

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN FRAMEWORK DE GESTIÓN DE APRENDIZAJE VIRTUAL PARA BRINDAR ACOMPAÑAMIENTO A ESTUDIANTES MEDIANTE EL USO DE UNA TÉCNICA DE INTELIGENCIA ARTIFICIAL INMERSA EN UN CHATBOT

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

CHÁVEZ MELO PATRICIO ALEJANDRO
patricio.chavez@epn.edu.ec

DIRECTOR: PAZ ARIAS HENRY PATRICIO
henry.paz@epn.edu.ec

Quito, mayo 2020

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Patricio Chávez, bajo mi supervisión.

**Paz Arias Henry Patricio
DIRECTOR DE PROYECTO**

DECLARACIÓN DE AUTORÍA

Yo, Patricio Chávez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Patricio Alejandro Chávez Melo

DEDICATORIA

A mis padres Azucena y Patricio.

Por brindarme su apoyo incondicional, por guiarme en los rumbos adecuados de la vida. Los quiero mucho.

A mis hermanos Fer, Carlos, Lore, Diani y Cris.

Por siempre estar ahí en los buenos y difíciles momentos de mi vida y en especial de mi etapa universitaria.

Patricio.

AGRADECIMIENTO

Agradezco a mis padres que han sido mis mentores, los que me han enseñado que todo en la vida requiere esfuerzo, que mientras más esfuerzo se necesite más grata será la recompensa, aquí el fruto de todo mi esfuerzo. Eternamente agradecido.

Un especial agradecimiento a mis hermanos, sin su apoyo en momentos clave de mi vida este logro no podría haberse realizado. Gracias.

Mis agradecimientos a la casa que me vio pasar por el proceso de aprendizaje, mi querida, “Escuela Politécnica Nacional del Ecuador”.

Yo siempre he dicho que la educación es la que mantiene a un pueblo culto y responsable de sus actos, la labor de un profesor es difícil ya que es el pilar en el cual este conocimiento florece, mis admiraciones y agradecimientos a todos los profes que antes de ser profesores son mis amigos. Gracias.

Finalmente quiero agradecer a usted Henry, por brindarme su guía y paciencia en este proceso final de mi etapa profesional universitaria.

Patricio.

ÍNDICE DE CONTENIDO

CERTIFICACIÓN	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE TABLAS	VII
ÍNDICE DE FIGURAS	VIII
RESUMEN	IX
ABSTRACT	X
1. INTRODUCCIÓN	1
1.1 Antecedentes	1
1.2 Objetivos	2
1.2.1 Objetivo General.....	2
1.2.2 Objetivos Específicos.....	2
1.3 Alcance	2
1.4 Herramientas	3
1.4.1 Moodle (versión 3.7.1).....	3
1.4.2 Moodle chat (mod/chat).....	3
1.4.3 Apache Server.....	3
1.4.4 MySQL.....	4
1.4.5 AIML (versión 1.0).....	4
1.4.6 Program O (versión 2.6.11).....	5
1.4.7 TreeTagger.....	5
1.4.8 TreeTaggerWrapper (versión en Python).....	5
1.4.9 Flask.....	6
1.4.10 Heroku.....	6
2. METODOLOGÍA	7
2.1 Definición de herramientas del framework	7
2.1.1 Análisis de herramientas LMS.....	7
2.1.2 Análisis de herramientas de CHATBOT.....	11
2.2 Implementación del LMS	13
2.2.1 Elementos de e-learning.....	14
2.2.2 Modalidades de e-learning.....	15
2.2.3 Definir Objetivos y Necesidades.....	16
2.2.4 Elaboración y/o Migración de Cursos.....	17

2.2.5	Montaje de Cursos	18
2.2.6	Desarrollo de los Cursos.....	19
2.2.7	Evaluación y Retroalimentación.....	19
2.3	Desarrollo del agente conversacional (Chatbot)	20
2.3.1	Métodos utilizados para desarrollar Chatbots.....	20
2.3.2	Etiquetador de Textos.....	22
2.3.3	Estructura del Chatbot.....	23
2.3.4	Determinación de la Base del Conocimiento (BC)	25
2.4	Integración del framework	37
2.4.1	Arquitectura del framework	38
2.4.2	Backend.....	39
2.4.3	Frontend	40
2.4.4	Estructura del framework	41
3.	RESULTADOS Y DISCUSIÓN	43
3.1	Resultados	43
3.1.1	ESCENARIO 1.....	44
3.1.2	ESCENARIO 2.....	46
3.2	Discusión.....	50
4.	CONCLUSIONES Y RECOMENDACIONES.....	51
4.1	Conclusiones	51
4.2	Recomendaciones	52
5.	REFERENCIAS BIBLIOGRÁFICAS	53
6.	Bibliography.....	53
7.	ANEXOS.....	57

ÍNDICE DE TABLAS

Tabla 1: Cuadro Comparativo LMS Código abierto. Fuente: [16]	10
Tabla 2: Cuadro Comparativo Chatbot AIML. Fuente: [10] y [22]	12
Tabla 3: Definición de Objetivos y Alcance del curso proceso e-learning.	16
Tabla 4: Estructura del curso proceso e-learning.	17
Tabla 5: Modelo de encuesta proceso e-learning. Fuente: [23]	19
Tabla 6: Comparativa métodos Chatbot Fuente: [25]	21
Tabla 7: Conjugaciones y Lema del verbo Terminar TreeTagger	22
Tabla 8: Puntuación Program O. Fuente: [10]	25
Tabla 9: Lista de preguntas y respuestas.	27
Tabla 10: División de Pronombre Interrogativo con su Contexto.	28
Tabla 11: Lista de preguntas etiquetadas y respuestas.	33
Tabla 12: Lista de preguntas etiquetadas y depuradas con sus respectivas respuestas.	34
Tabla 13: BC del Escenario 1	44
Tabla 14: Resultados Escenario 1	45
Tabla 15: BC del Escenario 2	46
Tabla 16: Resultados Escenario 2	47
Tabla 17: Promedio de tiempo por escenario	47
Tabla 18: Tipos de Respuesta del chatbot.	48
Tabla 19: Comparación con agente conversacional AIML en [50]	50

ÍNDICE DE FIGURAS

Figura 1: Proceso de aprendizaje e-learning. Fuente: [23]	15
Figura 2: Montaje del curso proceso e-learning	18
Figura 3: Modelo arquitectura cliente-servidor del framework	39
Figura 4: Vista del Chat modificado	41
Figura 5: Estructura del framework	42
Figura 6: Matriz de confusión de los resultados obtenidos.	49

RESUMEN

El presente trabajo se enfoca en brindar una solución informática, a través de un framework, para entidades que necesiten un sistema de gestión de capacitación o aprendizaje y algún tipo de acompañamiento virtual en forma de chatbot o agente conversacional, dirigido a sus estudiantes que tomen cursos bajo la metodología a distancia o semipresencial.

El framework se basa en la arquitectura cliente-servidor utilizando reglas de inferencia como base del conocimiento escritas en el lenguaje de etiquetas AIML. Al necesitar que el chatbot simule una conversación natural con los estudiantes se utilizará un etiquetador de textos a manera de un servicio REST que predefinirá el lema de cada texto antes de buscarlo en la base del conocimiento, mejorando su capacidad de entendimiento.

Palabras clave: Framework, Sistema de Gestión de Aprendizaje, Chatbot, Agente Conversacional, AIML, Etiquetador de Textos, REST

ABSTRACT

This work focuses in giving an informatic solution alternative, through a framework, for entities which need a training or learning management system management system and some kind of virtual accompaniment in a chatbot or conversational agent form, aimed at their students whom are taking courses under distance or blended methodology.

This framework is based on a client-server architecture using inference rules as a knowledge base written in tag languages (AIML). When we need that the chatbot simulates a natural conversation with students, we will use a text tagger in a REST service way which will predefine the lemma of each text before looking for it in the knowledge base, improving its understanding capacity.

Keywords: Framework, Learning Management System, Chatbot, Conversational Agent, AIML, Text Tagger, REST

1. INTRODUCCIÓN

1.1 Antecedentes

El “Instituto de Espiritualidad Santa Teresa del Niño Jesús” [1] dirigido por los Frailes Carmelitas Descalzos en Ecuador, es un instituto académico de formación Espiritual, que actualmente su oferta académica se basa en tres modalidades de estudio: “Semipresencial” (semestres, 3 a 4 años de duración), “A distancia” (módulos, duración indefinida) y “Cursos Cortos” (tema, de hasta 20 horas de duración). Hoy en día la necesidad principal que presenta el Instituto se encuentra en los cursos para la modalidad a distancia, que, a pesar de contar con materiales didácticos impresos, visuales, audio visuales, tableros didácticos, entre otros, para el proceso de enseñanza aprendizaje a través del correo electrónico, como único canal de comunicación entre el tutor y el estudiante, lo que genera inconvenientes tanto para el estudiante, pues el nivel de respuesta de su tutor toma mucho tiempo a pesar de que la inquietud no tenga un nivel de dificultad alto, y de igual manera, para el tutor académico, pues le toma mucho tiempo el proceso de calificación y retroalimentación, y lo más importante no se aprovechan los recursos tecnológicos que hoy día pueden mejorar para la educación que no es presencial.

Uno de los objetivos principales de la educación a distancia es brindar una alternativa a la demanda de personas que tienen algún tipo de impedimento para asistir de forma presencial, educación [2]. Por tanto, para simular una educación a distancia plena es necesario simular un acompañamiento físico en cualquier momento del tutor hacia sus estudiantes.

De acuerdo a la información obtenida mediante una entrevista [3], dirigida al director del “Instituto de Espiritualidad Santa Teresa del Niño Jesús”, se logró evidenciar dos problemas que afectan de manera clara la coordinación y gestión de procesos cotidianos del Instituto, estos problemas detectados fueron: primero, la falta de un modelo de gestión educativo; y, el segundo, la falta de acompañamiento continuo entre el tutor y sus estudiantes. Con ello, basado en este último, el presente trabajo pretende brindar una alternativa de solución a esta problemática, desarrollando un framework mediante el uso de herramientas de código abierto que conste de dos partes: un sistema de gestión de aprendizaje virtual (*Learning Management System*, LMS) [4] y un asistente virtual (chatbot).

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un framework de gestión de aprendizaje virtual para brindar un acompañamiento a los estudiantes mediante el uso de una técnica de inteligencia artificial inmersa en un chatbot.

1.2.2 Objetivos Específicos

- Implementar un LMS de código abierto que se acople a los requerimientos de los estudiantes y tutores facilitando un control dentro del proceso de gestión del aprendizaje.
- Desarrollar un chatbot con técnicas de inteligencia artificial para brindar un apoyo constante a cada estudiante facilitando el proceso de enseñanza y aprendizaje.
- Integrar el framework dentro de un ambiente e-learning para el acompañamiento virtual asincrónico en los cursos dirigidos a estudiantes.

1.3 Alcance

Este proyecto está encaminado a desarrollar un framework¹, utilizando herramientas de código abierto, para crear un ambiente web en el cual se almacene: la información personal y académica de los estudiantes, los cursos en los cuales están inscritos, las calificaciones de las tareas asignadas por un tutor; y finalmente un agente conversacional (chatbot) que brinde un acompañamiento a los estudiantes que ingresen a esta plataforma. La conversación con el chatbot se llevará a cabo a través de un servicio web.

¹ Framework: marco de trabajo en el cual se llevará a cabo un determinado proceso, o conjunto de herramientas

1.4 Herramientas

Para el desarrollo del framework se usarán varias herramientas que darán solución a partes específicas del problema, estas herramientas son:

- Un LMS que será la base del gestor del conocimiento.
- Un agente conversacional que utilice un lenguaje entendible tanto para el usuario como para sí mismo.
- Una plataforma en la nube para almacenar el servicio web que procesara las conversaciones con el agente conversacional.
- Un etiquetador de texto capaz de devolver el discurso y el lema de cada palabra para permitir que la conversación sea mas fluida entre el estudiante y el chatbot.

1.4.1 Moodle (versión 3.7.1)

Este LMS esta escrito principalmente en PHP, y distribuido bajo licencia de código abierto, fue creado con el propósito de brindar a estudiantes, profesores y administradores una plataforma robusta de gestión de la educación [5], Moodle además, utiliza MySQL para almacenar los datos del sistema.

1.4.2 Moodle chat (mod/chat)

Moodle posee una serie de módulos y/o complementos (plugins) que brindan algunas funcionalidades extras como por ejemplo el chat. Este complemento permite crear sesiones entre grupos definidos de estudiantes y profesores [6]. Se utilizará la versión Ajax de este modulo para el desarrollo del framework.

1.4.3 Apache Server

Es un servidor HTTP de código abierto desarrollado para los mas modernos sistemas operativos que existen [7], proporcionando seguridad y eficiencia a los servicios http de

acuerdo con protocolos y estándares actualizados. Permite levantar de manera fácil un ambiente MOODLE en sencillos pasos.

1.4.4 MySQL

Es un motor de base de datos relacional tan flexible que puede ser utilizado tanto en aplicaciones grandes como en pequeñas. Este motor se distribuye bajo licencias de código abierto y utiliza los estándares actuales de SQL [8]. Es el sistema de base de datos escogido por MOODLE.

1.4.5 AIML (versión 1.0)

El lenguaje de etiquetas para inteligencia artificial, (*Artificial Intelligence Markup Language*, AIML) es un lenguaje de etiquetas basado en XML desarrollado para crear agentes conversacionales de lenguaje natural. Escrito por Richard Wallace a inicios del año 1995 [9], este lenguaje esta basado en reglas, es decir, que se asocia al paradigma “manejado por datos” el cual es diferente al de programación secuencial. Las reglas son pequeñas proposiciones armadas por “una condición y una acción”, también referidas como una “premisa y conclusión”.

En su versión 1.0, AIML utiliza wildcards o comodines “_” y “*” para especificar, respectivamente, el patrón 1 o mas ocurrencias. Mientras que en la versión 2.0 se añaden mas comodines como el “^” y “#” que definen 0 o mas ocurrencias del texto respectivamente.

Las etiquetas mas básicas del lenguaje AIML son:

- **<aiml>**: esta etiqueta sirve para definir todo el contenido aiml dentro de un archivo.
- **<category>**: es una categoría o regla de la base del conocimiento. Esta etiqueta está compuesta por las etiquetas <pattern> y <template> que representan la premisa y conclusión de una proposición.
- **<pattern>**: define la premisa de la regla.
- **<template>**: define la conclusión de la regla.
- **<srai>**: (*“symbolic reduction artificial intelligence”*) sirve para concatenar las conclusiones de las reglas.

- **<star *>**: almacena el carácter o conjunto de caracteres de la premisa que se escribirán en la conclusión tal cual fueron almacenados.

1.4.6 Program O (versión 2.6.11)

Program O es un interprete de lenguaje AIML escrito en PHP. Utiliza MySQL como sistema de almacenamiento de datos e información del chatbot incluyendo la base del conocimiento escrita en los archivos AIML [10]. Soporta caracteres UNICODE² permitiendo expresar en lenguaje AIML a los lenguajes como español, chino, árabe, entre otros. También tiene un completo sitio de administración donde se pueden cambiar el nombre del chatbot y agregar, editar o eliminar archivos AIML. Posee además una API que devuelve un objeto JSON con la respuesta mas adecuada del chatbot, escogida entre todas las posibilidades de respuestas que forman la base del conocimiento.

1.4.7 TreeTagger

TreeTagger es una herramienta de anotación de texto que divide en lema y discurso de cada palabra [11]. Está escrita en Perl y utiliza los arboles de decisión del modelo de Markov para estimaciones mas confiables de parámetros contextuales.

1.4.8 TreeTaggerWrapper (versión en Python)

Esta herramienta, desarrollada por Laurent Pointal, se encuentra desarrollada en Python y contiene todo lo necesario para utilizar TreeTagger en un ambiente Python. La función tag_text() devuelve cada palabra con su definición o discurso y su lema. Es capaz de trabajar con casi 20 lenguajes naturales principales como el español, ingles, francés, entre otros. [12].

² UNICODE: estándar de codificación de caracteres que facilita la interpretación, visualización y transmisión de diferentes textos en varios idiomas.

1.4.9 Flask

Es un framework minimalista escrito en Python que permite crear aplicaciones web en pocos pasos y con sencillas líneas de código. Además permite crear servicios con la arquitectura REST sin la mayor cantidad de librerías externas [13].

1.4.10 Heroku

Heroku es una plataforma en la nube que permite levantar, administrar y monitorear aplicaciones escritas casi en cualquier lenguaje [14]. Es ideal para la creación de microservicios en ambientes tanto de prueba como de producción.

2. METODOLOGÍA

En esta sección se detallan los procesos de construcción del framework dividido en dos partes principales: la implementación del LMS y el desarrollo del CHATBOT. También se describen los pasos para la ejecución del framework con los cuales los estudiantes podrán resolver sus inquietudes dentro de un ambiente asíncrono “*e-learning*”.

2.1 Definición de herramientas del framework

El framework contemplara dos elementos principales, el LMS y el Chatbot, los cuales serán escogidos en esta sección luego de un análisis de las herramientas vanguardistas existentes en el mundo del código abierto.

2.1.1 Análisis de herramientas LMS

Un LMS es un sistema formado por un conjunto de servicios interactivos en línea que brindan a los estudiantes acceso a información, herramientas y recursos para respaldar el proceso de formación a través de Internet [15].

Debido a que en el presente trabajo se pretende desarrollar una solución de código abierto, solo se analizaran herramientas bajo este contexto. Inicialmente, definiremos los parámetros o indicadores que sean comunes dentro de la problemática definida; los cuales se tomarán en cuenta para encontrar la mejor opción de LMS. De acuerdo con los resultados hallados al aplicar instrumentos de investigación, con la entrevista [3] y con la investigación realizada por [16] los indicadores vamos a utilizar son:

- **Interactividad:** capacidad de interacción entre el usuario y el sistema informático.
- **Flexibilidad:** capacidad de implementar o agregar nuevas funcionalidades al sistema informático.

- **Funcionalidad:** conjunto de características capaces de brindar algún uso dentro de un sistema informático.
- **Soporte:** capacidad de encontrar información respecto al sistema informático.

De acuerdo con [16], se tomaron en cuenta varias herramientas para la administración de LMS enfocados en la interacción profesor-estudiante, analizando algunos aspectos de los cuales solo los pertinentes se detallan a continuación:

- **Autenticación de usuario:** es necesario que exista un proceso de ingreso para determinar el tipo de usuario y su rol posee dentro del LMS.
- **Autorización a los cursos:** debe existir un proceso de asignación de permisos a los cursos otorgados por algún tipo de usuario.
- **Servicios alojados:** es imprescindible mantener alojada la información tanto de los estudiantes como de los recursos utilizados en el proceso de aprendizaje.
- **Registro de estudiantes:** además de tener un registro de los roles de usuarios, para los estudiantes debe existir un registro a los cursos que el administrador crea pertinente.
- **Gestión de cursos:** existen cursos que tienen una modalidad secuencial, es decir que no se puede ingresar a una parte del curso sin haber culminado la parte que le antecede, resultan ser un pre-requisito obligatorio por cumplir.
- **Seguimiento de los estudiantes:** los profesores deben tener un acceso a las estadísticas que generaran sus estudiantes, como ejemplo: las veces que un estudiante entra al curso.
- **Apariencia:** la experiencia del estudiante y de los profesores debe ser amigable e intuitiva. El LMS debe tener una apariencia que permita que el aprendizaje se lleve acabo de manera fluida.

A demás, en otro análisis expuesto en [16], se especifican 5 divisiones del mercado emergente para LMS que se detallan a continuación:

- **Grandes Empresas:** son los proveedores de aplicaciones comerciales los que predominan este mercado de LMS.
- **PYMES:** son empresas que necesitan capacitación a un precio bajo, por lo cual optan por soluciones de código abierto.
- **Gobierno y Sector Público:** son entidades que necesitan brindar un servicio de capacitación a sus varios procesos y regulaciones. Por tanto, al ser un mercado tan amplio, se necesita de soluciones a gran escala. Últimamente se está optando por soluciones LMS de código abierto.
- **Educación Primaria y Secundaria:** en las escuelas y colegios predominan los proveedores LMS comerciales. Sin embargo, se están tomando en cuenta otras opciones de LMS para reemplazarlas.
- **Implementaciones rápidas:** este mercado es amplio debido a que existen entidades que solo necesitan capacitar a sus miembros por un tiempo específico o simplemente en un área del conocimiento. Su mejor opción son LMS de código abierto.

Según [16], las funcionalidades y elementos imprescindibles o mínimas que un LMS debe tener para que la gestión del aprendizaje o conocimiento sea reconocida como tal, son los siguientes:

- **Gestión del estudiante:** control de autoría, y edición de los recursos.
- **Mecanismos de acceso a recursos:** objetos de aprendizaje y otros tipos de gestión de contenidos.
- **Elaboración de recursos:** guía de creación de nuevos contenidos.
- **Administración cualitativa:** evaluación continua de los estudiantes.
- **Seguimiento al estudiante:** compartición de recursos entre estudiantes.

Congruentemente, luego de revisar los análisis de los diversos conceptos, elementos y procedimientos expuestos anteriormente que hacen de un LMS un sistema integro; considerando también que el entorno de solución deseado pertenece al grupo de herramientas

de código libre; y además, en acuerdo con el cuadro comparativo encontrado en [16] se presenta la comparación entre herramientas mostradas en la Tabla 1:

Tabla 1: Cuadro Comparativo LMS Código abierto. Fuente: [16]

INDICADORES	Moodle	Dotlrn	Chamilo	Sakai	Edu2.0
Flexibilidad	7.00	7.00	9.00	6.00	8.00
Funcionalidad	7.33	5.33	7.33	4.67	8.00
Interactividad	6.57	7.14	8.00	6.00	8.86
Soporte	10.00	5.33	7.00	5.33	5.33
PROMEDIO	7.8	6.2	7.8	5.5	7.5

Tomando en cuenta el análisis expuesto en la Tabla 1, la herramienta escogida es Moodle [5], como LMS debido a su alto grado de información para el desarrollo de nuevas funcionalidades en forma de complementos.

Moodle posee una gran cantidad de funcionalidades extra llamadas plug-in³ o complementos que se los puede gestionar desde el módulo de administración. Existen algunos complementos que ya forman parte de la versión estable de Moodle, que son guardados en la carpeta mod/...; como el complemento de “chat” que es el de mayor interés para el presente trabajo, inferido también como “mod/chat” [6], permite a los estudiantes y profesores mantener una conversación asincrónica de manera que se puedan compartir inquietudes, soluciones, datos o cualquier tipo de información digital. Este módulo permite crear sesiones de chat para grupos definidos de estudiantes que caduquen en un tiempo determinado. Existen 3 métodos para ejecutar este complemento:

- **AJAX:** “Asynchronous JavaScript and XML” [17], método por el cual la comunicación con el servidor es asíncrona. Este método puede fallar cuando existen varios estudiantes conectados al mismo tiempo.
- **Normal:** se mantienen las sesiones en forma de stream de datos. Es el método mas básico del chat.

³ Plug-in: aplicación en forma complemento que brinda una funcionalidad extra de manera específica a otra aplicación.

- **Chat server Daemon⁴**: se requiere enviar los datos del servidor para que el proceso funcione todo el tiempo; esto implica mayor capacidad de memoria y procesamiento dentro del servidor que almacena Moodle, haciéndolo menos eficiente.

De todos estos se utilizará el método por defecto “AJAX”. Esto debido a que con este método, no es necesario levantar otra instancia de servidor adicional, para el procesamiento de la información del chat. Además, una de las mayores ventajas del método “AJAX” es que no requiere refrescar la página entera si no simplemente los elementos HTML que necesite para crear una vista, mejorando así el tiempo de respuesta del servidor.

Debido a que una de las principales características de la arquitectura de Moodle [18] es ser Modular para poder agregar más funcionalidades y permitir un crecimiento organizado de la plataforma, para el Frontend se utilizará y modificará el complemento de “chat” de Moodle ya que posee las funcionalidades necesarias para comunicarse con el chatbot. Mientras que para el Backend se manejará un método asíncrono que consuma el servicio REST [19] de la API del servidor de Program O.

2.1.2 Análisis de herramientas de CHATBOT

En [20], se hace referencia a dos bots: T-bot y Q-bot que sirven para brindar un acompañamiento virtual y evaluar a los estudiantes dentro de un LMS, estos bots están basados en el proyecto A.L.I.C.E. [21] y su base del conocimiento está creada en un conjunto de categorías en archivos AIML.

En base a este proyecto y con relación a que el LMS que se usará en el framework es Moodle, cuyo principal lenguaje de programación es PHP, se llevará a cabo un análisis de dos de las herramientas de Código abierto. Existen muchos intérpretes de AIML [9], de los cuales Program O [10] y Program E [22], serán los analizados en esta sección, con los mismos

⁴ Daemon: proceso que se ejecuta en background dentro de un sistema por un tiempo indefinido para realizar una tarea determinada, por ejemplo; obtener la hora del sistema.

indicadores evaluados en la investigación de LMS; flexibilidad, interactividad, funcionalidad y soporte.

A continuación, en la Tabla 2 se presenta una comparativa de las características mas relevantes de los dos programas:

Tabla 2: Cuadro Comparativo Chatbot AIML. Fuente: [10] y [22]

CARACTERÍSTICAS	Program O	Program E
Estructura Básica	AIML, PHP y MySQL	AIML, PHP y MySQL
Integración con el Frontend	Posee tres métodos de uso: mediante una pagina HTML estática, mediante AJAX o mediante un API REST.	Necesita de Adobe Flash ActiveScript para conectarse con una pagina web HTML. No posee otra forma de conectarse con el Frontend de la aplicación.
Información de soporte actualizada	Tiene una rama en Github constantemente actualizada.	Única página encontrada no oficial fue [22], en la cual no se muestra información alguna del actual estado de la herramienta.
Nivel de conocimiento en programación	Se necesita un nivel mínimo de conocimiento en lenguaje XML ya que su instalación es intuitiva con información actualizada.	Se necesita un nivel medio de conocimiento en lenguaje ActiveScript ya que para su funcionamiento necesita Adobe Flash.
Licencia para desarrollo	GNU	GNU
AIML versión	La versión 1.0 soporta la versión 1.0 de AIML pero la versión beta 3.0 soporta AIML 2.0.	No existe información.
Requerimientos de instalación	<ul style="list-style-type: none"> • PHP 4.0 o mayor • PDO – Driver de MySQL • cURL • mbString • Zip • Apache Server • MySQL 	<ul style="list-style-type: none"> • PHP 4.0 o mayor • Adobe Flash • Apache Server • MySQL
Administración de los bots	Tiene una completa interfaz de administración de bots con una pagina en la cual se puede escribir la personalidad del bot. Posee una interfaz de corrección de palabras y omisión de palabras agresivas. También posee un modulo de log y debug de conversaciones para saber que esta pasando con los bots instalados.	La única funcionalidad que posee es la de interacción de la comunicación entre el usuario y el bot.

Luego de considerar estas dos herramientas para el desarrollo del framework se decidió utilizar “Program O” [10] como herramienta principal de chatbot debido a los siguientes factores:

- Facilidad de creación de bots desde el módulo de administración.
- Facilidad de integración con el frontend mediante la funcionalidad de API.
- Nivel básico de conocimiento en lenguaje XML para su implementación.

2.2 Implementación del LMS

El e-learning o aprendizaje electrónico puede ser definido como el uso de tecnologías de computación e internet para entregar una gran variedad de soluciones que permitan el aprendizaje y mejorar su rendimiento [15].

Dado este concepto, un LMS permite gestionar todos los procesos de aprendizaje a través de e-learning. Se basa en la entrega de contenidos, haciéndolos lo mas interactivos posible para que la única herramienta de aprendizaje no sea solo la de lectura, si que se incluyan otros elementos digitales [23] como videos, imágenes, pruebas o quiz⁵, entre otros.

La educación o aprendizaje virtual cuenta con dos ambientes a tomar en cuenta, el aprendizaje basado en el profesor o tutor y el aprendizaje basado en el estudiante (autoaprendizaje).

- **Aprendizaje basado en el profesor:** el profesor es el que orquesta todos los procesos de aprendizaje de acuerdo con su experiencia, lo cual limita el aprendizaje. Se complementa con materiales de aprendizaje o también llamados recursos como: libros, revistas, internet; pero en definitiva el profesor es la mayor fuente de información. También conocido como modelo de enseñanza presencial.
- **Aprendizaje basado en el estudiante:** en este ambiente el estudiante tiene la libertad de encontrar la información mediante internet, foros, tutorías, entre otros; haciendo que el estudiante sea el principal gestor de su formación.

⁵ Quiz: es una especie de cuestionario con preguntas en algún contexto en línea.

Para la implementación del LMS se utilizará la metodología “*e-learning*” fundamentada en el aprendizaje basado en el estudiante. Esta metodología será descrita en las siguientes secciones a continuación.

2.2.1 Elementos de e-learning

De acuerdo con [23], los elementos mas básicos de e-learning son:

- **LMS:** es el núcleo de toda actividad, ya que sobre este elemento se cargarán las herramientas, recursos y contenido de cursos determinados. Es el elemento que gestiona todo el aprendizaje dentro de una plataforma en internet.
Para que el LMS cumpla con sus actividades principales se necesita de algunas herramientas, como por ejemplo: **1)** Herramientas de distribución de contenidos como repositorios de imágenes, de texto o de libros. **2)** Herramientas de comunicación o de colaboración, ya sea de manera síncrona como chats, o de manera asíncrona como e-mail y foros. **3)** Herramientas de administración como el proceso de matriculación y gestión de usuarios.
- **Contenidos:** es toda la información que se posee para generar el conocimiento, es decir, todo el material de aprendizaje en cualquier forma de comunicación: libros virtuales, revistas, imágenes videos, entre otros.
- **Sistemas de Comunicación:** como se lo menciono anteriormente existen varios sistemas de comunicación síncrona entre el tutor y estudiante o entre estudiantes de manera síncrona como los chats, videoconferencias o pizarras interactivas. Mientras que sistemas de comunicación asíncrona incluyen al correo electrónico o foros.
- **Otros:** aquí se pueden identificar otro tipo de actividades como casos prácticos o de interés, actividades de evaluación, artículos de interés o links a lecturas adicionales; todo en virtud de reforzar el aprendizaje.

En la Figura 1, se representa de manera clara como se genera el proceso de aprendizaje mediante la metodología de e-learning.

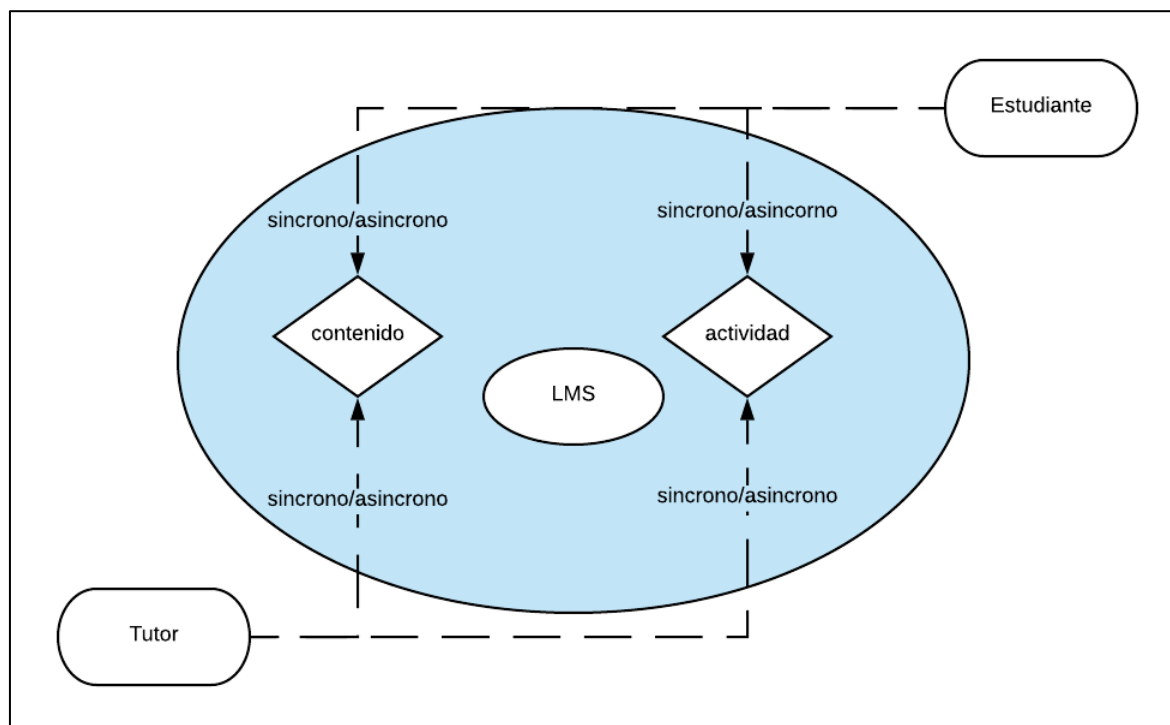


Figura 1: Proceso de aprendizaje e-learning. Fuente: [23]

2.2.2 Modalidades de e-learning

Existen una gran variedad de modalidades de e-learning [23]:

- **E-learning puro:** en esta modalidad se basa netamente en las TIC⁶. Se lleva acabo solo en internet sin ningún tipo de complemento presencial.
- **E-learning mixto o Blended learning:** en esta modalidad se adquiere lo mejor del proceso de aprendizaje del e-learning puro con sesiones de refuerzo presenciales.
- **Centrado en la tecnología:** modalidad que se enfoca en encontrar tecnologías para conocer la forma de transmitir los contenidos de la mejor manera, mas no en tener un tutor que este a cargo de un grupo de estudiantes.
- **Centrado en el profesor:** el profesor es el centro de toda fuente de información, por tanto, esta modalidad depende totalmente en la calidad de la interacción profesor-estudiante.

⁶ TIC: hace referencia a las tecnologías de información, que contemplan un conjunto de herramientas para el procesamiento, transmisión y almacenamiento de información.

- **Centrado en el estudiante:** aquí, el estudiante tiene la facilidad de planificar y gestionar su aprendizaje, teniendo a su alcance la mayoría de las actividades y recursos pedagógicos necesarios para cumplir con los objetivos de determinado curso.
- **Centrado en el contenido:** en esta modalidad se puede decir que los contenidos son lo suficientemente concretos y no es imprescindible la ayuda de un guía o tutor.

La implementación del LMS se llevará a cabo a través de la metodología de e-learning tomando en cuenta un ambiente de aprendizaje centrado en el estudiante. Este proceso de e-learning tiene 5 actividades principales: definir objetivos y necesidades, elaboración y/o migración de cursos, montaje de cursos, desarrollo de los cursos y finalmente, Evaluación y retroalimentación [23]. Las cuales se detallarán tomando como ejemplo ficticio el que una entidad necesita que sus desarrolladores aprendan los conceptos básicos de un chatbot con los conceptos de AIML. El curso ficticio será llamado "Curso básico de AIML" y será llevado por el proceso de e-learning desde su definición hasta su evaluación en las siguientes secciones.

2.2.3 Definir Objetivos y Necesidades

En esta actividad lo primordial es determinar los requerimientos de formación. Al ser la primera etapa en el proceso de formación e-learning es imprescindible clarificar el alcance, determinar su duración, definir para que, y para quienes se desarrollará el curso, definir la modalidad a ser utilizada, definir el nivel de conocimiento de los estudiantes y por ende los objetivos a ser alcanzados luego de culminar con el curso. Para lo cual se llenará la Tabla 3, sugerida por el autor basada en la información de [23].

Tabla 3: Definición de Objetivos y Alcance del curso proceso e-learning.

Datos	Requerimiento
Nombre	Curso básico de AIML
Duración	6 horas
Modalidad	Centrada en el estudiante.
Introducción	El presente curso pretende brindar una capacitación básica referente a los agentes conversacionales (chatbot) y aprender sobre su lenguaje principal AIML.
Dirigido	A los desarrolladores de la entidad.

Nivel de conocimiento	Se requiere un nivel mínimo de conocimiento por parte de los desarrolladores.
Necesidades	Para la entidad es necesario que los desarrolladores entiendan conceptos básicos de un chatbot para lo cual es necesario que puedan crear su propia base del conocimiento con el fin de crear flujos de conversaciones con los clientes de la entidad.
Objetivos	Al finalizar el curso el estudiante podrá usar las etiquetas mas básicas de AIML para crear una base del conocimiento de un chatbot.

2.2.4 Elaboración y/o Migración de Cursos

De acuerdo con [23] esta etapa del proceso de e-learning se refiere a la preparación del material para alcanzar los objetivos definidos en la actividad anterior. Aquí se pueden utilizar recursos que ya fueron utilizados anteriormente y actualizarlos de manera que se puedan utilizar nuevamente en algún otro curso o capacitación. Sin embargo, se debe procurar mantener la interacción entre los participantes y que la experiencia de curso sea atractiva mediante contenidos atractivos. Además, es necesario preparar contenido practico y enfocado a realizar actividades determinadas siguiendo la definición de la secuencia del curso. Para lo cual se sugiere tomar el siguiente formato elaborado por el autor (Tabla 4).

Tabla 4: Estructura del curso proceso e-learning.

Módulo	Secuencia Duración	Submódulo	Recursos
Introducción a los agentes conversacionales	Primero 1 hora		Lectura inicial, objetivos del curso, links de los chatbots mas populares.
Proyecto ALICE	Segundo 1 hora	<ul style="list-style-type: none"> • Historia • Como funciona? • Diferentes Interpretes AIML 	Lectura, video de transformación de los agentes conversacionales, Program O como interprete
Etiquetas Básicas de AIML	Tercero 1 hora	<ul style="list-style-type: none"> • Aprendizaje AIML • Hola mundo AIML 	Imágenes, Herramienta web de pruebas archivos AIML
Creación de la base del conocimiento	Cuarto 2 horas	<ul style="list-style-type: none"> • Base del conocimiento a partir de flujos. 	Foro, guía de ensamblaje de la base de conocimiento, link de pruebas archivos AIML.

Test	Quinto 1 hora		Dado un problema basado en la entidad resolverlo, y subirlo en forma de AIML
Total duración del curso = 6 horas			

2.2.5 Montaje de Cursos

En esta actividad se conduce y se plasma toda la estructura formulada en el punto anterior dentro del LMS. Según el material disponible que permita generar el entorno, será mas o menos difícil la carga de las actividades, contenido y recursos pedagógicos. Tomando en cuenta que el LMS escogido es “MOODLE”, se presenta la guía para crear un curso desde cero añadiendo lo necesario [24].

En la Figura 2 se detalla la estructura de los módulos y contenido montada dentro de Moodle.

The screenshot shows the Moodle course interface for "Introducción a los agentes conversacionales". The course structure is as follows:

- Proyecto ALICE** (Secciones o Módulos)
 - Video de la historia proyecto ALICE
 - Lectura
 - Program O
- Etiquetas Básicas AIML**
 - Imagenes Didacticas
 - AIML editor para pruebas
- Creación de la Base del Conocimiento (BC)**
 - CREACION DEL BC
 - Taller de creacion BC en AIML
- TEST**
 - Problema real

The sidebar on the left contains the following items:

- AIML
- Participantes
- Insignias
- Competencias
- Calificaciones
- General
- Introducción a los agentes conversacionales
- Proyecto ALICE
- Etiquetas Básicas AIML
- Creación de la Base del Conocimiento (BC)
- TEST
- Área personal
- Inicio del sitio
- Calendario

Figura 2: Montaje del curso proceso e-learning

2.2.6 Desarrollo de los Cursos

En esta actividad es realmente donde el aprendizaje se lleva a cabo. Los estudiantes comienzan a navegar por el contenido y las actividades del curso, adquiriendo la información que brinda el mismo. Se estudian las reacciones de los estudiantes, el nivel de dificultad, los tiempos de ejecución del programa de formación, lo cual servirá en la etapa de evaluación y retroalimentación del curso. Para almacenar la experiencia del estudiante se sugiere crear una encuesta voluntaria con preguntas simples y cortas, buscando entender cómo fue la experiencia del estudiante, la cual esta representada en la Tabla 5 [23].

Tabla 5: Modelo de encuesta proceso e-learning. Fuente: [23]

Pregunta	Valoración 1 -> bajo	5->alto
¿Como estuvo la calidad del material de apoyo del curso?	1()	2() 3() 4() 5()
¿El curso fue de su agrado?	1()	2() 3() 4() 5()
¿Se cumplió con el cronograma establecido del curso?	1()	2() 3() 4() 5()
¿ se le facilitó encontrar las diferentes actividades y contenido del curso?	1()	2() 3() 4() 5()
¿Volvería a tomar el curso , ¿si? ¿no? , por que?		

2.2.7 Evaluación y Retroalimentación

Esta es la actividad final del proceso de e-learning. En esta etapa se debe analizar la retroalimentación de los estudiantes que tomaron el curso, siempre guiándose por los objetivos del curso. La entidad que genere el curso debe valorar el nivel de satisfacción de los estudiantes tomando decisiones para resolver cualquier error y mejorar la calidad de los programas de formación que provee.

2.3 Desarrollo del agente conversacional (Chatbot)

En esta sección se analizará y desarrollará el chatbot con la herramienta Program O y el lenguaje AIML, además se utilizará el etiquetador de textos TreeTagger que será levantado a manera de servicio REST en la plataforma web Heroku y que servirá para mejorar la calidad de conversación humano-máquina.

2.3.1 Métodos más utilizados para desarrollar Chatbots

Para la primera parte del framework se muestran los métodos más destacados de la encuesta encontrada en el artículo [25] cuyo propósito es brindar un documento técnico comparando los lenguajes humano-máquina de acuerdo con los siguientes parámetros: **a)** Facilidad de Implementación y complejidad del lenguaje, **b)** Adquisición de conocimiento, **c)** Vínculos a ontologías⁷, y **d)** Construcción de un chatbot para una aplicación móvil.

En 1950, Alan Turing, publicó un artículo proponiendo una “Prueba de Turing” [26] tomando como criterio principal la inteligencia, es decir, define la capacidad que tiene un programa de computadora para imitar la interacción humana; lo suficiente para no ser detectado por un ser humano. ELIZA [27], es un proyecto desarrollado por Joseph Weizenbaum's capaz de engañar a los usuarios simulando una conversación natural con ellos. Los autores acordaron que la manera más óptima de facilitar la interacción humano-computador (*Human Computer Interaction, HCI*) es permitiendo a los usuarios al hablar o al escribir expresar de manera directa sus intereses, deseos o consultas. [25]

Varios programas fueron creados a raíz del proyecto ELIZA, uno de los más renombrados ALICE. Creado por William Wallace en 1995, este programa utiliza técnicas de reconocimiento y coincidencia de patrones escritas en un lenguaje de etiquetado derivado de XML⁸ llamado AIML.

⁷ Ontología: se refiere a la formulación de un riguroso esquema conceptual con el fin de facilitar la comunicación y el intercambio de información entre diferentes sistemas.

⁸ XML: “*Extensible Markup Language*” lenguaje de etiquetas creado con el fin de que se pueda entender tanto para humanos como para computadoras.

Otro proyecto creado a la par de ALICE es ChatScript [28]. Lenguaje creado para aceptar preguntas de un humano y enviar una respuesta correspondiente, este es un sistema de manipulación de lenguaje natural.

A partir de esta breve introducción se pone a consideración los dos métodos mas utilizados en la creación de chatbots o agentes conversacionales AIML y ChatScript. En la Tabla 6 se presenta la comparación realizada por los autores en [25].

Tabla 6: Comparativa métodos Chatbot Fuente: [25]

	AIML	ChatScript
Fácil de Implementar	No necesita de otras extensiones para ser implementado.	Es necesario utilizar la herramienta Flash, la cual esta obsoleta.
Complejidad del Lenguaje	Balance para mantener el lenguaje tan simple como sea posible.	Complejo sistema de agentes de manipulación del lenguaje natural y conversación.
Conexión a recursos externos	Freebase, Wikidata, DBPedia, Yago, Wordnet	WordNet, ConcepNet
Aprendizaje	Sintaxis definida para el aprendizaje de patrones y templates.	Si, con variables y hechos.
Implementación de Ontología personal	JENA library.	Si, Código Script.
Aplicación Móvil	Si	Si, pero con restricciones de versionamiento.

Enfocándose en los objetivos del presente trabajo se utilizará AIML como lenguaje base para el interprete del agente conversacional, esto debido a: **A)** facilidad de aprendizaje y **B)** simplicidad de lenguaje.

2.3.2 Etiquetador de Textos

Para la segunda parte del framework se definirá la estructura del chatbot a partir de la herramienta Program O que utiliza la versión 1.0 de AIML. El problema mas complejo de esta versión es la gran cantidad de archivos AIML que deben ser escritos manualmente para crear un chatbot funcional [29]. Problema solventado en la versión 2.0 (no soportada por Program O), creando el uso de nuevos comodines “^” y “#” que respectivamente permiten utilizar el patrón 0-N palabras desembarcando en la obtención de una base del conocimiento mas compacta y robusta.

Sin embargo, para el desarrollo de este framework se utilizará un etiquetador de textos basado en arboles de decisión. La herramienta a utilizar será TreeTagger [30], que de acuerdo con el trabajo en cuestión, obtuvo un 96.36% de precisión etiquetando con el formato “*Penn-TreeBank Data*” [31]. TreeTagger brinda un Corpus en el idioma español ya entrenado por Ancora [32], el cual será utilizado en este proyecto.

Esta herramienta proporcionará al chatbot un mayor nivel de flexibilidad de entendimiento de palabras naturales mejorando así el problema principal de AIML 1.0 antes mencionado. Esto se lograra gracias a que en el método de coincidencia de patrones [33] utilizado por AIML, solo se utilizaran los lemas de los diferentes textos escritos por los usuarios, etiquetados por la herramienta, mas no todas las conjugaciones de los tiempos del verbo que existan, resultando en un menor número de categorías y en un menor tiempo de búsqueda de las respuestas. A continuación, en la Tabla 7 se muestra un ejemplo de la utilización de la herramienta con varias conjugaciones del verbo “terminar”.

Tabla 7: Conjugaciones y Lema del verbo Terminar TreeTagger

Pronombre	Verbo	Etiqueta	Lema
Yo	termino (presente)	VLfin	terminar
Tú	terminas (presente)	VLfin	terminar
Él	termina (presente)	VLfin	terminar
Nosotros	terminamos (presente)	VLfin	terminar
Ustedes	terminan (presente)	VLfin	terminar
Ellos	terminan (presente)	VLfin	terminar

Yo	terminé (pretérito perfecto simple)	VLfin	terminar
Tú	terminaste (pretérito perfecto simple)	VLfin	terminar
Él	terminó (pretérito perfecto simple)	VLfin	terminar
Nosotros	terminamos (pretérito perfecto simple)	VLfin	terminar
Ustedes	terminaron (pretérito perfecto simple)	VLfin	terminar
Ellos	terminaron (pretérito perfecto simple)	VLfin	terminar
No aplica	terminado (participio)	VLadj	terminar
No aplica	Terminando (gerundio)	VLger	terminar

Finalmente, TreeTagger posee un “*wrapper*” [12], desarrollado en Python y distribuido bajo la licencia de código libre, el cual permite utilizar la función de etiquetar textos. Por motivos de solución se almacenará esta herramienta en un servicio REST en la nube, el cual reciba el texto escrito por el usuario y devuelva los lemas de cada palabra en ese texto. El servicio utiliza el método POST [34] que ejecuta la función de TreeTagger y devuelve un objeto JSON [35].

2.3.3 Estructura del Chatbot

Finalmente, en esta sección se describirá la funcionalidad del chatbot, tomando en cuenta la herramienta Program O. Esta herramienta usa un algoritmo de coincidencia de patrones para encontrar la mejor opción de respuesta dentro de la BC⁹. Para que Program O pueda realizar el proceso de interpretar lenguaje natural y devolver una respuesta simulando a una persona, necesita cumplir con lo siguientes pasos [36]:

- **Preprocesamiento:** fase en la que se cargan los archivos AIML escritos en categorías. Estas pasan por un proceso de transformación de todas las palabras en minúsculas. También es necesario que pasen por la eliminación de símbolos de escritura natural como signos de pregunta, interrogación o puntuación.
- **Parseo de archivos AIML:** en esta fase se dividen las palabras de las diferentes categorías en una estructura de datos, haciendo mas fácil la lectura de los datos para el interprete.

⁹ BC: Base del conocimiento de un agente conversacional.

- **Analizador de patrones:** en esta fase final, el interprete analiza las posibles coincidencias entre lo escrito por el usuario y lo escrito dentro la BC para brindar la mejor respuesta.

Para la fase de análisis de patrones, los interpretes AIML intentan hacer coincidir palabra por palabra hasta obtener la coincidencia de patrón más larga, ya que esta suele ser la mejor. Este comportamiento se puede describir en términos del Graphmaster, que es un conjunto de archivos y directorios. Tiene un conjunto de nodos llamados mapas y ramas que representan las primeras palabras de todos los patrones y símbolos comodín.

Suponiendo que la entrada del usuario comienza con la palabra X y que la raíz de esta estructura de árbol es una carpeta del sistema de archivos que contiene todos los patrones y plantillas; el algoritmo de coincidencia de patrones utiliza técnicas de búsqueda en profundidad. Esta técnica basada en el proyecto ALICE/AIML [37] se explica a continuación:

- Si la carpeta tiene una subcarpeta que comienza con un puntaje bajo, luego cambie a “_ /”, escanee a través de ella para capturar todas las palabras con el sufijo X; si no coincide entonces;
- Regrese a la carpeta, intente encontrar una subcarpeta que comience con la palabra X, si es así, cambie a "X /", busque la coincidencia de la cola de X; si no hay coincidencias, entonces;
- Regrese a la carpeta, intente encontrar una subcarpeta de inicio con notación de estrellas, si es así, cambie a "* /", intente todos los sufijos restantes de entrada después de "X" para ver si hay una coincidencia. Si no se encuentra ninguna coincidencia, vuelva a cambiar el directorio al padre de esta carpeta y vuelva a colocar "X" en el encabezado de la entrada.
- Cuando se encuentra una coincidencia, el proceso se detiene y el intérprete procesa la plantilla o template que pertenece a esa categoría para construir la salida.

De acuerdo con esta técnica, Program O posee una lista de puntuación para cada carpeta dividida en dos partes: a) coincidencia de frases como tal y b) coincidencia de palabras. Permite escoger los mejores resultados y si hubiera el caso que dos respuestas tienen una misma puntuación este devuelve la ultima cargada en memoria. A continuación la descripción de los puntajes en la Tabla 8:

Tabla 8: Puntuación Program O. Fuente: [10]

	patrón	puntaje
frase	underscore	100
	topic underscore	80
	topic direct	50
	topic star	10
	that underscore	45
	that direct	15
	that star	2
	direct pattern	10
	pattern direct	7
	star	1
	default	5
	palabra	direct
that direct		2
underscore		25
that underscore		25
star		1
that star		1
reject		-1000

2.3.4 Determinación de la Base del Conocimiento (BC)

La unidad mas básica de la BC dentro del contexto AIML es la categoría, que se encuentra definida por la estructura de formato patrón-plantilla ("*pattern-template*") es decir una estructura basada en reglas [38]. A demás esta basada también en la metodología de coincidencia de patrones.

Existen varias técnicas para escribir la base del conocimiento de un agente conversacional con la etiqueta <srai> [39]:

- Reducción Simbólica
- Divide y Vencerás
- Resolución de Sinónimos
- Detección de palabras

AIML posee tres tipos de clases de categorías, según [40]:

- **Atómicas:** son las categorías que se generan explícitamente cuando existe una coincidencia total con el patrón sin comodines utilizados. Ejemplo:

```
<category>
    <pattern>Hola</pattern>
    <template>Saludos humano</template>
</category>
```

- **Por defecto:** son las categorías que se ejecutan cuando no existe alguna coincidencia con alguna otra regla, se la crea a partir del uso del comodín ' * '. Ejemplo:

```
<category>
    <pattern>*</pattern>
    <template>No tengo respuesta para eso humano.</template>
</category>
```

- **Recursivas:** son las categorías que sirven para persuadir al usuario a ser mas específico sobre lo que busca encontrar en la conversación. Se utiliza la etiqueta <srai>, ejemplo:

```
<category>
    <pattern>Puedes decirme quien es Albert Einstein</pattern>
    <template><srai> Quien es Albert Einstein </srai></template>
</category>
```

```
<category>
    <pattern>Quien es Albert Einstein</pattern>
    <template>Es un científico</template>
</category>
```

Considerando la inexistencia de una metodología explícita para crear la BC de un chatbot en AIML, se tomará en cuenta el algoritmo presentado en [41] que define un proceso para generar la BC a partir de un glosario y un conjunto de preguntas frecuentes. Para el desarrollo del framework se utilizarán los siguientes pasos, basados en la información encontrada sobre el proceso:

- Clasificar las preguntas y respuestas.
- Identificar las reglas principales y derivadas.
- Etiquetar las preguntas.
- Depurar los archivos AIML.
- Creación de archivos de saludo, despedida, ayuda y categorías por defecto.

Estos pasos o tareas parten de un hecho: la entidad posee previamente la información que pretende cargar dentro del chatbot en un formato de cuestionario, en pocas palabras, cada pregunta es asignada a una respuesta o conjunto de respuestas. Por motivos de explicación el cuestionario a utilizar para armar la BC del agente será el de preguntas frecuentes “Estudiantes FAQ” [42] que almacenan una serie de preguntas generales de los estudiantes en la plataforma MOODLE. El cuestionario se detalla a continuación en la Tabla 9:

Tabla 9: Lista de preguntas y respuestas.

#	Pregunta	Respuesta
1	Cómo obtengo acceso a un curso?	Localiza o busca el curso deseado (Puedes elegir 'Todos los cursos...' en el bloque de 'Mis cursos') y elegir el nombre del curso.
2	Cómo cambio entre mis cursos?	Lo puedes cambiar mediante tu bloque de MIS CURSOS si esta disponible.
3	Cómo regreso a la página principal del curso?	Utiliza la Barra de navegación que está en la parte superior izquierda de la página.
4	Cómo puedo encontrar el curso?	Si no estás ya inscrito en un curso, puedes buscarlo por el nombre y descripción. Si ya estás inscrito te aparecerá en tu blog (si está disponible).
5	Por qué no tengo ningún correo y otros usuarios sí?	Pudiera ser que no te hayas suscrito a los foros que generan correos.
6	Por qué no hay botón de subir o grabar?	Puede que la tarea esté cerrada en este momento
7	Por qué no hay botón de subir o grabar?	Puede que la tarea no haya sido abierta aún

8	Por qué no hay botón de subir o grabar?	Simply ya has subido o grabado algo antes y la configuración impida que repitas las tareas
9	Por qué no puedo acceder a la plataforma?	Lo más probable es que simplemente olvidaste tu contraseña.
10	Por qué no puedo acceder a la plataforma?	Estas intentando acceder con una contraseña equivocada o la estás escribiendo incorrectamente.

2.3.4.1 Clasificar las preguntas y respuestas

Para llevar a cabo esta etapa, es necesario que las preguntas estén divididas en grupos de pronombres interrogativos con su contexto para poder obtener una categoría por cada uno de ellos. Finalmente se añade un archivo con las respuestas encontradas en ese contexto, la división se muestra en la Tabla 10:

Tabla 10: División de Pronombre Interrogativo con su Contexto.

Pronombre	Contexto
Cómo	Obtener acceso curso.
Cómo	Cambiar curso.
Cómo	Encontrar curso.
Por qué no	Tengo correos.
Por qué no	Hay botón de subir o grabar.
Por qué no	Puedo acceder a la plataforma.

Luego de haber dividido el pronombre con su contexto se procede a crear una categoría AIML por cada pronombre tomando en cuenta su contexto, así:

Categoría para la pregunta "Cómo":

<category>

<pattern>Cómo</pattern>

<template>

como? mmmm... creo que con eso puedes preguntarme como: acceder, navegar, cambiar, encontrar algún curso

</template>

</category>

Categoría para la pregunta “Por qué no”:

<category>

<pattern>Por qué no**</pattern>**

<template>

porque no... en ese contexto puedes preguntarme sobre la plataforma, correo, grabar o subir

</template>

</category>

Es decir, para cada contexto se tendrá que crear una categoría con cada respuesta respectivamente utilizando la etiqueta <that> encargada de mantener el contexto de la conversación con el chatbot en una sesión, así:

Para la pregunta “Cómo” y el contexto “Encontrar”:

<category>

<pattern>encontrar**</pattern>**

<that>

como? mmmm... creo que con eso puedes preguntarme como: acceder, navegar, cambiar, encontrar algún curso

</that>

<template>

Si no estás ya inscrito en un curso, puedes buscarlo por el nombre y descripción. Si ya estás inscrito te aparecerá en tu blog (si está disponible).

</template>

</category>

Para la pregunta “Por qué no” y el contexto “Correo”:

<category>

<pattern>correo</pattern>

<that>

porque no... en ese contexto puedes preguntarme sobre la plataforma, correo, grabar o subir

</that>

<template>

Pudiera ser que no te hayas suscrito a los foros que generan correos.

</template>

</category>

2.3.4.2 Identificar las reglas principales y derivadas

En este paso se deben reconocer las respuestas que son mas complejas, es decir, **a)** cuando se tiene una misma pregunta con varias respuestas, o **b)** si la respuesta es tan especifica que necesita un nivel de persuasión para que el usuario sea mas concreto y detalle bien lo que requiere.

Para ello, primero, se determinarán las respuestas que tengan una misma pregunta y se resolverán con la etiqueta <random>. Etiqueta encargada de devolver cualquiera de las opciones inscritas en el template de una regla, de esta manera:

<category>

<pattern>Por qué no hay botón de subir o grabar?</pattern>

<template>

<random>

**** Puede que la tarea esté cerrada en este momento ****

**** Puede que la tarea no haya sido abierta aún ****

**** Simplemente ya has subido algo antes y la configuración impida que repitas la tarea ****

</random>

</template>

</category>

Lo siguiente que se realizara, es escribir tal cual las preguntas y respuestas en forma de archivos AIML. A estas reglas se las llama principales ya que de estas derivan sus complementarias, obviando las preguntas que tienen mas de una respuesta anteriormente descritas, así:

<category>

<pattern> Cómo obtengo acceso a un curso? **</pattern>**

<template>

Localiza o busca el curso deseado (Puedes elegir 'Todos los cursos...' en el bloque de 'Mis cursos') y elegir el nombre del curso.

</template>

</category>

Adicionalmente, se deben determinar las respuestas que son mas complejas, del tipo, “si – entonces”, a estas reglas las llamaremos reglas derivadas, las cuales se resolverán con la etiqueta <that>. Etiqueta que permitirá resolver este tipo de preguntas dándoles un contexto, de la siguiente manera:

** Para ello se tomará en cuenta la respuesta 4 ubicada en la Tabla 9.

Caso de prueba: ***“Si no estás ya inscrito en un curso, puedes buscarlo por el nombre y descripción. Si ya estás inscrito te aparecerá en tu blog (si está disponible).”***

La respuesta se divide en la parte “SI” o premisa y la parte “ENTONCES” o conclusión, de esta manera:

- **Premisa:** ¿Estas inscrito en algún curso?
- **Conclusión (afirmativa):** te aparecerá en tu blog (si esta disponible).
- **Conclusión (negativa):** puedes buscarlo por el nombre y descripción.

Quedando la categoría AIML principal para la respuesta dividida en 2 diferentes categorías derivadas, una para cada conclusión, afirmativa o negativa, así:

Categoría Principal

```
<category>
  <pattern> Cómo puedo encontrar el curso? </pattern>
  <template>
    ¿Estas inscrito en ese curso?
  </template>
</category>
```

Categoría Afirmativa

```
<category>
  <pattern> SI </pattern>
  <that> estas inscrito en ese curso? </that>
  <template> te aparecerá en tu blog (si esta disponible). </template>
</category>
```

Categoría Negativa

```
<category>
  <pattern> NO </pattern>
  <that> estas inscrito en ese curso? </that>
  <template> puedes buscarlo por el nombre y descripción. </template>
</category>
```

2.3.4.3 Etiquetar las preguntas

Este proceso resulta primordial, debido a que la solución planteada para resolver las diferentes conjugaciones de tiempos de los verbos que se tomó es la de etiquetar las palabras para encontrar su lema, es decir, encontrar su forma canónica (base) [43]. Aplicando el patrón a la Tabla 9 para buscar las coincidencias posibles, queda descrita y etiquetada en la Tabla 11:

Tabla 11: Lista de preguntas etiquetadas y respuestas.

#	Pregunta etiquetada	Respuesta
1	cómo obtener acceso a un curso?	Localiza o busca el curso deseado (Puedes elegir 'Todos los cursos...' en el bloque de 'Mis cursos') y elegir el nombre del curso.
2	cómo cambio entre mi curso?	Lo puedes cambiar mediante tu bloque de MIS CURSOS si esta disponible.
3	cómo regreso a el página principal del curso?	Utiliza la Barra de navegación que está en la parte superior izquierda de la página.
4	cómo poder encontrar el curso?	Si no estás ya inscrito en un curso, puedes buscarlo por el nombre y descripción. Si ya estás inscrito te aparecerá en tu blog (si está disponible).
5	por qué no tener ninguno correo y otro usuario sí?	Pudiera ser que no te hayas suscrito a los foros que generan correos.
6	por qué no haber botón de subir o grabar?	Puede que la tarea esté cerrada en este momento
7	por qué no haber botón de subir o grabar?	Puede que la tarea no haya sido abierta aún
8	por qué no haber botón de subir o grabar?	Simplemente ya has subido o grabado algo antes y la configuración impida que repitas las tareas
9	por qué no poder acceder a el plataforma?	Lo más probable es que simplemente olvidaste tu contraseña.
10	por qué no poder acceder a el plataforma?	Estas intentando acceder con una contraseña equivocada o la estés escribiendo incorrectamente.

2.3.4.4 Depurar archivos AIML

En este punto casi se ha generado por completo la base del conocimiento a partir del conjunto de preguntas y respuestas. Para que la conversación con el chatbot sea lo mas natural posible, se deben sustituir las palabras como: pronombres, sustantivos posesivos o adjetivos cuantitativos por comodines “_” o “*”, esto con el objetivo de dejar solo las palabras que mas identifiquen un contexto. Además, eliminar los signos e interrogación y exclamación, de la siguiente manera expresada en la Tabla 12:

Tabla 12: Lista de preguntas etiquetadas y depuradas con sus respectivas respuestas.

#	Pregunta etiquetada y depurada (forma canónica)	Respuesta
1	cómo obtener acceso _ curso	Localiza o busca el curso deseado (Puedes elegir 'Todos los cursos...' en el bloque de 'Mis cursos') y elegir el nombre del curso.
2	cómo cambio _ curso	Lo puedes cambiar mediante tu bloque de MIS CURSOS si esta disponible.
3	cómo regreso _ página principal _ curso?	Utiliza la Barra de navegación que está en la parte superior izquierda de la página.
4	cómo poder encontrar _ curso	Si no estás ya inscrito en un curso, puedes buscarlo por el nombre y descripción. Si ya estás inscrito te aparecerá en tu blog (si está disponible).
5	por qué no tener ninguno correo _	Pudiera ser que no te hayas suscrito a los foros que generan correos.
6	por qué no haber botón _ subir _ grabar	Puede que la tarea esté cerrada en este momento
7	por qué no haber botón _ subir _ grabar	Puede que la tarea no haya sido abierta aún
8	por qué no haber botón _ subir _ grabar	Simplemente ya has subido o grabado algo antes y la configuración impida que repitas las tareas
9	por qué no poder acceder _ plataforma	Lo más probable es que simplemente olvidaste tu contraseña.
10	por qué no poder acceder _ plataforma	Estas intentando acceder con una contraseña equivocada o la estás escribiendo incorrectamente.

Añadiendo, debido a que en español se tienen palabras con tildes o diéresis y AIML 1.0 las detecta como palabras diferentes si tiene o no tilde; o cualquier otro símbolo. Para solucionar este inconveniente se deberán generar dos archivos AIML, uno por cada archivo que contemple la BC del chatbot, es decir, un archivo para palabras con tilde y otro para palabras sin tilde. Todo esto utilizando la etiqueta <srai> que nos permite devolver una misma conclusión para diferentes premisas, en este caso sea cual sea la premisa si la pregunta tenga o no la tilde se llegue a una misma conclusión.

Finalmente, en este paso, y en virtud de que AIML 1.0 funciona con comodines en la forma 1-N se utilizara la estrategia de resolución “*Keyword Detection*”. En el presente trabajo mejorada ya que en ves de detectar palabras se detectarán frases completas, por lo cual se deben crear

3 archivos por cada archivo de reglas: el primero con los comodines al inicio y al final de la frase, el segundo solo poniendo el comodín al final y el tercero solo poniendo el comodín al principio.

Esto detallado a continuación:

Categoría con comodines al inicio y al final

<category>

<pattern> _ cómo cambio _ curso _</pattern>

<template>

Lo puedes cambiar mediante tu bloque de MIS CURSOS si esta disponible.

</template>

</category>

Categoría con comodines al inicio

<category>

<pattern> _ cómo cambio _ curso </pattern>

<template>

Lo puedes cambiar mediante tu bloque de MIS CURSOS si esta disponible.

</template>

</category>

Categoría con comodines al final

<category>

<pattern> cómo cambio _ curso _</pattern>

<template>

Lo puedes cambiar mediante tu bloque de MIS CURSOS si esta disponible.

</template>

</category>

2.3.4.5 Creación de archivos de saludo, despedida, ayuda y categorías por defecto

En esta fase se crean los archivos AIML que mejoran la interacción humano-computador, simulando una conversación natural entre los dos entes. La herramienta Program O brinda la posibilidad de utilizar dos etiquetas, que serán de mucha ayuda al momento de crear las categorías por defecto, es decir cuando el chatbot no sepa que responder estas se lanzaran. Las etiquetas <google> y <wiki> permiten buscar resultados de la información no encontrada en la BC tanto dentro de la plataforma Wikipedia como en la plataforma Google. Detallados a continuación:

Categoría de saludo

```
<category>
  <pattern> Hola</pattern>
  <template>Saludos, en que puedo ayudarte?</template>
</category>
```

Categoría de despedida

```
<category>
  <pattern> Chao </pattern>
  <template>Un placer ayudarte.</template>
</category>
```

Categoría por defecto

```
<category>
  <pattern> * </pattern>
  <template>
    <random>
      <li>esto encontré en la Wikipedia: <wiki><star/></wiki></li>
      <li>esto encontré googleando: <google><star/></google></li>
    </random>
  </template>
</category>
```

El objetivo del archivo de ayuda es que el usuario pueda entender que tipo de preguntas el chatbot puede resolver, aquí se puede explicar brevemente como preguntar tomando en cuenta la sección 2.3.3.1 de este trabajo. Las etiquetas HTML básicas [44] pueden ser utilizadas aquí para dar un formato a la respuesta del chatbot.

Categoría de ayuda

<category>

<pattern> AYUDA </pattern>

<template>

<h4>Hola amigo!</h4>

Puedo ayudarte con preguntas de: **
</br>**

por que no puedo acceder a mi plataforma?**
</br>**

por que no tengo algun correo del sistema?**
</br>**

como obtener acceso a mis cursos?**
</br>**

como navegar entre mis cursos?**
</br>**

como puedo encontrar mis cursos?**
</br>
</br>**

O puedes escribir -> como **
</br>**

para ayudarte con algunas cosas de tus cursos en MOODLE**
</br>
</br>**

Ahora dime que puedo hacer por ti?

</template>

</category>

2.4 Integración del framework

Aquí se definirán los procesos y técnicas para integrar el chatbot con la interfaz web de Moodle dentro del plugin “mod/chat”. Para el consumo de los servicios creados, API chatbot de Program O y el del etiquetador de textos, se utilizará AJAX como el gestor de promesas en JavaScript modificando los archivos que forman parte del código principal de este complemento.

2.4.1 Arquitectura del framework

La arquitectura del framework es la misma arquitectura de Moodle [45], esto en función de que simplemente se esta modificando el complemento de chat. Al igual que la mayoría de los proyectos de código libre se enfocan en un núcleo de aplicación principal agregando una serie de funcionalidades ordenadas mas especificas llamadas complementos. Estos se comunican con el núcleo o Core de MOODLE mediante APIs con el método conocido como “*Transaction Script*” [46]. El uso de esta arquitectura es natural de una aplicación escrita en PHP.

La base de datos de MOODLE esta definida por alrededor de 250 tablas, las cuales se encuentran bien entendidas gracias a que la mayoría son complementos, esta estructura esta definida en el archivo “install.xml” descrito por cada uno de estos. Para el desarrollo del framework en cuestión se modificará este archivo para almacenar ciertos datos necesarios para la interacción de MOODLE con el chatbot como: la url del api, el nombre del agente conversacional, la sesión, entre otros.

Físicamente un complemento de MOODLE es solo una carpeta de scripts PHP, JavaScript, CSS, entre otros. Estos brindan estilos y comportamiento de las interfaces. Moodle Core se comunica con el complemento buscando puntos de entrada particulares, a menudo definidos en el archivo “lib.php” dentro del complemento.

Según [45] MOODLE adopto la librería de interfaces JavaScript de YAHOO, lo cual permite agregar fácilmente nuevas funcionalidades JavaScript a cada complemento.

A demás para el desarrollo del framework se utilizará el complemento de chat de MOODLE ubicado en la carpeta “mod/chat/gui_ajax” del código fuente de este. En esta carpeta se encuentran los archivos necesarios para modificar el funcionamiento de este módulo. En los archivos “index.php” y “module.js” se encuentran las funcionalidades y comportamiento del complemento. Archivos los cuales han sido modificados de su versión para cumplir con el objetivo de este trabajo, cambios definidos en la sección 2.4.2 y 2.4.3 de este capitulo.

Como se observa en la Figura 3, se encuentra detallada la arquitectura del framework a nivel de aplicación en forma de cliente-servidor [47] en el cual el usuario (cliente) envía la petición JavaScript en forma AJAX a Program O (servidor), esperando un tiempo prudente hasta que se resuelvan las peticiones y el servidor envié una respuesta, satisfactoria o no, nuevamente

al cliente. Todo este proceso se lo realiza de manera asíncrona, es decir, en la vista no se vera reflejado algún cambio hasta que se resuelvan todas las promesas [48], enviadas al servidor.

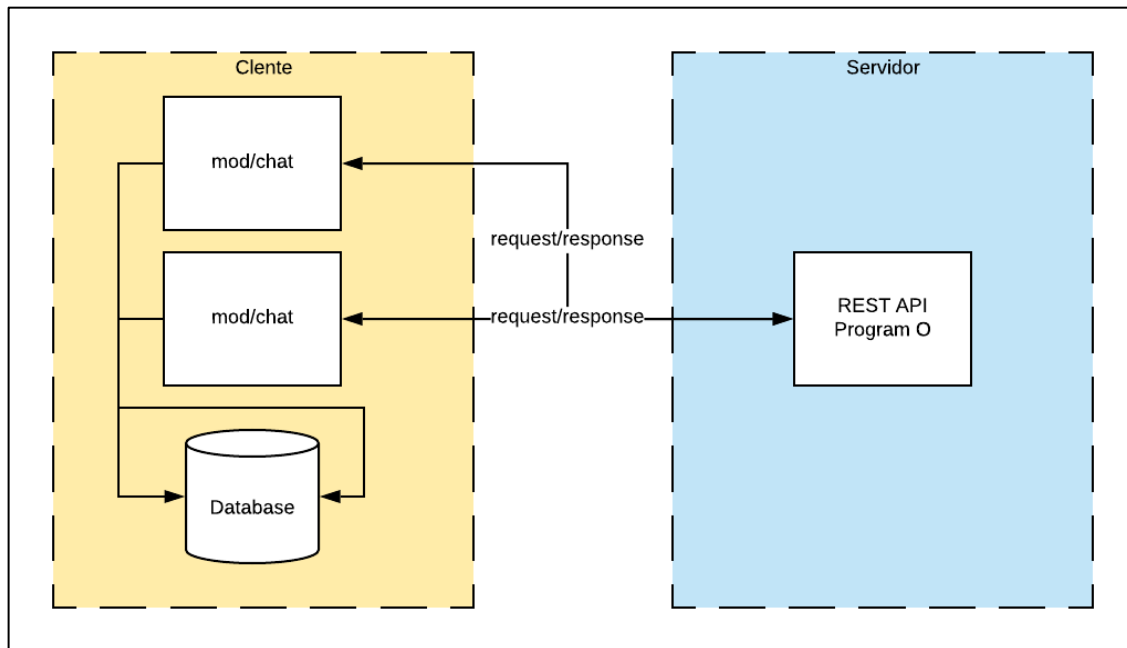


Figura 3: Modelo arquitectura cliente-servidor del framework

En el desarrollo web existen dos apartados que se deben diferenciar según [49]: el “*Backend*” y “*Frontend*” descritos a continuación.

2.4.2 Backend

El Backend es la parte mas funcional y menos visual de una aplicación web, aquí se desarrolla todo el comportamiento que la aplicación web tendrá. Las conexiones a servicios externos para agregar nuevas funciones como la conexión a la base de datos, gestión de usuarios, entre otros.

Los cambios realizados en el Backend del complemento de chat de MOODLE están descritos a continuación:

- Se agregaron tres campos a ser almacenados en la base de datos: “botdir”, “botname” y “botid”, los cuales fueron agregados en el archivo “../db/install.xml” en la tabla “chat” que servirán para almacenar el nombre del chatbot, la dirección de la API y su id. Estos datos se encuentran al crear un bot dentro de Program O en la sección de administrador.
- Se añadieron dos funciones JavaScript en el archivo “module.js” para que puedan consumir el servicio mediante AJAX tanto de la API de Program O como del servicio de etiquetar textos.
- Además se agregó la funcionalidad en el método “send” del archivo “module.js” el cual se encarga de almacenar las conversaciones dentro de las sesiones de un chat, esta funcionalidad agregada fue la de si el usuario digita el patrón “@botname” se hará el llamado en este orden: primero al etiquetador de textos el cual devuelve la forma canónica de la pregunta y segundo al servicio de chatbot.

2.4.3 Frontend

El Frontend es la parte mas visual de una aplicación web ya que interactúa directamente con los usuarios, así que debe ser lo mas intuitiva posible. La modificación se encuentra descrita a continuación:

- Se agregó un link con el nombre “Chatbot” el cual lanza una alerta con las indicaciones de como utilizar el chat, quedando así en la Figura 4:

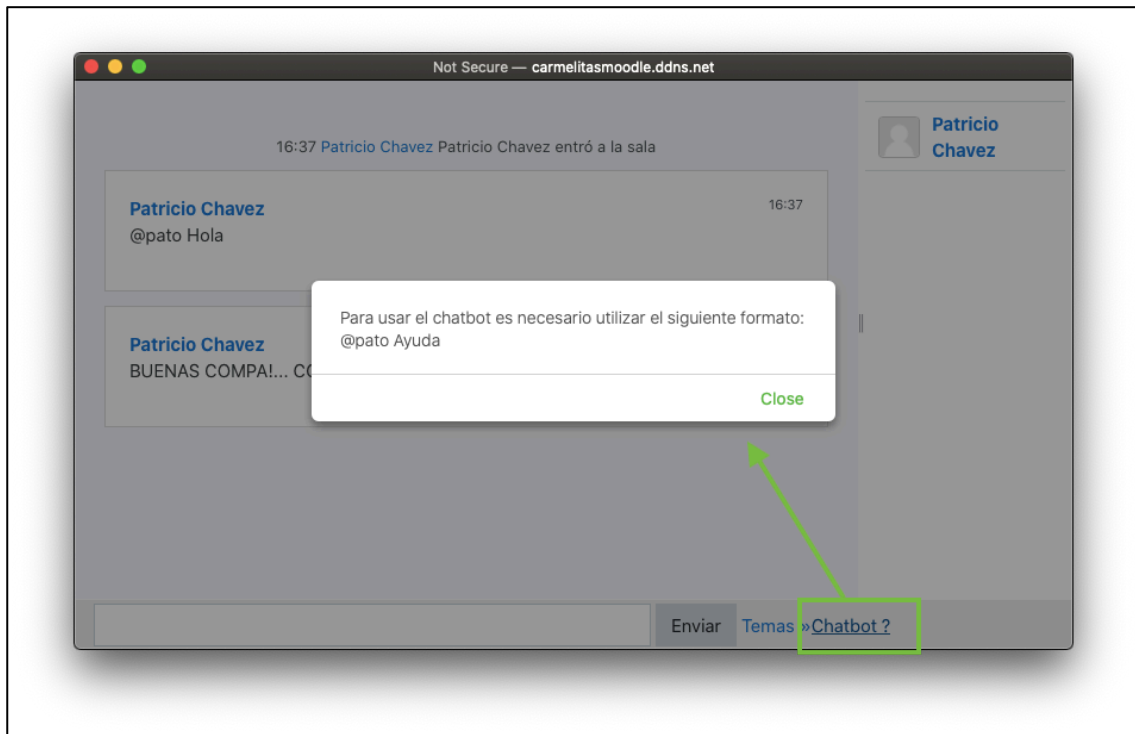


Figura 4: Vista del Chat modificado

2.4.4 Estructura del framework

En la Figura 5 se muestra de manera clara el funcionamiento del framework: **a)** como se realizan las peticiones y respuestas entre los componentes del Framework, **b)** como se consume servicio de etiquetar textos, **c)** la API de Program O y **d)** el complemento de chat de MOODLE.

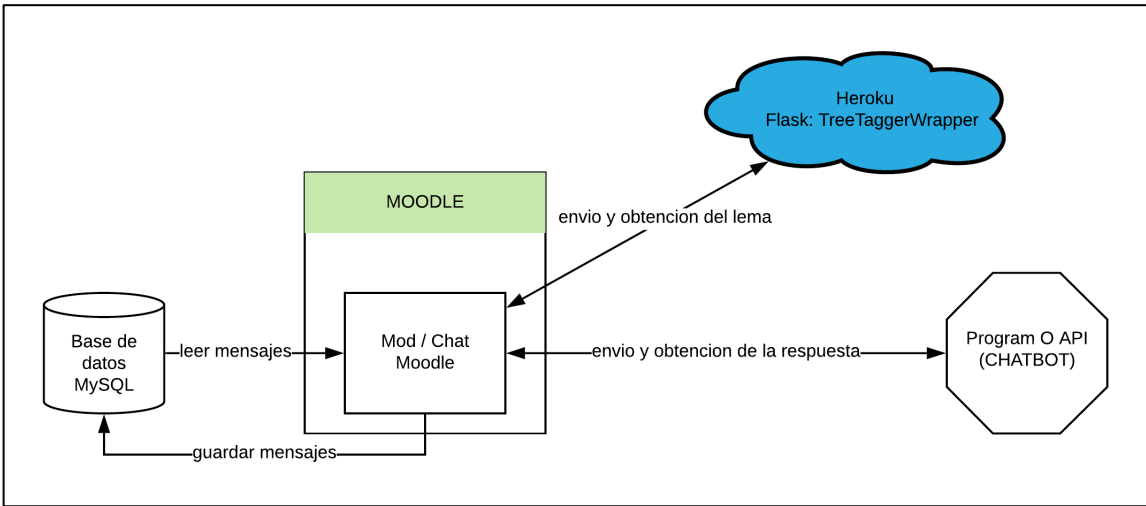


Figura 5: Estructura del framework

3. RESULTADOS Y DISCUSIÓN

En esta sección se ocuparán las mismas preguntas utilizadas en el capítulo 2, Tabla 9. Como escenario de prueba para así demostrar los resultados obtenidos luego de aplicar la técnica de coincidencia de patrones ALICE/AIML expresada anteriormente en la sección 2.3.3 de este trabajo. También en esta sección se desarrollara una matriz de confusión con los datos obtenidos de las pruebas del modelo de chatbot aplicado a diferentes personas reproduciendo la metodología tomada de [50], para evaluar el rendimiento de un agente conversacional basado en el proyecto ALICE/AIML.

3.1 Resultados

De acuerdo con [50], numerosos son los métodos utilizados para evaluar un chatbot, descritos a continuación:

- **Prueba de Turing:** esta prueba se basa en la idea propuesta por Turing en la década de los 1950s, que consistía en poner una persona frente a una maquina sin que esta supiera, y preguntarle a la persona, si creía que estaba hablando con una maquina o un ser humano.
- **Caja negra y Caja blanca:** las pruebas de caja blanca involucran el análisis de cada respuesta del chatbot a cada pregunta del usuario, mientras que las pruebas de caja negra se enfocan en evaluar todo el chatbot como tal: la satisfacción del usuario, tiempo de respuesta de una pregunta, la exactitud de la información obtenida por el usuario. Aplicando los dos tipos de pruebas se obtendrá una evaluación mas efectiva de un agente conversacional.
- **Métodos de evaluación algorítmica:** estas pruebas algorítmicas marcan el rendimiento de un chatbot dependiendo de un valor numérico generado por probabilidades, métodos como ROUGE o METEOR [51], considerados métodos no efectivos según [52].

Por motivos de efectividad se ha escogido el segundo método, caja blanca y caja negra tomando en cuenta los tiempos de respuesta a varias de las preguntas generadas en el capítulo anterior de este trabajo.

Para entender el funcionamiento del algoritmo de coincidencia de patrones del proyecto ALICE/AIML específicamente hablando del interprete Program O, el cual funciona mediante puntajes siendo los de mayor grado de coincidencia los que tienen mayores puntajes. Se harán varias preguntas al chatbot y utilizando el archivo de Debug que este nos proporciona verificar los diferentes puntajes alcanzados por el chatbot después de analizar las mejores respuestas dentro de la BC.

Los resultados mostrados a continuación están divididos en dos casos, el primero utilizando la forma canónica de las preguntas (forma base) con el etiquetador de textos llamado ESCENARIO 1, y el segundo utilizando solo el algoritmo de coincidencia de patrones llamado ESCENARIO 2, para lo cual se crearan dos bots que almacenaran las preguntas, en la forma indicada en cada escenario, de la Tabla 9 anteriormente expuesta.

En estas pruebas se tomaran en cuenta el score o puntaje que alcancen y el tiempo que se demora en la función “*score_matches()*”. Esta función va asignando los puntajes de acuerdo con los patrones encontrados en la lista de puntajes de la Tabla 8.

3.1.1 ESCENARIO 1

Para este contexto se utilizarán las preguntas en su forma canónica como la base del conocimiento, descritas anteriormente por [43].

Tabla 13: BC del Escenario 1

Pregunta almacenada en la BC para el escenario 1	Respuesta
cómo obtener acceso _ curso _	Localiza o busca el curso deseado (Puedes elegir 'Todos los cursos...' en el bloque de 'Mis cursos') y elegir el nombre del curso.
cómo cambio _ curso	Lo puedes cambiar mediante tu bloque de MIS CURSOS si esta disponible.

cómo regreso _ página principal _ curso	Utiliza la Barra de navegación que está en la parte superior izquierda de la página.
cómo poder encontrar _ curso _	Si no estás ya inscrito en un curso, puedes buscarlo por el nombre y descripción. Si ya estás inscrito te aparecerá en tu blog (si está disponible).
por qué no tener ninguno correo _	Pudiera ser que no te hayas suscrito a los foros que generan correos.
por qué no haber botón _ subir _ grabar	Puede que la tarea esté cerrada en este momento
por qué no haber botón _ subir _ grabar	Puede que la tarea no haya sido abierta aún
por qué no haber botón _ subir _ grabar	Simplemente ya has subido o grabado algo antes y la configuración impida que repitas las tareas
por qué no poder acceder _ plataforma	Lo más probable es que simplemente olvidaste tu contraseña.
por qué no poder acceder _ plataforma	Estas intentando acceder con una contraseña equivocada o la estás escribiendo incorrectamente.
*	Respuesta por default.

A partir de las preguntas del escenario 1 expresadas en la Tabla 13, se generaron varias preguntas de prueba las cuales están realizadas en varios tiempos verbales [53], indicadas en la Tabla 14 identificando los tiempos y puntajes encontrados.

Tabla 14: Resultados Escenario 1

Pregunta realizada al chatbot	Score Program O	Tiempo (milis)
Cómo puedo encontrar mis cursos dentro de mi plataforma?	159	0.566
Cómo puedo encontrar el curso de ingles?	159	0.798
Cómo obtengo acceso a mi curso en Moodle?	159	0.560
Por qué no hubo el botón de subir o grabar?	2	0.354
Por qué no pude acceder a mi plataforma?	138	0.523
Por qué no puedo acceder a la plataforma?	138	0.556
Por qué no podía acceder a la plataforma?	138	0.502

3.1.2 ESCENARIO 2

Para este escenario se utilizara la siguiente BC teniendo que aumentar varias categorías diferentes al las del escenario 1 para que el chatbot responda con una plantilla diferente de la por defecto, descrita en la Tabla 15 a continuación:

Tabla 15: BC del Escenario 2

Pregunta almacenada en la BC para el escenario 2	Respuesta
cómo obtener acceso _ curso _	Localiza o busca el curso deseado (Puedes elegir 'Todos los cursos...' en el bloque de 'Mis cursos') y elegir el nombre del curso.
cómo obtengo acceso _ curso _	Localiza o busca el curso deseado (Puedes elegir 'Todos los cursos...' en el bloque de 'Mis cursos') y elegir el nombre del curso.
cómo obtengo acceso _ cursos _	Localiza o busca el curso deseado (Puedes elegir 'Todos los cursos...' en el bloque de 'Mis cursos') y elegir el nombre del curso.
cómo cambio _ curso	Lo puedes cambiar mediante tu bloque de MIS CURSOS si esta disponible.
cómo regreso _ página principal _ curso	Utiliza la Barra de navegación que está en la parte superior izquierda de la página.
cómo poder encontrar _ curso _	Si no estás ya inscrito en un curso, puedes buscarlo por el nombre y descripción. Si ya estás inscrito te aparecerá en tu blog (si está disponible).
cómo puedo encontrar _ curso _	Si no estás ya inscrito en un curso, puedes buscarlo por el nombre y descripción. Si ya estás inscrito te aparecerá en tu blog (si está disponible).
por qué no tener ninguno correo _	Pudiera ser que no te hayas suscrito a los foros que generan correos.
por qué no haber botón _ subir _ grabar	Puede que la tarea esté cerrada en este momento
por qué no haber botón _ subir _ grabar	Puede que la tarea no haya sido abierta aún
por qué no haber botón _ subir _ grabar	Simplemente ya has subido o grabado algo antes y la configuración impida que repitas las tareas
por qué no hubo botón _ subir _ grabar	Simplemente ya has subido o grabado algo antes y la configuración impida que repitas las tareas
por qué no poder acceder _ plataforma	Lo más probable es que simplemente olvidaste tu contraseña.
por qué no poder acceder _ plataforma	Estas intentando acceder con una contraseña equivocada o la estés escribiendo incorrectamente.

por qué no pude acceder a la plataforma	Estas intentando acceder con una contraseña equivocada o la estás escribiendo incorrectamente.
por qué no puedo acceder a la plataforma	Estas intentando acceder con una contraseña equivocada o la estás escribiendo incorrectamente.
por qué no podía acceder a la plataforma	Estas intentando acceder con una contraseña equivocada o la estás escribiendo incorrectamente.
*	Respuesta por default.

Tomando en cuenta la BC detallada en la Tabla 15. Los resultados de la evaluación del algoritmo de coincidencia de patrones para el escenario 2 con las mismas preguntas realizadas en el escenario 1, fueron:

Tabla 16: Resultados Escenario 2

Pregunta realizada al chatbot	Score Program O	Tiempo (milis)
Cómo puedo encontrar mis cursos dentro de mi plataforma?	159	0.625
Cómo puedo encontrar el curso de inglés?	159	0.890
Cómo obtengo acceso a mi curso en Moodle?	159	0.670
Por qué no hubo el botón de subir o grabar?	2	0.401
Por qué no pude acceder a mi plataforma?	138	0.630
Por qué no puedo acceder a la plataforma?	138	0.702
Por qué no podía acceder a la plataforma?	138	0.544

Enfocándose en los resultados obtenidos luego de poner a prueba el chatbot con las preguntas expuestas tanto en el escenario 1 como en el 2, se sacará un promedio del tiempo que se demora en los dos escenarios. Tomando en cuenta que los scores obtenidos son los mismos ya que las preguntas en los dos escenarios fueron las mismas.

Quedando los resultados detallados así, mismos que serán discutidos posteriormente en la siguiente sección:

Tabla 17: Promedio de tiempo por escenario

	Número de categorías AIML	Tiempo Promedio (milis)
Escenario 1	8	0.551
Escenario 2	15	0.637

A simple vista en la Tabla 17 existe una pequeña diferencia en el tiempo de respuesta, pero si se lo lleva a gran escala se puede convertir en un problema de no satisfacción de los usuarios.

Finalmente, para concluir con la sección de resultados, se hicieron las correspondientes pruebas dirigidas a personas. Esta prueba consistía en que cada persona pueda interactuar directamente con el chatbot en un par de minutos con su propia sesión en MOODLE, luego de que la persona acabara con la prueba, simplemente los datos de la conversación quedan almacenados en el histórico de cada sesión, a continuación se procede a copiar estos datos y luego de una depuración rápida (eliminación de filas vacías) se procedió a clasificar de acuerdo con los casos de la tabla en [50], descrita a continuación:

Tabla 18: Tipos de Respuesta del chatbot.

FALSOS POSITIVOS (FP)	El chatbot respondió incorrectamente a una pregunta correcta del usuario.
FALSOS NEGATIVOS (FN)	El chatbot respondió incorrectamente a una pregunta incorrecta del usuario.
VERDADEROS POSITIVOS (VP)	El chatbot respondió correctamente a una pregunta correcta del usuario.
VERDADEROS NEGATIVOS (VN)	El chatbot respondió correctamente a una pregunta incorrecta del usuario.

Con los datos tabulados se procede a calcular el valor de las medidas de Sensibilidad [50], Exactitud [50], Precisión [54] y la Medida-F [50]. que permitirán tener una idea clara de cual fue el rendimiento del chatbot frente a las pruebas de cada usuario.

Las formulas de las diferentes métricas son :

$$Exactitud = \frac{VP}{VP+FP} \quad (1)$$

$$Sensibilidad = \frac{VP}{VP+FN} \quad (2)$$

$$MedidaF(\beta) = \frac{1}{\frac{\beta}{exactitud} + \frac{1-\beta}{sensibilidad}} \quad (3)$$

$$Precision = \frac{VP+FN}{VP+FP+VN+FN} \quad (4)$$

- (1) Se refiere al valor del grado de concordancia de la relación entre lo medido con lo real.
- (2) Se refiere al valor mínimo de medición de un instrumento de medida, en este caso el modelo presentado como solución a la problemática planteada.
- (3) Se refiere al valor combinado entre sensibilidad y exactitud que sirve para tener una mejor visión del rendimiento del modelo.
- (4) Se refiere al valor del porcentaje de respuestas correctas realizadas por el modelo (Chatbot) frente a las preguntas realizadas.

Tomando en cuenta que β tiene un valor de 0.8 para favorecer la exactitud sobre la sensibilidad según [50]. Los resultados obtenidos con los valores de las pruebas se detallan a continuación.

		RESPUESTAS CHATBOT	
		NEGATIVO	POSITIVO
RESPUESTAS REALES	VERDADERO FALSO	27	26
	VERDADERO VERDADERO	21	140

Figura 6: Matriz de confusión de los resultados obtenidos.

Estos resultados se discuten de inmediato en la siguiente sección del presente trabajo.

3.2 Discusión

Primeramente, los resultados obtenidos por el agente conversacional luego de realizar las pruebas de respuesta calculando el tiempo que demora en los dos diferentes escenarios Tabla 17, si bien la diferencia no es tan grande, en términos de porcentaje se puede decir que utilizando la forma canónica (base) de las palabras con la ayuda del lematizador es un 14% mas rápida en la capacidad de encontrar la mejor coincidencia del patrón, que utilizando la forma natural (no canónica). Teniendo en cuenta este resultado como realmente una alternativa de solución para lo que fue creado este framework.

No obstante, los resultados de Sensibilidad, Exactitud y Precisión de la matriz de confusión son un mejor indicador para finalmente inclinarse por la solución expuesta en el actual trabajo, además se compararon estos mismos valores expuestos en [50] ,de un agente conversacional para resolver tareas específicas basado en AIML con los del trabajo en cuestión.

Tabla 19: Comparación con agente conversacional AIML en [50].

Agente conversacional	Precisión	Exactitud	Sensibilidad	Medida-F
Presente trabajo	0.780	0.843	0.870	0.848
Mostrado en [50]	0.568	0.574	0.877	0.617

De acuerdo con la Tabla 19 se encontró que el modelo de agente conversacional propuesto en el presente trabajo tuvo valores similares en cuanto a sensibilidad refiere, mientras que en los demás valores de precisión, exactitud y la medida-F obtuvo un 22%, 27% y 23% aproximadamente de ventaja, dando como resultados prometedores para futuros trabajos de investigación o de diversa índole.

4. CONCLUSIONES Y RECOMENDACIONES

Aquí se identifican y generan las conclusiones y recomendaciones del trabajo.

4.1 Conclusiones

Se logro implementar el LMS MOODLE de acuerdo con las recomendaciones de [15], permitiendo una adecuación sencilla de un curso de prueba en formato curso secuencial, que fue utilizado para las pruebas de rendimiento del agente conversacional. Este curso fue planteado con tres diferentes secciones: introducción, conocimiento (todo el material de aprendizaje) y la de prueba; que evidencian si se cumplió o no con el objetivo del curso. Toda esta implementación se beneficia por la facilidad de encontrar información a cerca de MOODLE y sus cursos, además de que este sistema en su versión actual viene incluido con funcionalidades extras como sesiones de chat, videoconferencias, almacenamiento de archivos, pruebas rápidas o quiz al final de cada sección para tener una evaluación continua de los estudiantes que accedan al sistema.

MOODLE, al poseer una gran capacidad de flexibilidad en cuanto a modificaciones de su funcionalidad mediante complementos o plugin, facilito al presente trabajo cumplir con el desarrollo y acoplamiento del agente conversacional (Chatbot). El agente utilizado optimiza los tiempos de respuesta, además de la calidad de conversación, esto gracias a que el agente traduce las preguntas a un nivel de representación mas básico (canónico) para posteriormente aplicar el algoritmo de búsqueda por patrones específico del modelo ALICE/AIML, probado y que es sujeto de varias implementaciones a nivel empresarial.

En el framework presentado del trabajo en cuestión, se ha logrado evidenciar los niveles de satisfacción de los usuarios que interactuaron con el agente conversacional dentro del ambiente de gestión del aprendizaje MOODLE, mediante las pruebas de medición realizadas al modelo, en este sentido, se obtuvo un modelo con una precisión del 78%, lo cual es muy bueno hablando en términos técnicos ya que se pueden mejorar en trabajos a futuro tomando en cuenta las recomendaciones de la siguiente sección del presente trabajo.

4.2 Recomendaciones

Para la adecuación de cursos, es necesario como cualquier otro material didáctico, generar recursos como presentaciones, imágenes, videos, links de acuerdo con las necesidades que la entidad o persona que utilicen este framework tengan. Cabe decir que a mayor facilidad de acceder a la información en un medio digital mayor será la satisfacción del usuario que lo utiliza.

En cuanto a mejorar la calidad y presentación de la información y recursos en un LMS, que se presentan en los cursos se pueden aplicar capacitaciones dirigidas a los administradores sobre las funcionalidades y desarrollos que existen dentro de la plataforma MOODLE. Tomando en cuenta además su gran ventaja que son distribuido bajo licencias libres y con información detallada. Todo esto con el objetivo de mejorar la experiencia del estudiante y profesor dentro de un LMS.

Se necesita hacer un previo análisis para definir el alcance del agente conversacional para luego plasmarlo a través de archivos AIML. Para poder generar correctamente la base del conocimiento inferido como núcleo del framework; además para que el agente conversacional sea lo mas amigable posible con el usuario es necesario complementar la raíz de la base del conocimiento con archivos que se encuentran fácilmente en la web, archivos como: saludos.AIML, sentimientos.AIML, deportes.AIML, entre otros. Que ayudan a mejorar la calidad de conversación.

Finalmente se recomienda realizar un proceso de pruebas exhaustivo para verificar el rendimiento del agente conversacional, pruebas reales con las personas que utilizaran el sistema, ya que se obtendrá un porcentaje mas real de la satisfacción de estos, así como el estado actual del agente conversacional.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] F. C. Descalzos, «Instituto de Espiritualidad Santa Teresa del Niño Jesús,» [En línea]. Available: <http://www.carmelitasdescalzosecuador.org/index.php/instituto>.
- [2] L. G. Aretio, «OBJETIVOS Y FUNCIONES DE LA EDUCACIÓN A DISTANCIA,» *Actas del Congreso Internacional de Filosofía de la Educación*, 1990.
- [3] J. L. Auquilla, Interviewee, *Problemática del Instituto*. [Entrevista]. 27 Agosto 2019.
- [4] A. F.-P. Cesteros, «Las plataformas e-learning para la enseñanza y el aprendizaje universitario en Internet,» 2019. [En línea]. Available: https://eprints.ucm.es/10682/1/capituloE_learning.pdf.
- [5] MOODLE, «Acerca de Moodle,» 26 Diciembre 2019. [En línea]. Available: https://docs.moodle.org/all/es/Acerca_de_Moodle. [Último acceso: Diciembre 2019].
- [6] MOODLE, «Configuraciones de Chat,» 10 Noviembre 2019. [En línea]. Available: https://docs.moodle.org/all/es/Configuraciones_de_chat.
- [7] APACHE, «Apache HTTP server project,» 27 octubre 2019. [En línea]. Available: <https://httpd.apache.org>.
- [8] W3SCHOOLS, «PHP MySQL Database,» 29 Noviembre 2019. [En línea]. Available: https://www.w3schools.com/php/php_mysql_intro.asp.
- [9] W. Paluszynski, «Introduction to AIML,» [En línea]. Available: https://kcir.pwr.edu.pl/~witold/ai/aie_aiml_d.pdf.
- [10] PROGRAM-O, «Program-O,» 20 Julio 2019. [En línea]. Available: <https://github.com/Program-O/Program-O>.
- [11] H. Schmid, «TreeTagger - a part-of-speech tagger for many languages,» 20 Diciembre 2019. [En línea]. Available: <https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>.
- [12] L. Pointal, «TreeTagger Python Wrapper's documentation!,» 2019. [En línea]. Available: <https://treetaggerwrapper.readthedocs.io/en/latest/>.
- [13] FLASK, «FLASK,» 2019. [En línea]. Available: <https://www.palletsprojects.com/p/flask/>.
- [14] Heroku, «What is Heroku?,» 2019. [En línea]. Available: <https://www.heroku.com/what>.
- [15] FAO United Nations, « E-learning methodologies A guide for designing and developing e-learning courses,» 2011. [En línea]. Available: <http://www.fao.org/3/i2516e/i2516e.pdf>.

- [16] D. M. d. C. Fiallos y J. F. Pagoaga Acosta, «Análisis de varias plataformas educativas (LMS) existentes en el mercado que le permitirán almacenar sus aulas virtuales,» 2019. [En línea]. Available: https://www.academia.edu/8354499/Análisis_de_varias_plataformas_educativas_LMS_existentes_en_el_mercado_que_le_permitirán_almacenar_sus_aulas_virtuales.
- [17] W3Schools, «AJAX introduction,» 2019. [En línea]. Available: https://www.w3schools.com/xml/ajax_intro.asp.
- [18] MOODLE, «Arquitectura Moodle,» 2019. [En línea]. Available: https://docs.moodle.org/all/es/Arquitectura_de_Moodle.
- [19] Dept. Ciencias de la Computacion e IA, «Acceso a Servicios REST,» 2019. [En línea]. Available: <http://www.jtech.ua.es/dadm/restringido/red/sesion02-apuntes.pdf>.
- [20] F. & B. J. & R. D. & R. E. & L. N. M. Mikic Fonte, «T-Bot and Q-Bot: A couple of AIML-based Bots for tutoring courses and evaluating students,» de *Frontiers in Education Conference*, 2008.
- [21] A. Bayan y E. Atwell, «ALICE chatbot: Trials and Outputs,» *Computación y Sistemas*, vol. 19, 2015.
- [22] Program E, «Programe E ALICE AIML,» 2019. [En línea]. Available: <http://www.nathalielawhead.com/candybox/programe-e-alice-aiml-bot-using-flash-php-actionscript-walkthrough-source-files>.
- [23] E. NARANJO, «IMPLANTACIÓN DE UNA HERRAMIENTA E-LEARNING DE CAPACITACIÓN Y EVALUACIÓN DEL PERSONAL DE SEGUROS EQUINOCCIAL S.A.,» 2011.
- [24] MOODLE, «Página principal del curso,» 2020. [En línea]. Available: https://docs.moodle.org/all/es/P%C3%A1gina_principal_del_curso.
- [25] S. Madhumitha, B. Keerthana y B. Hemalatha, «Interactive Chatbot Using AIML,» *Advanced Networking & applications (IJANA)*.
- [26] A. Turing, «COMPUTING MACHINERY AND INTELLIGENCE,» *Computing Machinery and Intelligence*, pp. 433-460, 1950.
- [27] J. Weisenbaum, «ELIZA - A computer program for the study of natural language communication between man and machine,» *Communications of ACM*, vol. 9, nº 1, 1966.
- [28] B. Wilcox, «ChatScript Basic User's Manual,» 2013. [En línea]. Available: <http://chatscript.sourceforge.net/Documentation/ChatScript%20Basic%20User%20Manual.pdf>.

- [29] S. Arsovski, I. Muniru y A. Cheok, «ANALYSIS OF THE CHATBOT OPEN SOURCE LANGUAGES AIML AND CHATSCRIPT: A Review,» n° 10.13140/RG.2.2.34160.15367, 2017.
- [30] H. Schmid, «Probabilistic Part-of-Speech Tagging Using Decision Trees,» de *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK, 1994.
- [31] A. Taylor, M. Marcus y B. Santorini, «The Penn Treebank: An overview,» 2003.
- [32] Centro de Linguística y Computación, «Corpus,» [En línea]. Available: <http://clic.ub.edu/corpus/en/documentation>.
- [33] T. Hak y J. Dul, «Pattern Matching,» *Erasmus Research Institute of Management (ERIM)*, 2009.
- [34] W3Schools, «HTTP Request Methods,» [En línea]. Available: https://www.w3schools.com/tags/ref_httpmethods.asp.
- [35] JSON org, «Introducing Json,» [En línea]. Available: <https://www.json.org/json-en.html>.
- [36] S. ROa, «Estudio y uso de chatbots para el aprendizaje de inglés,» [En línea]. Available: http://oa.upm.es/43327/3/TFG_SERGIO_RODRIGUEZ_ROA.pdf.
- [37] B. Abushawar y E. Atwell, «ALICE chatbot: Trial and outputs,» *Computación y Sistemas*, vol. 19, 2015.
- [38] J. Cahn, «CHATBOT: Architecture, Design, & Development,» 2017.
- [39] N. R. y. S. S. B. R. Ranoliya, «Chatbot for university related FAQs,» *2017 International Conference on Advances in Computing, Communications and Informatics*, pp. 1525-1530, 2017.
- [40] B. Ranoliya, N. Raghuvanshi y S. Singh, Chatbot for University Related FAQ's, pp. 1525-1530.
- [41] D. G. Giovanni, «Building an AIML Chatter Bot Knowledge-Base Starting from a FAQ and a Glossary,» *Journal of e-learning an knowledge society*, vol. 6, pp. 75-83, 2010.
- [42] MOODLE, «Estudiante FAQ,» 2019. [En línea]. Available: https://docs.moodle.org/all/es/Estudiante_FAQ.
- [43] R. Delgado y T. Kobayashi, *Proceedings of the Paralinguistic Information and its Integration in Spoken Dialogue Systems Workshop*, 2011.
- [44] B. S. Marco, «Lista de etiquetas HTML,» 2020. [En línea]. Available: <https://www.mclibre.org/consultar/htmlcss/html/html-etiquetas.html>.

- [45] MOODLE, «Moodle Architecture,» 2020. [En línea]. Available: https://docs.moodle.org/dev/Moodle_architecture.
- [46] P. Sousa, «Learning patterns of application architecture by looking at code,» *Knowledge Engineering and Decision Support Group Instituto Superior de Engenharia do Porto*, 2007.
- [47] S. Shakirat, «Client-Server Model,» *IOSR Journal of Computer Engineering*, vol. 16, pp. 57-71, 2014.
- [48] K. M. Anderson, «Asynchronous Programming in Javascript, Part 2,» 2015. [En línea]. Available: <https://www.cs.colorado.edu/~kena/classes/5828/f15/lectures/30-asyncjavascriptpromises.pdf>.
- [49] H. M. Abdullah and A. M. Zeki, «Frontend and Backend Web Technologies in Social Networking Sites: Facebook as an Example,» de *2014 3rd International Conference on Advanced Computer Science Applications and Technologies*, Amman, 2014.
- [50] N. a. S. A. a. B. G. Teckchandani, «AIML and Sequence-to-Sequence Models to Build Artificial Intelligence Chatbots: Insights from a Comparative Analysis,» 2019, pp. 323-333.
- [51] R. L. I. S. M. N. L. C. a. J. P. C. Liu, «How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation,» de *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.
- [52] S. R. T. W. a. W. Z. K. Papineni, «BLEU: a method for automatic evaluation of machine translation,» de *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002.
- [53] I. Ramirez, «Tabla de modos y tiempos verbales,» 2020. [En línea]. Available: https://www.academia.edu/11282899/Tabla_de_modos_y_tiempos_verbales.
- [54] D. M. a. J. P. T. Ryan Green, «Precision and Recall of Machine Translation,» *Computer Science Department, New York University*, 23 01 2018.

6. ANEXOS

En el CD se encuentran adjuntos los anexos detallados a continuación:

ANEXO I Entrevista realizada en el actual trabajo con sus respectivas respuestas.

ANEXO II Código fuente del framework del chatbot.

ANEXO III Manual de instalación del framework dentro de un ambiente MOODLE.

ANEXO IV Archivos con formato .aiml con las categorías procesadas por el agente conversacional del actual trabajo.

ANEXO V Archivo con formato Excel con los resultados obtenidos en las pruebas con su respectiva hoja de calculo de la matriz de confusión.

ANEXO I

Problemática del Instituto

1. ¿Cómo esta el estado actual del instituto?

Actualmente, el instituto trabaja con la modalidad de cursos semipresenciales, los cuales están dictados por voluntarios que se presentan a través del único medio de comunicación del correo electrónico, pero a pesar de esto existe un alto nivel de personas que no culminan el curso.

Además existe una modalidad de cursos, los cuales están dirigidos a voluntarios que no tienen acceso a internet al menos pasando uno o dos meses porque están realizando su trabajo de voluntariado en lugares como la selva ecuatoriana. Para estas personas es difícil tener una constante interacción con algún medio de comunicación así que para ellos no existe un límite de tiempo en el cual acabar con el curso brindado por el instituto, llegando así a tomar inclusive años para que estas personas logren culminar con su curso.

Finalmente el instituto trabaja con un libro (físico) proporcionado por una entidad española, en el cual están las tareas al final de cada capítulo, simplemente la persona se guía en este recurso y cuando necesita algo envía un correo al tutor asignado, teniendo en cuenta, esto representa una pérdida de tiempo la cual no se daría si se tuviera una plataforma en la cual almacenar toda la información de los cursos.

2. ¿Cuáles con las necesidades del instituto?

Pues se necesitan varias cosas, la primera es que exista un lugar en donde se pueda centralizar la información y sea accesible tanto para estudiantes, profesores como para los administradores. Para mejorar la calidad del acompañamiento del profesor se necesita de una herramienta que permita a los estudiantes estar constantemente actualizados sobre las tareas que tienen, y que un profesor pueda estar la mayoría del tiempo solventando dudas de sus estudiantes. Ya que normalmente en el ámbito en el que el instituto se desarrolla es en el filosófico, así que existen muchas preguntas que se necesitan estén resueltas con prontitud.

3. ¿Qué se espera al final de este proyecto?

Finalmente lo que esperamos del proyecto es poder estandarizar y llevar un control de los estudiantes que tenemos, el estado y la calidad de los recursos que se les presenta a nuestros estudiantes. Además lograr una satisfacción de nuestros estudiantes para así generar un ambiente y más personas puedan acceder a nuestros cursos.