



Sistema Inteligente de Tutoría para entrenamiento procedimental apoyado en un diálogo adaptado al contexto

Intelligent Tutoring System for procedural training supported in a dialogue adapted to the context

Jose Paladines Morán¹

Universidad Estatal del Sur de Manabí, Ecuador

Jaime Ramírez Rodríguez²

Universidad Politécnica de Madrid, España

Recibido: 01-10 -2017

Aceptado: 11-12 -2017

CITA RECOMENDADA

Paladines, J. y Ramírez, J. (2017) Sistema inteligente de tutoría para entrenamiento procedimental apoyado en un diálogo adaptado al contexto. *Hamut'ay*, 4 (2), 17-29.

Recuperado de: <http://dx.doi.org/10.21503/hamu.v4i2.1469>

RESUMEN

El objetivo de este documento es presentar una propuesta de un Sistema Inteligente de Tutoría para entrenamiento procedimental en un entorno virtual 2D/3D, capaz de mantener un diálogo en lenguaje natural basado en el contexto. De esta manera, cada alumno será capacitado por medio de un diálogo en lenguaje natural que tome en cuenta sus características específicas, su progreso en el desarrollo de la tarea y el entorno donde se realiza la tarea. Por lo tanto, la retroalimentación de tutoría será el resultado de un diálogo adaptado al contexto. Para dar soporte al diálogo, se utilizará un gestor de diálogo, construido sobre alguna de las plataformas conocidas para la creación de gestores de diálogo actualmente disponibles.

Palabras Clave: Sistema Inteligente de Tutoría, Procesamiento de Lenguaje Natural, Diálogo adaptado al Contexto.

ABSTRACT

In order to provide support environments for education, the aim of this paper is to present a proposal for an Intelligent Tutoring System for procedural training in a 2D / 3D virtual environment, capable of maintaining a dialogue in natural language based on the context. In this way, each student will be trained through a dialogue in natural language that takes into account the students' specific characteristics, their progress in the development of the task and the environment where the task is performed. Therefore, the tutoring feedback will be the result of a dialogue adapted to the context. To support the dialogue, a dialogue manager will be used, built on one of the currently available

¹ Ingeniero en Sistemas, Facultad de Ciencias Técnicas de la Universidad Estatal del Sur de Manabí, Ecuador. Estudiante de Doctorado en el programa de Software, Sistemas y Computación. E-mail: jose.paladines@unesum.edu.ec

² Doctor en Informática, Profesor contratado en la Escuela Técnica Superior de Ingenieros de la Universidad Politécnica de Madrid. E-mail: jramirez@fi.upm.es



well-known platforms for the creation of dialogue managers.

Keywords: Intelligent tutoring system, Natural Language Process, Dialogue adapted to context.

INTRODUCCIÓN

Hoy en día, los entornos de aprendizaje están cambiando significativamente gracias a la incorporación de diversas tecnologías que están promoviendo un proceso de aprendizaje más personalizado. Así, tenemos por un lado, a los entornos virtuales 2D/3D que aportan una mejor experiencia al momento de explorar el entorno educativo, ya que facilitan el desarrollo de tareas que mejoran la transferencia de conocimientos y habilidades a situaciones reales mediante la contextualización del aprendizaje (Dalgarmo & Lee, 2010); y por otro lado se encuentran los Sistemas Inteligentes de Tutoría (SITs), que son sistemas basados en la aplicación de técnicas de la Inteligencia Artificial que emulan a un profesor humano en ciertas tareas docentes, sobre todo, en aquellas relacionadas con la supervisión de los ejercicios prácticos de los alumnos (Murray, 1999).

Aunque el modelado de los SITs representa, en general, una tarea compleja, se han desarrollado algunos SITs para diferentes propósitos con interfaces conversacionales (Graesser et al 2001); uno de ellos es Autotutor (A. Graesser & et al., 2005), un SIT diseñado para la enseñanza de computación y física newtoniana que dispone de un sistema de diálogo de iniciativa mixta a través de preguntas que invitan a largas explicaciones y razonamientos. Sin embargo, después de estudiar el estado del arte en la materia, podemos afirmar que existen muy pocos agentes conversacionales que se integran con los SITs para el entrenamiento de procedimientos, entre ellos sobresalen Jacob (Evers & Nijholt, 2000) y Paco (Rickel, Lesh, Rich, Sidner, & Gertner, 2001) que mostraremos en los trabajos relacionados, y que a pesar de dar tutoría procedimental, están concebidos para que el estudiante interactúe con el sistema a través de un avatar en un entorno virtual 2D/3D y dialogue con el sistema mediante el uso del lenguaje natural.

Dado que el lenguaje natural es un aspecto fundamental del comportamiento humano, los sistemas de lenguaje natural deben tener en consideración tanto el conocimiento acerca de la estructura del lenguaje (las palabras, el significado de las palabras, las reglas gramaticales, etc.), como el conocimiento del entorno donde se desarrolla el diálogo (Hung, Gonzalez, & DeMara, 2009). Para facilitar esta tarea, se deben utilizar técnicas de Procesamiento del Lenguaje Natural (PLN), que hoy en día ya están proporcionando a los programas de computadora la capacidad de entender el habla humana.

Basado en lo expuesto, este documento tiene como objetivo presentar una propuesta de un SIT para el entrenamiento procedimental en un entorno virtual 2D/3D, capaz de mantener un diálogo en lenguaje natural basado en el contexto. De esta forma, cada alumno será entrenado en el entorno virtual por medio de un diálogo en lenguaje natural tomando en cuenta sus características específicas, su progreso en el desarrollo de la tarea y el ambiente donde realiza la tarea. Por lo tanto, la retroalimentación de tutoría incluirá un diálogo contextualizado o adaptado al contexto.

Para dar soporte al diálogo, utilizaremos un gestor de diálogo que será construido sobre alguna de las plataformas conocidas y actualmente disponibles para la creación de gestores de diálogo. Cabe indicar que estas plataformas han alcanzado un notable grado de madurez, entre otros motivos, por las facilidades que ofrecen para interpretar y procesar el lenguaje natural en diferentes dominios de aplicación; a tal punto que han transformado la forma en que interactuamos con la tecnología informática.

Este trabajo representa un primer paso en el proceso de diseño de un sistema con las características mencionadas anteriormente. El resto de este artículo está estructurado de la siguiente manera:

la sección 2 presenta los trabajos más relacionados con los objetivos de este artículo; la sección 3 se centra en algunas plataformas bien conocidas que facilitan la construcción de gestores de diálogo; la sección 4 describe varias características deseadas del sistema propuesto y presenta la arquitectura del SIT propuesto; la sección 5 detalla tres ejemplos de aplicación de la arquitectura propuesta; y finalmente, la sección 6 presenta las conclusiones del trabajo y menciona algunas líneas de trabajo futuro.

Trabajos relacionados

En esta sección, nos centraremos en presentar los SITs con PLN construidos para ambientes virtuales en 2D/3D. Dentro de este grupo, distinguiremos entre SITs orientados a la enseñanza de conceptos y al entrenamiento de tareas (procedimientos). Sin embargo, antes de presentar este grupo de SITs, consideramos necesario mencionar algunos SITs para entrenamiento de procedimientos que, a pesar de no contar con PLN, son sistemas notables que pueden supervisar el comportamiento del estudiante y proporcionar una valiosa retroalimentación de tutoría. Estos sistemas son:

Steve (Rickel & Johnson, 1999), es un agente animado diseñado para entrenar al personal de la marina para operar y mantener maquinaria compleja en un entorno 3D; incorpora características novedosas, aunque en un estado elemental, que le permite mantener una activa interacción con el alumno y detectar su actitud frente a la tutoría (atento, distraído, falta de motivación); cuenta con módulos que le proporcionan características y habilidades que son independientes del dominio, lo cual no excluye la posibilidad de capacitar a Steve para trabajar en un dominio específico.

Lahystotrain (Los Arcos et al., 2000), es un sistema desarrollado para entrenar a cirujanos en operaciones de laparoscopia e histeroscopia compuesto por una simulación de Realidad Virtual y un ATS (Advanced Training System). La formación se adquiere a través de dos fases, en la primera el estudiante adquiere conocimientos teóricos relacionados con las patologías a tratar y en la segunda se ponen a prueba dentro del simulador las habilidades prácticas adquiridas. Durante la prueba de conoci-

mientos el SIT puede proporcionar dos tipos de explicaciones, proactivas y reactivas: las proactivas están orientadas a aspectos relacionadas con el procedimiento y las reactivas a la corrección de errores.

SafeChild (Gu, Sosnovsky, & Ullrich, 2015), es un sistema diseñado para enseñar a los niños a desenvolverse con seguridad como peatones en situaciones con tráfico. Este sistema proporciona a los niños un entrenamiento guiado y asistido en una amplia gama de escenarios de entrenamiento, y consta de tres componentes principales: un entorno de una ciudad en realidad virtual como área para el entrenamiento, una interfaz flexible para interactuar con el entorno y un SIT que monitorea el comportamiento del alumno detectando errores individuales y deficiencias en sus habilidades cognitivas de alto nivel.

Todos estos sistemas proporcionan una retroalimentación de tutoría que comprende: dar consejos cuando el estudiante comete un error o pide ayuda. En este tipo de sistemas, sin embargo, los mensajes de tutoría se limitan a explicaciones prefabricadas asociadas con errores o pistas. Además, estos sistemas no pueden entender y contestar preguntas abiertas del estudiante, ni evaluar los conocimientos del estudiante a partir de sus comentarios durante la práctica.

En cuanto a los SITs con PLN, en primer lugar, mencionaremos algunos sistemas cuya retroalimentación de tutoría está dirigida a la enseñanza de conceptos. Entre ellos tenemos:

CircSim (Glass, 2001), considerado como uno de los primeros SITs dotado de una cierta capacidad de diálogo en lenguaje natural, es un sistema que enseña el mecanismo baroreceptor reflejo de la presión arterial a estudiantes de primer año de medicina. El diálogo está bajo el control del tutor inteligente, de forma que el sistema hace preguntas y el estudiante contesta con texto libre imitando el estilo socrático de tutoría. Las aportaciones del estudiante son sometidas a un análisis sintáctico apoyado en un extenso catálogo de conceptos erróneos, respuestas erróneas y otros insumos inesperados que presentan oportunidades de tutoría. No obstante, las preguntas sólo admiten respuestas cortas de una o dos palabras, por lo que su rango de respuesta es limitado y no induce a un lenguaje que contenga mayores explicaciones.

Why2-Atlas (VanLehn et al, 2002), facilita la enseñanza de la física cualitativa haciendo que los estudiantes escriban largas explicaciones (ensayos) sobre fenómenos mecánicos simples. Utiliza el análisis sintáctico profundo y el razonamiento abductivo para convertir el ensayo del estudiante en una prueba lógica y descubrir sus conceptos erróneos. Está compuesto por varios módulos, entre los cuales sobresale el entendedor de nivel de sentencias (ENS) que utiliza un preprocesador léxico, un analizador, un módulo de reparación (todos ellos conforman un paquete denominado CARMEL) y un analizador estadístico.

Autotutor (Graesser et al 2005), es un sistema desarrollado para la enseñanza de la computación y la física newtoniana. Mantiene un diálogo basado en turnos con un método de PLN bastante elaborado que permite diálogos con iniciativa mixta para facilitar la iniciación de nuevos temas de discusión. Las conversaciones se realizan de manera escrita, pero posee un sistema de síntesis de voz y una representación gráfica de un tutor con ciertas expresiones faciales y gestos. Este sistema utiliza Análisis Semántico Latente (ASL) para comparar las explicaciones de los estudiantes con un conjunto de expectativas (respuestas ideales) e ideas erróneas (respuestas incorrectas).

Tanto *Why2-Atlas* como *Autotutor* ayudan al estudiante a construir explicaciones correctas a través del diálogo, pero difieren en las técnicas de PLN empleadas para interpretar las expresiones de los estudiantes.

Beetle II (Dzikovska et al 2014), es un sistema desarrollado para enseñar la construcción de un circuito electrónico. Permite entradas en lenguaje natural sin restricciones y la aplicación de diferentes estrategias de diálogo para realizar una retroalimentación adaptativa dinámica. Implementa un sistema de diálogo orientado a tareas que es más dependiente del dominio que el de *Autotutor*. *Beetle* implementa una estrategia pedagógica “Pre-decir-Verificar-Evaluar” para invitar al estudiante a pensar profundamente acerca de los fenómenos que observa y a explicar sus predicciones/errores en una simulación. Para guiar al estudiante, *Beetle II* aplica tácticas de tutoría que pueden usarse en diferentes contextos de resolución de problemas. Ade-

más, *Beetle* utiliza una ontología para representar el conocimiento del dominio.

A continuación, vamos a mencionar los únicos SITs que se han encontrado en la literatura que cuentan con PLN para el entrenamiento de procedimientos, estos son Jacob y Paco.

Jacob (Evers & Nijholt, 2000), es un agente pedagógico que enseña a resolver el problema de las Torres de Hanoi en un entorno virtual 3D. Este sistema integra conocimientos de diferentes disciplinas, tales como: SITs, realidad virtual, tecnología de agentes y PLN. El modelo de tarea y el modelo de instrucción forman parte de la mente de Jacob y actúan como controladores que observan el mundo abstracto y tratan de alcanzar los objetivos específicos de instrucción manipulando el cuerpo de Jacob. El modelo de instrucción encapsula el conocimiento de la tarea y las capacidades de comprensión del lenguaje natural permitiendo que la interacción con el usuario se de a través de acciones relativas a las tareas, en las que el alumno introduce por teclado los enunciados que luego son analizados e interpretados utilizando una representación apropiada del conocimiento del entorno virtual. Seguidamente, el sistema responde a través de declaraciones o realizando acciones que han sido definidas a través de plantillas.

Paco (Rickel, Lesh, Rich, Sidner, & Gertner, 2002), es un agente pedagógico que implementa su estrategia de tutoría por medio de un sistema de diálogo colaborativo basado en Collagen. En este sistema, el diálogo se va generando por medio de reglas de actos de discurso, las cuales, a su vez, se apoyan en el modelo de tareas, el modelo del estudiante y la interacción con el usuario. La formación procedimental está basada en una simulación que implementa el manejo de los motores de turbina de gas que impulsan los barcos. En este contexto, Paco elige repetidamente un escenario, es decir, una tarea a realizar a partir de un estado de simulación particular, para que los estudiantes apliquen con flexibilidad procedimientos bien definidos en una variedad de situaciones de un entorno de trabajo real.

Tanto *Jacob* como *Paco* proporcionan ayuda cuando se solicita retroalimentación positiva, aunque en el caso de Jacob es ocasional. En general, Jacob

y Paco tienen muchas limitaciones con respecto al diálogo en lenguaje natural. Sin embargo, a pesar de no poder entender las preguntas abiertas de los estudiantes, proporcionan respuestas basadas en un cierto nivel de comprensión de lo que requiere el estudiante en cada momento.

Plataformas para crear un gestor de diálogo

El PLN surge con la finalidad de hacer posible la comprensión y el procesamiento de información expresada en lenguaje natural, facilitando el desarrollo de determinadas tareas como el análisis de opiniones, la traducción automática, los sistemas de diálogo interactivos, reconocimiento de patrones, clasificación de entidades, entre otras (Dudhabaware & Madankar, 2014).

Gracias a los servicios en la nube, existen plataformas con frameworks especializados para implementar sistemas de diálogo. Los sistemas de diálogo proporcionan a las personas una vía de interacción con una aplicación informática a través de diálogos escritos (e incluso orales), similares a los que entablan entre sí los seres humanos sobre un asunto concreto (Mallios & Bourbakis, 2016).

Estas plataformas están alcanzando un alto grado de madurez en el tratamiento de las estructuras de diálogo, lo cual ha provocado una significativa proliferación de su uso en diferentes dominios y ha elevado la complejidad de los sistemas desarrollados.

Para lograr su cometido, las plataformas utilizan el poder del aprendizaje automático para resolver el difícil problema de extraer el significado de las entradas del usuario en lenguaje natural. De esta manera, cualquier aplicación cliente que converse con usuarios puede pasar la entrada del usuario a un gestor de diálogo construido con una de estas plataformas y recibir los resultados derivados de una comprensión del lenguaje natural. Esto es posible sin necesidad de que el programador de la aplicación cliente conozca los algoritmos de aprendizaje automático que se están empleando.

Entre las plataformas más populares se encuentran *Api.ai* de Google, *Wit.ai* de Facebook Messenger, *Watson* de IBM, *LUIS* de Microsoft, entre otras. Cada una de estas plataformas tienen APIs

para desarrolladores disponibles en diversos sistemas operativos, dispositivos y lenguajes de programación. Para estructurar el flujo de conversación o diálogo, estas plataformas utilizan algunos elementos comunes que, en su conjunto, facilitan el PLN, entre ellos están: el enunciado, las intenciones y las entidades (Singh, 2017).

El enunciado es la entrada del usuario que la aplicación necesita interpretar, que en algunas ocasiones no están bien formuladas.

Las intenciones son como verbos en una oración, representan las acciones que el usuario desea realizar, es decir el propósito expresado en la entrada del usuario.

Las entidades serían como los sustantivos de una oración, representan una instancia de una clase de objeto que es relevante para la intención de un usuario.

Algo sobresaliente de estas plataformas es que permiten que el diálogo pueda ser ajustado a las diferentes situaciones de un determinado escenario por medio de una estructura denominada contexto, que se utiliza de forma interna y externa para pasar información entre una aplicación cliente y el gestor de diálogo.

Una aplicación desarrollada en cualquiera de estas plataformas, toma un enunciado del usuario y extrae intenciones y entidades que corresponden a actividades relativas a la lógica de la aplicación cliente. De este modo, esta aplicación puede tomar la acción más apropiada basándose en las intenciones del usuario y así generar un diálogo efectivo.

Las plataformas mencionadas, a pesar de estar orientadas a un fin similar, tienen características diferenciadas. Por ello, la elección de la plataforma idónea dependerá de la aplicación que se vaya a construir, teniendo en cuenta básicamente el modelado del flujo de la conversación que es en sí lo que va a definir el comportamiento de la aplicación.

Api.ai actualmente llamada *Dialogflow*, a través de los intentos, crea un enlace entre lo que dice un usuario y qué acción debe tomar el sistema; además brinda servicios de integración con otras

plataformas (Google, 2015). Los contextos son útiles para diferenciar peticiones que podrían tener un significado diferente en relación a las peticiones anteriores. Los intentos y contextos permiten crear máquinas de estado que modelan grandes y complejos flujos de conversación. Asimismo, dispone de una sección de entrenamiento en versión beta.

Wat.ai dispone de entidades predefinidas y considera a las intenciones del usuario como historias (Facebook, 2015). Cada historia representa un ejemplo de una posible conversación. Una historia puede ser vista como un gráfico de las intenciones a las que se pueden agregar ramas que se activan en condiciones tales como la existencia o no de valores de variables específicas. A pesar de ello, las historias están en versión beta y en ocasiones es difícil seguir en ellas el flujo de la conversación.

Watson dispone de muchos servicios cognitivos que favorecen el desarrollo de determinadas tareas. Entre esos sistemas cognitivos cabe destacar el Servicio Conversation que permite la implementación de un sistema de diálogo que es controlado por un espacio de trabajo en el que se definen las intenciones, las entidades y las variables de contexto (IBM, 2016). Los usuarios interactúan con su aplicación a través de su interfaz de usuario proporcionando enunciados. Para cada enunciado proporcionado, la aplicación se conecta al espacio de trabajo y el servicio de conversación interpreta el enunciado del usuario, proporciona una respuesta y dirige el flujo de conversación.

Para *LUIS* los enunciados, los intentos y las entidades son los conceptos claves para modelar el diálogo. *LUIS* es incapaz de reconocer palabras raras, por lo tanto a parte de las entidades puede utilizar una lista de frases para generalizar e identificar nuevos elementos de una categoría (Microsoft, 2016). Además, se pueden agregar funciones al modelo para proporcionar sugerencias sobre cómo reconocer las entradas que se desean etiquetar o clasificar. Estas funciones ayudan a reconocer tanto los intentos como las entidades a través de términos relacionados o patrones para reconocer dichos términos.

Cabe señalar que, la mayoría de las plataformas disponen de entidades predefinidas que dan una

mayor robustez a la aplicación. Sin embargo, tanto *Watson* como *LUIS* proponen un marco de desarrollo un poco más complejo y más potente que las otras plataformas, ya que disponen de servicios que aprenden de forma rápida y por tanto minimizan el esfuerzo requerido para crear los sistemas de diálogo (IBM, 2016; Microsoft, 2016).

En general, independientemente de la plataforma que utilicemos, sostenemos que para poner en marcha el gestor de diálogo es necesario entrenar su espacio de trabajo, es decir: las intenciones necesitan que se les suministren muchos ejemplos de enunciados de usuario; y las entidades requieren que se les refieran todos sus posibles valores junto a los sinónimos que probablemente los usuarios utilizarán.

Características de un Sistema Inteligente de Tutoría para entrenamiento procedimental en un entorno virtual 2D/3D con Procesamiento de Lenguaje Natural

Como podemos ver en la sección de trabajos relacionados, hay una notable escasez de SITs para entrenamiento procedimental en un entorno virtual 2D/3D que cuenten con PLN. Este hecho ha motivado el trabajo que vamos a presentar en este artículo, que consiste en una arquitectura para un SIT dirigido al entrenamiento procedimental que estará equipada con un gestor de diálogo, que será implementado por medio de una de las plataformas explicadas en la sección anterior. Antes de presentar esta arquitectura, vamos a explicar de qué manera se va a utilizar el contexto en la gestión del diálogo con el alumno.

Con el fin de que el estudiante reciba una retroalimentación de tutoría apropiada, es necesario contar no sólo con la frase del estudiante, sino con una noción del contexto en el cual se dice dicha frase. Por lo tanto, el gestor de diálogo asociado al SIT deberá disponer de la suficiente información como para poder proporcionar un diálogo contextualizado. Para proporcionar este tipo de diálogo, el contexto que manejará el gestor deberá contener información relativa a:

- Los conocimientos del estudiante,
- El entorno virtual (con información estática y dinámica), y

- El progreso actual del alumno en la actividad a realizar.

Basándose en esta información, el SIT podrá suministrar una tutoría personalizada a través de un diálogo contextual adaptado a: lo que el estudiante sabe y no sabe; la ubicación de su avatar en el entorno virtual en el momento del diálogo; la fase de la actividad en que se encuentra el estudiante, etc.

Algunos ejemplos de la retroalimentación de tutoría personalizada que proporcionará un diálogo contextualizado son los siguientes:

- Responder a preguntas sobre la ubicación de un objeto en el entorno virtual o cómo llegar a dicho objeto, incluso cuando este objeto está lejos del avatar del estudiante.
- Responder a las preguntas sobre la próxima acción que se debe realizar.
- Recomendar actividades de aprendizaje para cubrir las lagunas de conocimientos demostradas por el estudiante.
- Proporcionar pistas proactivamente para guiar al estudiante con la ejecución de una tarea, si se observa que el estudiante necesita ayuda, aunque no la esté pidiendo.
- Fomentar un diálogo afectivo para mitigar la inactividad de los estudiantes o sus momentos de desánimo durante la práctica.

Arquitectura Propuesta

Para una mejor comprensión de la arquitectura propuesta, en la Figura 1 se muestra dicha arquitectura, la cual, como vamos a justificar a continuación, brindará una tutoría personalizada teniendo como eje central el entorno virtual donde se realiza el entrenamiento.

La arquitectura está integrada por cuatro componentes principales: el entorno de entrenamiento procedimental, el sistema de Comprensión de Lenguaje Natural (CLN), el SIT y el gestor de diálogo. Como se va a explicar a continuación, cada uno de ellos cumple distintas funciones con la finalidad de sustentar la tutoría en lenguaje natural.

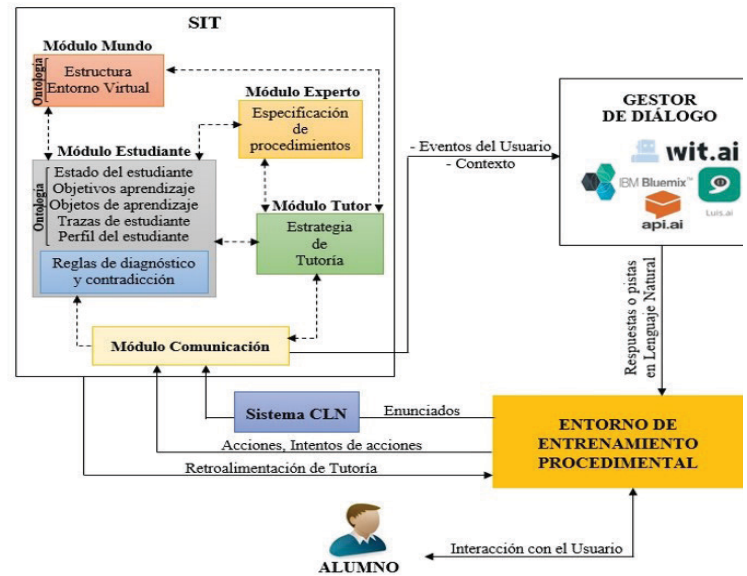


Figura 1. Arquitectura de un SIT con PLN para entrenamiento procedimental

El *entorno de entrenamiento procedimental*, es el escenario que simula al mundo real donde se llevan a cabo las actividades de entrenamiento. Este escenario puede ser diseñado con elementos 3D para dar más realismo a las tareas que el estudiante debe realizar a lo largo de la práctica.

El *sistema de comprensión del lenguaje natural*, se encargará de recibir las frases del usuario, las interpretará y extraerá de ellas las entidades y relaciones que las componen. Luego preprocesará el enunciado del usuario, de manera que luego los módulos del SIT puedan trabajar con la semántica del enunciado. Este sistema, se podría implementar utilizando, el servicio de “Natural Language Understanding” de Watson (IBM, 2015).

El *sistema inteligente de tutoría*, integrará los módulos correspondientes a un SIT clásico, es decir, el módulo del estudiante, el módulo de tutoría, el módulo experto y el módulo de comunicación. Adicionalmente, se agregará al SIT un módulo del mundo, para representar las características del entorno virtual de formación, a fin de posibilitar un efectivo proceso de tutoría, y contendrá información sobre los aspectos constituyentes del entorno relacionados con los avatares y los objetos 2D/3D del entorno virtual. Además, este módulo representará la estructura física de los escenarios y su contenido para que el sistema pueda responder a una pregunta sobre la situación de un objeto o

cómo ir de un lugar a otro.

Cabe señalar que, la información expuesta, no solo será necesaria para la tutoría, sino también para validar las acciones realizadas por el estudiante. Otros dos módulos que tendrán un papel clave en nuestra propuesta serán el módulo del estudiante y el módulo experto:

El *módulo del estudiante* contendrá información relacionada con las acciones del estudiante; sus movimientos a través del entorno virtual; el tiempo que necesitó para realizar una acción o un grupo de acciones; las preguntas que hizo; las sugerencias que recibió por parte del tutor, etc. A partir de esta información el mismo módulo inferirá, con un cierto nivel de fiabilidad, los conocimientos del alumno, lo cual, a su vez, resultará útil para decidir la mejor estrategia de tutoría. Para lograr este fin adoptaremos el modelo del estudiante propuesto en (Clemente, Ramírez, & de Antonio, 2011).

El *módulo experto* especificará el conocimiento que debe ser enseñado al estudiante. Como estamos abordando el entrenamiento procedimental, el modelo contendrá una descripción del procedimiento a ser aprendido.

Dentro del SIT, el contexto del diálogo se definirá por medio de una ontología y se rellenará con información proveniente del modelo del estudiante (estado de los conocimientos, progreso de la actividad, trayectoria del estudiante en el entorno virtual), el modelo experto (el plan correcto, la siguiente acción correcta) y el modelo del mundo (estructura del mundo virtual, posición del avatar del estudiante, descripciones de los objetos). Luego, antes de pasar el contexto al gestor de diálogo, la representación ontológica del contexto se traducirá a otra representación comprensible para el gestor de diálogo.

El *gestor de diálogo* contendrá la definición de la estructura del diálogo, las intenciones y las entidades específicamente destinadas al entorno de formación, así como el sistema de ejecución de la plataforma elegida. Este componente será responsable de la comunicación con el usuario teniendo en cuenta la información contextual proporcionada por los modelos que integran el SIT.

El usuario puede interactuar con el entorno de formación procedimental generando eventos tales como intentos de acciones, preguntas, etc., que serán proporcionados al SIT a lo largo de la práctica.

El siguiente pseudocódigo describe cómo el gestor de diálogo puede ser integrado en el proceso de tutoría aplicado por el SIT en un entorno de entrenamiento procedimental.

- 1: Tutor da un mensaje de bienvenida con posible pista
- 2: Espera a que el alumno genere un evento (intento de acción o frase) o transcurra el tiempo de espera
- 3: Si el estudiante intenta realizar alguna acción antes del tiempo de espera
- 4: Si la acción es correcta
- 5: Se pide al gestor de diálogo que dé OK
- 6: Se deja que el estudiante realice la acción
- 7: Caso contrario
- 8: Se solicita al gestor de diálogo que envíe un mensaje de error, si es pedagógicamente adecuado.
- 9: Se bloquea o impide la ejecución de la acción, si es pedagógicamente adecuado.
- 10: Fin de Si
- 11: Si el estudiante parece necesitar una pista según su modelo de estudiante
- 12: Se envía al gestor de diálogo el evento y contexto del usuario
- 13: Se da al estudiante la pista (la respuesta del gestor de diálogo) ajustada a su estado actual de conocimientos.
- 14: Fin de Si
- 15: Fin de Si
- 16: Si el estudiante dice algo antes del tiempo de espera
- 17: Se proporciona la frase y el contexto al gestor de diálogo
- 18: Se muestra al estudiante la respuesta del gestor de diálogo
- 19: Fin de Si
- 20: Si el estudiante no intenta ninguna acción ni dice nada, y expira el plazo de espera
- 21: Se envía al gestor de diálogo el evento (excesiva inactividad) y el contexto
- 22: Se da una pista ajustada a su estado actual de conocimientos
- 23: Fin de Si
- 24: Actualiza el módulo del estudiante y el módulo del mundo.
- 25: Retorna al paso 2

Ejemplos de Aplicación de la Arquitectura Propuesta

Para una mejor comprensión del proceso de tutoría explicado en el apartado anterior y de cómo intervienen en este proceso los módulos para la generación del diálogo en lenguaje natural, presentaremos tres ejemplos basados en una práctica “Preparación del Medio de Cultivo” que se realiza en un Laboratorio Virtual de Biotecnología³. Este laboratorio fue desarrollado sobre la plataforma de mundos virtuales OpenSimulator, como parte de una tesis de máster (Riofrío-Luzcando, Ramírez, y Berrocal-Lobo, 2012). Para realizar la práctica, el alumno controla un avatar y cuenta con la ayuda de un tutor automático, que le va dando indicaciones sobre las acciones a realizar en cada momento y le muestra mensajes de error, cuando se equivoca. Estas indicaciones y mensajes de error consisten en mensajes previamente configurados y asociados a cada una de las acciones que se deben realizar en la práctica.

Este Laboratorio Virtual de Biotecnología cuenta con cinco salas: hall, sala principal, sala de autoclaves, sala de fitotrón, sala de cabinas de flujo de laminar. La mayor parte de las tareas se realizan en la sala principal donde se encuentran la mayoría de los químicos y el instrumental tal como se observa en la Figura 2.

La versión actual del tutor automático no implementa la arquitectura propuesta en el apartado anterior. No obstante, los ejemplos que se van a explicar a continuación describen el comportamiento del SIT como si éste realmente estuviera dotado de la arquitectura propuesta e implementara el proceso de tutoría ya mencionado.

Para una mejor definición de los ejemplos que veremos a continuación se ha utilizado como base la Figura 3 de la arquitectura propuesta, de tal forma que se han dibujado círculos numerados sobre ella para especificar la secuencia de pasos que ejecutan los módulos del SIT en cada escenario.

En este ejemplo presentamos un caso relacionado con la línea 3 del pseudocódigo, es decir, cuando el alumno *intenta realizar una acción y ésta es correcta*.

3. En http://youtu.be/mAFREZ5_iak se puede encontrar un video de este laboratorio virtual



Figura 2. Sala principal del Laboratorio Virtual de Biotecnología

Ejemplo No. 1: El alumno intenta coger sacarosa para agregarlo a la mezcla y la acción es correcta de acuerdo al plan. La secuencia es la siguiente:

1. El alumno intenta coger un elemento químico: “sacarosa”.
2. El entorno de entrenamiento procedimental retransmite el intento de acción (evento) al módulo comunicación.
3. El módulo comunicación envía el evento al módulo estudiante.
 - El módulo estudiante registra en su ontología que el estudiante intenta realizar una acción.

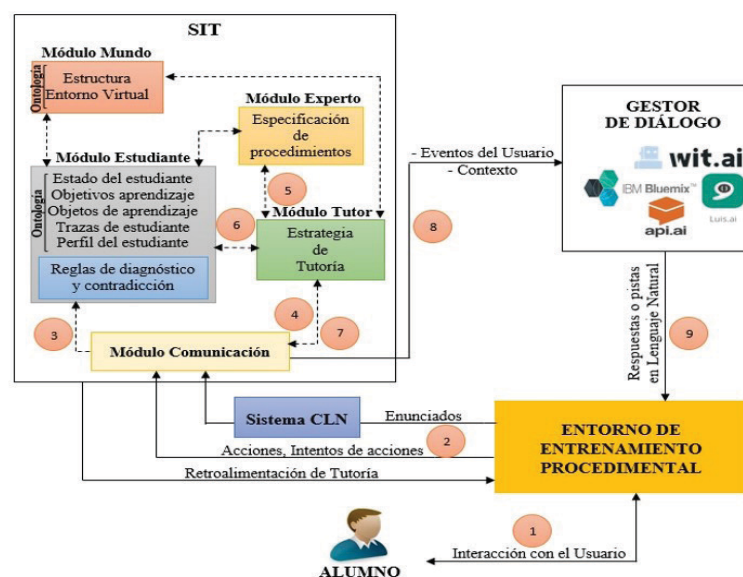


Figura 3. El alumno intenta agregar un químico a la mezcla y la acción es correcta

- El objetivo de aprendizaje que se evaluará es: “El alumno debe incorporar sacarosa a la mezcla”.
- El módulo comunicación envía el evento al módulo tutor.
 - El módulo tutor le consulta al módulo experto si el evento es correcto y éste devuelve: “Correcta, OK”.
 - El módulo tutor le dice al módulo estudiante: “La acción coger sacarosa es correcta”
 - El módulo estudiante infiere mediante una regla de diagnóstico que el estudiante ha adquirido dicho objetivo de aprendizaje o refuerza su creencia en dicha adquisición, ya que acaba de obtener una nueva evidencia de ello.
 - El módulo tutor le dice al módulo comunicación “La acción coger sacarosa es correcta”.
 - El módulo comunicación le indica al gestor de diálogo que la acción es correcta.
 - El gestor de diálogo envía al entorno de entrenamiento procedimental el mensaje “Acción Correcta”.

En el siguiente ejemplo mostraremos un caso relativo a la línea 20 del pseudocódigo, es decir, cuando el alumno *ha permanecido inactivo demasiado tiempo*.

Ejemplo No. 2: El alumno tiene en la mano un vaso lleno con agua destilada, pero no sabe a dónde llevarlo y se queda parado. De acuerdo al plan, la siguiente acción correcta es *llevar ese vaso a un agitador y soltarlo allí*. Esta es la secuencia

- El alumno permanece inactivo durante un tiempo lo suficientemente grande como para que el SIT infiera que no sabe qué hacer a continuación.
- El entorno de entrenamiento procedimental retransmite la inactividad del alumno al módulo comunicación.
- El módulo comunicación notifica la excesiva inactividad del alumno al módulo estudiante.
 - El módulo estudiante registra en su ontología la excesiva inactividad del alumno.

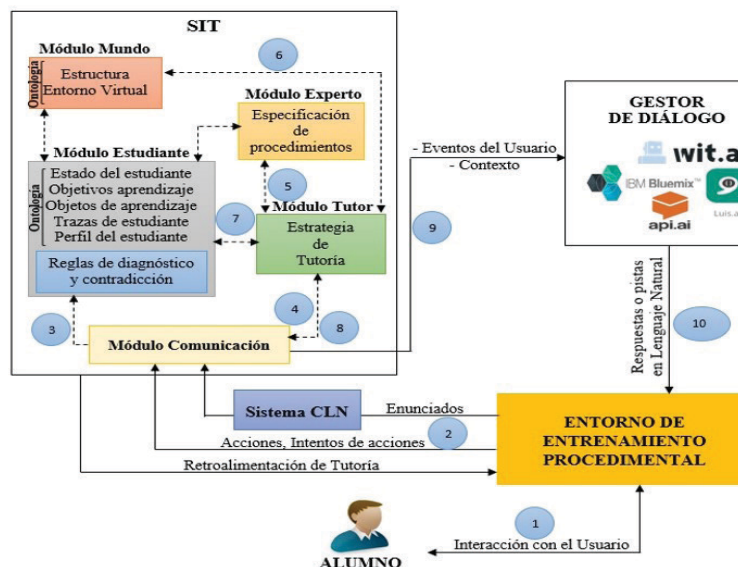


Figura 4. El alumno desconoce la siguiente acción que debe realizar

- El módulo estudiante a través de las reglas de diagnóstico infiere que el alumno no conoce la siguiente acción.
 - El objetivo de aprendizaje que se evaluará es: “El alumno debe llevar el vaso lleno de agua destilada al agitador”.
 - Por tanto, el estado de conocimientos del estudiante contendrá: “El alumno no ha adquirido ese objetivo con un cierto grado de certeza”.
- El módulo comunicación notifica la excesiva inactividad del alumno al módulo tutor.
 - El módulo tutor le consulta al módulo experto cuál es la siguiente acción y éste devuelve: “Llevar el vaso de agua destilada al agitador”
 - El módulo tutor una vez que ha averiguado cuál es la siguiente acción, le pregunta al módulo mundo ¿Dónde está el agitador?, ¿Dónde está el alumno? y éste devuelve:
 - “El agitador está en la sala principal”
 - “El estudiante está en la sala de autoclave”
 - El módulo tutor le pregunta al módulo estudiante: ¿Cuál fue la última acción que realizó el alumno? ¿Cuál fue el rendimiento global del estudiante hasta el momento en la práctica? El módulo estudiante le dirá al tutor:
 - “El alumno llenó un vaso con agua destilada”

“El rendimiento fue flojo (ha cometido muchos errores)”

8. El módulo tutor le envía al módulo comunicación la información recabada del módulo mundo y módulo estudiante con la que se elaborará la información del contexto. Además, le indica que el alumno va a necesitar una pista concreta, dado su rendimiento hasta el momento en la práctica.

9. El módulo comunicación envía al gestor de diálogo el evento “excesiva inactividad del alumno” y el contexto que contiene la siguiente información:

SIG_ACCION_CORRECTA: “Llevar el vaso de agua destilada al agitador”

“El agitador está en la sala principal”

“El estudiante está en la sala de autoclave”

“El alumno llenó un vaso con agua destilada”

Nivel de Pista: Concreto

10. El gestor de diálogo envía al entorno de entrenamiento procedimental la pista de acuerdo con el nivel que decidió el módulo tutor.

Concreto: “Debes salir de la sala de autoclave y llevar el vaso al agitador que está en la sala principal”

General: “Debes llevar el vaso al agitador que está en la sala principal”

En el siguiente ejemplo presentamos un caso que guarda relación con la línea 16 del pseudocódigo, es decir, cuando el alumno dice algo antes del tiempo de espera.

Ejemplo No. 3: El alumno pregunta por un nuevo químico que necesita para la mezcla. La secuencia es como sigue:

1. El alumno pregunta ¿Dónde está el agar?
2. El Entorno de entrenamiento procedimental retransmite la pregunta al Sistema CLN.
3. El Sistema CLN interpreta la pregunta y la descompone en entidades y relaciones y las envía al módulo de comunicación del SIT
4. El módulo comunicación envía la pregunta descompuesta en entidades y relaciones al módulo estudiante.

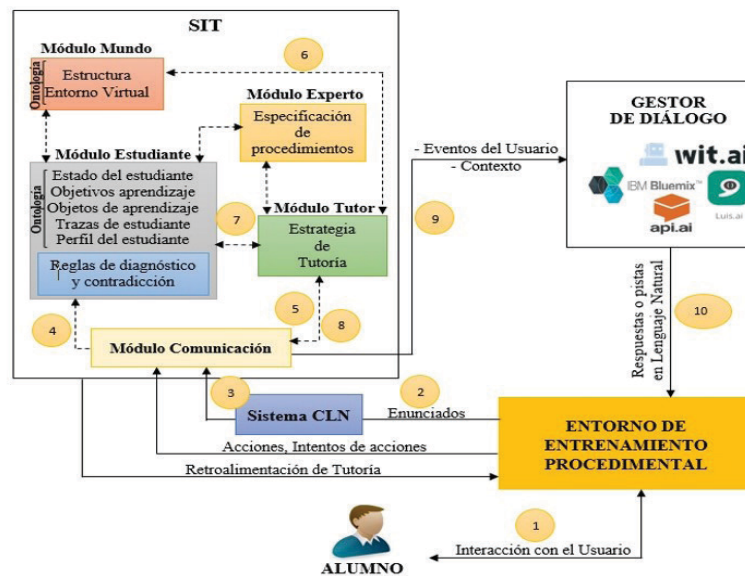


Figura 5. El alumno dice algo antes del tiempo de espera

- El módulo estudiante a través de las reglas de diagnóstico deduce que el alumno no sabe dónde está el agar y actualiza el estado del estudiante.
 - El objetivo de aprendizaje que se evaluará es: “El alumno sabe dónde está el agar”.
 - Por tanto, el estado de conocimientos del estudiante contendrá: “El alumno no ha adquirido ese objetivo con un cierto grado de certeza”.
5. El módulo comunicación envía la pregunta al módulo tutor ¿Dónde está el agar?.
 6. El módulo tutor una vez que le llega la pregunta del alumno, le pregunta al módulo mundo ¿Dónde está el agar? ¿Dónde está el estudiante? y éste devuelve:
 - El agar está dentro de la vitrina” y “La vitrina está en la sala principal”
 - “El estudiante está en la sala principal”
 7. El módulo tutor le pregunta al módulo estudiante: ¿Sabe el alumno dónde está la vitrina? ¿Cuál fue el rendimiento global del estudiante hasta el momento en la práctica?

Como el alumno ya ha tomado anteriormente otro químico de la vitrina y esto quedó registrado en la ontología del estudiante, el módulo estudiante le contestará al módulo tutor:

“Sí, el alumno ya sabe dónde está la vitrina”.

“El rendimiento fue bueno (cometió pocos errores)”

8. El módulo tutor envía al módulo comunicación la información recabada del módulo mundo y módulo estudiante con la que se elaborará la información del contexto.

9. El módulo comunicación envía al gestor de diálogo la pregunta del alumno ¿Dónde está el agar? y el contexto con la siguiente información:

“El estudiante está en la sala principal”

“El agar está en la vitrina”

“La vitrina está en el la sala principal”

“El estudiante sabe dónde está la vitrina”

Nivel de Pista: General

10. El gestor de diálogo envía al entorno de entrenamiento procedimental la pista de acuerdo al nivel que decidió el módulo tutor.

General: “El agar está en el mismo sitio que el otro químico que añadiste antes”

Concreto: “El agar está en la vitrina de la sala principal”

DISCUSIÓN Y CONCLUSIONES

Hemos presentado una propuesta de un SIT para el entrenamiento procedimental con PLN en el que se muestran algunas de sus características y la arquitectura del sistema que abarcaría el SIT junto al gestor de diálogo. Para dar soporte al diálogo hemos propuesto utilizar un gestor de diálogo que se podría construir sobre la plataforma de Watson – IBM y su Servicio Conversation. El motivo de esta elección es que esta plataforma es la más robusta al momento de integrar las variables de contexto con los componentes de la estructura del diálogo, así como por la posibilidad de utilizar el Servicio de Natural Language Understanding que proporcionará el sistema de comprensión en lenguaje natural.

Para detallar la solución propuesta, se ha presentado un pseudocódigo que describe cómo se integraría el gestor de diálogo en el proceso de tutoría

del SIT, y ejemplos para tres casos diferentes, en los que se muestra la interacción entre los módulos de la arquitectura.

Este artículo describe los primeros resultados a los que se ha llegado en una tesis doctoral todavía en curso. Además de lo que hemos presentado, en el futuro planeamos refinar la arquitectura presentada: profundizando en el concepto de “contexto”, y diseñando una estructura de diálogo que aproveche la información de este contexto para proporcionar mensajes en lenguaje natural adaptados a cada estudiante y a su situación en el entorno virtual.

AGRADECIMIENTO

Agradece el apoyo económico brindado por parte de la Universidad Estatal del Sur de Manabí – Ecuador.

REFERENCIAS BIBLIOGRÁFICAS

- Clemente, J., Ramírez, J., & de Antonio, A. (2011). A proposal for student modeling based on ontologies and diagnosis rules. *Expert Systems with Applications*, 38(7), 8066–8078. Recuperado de: <https://doi.org/10.1016/j.eswa.2010.12.146>
- Dalgarmo, B., & Lee, M. J. W. (2010). What are the learning affordances of 3-D virtual environments? *British Journal of Educational Technology*, 41, 10–32. Recuperado de: <https://doi.org/10.1111/j.1467-8535.2009.01038.x>
- Dudhabaware, R., & Madankar, M. (2014). Review on Natural Language Processing Tasks for Text Documents. In *IEEE International Conference on Computational Intelligence and Computing Research* (pp. 1–5). Coimbatore. Recuperado de: <https://doi.org/10.1109/IC-CIC.2014.7238427>
- Dzikovska, M., Steinhauer, N., Farrow, E., Moore, J., & Campbell, G. (2014). BEETLE II: Deep Natural Language Understanding and Automatic Feedback Generation for Intelligent Tutoring in Basic Electricity and Electronics. *International Journal of Artificial Intelligence in Education*, 24(3), 284–332. Recuperado de: <https://doi.org/10.1007/s40593-014-0017-9>
- Evers, M., & Nijholt, A. (2000). Jacob - An Animated Instruction Agent in Virtual Reality. In T. Tan, Y. Shi, & W. Gao (Eds.), *Advances in Multimodal Interfaces --- ICMI*

- 2000: Third International Conference Beijing, China, October 14--16, 2000 Proceedings (pp. 526–533). Springer Berlin Heidelberg.
- Facebook, M. (2015). Wit.ai. Recuperado de: <https://wit.ai/docs/recipes>
- Glass, M. (2001). Processing Language Input in the CIRC-SIM-Tutor Intelligent Tutoring System. In J. D. M. et Al (Ed.), *Artificial Intelligence in Education* (pp. 210–221). San Antonio: IOS Press.
- Google. (2015). Dialogflow. Recuperado de: <https://dialogflow.com/docs/intents>
- Graesser, A. C., Chipman, P., Haynes, B., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. In *IEEE Transactions in Education* (pp. 612–618).
- Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001). Intelligent Tutoring Systems with Conversational Dialogue. *AI Mag.*, 22(4), 39–51. Recuperado de: <http://dl.acm.org/citation.cfm?id=567363.567366>
- Gu, Y., Sosnovsky, S., & Ullrich, C. (2015). SafeChild: An Intelligent Virtual Reality Environment for Training Pedestrian Safety Skills. In G. Conole, T. Klobu?ar, C. Rensing, J. Konert, & É. Lavoué (Eds.), *Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning, EC-TEL 2015, Toledo, Spain, September 15-18, 2015, Proceedings* (pp. 141–154). Cham: Springer International Publishing. Recuperado de: https://doi.org/10.1007/978-3-319-24258-3_11
- Hung, V., Gonzalez, A., & DeMara, R. (2009). Towards a Context-Based Dialog Management Layer for Expert Systems. In *International Conference on Information, Process, and Knowledge Management* (pp. 60–65). IEEE. Recuperado de: <https://doi.org/10.1109/eKNOW.2009.10>
- IBM. (2015). Natural Language Understanding. Recuperado de: <https://www.ibm.com/watson/services/natural-language-understanding/>
- IBM. (2016). Watson conversation. Recuperado de: <https://console.bluemix.net/docs/services/conversation/getting-started.html#gettingstarted>
- Los Arcos, J., Muller, W., Fuente, O., Orúe, L., Arroyo, E., Leaznibarrutia, I., & Santander, J. (2000). LAHYSTO-TRAIN: Integration of Virtual Environments and ITS for Surgery Training. In G. Gauthier, C. Frasson, K. VanLehn (Eds.): *ITS 2000, LNCS 1839* (pp. 43–52). Springer-Verlag. Recuperado de: https://doi.org/10.1007/3-540-45108-0_8
- Mallios, S., & Bourbakis, N. (2016). A survey on human machine dialogue systems. In *7th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1–7). Recuperado de: <https://doi.org/10.1109/IISA.2016.7785371>
- Microsoft. (2016). Learn about Language Understanding Intelligent Service (LUIS). Recuperado de: <https://docs.microsoft.com/en-us/azure/cognitive-services/LUIS/Home>
- Murray, T. (1999). Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art. *Internacional Journal of Artificial Intelligence in Education (IJAIED)*, 10, 98–129.
- Rickel, J., & Johnson, W. L. (1999). Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence*, 13, 343–382.
- Rickel, J., Lesh, N., Rich, C., Sidner, C., & Gertner, A. (2002). Collaborative Discourse Theory as a Foundation for Tutorial Dialogue. In *Proceedings of Sixth International Conference on Intelligent Tutoring System* (pp. 524–551). Springer-Verlang. Recuperado de: https://doi.org/10.1007/3-540-47987-2_56
- Riofrío-Luzcando, D., Ramírez, J., y Berrocal-Lobo, M. (2012). *Diseño e Implementación de un Laboratorio Virtual de Biotecnología*. Tesis de Máster, ETS de Ingenieros Informáticos Universidad Politécnica de Madrid, Madrid, España.
- Singh, B. (2017). Chat Bots — Designing Intents and Entities for your NLP Models. Recuperado de: <https://blogs.msdn.microsoft.com/brijrajsingh/2017/01/29/chat-bots?-?designing-intents-and-entities-for-your-nlp-models/>
- VanLehn, K., Jordan, P. W., Rosé, C. P., Bhembe, D., Böttner, M., Gaydos, A., ... Srivastava, R. (2002). The Architecture of Why2-Atlas: A Coach for Qualitative Physics Essay Writing. In S. A. Cerri, G. Gouardères, & F. Paragaçu (Eds.), *Intelligent Tutoring Systems: 6th International Conference, ITS 2002 Biarritz, France and San Sebastian, Spain, June 2--7, 2002 Proceedings* (pp. 158–167). Springer Berlin Heidelberg.