chat

4. Conclusiones y líneas futuras

En los trabajos (Kerlyl et al., 2006), (Kerry et al., 2009; García et al., 2009; Lima, Algreto-Badillo, Moreno, Bautista, & Rosales, s. f.; Xu, Liu, Guo, Sinha, & Akkiraju, 2017; Acedo & Ramírez, 2017; Cambizaca & Francisco, 2018) se describen los detalles de implementación de agentes conversacionales en diferentes ámbitos de aplicación. Un factor común es la diversidad de plataformas de ejecución.

Disponer de una plataforma de ejecución de Agentes Conversacionales (ChatBots) implementada sobre software Open Source, proveída desde la academia, representa una oportunidad para propender a la innovación transparentando la complejidad de los componentes de la plataforma de ejecución. Este conocimiento puede ser un

motor para la innovación, la resolución de problemas y el planteamiento creativo de posibilidades (Ramírez-Montoya & García-Peñalvo, 2018).

En concepto la plataforma permitirá realizar experimentación sobre modelos de conversaciones a través de ChatBots, pese a que la plataforma no está integrada aún, el software *martinbigio/chatterbot* puede usarse de inmediato como versión beta, de estas interacciones se obtendrán datos e ideas para evolucionarla.

Un debate que empieza a tomar fuerza, es la conveniencia o no de entregar datos (en el contexto de IA: inteligencia) a los grandes fabricantes de software y sus herramientas de IA. La plataforma propuesta, permitiría evaluar si es posible o no alejarse de los grandes fabricantes.

No es descartable que, como fruto de la investigación, la plataforma pueda ser llevada a entornos de producción empresarial.

Agradecimientos

Este trabajo forma parte de las actividades académicas y de investigación del Grupo de Investigación GITIC, agradecemos al profesorado, autoridades y alumnado de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja (UNL).

Contribuciones de autoría

Jose-Luis Granda y Luis Chamba-Eras y concibieron la propuesta, la arquitectura, las vistas. Así mismo, Milton Labanda-Jaramillo, Edison Coronel-Romero, Rene Guaman-Quinche y Carlos Maldonado-Ortega, revisaron la redacción del texto y aportaron comentarios en la revisión de literatura sobre el objeto de estudio, la especificación de la arquitectura y otros elementos para la mejora el artículo y aprobaron la versión final para su publicación.

Referencias

Acedo, E. G., & Ramírez, P. S. (2017). BotMentor: Bot de ayuda al estudiante en la plataforma Telegram.

Bigio, M. (2018). *chatterbot: A simple intelligent agent for a call center, written in Java (final project at itba.edu.ar)*. Java. Recuperado de <https://github.com/martinbigio/chatterbot> (Original work published 2012)

Cambizaca, T., & Francisco, A. (2018). Desarrollo de un chatbot que ayude a responder a preguntas frecuentes referentes a becas en la Universidad Técnica

Particular de Loja. Recuperado de <http://dspace.utpl.edu.ec/handle/20.500.11962/21874>

Extensible Messaging and Presence Protocol. (2018). En *Wikipedia, la enciclopedia libre*. Recuperado de https://es.wikipedia.org/w/index.php?title=Extensible_Messaging_and_Protocol&oldid=108402227

García, E. G., López, R. J., Moreno, J. J. M., Abad, A. S., Blasco, B. C., & Pol, A. P. (2009). La metodología del Data Mining. Una aplicación al consumo de alcohol en adolescentes. *Adicciones*, 21(1), 65–80.

Realtime, Ignite. (2008). Ignite Realtime: Openfire Server. Recuperado 25 de julio de 2018, de <https://www.igniterealtime.org/projects/openfire/index.jsp>

Kerlyl, A., Hall, P., & Bull, S. (2006). Bringing Chatbots into education: Towards Natural Language Negotiation of Open Learner Models. En *Applications and Innovations in Intelligent Systems XIV* (pp. 179-192). Springer, London. https://doi.org/10.1007/978-1-84628-666-7_14

Kerry, A., Ellis, R., & Bull, S. (2009). Conversational agents in E-Learning. En *Applications and innovations in intelligent systems XVI* (pp. 169–182). Springer.

Lima, F. T., Algreto-Badillo, I., Moreno, N. T., Bautista, D. P., & Rosales, L. A. M. (s. f.). Recorrido Virtual y Agente Conversacional Usando Photosynth y Python para un Sitio Web Publicitario.

Open Source Initiative OSI - Jabber Open Source License:Licensing | Open Source Initiative. (s. f.). Recuperado 25 de julio de 2018, de <https://opensource.org/licenses/jabberpl.php>

Rahman, A. M., Al Mamun, A., & Islam, A. (2017). Programming challenges of chatbot: Current and future prospective. En *Humanitarian Technology Conference (R10-HTC), 2017 IEEE Region 10* (pp. 75–78). IEEE.

Ramírez-Montoya, M. S., & García-Peñalvo, F. J. (2018). Co-creación e innovación abierta: Revisión sistemática de literatura. Recuperado de <https://repositorio.grial.eu/handle/grial/1048>

Russell, S. J., & Norvig, P. (1995). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.

Russell, S. J., & Norvig, P. (2004). *Inteligencia Artificial: un enfoque moderno*.

Saint-Andre, P. (2011). *Extensible messaging and presence protocol (XMPP): Core*.

Wallace, R. S. (2009). The anatomy of ALICE. En *Parsing the Turing Test* (pp. 181–210). Springer.

William Vorhies. (2017, noviembre 7). Beginners Guide to Chatbots. Recuperado 10 de julio de 2018, de <https://www.datasciencecentral.com/profiles/blogs/beginners-guide-to-chatbots>

XMPP | The XMPP Blog. (s. f.). Recuperado 24 de julio de 2018, de <https://xmpp.org/2015/03/no-its-not-the-end-of-xmpp-for-google-talk/>

Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A New Chatbot for Customer Service on Social Media. En *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 3506–3510). New York, NY, USA: ACM. <https://doi.org/10.1145/3025453.3025496>

Xuefu, B., & Ming, Y. (2012). Design and implementation of web Instant Message System based on XMPP. En *2012 IEEE International Conference on Computer Science and Automation Engineering* (pp. 83-88). <https://doi.org/10.1109/ICSESS.2012.6269411>