

# Investigación

## Agente Virtual Inteligente Aplicado a un Entorno Educativo

Celia G. Róspide y Cristina Puente

Revista de Investigación



Volumen II, Número 2, pp. 195-208, ISSN 2174-0410  
Recepción: 20 Sep'12; Aceptación: 26 Sep'12

1 de octubre de 2012

### Resumen

El presente artículo presenta el desarrollo de un agente virtual inteligente o chatbot cuyo dominio de conocimiento se corresponde con el temario de una asignatura académica, con el objetivo de lograr su interacción con el alumno y resolver sus dudas sobre la materia, suponiendo, así, un refuerzo en su proceso de aprendizaje.

**Palabras Clave:** Agente Virtual Educativo Inteligente, Chatbot, e-Learning, Respuesta Flexible.

### Abstract

The current paper introduces the development of a virtual agent or chatbot, whose knowledge domain belongs to the contents of an academic subject. The goal of the process is to achieve interaction between the application and the student, and, therefore, to help him in his learning process.

**Keywords:** Intelligent Virtual Agent, Chatbot, e-Learning, Flexible Answer.

## 1. Introducción

Desde siempre, y más en los últimos tiempos, vivimos en un mundo en el cual debemos esforzarnos cada día por lograr una sociedad más capacitada y preparada. Para ello, la educación se convierte en un aspecto de vital importancia, debiendo entenderse como un bien compartido.

Respecto a este punto, cabe recuperar las palabras de Velásquez Córdoba [1] cuando, citando a Víctor Guédez [2] subrayaba un modelo de educación fundamentado en la *motivación a ser más*, apelando para ello a la colaboración de los demás. Este prototipo se apoya en una idea de educación que se aleja del sistema individual de aprendizaje, basándose en una trama de intercambio de conocimientos entre personas en desarrollo constante.

La implantación de un nuevo sistema educativo, con una reducción de clases presenciales y un incremento de trabajos prácticos del alumno, propicia que se creen nuevas herramientas para solventar de manera automática las dudas que pudieran aparecer referentes a una materia de estudio. En este orden de magnitud, se materializan numerosas posibilidades de potenciación de la educación a través de la participación activa, entre las que se encuentra la ofrecida por las

nuevas tendencias tecnológicas y, en concreto, la que puede ser brindada por un agente virtual o chatbot.

Un chatbot puede definirse como un programa diseñado para mantener una conversación con una persona. Recientemente, este tipo de aplicaciones han visto aumentada su capacidad tras haber sido dotadas de Inteligencia Artificial, la cual aproxima su comportamiento al generalmente ejercido por el ser humano, debiendo éste exhibir, según Kasabov [3], las siguientes características:

- Aprender nuevos problemas e incrementar normas de solución.
- Capacidad de adaptación en línea y en tiempo real.
- Ser capaz de analizar condiciones en términos de comportamiento, el error y el éxito.
- Aprender y mejorar a través de la interacción con el medio ambiente (realización).
- Aprender rápidamente del estudio de grandes cantidades de datos.
- Basarse en una memoria de almacenamiento masivo, con la posterior recuperación de dicha capacidad.

Dicha inteligencia distingue varios tipos de procesos válidos para la obtención de resultados racionales, entre los que destacan dos de ellos por su relación con la base sobre la que se cimientan los chatbots: la ejecución de una respuesta predeterminada para cada entrada, y la búsqueda del estado requerido dentro del conjunto de estados producidos en función de la entrada recibida.

Generalmente, cuando se habla de agente virtual suele aludirse a una entidad de software que consta de un propósito específico. Los agentes tienen sus propias ideas sobre las tareas a ejecutar. Con objetivos concretos se distinguen los agentes de las aplicaciones multifunción, que son típicamente más pequeñas, según explicó Hayes-Roth [4].

En la actualidad, son diversos los usos de que se provee a estos agentes, siendo los siguientes, quizá, los más representativos: la exploración de datos en la red, la atención de clientes en sitios web comerciales (compras, ventas y comercio electrónico), la consulta sobre productos, la explicación de manuales de instrucciones, la administración de canales de IRC<sup>1</sup> actuando como moderadores (con capacidad, incluso, para expulsar y banear usuarios), o el mero entretenimiento.

Con todo, su empleo en el ámbito de la educación no se encuentra aún demasiado explotado. En la última década han surgido nuevos tipos de sistemas, basados en robots virtuales, que han demostrado ser de gran apoyo en la capacitación de alumnos y en el trabajo de los profesores. Es el caso de los Sistemas Tutores Inteligentes (STI), los cuales buscan modelar el sistema de forma que pueda adaptarse al comportamiento del estudiante, identificando la forma en que éste resuelve un problema, a fin de poder brindarle ayudas cognitivas cuando lo requiera.

Un tutor inteligente es, según Vanlehn [5]:

*Un sistema de software que utiliza técnicas de Inteligencia Artificial (IA) para representar el conocimiento e interactúa con los estudiantes para enseñárselo.*

Por otro lado, una nueva definición [6] describe los STI como

*Sistemas que modelan la enseñanza, el aprendizaje, la comunicación y el dominio del conocimiento del especialista y el entendimiento del estudiante sobre ese dominio.*

<sup>1</sup> Se le llama IRC (Internet Relay Chat) a una red de comunicación en tiempo real, similar al Messenger, en la que se puede hablar con un grupo de usuarios al mismo tiempo.

Sin embargo, estos agentes no están orientados a la resolución de dudas académicas sobre una asignatura concreta.

## 2. Características y objetivos fundamentales

El sistema inteligente desarrollado pretende facilitar el aprendizaje de la materia por parte del alumno, basándose en la interacción usuario-máquina, por medio de la cual el estudiante formulará sus dudas concretas acerca de la materia. Dicha comunicación, que podrá llevarse a cabo con total flexibilidad, proporcionará de manera orientativa la información que el alumno desea obtener. Asimismo, se tendrá constancia de la evolución del mismo a lo largo del curso, de manera que la información proporcionada podrá verse ampliada, de forma automática, con aquellos contenidos en los que el alumno necesite hacer especial incisión.

Gracias a la solidez de su arquitectura, el sistema permite la ejecución simultánea de sus funcionalidades por parte de varios usuarios, los cuales dispondrán de una información actualizada en todo momento. Por tanto, el agente creado puede definirse como un sistema interactivo, capaz de proveer a su interlocutor de una respuesta en tiempo real.

Cabe mencionar que el idioma nativo de la base de conocimientos del agente es el inglés. Este hecho ayuda notablemente a agilizar la programación y el análisis de las cadenas de entrada introducidas por el alumno, puesto que la estructura de formulación de preguntas en esta lengua es mucho más parametrizable y estricta que la del castellano.

De hecho, el castellano es un idioma cuya gramática presenta una dificultad latente, destacando las formas verbales, tiempos y modos. La estructura gramatical que compone las frases admite una amplia variedad de formulación. Además, el vocabulario permite una gran cantidad de giros, sinónimos y variantes. Por esta razón, se ha escogido el inglés como lengua materna, debiendo, consecuentemente, establecerse la comunicación con el agente en esta lengua.

Además, una nueva ventaja de su empleo es el aumento de la visión internacional del desarrollo, así como su integración con numerosas herramientas semánticas, la mayoría de ellas construidas en base al inglés. Con ello, la mejora y posible adaptación de este sistema a otros entornos, cobra una probabilidad de materialización mayor.

Por otra parte, y para garantizar su correcto ejercicio, el sistema debe ser capaz de cumplir dos objetivos fundamentales:

- Comprender las preguntas formuladas por el alumno, distinguiendo entre las que atañen a la materia de la asignatura y las que presentan un carácter general.
- Elaborar una respuesta en consonancia con lo cuestionado, pudiendo resolver las dudas del alumno sobre el contenido de la disciplina.

Para el cumplimiento del primero de ellos es necesario tener en cuenta la siguiente directriz, revelada por el estudio del estado del arte: el mejor método de comprensión empleado por un chatbot es el análisis de los patrones de diálogo empleados por su interlocutor humano. Es decir, el sistema analizará de forma individual la entrada introducida por el usuario, detectando en ella patrones considerados clave para la posterior elaboración de la respuesta. Este procedimiento, otorgará una total flexibilidad a la estructura que debe presentar la pregunta del alumno.

Una vez determinados los patrones empleados por el usuario, se comprobará la correspondencia de los mismos con el contenido de una base de datos, conformando la conveniente respuesta a la pregunta formulada.

La consecución de ambos objetivos conlleva la facilitación del proceso de aprendizaje del alumno, pudiéndose éste realizar a través de un mecanismo flexible, dinámico e interactivo, el cual, se espera, atraerá al estudiante y fomentará su implicación en la materia.

## 2.1. Integración

El sistema estará integrado en una plataforma de e-Learning, por medio de la cual el alumno podrá iniciar el proceso de comunicación con el chatbot.

El término e-Learning engloba procesos de enseñanza-aprendizaje por medio de Internet. Una plataforma contenedora de esa modalidad, como puede ser *Moodle*, ofrece ambientes de aprendizaje diseñados e integrados en un entorno privado dotado de las herramientas necesarias para el aprendizaje y el mejor seguimiento de la evolución del alumno.

## 2.2. Aspectos distintivos

El lenguaje AIML, extensión del lenguaje XML, fue diseñado específicamente para ayudar en la creación de la primera entidad chatbot informática de lenguaje artificial online, A.L.I.C.E. Por tanto, se trata de un lenguaje especializado en la creación de agentes software con lenguaje natural.

Sin embargo, en este sistema no se utiliza AIML para interpretar el lenguaje por los siguientes motivos:

- No existe una plataforma estandarizada que permita el funcionamiento del código AIML. Para utilizarlo, es imprescindible el empleo de HTML, Program D ó E, y acceder al sitio web pandorabots.
- AIML es un lenguaje muy rígido. Realiza matcheos exactos, es decir, el usuario debe ingresar exactamente la frase que el bot tiene almacenada en su base de conocimientos, de manera que para comprender al usuario el agente debería tener indicadas todas formas y expresiones posibles que el usuario podría utilizar.
- Elaborar una respuesta en consonancia con lo cuestionado, pudiendo resolver las dudas del alumno sobre el contenido de la disciplina.
- AIML no soporta la aceptación de variantes adicionales de entrada, como pueden ser:
  - Palabras de diez letras.
  - Sólo dígitos.
  - Sólo signos.
- AIML no permite operar con el valor ingresado en la entrada por el usuario.

## 3. Arquitectura

La arquitectura de todo proyecto informático puede entenderse como la disposición conjunta y ordenada de elementos software y hardware para cumplir con una determinada función. Evidentemente, la combinación de arquitecturas muy distintas e inconsistentes deriva en la obtención de un proyecto ingobernable, tanto o más cuanto mayor sea la envergadura del mismo.

La arquitectura externa del sistema refleja todos los elementos, componentes o sistemas que intervienen en la implantación del sistema global. Se instaura la arquitectura Cliente-Servidor,

la cual responde a un modelo de aplicación en el que las tareas se reparten entre el proveedor de recursos, el servidor, y el demandante, el cliente. Este último se erige como la parte activa de la relación, estando diseñado con la asunción de recibir respuesta por cada petición que realice. El servidor, fracción pasiva, se encarga de procesar dichas peticiones y enviar una respuesta a las mismas, orientando siempre su actuación hacia la maximización de la eficiencia.

La arquitectura interna describe el alcance del proyecto dentro del ámbito de las unidades funcionales programadas para la ejecución de las diferentes funcionalidades. Se ha escogido la aplicación del patrón Modelo –Vista – Controlador (MVC), gracias al cual se separa la lógica de negocio del interfaz del usuario, facilitando la evolución por separado de ambos aspectos, e incrementando la reutilización y la flexibilidad de la aplicación. La lógica del interfaz va a cambiar con más frecuencia que la información almacenada en la base de datos (lógica de negocio), de manera que este diseño impedirá que, si hay necesidad de modificar el interfaz, haya que cambiar trabajosamente los datos asociados.

Además, es imprescindible la interacción con una base de datos, soporte de almacenamiento del sistema. Entre sus registros, contará con la información relacionada con el acceso de usuarios al sistema, el dominio de conocimiento del chatbot (apuntes de la asignatura) y el procedimiento para reconocer los patrones introducidos por el usuario en la entrada.

En base a las arquitecturas externa e interna definidas, se expone una imagen que recoge el tránsito de información y el lugar físico (cliente o servidor) en el que se desarrollan cada una de las operaciones a llevar a cabo.



Figura 1. Arquitectura específica del sistema

Como se puede apreciar en la figura, el cliente web consta de dos partes fundamentalmente, el interfaz y la sección de validación de entrada. Para implementar el estilo del interfaz se han empleado HTML5, CSS y J-Query.

El módulo de validación de entrada emplea AJAX, JQuery y Javascript, para la actualización asíncrona de la vista, y es el encargado de enviar la pregunta al servidor y esperar su respuesta. La funcionalidad aportada por AJAX es vital para este tipo de sistemas; ya que permite realizar cambios sobre la página sin necesidad de recargarla de nuevo, lo que aumenta la interactividad, velocidad y usabilidad del sistema.

El servidor web contiene tres partes fundamentalmente: el módulo de análisis de pregunta,

programado en PHP, el algoritmo de respuesta, desarrollado con el analizador morfológico Flex, y la base de datos anteriormente mencionada.

## 4. Implementación

Como contenido fundamental de esta sección se especifican los aspectos y técnicas de desarrollo exclusivas del proyecto, los cuales le otorgan su distinción y particularidad y definen los métodos idóneos para llevar a cabo su implementación.

Asimismo, se presentan los diferentes módulos que componen el núcleo central de desarrollo, cimentados sobre la arquitectura anteriormente descrita, detallando sus funciones principales y funcionamiento, así como el modo en que se logra su interconexión.

### 4.1. Técnicas específicas

En primer lugar, cabe destacar que el bot responderá únicamente a cuestiones que versen sobre la asignatura; es decir, estará completamente ceñido a la materia, salvo alguna honrosa excepción, propia de todo chatbot. Éstos, comúnmente, tienen la capacidad de responder a preguntas, consideradas de baja prioridad, acerca de conocimientos que deben tener sobre sí mismos. “¿Quién es tu creador?”, “¿Dónde vives?” o “¿Cómo te llamas?”, son algunas de las preguntas, “extraoficiales”, que este sistema podrá contestar al usuario que se las formule.

Un punto importante en el desarrollo de este sistema consiste en otorgar flexibilidad y realismo al proceso de comunicación con el chatbot; para ello, se opta por utilizar diferentes estrategias:

- Dar ilusión de estar escuchando. Para ello se incluyen subcadenas de la entrada del usuario en la respuesta.
- Admitir ignorancia. El programa reconoce no saber la respuesta a algunas preguntas.
- Realizar un adecuado cambio de tema. Arrastrar al usuario hacia la conversación que el bot quiere, en lugar de ser él quien la elija.
- Reconocer cuándo el usuario divaga y no formula una pregunta orientada a conocer algo sobre la materia, instándole a hacerlo.

Una funcionalidad añadida al simple procedimiento de pregunta-respuesta es que el bot, tras reconocer una pregunta sobre la disciplina que debe resolver al usuario, además de proporcionarle la correspondiente respuesta, recurre a la página de los apuntes de la asignatura en la que se verifica la contestación ofrecida, mostrándosela por pantalla. Este hecho, evidentemente, refuerza la explicación proporcionada y, de alguna manera, sirve como referencia visual al alumno.

Otro aspecto a tener en cuenta es el seguimiento que se realiza de la evolución del alumno durante el curso. El sistema consulta el contenido de una tabla de la base de datos en la que se encuentran almacenados los aspectos en los que el estudiante ha fallado en el último examen de la asignatura, haciéndole un recordatorio de manera automática del contenido que debería revisar.

Con el fin de realizar una correcta clasificación de las preguntas realizadas por el usuario, que ayude a su mejor tratamiento y comprensión, se lleva a cabo su catalogación según tipo y prioridad.

- De alta prioridad. Son aquellas consultas que contienen patrones considerados clave por el sistema. Admitir ignorancia. El programa reconoce no saber la respuesta a algunas preguntas.
- De baja prioridad. Aquellas relacionadas con los conocimientos que el robot debe tener acerca de él mismo o de la asignatura (¿Cómo te llamas? – ‘What is your name?’, ¿Cuándo será el próximo examen? – ‘When will the next exam take place?’).
- De orientación. Se proporcionan cuando no se encuentran las consultas ingresadas, con el objetivo de orientar al alumno hacia lo que desea saber realmente (¿Qué quieres saber de...? – ‘What do you want to know about...?’, ¿Te refieres a...? – ‘Do you mean...?’).

## 4.2. Módulos

Como se dejó constancia en la especificación de la arquitectura, el sistema contará con varios módulos, especializados cada uno en una función concreta.

### 4.2.1. Interfaz

A través del interfaz se produce la interacción del usuario con el sistema; se trata del elemento virtual que identifica visualmente a la aplicación. El aspecto que más destaca ante el usuario es el avatar gráfico que representa al bot. Este, cambiará su imagen en función del contenido hacia el que haya evolucionado la conversación, constando de un total de 9 imágenes distintas, siendo algunas de ellas las siguientes:

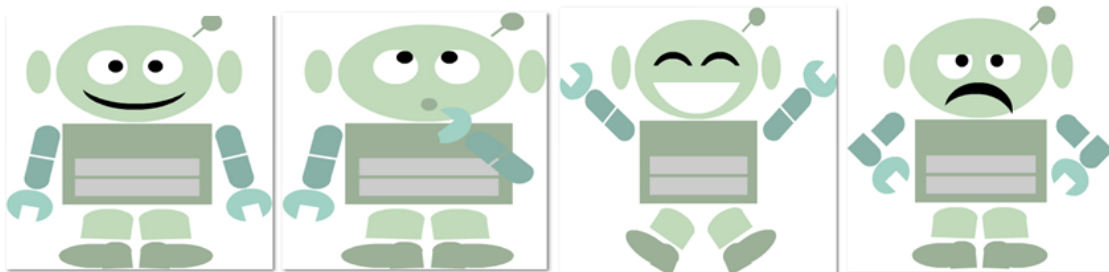


Figura 2. Ejemplo de avatares del chatbot

### 4.2.2. Validación de Entrada

El módulo de Validación de Entrada se encarga de dar la bienvenida al usuario al sistema y de realizar una primera consulta en base de datos para comprobar si el usuario debería repasar contenidos de la asignatura, en función de los resultados obtenidos en sus exámenes. En caso de que así sea, se le mostrará la siguiente pantalla, donde se le indicará que sus resultados obtenidos no han sido demasiado buenos, y que debería revisar los conceptos recogidos en las ventanas que aparecen a su derecha:

Este módulo, además, tiene como objetivo enviar la pregunta al servidor (previa validación de la misma), donde se analizará, y esperar la respuesta, incluyéndola, tras ser recibida, al final del área de texto. La validación de la entrada se realiza eliminando de ella los símbolos y caracteres que puedan dificultar su comprensión por los siguientes módulos que la traten (?, !, [, ], etc.), según su equivalencia con los correspondientes códigos ASCII.



Figura 3. Propuesta de revisión de contenidos

#### 4.2.3. Análisis de la pregunta

El módulo de Análisis de la Pregunta, íntegramente programado en PHP y albergado en el servidor web, tiene como objetivo, como su propio nombre indica, analizar el contenido de la entrada del usuario, para detectar en ella los patrones coincidentes con el contenido de las tablas *Saludos*, *Estados*, *Insultos*, e *Información* de la base de datos. Así, al llegar una pregunta, se separan cada una de las palabras de la misma en un vector, siendo contrastados a continuación los términos clave con el contenido de las citadas tablas. Dichos procedimientos de comparación pueden considerarse sub-módulos dentro del análisis.

El primer sub-módulo que entra en acción es el que opera sobre la tabla de Insultos. El sistema se encarga de detectar palabras malsonantes en la pregunta, y de proporcionar una respuesta adecuada a las circunstancias. Se contabiliza el número de veces que el alumno ha incluido una palabra de este tipo, hasta un máximo de tres, momento en el que es expulsado de la aplicación.

Una vez pasado el filtro impuesto por el primer sub-módulo, se procede a comprobar si el usuario ha saludado. En caso afirmativo, el bot hará lo propio, dándole la bienvenida.

El sistema, asimismo, tiene contabilizadas el número de ocasiones en las que el usuario saluda, de manera que, en caso de realizarlo más de una vez, le recordará que ya lo hizo previamente, animándole a formular una pregunta relacionada con la asignatura.

El tercer sub-módulo que analiza la entrada es el de *Estados*, el cual se encarga de dar respuesta a preguntas destinadas a conocer cómo se encuentra el bot, cuál es su estado.

Por último, el sub-módulo de *Información* finaliza el análisis. Su objetivo es contestar cuestiones básicas sobre el propio chatbot (quién te creó, dónde vives, etc.) y la asignatura (cuáles son la hora y aula del próximo examen, cuál es despacho de tutorías, quién es el profesor de la asignatura, etc.)





Figura 4. Secuencia de saludos

Destaca que, para ciertas repuestas, el sistema abre un *pop-up* con el documento que las certifica. Por ejemplo, si el usuario pregunta que en qué aula tendrá lugar el próximo examen de la asignatura, el chatbot añadirá a su respuesta el documento con toda la información acerca del examen.

En caso de que ninguno de los sub-módulos anteriores haya sido capaz de proporcionar una respuesta adecuada (siempre que la pregunta cubra las características propias de ellos), la entrada pasará íntegra al Algoritmo de Respuesta, que analizará de una manera más profunda la composición de la misma.

El modo de envío de la cadena al Algoritmo de Respuesta será a través de un fichero, añadiendo a la cadena el carácter '#', el cual indicará el fin de la entrada.

#### 4.2.4. Algoritmo de Respuesta

Cuando se habla de Algoritmo de Respuesta se quiere hacer referencia, fundamentalmente, a aquéllas respuestas académicas extraídas de los apuntes de la asignatura que el bot proporciona al alumno.

Dicho Algoritmo ha sido desarrollado en Flex, herramienta de análisis morfológico diseñada para generar escáneres, es decir, programas que reconocen patrones léxicos en un texto. Flex es capaz de reconocer las expresiones regulares previamente definidas, según la descripción del escáner a generar, en la cadena de entrada, la cual divide en tokens<sup>2</sup> para su correcto análisis.

Las palabras que este sistema reconocerá como patrones son las siguientes:

- Conceptos propios y específicos de la materia educativa.
- Términos que determinen lo que el usuario quiere saber de los conceptos de la materia

<sup>2</sup> Un token es una unidad sintáctica gramatical con significado propio.

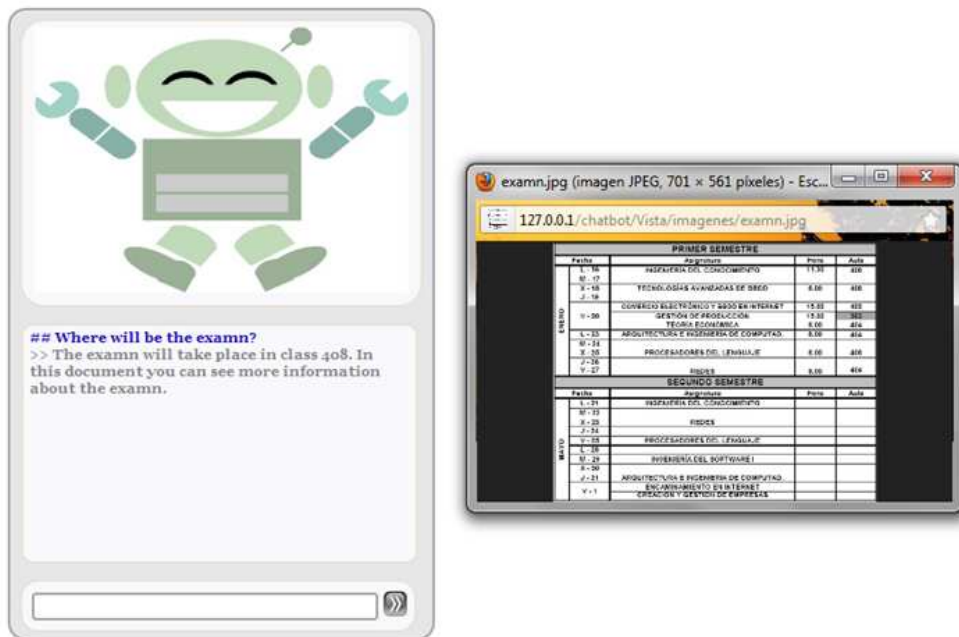


Figura 5. Pregunta por el aula en la que tendrá lugar el examen

(función, tipos, etc.).

- Pronombres interrogativos que conduzcan a la formulación de la clase de pregunta (qué, cuál, etc.).
- Palabras y expresiones utilizadas típicamente para despedirse.
- Vocablos empleados para expresar agradecimiento.
- Expresiones para referenciar coloquialmente al interlocutor.

Las reglas de reconocimiento de dichos patrones tienen la estructura [patrón]-[acción], o lo que es lo mismo, si se reconoce en la cadena de entrada alguno de los patrones establecidos, se ejecutará la acción que se detalle a continuación, escrita en código C. Esa acción puede conllevar el inicio de un nuevo estado.

La organización del código Flex se divide por estados, los cuales determinan la estructura lógica de la cadena introducida. De acuerdo a ello, las palabras de la cadena de entrada van transitando de estado a estado, procurando la formulación de las preguntas con una completa concordancia gramatical.

Los distintos estados en los que queda dividido el código Flex, y algunas de sus funciones principales son:

- **INITIAL.** Se encarga de dirigir la ejecución hacia los correspondientes estados en función del token recibido, tal y como se muestra en la Figura 5, controlar cuándo el usuario no formula una pregunta correcta, o elaborar una respuesta cuando el usuario no pregunte nada acerca de la asignatura.
- **PREGUNTA.** A él se llega cuando, desde INITIAL, se ha detectado un pronombre interrogativo. Se encarga de dirigir la ejecución hacia el estado correspondiente según la entrada, y de detectar preguntas ajenas al contenido de la asignatura. Para ello, lleva un contador

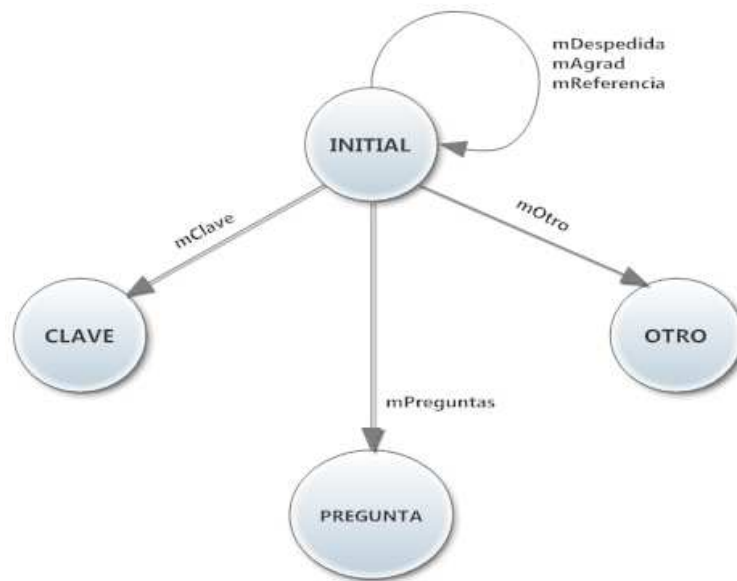


Figura 6. Autómata del estado INITIAL

del número de palabras que transcurren entre la aparición del pronombre interrogativo y la siguiente clave y, si es mayor que 3, deshecha la posibilidad de que la pregunta verse sobre la materia.

- **OTRO.** A este estado se transita cuando se ha empleado un término que determine lo que el usuario quiere saber de los conceptos de la materia. Una vez en él, por tanto, sólo se considerará la llegada de patrones reconocidos como conceptos de la asignatura. Además tiene la capacidad de detectar, en una estructura anidada, la posible formulación errónea de las preguntas.
- **CLAVE.** Es el estado receptor de la cadena cuando se ha introducido un término clave de la asignatura. Su función principal consiste en detectar si la pregunta ha sido formulada correctamente, o bien no se ha mencionado nada significativo acerca de dicha clave.
- **INTERMEDIO.** Es el estado al que se transita después de la formulación correcta de una pregunta, es decir, cuando existe una estructura anidada de preguntas realizadas (detallada más adelante).

Dichos estados comparten la función de almacenar en un fichero el conjunto de palabras clave encontradas en la entrada del usuario, para realizar su posterior cotejamiento con el contenido de la base de datos.

Por otro lado, el empleo de Flex permite otorgar al sistema de las siguientes habilidades:

- Almacenamiento de un **histórico** de lo cuestionado en la iteración anterior.
- Admisión de la formulación de cualquier número de preguntas en la misma entrada, permitiéndose la introducción de **varias estructuras anidadas**, como podría ser *'Could you tell me how does the semantic analysis work and what is a token?'*. Para ello, cuando se detecta un término de la asignatura tras haber empleado con anterioridad un pronombre interrogativo y/o un término que determine lo que el usuario quiere saber de los conceptos de la materia, siendo la estructura de la frase gramaticalmente correcta, se almacena la palabra correcto en el fichero a cotejar, la cual indica que el contenido anterior del fichero

conforma una búsqueda positiva en la base de datos. A continuación, se salta al estado INTERMEDIO, para continuar con el análisis de los siguientes vocablos.

- Advertir la hora del día en que se está produciendo la conversación, e incluir referencias sobre lo temprano o tarde que es, así como despedirse con 'Good night' en caso de superar determinada una hora.
- Detección e inclusión de referencias amistosas a su interlocutor en la respuesta.
- Conocimiento de la duración de la conversación (por medio de un fichero que actúa como contador del número de frases dichas por un mismo usuario) y, en caso de brevedad de la misma, ser capaz de comunicarle al alumno su asombro y preguntarle si no desea resolver ninguna duda adicional.

Al finalizar el análisis de la cadena, se devuelve el control al motor PHP, donde se comienza una búsqueda en base de datos de la posible correspondencia entre el contenido de ésta y las palabras clave detectadas.

La tabla sobre la que se realiza la búsqueda está constituida a través de una **ontología** de términos de la asignatura en función de la diapositiva de los apuntes a la que hacen referencia; es decir, cada diapositiva contiene una serie de palabras clave, asociadas de manera unívoca, en su conjunto, a ella. Con este método, no es necesario indicarle al sistema todas las formas y expresiones posibles con las que el usuario puede hacer referencia a una pregunta concreta, sino que únicamente se requiere la manifestación de ciertas palabras clave, de las que se realiza su **composición de lugar**.

Al realizar la búsqueda en base de datos por medio de dichas palabras, podrá existir una concordancia total de las palabras clave a una diapositiva concreta, o no. En caso de que dicha concordancia sea total, se elaborará la correspondiente respuesta a lo preguntado y, además, se abrirá un *pop-up* que muestre la diapositiva que certifique la respuesta dada.

Sin embargo, si la concordancia es parcial, habiendo varias diapositivas involucradas en el proceso, se realizará una operación denominada de **máxima coincidencia**, asignando un porcentaje que simbolice el grado de pertenencia de la pregunta a cada una de las diapositivas. La que alcance el mayor valor, será la que se ofrecerá como respuesta.

## 5. Conclusiones y trabajos futuros

Tras la elaboración del proyecto, se han obtenido varias conclusiones en relación a los agentes virtuales:

- El usuario no disfruta en una conversación mantenida con una máquina, por tanto, los chatbots han de desarrollarse de tal modo que sean capaces de imitar el comportamiento humano.
- El mejor método de comprensión empleado por un chatbot es el análisis de los patrones de diálogo empleados por su interlocutor humano.
- Una mayor aproximación a la entrada literal del usuario, conlleva una mayor probabilidad de fallo en la respuesta ofrecida.
- Elaborar un chatbot de uso específico a partir de un diseño de uso general limita el desarrollo.

En cuanto a los trabajos futuros, además de los puntos orientados al contexto educativo citados anteriormente, este desarrollo constituirá una sólida base para la adaptación del prototipo a diversos entornos; destacando, por encima de todos, la aplicación.

## Referencias

- [1] VELÁSQUEZ CÓRDOBA, Luis Fernando. *Compromiso y trascendencia de la educación*. Revista electrónica de Psicología Social. FUNLAM, Bogotá, 2007.
- [2] GUÉDEZ, Víctor. *Educación y Proyecto Histórico Pedagógico*. Universidad Nacional Abierta, Fondo Editorial del Vicerrectorado Académico, Caracas, 1987.
- [3] KASABOV, Nikola. *Advanced Neuro-Fuzzy Engineering for Building Intelligent Adaptive Information Systems*. in: *Fuzzy Systems Design: Social and Engineering Applications*. L.Reznik, V.Dimitrov and J.Kacprzyk (eds) Heidelberg, Physica-Verlag 249-262, 1998.
- [4] HAYES-ROTH, Barbara. *An Architecture for Adaptive Intelligent Systems*. Artificial Intelligence, Vol. 72, 329-365, Elsevier, Stanford, 1995.
- [5] VANLHEN, Kurt. *Student Modeling*. Foundations of Intelligent Tutoring Systems, 55-78, Hillsdale. N.J. Lawrence Erlbaum Associates, 1998.
- [6] WOLF, B. *Context Dependent Planning in a Machine Tutor*. Ph.D. Dissertation, University of Massachusetts, Amherst, Massachusetts, 1984.

### Sobre las autoras:

*Nombre:* Celia Gómez Róspide

*Correo electrónico:* cgrospide@gmail.com

*Institución:* ICAI, Universidad Pontificia Comillas.

*Nombre:* Cristina Puente Águeda

*Correo electrónico:* cpuente2@upcomillas.es

*Institución:* ICAI, Universidad Pontificia Comillas.

