



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**ROBOTS DE CHARLA COMO HERRAMIENTA INFORMATIVA PARA LOS ESTUDIANTES
DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Carlos Estuardo Cano López

Asesorado por el Ing. Miguel Antonio Morales Guillén

Guatemala, febrero de 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ROBOTS DE CHARLA COMO HERRAMIENTA INFORMATIVA PARA LOS ESTUDIANTES
DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

CARLOS ESTUARDO CANO LÓPEZ

ASESORADO POR EL ING. MIGUEL ANTONIO MORALES GUILLÉN

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, FEBRERO DE 2019

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Luis Diego Aguilar Ralón
VOCAL V	Br. Christian Daniel Estrada Santizo
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADORA	Inga. Virginia Victoria Tala Ayerdi
EXAMINADOR	Ing. Óscar Alejandro Paz Campos
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
SECRETARIA	Inga. Mayra Grisela Corado García (a.i.)

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**ROBOTS DE CHARLA COMO HERRAMIENTA INFORMATIVA PARA LOS
ESTUDIANTES DE LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS DE LA
FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 27 de noviembre de 2010.

Carlos Estuardo Cano López

ACTO QUE DEDICO A:

Dios	Por iluminar cada paso de mi vida y permitirme llegar hasta este momento importante.
Mis padres	Por su paciencia, comprensión y apoyo incondicional.
Mi familia	Por acompañarme en todos los momentos importantes.
Mi esposa	María René Corzo, por su apoyo incondicional, cariño y amistad.
Mi asesor	Miguel Antonio Morales Guillén, por su colaboración desinteresada para la elaboración de este trabajo de graduación.
Mis amigos	Por compartir gratos momentos y también los que no lo fueron, los cuales me han hecho crecer y mejorar.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	V
LISTA DE SÍMBOLOS	VIII
GLOSARIO	IX
RESUMEN	XIII
OBJETIVOS.....	XV
INTRODUCCIÓN	XVIII
1. MARCO TEÓRICO CONCEPTUAL	1
1.1. ¿Qué es un robot de charla?	1
1.2. ¿Cómo funciona el robot de charla?.....	2
1.2.1. Tipos de inteligencia artificial en los robots de charla.....	3
1.2.2. Los robots de charla con inteligencia artificial débil.....	3
1.2.3. Los robots de charla con inteligencia artificial fuerte	4
1.3. La prueba de Turing	5
1.4. Comportamiento de los robots de charla según el contexto.....	5
1.5. Uso de los robots de charla a través del tiempo.....	5
1.6. Robots de charla en internet.....	7
1.7. Interactuando con los robots a través del reconocimiento de voz.....	8
2. PROCEDIMIENTOS PARA DAR MANTENIMIENTO AL CÓDIGO FUENTE DEL ROBOT DE CHARLA.....	11

2.1.	Requisitos del sistema	11
2.1.1.	Lenguaje de programación.....	11
2.1.2.	Base de datos	12
2.1.3.	¿Qué es un servidor web?	13
2.1.3.1.	¿Por qué se utiliza?.....	13
2.1.4.	Sistema de control de versiones	14
2.1.4.1.	Subversión ¿Por qué utilizarlo?.....	13
2.2.	Preinstalación del robot de charla	14
2.2.1.	Sistema operativo Windows	15
2.2.2.	Sistema operativo Linux.....	16
2.2.3.	Sistema operativo Mac OS.....	17
2.3.	Preconfiguración del robot de charla.....	18
2.3.1.	Navegador predeterminado para ambiente de desarrollo.....	18
2.3.2.	Revisión de la configuración del servidor	19
2.3.3.	Obtener una copia local del código fuente	20
3.	PROCEDIMIENTOS PARA USO Y ADMINISTRACIÓN DEL ROBOT DE CHARLA	21
3.1.	Requisitos del sistema	21
3.1.1.	Windows.....	21
3.1.2.	Mac.....	22
3.1.3.	Linux.....	22
3.2.	Procedimiento para hacer uso del robot de charla	22
3.3.	Procedimiento para administrar el robot de charla	24
3.3.1.	Sección de pestañas	27
3.3.1.1.	Inicio	27
3.3.1.2.	Documentación.....	28
3.3.1.3.	Estadísticas.....	29

	3.3.1.4.	Contacto	31
	3.3.1.5.	Salir.....	31
3.3.2.		Sección de navegación.....	32
	3.3.2.1.	Registros.....	32
	3.3.2.2.	Personalidad.....	33
	3.3.2.3.	Enseñar	35
	3.3.2.4.	Subir AIML	37
	3.3.2.5.	Corrector ortográfico	39
	3.3.2.6.	Buscar / Editar AIML	41
	3.3.2.7.	Demo chat	43
4.		CREANDO CONOCIMIENTO EN EL CEREBRO DEL ROBOT DE CHARLA HACIENDO USO DEL LENGUAJE AIML	45
4.1.		Categorías.....	45
	4.1.1.	Recursión.....	47
		4.1.1.1. Reducción simbólica.....	47
		4.1.1.2. División y conquista.....	48
		4.1.1.3. Sinónimos	48
		4.1.1.4. Corrección de ortografía y gramática...	49
		4.1.1.5. Palabras clave	50
		4.1.1.6. Condicionales	51
	4.1.2.	Focalización.....	52
	4.1.3.	Contexto	52
4.2.		Creando conocimiento desde un archivo AIML	56
	4.2.1.	Reglas para crear un correcto contenido en español	56
	4.2.2.	Cargando archivos AIML al cerebro del robot.....	59
4.3.		Creando conocimiento desde el área de administración	59

5.	BOTICÓN EL ROBOT DE CHARLA PARA LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS	61
5.1.	Modificaciones al proyecto Program-O	62
5.1.1.	¿Qué es Program-O?	62
5.1.2.	¿Por qué fue necesario modificar Program-O?	62
5.1.3.	Modificaciones realizadas al front-end	63
5.1.4.	Modificaciones visuales al diseño del sitio	63
5.1.5.	Modificaciones realizadas a la base de datos	63
5.1.6.	Precargando información	64
5.1.7.	Modificaciones realizadas al back-end	64
5.1.7.1.	Parser para carar archivos AIML	65
5.1.7.2.	Integración con los estándares de la Universidad Virtual	65
5.1.7.3.	Personalidad del Boticón	65
5.1.7.4.	Regionalización de la base de datos	66
5.1.7.5.	Eliminación de bugs	67
5.1.8.	Mejoras en general	68
5.2.	Automatización para los procesos de la Universidad Virtual	68
5.2.1.	Creación de contenido para horario de cursos	68
5.3.	¿Qué hacer en caso pierda la contraseña?	70
	CONCLUSIONES	71
	RECOMENDACIONES	73
	BIBLIOGRAFÍA	77
	APÉNDICE	79

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Flujo del robot de charla.....	2
2.	Evolución de los robots de charla.....	6
3.	Campo de texto con soporte de reconocimiento de voz.....	9
4.	Menú de opciones WAMP	16
5.	Menú de opciones XAMPP.....	17
6.	Menú de opciones MAMP	18
7.	Interfaz de usuario del robot de charla - conversación.....	23
8.	Interfaz de usuario del robot de charla - inicio.....	24
9.	Página de inicio – administrador	27
10.	Página de inicio – documentación.....	28
11.	Página de estadísticas – administrador.....	31
12.	Página de registros – administrador.....	33
13.	Página de personalidad –administrador	34
14.	Página de enseñanza – administrador	36
15.	Página de carga de archivos AIML – administrador.....	38
16.	Página del corrector ortográfico – administrador.....	40
17.	Página del corrector ortográfico – resultados.....	40
18.	Página del corrector ortográfico - añadir	41
19.	Página del editor AIML – administrador	42
20.	Página del editor AIML – actualizar.....	43
21.	Seleccionar el tipo de codificación UTF-8 – Notepad++.....	57
22.	Observaciones generales sobre archivos AIML en español.....	58

TABLAS

I.	Versiones de software necesarias	19
II.	Horario del segundo semestre para el curso IPC1	69

LISTA DE SÍMBOLOS

Símbolo	Significado
B	Es una secuencia de bits contiguos
GB	Es una unidad de almacenamiento de información, que equivale a 10^9 bytes
Hz	Es la unidad de frecuencia, representa un ciclo por cada segundo
MHz	Equivale a 10^6 hertzios
MB	Equivale a 10^6 bytes

GLOSARIO

.Net	Es un framework propiedad de Microsoft, independiente de la plataforma de hardware que permite un rápido desarrollo de aplicaciones.
AOL	Es una empresa estadounidense con sede en Nueva York que provee servicios de internet y medios.
<i>Back-end</i>	Comprende los componentes que procesan la salida del <i>front-end</i> .
Boticon	Robot de charla desarrollado especialmente para brindar información relacionada con la Escuela de Ciencias y Sistemas de la Universidad de San Carlos de Guatemala.
<i>Botmaster</i>	Persona o sistema responsable del mantenimiento del robot de charla.
<i>Bug</i>	Se le llama así a los defectos, deficiencias o errores en el software.
<i>Chat</i>	Se refiere a una comunicación escrita en la que se envían y reciben mensajes entre dos o más personas instantáneamente.

Framework	Estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.
Front-end	Es la parte de un sistema de software que interactúa directamente con el usuario.
GTalk	Programa de mensajería instantánea que utiliza el protocolo XMPP para hablar y chatear.
Hardware	Se refiere a todas las partes físicas tangibles de una computadora.
ICQ	Programa de mensajería instantánea para chatear.
Internet	Conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP.
IRC	Protocolo de comunicación basado en texto que permite chatear.
Java	Lenguaje de programación orientado a objetos con instancia, basada en gran parte en los lenguajes de programación C y C++.
Lenguaje de programación	Es un idioma artificial diseñado para dar instrucciones a máquinas como las computadoras.

Lenguaje de programación compilado	Es un lenguaje de programación traducido a otro lenguaje de programación a través de un programa generando un programa equivalente que la máquina sea capaz de interpretar.
Lenguaje de programación interpretado	Lenguaje de programación diseñado para ser ejecutado por un programa intérprete, contrario a los lenguajes compilados.
PayPal	Es una empresa que pertenece al sector del comercio electrónico que hace uso del Internet para transferir dinero entre usuarios.
Licencia Pública General de GNU	Un tipo de licencia creada por la fundación Free Software Foundation orientada a proteger la libre distribución, modificación y uso de software.
Multihilo	Capacidad del hardware para ejecutar múltiples subprocesos.
MyCyber Twin	Es un programa que provee soporte a personas a través de sitios web o dispositivos móviles. Sitio oficial http://site.mycybertwin.com/ .
Microsoft Outlook	Es un programa para manejar los correos electrónicos, propiedad de Microsoft y forma parte de la suite Microsoft Office.

Mozilla Thunderbird	Es un programa para manejar los correos electrónicos, propiedad de la Fundación Mozilla. Funciona en distintos sistemas operativos.
Notepad ++	Es un editor de texto con opciones avanzadas para desarrolladores y programadores. Posee una licencia de código fuente libre con soporte para varios lenguajes de programación.
Parser	Es un programa que analiza la sintaxis de un lenguaje de programación.
Scripting	Es un programa interpretado que generalmente se almacena en un archivo de texto plano, su uso común es para realizar tareas generales.
SmarterChild	Es un robot de charla capaz de adaptar su personalidad y aprender de sus usuarios.
Software	Comprende el conjunto de componentes que hacen posible la realización de tareas específicas, contrario a los componentes de hardware.
Textpad	Popular editor de texto de propósito general y de fácil uso para manipular archivos texto plano.
Windows live	Conjunto de servicios y productos de la empresa multinacional Microsoft.

RESUMEN

Se le llama robot de charla al programa creado para simular la interacción del lenguaje humano a través del habla o texto. Estos programas tienen como objetivo fundamental mantener una conversación con una persona haciendo uso de la inteligencia artificial u otros métodos de programación.

Los robots de charla utilizan como en cualquier otro medio de comunicación humano: un código, un canal, un emisor y un receptor. Estos cuatro aspectos naturales han sido adoptados virtualmente por los robots de charla para tratar de dar lugar a una conversación fluida y natural.

Los robots de charla con inteligencia artificial del tipo débil son los más comunes de encontrar ya que son relativamente simples de programar y enseñar, debido a que no tienen la habilidad de razonar y pueden ser programados en casi cualquier lenguaje.

Hasta el momento no es posible que un robot de charla sea capaz de seguir una conversación más o menos lógica a menos que se utilicen frases coherentes y de fácil comprensión.

En la actualidad los robots de charla se encuentran en varios nichos de *internet*, pero principalmente en las páginas *web* y la mensajería instantánea. El beneficio de utilizar estos robots de charla es brindar información puntual de manera rápida.

OBJETIVOS

General

Brindar una plataforma interactiva en la que los estudiantes puedan obtener instantáneamente respuestas a preguntas frecuentes.

Específicos

1. Utilizar un robot de charla que provea información rápida y concreta de temas específicos a los estudiantes.
2. Utilizar como plataforma el sitio web de la escuela de ciencias.
3. Embeber el robot de charla en un sitio web.
4. Simular una conversación, el usuario haciendo uso del teclado para escribir preguntas y el robot de charla contestando las respuestas en pantalla.
5. Proporcionar un módulo administrador (de fácil uso) para que el personal docente y administrativo enriquezca la información que provee el robot de charla.
6. Eliminar los tiempos de espera en atención a estudiantes cuando soliciten información específica.

7. Apoyar al personal docente y administrativo en las tareas informativas del día a día delegando las tareas que consideren pertinentes al robot de charla.

8. Brindar retroalimentación enviado a través de informes estadísticos al personal docente y administrativo, sobre las búsquedas / consultas que realizan los estudiantes.

INTRODUCCIÓN

Los robots de charla son algoritmos, programas que hacen uso del procesamiento del lenguaje natural en un sistema de preguntas y respuestas. Hoy en día existen muchas dificultades asociadas al reconocimiento del lenguaje humano, a su interpretación y la posibilidad de las máquinas de entender lo que comunica una persona de manera escrita.

Cualquier conversación toma lugar bajo un determinado contexto, siendo esta una palabra que abarca un contenido amplio. Para este trabajo queda restringida a su significado en una conversación tradicional de *chat*, sin sonidos, sin imágenes sensoriales que complementen el diálogo restringiendo el significado correcto de algunas frases. En este tipo de conversaciones el robot de charla debe deducir la información de su interlocutor a partir de lo que él le diga.

El tipo de robot de charla que se presenta en este trabajo funciona básicamente buscando palabras clave en la frase de entrada, palabras que fueron asociadas a conjuntos de reglas que descomponen la frase original y arman una respuesta simulando una conversación. Cuando una frase u oración del usuario coincide con una de las frases aprendidas por el robot, se devuelve la respuesta asociada. Sin embargo, existen muchas frases de entradas cortas en las que este comportamiento no es suficiente y se hace necesaria una evaluación del contexto. Este tipo de comportamiento es una primera aproximación al seguimiento del contexto.

¿Cómo funciona internamente un robot de charla? ¿Qué tipo de robot de charla se adapta mejor a las necesidades de la Escuela de Ciencias y Sistemas? este trabajo es una adaptación al español del motor de *chat* e intérprete de archivos AIML del conocido proyecto Program-O, escrito en PHP y utilizando MySQL como motor de base de datos, estas son algunas de las respuestas que se brindan a continuación en este trabajo de graduación.

1. MARCO TEÓRICO CONCEPTUAL

A través de los años a los robots de charla se les ha llamado también *chatterbots*, *bots* de charla, *bots* de *chat*, agentes de conversación, agentes virtuales, o *chatbots* y empezaron a ganar notoriedad gracias a ELIZA, un programa de Joseph Weizenbaum publicado en 1966, el cual hacía creer a algunos usuarios que estaban hablando con una persona real al momento de someterlo a la prueba de Turing.

1.1. ¿Qué es un robot de charla?

Se le llama robot de charla al programa creado para simular la interacción del lenguaje humano a través del habla o texto. Estos programas tienen como objetivo fundamental mantener una conversación con una persona haciendo uso de la inteligencia artificial u otros métodos de programación. La mayoría de los robots de charla simplemente buscan palabras clave, frases y patrones que han sido programados en sus bases de datos y algunos otros utilizan técnicas más avanzadas. Hasta la fecha ningún robot de charla ha sido capaz de engañar completamente a los humanos para hacerles creer que no es un robot.

Tradicionalmente la forma de comunicarse con el robot de charla ha sido mediante el uso del teclado y monitor, pero gracias a los avances tecnológicos se puede hacer uso de métodos alternos como el sonido. Este método no se emplea en este trabajo de investigación ya que aún no es tan popular ni ampliamente utilizado como el método tradicional.

Los robots de charla utilizan como en cualquier otro medio de comunicación humano: un código, un canal, un emisor y un receptor. Estos cuatro aspectos naturales han sido adoptados virtualmente por los robots de charla para tratar de dar lugar a una conversación fluida y natural.

1.2. ¿Cómo funciona el robot de charla?

El funcionamiento de cada uno de los robots de charla varía entre uno y otro. A continuación se describe sobre el funcionamiento específico de Boticon un robot de charla desarrollado especialmente para la Escuela de Ciencias y Sistemas.

El flujo de funcionamiento entre el robot de charla y su usuario puede ser observado en la siguiente gráfica.

Figura 1. Flujo del robot de charla



Fuente: elaboración propia, con programa de PhpStorm.

El robot de charla está a la espera de un estímulo de entrada, es decir, preguntas y/o frases que el usuario escribe desde una interfaz *web* diseñada especialmente para transferir meta información a la interfaz de la aplicación, la cual consulta a la base de datos en busca de patrones previamente cargados desde archivos AIML.

Todas las concordancias de patrones son devueltas desde la base de datos hasta la interfaz de la aplicación y posteriormente hasta la interfaz *web*.

1.2.1. Tipos de inteligencia artificial en los robots de charla

Debido a la programación avanzada y cantidad de tiempo que requiere enseñar a un robot de charla a comunicarse adecuadamente, existen los del tipo débil y fuerte.

1.2.2. Los robots de charla con inteligencia artificial débil

Son los más comunes de encontrar ya que son relativamente simples de programar y enseñar, debido a que no tienen la habilidad de razonar y pueden ser programados en casi cualquier lenguaje, muchos de ellos están escritos en PHP, .Net, Java, etc. Son de uso común en salas de charla IRC, en la mensajería instantánea como AOL, Yahoo!, Windows Live, ICQ y servicios basados en el protocolo XMPP como gTalk.

Existen dos formas de programar los cerebros para estos robots: la primera es creando una base de datos vacía sin que el robot de charla sepa absolutamente nada y guarde lo que la gente dice y sus respuestas a los comentarios que están en la base de datos hasta que alguien más use la misma frase.

Existen varios inconvenientes al utilizar este método, el primero de ellos es la cantidad de horas de tiempo de conversación que necesita el robot de charla para aprender suficiente información y mantener una conversación decente, además son susceptibles de aprender malos hábitos de las personas que les hablan ya que muchas abusan y les enseñan cosas sucias y/o

malintencionadas, es por esto que generalmente no pueden tener más que una simple y superficial conversación que consiste en los comentarios de gente que previamente hizo.

La segunda forma es a través de programación directa con preguntas, frases y palabras y como debe responder a estas, la desventaja de usar este método es que requiere una cantidad de tiempo considerable para el programador ya que el robot solo responderá a frases que existan en su diccionario de datos, pero la ventaja es que puede dirigir una conversación a través de caminos y recordar ciertos detalles sobre los usuarios, además son excelentes para conversaciones de temas especializados. La aplicación de software del presente trabajo utiliza esta forma de programación.

1.2.3. Los robots de charla con inteligencia artificial fuerte

Utilizan métodos sofisticados incluyendo reglas gramaticales, definición de conocimientos y otras vías de entendimiento natural.

Además aplican un método de respuesta mucho más complicado e intentan comprender e interpretar lo que se está diciendo. Para que un robot de charla sea categorizado del tipo fuerte, no debe simplemente dar respuestas pre-programadas debe ser capaz de mostrar cierto nivel de entendimiento y crear respuestas por su propia cuenta. Un ejemplo de este tipo de robot se puede encontrar en el sitio MyCyberTwin, el cual provee soporte a clientes de banca e incluso para la NASA. Y no solo las empresas grandes se dedican a trabajar con este método, existen individuos como Rollo Carpenter's quien posee un robot de charla conocido como Jabberwacky el cual ha aprendido de las charlas con otras personas a través del tiempo.

1.3. La prueba de Turing

Esta prueba consiste en que un humano actúa como juez en una conversación natural con un robot de charla actuando como un humano, se intenta desarrollar una conversación natural y si el juez no puede identificar claramente más del 50% de las veces si el robot es un humano o no, se considera que ha pasado la prueba. La prueba fue descrita por Alan Turing en 1950 y aún se utiliza en la actualidad para premiar anualmente el concurso por el premio Loebner de robots de charla. Existen tres tipos diferentes de variantes de esta prueba, pero hasta la fecha ningún robot ha pasado oficialmente una prueba de Turing.

1.4. Comportamiento de los robots de charla según el contexto

Hasta el momento no es posible que un robot de charla sea capaz de seguir una conversación más o menos lógica a menos que se utilicen frases coherentes y de fácil comprensión. Esto permite establecer una conversación con el robot de charla – para un contexto dado – debido a que estos tienen en cuenta las palabras claves del interlocutor permitiendo usar un conjunto de respuestas preparadas de antemano.

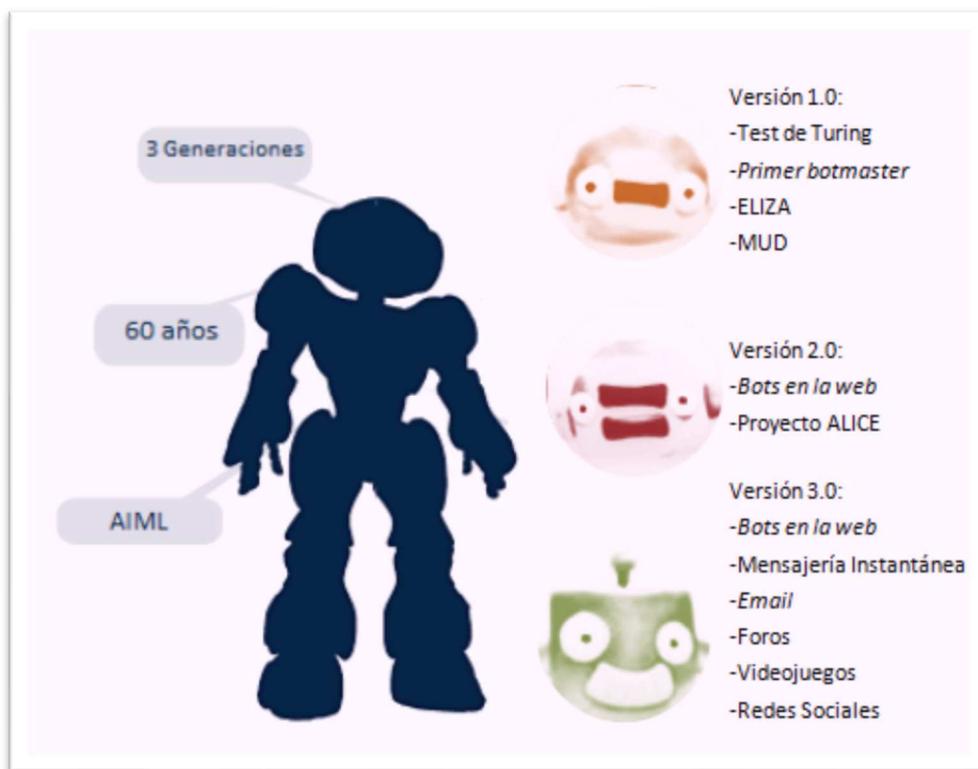
1.5. Uso de los robots de charla a través del tiempo

A través del tiempo los robots de charla han evolucionado de ser simples sentencias de *software* a convertirse en un lenguaje especializado de patrones y estímulos funcionando en nichos específicos, tales como: mensajería instantánea, correo electrónico, *posts* en foros, juegos y redes sociales.

El concepto del robot de charla fue concebido por primera vez por Alan Turing en 1950, pero fue hasta que Joseph Weizenbaum se convirtiera en el primer *botmaster* creando el programa ELIZA. Fue ELIZA el primer programa de inteligencia artificial que aplicó el concepto de estímulo – respuesta.

La siguiente infografía muestra de manera resumida los acontecimientos más importantes desde los tempranos inicios de los robots de charla hasta la fecha.

Figura 2. **Evolución de los robots de charla**



Fuente: elaboración propia, con programa de PhpStorm.

1.6. Robots de charla en internet

En la actualidad los robots de charla se pueden encontrar en varios nichos de *internet*, pero principalmente en las páginas *web* y la mensajería instantánea. El beneficio de utilizar estos robots de charla es brindar información puntual de manera rápida, es por eso que portales en línea como PayPal los utilizan para ofrecer soporte a sus clientes, PayPal estima que manejan alrededor de 400,000 conversaciones al mes.

Otra empresa que aprovecha los beneficios de los robots de charla es Coca Cola, proporcionando a los usuarios información general sobre la marca así como recomendaciones e información pertinente a los procesos de mercadeo y publicidad.

Por citar algunos ejemplos de robots de charla de tipo débil, se menciona el SmarterChild y *bots* individuales creados para fines de entretenimiento o para inundar de *spam* e información no deseada las salas de *chat*.

Pero también existen compañías que utilizan sistemas con una mezcla de inteligencia artificial débil y fuerte como es el caso de MyCyberTwin, ellos permiten a individuos ajustar a la medida un robot de charla, además de trabajar con equipos de negocio para implementar agentes virtuales embebidos en sitios *web*.

1.7. Interactuando con los robots a través del reconocimiento de voz

El reconocimiento de voz es un soporte que ha sido agregado recientemente a la quinta revisión importante del lenguaje básico de la *WorldWide Web* conocido como HTML5 e implementado por Google bajo el nombre de Speech API. Hasta el momento funciona únicamente con el navegador Google Chrome versión 10 corriendo en un sistema operativo Windows 7 y utilizando un parámetro adicional para habilitar la API. Se espera que esta funcionalidad sea soportada más adelante por el resto de navegadores populares, como opera, Firefox e Internet Explorer en diferentes sistemas operativos.

La API de reconocimiento de voz al momento en que se escribe este apartado reconoce únicamente el idioma inglés, sin embargo es importante incluirlo como referencia para el futuro administrador del *bot*.

Para activar el soporte de reconocimiento de voz basta con agregar un atributo adicional a la etiqueta de campo de texto en nuestro código HTML:

Campo de texto sin soporte de reconocimiento de voz

```
<input type=text name=ejemplo/>
```

Campo de texto con soporte de reconocimiento de voz

```
<input type=text name=ejemplo x-webkit-speech />
```

Para iniciar con el proceso se debe hacer clic en el icono del micrófono que aparece junto al campo de texto, el Google Speech API detectara el momento en que termines de hablar y automáticamente escribirá en pantalla el texto detectado, haz clic en el botón si deseas enviar el texto al *bot*.

Figura 3. **Campo de texto con soporte de reconocimiento de voz**



Fuente: elaboración propia, con programa de PhpStorm.

Se debe tomar en cuenta que la calidad de reconocimiento puede variar debido a factores que afectan la calidad del sonido, por ejemplo: ruido ambiental, el acento de la persona que habla, la calidad de fabricación del micrófono, el tipo de tarjeta de sonido de la computadora, entre otros.

Si no observas el dibujo del micrófono puede que la API de reconocimiento de voz no esté habilitada por defecto, si este es el caso deberás crear un acceso directo y agregar en el campo Objetivo de sus propiedades --enable-speech-input. Por ejemplo:

```
C:\Program Files\Google\Chrome\Application\chrome.exe --enable-speech-input
```

El soporte de reconocimiento de voz que se describe en este apartado es válido para cualquier implementación de robot de charla basado en Pandorabots. Al momento de escribir esta tesis el soporte de reconocimiento de voz es soportado únicamente para el idioma inglés y por navegadores basados en *webkit*, por lo que recomienda utilizar una versión de *Google Chrome 20.x* o superior.

2. PROCEDIMIENTOS PARA DAR MANTENIMIENTO AL CÓDIGO FUENTE DEL ROBOT DE CHARLA

Este trabajo es una adaptación al español del motor de *chat* e intérprete de archivos AIML del conocido proyecto Program-O, fue escrito en PHP utilizando MySQL como motor de base de datos. Está compuesto por un intérprete de archivos AIML el cual proporciona preguntas y respuestas al cerebro del robot y un núcleo de reglas que lo adhieren a las especificaciones e interfaces de *chat*.

Program-O fue lanzado en febrero de 2010 como continuidad al desatendido proyecto Program-E (que por si fuera poco estaba plagado de *bugs*) a cargo de Elizabeth Perreau bajo la Licencia Pública General de *GNU*.

2.1. Requisitos del sistema

El apartado 2.2 se listan algunos programas que instalan y configuran automáticamente el lenguaje de programación, la base de datos y el servidor *web* que se describen a continuación.

2.1.1. Lenguaje de programación

PHP es un lenguaje de *scripting* ampliamente utilizado para fines generales. Es Especialmente adecuado para el desarrollo *web* y puede ser embebido en páginas HTML.

Es usado principalmente en interpretación del lado del servidor (*server-side scripting*) y las razones principales para utilizar este lenguaje de programación es porque puede ser integrado fácilmente en la mayoría de servidores *web* y en casi todos los sistemas operativos y plataformas sin costo alguno.

La Escuela de Sistemas forja un fuerte perfil en el desarrollo de *software* basado en el popular lenguaje de programación C++. Este factor aunque no fue fundamental para la decisión de utilizarlo para desarrollar el robot de charla es beneficioso para darle mantenimiento porque la curva de aprendizaje entre los programadores de C++ y PHP es muy corta, lo que trae beneficios al proyecto en los tiempos de desarrollo.

El robot de charla fue programado utilizando la versión 5.3.0 del lenguaje de programación PHP. Si bien es posible que funcione con versiones 5.x.x en adelante, se recomienda revisar la bitácora para encontrar posibles cambios que afecten el comportamiento del mismo.

2.1.2. Base de datos

MySQL es un sistema de gestión de base de datos relacional, multi hilo y multiusuario. Fue la elección de base de datos para este proyecto por su gran popularidad en aplicaciones *web*. Esta popularidad como base de datos para la *web* está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

Es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones *web* hay baja

conurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. MySQL corren en más de 20 plataformas, incluidas Windows, Linux y Mac OS.

La información que persiste en la base de datos del robot de charla utiliza la versión 5.1.36 de MySQL, si bien es posible que funcione con versiones 5.x.x en adelante se recomienda revisar la bitácora para encontrar posibles cambios que afecten el comportamiento del mismo.

2.1.3. ¿Qué es un servidor *web*?

Es un programa que está diseñado para procesar cualquier aplicación en el lado del servidor a través de conexiones bidireccionales y/o unidireccionales, generando dos tipos de peticiones y respuestas con los clientes: síncronas o asíncronas, o bien, cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador *web*. Para la transmisión de todos estos datos suele utilizarse algún protocolo, siendo el más popular para estas comunicaciones el HTTP, el cual pertenece a la capa de aplicación en el modelo OSI.

2.1.3.1. ¿Por qué se utiliza?

Apache es ampliamente utilizado en *Internet* para enviar páginas *web* estáticas y dinámicas. Este proyecto así como muchas aplicaciones *web* están diseñados para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable.

Es el componente de servidor *web* en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python (y ahora también Ruby).

Las páginas dinámicas y estáticas servidas por Apache utilizan la versión 2.2.11, si bien es posible que funcione con versiones 2.x.x en adelante se recomienda revisar la bitácora para encontrar posibles cambios que afecten el comportamiento del mismo.

2.1.4. Sistema de control de versiones

Un sistema de control de versiones automatiza tareas comunes sobre los cambios que se realizan en los archivos: recuperarlos, guardarlos, mezclarlos e identificarlos.

2.1.4.1. Subversión ¿Por qué utilizarlo?

Porque durante el desarrollo y/o mantenimiento de un proyecto los archivos son modificados frecuentemente siendo conveniente mantener un histórico de los cambios que se realizan. El cliente más popular de subversión es TortoiseSVN.

2.2. Preinstalación del robot de charla

El desarrollo del robot de charla tiene como objetivo primordial operar en múltiples sistemas operativos, es por eso que previamente se justificó el uso de PHP, MySQL y Apache como lenguaje de programación, base de datos y servidor *web* respectivamente.

La instalación individual de cada uno de estos componentes requiere de configuraciones adicionales para cada una de las plataformas y ambientes de trabajo (desarrollo, pruebas y producción). WAMP, XAMPP y MAMP consisten cada uno en un sistema de infraestructura que brinda una rápida y fácil solución a los problemas previamente mencionados.

Es sumamente recomendable utilizar un *software* de control de versiones sin importar el sistema operativo en el que se esté trabajando, existen muchas versiones gratuitas y queda a discreción del desarrollador seleccionar el que mejor se acomode a sus conocimientos, preferencias y necesidades. En la sección 2.3.3 se proporciona mayor información.

A continuación se presenta el *software* recomendado para los sistemas operativos más utilizados.

2.2.1. Sistema operativo Windows

WAMP es una suite que permite reproducir el ambiente del servidor de producción y cuyo uso es muy intuitivo brindándonos la capacidad de ajustar el ambiente de trabajo sin necesidad de tocar los archivos de configuración. Una vez instalado se tiene la posibilidad de agregar cuantos lanzamientos de Apache, MySQL y PHP se desee.

Una guía rápida de instalación paso a paso, foros y soporte se puede encontrar en <http://www.wampserver.com/en/presentation.php>. Si suscitaran errores de instalación, configuración o uso se pueden enviar preguntas o consultar el foro de WAMP en <http://www.wampserver.com/phorum/>. WAMP puede ser descargado desde <http://www.wampserver.com/>

Figura 4. **Menú de opciones WAMP**



Fuente: WAMP Server, http://www.wampserver.com/data/image_menu_wamp.gif.

Consulta: 29 de julio de 2012.

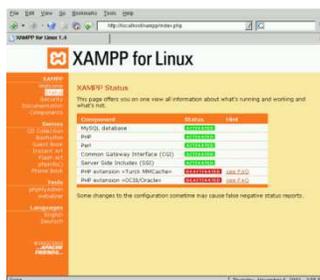
2.2.2. Sistema operativo Linux

XAMPP es la suite por excelencia para una rápida instalación y configuración de nuestro ambiente de trabajo en cualquier distribución Linux. La instalación consiste en 4 sencillos pasos que van desde la descarga del *software* hasta las pruebas que verifican el correcto funcionamiento. Una característica importante en la que hacen énfasis los creadores es que XAMPP no fue ideado para ambientes de desarrollo ya que ha sido configurado tan abierto como sea posible para permitir al programador hacer lo que este quiera.

Los sistemas Linux se han caracterizado desde hace tiempo por la robusta seguridad en sus sistemas de *software*, pero debido a la razón con la que fue ideado XAMPP instalarlo en ambientes de producción puede ser fatal.

Una guía rápida de instalación, noticias, foros y soporte de XAMPP puede ser encontrado en <http://www.apachefriends.org/en/index.html>

Figura 5. **Menú de opciones XAMPP**



Fuente: Pictures of XAMPP for Linux, <http://www.apachefriends.org/en/xampp-linux-screenshots.html>. Consulta: 29 de julio de 2012.

Si necesita configurar Apache, PHP y MySQL en un ambiente de producción Linux es indispensable que consulte con su administrador de sistemas, él deberá orientarle de la mejor manera. Si no dispone de él, existe una inmensa cantidad de información y documentos sobre la configuración e instalación de cada uno de estos paquetes en sus respectivos sitios oficiales.

2.2.3. Sistema operativo Mac OS

MAMP no difiere de las dos suites mencionadas previamente en cuanto a facilidad de uso, y al igual que XAMPP fue creado principalmente para ambientes de desarrollo en PHP y no es recomendado para instalar en servidores de producción.

Utilizar MAMP no tiene ningún costo asociado y se distribuye gratuitamente bajo la licencia pública general de GNU.

Figura 6. **Menú de opciones MAMP**



Fuente: Screenshots of MAMP, <http://www.mamp.info/en/mamp-pro/screenshots.html>.

Consulta: 29 de julio de 2012.

2.3. Preconfiguración del robot de charla

A continuación se describen los pasos para preparar el ambiente de desarrollo que permita hacer cambios y ajustes al comportamiento del robot de charla.

2.3.1. Navegador predeterminado para ambiente de desarrollo

En algunos casos WAMP, XAMPP o MAMP podría necesitar que seleccione la ruta de acceso a su navegador predeterminado con el fin de desplegar el estado de instalación y sus respectivas configuraciones de PHP, Apache y MySQL.

Es muy recomendable utilizar el navegador Firefox porque brinda un conjunto de herramientas para desarrollo *web* que son gratuitas, ahorran gran cantidad de tiempo en el trabajo, son de calidad y facilitan la programación del robot de charla.

2.3.2. Revisión de la configuración del servidor

Las siguientes versiones fueron utilizadas durante el desarrollo del presente trabajo. Es posible utilizar diferentes versiones pero se recomienda revisar la bitácora de cambios para evitar funcionamiento errático en el comportamiento del robot de charla.

Tabla I. **Versiones de software necesarias**

Servidor <i>web</i> Apache	2.2.11
Lenguaje de programación PHP	5.3.0
Base de datos MySQL	5.1.36
Control de versiones TortoiseSVN	1.6.15

Fuente: elaboración propia, con programa Excel de Microsoft Office.

Además es necesario que las siguientes extensiones de PHP estén instaladas y habilitadas en el servidor: Core, date, iconv, pcre, tokenizer, PDO, xmlreader, mysql, bcmath, ereg, json, Reflection, zip, Phar, xmlwriter, mysqli, calendar, filter, mcrypt, sesión, zlib, SimpleXML, apache2handler, pdo_mysql, com_dotnet, ftp, mysqlnd, SPL, libxml, wddx, gd, pdo_sqlite, ctype, hash, odbc, standard, dom, xml, mbstring, mhash.

2.3.3. Obtener una copia local del código fuente

Este proyecto fue desarrollado haciendo uso de un sistema estándar de control de versiones. Una vez instalado el cliente SVN se procede a hacer checkout en la carpeta ubicada en la ruta configurada por Apache como directorio público `www`.

El repositorio se encuentra hospedado en internet en el sitio *xp-dev.com* y puedes acceder públicamente utilizando la cuenta de invitado con los siguientes parámetros:

- Ruta del repositorio: `http://svn3.xp-dev.com/svn/Boticon/trunk`
- Usuario: `desarrollador`
- Contraseña: `boticon`
- Instalación del robot de charla

Es indispensable haber realizado las configuraciones previas a la instalación, consulte el apartado 2.3 antes de proceder a instalar el robot de charla. El manual de instalación puede ser encontrado en la sección de Anexos. La utilización del control de versiones es un paso necesario para obtener la copia oficial del código fuente mas actualizado, sin embargo no es un paso obligatorio.

3. PROCEDIMIENTOS PARA USO Y ADMINISTRACIÓN DEL ROBOT DE CHARLA

Entre los requisitos técnicos para el uso y administración aunque no es indispensable el uso de una computadora portátil o de escritorio para administrar el contenido del robot de charla es recomendable porque la interfaz fue diseñada especialmente para ello utilizando la última versión del navegador Firefox.

3.1. Requisitos del sistema

Se requiere de un dispositivo con acceso a internet que agrupe como mínimo las siguientes características:

3.1.1. Windows

- Sistema operativo: Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows 7.
- Pentium 233 MHz (Recomendado: Pentium 500MHz o superior)
- 64 MB RAM (Recomendado: 128 MB RAM o más)
- 52 MB de espacio en disco

3.1.2. Mac

- Sistema operativo: Mac OS X 10.4 o superior
- Computadora Macintosh con procesador Intel x86 o PowerPC G3, G4, o G5.
- 128 MB RAM (Recomendado: 256 MB RAM o más)
- 200 MB de espacio en disco

3.1.3. Linux

Tome en cuenta que las distribuciones de Linux proporcionan paquetes para tu sistema operativo con requisitos diferentes.

- CPU Intel Pentium II o AMD K6-III+ 233 MHz (Recomendado: 500MHz o superior).
- 64 MB RAM (Recomendado: 128 MB RAM o más)
- 52 MB de espacio en disco

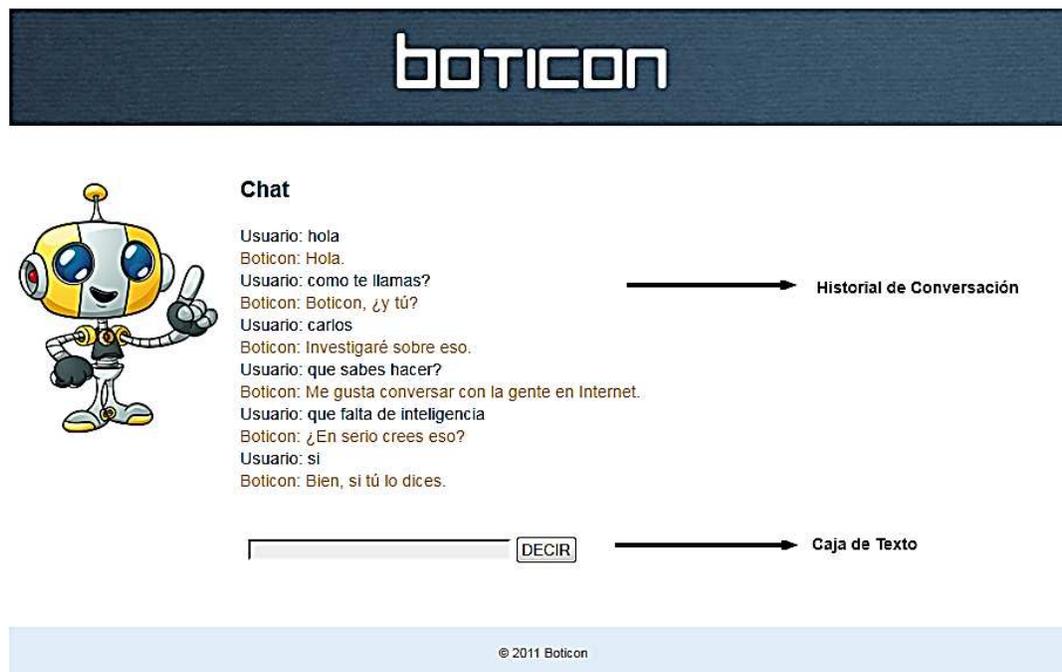
3.2. Procedimiento para hacer uso del robot de charla

Hacer uso del robot de charla es tan sencillo como abrir una página *web* y empezar a chatear. Si usa al robot desde una copia local en la computadora consulte el paso 15 de la sección de anexos.

La Universidad Virtual de la Escuela de Sistemas tiene una versión funcional del robot de charla, con información relevante sobre gestiones administrativas de la Escuela, información sobre trabajos de graduación relacionados con el departamento de lingüística e información trivial sobre la personalidad del robot. Ingrese a <https://ecys.ingenieria-usac.edu.gt/boticon/>.

La interfaz es muy intuitiva y sencilla, dispone de una caja de texto en la que se escribe al robot sobre información que desea conocer y un espacio para ver el historial de la conversación, el cual muestra hasta 13 líneas de conversación al mismo tiempo. La cantidad de información que puede proporcionar el robot depende de la cantidad de información que el botmaster le ha enseñado.

Figura 7. **Interfaz de usuario del robot de charla - conversación**



Fuente: elaboración propia, con programa de PhpStorm.

3.3. Procedimiento para administrar el robot de charla

Administrar el robot de charla no es un proceso sencillo como hacer uso de él, para acceder a esta área necesita una cuenta con privilegios de administrador. Abra una página web y escriba <https://ecys.ingenieria-usac.edu.gt/boticon/admin/>, debe autenticarse y así tener acceso.

El área de administración está diseñada para presentar información relevante en 4 secciones importantes: pestañas, navegación, contenido y pie de página. A continuación se describe el resumen sobre las características.

Figura 8. Interfaz de usuario del robot de charla - inicio



Fuente: elaboración propia, con programa de PhpStorm.

La sección de pestañas contiene información de administración general sobre el robot. La acción de hacer clic en cada una de las opciones de esta sección resaltará el color amarillo y desplegará su contenido correspondiente. Las opciones disponibles son:

- Inicio: pantalla de bienvenida para el administrador del robot
- Documentación: información relevante para crear información en el cerebro del robot utilizando el lenguaje AIML.
- Estadísticas: hechos numéricos sobre las conversaciones de los usuarios con el robot.
- Contacto: solicitud de información vía correo electrónico
- Salir: finaliza la sesión de administración

La sección de navegación agrupa funcionalidades propias del robot, que permiten administrar su comportamiento, contenido y personalidad así como bitácoras de conversaciones con todos los usuarios con que el botmaster pueda recibir retroalimentación sobre temas y frases que utilizan los usuarios con el fin de crear información relevante en el cerebro del robot. Las opciones disponibles son:

- Registros: bitácora de cada una de las conversaciones que han tenido los usuarios con el robot de charla.
- Personalidad: permite la modificación del conjunto de características del robot.
- Enseñar: permite crear conocimiento en el cerebro del robot de charla para luego comunicarlo a los usuarios.
- Subir AIML: opción para crear conocimiento en el cerebro del robot a partir de un conjunto de conocimientos almacenados en archivos de tipo XML.

- Corrector Ortográfico: analiza palabras con el fin de detectar y corregir de forma automática faltas ortográficas en las preguntas que hacen al robot.
- Buscar / Editar AIML: permite la búsqueda y edición de palabras almacenadas en el cerebro del robot.
- *Demo Chat*: permite charla con el robot sin afectar las estadísticas y registros del mismo.

La función principal de la sección de contenido es mostrar la información pertinente a cada una de las opciones de la sección de pestañas y navegación. En esta área el botmaster deberá hacer uso de formularios que actualizan la personalidad del robot, que le enseñan a responder ante preguntas formuladas por los usuarios, que corrigen la mala escritura de los usuarios, y que ponen a prueba la capacidad de conversación ante la generalidad de preguntas que son capaces de hacer los usuarios.

La sección del pie de página muestra información sobre el usuario que ha iniciado sesión, desde que se conecta IP, a qué hora y cuándo se conectó por última vez y en la sesión actual.

En las siguientes secciones se ampliará el funcionamiento de cada una de las opciones presentadas anteriormente y en el capítulo 5, ejemplos funcionales creados especialmente para el administrador de la Universidad Virtual de la Escuela de Ciencias y Sistemas de la Universidad de San Carlos de Guatemala.

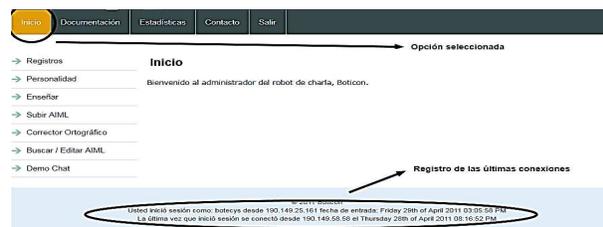
3.3.1. Sección de pestañas

La sección de pestañas está comprendida por cinco opciones: inicio, documentación, estadísticas, contacto y salir.

3.3.1.1. Inicio

La página de inicio es la predeterminada a desplegar al momento de iniciar sesión en el área de administración del robot. Todas las opciones que se ven en esta página pueden ser accedidas en cualquier momento que lo necesite. Siempre que se inicie la sesión, preste especial atención al pie de página para corroborar que usted ha iniciado una sesión previa y la integridad de la información del robot no ha sido expuesta a personas no autorizadas. Si tiene sospechas que alguien no autorizado ha iniciado sesión con su cuenta, comuníquese inmediatamente con el administrador de la Universidad Virtual para que cambie su contraseña y disponga de las medidas necesarias. Consulte el capítulo 5 para mayor información.

Figura 9. **Página de inicio – administrador**

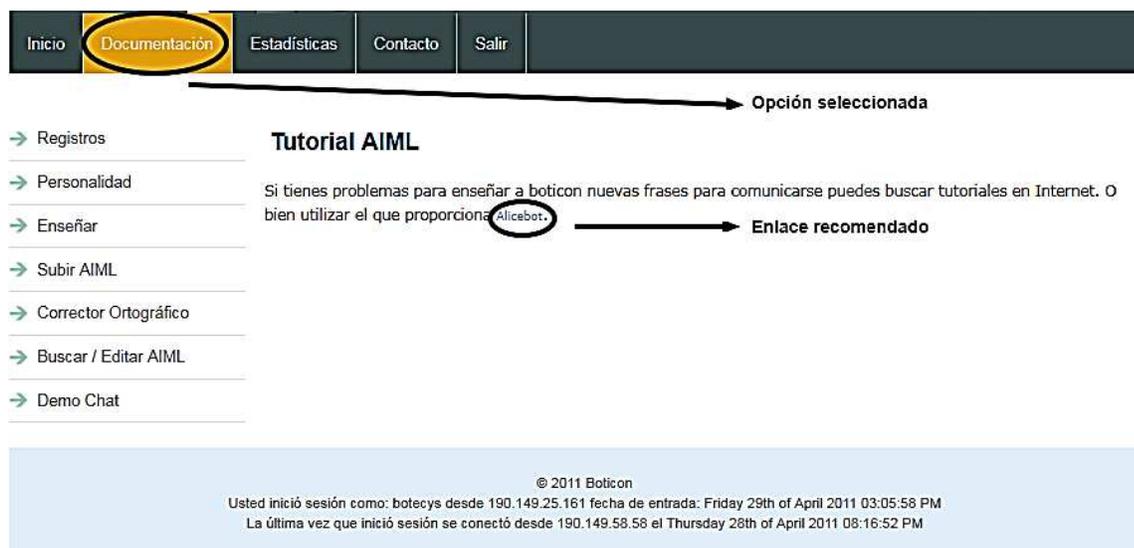


Fuente: elaboración propia, con programa de PhpStorm.

3.3.1.2. Documentación

La página de documentación es la utilizada para brindar soporte rápido a las tareas de administración del robot de charla. Una de las tareas más importantes es crear conocimiento en el cerebro del robot, por lo que es indispensable que el *botmaster* conozca a detalle el funcionamiento, opciones y significado del lenguaje de programación AIML. En esta sección encontrará un enlace con soporte rápido, ejemplos y guías para enseñar al robot nuevas categorías de información. El capítulo 4 recopila los ejemplos más importantes tomados de varios tutoriales y de mi experiencia creando los archivos que fueron cargados en el cerebro del robot al momento de desarrollarlo para la Universidad Virtual.

Figura 10. Página de inicio – documentación



Fuente: elaboración propia, con programa de PhpStorm.

3.3.1.3. Estadísticas

La página de estadísticas muestra información relevante sobre el número de conversaciones y líneas de *chat* por sesión de usuario. Es importante para el administrador del robot saber interpretar esta información: un número alto de conversaciones no quiere decir necesariamente que los usuarios estén usando al robot como herramienta informativa y no hay que esperar que las estadísticas de visitas sean altos, mientras el robot no tenga un amplio contenido que informar, incluso es recomendable que el robot de charla posea bastante conocimiento sobre eventos triviales y de la vida diaria si se desea que sea popular entre los usuarios.

También se recomienda que antes de hacer publicidad sobre este novedoso servicio en la Escuela, se hagan pruebas de satisfacción entre los estudiantes, hasta alcanzar una aceptación alta, con el fin de obtener la mayor satisfacción posible desde su primer uso. Esto generará que los mismos estudiantes hagan popular al servicio, entusiasmándose a seguir utilizándolo en una futura ocasión y recomendándolo entre sus compañeros y amigos.

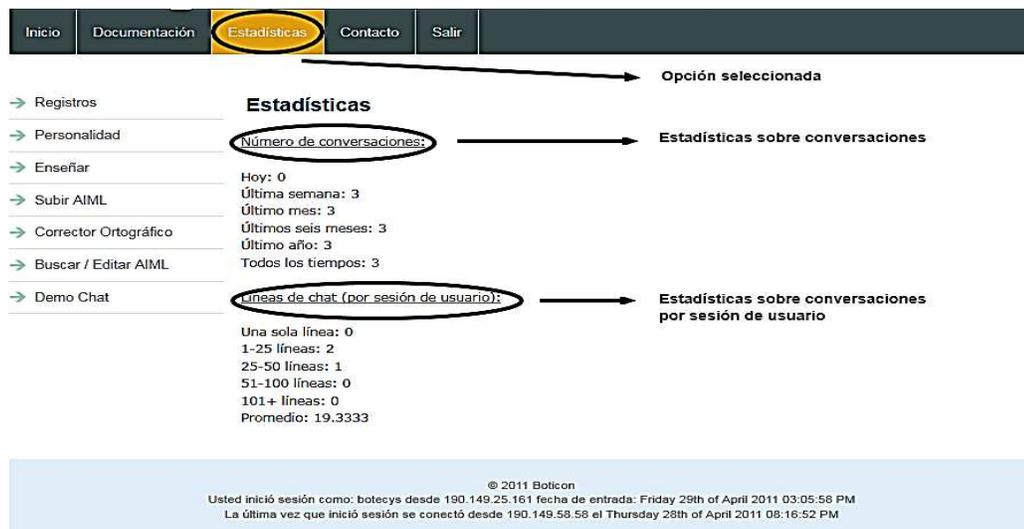
Se analizarán algunos ejemplos con el fin de indicar al administrador en qué área es deficiente o está fallando el robot con su cometido.

- Bajo número de conversaciones con una sola línea de *chat*: este es el peor de los casos, indica que nadie está utilizando el servicio.
- Alto número de conversaciones con una sola línea de *chat*: aún es un mal caso, porque a pesar que hay muchas conversaciones únicamente le hacen una pregunta y terminan por finalizar el servicio.

- Una de las razones puede ser el pobre contenido del robot, se recomienda analizar la página de registros para recopilar mayor información.
- Bajo número de conversaciones con un máximo de 25 líneas de *chat*: es un mal indicador, porque el servicio es impopular y puede ser debido al pobre contenido, o los usuarios no saben que más preguntar al robot. Podría entrenarse al robot para informar al usuario sobre los conocimientos que puede ofrecer.
- Bajo número de conversaciones con un rango de 25 a 100 líneas de *chat*: se puede considerar como un servicio normal, los usuarios están consultando bastante información del robot. El número de conversaciones podría aumentar al promocionar el servicio en diferentes medios: impresos, en el sitio de la UV, a través de catedráticos, etc.
- Alto número de conversaciones con un rango de 25 a 100 líneas de *chat*: un claro indicador que el servicio se ha vuelto popular y una buena herramienta informativa, existe una alta satisfacción de usuarios y se recomienda no descuidar la actualización constante de conocimientos en el robot.

Por último se menciona que el promedio de líneas de *chat* por usuario se calcula en base al número de sesiones distintas, es decir: el usuario Carlos chateó con el robot en 3 diferentes ocasiones, en cada ocasión escribió: 5, 20 y 15 líneas respectivamente, la suma total es de 40 líneas por sesión. El usuario Estuardo chateó con el robot en 2 diferentes ocasiones: 8 y 16 líneas respectivamente, la suma total es de 24 líneas por sesión. El promedio se calcula sobre la suma de todas las líneas dividido el número de conversaciones por sesión de usuario, es decir $(40 + 24) / 2 = 32$ líneas.

Figura 11. **Página de estadísticas – administrador**



Fuente: elaboración propia, con programa de PhpStorm.

3.3.1.4. **Contacto**

El enlace contacto no lleva a ninguna página, es una acción que lanza la aplicación que maneja al usuario por defecto los mensajes de correo (*Outlook, Thunderbird, Windows Live Mail, etc.*) y rellena los campos Para, De y Asunto permitiendo ahorrar tiempo y preparando el mensaje de correo para que el administrador solicite ayuda y/o sea asistido en la ejecución de alguna tarea o problema con el sistema.

3.3.1.5. **Salir**

Esta acción finaliza la sesión como administrador. No puede ingresar de nuevo al área de administración hasta que inicie sesión nuevamente.

3.3.2. Sección de navegación

La sección de navegación está comprendida por siete opciones: registros, personalidad, enseñar, subir aiml, corrector ortográfico, buscar / editar aiml y *demo chat*. Al igual que la sección de pestañas se puede acceder en cualquier momento a estas opciones desde cualquier página del administrador.

3.3.2.1. Registros

La página de registros mantiene una bitácora de todas las conversaciones realizadas por los usuarios. La bitácora proporciona detalles como fecha y hora de la conversación y un registro completo de las preguntas y respuestas. Dada la naturaleza de la aplicación y la cantidad de información que es registrada por la bitácora se dispone de un filtro que organiza a través del tiempo las conversaciones ayudando al administrador a recuperar información relevante en un período de tiempo seleccionado.

La aplicación del filtro consiste en dos pasos: primero se debe seleccionar el período de tiempo, el cual puede ser uno de los siguientes: últimos 20, hoy, semana previa, 2 semanas previas, mes previo, últimos 6 meses, últimos 12 meses y todos los tiempos. Como segundo paso se debe hacer clic en el botón *show*, esta acción aplica el filtro a todas las conversaciones existentes en la bitácora listando del lado derecho únicamente las existentes durante ese período de tiempo.

Figura 12. **Página de registros – administrador**



Fuente: elaboración propia, con programa de PhpStorm.

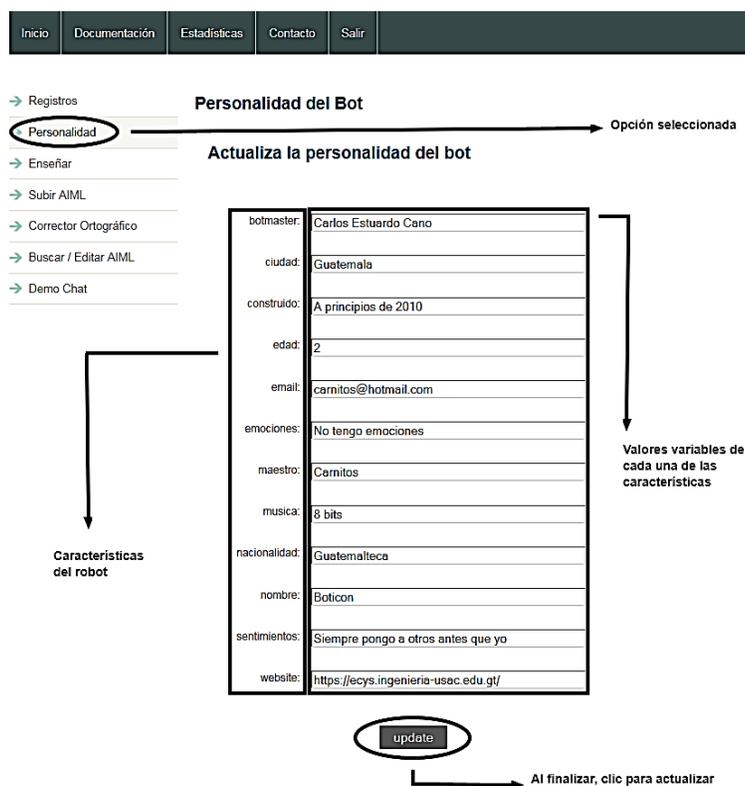
3.3.2.2. Personalidad

La página de personalidad permite la modificación por parte del administrador sobre el conjunto de características del robot de charla. Las características son constantes y el valor de cada una de ellas es variable adecuándose a las necesidades de la Escuela y a la región en la que habitan sus usuarios. Las características pueden estar ligadas a las mismas respuestas que da el robot en las conversaciones por lo que se recomienda ser extremadamente cauteloso al cambiar sus valores. Debido a esto no es posible agregar o eliminar características desde el área de administración, y en caso se

deban agregar, eliminar o modificar es el encargado del desarrollo del proyecto o el programador en cuestión quién debe consultar el capítulo 5 en donde se explica técnicamente que se necesita para cumplir con ese cometido.

Actualizar la información de la personalidad es tan sencillo como editar en valor de las cajas de texto y finalizar dando clic al botón update. No se recomienda por ningún motivo actualizar las características con valores vacíos porque limitará al robot en la calidad de información que muestre a sus usuarios. Si un campo no se llegara a utilizar en ninguna conversación entonces se puede considerar eliminarlo, de lo contrario es mejor dejarlos.

Figura 13. **Página de personalidad –administrador**



Fuente: elaboración propia, con programa de PhpStorm.

3.3.2.3. Enseñar

La página de enseñar permite la creación de nuevo contenido en el cerebro del robot de charla utilizando el formato estándar AIML, el cual se basa en cuatro campos importantes: tópico, previa respuesta, entrada de usuario y respuesta del *bot*. En cada uno de los campos es obligatorio escribir sin faltas de ortografía cada una de las palabras.

Además, es su responsabilidad revisar que cada palabra que use tilde, eñe y/o diéresis exista en el diccionario ortográfico del robot, de no hacerlo así el robot no podrá ser capaz de encontrar patrones de preguntas y por consiguiente no podrá retornar la información al usuario. Consulte el apartado 3.3.2.5 para obtener consejos y aprender a ingresar palabras al diccionario. Consulte el capítulo 4 si aún no conoce cómo manipular el lenguaje AIML.

El campo tópico significa que el *bot* sólo podrá acceder a esta respuesta si una categoría AIML anterior ha puesto este tema como el tema actual. En la gran mayoría de los casos, este campo siempre estará vacío.

El campo previa respuesta significa que el *bot* debe haber respondido con este texto en su última respuesta. Se debe escribir siempre en mayúsculas.

El campo entrada usuario es la entrada que los usuarios utilizan para acceder a las respuestas del *bot*. Se debe escribir siempre en mayúsculas.

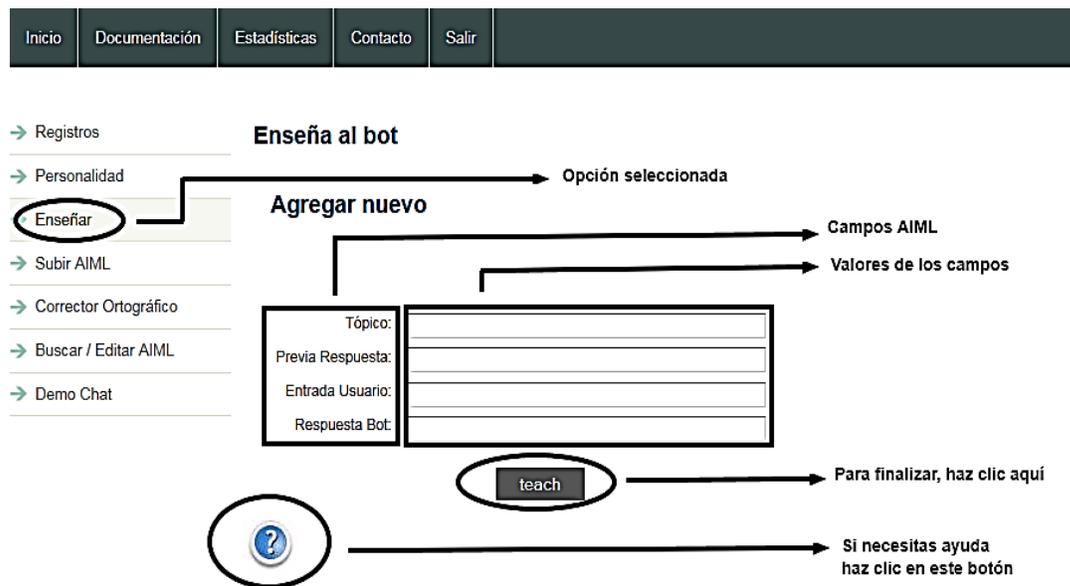
El campo respuesta *bot* es el texto que el *bot* despliega cuando el campo entrada usuario ha sido encontrado. La respuesta se escribe en mayúsculas y minúsculas, tal como se desee mostrar en pantalla.

A continuación algunos ejemplos:

- Ejemplo simple:
Entrada de usuario: HOLA COMO ESTÁS
Respuesta del *bot*: Yo estoy bien gracias y tú?
- Ejemplo intermedio:
Previa respuesta del *bot*: YO ESTOY BIEN GRACIAS Y TU
Entrada de usuario: BIEN GRACIAS
Respuesta del *bot*: Eso es muy bueno escuchar

Se puede encontrar una gran cantidad de información en la red acerca de cómo escribir AIML bien formados. Si desea obtener más información acerca de la escritura AIML visite <http://www.alicebot.org/documentation/aiml-primer.html>.

Figura 14. **Página de enseñanza – administrador**



Fuente: elaboración propia, con programa de PhpStorm.

3.3.2.4. Subir AIML

La página de subir archivos AIML permite la carga de un conjunto de enseñanzas almacenadas en un archivo de texto plano de tipo XML. Crear archivos de contenido AIML tiene ventajas sobre la creación de contenido directo:

- Respaldo de información, si se desea migrar la información de un tipo de robot de charla a otro en el que ambos comparten la misma información es tan simple como buscar el archivo AIML y solicitar al sistema que cargue automáticamente la información.
- Mejor comprensión, crear contenido directamente desde el administrador utilizando la opción enseñar es recomendable siempre que se desee crear una sola entrada de información o cuando se disponga de poco tiempo. Por el contrario y en la mayoría de los casos cuando inicia el proceso de retroalimentar del robot es necesario crear muchas entradas, al hacerlo en un archivo de texto plano permite comprender y organizar de manera eficiente toda la información que se está gestionando porque el archivo AIML posee una estructura jerárquica igual que los campos utilizados en la página enseñar.

Subir archivos AIML es tan fácil como hacer clic en el botón *Browse*, buscar y seleccionar el archivo .AIML que se desee cargar al sistema y finalizar haciendo clic en el botón *upload*. Cualquier error relacionado con el archivo o con su estructura será desplegado. Si aún no sabes crear archivos AIML se recomienda que consulte el capítulo 4.

Si tiene problemas al cargar archivos AIML consulta en la siguiente lista común de errores:

- El archivo AIML tiene un tamaño muy grande
- Comentarios dentro del archivo no son permitidos
- Asegurarse que el archivo que se está intentando subir tiene privilegios de lectura/escritura en él (CHMOD 755).
- Reemplazo de todo por encima de la primer <category> con una simple etiqueta <aiml>.

Además puede encontrar una gran cantidad de información en la red acerca de cómo escribir AIML bien formados. Si desea obtener más información acerca de la escritura de archivos AIML visite la siguiente página web: <http://www.alicebot.org/documentation/aiml-primer.html>.

Figura 15. **Página de carga de archivos AIML – administrador**



Fuente: elaboración propia, con programa de PhpStorm.

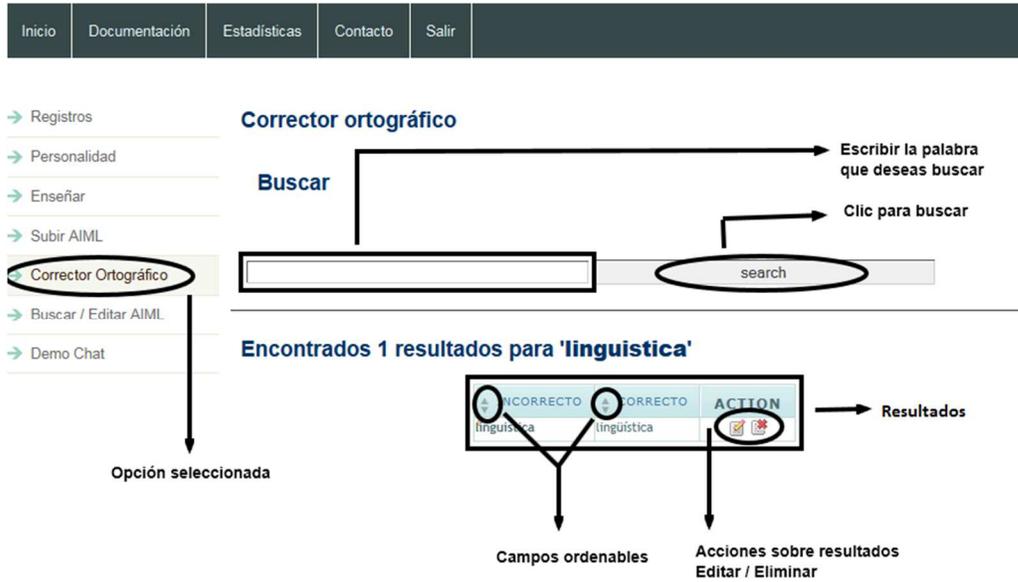
3.3.2.5. Corrector ortográfico

La página del corrector ortográfico permite buscar, agregar y modificar palabras con el fin de detectar y corregir de forma automática faltas ortográficas. Esta herramienta es indispensable para crear contenido de calidad porque sin un diccionario de palabras adecuado, el robot hará diferencia entre palabras como “lingüística” y “lingüística”, generando deficiencias para encontrar patrones de preguntas y por consiguiente no existirá información que mostrar al usuario.

Pero se debe tener extremo cuidado porque agregar cualquier palabra que se tilde no es la regla, suponiendo que cuando el usuario pregunte ¿Cómo te llamas? el robot debe contestar Mi nombre es Boticon, un error sería no pensar que la palabra como con y sin tilde es dependiente del contexto en el que se utilice: cómo en una pregunta lleva tilde y en una oración puede no llevarla, yo como mucho. Ya que el robot no es capaz de entender el contexto en el que se desarrolla la conversación se debe evitar ingresar la palabra como en el corrector ortográfico, y dependerá totalmente del usuario el escribir correctamente la palabra. Es decir, la palabra debe ser agregada al diccionario sí y sólo sí la palabra se tilda sin importar el contexto por ejemplo, lingüística.

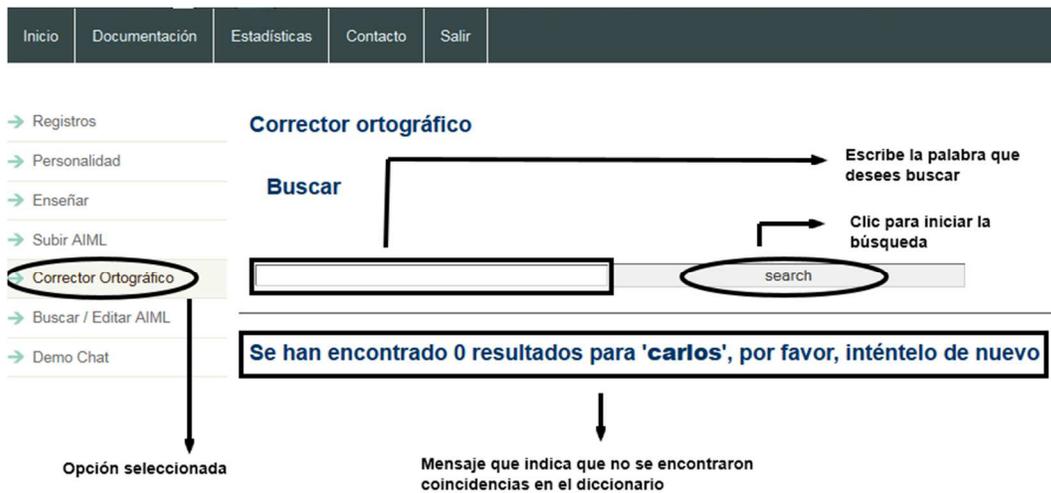
Conociendo las reglas para agregar las palabras al corrector, los pasos a seguir son: buscar la palabra que se va a agregar (este paso es recomendado, sin embargo no es obligatorio), si el buscador lista la palabra entonces no será necesario agregarla al diccionario. Al encontrarla aparecen dos opciones: editar la palabra y/o eliminarla. Si se editan aparecerán dos campos que permiten actualizar la entrada al diccionario. Si se elimina un mensaje advierte si está seguro, pudiendo cancelar la acción.

Figura 16. **Página del corrector ortográfico – administrador**



Fuente: elaboración propia, con programa de PhpStorm.

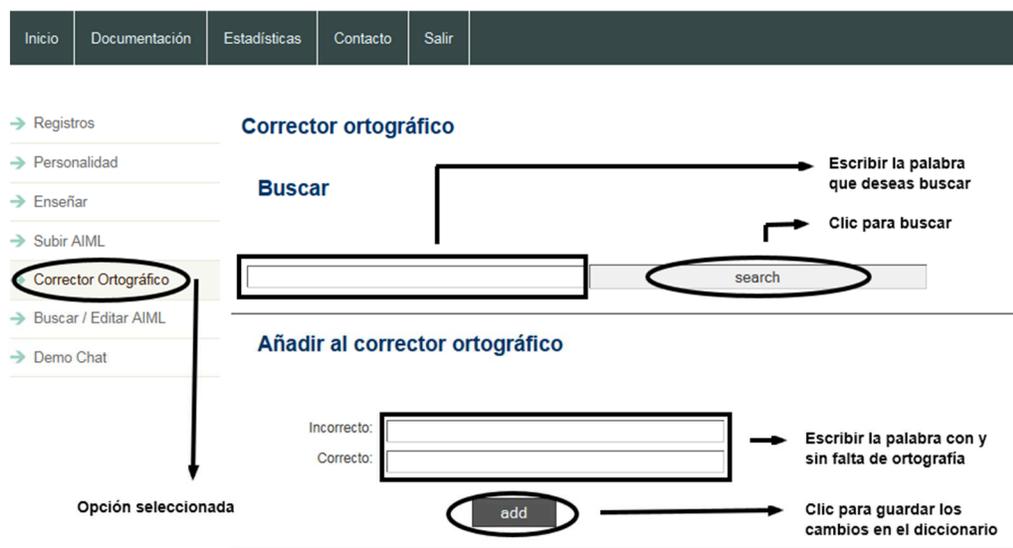
Figura 17. **Página del corrector ortográfico – resultados**



Fuente: elaboración propia, con programa de PhpStorm.

Si no existe ninguna entrada de la palabra que deseas agregar al diccionario, haz clic nuevamente en corrector ortográfico y procede a escribir la palabra con y sin error ortográfico en los campos de texto correspondientes. Finaliza la operación haciendo clic en *add*.

Figura 18. **Página del corrector ortográfico - añadir**



Fuente: elaboración propia, con programa de PhpStorm.

3.3.2.6. **Buscar / Editar AIML**

La página de buscar y editar entradas AIML permite la modificación y eliminación de un conjunto de enseñanzas almacenadas en el cerebro del Robot de Charla. Consulte el capítulo 4 si aún no conoce como manipular el lenguaje AIML. El proceso es muy sencillo: primero se escribe la palabra que se está buscando y luego ejecuta la búsqueda presionando el botón *search*, de encontrarse resultados serán listados en una tabla que posee columnas

ordenables ascendente y descendente: tópico, previa respuesta, respuesta *bot*, archivo y acción.

Estas cuatro columnas coinciden con la estructura AIML: *topic*, *that*, *pattern* y *template* respectivamente. Para cada una de las filas de la tabla tiene la opción de editar y/o eliminar, si edita la entrada se mostrarán cinco campos de los que dispone para actualizar, si elimina la entrada un cuadro de información le pregunta si está seguro o desea cancelar la acción.

Figura 19. **Página del editor AIML – administrador**

Buscar y Editar AIML

Acciones sobre resultados (Editar / Eliminar)

Buscar

Escribe la palabra que desees buscar

Clic para iniciar la búsqueda

search

Encontrados 30 resultados para 'linguística'

Resultados encontrados

TÓPICO	PREVIA RESPUESTA	ENTRADA USUARIO	RESPUESTA BOT	ARCHIVO	ACCIÓN
		CUÁL ES EL CORREO DE LA OFICINA DE LINGÜÍSTICA	linguistica@ing.usac.edu.gt	gestiones_linguistica.aiml	[icon] [icon]
		A QUE HORAS ATIENDEN EN LA OFICINA DE LINGÜÍSTICA	CUÁLES SON LOS HORARIOS DE ATENCIÓN DE LA OFICINA DE LINGÜÍSTICA	gestiones_linguistica.aiml	[icon] [icon]
		CUÁLES SON LOS HORARIOS DE ATENCIÓN DE LA OFICINA DE LINGÜÍSTICA	Matutino: martes, jueves y viernes de 10:00 a 12:00 horas. Vespertino: martes y jueves: de 16:00 a 19:00 (lun, mie, vie; sólo con cita).	gestiones_linguistica.aiml	[icon] [icon]
		EN DONDE PUEDO	SE DEBE PEDIR CITA PARA ENTREGAR EL	gestiones_linguistica.aiml	[icon] [icon]

Opción seleccionada

Resultados

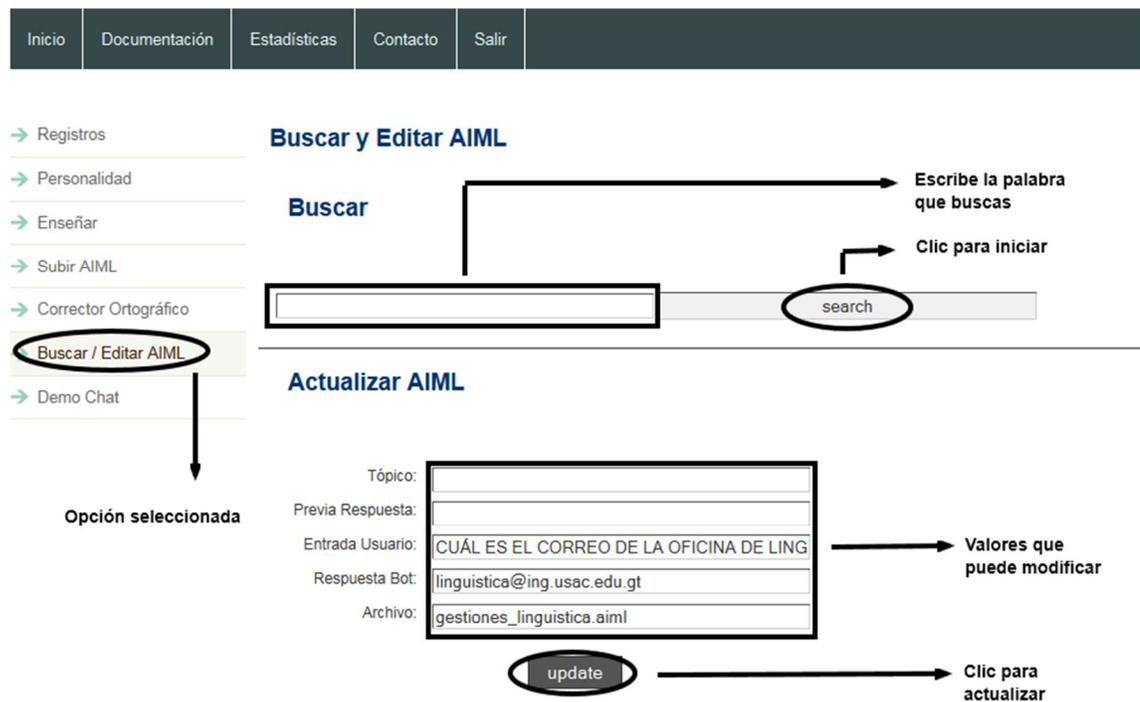
Columnas ordenables

Fuente: elaboración propia, con programa de PhpStorm.

El campo Entrada Usuario siempre debe contener valores en mayúsculas. Es lo mismo que utilizar la etiqueta *<pattern>* del archivo AIML.

El campo Respuesta *Bot* guarda el valor de la respuesta del robot y la mostrará tal cual lo haya guardado, incluyendo mayúsculas y minúsculas. Es lo mismo que utilizar la etiqueta `<template>` del archivo AIML.

Figura 20. **Página del editor AIML – actualizar**



Fuente: elaboración propia, con programa de PhpStorm.

3.3.2.7. *Demo chat*

La página de *demo chat* permite charlar con el robot sin afectar las estadísticas y registros del mismo. El Robot de Charla se comporta de la misma manera como se comporta con los usuarios.

4. CREANDO CONOCIMIENTO EN EL CEREBRO DEL ROBOT DE CHARLA HACIENDO USO DEL LENGUAJE AIML

Información general sobre el lenguaje AIML (*Artificial Intelligence Mark-up Language*) permite a las personas crear conocimiento en los robots basados en la tecnología de software libre ALICE.

AIML fue desarrollado por el doctor Richard S. Wallace y la comunidad del software libre durante el período 1995-2000. Fue adaptado originalmente de una gramática que no es del tipo XML también llamado AIML y formó la base para el primer Alicebot llamado A.L.I.C.E.

AIML, describe una clase de objetos de datos llamados objetos AIML y en parte describe el comportamiento de los programas de computadora que los procesan. Los objetos AIML están formados por unidades llamadas tópicos o temas y categorías.

4.1. Categorías

La unidad básica de conocimiento en AIML es llamada categoría. Cada categoría se compone de una pregunta de entrada, una respuesta de salida, y un contexto opcional. La pregunta, o estímulo, se llama patrón o *<pattern>*. A la respuesta, se le conoce como plantilla o *<template>*. Los dos tipos de contexto opcionales se denominan eso o *<that>* y tópico o *<topic>*. El lenguaje de patrones AIML es simple, consiste solamente de palabras, espacios y los símbolos de comodín *_* y ***. Las palabras pueden consistir de letras y números, pero no de otros caracteres.

El lenguaje de patrones es de caso invariante (no es sensible a mayúsculas o minúsculas). Las palabras están separadas por un espacio y los caracteres comodines funcionan como las palabras.

Las primeras versiones de AIML sólo permiten un carácter comodín por patrón. El AIML 1.01 estándar permite múltiples comodines en cada patrón, pero el lenguaje está diseñado para ser tan simple como sea posible, más sencillo incluso que las expresiones regulares. La plantilla es la respuesta AIML. En su forma más simple, la plantilla se compone de sólo texto sin formato.

En términos más generales, las etiquetas AIML transforman la respuesta en un mini programa de Computadora que puede guardar información, activar otros programas, dar respuestas condicionales y de forma recursiva llamar al reconocimiento de patrones para insertar las respuestas de otras categorías.

La parte opcional del contexto de la categoría se compone de dos variantes, llamadas *<that>* y *<topic>*. La etiqueta *<that>* aparece dentro de la categoría y su patrón de expresión debe coincidir con la última declaración del robot. Recordando que una última declaración es importante si el robot hace una pregunta. La etiqueta *<topic>* aparece fuera de la categoría, y recoge un conjunto de categorías juntas. El tema se puede establecer dentro de cualquier etiqueta *<template>*.

AIML no es exactamente lo mismo que una simple base de datos de preguntas y respuestas. La coincidencia de patrones de consulta del lenguaje es mucho más simple que algo como SQL. Pero una plantilla de categoría puede contener la etiqueta recursiva *<srail>*, por lo que la salida no depende sólo de una categoría encontrada, sino también de cualquier otra forma recursiva llegada a través de *<srail>*.

4.1.1. Recursión

AIML implementa recursividad con el operador `<srai>`. No existe acuerdo sobre el significado de las siglas. El A.I. es sinónimo de inteligencia artificial, pero S.R. puede significar estímulo-respuesta, reescritura sintáctica, reducción simbólica, recursividad simple o resolución sinónima. El desacuerdo sobre el acrónimo refleja la variedad de aplicaciones para `<srai>` en AIML. Cada uno de estos se describe con más detalle a continuación:

4.1.1.1. Reducción simbólica

Se refiere al proceso de simplificación de formas gramaticales complejas en otras más simples. Usualmente, los patrones atómicos en las categorías que almacenan el conocimiento del robot se expresan en los términos más simples posibles, por ejemplo, tendemos a preferirlos patrones como ¿Quiénes Sócrates? a aquellos como ¿Sabe usted quién es Sócrates?, cuando almacenamos información biográfica acerca de Sócrates.

Muchas de las formas más complejas se reducen a formas más simples utilizando categorías AIML diseñadas para la reducción simbólica:

```
<category>
<pattern>   SABE   USTED   QUIEN   ES   *   </pattern>
<template> <srai> QUIEN ES <star/> </srai> </template>
</category>
```

Cualquiera que sea la entrada que coincide con este patrón, la parte ligada al comodín * puede ser insertado en la respuesta con la etiqueta `<star/>`.

Esta categoría reduce cualquier entrada de la forma ¿Sabe usted quién es X? a ¿Quién es X?, donde X es cualquier palabra ingresada por el usuario.

4.1.1.2. División y conquista

Muchas oraciones individuales se pueden reducir adoso más oraciones subsecuentes y la respuesta formada por la combinación de cada una de las respuestas. Una frase que comienza con la palabra Sí, por ejemplo, si tiene más de una palabra, puede ser tratada como la sub oración Sí más lo que le sigue.

```
<category>
<pattern>SI *</pattern>
<template><srai>SI</srai><sr/></template>
</category>
```

La etiqueta <sr/> es simplemente una abreviatura de las etiquetas <srai>, <star/> o </srai>.

4.1.1.3. Sinónimos

El estándar AIML1.01 no permite más de un patrón por cada categoría. Los sinónimos son tal vez la aplicación más común de la etiqueta <srai>. Muchas maneras de decir lo mismo se puede reducir a una categoría, quien es la que contiene la respuesta.

```
<category>
<pattern>HOLA</pattern>
<template>Que tal!</template>
```

```
</category>
```

```
<category>
```

```
<pattern>HEY</pattern>
```

```
<template><srai>HOLA</srai></template>
```

```
</category>
```

```
<category>
```

```
<pattern>BUENAS</pattern>
```

```
<template><srai>HOLA</srai></template>
```

```
</category>
```

```
<category>
```

```
<pattern>WENAS</pattern>
```

```
<template><srai>HOLA</srai></template>
```

```
</category>
```

4.1.1.4. Corrección de ortografía y gramática

El error ortográfico más común es el uso de las contracciones. Una pequeña cantidad de contexto gramatical suele ser necesaria para capturar este error.

```
<category>
```

```
<pattern>ESCRIBÍ UNA CARTA AL *</pattern>
```

```
<template>Creo que quieres decir
```

```
<srai>ESCRIBÍ UNA CARTA A EL <star/></srai>
```

```
</template>
```

```
</category>
```

Aquí el robot corrige la entrada del usuario y actúa como un tutor del idioma.

4.1.1.5. Palabras clave

Con frecuencia sería oportuno escribir una plantilla de AIML que se activa por la aparición de una palabra clave en cualquier lugar de la frase de entrada. El formato general de las cuatro categorías AIML se ilustra con este ejemplo tomado del robot de charla ELIZA:

```
<category>
<pattern>MAMÁ</pattern>
<template>Cuéntame más sobre tú familia. </template>
</category>
```

```
<category>
<pattern>_ MAMÁ</pattern>
<template><srai>MAMÁ</srai></template>
</category>
```

```
<category>
<pattern>MAMÁ _</pattern>
<template><srai>MAMÁ</srai></template>
</category>
```

```
<category>
<pattern>_ MAMÁ *</pattern>
<template><srai>MAMÁ</srai></template>
</category>
```

La primera categoría detecta la palabra clave que aparece por sí misma y proporciona la respuesta genérica. La segunda categoría detecta la palabra clave como el sufijo de una oración. El tercero la detecta como el prefijo de una frase de entrada, y finalmente la última categoría detecta la palabra clave como un infijo. En cada una de las tres últimas categorías la etiqueta <srai> se utiliza para vincular a la primera categoría, de modo que los cuatro casos producen la misma respuesta, pero necesita ser escrito y almacenado sólo una vez.

4.1.1.6. Condicionales

Es posible escribir etiquetas condicionales en AIML, utilizando sólo la etiqueta <srai>. Se debe tener en cuenta tres categorías.

```
<category>
<pattern>QUIÉN ES EL</pattern>
<template><srai>QUIEN ES EL<getname=el/></srai></template>
</category>
```

```
<category>
<pattern>QUIEN ES EL *</pattern>
<template>Él es<getname=el/>.</template>
</category>
```

```
<category>
<pattern>QUIEN ES EL UNKNOWN</pattern>
<template>No sé quién es él.</template>
</category>
```

A condición de que él se inicializa a *unknown*, las categorías ejecutan un salto condicional en función de si él se ha establecido. Para comodidad del botmaster, AIML también proporciona la función equivalente a través de la etiqueta <condition>.

4.1.2. Focalización

Las personalidades de los robots son creadas y mejoradas a través de un proceso cíclico de aprendizaje supervisado llamado focalización. La focalización es un ciclo de incorporación entre el usuario, el robot y el botmaster, en donde las entradas de los usuarios que no encuentran coincidencias completas entre las categorías se registran por el *bot* y se entregan como focalización al botmaster, que crea las respuestas adecuadas, a partir de las consultas más comunes. El ciclo de focalización produce una personalidad cada vez más refinada para el robot.

El arte de escribir AIML es más evidente en la creación de categorías predeterminadas, que ofrecen respuestas evasivas a una amplia gama de insumos.

4.1.3. Contexto

La etiqueta <that> en AIML se refiere al enunciado previo del robot. En concreto, si el robot responde con una frase de múltiples oraciones, el valor de la etiqueta <that> se establece en la última oración de la secuencia. La elección de la etiqueta <that> está motivado por su uso en el lenguaje ordinario, a continuación algunos ejemplos:

```

<category>
<pattern>Sí</pattern>
<that>TE GUSTAN LAS PELÍCULAS</that>
<template>¿Cuál es tu película favorita?</template>
</category>

```

Esta categoría se activa cuando el cliente dice que sí. El robot debe averiguar a que está diciendo el usuario sí. Si el robot le preguntó: ¿Te gustan las películas?, esta categoría concuerda, y la respuesta, ¿cuál es tu película favorita?, continúa la conversación en el mismo sentido.

Un ejemplo interesante del uso de la etiqueta *<that>* son categorías que habilitan al robot a responder bromas como tnoc-tnoc. Las categorías:

```

<category>
<pattern>TNOCTNOC</pattern>
<template>¿Quién es?</template>
</category>

```

```

<category>
<pattern>*</pattern>
<that>¿QUIÉN ES?</that>
<template><person/>quién?</template>
</category>

```

```

<category>
<pattern>*</pattern>
<that>* QUIÉN</that>
<template>Jaja muy gracioso, <getname=nombre/>.</template>

```

</category>

Produciría el siguiente diálogo:

Usuario: Tnoc tnoc.

Robot: ¿Quién es?

Usuario: Miguelito.

Robot: ¿Miguelito quién?

Usuario: El ratón Miguelito.

Robot: Jaja muy gracioso, Miguelito.

Internamente, el intérprete AIML almacena el patrón de entrada, el patrón *that* y el patrón *topic* a lo largo de una trayectoria única, como: INPUT *<that>* THAT *<topic>*. Cuando los valores de *<that>* o *<topic>* no se especifican, el programa establece implícitamente los valores del correspondiente patrón *THAT* o *TOPIC* al comodín*.

La primera parte de la ruta a coincidir es la entrada. Si más de una categoría tienen el mismo patrón de entrada, el programa puede distinguir entre ellos en función del valor de la etiqueta *<that>*. Si dos o más categorías tienen la misma etiqueta *<pattern>* y la misma etiqueta *<that>*, el último paso es elegir la respuesta basada en la etiqueta *<topic>*.

Esta estructura sugiere una regla de diseño: nunca utilice la etiqueta *<that>* a menos que haya escrito dos categorías con la misma etiqueta *<pattern>*, y nunca use la etiqueta *<topic>* a menos que escriba dos categorías con la misma etiqueta *<pattern>* y *<that>*. Sin embargo, una de las aplicaciones más útiles para la etiqueta *<topic>* es crear sujetos dependientes como:

```
<topic name=CARROS>
<category>
<pattern>*</pattern>
<template>
<random>
<li>¿Cuál es tu carro favorito?</li>
<li>¿Qué tipo de carro manejas?</li>
<li>¿Obtienes muchas multas de la municipalidad?</li>
</random>
</template>
```

Teniendo en cuenta la enorme cantidad de cosas que las personas podrían decir que son gramaticalmente correctas o semánticamente significativas, el número de cosas que la gente realmente dice es sorprendentemente pequeño. Steven Pinker, en su libro *Cómo funciona la mente* escribió.

Se supone que tienes diez opciones para la primera palabra para comenzar una oración, diez para la segunda palabra (obteniendo cien inicios de dos palabras), diez para la tercera palabra (obteniendo mil inicios de tres palabras), y así sucesivamente. (Diez es, de hecho, la media geométrica aproximada del número de opciones de palabras disponibles en cada momento que se construye una oración gramatical y sensible). Un poco de aritmética muestra que el número de oraciones de 20 palabras o menos (no es una inusual) es de aproximadamente 1 020.

Afortunadamente para los programadores de robots de *chat*, los cálculos de Pinker están muy lejos. Los experimentos con A.L.I.C.E. indican que el número de opciones para la primera palabra es más de diez, pero sólo

alrededor de dos mil. En concreto, alrededor de 2 000 palabras cubren el 95% de todas las entradas de las primeras palabras de A.L.I.C.E. El número de opciones para la segunda palabra es de sólo dos. Para estar seguro, hay algunas primeras palabras (yo y usted, por ejemplo) que tienen muchas segundas palabras posibles, pero el promedio general es de menos de dos palabras. El promedio disminuye en el factor de ramificación con cada palabra sucesiva.

4.2. Creando conocimiento desde un archivo AIML

Crear contenido en español correctamente desde un archivo AIML es un proceso poco documentado debido a que la gran mayoría de robots de charla son desarrollados en inglés por la popularidad mundial del idioma y la comodidad de manejar un conjunto de caracteres que no deben ser tildados, o utilicen modificaciones en las letras (eñe, diéresis, etc.) para ser escritos correctamente.

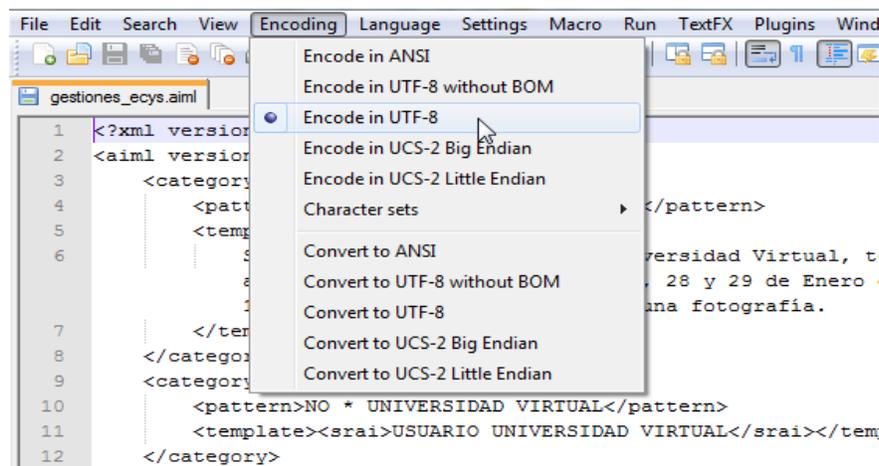
4.2.1. Reglas para crear un correcto contenido en español

Si se necesita crear contenido desde un archivo AIML es indispensable como primer paso instalar un editor de texto plano avanzado, por ejemplo: *Notepad++* o *Textpad* tienen excelente reputación entre las personas que se dedican a las tecnologías de la información por la cantidad de opciones, complementos, facilidad de uso y capacidad de manejar opciones avanzadas sobre textos planos.

Luego de abrir el editor de texto avanzado (en este caso se mostraran algunas capturas de pantalla utilizando *Notepad++*). Se creará un nuevo archivo con un tipo de codificación utf-8, inmediatamente después se guarda el archivo

con un nombre cualquiera y una extensión AIML, por ejemplo: gestiones_ecys.aiml.

Figura 21. **Seleccionar el tipo de codificación UTF-8 – Notepad++**



Fuente: elaboración propia, con programa de PhpStorm.

Notepad++ puede crear formatos de colores para diferentes tipos de archivos. AIML es un tipo de archivo basado en XML, ir al menú de opciones, seleccionar Lenguaje -> XML, inmediatamente asignará colores adecuados que brindan una mejor lectura y comprensión del archivo. Existen 3 líneas que siempre se deben escribir antes de empezar a crear contenido en lenguaje AIML.

```
<?xml version=1.0 encoding=utf-8?>  
<aiml version=1.0>  
</aiml>
```

La primera línea indica al *parser* de archivos AIML que el tipo de caracteres a interpretar corresponde al juego de caracteres UTF-8 (entre los

que se encuentran las tildes, diéresis, etc.). Es vital que se indique la codificación en ambos: en el tipo de archivo y dentro del código AIML. Dentro de las etiquetas <aiml> se inicia a escribir el contenido deseado, sin olvidar la información general sobre archivos AIML descrita en la sección 4.1.

Figura 22. Observaciones generales sobre archivos AIML en español

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <aiml version="1.0">
3   <category>
4     <pattern>USUARIO UNIVERSIDAD VIRTUAL</pattern>
5     <template>
6       Si necesitas usuario para la Universidad Virtual, te
7       estaremos atendiendo en el salón DSI el 27, 28 y 29 de
8       Enero de 9:00 a 12:00. Te tomaremos tus datos y una
9       fotografía.
10    </template>
11  </category>
12  <category>
13    <pattern>NO * UNIVERSIDAD VIRTUAL</pattern>
14    <template><srai>USUARIO UNIVERSIDAD VIRTUAL</srai></template>
15  </category>
16  <category>
17    <pattern>* NUEVO * UNIVERSIDAD VIRTUAL</pattern>
18    <template><srai>USUARIO UNIVERSIDAD VIRTUAL</srai></template>
19  </category>
20 </aiml>
```

length: 639 lines: 18 Ln: 18 Col: 1 Sel: 0 Dos\Windows UTF-8 INS

Fuente: elaboración propia, con programa de PhpStorm.

En la primera línea se resalta el tipo de codificación a utilizar, las líneas 4 y 11 fueron resaltadas para hacer énfasis que el contenido escrito dentro de esas etiquetas siempre debe ir en mayúsculas y por último se ha resaltado en la parte inferior de la imagen la codificación del tipo de archivo, también UTF-8.

4.2.2. Cargando archivos AIML al cerebro del robot

Una vez terminado el contenido AIML se procede a cargar el contenido en el cerebro del robot de charla. Se abre una nueva ventana en el navegador *web* y se inicia sesión en <https://ecys.ingenieria-usac.edu.gt/boticon/admin/>.

Luego se consulta la sección 3.3.2.4. Si desea probar si la información que se enseñó al robot funciona adecuadamente se consulta la sección 3.3.2.7.

Si necesita archivos de recursos o ejemplos puede solicitar al administrador de la UV que los proporcione o bien puede consultar la sección 2.3.3. Y navegar hasta *Boticon \ admin \ resources*.

4.3. Creando conocimiento desde el área de administración

Tal como se ha mencionado en otros capítulos crear conocimiento desde el área de administración es un proceso más rápido si solo se desea agregar o modificar una entrada. No se recomienda utilizar este método si el *botmaster* quiere cargar demasiada información. Consulte la sección 3.3.2.3.

5. BOTICÓN EL ROBOT DE CHARLA PARA LA ESCUELA DE INGENIERÍA EN CIENCIAS Y SISTEMAS

La característica del robot de charla permite a sus usuarios utilizarlo como una poderosa herramienta informativa.

Puede ser consultado por cualquier persona, desde cualquier dispositivo que soporte de un navegador *web* con acceso a *internet*.

Por la naturaleza de la mensajería instantánea y la capacidad de *internet* de estar disponible en cualquier momento en cualquier lugar, el robot de charla reduce los tiempos de atención para los usuarios.

Ofrece un apoyo informativo para el personal docente y administrativo, evitando la tediosa tarea de repetir una y otra vez la misma información y aprovechando ese tiempo para sus respectivas labores.

- Permite un número ilimitado de operadores, conversaciones y usuarios
- Facilidad de acceso a los contenidos y actividades informativas

El robot de charla le permite al estudiante eliminar la necesidad de desplazamientos para llegar hasta la Escuela y poder informarse sobre las actividades, trámites, etc.

5.1. Modificaciones al proyecto Program-O

En estas secciones se describirán procesos sumamente técnicos escritos especialmente para desarrolladores y programadores que deseen conocer el funcionamiento a interno de la aplicación.

5.1.1. ¿Qué es Program-O?

Tal como se describe al inicio del capítulo 2, *Program-O* es un proyecto que nace a partir de un proyecto previo llamado *Program-E*. *Program-O* es un intérprete *AIML* escrito en *PHP* con *MySQL*, que permite hacer nuestros propios robots de charla y hospedarlos en un sitio *web*, el motor del *chat* está escrito en *PHP*. Si deseas consultar más información visita las siguientes direcciones: <http://www.program-o.com>, <http://sourceforge.net/projects/program-o/>.

5.1.2. ¿Por qué fue necesario modificar Program-O?

Porque el proyecto *Program-O* no soporta múltiples idiomas, fue pensado y desarrollado desde un principio para soportar únicamente caracteres del idioma inglés. La adaptación al idioma español conllevó cambios en la vista de las páginas *web*, cambios en la codificación de caracteres de la base de datos, correcciones a las consultas desde el *back-end* hacia la base de datos, correcciones al intérprete de archivos *AIML*, creación de archivos de contenido específicos para la Escuela de Sistemas, eliminar funcionalidad innecesaria, correcciones a deficiencias y fallos de programación, traducción al español de las páginas *web* del instalador, administrador y robot.

Para un detalle de todos los cambios realizados se recomienda ir a la sección 2.3.3.

5.1.3. Modificaciones realizadas al *front-end*

Las modificaciones se centraron básicamente en traducir el proceso de instalación, el administrador y mensajes de las páginas *web* para que sea comprendida por todos los estudiantes de la Escuela de Sistemas.

Una modificación muy importante sufrieron todas las páginas *web* estableciendo en el encabezado una codificación UTF-8 para que los caracteres pudieran ser legibles, escritos y desplegados correctamente.

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

5.1.4. Modificaciones visuales al diseño del sitio

Se creó una nueva página, más amigable, para que los usuarios interactúen con el robot de charla. Así mismo el logo de Boticón fue comprado y tiene derechos de *copyright* (condicionados), cualquier información adicional póngase en contacto con el autor.

En la medida de lo posible se intentó desacoplar la capa de vista de la capa *back-end* para acelerar las fases de mantenimiento.

5.1.5. Modificaciones realizadas a la base de datos

Las modificaciones no cambiaron el diseño, nombres de tablas, o campos de la base de datos como tal, únicamente se alteró la codificación de caracteres en las tablas para que pudieran soportar el idioma español.

5.1.6. Precargando información

Al momento de ejecutar el proceso de instalación del robot de charla todos los archivos que se encuentran en */admin/aimlsql/*.sql* serán cargados en la memoria del robot (Manual de instalación - paso 7). Los archivos del proyecto original fueron removidos y reemplazados por uno solo que brinda información característica de nuestro robot, este archivo está compuesto por un conjunto de sentencias INSERT que deben tener la sintaxis de MySQL para realizar exitosamente el ingreso de información en la tabla AIML.

```
INSERT INTO `aiml` VALUES(  
  AUTO_INDENT,  
  "AIML category rules",  
  "AIML pattern rules",  
  "AIML that rules",  
  "AIML template rules",  
  "AIML topic rules",  
  "AIML filename")
```

Se crearon 5 archivos de recursos AIML específicos para la Escuela de Sistemas, se pueden obtener en */admin/resources/*.aiml*. Estos archivos no se cargan automáticamente con la instalación, pero están preparados para ser cargados utilizando la opción "Subir AIML" desde el área de administración.

5.1.7. Modificaciones realizadas al *back-end*

El *back-end* de Program-O fue la capa que recibió los cambios de mayor trascendencia, muchos de estos cambios se deben a la necesidad de agregar soporte a caracteres del idioma español.

5.1.7.1. Parser para cargar archivos AIML

El *parser* de archivos se encarga de interpretar cada una de las palabras, etiquetas y estructura jerárquica del archivo en cuestión. La restricción de 75kb que se menciona en la ayuda se debe a que en algunas ocasiones cuando el archivo es muy grande; Apache aborta la conexión por *timeout* aunque el intérprete no haya terminado de cargar todo el archivo. Es recomendable analizar el ambiente de la Universidad Virtual con el fin de determinar el máximo tamaño permitido.

5.1.7.2. Integración con los estándares de la Universidad Virtual

Por recomendación del administrador de la Universidad Virtual, se implementó por defecto una contraseña fuerte de 16 bits, esta contraseña solo puede ser cambiada por el administrador de la UV.

5.1.7.3. Personalidad de Boticón

Los campos de la personalidad fueron creados especialmente para este robot de charla y difieren un poco respecto al proyecto original. La razón de eliminar estos campos es que mucha de la información no se estaba utilizando y así conforme se vaya necesitando se irán agregando más campos.

En caso se desee agregar más campos que definan la personalidad de Boticón, se debe insertar el / los registros necesarios en la tabla *botpersonality*. Automáticamente se desplegarán los cambios en pantalla cuando refresque la página.

```
INSERT INTO `programobot`.`botpersonality` (`id`, `bot`, `name`, `value`)
VALUES(
  AUTO_INCREMENT,
  1,
  'PERSONALITY_NAME,
  'VALUE')
```

Cualquier característica de la personalidad se puede utilizar desde la estructura AIML escribiendo `<botname="PERSONALITY_NAME"/>`.

Para soportar caracteres latinos se implementó la función `mysql_real_escape_string`, la cual escapa caracteres especiales en la cadena no escapada, teniendo en cuenta el conjunto de caracteres actual para que sea seguro usarla en `mysql_query()`.

5.1.7.4. Regionalización de la base de datos

Era importante utilizar la regionalización adecuada para el área geográfica de Boticón, porque las estadísticas, inicio de sesiones, etc. llevan estampadas la hora y fecha que el proyecto indique. El valor predeterminado de la base de datos era Europa / Inglaterra, y la validación de la regionalización estaba mal implementada, causando siempre errores. Se corrigió la región y la validación.

```
@date_default_timezone_set('America/Guatemala');
```

Otro error ocasionado por una mala asignación de la región se encontraba en el inicio de sesión, generando información incorrecta en las fechas y horas en que iniciaba y finalizaba sesión el administrador del sistema.

5.1.7.5. Eliminación de *bugs*

Se aplicaron actualizaciones menores en distintas áreas de código, siendo una de las más importantes el inicio de sesión donde personas podían explotar un fallo de seguridad y entrar al área de administración sin previa autorización.

Conversión a minúsculas en enlaces de etiquetas `<a href>`, esta conversión a minúsculas generaba enlaces rotos, muchas páginas en el mundo (incluyendo a la Universidad Virtual) capitalizan las letras y Apache no es capaz de encontrar los enlaces si es sensible al contexto. Este *bug* se eliminó permitiendo al *botmaster* escribir los *links* tal y como son.

Otro importante error de programación que se corrigió en la función *CleanText* es la cual eliminaba caracteres latinos por considerarlos errores de escritura por parte del usuario, ya sea al momento de cargar un archivo AIML o guardarlo desde el administrador.

```
$that = preg_replace('/[^a-z0-9áéíóúÁÉÍÓÚñÑüÿ]/i', '', $that);
```

Las estadísticas de conversaciones mostraban errores de conceptualización que ocasionaba al administrador incongruencia de información respecto a los datos reales.

```
$sql = "SELECT count(subq.inputs) AS TOT FROM (SELECT count(date(timestamp)) AS INPUTS, date(timestamp) AS DATE FROM `\$dbn`.`conversation_log` $sqladd GROUP BY date ORDER BY date) AS SUBQ";
```

5.1.8. Mejoras en general

La cantidad de líneas por conversación se amplió de 1 a 5 permitiendo un máximo de 13 líneas, brindando al usuario una mejor comprensión de la conversación con Boticon.

```
$convoLines = 5;
```

Creación de un archivo de modificación de directivas de Apache que obliga al servidor *web* a no mostrar *warnings* al usuario y administrador del sistema. Ya que ellos no tienen por qué estar enterados de estos mensajes.

```
php_valueerror_reporting 0
```

Instrucciones de instalación claras. Se mejoraron los mensajes de instalación y se creó un manual de instalación con mensajes claros y que acciones llevar a cabo en caso se presenten errores.

5.2. Automatización para los procesos de la Universidad Virtual

Con base en el archivo de entrada que provee control académico a la Escuela de Sistemas, se muestra a continuación un ejemplo de cómo se debe analizar el archivo para crear contenido en el robot de charla.

5.2.1. Creación de contenido para horario de cursos

La creación de contenido para horario de cursos, permite analizar la siguiente tabla de información.

Tabla II. **Horario del segundo semestre para el curso IPC1**

770	IPC1	A	:10	:50		X	X			009900038	CESAR BATZ SAQUIMUX
770	IPC1	B	:10	:50		X	X			020071281	NEFTALI DE JESUS CALDERO MENDEZ
770	IPC1	C	:10	:50		X	X			990000045	WALTER E. MINCHEZ SUTUC
770	IPC1	D	:10	:50		X	X			020040258	JULIO CESAR ESCOBAR DE LEON

Fuente: elaboración propia, con programa Excel de Microsoft Office.

Se desea entrenar al robot de charla para que brinde a los estudiantes esa información. Haciendo uso de los conceptos mencionados en el capítulo 4 se procede a crear un archivo AIML con las siguientes categorías:

```
<?xml version="1.0" encoding="utf-8"?>
<aiml version="1.0">
<category>
<pattern>INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1
SEGUNDO SEMESTRE 2011</pattern>
<template>El horario del curso Introducción a la Programación y
```

Computación 1 con el código 0770 es: Martes y Jueves de 7:10 a 8:50.

En la sección A imparte clases el Ing. CésarBatzSaquimux.

En la sección B la Inga. Neftalí de Jesús Calderón Mendez.

En la sección C el Ing. Walter E. MinchezSutuc.

En la sección D el Ing. Julio César Escobar De León.

</template>

</category>

<category>

<pattern> HORARIO * INTRODUCCIÓN A LA PROGRAMACIÓN Y
COMPUTACIÓN 1 SEGUNDO SEMESTRE 2011</pattern>*

*<template><srai>INTRODUCCIÓN A LA PROGRAMACIÓN Y
COMPUTACIÓN 1 SEGUNDO SEMESTRE 2011</srai></template>*

</category>

</aiml>

5.3. ¿Qué hacer en caso pierda la contraseña?

Se sugiere ponerse en contacto inmediatamente con el administrador de la Universidad Virtual explicando su situación. Si ha perdido la contraseña y no hay forma de recuperarla tendrán que asignarle una nueva contraseña codificada en MD5.

CONCLUSIONES

1. El contenido en español desde un archivo AIML es un proceso poco documentado, debido a que la gran mayoría de robots de charla son desarrollados en inglés por la popularidad mundial del idioma y la comodidad de manejar un conjunto de caracteres que no deben ser tildados o utilicen modificaciones en las letras (eñe, diéresis, etc.) para ser escritos correctamente.
2. Crear contenido desde archivos AIML tiene mayores ventajas que desventajas, pero es necesario seguir los pasos establecidos para una correcta carga e interpretación.
3. Es mejor utilizar un robot de charla del tipo débil porque los robots del tipo fuerte necesitan una extensa cantidad de horas de tiempo de aprendizaje para almacenar suficiente información y mantener una conversación, además son susceptibles a aprender malos hábitos de las personas que les hablan ya que muchas abusan y les enseñan cosas sucias y/o malintencionadas, entorpeciendo el proceso de aprendizaje y enseñando conversaciones superficiales que consisten en comentarios de gente que previamente hizo.
4. Hasta el momento no es posible que un robot de charla sea capaz de seguir una conversación más o menos lógica a menos que se utilicen frases coherentes y de fácil comprensión.

5. En la actualidad los robots de charla se pueden encontrar en varios nichos de internet, pero principalmente en las páginas web y la mensajería instantánea. El beneficio de utilizar estos robots de charla es brindar información puntual de manera rápida, es por eso que portales en línea como *PayPal* los utilizan para ofrecer soporte a sus clientes.

6. Este trabajo es una adaptación al español del motor de *chat* e intérprete de archivos AIML del conocido proyecto *Program-O*. Fue escrito en PHP utilizando MySQL como motor de base de datos, está compuesto por un intérprete de archivos AIML el cual proporciona preguntas y respuestas al cerebro del robot y un núcleo de reglas que lo adhieren a las especificaciones e interfaces de *chat*.

7. La página de estadística muestra información relevante sobre el número de conversaciones y líneas de *chat* por sesión de usuario. Es importante para el administrador del robot saber interpretar esta información: un número alto de conversaciones no quiere decir necesariamente que los usuarios estén usando al robot como herramienta informativa y no hay que esperar que las estadísticas de visitas sean altas mientras el robot no tenga un amplio contenido que informar, incluso es recomendable que el robot de charla posea bastante conocimiento sobre eventos triviales y de la vida diaria si se desea que sea popular entre los usuarios.

RECOMENDACIONES

1. Instalar códigos de rastreo de páginas *web* para tener un mejor control del comportamiento de navegación de los visitantes en el sistema, ya que las estadísticas del robot de charla no brindan ese tipo de información enriquecida. Información sobre hábitos y tiempos de interacción que permita identificar características que gusten a los usuarios o que permitan detectar posibles fallas en el servicio ya sea por contenido de baja calidad o falta del mismo.
2. Hacer crecer la popularidad del robot de charla, lanzando el servicio a través de protocolos de mensajería instantánea que hagan uso del *back-end* de este proyecto. Estos protocolos son ampliamente utilizados en dispositivos móviles por lo que brindan facilidad de acceso y portabilidad a los usuarios del sistema.
3. Mejorar el registro de conversaciones del área de administración con información relevante en cuanto se tenga disponible, por ejemplo: además de mostrar el Id del usuario se podría mostrar su nombre en caso se lo haya hecho saber al robot de charla, mejorando la usabilidad de los filtros a través del tiempo.
4. Utilizar un patrón de desarrollo Modelo – Vista – Controlador, que facilite el mantenimiento de código y permita ciclos de programación más rápidos. La implementación de este patrón también ayudaría a facilitar el lanzamiento del robot de charla a través de los protocolos de mensajería instantánea mencionados en el inciso 2.

5. Siempre que se vaya a crear, modificar o eliminar información en archivos AIML se debe utilizar un editor de archivos de texto plano avanzado como notepad++ o textpad, y un teclado con configuración regional en español para una escritura más rápida de caracteres latinos.
6. Siempre que se modifique información del robot de charla verificar que el cambio no afecte el comportamiento de otros patrones de información en el cerebro del robot de charla.
7. Eliminar los archivos de instalación para evitar que piratas informáticos reescriban la configuración del sistema del robot de charla y eliminen información de manera intencionada.
8. Implementar un sistema de cache que mejore el rendimiento de la base de datos, acelerando la velocidad de respuesta de preguntas de los usuarios.
9. Consultar constantemente los foros de *Program-O* en búsqueda de parches y errores, ser cauteloso con los cambios y probarlos en un ambiente de desarrollo antes de lanzarlos a producción.
10. Ejecutar pruebas con archivos AIML del tamaño máximo que el servidor Apache permite cargar hacia el servidor de la Universidad Virtual, con el fin de evitar pérdida de información y de integridad de los datos en el cerebro del robot de charla.
11. Implementar un sistema seguro de recuperación de contraseña para el área de administración del robot de charla.

12. Realizar una fase de control de calidad para eliminar cualquier deficiencia o fallo en el sistema actual

13. Elaborar pruebas de satisfacción entre los estudiantes, hasta alcanzar una aceptación alta, con el fin de obtener la mayor satisfacción posible desde su primer uso, esto generará que los mismos estudiantes hagan popular al servicio, entusiasmándose a seguir utilizándolo en una futura ocasión y proponerlo entre sus compañeros y amigos.

BIBLIOGRAFÍA

1. FITRIANIE, Siska. *My Eliza A multimodal Communication System*. [en línea]. <<http://www.kbs.twi.tudelft.nl/Publications/MSc/2002-Fitriani-MSc.html>>. [Consulta: 2 de enero de 2012].
2. TURING, Alan. M. *Computing Machinery and Intelligence*. [en línea]. <<http://www.csee.umbc.edu/courses/471/papers/turing.pdf>>. [Consulta: 15 de febrero de 2012].
3. WALLACE, Richard S. *The Elements of AIML Style*. [en línea]. <<https://files.ifi.uzh.ch/cl/hess/classes/seminare/chatbots/style.pdf>>. [Consulta: 22 de marzo de 2012].
4. WEIZENBAUM, Joseph. *ELIZA. A computer Program for the Study of natural Language Communication Between Man And Machine*. [en línea]. <<http://www.cse.buffalo.edu/~rapaport/572/S02/weizenbaum.eliza.1966.pdf>>. [Consulta: 28 de marzo de 2012].
5. WELLMAN, Dan. *jQuery UI 1.6 – The User Interface Library for jQuery*. [en línea]. <<http://www.packtpub.com/jquery-ui-16-user-interface-library-jquery/book>>. ISBN: 1847195121. [Consulta: 7 de junio de 2012].

APÉNDICE

Apéndice I. **Manual de Instalación del robot de charla – Boticón**

Tiempo aproximado de instalación: 15 minutos.

- Creación del esquema: se deberá crear un nuevo esquema en la base de datos que tenga por nombre “programobot”.

```
mysql>createschemaprogramobot;
```

- Creación de usuario: es necesario crear una cuenta con permisos para ejecutar sentencias: insert, delete, update, select y createtable.

```
mysql> create user carlos identified by 'nuPru?*y-ru5Uk*t';
```

```
mysql> grant create, insert, delete, update, select on programobot.* to 'carlos' identified by 'nuPru?*y-ru5Uk*t';
```

```
mysql>flushprivileges;
```

- Configuración de usuario y base de datos en los archivos de instalación:

Ajustar el archivo de configuración /admin/funcs/config.php

Ajustar el archivo de configuración /bot/config.php

```
$dbh = "localhost"; //server location (localhost should be ok for this)
```

```
$dbn = "programobot"; //databasename/prefix
```

```
$dbu = "carlos"; //databaseusername
```

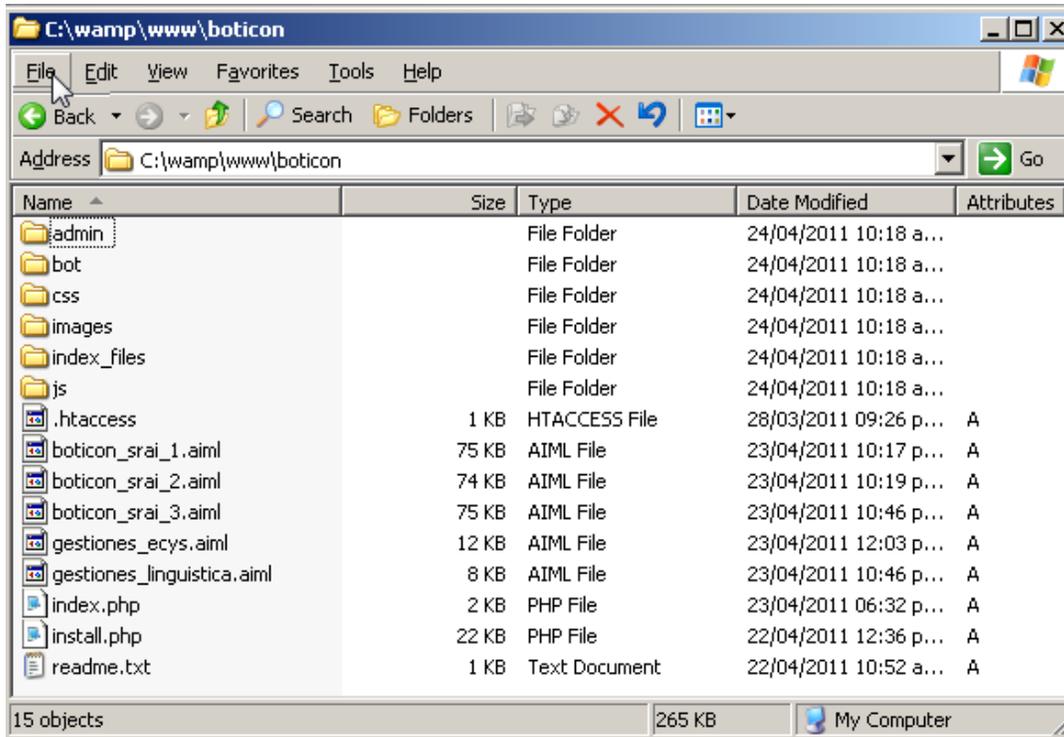
```
$dbp = "nuPru?*y-ru5Uk*t"; //databasepassword
```

Continuación apéndice I.

- Copiar archivos fuente al servidor:

Los archivos de esta aplicación son independientes de otras páginas por lo que se puede y es muy recomendable crear una carpeta dentro de www que identifique a este proyecto, por ejemplo: boticon. Luego de crear la carpeta proceder a copiar todos los archivos contenidos dentro de “boticon.zip”.

Archivos que conforman el proyecto



Name	Size	Type	Date Modified	Attributes
admin		File Folder	24/04/2011 10:18 a...	
bot		File Folder	24/04/2011 10:18 a...	
css		File Folder	24/04/2011 10:18 a...	
images		File Folder	24/04/2011 10:18 a...	
index_files		File Folder	24/04/2011 10:18 a...	
js		File Folder	24/04/2011 10:18 a...	
.htaccess	1 KB	HTACCESS File	28/03/2011 09:26 p...	A
boticon_srai_1.aiml	75 KB	AIML File	23/04/2011 10:17 p...	A
boticon_srai_2.aiml	74 KB	AIML File	23/04/2011 10:19 p...	A
boticon_srai_3.aiml	75 KB	AIML File	23/04/2011 10:46 p...	A
gestionecys.aiml	12 KB	AIML File	23/04/2011 12:03 p...	A
gestionelinguistica.aiml	8 KB	AIML File	23/04/2011 10:46 p...	A
index.php	2 KB	PHP File	23/04/2011 06:32 p...	A
install.php	22 KB	PHP File	22/04/2011 12:36 p...	A
readme.txt	1 KB	Text Document	22/04/2011 10:52 a...	A

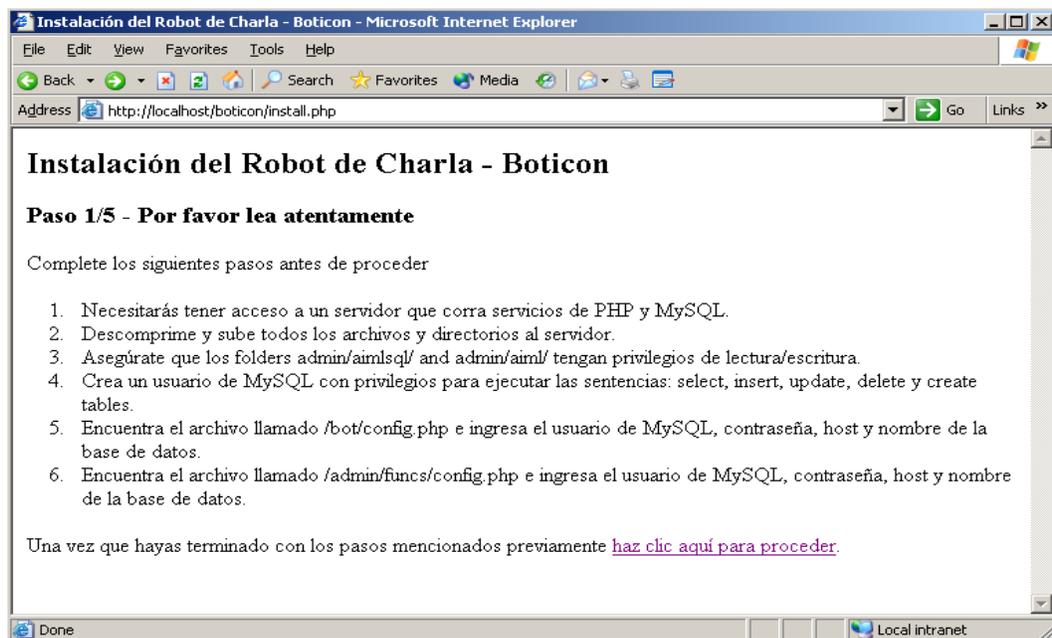
Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del *bot* – Paso 1: las siguientes capturas de pantalla muestran una instalación exitosa, cualquier mensaje de error que pueda suceder se lo hará saber el instalador. Abrir una nueva pestaña del navegador, e ir a esta dirección:

<http://localhost/boticon/install.php>

Primer paso de instalación del robot

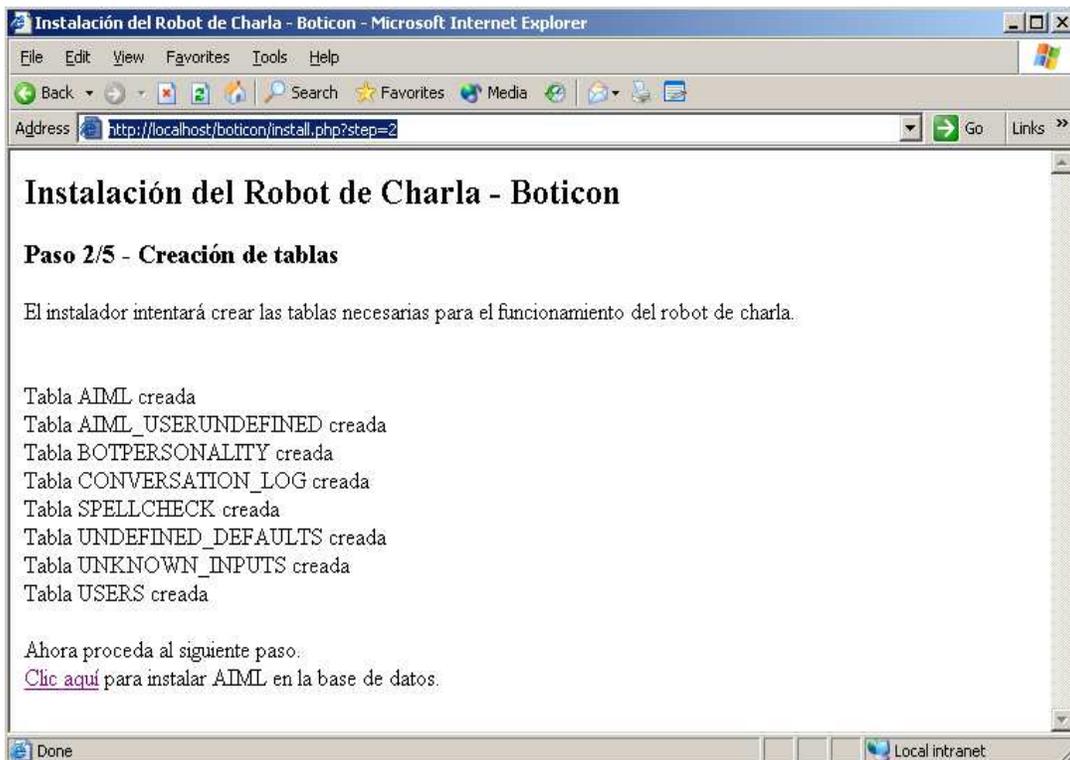


Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del *bot* – Paso 2: el asistente creará 8 tablas.

Segundo paso de instalación del robot

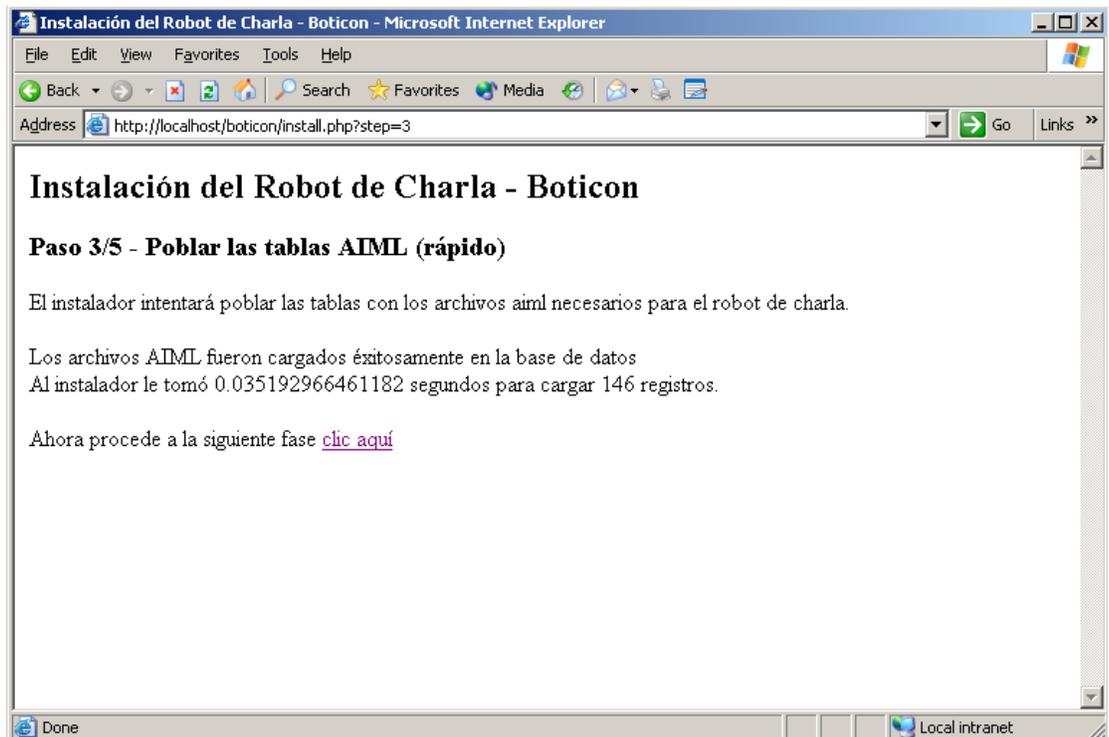


Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del *bot* – Paso 3 y 4: carga de datos a las tablas.

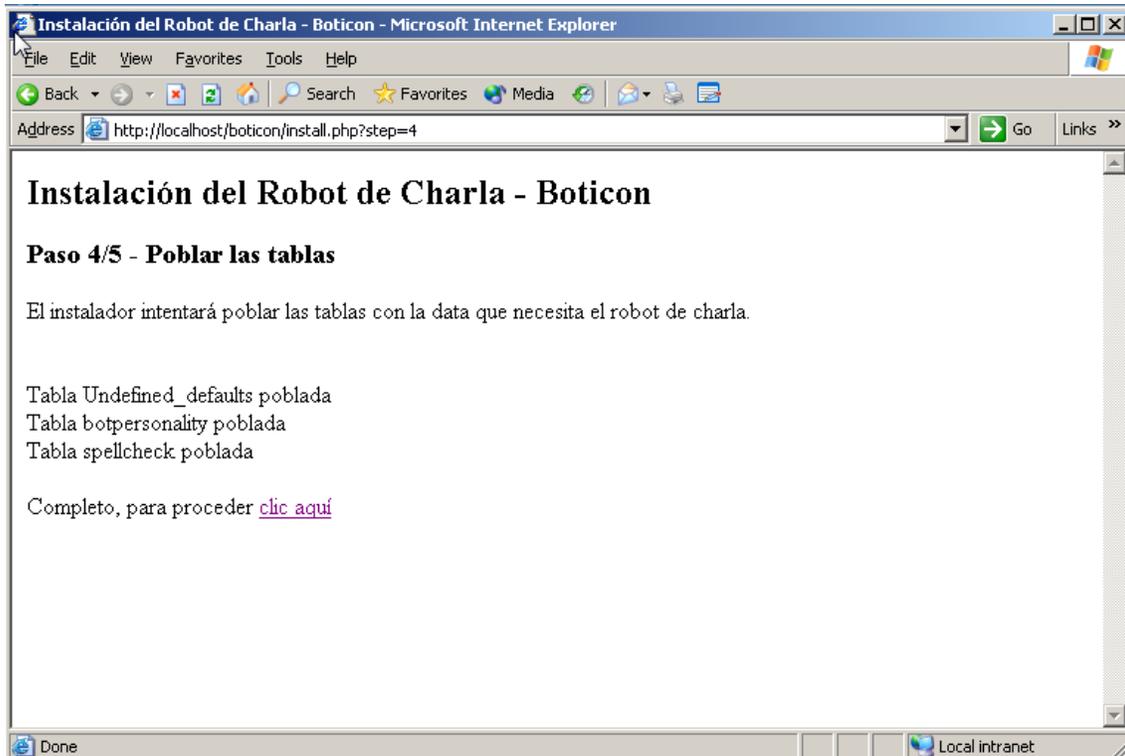
Tercer paso de instalación del robot



Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

Cuarto paso de instalación del robot



Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del *bot* – Paso 5: resumen de instalación. Proceda a instalar el administrador haciendo clic en el penúltimo enlace.

Quinto paso de instalación del robot



Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del administrador del *bot* – Paso 1: Después de instalar el *bot* será necesario instalar también el área de administración del mismo. Puede ejecutar este instalador haciendo clic en el penúltimo enlace del paso anterior o bien escribiendo en una pestaña: http://localhost/boticon/admin/install_admin.php

Primer paso de instalación del administrador

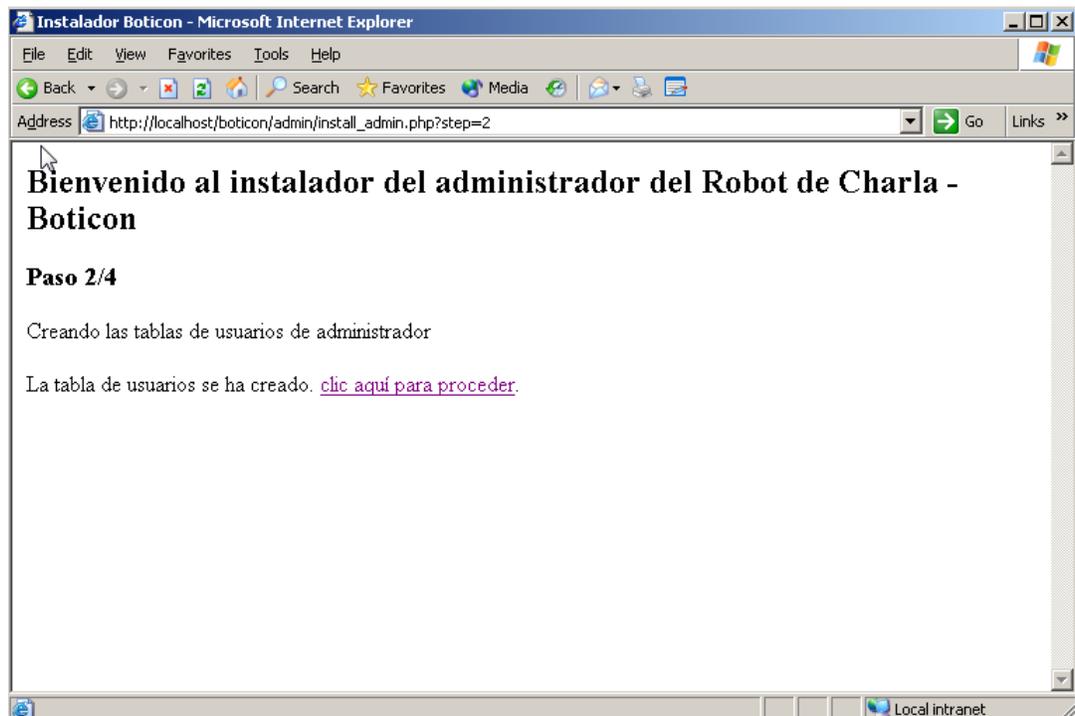


Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del administrador del *bot* – Paso 2: creación de la tabla de administrador.

Segundo paso de instalación del administrador

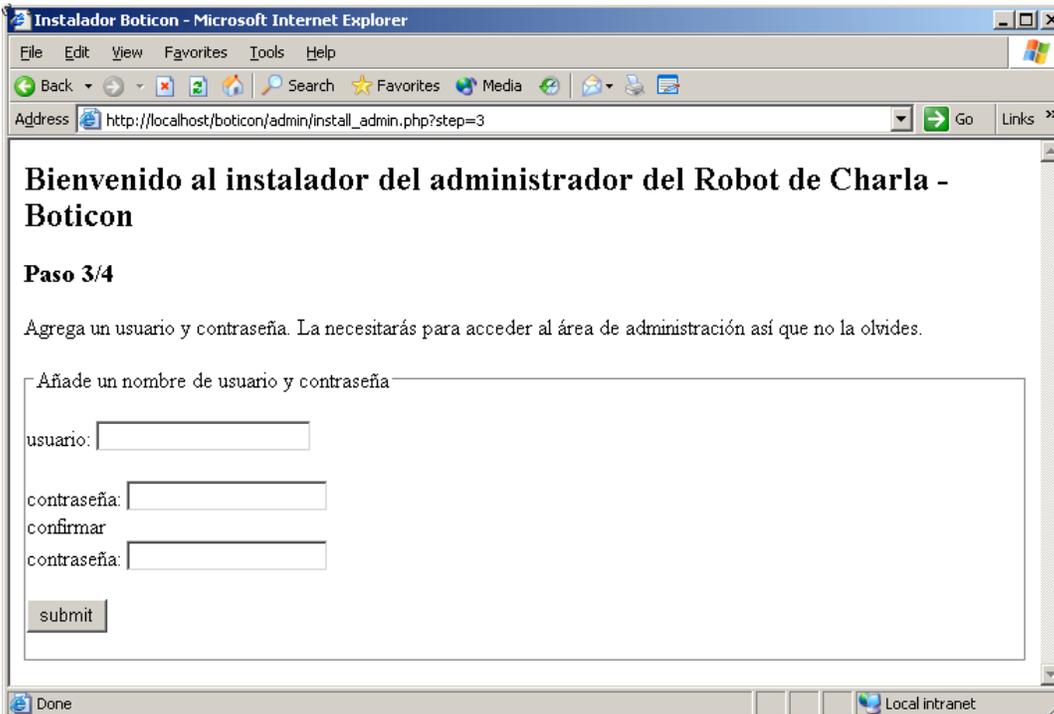


Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del administrador del *bot* – Paso 3: creación de acceso para el área de administración. El campo de usuario no deberá exceder los 10 caracteres. Por ejemplo, usuario: botecys, contraseña: sUHAr2swu-edap=W

Tercer paso de instalación del administrador



Instalador Boticon - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address http://localhost/boticon/admin/install_admin.php?step=3 Go Links >>

Bienvenido al instalador del administrador del Robot de Charla - Boticon

Paso 3/4

Agrega un usuario y contraseña. La necesitarás para acceder al área de administración así que no la olvides.

Añade un nombre de usuario y contraseña

usuario:

contraseña:

confirmar
contraseña:

Done Local intranet

Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del administrador del *bot* – Paso 3a: Luego de ingresar el usuario y confirmar las contraseñas verá una pantalla como la siguiente.

Tercer paso de instalación del administrador



Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Asistente de instalación del administrador del *bot* – Paso 4: revisión de instalación, de ser exitosa verá un mensaje como el siguiente.

Cuarto paso de instalación del administrador

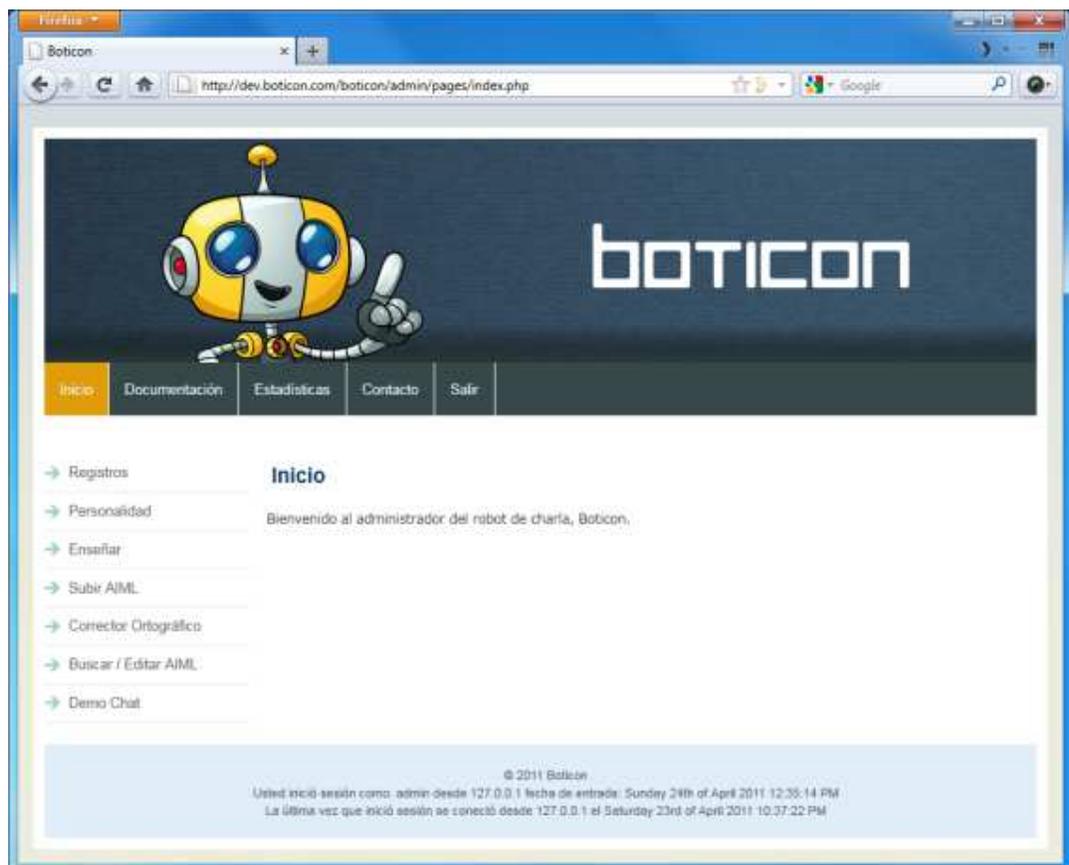


Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Accediendo al área de administración: ir al área de administración <http://localhost/boticon/admin/index.php> e ingresar el usuario y contraseña creados en el paso 11, de ser correcta la combinación de usuario/contraseña verá una página como la siguiente.

Página de bienvenida del administrador

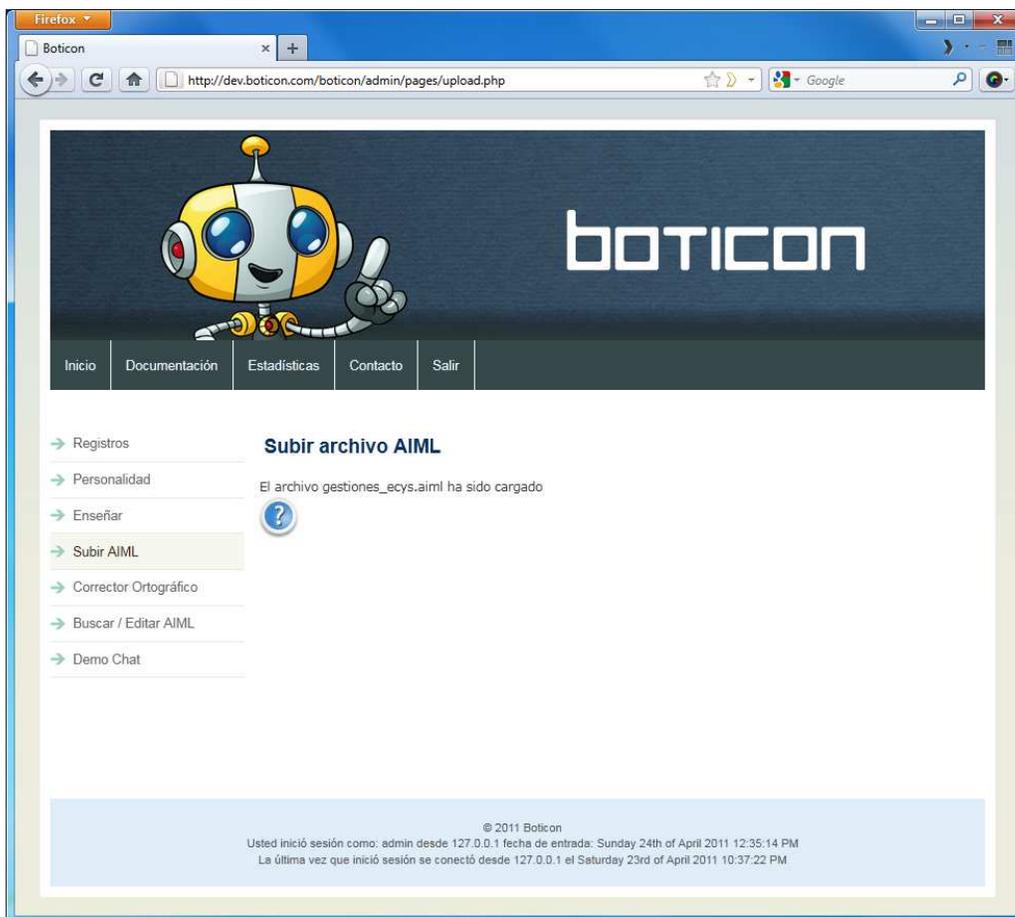


Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Cargando archivos para el cerebro del *bot* (opcional): vaya a la pestaña de Inicio, haga clic en “Subir AIML” y cargue manualmente los archivos .aiml que se encuentra dentro de la carpeta “admin/resources”. Finalice haciendo clic en cargar, un mensaje le informará si el archivo ha sido cargado con éxito.

Página de carga de archivos AIML

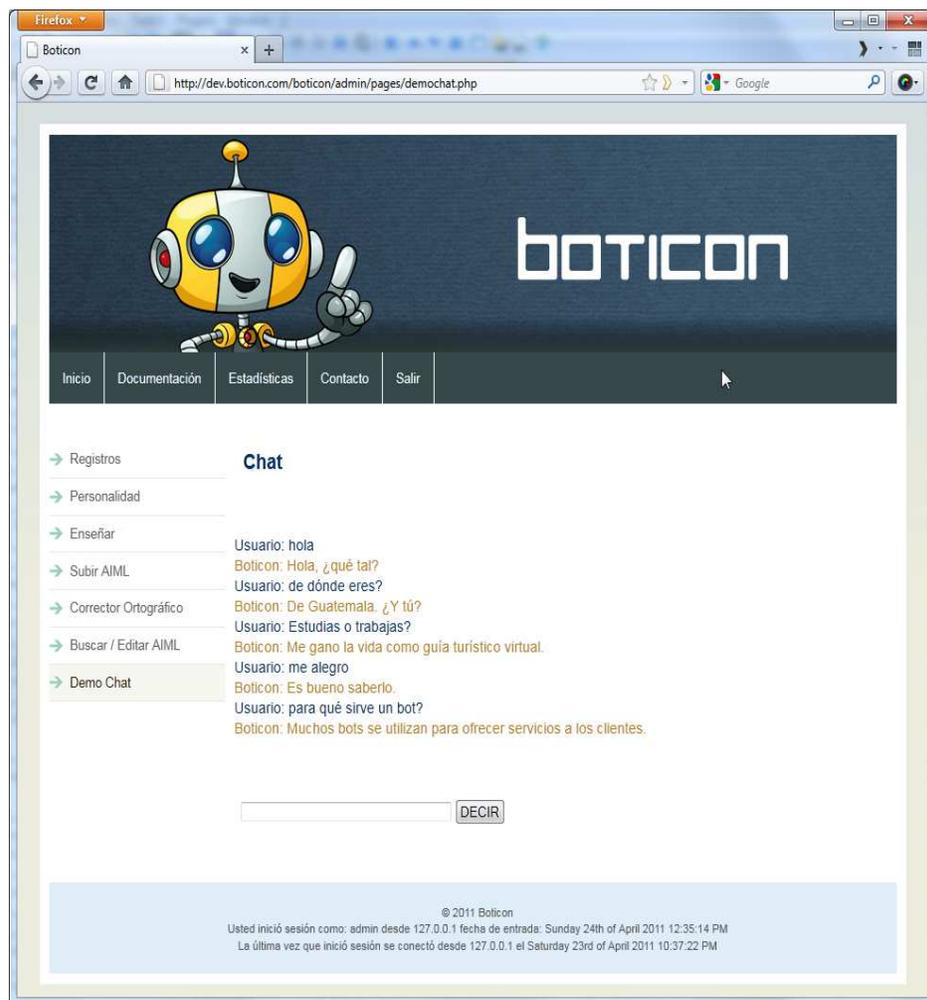


Fuente: elaboración propia, con programa de PhpStorm.

Continuación apéndice I.

- Verificando el funcionamiento del *bot*: las pruebas las puede hacer desde el área de administración <http://localhost/boticon/admin/pages/index.php> o fuera de ella <http://localhost/boticon/> Si cargaste los archivos del ejemplo puedes escribir algunas de esas preguntas en la caja de texto.

Página para chatear con el robot



Fuente: elaboración propia, con programa de PhpStorm.

