

Entorno de desarrollo de bots basados en ontologías para la moderación de debates

Desarrollo de un Bot para moderar debates de carácter político

Jaime Ramos Romero



Máster en Inteligencia Artificial Avanzada
Escuela Técnica Superior de Ingeniería Informática

Directores:

Ángeles Manjarrés Riesco

Simon Pickin

Autorización de Difusión y Uso

Autorizo a la Universidad Nacional de Educación a Distancia a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor, tanto la propia memoria, como el código, la documentación y/o el software desarrollado.

Jaime Ramos Romero



Copyleft by Jaime Ramos Romero, released under the license Creative Commons Attribution Share-Alike International 4.0 available at:

<https://creativecommons.org/licenses/by-sa/4.0/>

Agradecimientos

Después superar un año en el que las cosas no han ido como esperaban, al final se ha consigo sacar adelante este proyecto, cuya finalidad es mostrar los conocimientos adquiridos durante el desarrollo del máster y abrir nuevas oportunidades hacia la investigación.

Quiero expresar mi agradecimiento a todas aquellas personas que de alguna manera u otra me han apoyado para llegar hasta aquí, quienes me ayudaron a superar los baches y dificultades de este camino, aunque ahora no todas se encuentren aquí.

Destaco mi agradecimiento a los directores Ángeles y Simon, cuyo gran apoyo e interés por el proyecto han sido de gran ayuda para llevar a cabo este trabajo.

También me gustaría dar mi agradecimiento a amigos y familiares, pues con apoyo y comprensión sirvió de base para finalizar este trabajo.

Y por último, a todo aquél que quiera dedicar su tiempo para indagar en lo que se expone en estas páginas. Espero que pueda servirle de ayuda o transmitirle algún conocimiento que pueda derivar en otros trabajos.

“El hombre sabio no debe abstenerse de participar en el gobierno del Estado, pues es un delito renunciar a ser útil a los necesitados y un cobardía ceder el paso a los indignos.”

Epicteto de Frigia

Índice general

CAPÍTULO 1 - INTRODUCCIÓN.....	1
1.1 ANTECEDENTES.....	1
1.1.1 <i>Relación de la Informática y la Inteligencia Artificial en la Ciencia Política.....</i>	<i>1</i>
1.1.2 <i>Motivación.....</i>	<i>2</i>
1.1.3 <i>Visión general del Proyecto.....</i>	<i>3</i>
1.2 CONTEXTO.....	4
1.3 DEFINICIÓN DEL PROBLEMA E HIPÓTESIS DEL TRABAJO.....	5
1.4 OBJETIVOS DEL PROYECTO.....	5
1.5 METODOLOGÍA DE INVESTIGACIÓN.....	6
1.6 ESTRUCTURA DEL DOCUMENTO.....	8
CAPÍTULO 2 - ESTADO DEL ARTE.....	9
2.1 ONTOLOGÍAS.....	9
2.1.1 <i>Political Ontology – Colin Hay.....</i>	<i>9</i>
2.1.2 <i>Rawls’s political ontology.....</i>	<i>10</i>
2.1.3 <i>The political ontology of European integration.....</i>	<i>10</i>
2.1.4 <i>Constitution as Executive Order The Administrative State and the Political Ontology of “We the People”.....</i>	<i>10</i>
2.1.5 <i>Ontology-enabled E-Government Service Configuration - the OntoGov Approach.....</i>	<i>10</i>
2.1.6 <i>Ontologies for e-Government.....</i>	<i>11</i>
2.1.7 <i>Semantic Web Technologies for Information Management within e-Government Services.....</i>	<i>11</i>
2.1.8 <i>Integration of Government Services using Semantic Technologies.....</i>	<i>11</i>
2.1.9 <i>Un Modelo Ontológico para el Gobierno Electrónico.....</i>	<i>12</i>
2.2 EXTRACCIÓN DE INFORMACIÓN DE TEXTOS BASADA EN ONTOLOGÍAS.....	13
2.2.1 <i>Metodología para la población automática de ontologías: aplicación en los dominios de medicina y turismo.....</i>	<i>13</i>
2.2.2 <i>Los proyectos SINAMED e ISIS: Mejoras en el Acceso a la Información Biomédica mediante la integración de Generación de Resúmenes, Categorización Automática de Textos y Ontologías.....</i>	<i>14</i>
2.2.3 <i>Applying Ontologies for Web Text Mining Using Heterogeneous Information.....</i>	<i>14</i>
2.2.4 <i>Ontology learning from text: A look back and into the future.....</i>	<i>15</i>
2.2.5 <i>Evaluating techniques for learning non-taxonomic relationships of ontologies from text.....</i>	<i>15</i>
2.3 PARTICIPACIÓN POLÍTICA Y CIUDADANA.....	16
2.3.1 <i>DemoCritics.....</i>	<i>16</i>
2.3.2 <i>Poletika.org.....</i>	<i>17</i>
2.3.3 <i>Appgree.....</i>	<i>18</i>
2.3.4 <i>Agora Voting.....</i>	<i>19</i>
2.4 BOTS BASADOS EN ONTOLOGÍAS.....	20
2.4.1 <i>OntBot: Ontology based chatbot.....</i>	<i>20</i>
2.4.2 <i>Chatbots as Interface to Ontologies.....</i>	<i>21</i>
2.4.3 <i>MooBot & Nightbot – Twitch Chat Bot.....</i>	<i>21</i>
2.4.4 <i>PolitiBot.....</i>	<i>22</i>
2.5 CONCLUSIONES.....	23
2.5.1 <i>Herramientas de participación política y ciudadana.....</i>	<i>23</i>
2.5.2 <i>Ontologías relacionadas con la ciencia política.....</i>	<i>24</i>
2.5.3 <i>Bots basados en extracción de información basados en ontologías.....</i>	<i>25</i>

CAPÍTULO 3 - METODOLOGÍA.....	26
3.1 DISEÑO DE LA ONTOLOGÍA POLÍTICA.....	26
3.2 DESARROLLO DE @DEMOCRITICSBOT.....	27
3.3 INTEGRACIÓN DE LA BASE DE DATOS DE DEMOCRITICS CON @DEMOCRITICSBOT.....	27
3.4 USO DE SOFTWARE LIBRE.....	28
3.5 REPOSITORIO Y CONTROL DE VERSIONES.....	28
CAPÍTULO 4 - TECNOLOGÍAS DEL PROYECTO.....	29
4.1 TECNOLOGÍAS PARA ONTOLOGÍAS.....	30
4.1.1 OWL.....	30
4.1.2 protégé.....	30
4.2 TECNOLOGÍAS PARA EL BOT CONVERSACIONAL.....	30
4.2.1 Telegram Messenger.....	30
4.2.2 Telegram Bot API.....	31
4.2.3 Python.....	31
4.2.4 python-telegram-bot.....	32
4.2.5 WordNet.....	32
4.2.6 Natural Language Toolkit - NLTK.....	32
4.2.7 Cloud9.....	33
4.2.8 Git – GitHub.....	33
4.2.9 PHP.....	33
4.2.10 OntoSpy's.....	34
4.3 TECNOLOGÍAS PARA LA INTEGRACIÓN CON DEMOCRITICS.....	34
4.3.1 Laravel.....	34
4.3.2 MySQL.....	34
4.3.3 OpenShift.....	34
CAPÍTULO 5 - ARQUITECTURA DEL PROYECTO.....	36
5.1 TELEGRAM MTPROTO MOBILE PROTOCOLO.....	36
5.2 UNA ARQUITECTURA PARA BOTS MODERADORES-CONSULTORES BASADOS EN ONTOLOGÍAS.....	38
5.3 ARQUITECTURA DE LA APLICACIÓN.....	39
5.3.1 Telegram Bot API.....	40
5.3.2 python-telegram-bot.....	41
5.3.3 Domain-language-processing.....	43
5.3.4 API Rest.....	43
5.3.5 Ontology-connector.....	44
CAPÍTULO 6 - DISEÑO DE UNA ONTOLOGÍA POLÍTICA.....	45
6.1 ENTREVISTAS CON EXPERTOS.....	45
6.2 DIFERENTES PERSPECTIVAS EN UNA ONTOLOGÍA COMÚN.....	47
6.3 IDENTIFICACIÓN DE LOS TÉRMINOS.....	48
6.4 METODOLOGÍAS PARA EL DESARROLLO DE ONTOLOGÍAS.....	49
6.5 IMPLEMENTACIÓN EN PROTÉGÉ.....	50
CAPÍTULO 7 - DISEÑO DE @DEMOCRITICSBOT.....	51
7.0.1 Comandos personalizados.....	51
7.0.2 Comando /start.....	52
7.0.3 Comando /help.....	52
7.0.4 Comando /informacion.....	53
7.0.5 Comando /debate.....	54
7.0.6 Comando /about.....	54
7.0.7 Otros comandos por implementar.....	54
7.1 INTEGRACIÓN DE @DEMOCRITICSBOT CON LA BASE DE DATOS DE DEMOCRITICS.....	55

CAPÍTULO 8 - RESULTADOS Y TRABAJO FUTURO.....	57
CAPÍTULO 9 - CONCLUSIONES.....	59
BIBLIOGRAFÍA.....	61

Índice de figuras

Figura 1: Modelo de la Ontología de Gobierno Electrónico [10].....	12
Figura 2: Proceso para la conversión de la información en una ontología [51].....	15
Figura 3: Algunas capturas de DemoCritics[12].....	17
Figura 4: Poletika [13], Comparativa de los programas electorales.....	18
Figura 5: Algunas capturas de la aplicación Appgree [14].....	19
Figura 6: Diferentes capturas de una conversación privada con @politibot [18].....	22
Figura 7: Diferentes teclados virtuales con respuestas personalizadas [20].....	31
Figura 8: Tráfico de los datos en los chats normales mediante MTProto [54].....	36
Figura 9: Tráfico de los datos en los chats normales mediante MTProto [55].....	37
Figura 10: Arquitectura propuesta para el bot-moderador-consultor.....	39
Figura 11: Estructura general de la aplicación.....	40
Figura 12: Relación de clases en la Telegram Bot API.....	41
Figura 13: Arquitectura de Laravel, modelo MVC [56].....	43
Figura 14: Algunas clases de la ontología pertenecientes al sector terciario.....	45
Figura 15: Diferentes cargos de un político.....	46
Figura 16: Principales áreas o categorías del programa de Ahora Madrid.....	47
Figura 17: Tareas que componen la metodología Methontology [38].....	49
Figura 18: Ontología política implementada en protégé.....	50
Figura 19: Solicitando información a @DemoCriticsBot.....	52
Figura 20: Captura de @DemoCriticsBot desde Telegram Desktop.....	53
Figura 21: Estructura de la base de datos de DemoCritics [12].....	55

Índice de tablas

Tabla 1: Software libre utilizado en el desarrollo del proyecto.....	28
Tabla 2: Tecnologías utilizadas en el proyecto.....	29
Tabla 3: Componentes principales para la arquitectura del bot-moderador basado en ontologías.....	38

Resumen

Actualmente vivimos en una sociedad en constante cambio, con grandes avances que se producen a una velocidad mayor a la que estaban acostumbradas las sociedades de generaciones anteriores. Estos cambios han generado profundas alteraciones en un sistema político-social que parecía funcionar hasta ahora. En las últimas décadas, movimientos sociales emergidos por todo el mundo y una vertiginosa evolución de las tecnologías de la información y la comunicación han irrumpido para incorporar nuevas herramientas democráticas que fomentan la participación de la ciudadanía en política.

Este proyecto partirá sobre la idea de un Trabajo de Fin de Grado realizado en 2015; “*DemoCritics*, Aplicación Android de participación política con edición colaborativa en tiempo real”. El propósito de este proyecto fue incentivar la participación política de la ciudadanía a través de una aplicación que permite visualizar programas electorales, analizar y valorar propuestas, debatir, etcétera.

Partiendo del contexto de *DemoCritics*, se plantea el objetivo de extender su funcionalidad utilizando tecnologías inteligentes basadas en la web semántica, posibilitando un uso diferente al que aportan otras herramientas de participación ciudadana. Este objetivo se concreta en el desarrollo de un bot que realice tareas básicas de moderación de debates en un contexto electoral, y que sea capaz de extraer y utilizar la información de una ontología política para centrar los debates de los usuarios, extraer información de textos específicos o realizar otras tareas requeridas por los usuarios para enriquecer el debate.

Como resultado del proyecto se proponen una arquitectura y plataforma de desarrollo de bots moderadores de debates basados en ontologías, reutilizable en diferentes dominios de debate. Asimismo se desarrollan una ontología en el dominio de la política, y un bot moderador de debates políticos, *DemoCriticsBot*, basado en la arquitectura y plataforma antes mencionados.

DemoCriticsBot utiliza la API desarrollada por Telegram, la aplicación de mensajería instantánea para dispositivos móviles. Estará encargado de realizar las tareas principales de un moderador para desarrollar un debate político. Asimismo podrá ofrecer cierta información a poner en valor sobre temas relacionados con el debate seleccionado, dando la posibilidad a los usuarios de consultar textos, noticias o gráficos de actualidad pertinentes.

Este chat bot inteligente dará un enfoque, inviable hasta ahora, a las herramientas de participación ciudadana, gracias a la utilización de técnicas de inteligencia artificial y, en particular, tecnologías de la web semántica.

Palabras Clave: Política, Democracia, Participación Ciudadana, Bots, Ontologías, web semántica, Elecciones, Debates, Programas Electorales, Inteligencia Artificial, Extracción de Información, Procesamiento de lenguaje natural.

Abstract

Today we live in a changing society, with breakthroughs occurring at a faster rate to which they were accustomed societies from previous generations. These changes have led to profound disruptions happen in a political and social system that seemed to work so far. In recent decades, social movements emerged throughout the world and the fast pace evolution of information and communications technologies provided new tools that encourage democratic participation of citizens in politics.

This project starts from the main idea of a final degree project completed on 2015; “DemoCritics, Android real time collaborative editing political participation app”. This project’s purpose is to encourage citizenship to political participation using an application that makes possible to read electoral programs, analyze and evaluate different proposals, discuss, etc.

Based on “DemoCritics” context, we will attempt to extend its functionality using intelligent technologies based on semantic web, which will enable a different use to the ones that other citizenship participation tools provide. We will focus on creating a bot able to perform basic debating moderating tasks in an electoral context. The bot will be able to extract and use information from a political ontology in order to focus user discussions, extract information from specific texts and perform other tasks required by users to improve the discussion.

This project's result is an architecture and platform for developing discussion moderator bots based on ontologies, reusable in different contexts of discussion. Likewise an ontology in the domain of politics and a bot moderator of political debates, *DemoCriticsBot*, based on the aforementioned architecture and platform, are also developed .

DemoCriticsBot makes use of the API developed by Telegram, the instant messaging application for mobile devices. This bot will carry out moderator main tasks needed for a political debate to develop. Also, this bot may provide certain valuable information about topics related to the selected topic, improving discussion level and enabling users to consult texts, news and news charts related.

This smart chat bot will provide a different, not feasible until now, approach from the ones used in other citizenship participation tools so far, due to the use of artificial intelligence and particularly semantic web technologies.

Keywords: Politics, Democracy, Citizen Participation, Bots, Ontologies, Semantic Web, Elections, Proceedings, Electoral Programs, Artificial Intelligence, Information Extraction, Word Processing.

Capítulo 1 - Introducción

En este primer capítulo se exponen algunas observaciones acerca de las necesidades y motivaciones del proyecto. Posteriormente se detallarán los objetivos generales y la estructura general del documento.

1.1 Antecedentes

En los siguientes apartados discutiremos brevemente sobre el estado actual de la democracia, la política y los actuales sistemas de gobierno. Analizaremos la relación de la tecnología, las plataformas ciudadanas y la importancia de las herramientas como motor de cambio y la motivación base para el desarrollo del proyecto.

1.1.1 Relación de la Informática y la Inteligencia Artificial en la Ciencia Política

Aunque no sea evidente a primera vista, Informática y Ciencia Política son disciplinas que guardan una estrecha relación. Cuando realizamos el desarrollo de ciertas aplicaciones o entornos, u otras actividades directamente relacionadas con la informática que pueden tener consecuencias sociales y políticas, deberemos tener en cuenta la base de la sociedad donde nos encontramos. Cuando desarrollamos una aplicación que registra el sexo, la edad o la nacionalidad entre otros datos que podría recopilar cualquier aplicación acerca de un usuario, estamos haciendo política pues, en función de los valores sociales en el contexto en que la aplicación se implante, este hecho tendrá unas u otras consecuencias.

Del mismo modo es evidente la relación de la política con la inteligencia artificial, quizá a día de hoy no muy destacada pero previsible en un futuro cercano, ya que es fácil visualizar las consecuencias y efectos que podría tener la implantación de sistemas basados en inteligencia artificial para actividades de gobierno. Tales actividades podrían implicar desde la regulación de artefactos autónomos al desarrollo de sistemas automatizados que pudieran manejar datos sensibles de un estado. El desarrollo de sistemas inteligentes y autónomos pueden tener consecuencias positivas o negativas directamente en la sociedad. Es por ello que se precisa seguir una serie de normas éticas y asumir las responsabilidades a la hora de poner en práctica estos sistemas, con el fin de garantizar el progreso de una sociedad y garantizar los derechos humanos.

Cada vez son más las herramientas, aplicaciones y utilidades que ofrece la tecnología para llevar a cabo actividades que antes podrían ser impensables. Esto provocará muy pronto importantes cambios en nuestra vida social y en la forma de gobernar de los estados democráticos.

1.1.2 Motivación

En los últimos años la situación política en nuestro país ha sufrido constantes cambios. Después de una crisis económica, reformas educativas polémicas, casos de corrupción y el surgimiento de nuevos partidos en el panorama político, se han incentivado el interés y participación de los ciudadanos en la vida política. El surgimiento de colectivos y asociaciones, junto con el avance de las tecnologías de la información, han permitido desarrollar un conjunto de herramientas de participación ciudadana que fomentan adicionalmente el interés de participar, informarse y organizarse en tareas relacionadas con la actividad política.

Si bien la existencia de estas aplicaciones supone un gran avance en la sociedad para organizarnos y realizar tareas que antes no podríamos hacer desde cualquier lugar, aún no se han explotado las posibilidades de integrar estas herramientas dotándolas de inteligencia como en proyectos más ambiciosos en otros contextos.

Es por ello que encontramos interés en aplicar determinadas técnicas de inteligencia artificial para no sólo extender la funcionalidad de estas herramientas, sino para ir más allá y darles un enfoque diferente. Pues a pesar de que existen soluciones informáticas para determinadas tareas (analizaremos algunas en el estado del arte, ver sección 2.3), hasta la fecha ninguna ha sido dotada de inteligencia para enriquecer la funcionalidad que ofrece o conseguir autonomía.

Puesto que en el mencionado proyecto fin de grado se desarrolló una aplicación para la consulta y discusión de programas y propuestas electorales, resulta interesante dotarla de cierta inteligencia que facilite la interacción de los usuarios y su discusión sobre temas relacionados con la política, y aporte al mismo tiempo una dimensión educativa. En definitiva, el objetivo será dar soporte al debate político mediante un moderador-consultor-politólogo autónomo en forma de chat-bot.

Utilizar las tecnologías de la web semántica pueden resultar de interés para manejar la información contenida en bases de datos y documentación diversa, particularmente programas electorales, artículos o noticias, leyes, etcétera. El uso de ontologías para

clasificar la información junto con técnicas para la extracción de la información de bases de datos y textos en distintos formatos son los campos de investigación del ámbito de la Inteligencia Artificial de interés para nuestro propósito de dotar de inteligencia a un bot que actúe de forma autónoma para realizar ciertas labores de moderación y consultoría en el dominio del debate político.

1.1.3 Visión general del Proyecto

Para desarrollar el presente proyecto, se parte de la idea base de un Trabajo de Fin de Grado de la Universidad Complutense de Madrid, realizado el pasado año 2015. En este proyecto se desarrolla la aplicación *DemoCritics*, una App desarrollada en Android que facilita la lectura, valoración y discusión de programas electorales así como también el desarrollo de nuevas propuestas de gobierno con edición colaborativa en tiempo real. Podríamos establecer la idea base como acercar a la ciudadanía el panorama general político que podría resultar útil para informarse sobre los programas electorales, las propuestas de los candidatos y las valoraciones antes de acudir a votar.

De esta forma se intenta incentivar la participación ciudadana a través de los dispositivos móviles, con los que pasamos la mayor parte de nuestro tiempo a diario. Una forma alternativa a la tradicional en la que el usuario tendría que descargarse el programa electoral de cada partido o informarse en otros medios.

Partiendo de la idea principal de *DemoCritics*, en este proyecto queremos seguir incentivando la participación política de la ciudadanía facilitando su uso y comprensión. Por ello se plantea el desarrollo de un Bot que mediante técnicas de Inteligencia Artificial pueda moderar debates políticos sobre temas concretos, de forma que sea capaz de orientar a los usuarios sobre el tema del debate, facilitarles información de actualidad acerca de los temas que se estén debatiendo o simplemente poner a disposición del debate otra información de consulta que pueda resultar de interés.

La inteligencia del bot se basaría en las técnicas utilizadas para extraer y relacionar el conocimiento de la ontología política desarrollada para dar soporte a los usuarios. El bot desarrollado también sería capaz de buscar términos relacionados con las solicitudes de los usuarios, extrayendo información en textos especializados como programas electorales.

Con esta idea cualquier usuario sin la necesidad de contar con ciertos conocimientos sobre ciencia política, podría participar en debates de carácter político con la asistencia

de un Bot. Así por una parte ofrecemos una asistencia autónoma, a la vez que los usuarios aprenden sobre diversos ámbitos de la política o el estado del país.

La aplicación descrita implicaría por un lado el diseño de una Ontología en OWL para definir los principales términos políticos que podrá manejar el Bot y, por otro lado, el desarrollo del Bot que haga uso de la ontología desarrollada y una base de conocimientos basada en el proyecto *DemoCritics*.

1.2 Contexto

Podemos identificar el contexto del proyecto en tres grandes áreas, hablaremos más detalladamente de cada una de ellas en el Estado del Arte (capítulo 2). En primer lugar haremos referencia a las herramientas destinadas a la participación política ciudadana. Como ya hemos mencionado antes, estas herramientas se limitan a la conexión de un determinado grupo de usuarios para realizar una actividad específica, ya sea una votación, un debate, el desarrollo de un documento o cualquier otra tarea colaborativa. Sin embargo estas herramientas carecen de inteligencia o componentes de automatización que den soporte a los usuarios o sean capaces de extraer información de textos o sitios especializados, sin verse obligado a realizar tareas manuales de inserción de información o contenido.

Por otra parte, las ontologías desarrolladas hasta la fecha (ver sección 2.1), se centran demasiado en el gobierno electrónico y a dar apoyo a los servicios y organismos del estado. Estas ontologías parten de la base de un sistema de gobierno, sin dar cabida al conjunto de términos relacionados con la ciencia política que puedan estar o no registrado en un gobierno formal. Es por ello que se precisa una ontología más abierta al ciudadano, orientada a un conjunto de términos donde agrupar los factores que intervienen en un estado.

Por último, los bots desarrollados hasta la fecha, se han orientado a la “humanización” y a mantener conversaciones fluidas con los usuarios. Sin embargo no se ha procurado a la recuperación de información compleja y adecuada al contexto para dar soporte a los debates especializados en los chats con usuarios.

El desarrollo de un bot moderador de debates políticos tal como se ha descrito en la sección anterior traería consigo aportaciones en el campo de investigación de la recuperación de información basada en ontologías aplicada al desarrollo de bots intelligen-

tes y, específicamente, al desarrollo de bots moderadores de debates en dominios específicos con carácter educativo.

1.3 Definición del problema e Hipótesis del trabajo

En este trabajo se planteará la arquitectura e implementación para facilitar un bot-moderador que sea capaz de extraer información de textos y relacionarlos con una ontología política. Además este bot será capaz de interactuar con los usuarios realizando las tareas básicas para poner en marcha un debate de contenido político.

Podemos definir nuestra hipótesis de trabajo en los siguientes puntos:

- Es posible desarrollar una ontología política que acoja todos los términos y relaciones que podrían aparecer en el debate de una campaña electoral ante unas posibles elecciones.
- Las aplicaciones actuales relacionadas con la participación política no explotan las oportunidades de la técnica en la actualidad.
- Es posible proponer una arquitectura con ciertos componentes y herramientas software que permitan a un bot-moderador obtener y relacionar información de textos en base a una ontología política.
- Es posible desarrollar un espacio en el que debatir sobre diferentes temas de índole política que sea asistido y moderador por un bot-moderador.

1.4 Objetivos del Proyecto

Podemos resumir los objetivos del proyecto en tres puntos:

- i. Concebir una arquitectura y proporcionar unos componentes software basados en técnicas de inteligencia artificial para el desarrollo de bots-moderadores-consultores de debates, basados en ontologías del dominio objeto de debate y en la extracción de información en textos y bases de datos con conocimiento experto del dominio.
- ii. Desarrollar una ontología del dominio de la política.
- iii. Ilustrar el uso de la arquitectura y componentes software propuestos con la implementación de un bot moderador-consultor de debates en un contexto político, que haga uso de la ontología desarrollada.

En lo que respecta al último objetivo, se ha desarrollado un Bot en Telegram, mediante el cual cualquier usuario puede crear debates de carácter político con los integrantes

que quiera invitar. Este Bot hará uso de la API pública de Telegram y por tanto podrá ser accesible de forma pública desde cualquier usuario de la aplicación de mensajería instantánea.

Las tareas principales del Bot serán utilizar la ontología desarrollada y posteriormente extraer información de textos basada en la base de conocimientos del proyecto *DemoCritics*, utilizando técnicas de inteligencia artificial, de tal forma que el Bot preste soporte para moderar y aportar información de utilidad para enriquecer el debate. Por último se estudiarán otras funcionalidades relacionadas con el seguimiento de los usuarios: estudiar el progreso en los debates, las intervenciones o estadísticas de uso. Por tanto las principales fases del proyecto serán:

- 1ª Parte: Diseño de una Ontología de términos políticos en OWL.
 - Desarrollo de una ontología que acoja los principales términos políticos relacionados con un estado democrático utilizando el entorno *protégé*.
- 2ª parte: Desarrollo de un Bot en Telegram.
 - Desarrollo de un Bot en Telegram utilizando la API pública y librerías en Python que permitan la interacción con los usuarios de la aplicación de mensajería.
 - Implementación de técnicas de reconocimiento de textos y uso de la ontología y base de conocimientos para ofrecer asistencia a los usuarios.
- 3ª parte: Integración de @DemoCriticsBot con el proyecto *DemoCritics*
 - Trataremos de reutilizar la base de conocimientos del proyecto *DemoCritics* en base a lo relacionado con los programas electorales.
 - Se ampliará la base de datos para añadir algunas funcionalidades que necesite el bot.

1.5 Metodología de investigación

Los objetivos citados en la sección anterior conlleva al estudio de las investigaciones publicadas sobre ontologías para ciencia política, y las investigaciones del desarrollo de bots basados en ontologías para la extracción de información.

Para alcanzar los objetivos propuestos se exponen a continuación los siguientes objetivos o metas generales que marcarán la ruta del proyecto:

- I. Realizar un estudio del estado de las ontologías publicadas hasta la fecha en relación a la ciencia política.
 - Analizar y comprender la necesidad del desarrollo de una ontología para definir los términos y relaciones en la ciencia política.
 - Comprender la perspectiva teórica-filosófica de aquellos trabajos que requieren el estudio de una ontología para un contexto político.
 - Estudiar las soluciones propuestas hasta la fecha relacionadas con las ontologías para el gobierno electrónico.
- II. Definir una ontología que agrupe términos y relaciones en relación con la ciencia política.
 - Analizar las diferentes perspectivas planteadas en el estado del arte y adquirida en la visión de entrevistas con expertos.
 - Identificar los términos, instancias y relaciones para el desarrollo de una ontología política.
 - Implementar la ontología basada en las tecnologías de la web semántica.
- III. Realizar un estudio del estado de los bots conversacionales basados en ontologías.
 - Analizar las tecnologías y plataformas que forman los chat-bots, explorando sus limitaciones.
- IV. Definir una arquitectura de propósito general que integre un chat-bot basado en la extracción de información mediante la ontología desarrollada.
 - Identificar los principales componentes que formen parte de la arquitectura, garantizando el uso en otros dominios.
 - Implementar técnicas de extracción de la información basada en ontologías, haciendo uso de diccionarios y otras herramientas.
 - Integrar la base de datos del proyecto *DemoCritics* para enriquecer el sistema de conocimientos del bot desarrollado.

Para obtener más detalles de la metodología y pasos seguidos en el proyecto, se recomienda leer la sección 3 del presente documento, donde se detallan con más énfasis los objetivos anteriormente citados a modo resumen.

1.6 Estructura del documento

La idea fundamental de la presente memoria es dar a conocer los aspectos más destacados del proceso de desarrollo, las intenciones del proyecto y la investigación que se ha llevado a cabo. Se analizarán otras aplicaciones de carácter similar desarrolladas hasta la fecha para valorar las aportaciones del proyecto.

A lo largo del documento el lector se encontrará con una estructura representada en capítulos en la que se detallarán las diferentes etapas del desarrollo del proyecto:

- **Capítulo 2 – Estado del Arte:** se realiza un pequeño análisis sobre las principales investigaciones o desarrollos relacionados con el proyecto en cuestión, enfocando por una parte las aplicaciones relacionadas con la política y participación ciudadana y por otra el procesamiento de textos basado en técnicas de Inteligencia Artificial.
- **Capítulo 3 – Metodologías del Proyecto:** se exponen las metodologías llevadas a cabo para el desarrollo del proyecto.
- **Capítulo 4 – Tecnologías del Proyecto:** se dan a conocer las tecnologías, herramientas y plataformas que se han utilizado en el proyecto.
- **Capítulo 5 – Arquitectura del Proyecto:** se da una idea general de implementación y funcionamiento de la conexión entre las plataformas que facilitan la ejecución del Bot.
- **Capítulo 6 – Desarrollo de una ontología política:** se detallan las motivaciones, el proceso de investigación y el desarrollo de una ontología de términos políticos.
- **Capítulo 7 – Desarrollo de @DemoCriticsBot:** proceso de implementación del bot @DemoCriticsBot en la plataforma Telegram.
- **Capítulo 8 – Resultados y trabajo futuro:** se discuten y analizan los resultados obtenidos tras el desarrollo del proyecto. Se pone en valor los objetivos alcanzados y se discute el trabajo próximo a realizar para mejorar y continuar con el proyecto.
- **Capítulo 9 – Conclusiones:** se extraen las conclusiones acerca del resultado final del Trabajo de Fin de Máster.

Capítulo 2 - Estado del Arte

En este capítulo se explorarán las diferentes investigaciones y soluciones Software que podemos encontrar en la actualidad. Se analizará de forma resumida cada una de ellas para dar una visión general de las características, la funcionalidad y las aportaciones realizadas hasta la fecha.

Este capítulo quedará dividido en cuatro apartados que corresponderán al tipo de aplicación, plataforma o desarrollo relacionado con las publicadas hasta la fecha. En primer lugar se analizarán las Ontologías (ver sección 2.1), posteriormente expondremos algunas aplicaciones o plataformas que traten la extracción de la información sobre textos (ver sección 2.2). Por último, analizaremos algunas de las aplicaciones que han surgido últimamente que fomentan la participación política (ver sección 2.3), y finalmente haremos un breve repaso a los bots basados en ontologías (ver sección 2.4).

Al final del capítulo (ver sección 2.5) se pondrán en valor las aportaciones del presente proyecto en comparación con los proyectos analizados en este capítulo.

2.1 Ontologías

En esta sección trataremos de cubrir aquellas ontologías publicadas relacionadas con la ciencia política o la gobernabilidad del estado, que pueden servirnos de utilidad para el Gobierno Electrónico (e-Government). Analizaremos las más relevantes hasta la fecha y pondremos en valor el uso o utilidad que se le ha pretendido dar, si bien no han sido utilizadas en aplicaciones externas similares a la arquitectura planteada en este proyecto. En particular, en el caso de los primeros trabajos referenciados (secciones 2.1.1 a 2.1.4) se trata de estudios puramente teóricos en el campo de las ciencias políticas. El resto de los trabajos tienen algún tipo de aplicación relacionado con tareas de gestión y control de la información.

2.1.1 *Political Ontology – Colin Hay*

El profesor de ciencia política Colin Hay de la Universidad de Sheffield, publicó en 2006 en el libro *The Oxford Handbook of Contextual Political Analysis* [47], un capítulo que hacía la referencia al desarrollo de una ontología para el análisis político. Se discute la medida en la que la ciencia política se hizo más inestable y cómo generaba

cada vez mayor desconfianza. Realiza un análisis sobre el término de ontología política y las herramientas de análisis político.

2.1.2 *Rawls's political ontology*

La tesis publicada por Philip Pettit de la Universidad de Princeton [48], trata sobre una ontología implícita de las personas y la relación que se genera entre el pueblo y el Estado, de forma a la manera en la que pensamos en términos normativos acerca de la política.

2.1.3 *The political ontology of European integration*

En este artículo [49], se discute la ontología política de la Unión Europea a través de un análisis de dos posiciones. Se argumenta que el exclusivo marco ontológico y los dualismos que reproduce (objetivo-subjetivo, individuo-institución, socialización de cálculo, el interés en normas, supranacional, nacional, etc.) impide que una gran cantidad de investigación de la elaboración de una compleja descripción empírica de integración de la Unión Europea. La alternativa busca ir más allá de estos dualismos de desarrollar un enfoque relacional que tenga en cuenta el carácter interactivo de las instituciones, los individuos y grupos que habitan en ellos.

2.1.4 *Constitution as Executive Order The Administrative State and the Political Ontology of "We the People"*

El artículo de la revista *Administration & Society* [50], contribuye a los debates sobre el giro ontológico y sus implicaciones para la democracia mediante la propuesta de una comprensión experimental de una ontología política. Se analiza por qué el cambio de la epistemología a la ontología en los estudios de ciencia y tecnología han sido definitivos para el estudio de la política y de la democracia. Se distinguen hasta tres diferentes conceptos de la ontología política: teóricos, empíricos y experimentales.

2.1.5 *Ontology-enabled E-Government Service Configuration - the OntoGov Approach*

En el año 2004 el Instituto Tecnológico de Karlsruhe, desarrolló el proyecto OntoGov [1], mediante el cual agilizaría la reconfiguración y servicios de la administración pública. OntoGov propuso la definición de una ontología genérica de alto nivel para el ciclo de vida de servicio de Gobierno Electrónico (e-Gov).

Al mismo tiempo, se propuso una ontología destinada a los servicios públicos del gobierno electrónico donde se introdujo una aproximación del modelado de la ontología. La plataforma está desarrollada en Java cuyo código está disponible bajo una licencia GPL en el repositorio *SourceForge*. Se plantea como una solución administrativa para agilizar el funcionamiento de la administración pública de los gobiernos.

2.1.6 *Ontologies for e-Government*

Las “Ontologías para el eGovernment” [2] es un proyecto desarrollado por TopQuadrant y Ralph Hodgson, fue uno de los primeros trabajos que propuso la creación de una ontología para el gobierno electrónico. El proyecto modeló desde un primer momento las agencias de Estados Unidos y su estructura. El resultado de la obra se publicó en el sitio web www.oegov.org donde se pueden consultar las diferentes ontologías en OWL publicadas con una licencia *Creative Commons Attribution-Share Alike 3.0*.

Sin embargo la intención de este proyecto no fue más allá de modelar las diferentes agencias gubernamentales de los Estados Unidos, sin aprovechar su utilidad o llevarla más allá en un caso práctico, el proyecto no ha sido actualizado desde 2010.

2.1.7 *Semantic Web Technologies for Information Management within e-Government Services*

Este artículo propone el caso de desarrollar una ontología dirigida al gobierno electrónico basada en las tecnologías de la web semántica [3], con la intención de cómo relacionar las áreas de gestión de la información y el conocimiento en estos sistemas. Se analizan las diferentes tecnologías de la web semántica para representar y modelar la información basada en ontologías. También se propone algunas propuestas de arquitectura para implementar una ontología en un servicio de gobierno electrónico.

El artículo finaliza proponiendo una serie de objetivos o desafíos para la cooperación entre los desarrolladores de sistemas y los gestores de la información, para así fomentar el uso de las tecnologías de la web semántica en posibles enfoques para el e-Government.

2.1.8 *Integration of Government Services using Semantic Technologies*

En 2011 varios autores de la Universidad Técnica de Košice, desarrollaron el proyecto Access-eGov [4]. El objetivo del proyecto era mejorar la accesibilidad y la conectivi-

dad de los servicios gubernamentales para ciudadanos y empresas por medio de la creación de escenarios integrados y proporcionar orientación a los usuarios. El sistema ayuda al usuario para identificar y facilitar el acceso a trámites y consultas relacionadas con el sistema de gobierno electrónico. El proyecto Access-eGov fue desarrollado con herramientas software que permiten la integración de servicios utilizando tecnologías semánticas. Además se desarrolló una metodología para el desarrollo y diseño de una ontología. Fue desarrollado con ontologías específicas y evaluado en escenarios reales. El código del proyecto fue publicado bajo una licencia de software libre en su página oficial (<http://www.access-egov.org/>), aunque actualmente no está disponible por abandono del proyecto.

2.1.9 Un Modelo Ontológico para el Gobierno Electrónico

Dos autores de la Universidad Nacional de Misiones [10], plantean el problema de la toma de decisiones depende de cierta información que debe suministrarse con un formato de datos enriquecido. Hacen referencia a las grandes cantidades de información con las que cuentan las agencias gubernamentales y la falta de organización o clasificación de los datos puedan dificultar la difusión y uso de los mismos.

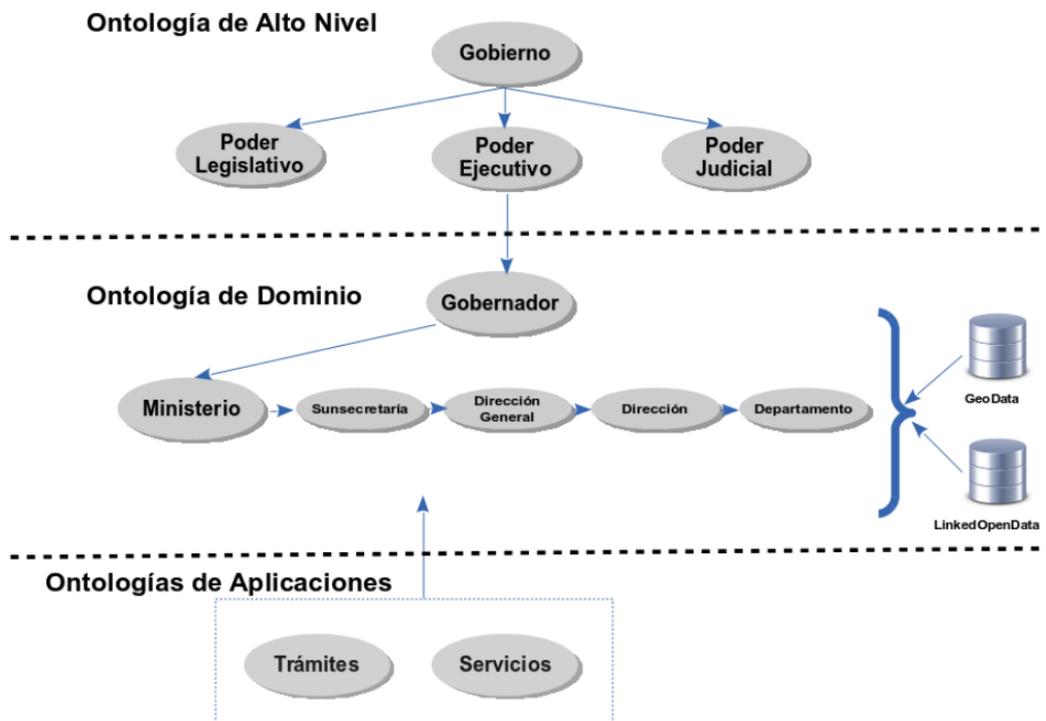


Figura 1: Modelo de la Ontología de Gobierno Electrónico [10].

Por ello proponen un modelo para la representación conceptual de las unidades de organización del Estado, vistos como entidades georeferenciadas del Gobierno Electrónico, basado en ontologías diseñadas bajo los principios de los Datos Abiertos Vinculados. La aplicación del modelo propuesto, es una ontología que forma parte del Portal de Datos Abiertos del gobierno de la Provincia de Misiones [11]. Para el desarrollo de la OGE (Ontología de Gobierno Electrónico) se utilizó la metodología Methontology [38]. La figura 1 muestra las clases principales del desarrollo de la Ontología para el Gobierno Electrónico en alto nivel.

Como citamos anteriormente la ontología es utilizada actualmente en el portal de datos abiertos en que pone a la disposición de los ciudadanos el Gobierno de la Provincia de Misiones (Argentina), para aumentar el acceso público a conjuntos de datos de valor y que sean legibles por máquinas.

2.2 Extracción de información de textos basada en ontologías

En este sub-capítulo trataremos las investigaciones de mayor relevancia y que guarden mayor con nuestro proyecto, sobre las técnicas de extracción de información de textos para su posterior procesamiento.

Existen numerosas investigaciones sobre la extracción de la información basada en textos. Algunos como el sistema *MULTIMEDICA* [5], que extrae información de textos en el dominio biométrico. También existen algunos proyectos que tratan de obtener información de los mensajes publicados en las redes sociales como Twitter [6]. Otros sistemas tratan de extraer la información de libros especializados de texto, como la reciente tesis publicada que se basa en la extracción automática de información semántica organizada en estructuras sintácticas [6].

Como el campo de la extracción de la información basada en textos complejos es bastante amplio, a continuación nos centraremos en aquellos trabajos basados en ontologías para extraer información de textos.

2.2.1 Metodología para la población automática de ontologías: aplicación en los dominios de medicina y turismo.

Esta tesis doctoral en Documentación, con mención de doctorado europeo, realizada por una alumna de la Universidad de Murcia [9], plantea el problema de la instancia- ción automática de ontologías a partir de texto en lenguaje natural. Destaca la cons-

trucción automática de ontologías que plantea la creación semi-automática de conceptos, relaciones, instancias y atributos ontológicos. Pone en valor el echo de tener que actualizar o modificar determinadas ontologías mediante la creación de nuevas instancias o la modificación de ciertas relaciones que pueden sufrir la ontología con el paso del tiempo.

Utilizando los dominios de medicina y turismo para la instanciación de una ontología, divide el proceso en cuatro fases. Comenzando por la fase de procesamiento del lenguaje natural, reconocimiento e identificación de las entidades nombrada, población de la ontología y verificación de la consistencia de la ontología resultante del proceso.

2.2.2 Los proyectos SINAMED e ISIS: Mejoras en el Acceso a la Información Biomédica mediante la integración de Generación de Resúmenes, Categorización Automática de Textos y Ontologías.

Los proyectos SINAMED e ISIS [7] son dos sistemas que tratan de categorizar el texto, extraer resúmenes utilizando ontologías para representar objetos y relaciones sobre un dominio biomédico específico: los historiales clínicos de pacientes y la información científica biomédica asociada. Utilizan lenguajes como UMLS (*Unified Medical Language System*) para la categorización del texto o la generación de resúmenes.

Años más tarde la Universidad de Jaén publicó un artículo para la Categorización de textos biomédicos usando UMLS [8]. El artículo presente un sistema automático para la categorización de texto multi-etiqueta que hace uso de UMLS, al igual que el trabajo anterior. Este sistema fue enriquecido utilizando las ontologías que incluye UMLS, cuyos resultados destacaron por encima de una categorización tradicional.

2.2.3 Applying Ontologies for Web Text Mining Using Heterogeneous Information.

Un estudio reciente publicado en 2014 trata sobre la extracción de la información en grandes documentos de textos en los que aplican algoritmos tradicionales basados en *text mining* [51]. Durante la extracción de esta información en múltiples formatos, se establece una serie de reglas que identifican las relaciones entre ellos. El objetivo final será construir una ontología que clasifique toda la información obtenida en base a unas reglas y filtros que permitan definir los objetos y relaciones de la ontología, utilizando OWL para la definición de la misma. La figura 2 muestra un esquema conceptual del proceso que se lleva al extraer la información y generar una ontología.

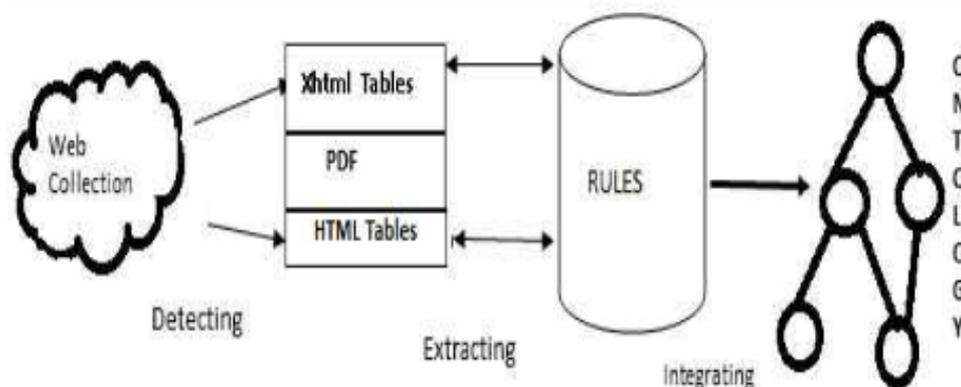


Figura 2: Proceso para la conversión de la información en una ontología [51].

Es interesante cómo generan la ontología a través de la información adquirida, pues esto nos puede permitir desarrollar una serie de ontologías en múltiples dominios sin ser expertos del tema. Aunque la automatización a la hora de definir la ontología puede perjudicar en la calidad o el conocimiento que tengamos sobre el dominio que estemos trabajando.

2.2.4 *Ontology learning from text: A look back and into the future.*

En el artículo publicado por Universidad de Australia Occidental [52], trata de dar un repaso a los avances de la web semántica relacionando las necesidades que poseen los actuales sistemas basados en información de texto. El crecimiento de estos sistemas demanda el uso de las tecnologías de la web semántica basadas en ontologías para representar la información. Exponen cómo algunas áreas como el procesamiento del lenguaje natural han contribuido a la investigación del aprendizaje de las ontologías basadas en textos en la última década.

El artículo nos mostrarán los avances hasta la fecha sobre el aprendizaje de las ontologías, la representación del conocimiento y los desafíos pendientes que definirán las próximas líneas de investigación de este área en el futuro. Un artículo bastante extenso e interesante para hacerse una idea global de los avances y logros de este área de investigación.

2.2.5 *Evaluating techniques for learning non-taxonomic relationships of ontologies from text.*

La investigación llevada a cabo por la *Federal University of Maranhão*, trata de evaluar las técnicas de aprendizaje y extracción automática de relaciones no-taxonómicas

de una ontología sobre un texto [53]. Estas técnicas se basan en el uso del procesamiento del lenguaje natural y otras técnicas de aprendizaje automático. Proponen tres métodos para evaluar las ontologías adquiridas del resultado del aprendizaje: evaluándolas en una aplicación ejecutable, contando con expertos en el dominio o comparándola con una ontología de referencia definida anteriormente. El artículo muestra dos procedimientos para evaluar las técnicas de aprendizaje de las relaciones no-taxonómicas basados en la comparación de las relaciones una ontología de referencia. Estos procedimientos los utilizarán para extraer relaciones en los dominios relacionados con la biología y derecho familiar.

2.3 Participación Política y Ciudadana

En esta sección exploraremos las soluciones que han surgido en los últimos años relacionadas con el objetivo principal del proyecto; el fomento y difusión de la actualidad política basada en las herramientas digitales como medio de información e interés ciudadano.

2.3.1 *DemoCritics*

DemoCritics [12] es un Proyecto de Fin de Grado desarrollado por dos alumnos de ingeniería informática de la Universidad Complutense de Madrid. Este proyecto en forma de App para Android, pone en valor dos objetivos principales. Por un lado, facilitar el acceso a la ciudadanía de los programas electorales en un formato más visible y adaptado para plataformas móviles. Por otro, la discusión y elaboración de propuestas, leyes y programas fomentando la participación y colaboración. De esta forma agrupamos a dos tipos de usuarios dentro de una misma aplicación. Por un lado aquel que desconoce los programas electorales y quiere informarse sobre aquellas secciones que le puedan interesar por su profesión o intereses. Mientras que por otro tenemos al usuario más activo que le interesa participar desarrollando nuevas propuestas, realizando opiniones o debatiendo. En la figura 3 podemos ver algunas capturas de pantalla de la aplicación en relación con los programas electorales, la categorización de los mismos y las propuestas ciudadanas.

Hasta este punto se consigue poner en valor una nueva forma de leer los programas, clasificado por categorías y adaptado a las nuevas tecnologías. Pues hasta ahora sólo disponíamos de un largo PDF por el que no sabríamos por dónde empezar. Y por tanto

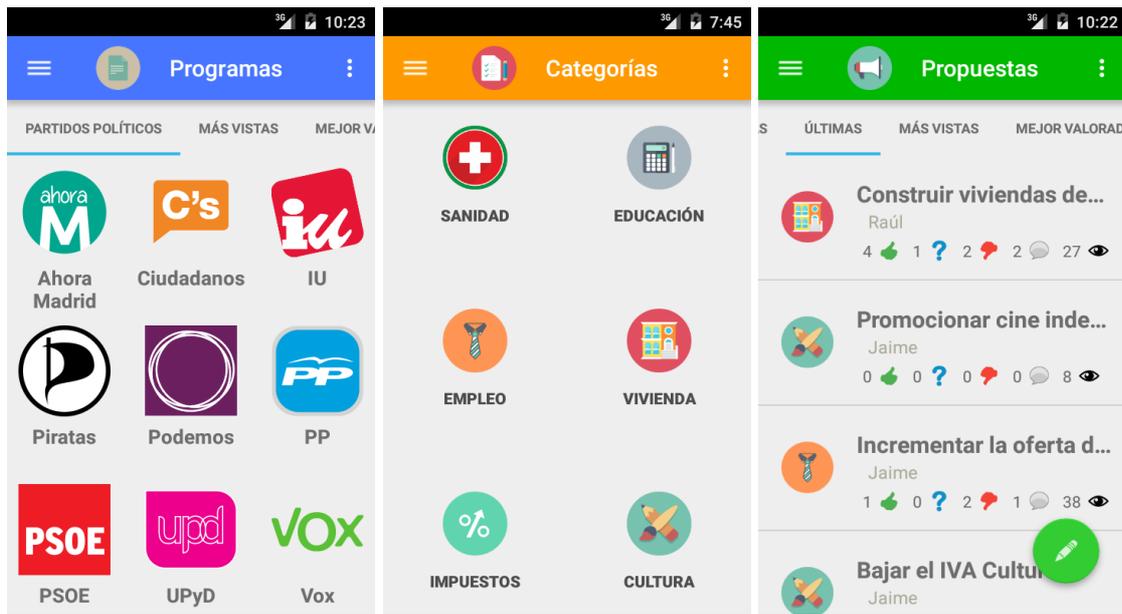


Figura 3: Algunas capturas de DemoCritics[12].

fomentar la participación ciudadana en la elaboración de propuestas colaborativas que podrán ser votadas, debatidas y valoradas.

2.3.2 Poletika.org

Esta aplicación [13] pone en valor la posibilidad de acceder a toda la información relacionada con una campaña electoral o los problemas y soluciones relacionadas en el dominio político, desde cualquier dispositivo móvil. Fomentando la participación en política a los usuarios menos familiarizados y ofreciendo una alternativa que facilita el acceso a la información.

Poletika cuyo significado es política + ética, surge en las pasadas elecciones a las cortes generales del 20 de Diciembre de 2015. Esta herramienta presentada en una página web, pretende ofrecer de forma más clara y precisa las medidas, declaraciones y programas electorales de los principales candidatos para las elecciones generales.

Poletika nos facilita resúmenes por temas, ideas y comparativas sobre lo que tratan los programas electorales de los candidatos e incluso las declaraciones que hacen sobre algunos temas. Así cualquier persona podría saber las políticas que políticas en una determinada materia en comparación con todos los partidos. También tienen pendiente un sistema de cumplimiento que vigila los compromisos y el programa del gobierno

durante una legislatura. La figura 4 muestra un análisis de los programas de los partidos en las diferentes materias que están clasificados.

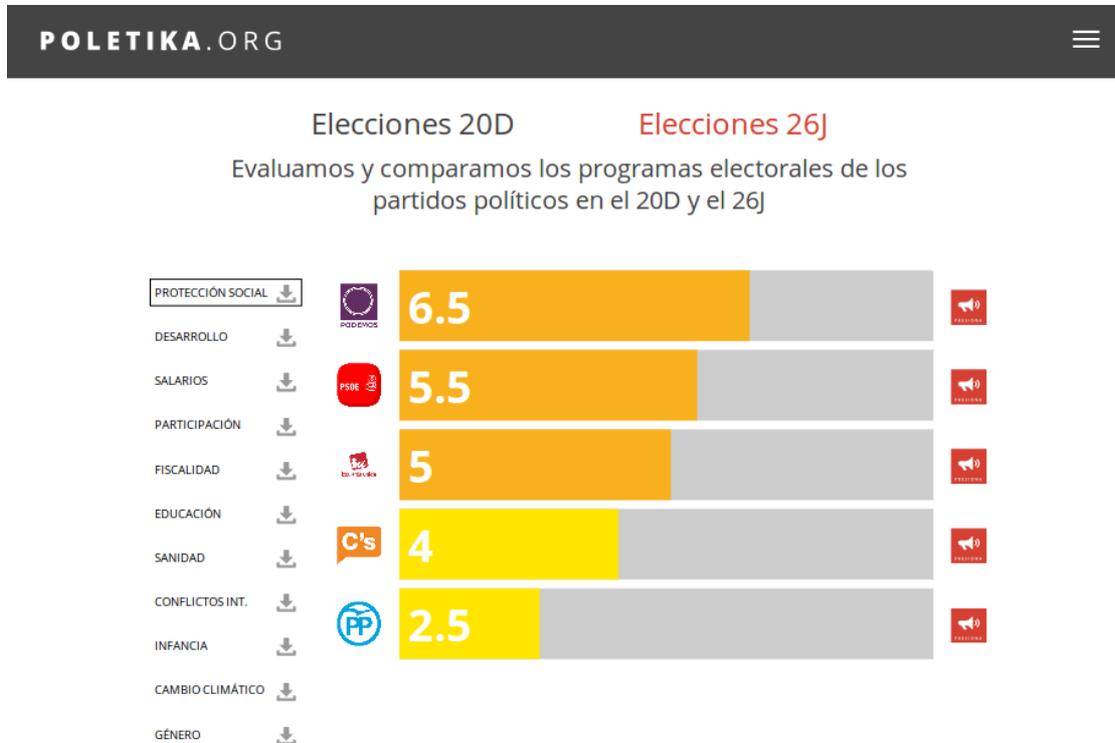


Figura 4: Poletika [13], Comparativa de los programas electorales.

Este sistema cumple con el objetivo de difundir la información a la ciudadanía en un formato más visible y objetivo frente al programa electoral, así como presionar a los políticos para que expresen su opinión en aquellos temas donde guardan silencio o no se expresan adecuadamente. Sin embargo Poletika cuenta con un problema sobre la generación de todo el contenido, resúmenes, puntuaciones, comparativas... Y es que todos ellos son generados por un equipo detrás. Pues aunque pueda parecer imparcial en un primer momento, siempre podremos caer en algunas preferencias o ideas ideológicas que nos hagan favorecer o penalizar a ciertos partidos. Sin embargo cumple con la función de difundir el conocimiento y fomentar el interés de la ciudadanía en materia política.

2.3.3 Appgree

Appgree [14] es una plataforma que cuenta con más de 2 millones de usuarios, desarrollada con el objetivo de discutir una serie de ideas y dialogar con los usuarios independiente de donde se encuentren a través de esta plataforma. En la aplicación puede-

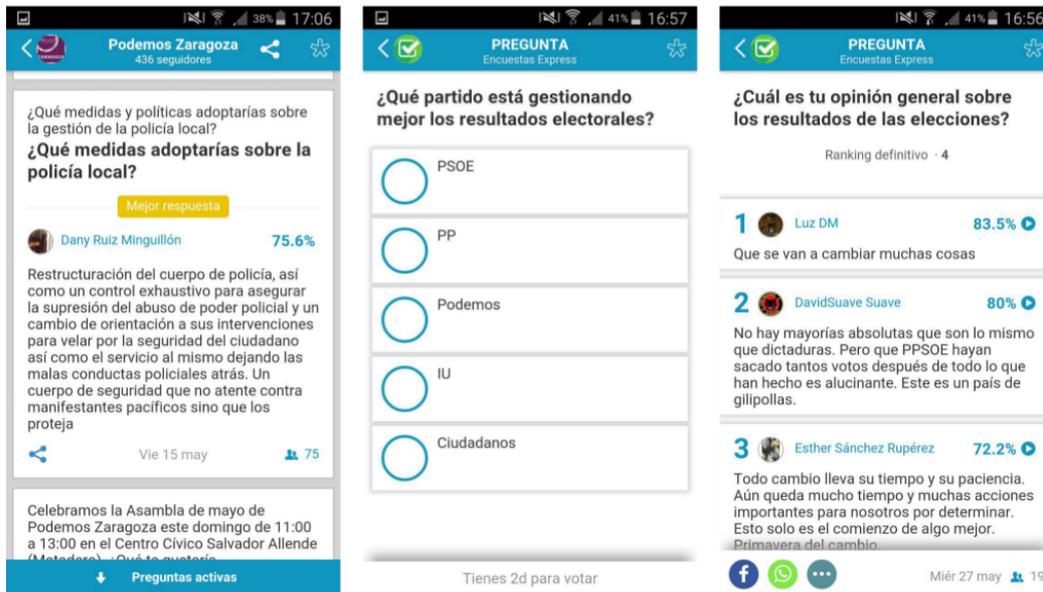


Figura 5: Algunas capturas de la aplicación Appgree [14].

mos acceder a una lista de canales: que aglutinan encuestas, propuestas y preguntas, pudiendo votar y ver los resultados actuales de votación y respuestas. La figura 5 muestra algunas de las interacciones más comunes con la aplicación. Para fomentar la participación potencian mucho el uso de listas de preguntas más candentes y recientes. Además se pueden compartir preguntas por redes sociales para aumentar su difusión. Actualmente es utilizada por colectivos, asociaciones y partidos políticos con la idea de poder llegar a todos los usuarios sin la necesidad de estar a la misma hora y lugar [17].

2.3.4 Agora Voting

Agora Voting es un sistema de votaciones online, que permite realizar procesos de votaciones manteniendo el anonimato de los votos y garantizando la privacidad mediante la encriptación de los mismos. Ha sido utilizado en procesos de primarias para elección de candidatos en algunas plataformas ciudadanas como *Barcelona En Comú* o el partido político *Podemos* [16].

Este sistema revoluciona la forma tradicional de votar en urnas como llevamos haciendo durante décadas. Pone en valor la posibilidad de ejercer el derecho al voto desde cualquier lugar, dependiendo tan solo de una conexión a Internet. Evitando así tener que ir a votar en un día y hora señalados o no teniendo solicitar otros medios de voto

mediante correo que puede ocasionar retrasos u otros problemas con el envío y los plazos.

Aunque si bien es cierto, el sistema respeta la privacidad y el anonimato del voto, ni si quiera el administrador del sistema puede ver la respuesta de cada voto. Puede poner el peligro la identidad o veracidad del usuario si este procedimiento pudiera caer en manos de otra persona. Por ello se debería implementar un sistema de doble autenticación, ya fuera mediante un dispositivo móvil personal, un certificado o incluso mediante presencia virtual vía webcam. Para así asegurarnos de que realmente la persona ha ejercido su derecho al voto libremente.

Un artículo de la Universidad de Alicante [15] analiza las potencialidades y limitaciones de estas aplicaciones y plataformas para la deliberación y la toma de decisiones ciudadanas. Referenciando a *Appgree*, *Loomio*, *Agoravoting*, *Sarepolis* y *Civitania*.

2.4 Bots basados en ontologías

En esta última sección del Estado del Arte, analizaremos los Bots conversacionales basados en ontologías hasta la fecha. También citaremos los chat-bots, aquellos bots basados en chats de texto. Posteriormente se citarán aquellos aspectos destacados en el presente proyecto que no están cubiertos en los trabajos citados.

2.4.1 *OntBot: Ontology based chatbot*

En 2011 la revista de Innovation in Information & Communication Technology (ISIICT) [19], publicó un artículo en el que desarrollan una ontología con el propósito de operar con chat-bots (OntBot). OntBot utiliza algunas técnicas de mapeo para transformar la información de las ontologías y el conocimiento adquirido en una base de datos relacional que utilizará como conocimiento para operar en los chats con usuarios. En este enfoque se plantean una serie de inconvenientes en el manejo de los chat-bots habituales, como la necesidad de aprender y utilizar el lenguaje AIML (*Artificial Intelligence Mark-up Language*), un lenguaje de programación basado en XML diseñado para utilizarlo en el chat-bot A.L.I.C.E (*Artificial Linguistic Internet Computer Entity*). OntBot es capaz de realizar interacciones sencillas con los usuarios utilizando su lenguaje natural y el apoyo de los diferentes dominios de la aplicación.

2.4.2 Chatbots as Interface to Ontologies

El libro *Advances onto the Internet of Things* publicado en 2014 [21], contiene un capítulo que referencia a los chat-bots como interfaz para las ontologías. Critican el enfoque de cómo se han desarrollado los agentes conversacionales hasta la fecha, utilizando unos patrones de reglas por defecto que llevan a diálogos de baja credibilidad, tomando unas reglas basado en un conocimiento restrictivo y limitado.

Describen las diferentes arquitecturas hasta la fecha de chat-bots, explotando el uso de las ontologías con el objetivo de obtener una información más clara y superar las principales limitaciones de los chat-bots: el conocimiento basado en la construcción de un mecanismo de diálogo.

2.4.3 MooBot & Nightbot – Twitch Chat Bot

MooBot [41] es un chat-bot disponible para la plataforma Twitch [43], una plataforma que ofrece un servicio de streaming en directo principalmente orientado a los videojuegos. MooBot es capaz de realizar tareas de moderación en salas de Twitch que han sido creadas para comentar un streaming. También puede prevenir los mensajes spam, envío de hiperenlaces y reglas personalizadas. Permite definir comandos personalizados (muy similares a los Bots de Telegram [44]), crear encuestas y enviar notificaciones sobre la actividad de los usuarios en el chat.

Nightbot [42] es otro chat bot que podemos integrar en Twitch para moderar nuestros chats. En comparación con MooBot, Nightbot dispone de una amplia gama de configuraciones para tareas de moderación relacionadas con el spam. Nos permitirá añadir reglas basadas en bloquear contenido obsceno, hiperenlaces, mayúsculas, emoticonos, etcétera. Al igual que MooBot, cuenta con comandos personalizados para solicitar acciones al bot, puede crear encuestas a través de comandos, y además posee un canal de estadísticas de moderación que podemos programar para que nos avise con una frecuencia determinada.

Estos chat-bots presentados para la plataforma Twitch son bastante interesantes, pero si bien es cierto que requieren de la configuración, personalización y mantenimiento del usuario. Pues aunque cuentan con herramientas que permiten configurar ciertas acciones de nuestro bot, estos bots no cuentan con técnicas de inteligencia que les permi-

tan funcionar desde el primer momento, determinar tareas de moderación según su juicio o realizar tareas de aprendizaje basadas en los chats realizados en el pasado.

2.4.4 PolitiBot

Politibot [18] fue desarrollado por un grupo de desarrolladores, sociólogos y periodistas para ponerlo a prueba en la repetición de las elecciones generales en España en 2016. Se trata de un bot desarrollado en la plataforma de mensajería instantánea Telegram, que nos ofrece estar al día de las últimas novedades relacionadas con la actualidad política, sondeos, artículos, gráficos, etcétera.

Para iniciar la interacción o solicitar una información concreta, Politibot nos ofrece la posibilidad de realizar una serie de acciones a través de unos comandos u opciones desde la interfaz en nuestro chat personal. Además de enviarnos notificaciones, preguntas, encuestas u otra información que pueda servirnos de utilidad. En la figura 6 podemos ver algunas de las interacciones más frecuentes con @politibot en un chat privado.

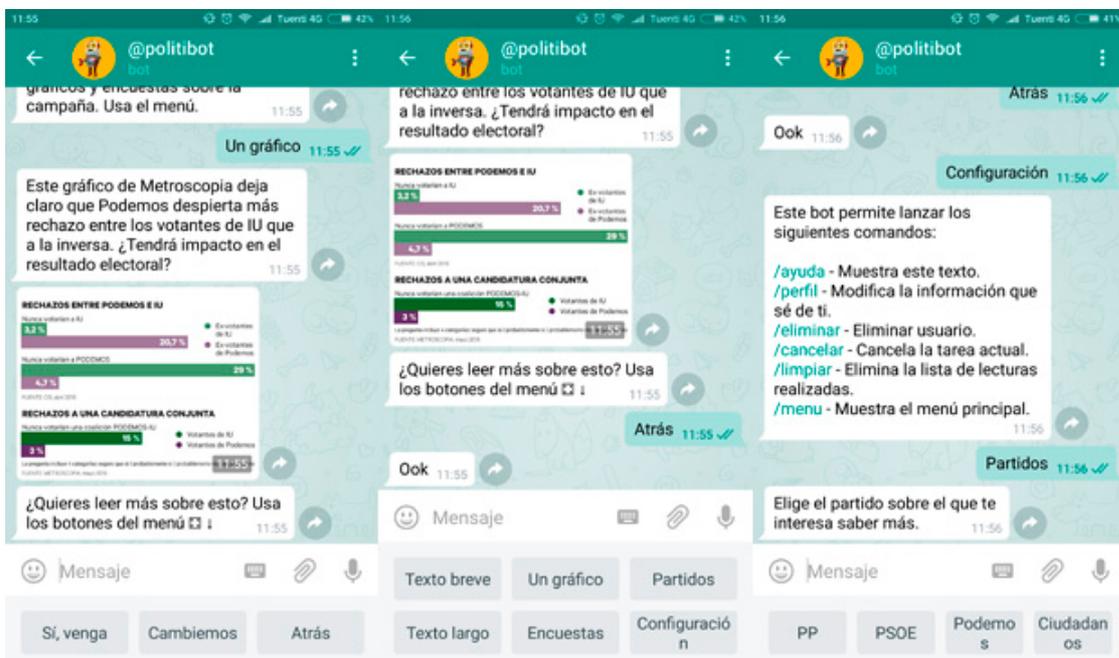


Figura 6: Diferentes capturas de una conversación privada con @politibot [18].

Utilizando la información que el usuario confía en Politibot (sexo, edad y ubicación), es capaz de generar contenido personalizado acorde con los datos relacionados con el usuario. En ocasiones Politibot utiliza una serie de preguntas a las que el usuario puede

contestar expresando su opinión sobre la actualidad política y así recopilar datos y estadísticas sobre los usuarios.

Aunque la versión actual de Politibot no cuenta con ninguna técnica de inteligencia artificial hasta la fecha, ni el uso de ontologías como base de conocimiento, nos ha parecido curioso incluirlo en el estado del arte de este proyecto por la relación que guarda con la temática y arquitectura de nuestro bot.

2.5 Conclusiones

Visto el enfoque de las diferentes aplicaciones e investigaciones analizadas en las secciones anteriores, este proyecto intentará dar un enfoque diferente a los publicados hasta la fecha, visualizando las carencias o diferentes enfoques en cada área.

2.5.1 Herramientas de participación política y ciudadana

Las aplicaciones relacionadas con la participación política y ciudadana, deberían ir más allá de intentar agrupar a un determinado colectivo de la población en un determinado lugar. Pues el hecho de que los ciudadanos se vean implicados en algunas acciones políticas, no difiere mucho de la participación que llevamos adquiriendo en los foros o en las construcciones de *wikis* colaborativas, tecnologías que no aportan nada nuevo al dominio conocido. Estas aplicaciones utilizan tecnologías para la comunicación o el desarrollo de contenidos en diferentes plataformas, pero no integran técnicas de inteligencia artificial, que podrían generar nuevas formas de uso o aprendizaje.

La arquitectura y plataforma propuesta en el proyecto plantea el desarrollo de un bot moderador-politólogo para dar soporte y apoyo en los debates de carácter político o electoral. Haciendo uso de la ontología política desarrollada para extraer información de textos y/o programas electorales que enriquecerán el contenido del debate. Para ello se utilizarán técnicas de extracción de información basadas en ontologías y otras técnicas relacionadas con el procesamiento del lenguaje natural como búsquedas y consultas en diccionarios.

Este enfoque plantea incentivar la participación política de una forma más activa, tratando de que el usuario pueda aprender sobre aquellos temas que desconoce o más le interesan. Fomentando el interés de adentrarse en debates de contenido específico, que serán asistidos mediante información específica que pueda enriquecer el debate y los

contenidos del mismo, utilizando técnicas de inteligencia artificial al igual que se llevan aplicando desde hace tiempo en otros contextos.

2.5.2 *Ontologías relacionadas con la ciencia política*

Desde el campo de las ontologías relacionadas con la ciencia política, hemos podido encontrar algunos trabajos relacionados con tareas administrativas de un gobierno electrónico. Estas ontologías están orientadas al tratamiento y clasificación de la información para tareas de gestión y administración. Lo que puede automatizar y agilizar diversos trámites que puedan requerir acciones de un ciudadano o institución. Además al estar centradas en el campo de la gestión o *eGovernment*, no terminan de cubrir los términos en relación a la política general del un estado. Estos términos y relaciones referencian aquellos implicados directamente con el estado, olvidando otros términos que no tengan relación o gestión directa con un gobierno pero que pertenecen al dominio de la ciencia política y pueden ser incluidos en la ontología. Hasta la fecha las referencias halladas en la literatura no terminan de integrar la visión de la ciencia política desde diferentes niveles. Esto puede denotar una diferencia o brecha entre expertos sobre el conocimiento del dominio con el ciudadano medio, que no abarca más conocimientos que los relacionados directamente con su entorno laboral o familiar.

Sin embargo el planteamiento que trataremos de darle a nuestra ontología será de una labor ilustrativa, de forma que podamos reflejar aquellos términos relacionados con la ciencia política que pueda manejar cualquier ciudadano, aquellos términos que se reflejen en un programa electoral o sean motivo de discusión y/o debate. Destacaremos la metodología peculiar llevada a cabo para definir la ontología, basada en el análisis de determinados dominios y entrevistas con expertos para llegar a un conjunto de términos y relaciones.

De esta forma intentaremos agrupar todos los términos que podrían surgir en un debate relacionado con la ciencia política, independiente del conocimiento que tengan los ciudadanos o la relación que tengan con el estado. Así podremos tomar una ontología para clasificar información, temas o debates relacionados con la ciencia política que pueda ser vista desde diferente perspectivas o conocimiento del dominio.

2.5.3 Bots basados en extracción de información basados en ontologías

En el ámbito de los bots conversacionales también llamados chat-bots, hemos visto el enfoque “humanista” que se basa en torno a mantener conversaciones o diálogos con los usuarios, intentando imitar o simular el comportamiento humano. Estos bots se centran principalmente en tareas de moderación, análisis o monitorización de la actividad que tienen los usuarios con el sistema. Actuando mediante tareas configuradas para definir un comportamiento con los usuarios como prevenir mensajes de spam, recopilar mensajes, expulsarles de la sala o mostrar algún tipo de información sobre el estado del sistema. Sin embargo estos bots no incorporan técnicas de inteligencia artificial para extraer información compleja y adecuada al contexto de las conversaciones con los usuarios para dar soporte a los debates en los chats.

Por ello en el desarrollo del bot-moderador del presente proyecto podemos ver un uso paralelo que va más allá de mantener la conversación con los usuarios. Es decir, de realizar las acciones de un asistente o consultor que pueda manejar cierta información dinámica sin restringirse a un diálogo establecido o unas pautas estrictas. El bot resultante utilizará técnicas de inteligencia artificial para extraer la información de textos y mensajes, procesar la información recibida de los usuarios y contrastarla en su base de conocimientos. Haciendo uso de la ontología desarrollada y un diccionario de términos y sinónimos para contrastar la información de la base de conocimientos del bot con los usuarios.

El dominio utilizado para la temática de los debates será un debate político, enfocado a una campaña electoral donde los usuarios podrán debatir entre las diferentes medidas y propuestas que ofertan los partidos políticos. El bot hará uso de la ontología para centrar el debate, ofrecerles información relacionada de consulta, o aceptar las peticiones de los usuarios para facilitarles el acceso a los datos o la información relacionada con el debate. Así los usuarios podrán contrastar las intervenciones de los demás usuarios con la información sugerida por el bot.

Capítulo 3 - Metodología

En este capítulo se expondrán las diferentes metodologías que se han llevado a cabo para el desarrollo del Trabajo de Fin de Máster.

3.1 Diseño de la Ontología Política

Para llevar a cabo el desarrollo de una Ontología relacionada con términos de un gobierno democrático, podríamos resumirla en las siguientes fases:

- Estudio de la bibliografía existente sobre ontologías de la política en el ámbito de las ciencias políticas.
- Estudiar los diferentes organismos y entidades que componen el gobierno de un estado, en nuestro caso particular el del gobierno de España.
- Analizar las materias sobre las diferentes propuestas y soluciones que tratan los programas electorales de los partidos políticos que se presentan a las elecciones.
- Identificar los términos más relevantes, dependencias y relaciones.
- Implementar la ontología en OWL en la plataforma *protégé*.
- Clasificar las principales temáticas que podrían surgir en un debate político a través de la ontología desarrollada.

Para el desarrollo de la ontología una de las dificultades más complicadas en la fase de especificación, fue tener en cuenta las siguientes perspectivas.

- La perspectiva teórica-filosófica de la ciencia política, cuyo enfoque podemos obtenerlo de los artículos seleccionados en el estado del arte (ver secciones 2.1.1, 2.1.2, 2.1.3 y 2.1.4).
- La perspectiva de la política común, relegada en los programas electorales y servicios públicos (ver secciones 2.1.5, 2.1.6, 2.1.7, 2.1.8 y 2.1.9).
- La perspectiva del ciudadano medio.

En el capítulo 4 del presente documento se detalla el desarrollo de la ontología, las motivaciones que llevaron a ello y el proceso de definición de la ontología política.

3.2 Desarrollo de @DemoCriticsBot

El desarrollo de @DemoCriticsBot como el núcleo principal del proyecto, supuso la elección de una plataforma que nos permitiera el uso de un bot conversacional para interactuar con usuarios. Para ello se llevó a cabo la siguiente metodología:

- Estudiar las posibles soluciones software de bots para poder desarrollar un bot capaz de conversar e interactuar con más de un usuario.
- Analizar las necesidades del Bot, siendo estas las características principales de su funcionamiento de forma autónoma o a petición de los usuarios.
- Implementar el Bot en una plataforma que permita llevar a cabo las necesidades estudiadas.
- Probar el funcionamiento del bot en un escenario real con diferentes usuarios.

El capítulo 5 de la memoria detalla todo el proceso de desarrollo de @DemoCriticsBot.

3.3 Integración de la base de datos de DemoCritics con @DemoCriticsBot

En la última parte del proyecto se planteó la idea de integrar la base de conocimientos del proyecto *DemoCritics* con @DemoCriticsBot, de tal forma que el bot pudiera enriquecer su inteligencia ofreciendo más información y funcionalidades a los usuarios. Para este proceso se llevaron a cabo los siguientes pasos:

- Analizar las necesidades de @DemoCriticsBot para enriquecer su base de conocimientos, descubriendo las necesidades o características que podrían mejorar su conocimiento a través de la base de datos del proyecto *DemoCritics*.
- Descubrir las necesidades de acceso al conocimiento por parte del Bot para poderlas integrar en la base de conocimiento de *DemoCritics*.
- Modificar la base de datos del proyecto *DemoCritics* adaptándose a las necesidades de acceso de @DemoCriticsBot.
- Implementar las peticiones al sistema de conocimiento mediante la API REST desarrollada en Laravel para facilitar el acceso a las consultas que tenga que realizar el bot en cada momento.

En sección 7.2 del documento se detalla el proceso de integración de @DemoCriticsBot con *DemoCritics*.

3.4 Uso de Software Libre

Para el desarrollo del software que compone el proyecto se ha dado prioridad a las soluciones de software libre. Lo que nos permite una aportación al común, fomentar el uso de estas herramientas y permitir que cualquier persona pueda hacer uso del código desarrollado. Se ha utilizado software libre tanto para las herramientas necesarias de desarrollo de código, servidores de back-end y documentación. Así como también el proyecto está disponible en la comunidad de desarrolladores GitHub con una licencia GNU GPLv3 [22]. La tabla 1 muestra un breve resumen de las diferentes herramientas utilizadas para el proyecto acompañadas de las licencias que poseen.

3.5 Repositorio y control de versiones

Para el desarrollo del proyecto se llevó a cabo una política de control de versiones, utilizando GIT como herramienta para mantener las copias locales del proyecto. Posteriormente el proyecto se subió al portar web GitHub, donde se pueden visualizar los *commits* con cambios y modificaciones del proyecto. La organización con los diferentes repositorios que engloban al proyecto, puede consultarte en la siguiente dirección: <https://github.com/PoliticalBot>.

Todo el repositorio queda disponible para su visualización, modificación o uso bajo una licencia de software libre GNU GPL v3.

Software	Licencia
Protégé	BSD 2-clause
Cloud9 IDE	GNU GPL v3
Laravel	MIT License
phpMyAdmin	GNU GPL v2
MySQL	GNU GPL
PHP	PHP License
WordNet	BSD license
Python	PFSL
python-telegram-bot	GNU LPGL v3
NLTK	Apache License 2.0
OntoSpy	GNU GPL v3
Telegram Desktop	GNU GPL v3
OpenShift	Apache License 2.0

Tabla 1: Software libre utilizado en el desarrollo del proyecto.

Capítulo 4 - Tecnologías del Proyecto

En las siguientes secciones se expone una breve descripción de las diferentes tecnologías y herramientas que se han utilizado en el desarrollo del proyecto. En particular se hablarán de las herramientas utilizadas en el desarrollo de la ontología política (ver capítulo 6), el desarrollo de @DemoCriticsBot (ver capítulo 7) y las herramientas de la integración de @DemoCriticsBot con *DemoCritics*.

Tipo	Nombre	Uso	Sección
Framework	Telegram Bot API	Funciones y uso para la interacción de Bots en conversaciones con usuarios de Telegram.	4.2.2
	python-telegram-bot	Librería para Python que realiza las llamadas a la Telegram Bot API.	4.2.4
	NTK 3.0	Kit de herramientas para la consulta de palabras y sinónimos en el diccionario WordNet.	4.2.6
	OntoSpy's	Librería desarrollada en Python para el procesamiento de Ontologías en OWL.	4.2.10
	Laravel 5	Ofrece la información alojada en la base de datos mediante peticiones HTTP a una API Rest.	4.3.1
Servidor	Cloud9	Aloja y mantiene en ejecución el Bot desarrollado para Telegram.	4.2.7
	OpenShift	Aloja la API Rest para las consultas a la base de datos.	4.3.3
Lenguaje	OWL	Lenguaje utilizado para definir la ontología política.	4.1.1
	Python	Lenguaje en el que está programado el Bot de Telegram.	4.2.3
	PHP	Lenguaje para definir la API Rest e interactuar con la base de datos.	4.2.9
Base de Datos	WordNet	Diccionario de consulta de términos y sinónimos.	4.2.5
	MySQL	Lenguaje para realizar consultas en la base de datos.	4.3.2
	phpMyAdmin	Plataforma para modificar la base de datos propuesta por <i>DemoCritics</i> .	
Control de Versiones	Git - GitHub	Repositorio y línea de comandos para alojar los espacios de trabajo del proyecto.	4.2.8
IDE	protégé	IDE para implementar la ontología política en OWL.	4.1.2
	Cloud9	IDE para desarrollar y ejecutar el Bot implementado en Python.	4.2.7

Tabla 2: Tecnologías utilizadas en el proyecto.

Para hacerse una idea general de las tecnologías y herramientas utilizadas se recomienda ver el gráfico que se muestra en la tabla 2.

4.1 Tecnologías para Ontologías

4.1.1 OWL

OWL [23] es el acrónimo de *Ontology Web Language*, un lenguaje de marcado para publicar y definir ontologías en la *World Wide Web* (WWW). Fue diseñado para las aplicaciones que necesitan procesar el contenido de cierto tipo de información en lugar de mostrar información al usuario. OWL facilita un modelo de marcado construido sobre RDF y codificado en XML. El proyecto utilizará el lenguaje OWL para definir la ontología de términos políticos.

4.1.2 protégé

protégé [24] es un editor libre de código abierto con licencia BSD desarrollado por la Universidad de Standford, y en colaboración con la Universidad de Mánchester. Desde el editor podemos desarrollar e implementar nuestras propias ontologías en los lenguajes OWL y RDF de forma visual. Está desarrollado en Java y utiliza frecuentemente la librería Swing para su entorno gráfico. Es posible añadir ciertos plugins que necesitemos en nuestro proyecto, y cuenta con el razonador libre HermiT que nos permite realizar todo tipo de consultas sobre la ontología desarrollada.

También disponen del portal *WebProtégé* el cual nos permite disfrutar de todas las características de la versión de escritorio de *protégé* y acceder a un repositorio de ontologías desarrolladas por el mismo editor. Se utilizará el editor *protégé* para implementar y evaluar la ontología desarrollada en OWL.

4.2 Tecnologías para el Bot conversacional

4.2.1 Telegram Messenger

Telegram [25] es una aplicación de mensajería instantánea lanzada en 2013 para los dispositivos móviles basados en iOS, Android y Windows Phone. Al igual que otras aplicaciones de mensajería instantánea como *WhatsApp*, con Telegram podemos mantener conversaciones privadas, en grupo, cifrar nuestros chats, suscribirnos a canales e interactuar con bots conversacionales creados por otros usuarios. Los clientes tanto

para web, escritorio y móvil están publicados bajo una licencia de software libre GNU GPLv3. Actualmente dispone de más de 2 millones de usuarios activos para Android.

Se utilizará la aplicación de escritorio Telegram Desktop para evaluar y poner a prueba las funcionalidades del bot desarrollado.

4.2.2 Telegram Bot API

A principios de 2015 Telegram publica su Bot API [20] que permite a los desarrolladores implementar Bots para la aplicación. Tras crear nuestro bot recibiremos un identificador con el que podremos realizar una serie de llamadas a la API para el desarrollo del bot. La API es bastante completa y suele ser actualizada con frecuencia, entre otras funciones nos permite interactuar con los usuarios de la aplicación, mandarles mensajes de texto, imágenes, vídeos, etcétera. También podemos recopilar cierta información sobre el usuario con el que estamos interactuando si ha decidido conversar con el bot.

Además tendremos la opción de crear teclados virtuales (ver figura 7) cuando queramos recoger la opinión del usuario u ofrecerle un menú de características para que el usuario pueda desplazarse como si se tratara de un menú de opciones.

La conexión al Bot de Telegram y las llamadas a la API se realizarán desde la interfaz python-telegram-bot (ver sección 4.2.4).

4.2.3 Python

Python es un lenguaje de programación cuya filosofía es proporcionar un lenguaje libre y flexible al programador, con una sintaxis que favorezca un código legible. Es un

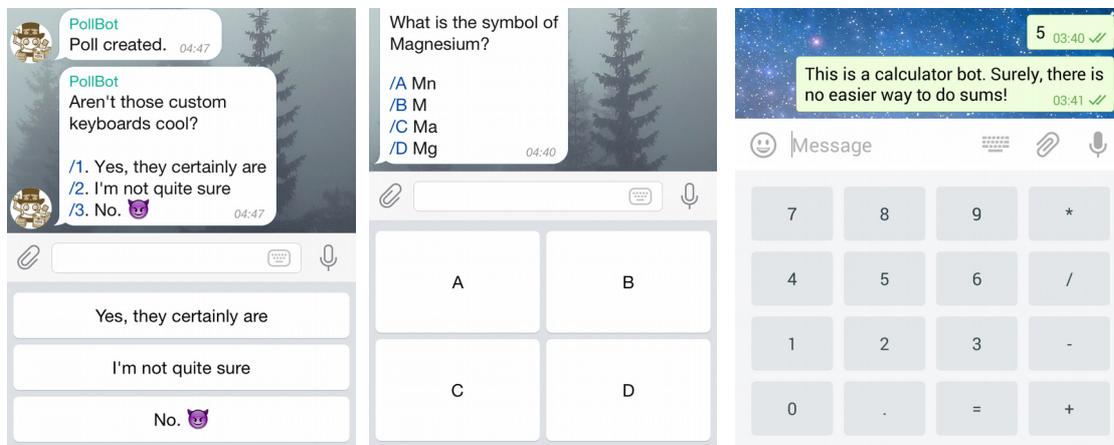


Figura 7: Diferentes teclados virtuales con respuestas personalizadas [20].

lenguaje de programación multiparadigma, es decir, que soporta orientación a objetos, programación imperativa, funcional y herencia múltiple.

Actualmente es administrado por la *Python Software Foundation* y posee una licencia de código abierto denominada *Python Software Foundation License*, que es compatible con la licencia GNU GPLv2.

4.2.4 *python-telegram-bot*

Es una librería publicada en la comunidad GitHub que ofrece una interfaz en python para realizar llamadas y acciones a la Telegram Bot API (ver sección 4.2.2). Es compatible con las versiones de Python 2.7, 3.3+ y con el intérprete y compilador PyPy. Actualmente soporta todos los métodos publicados en la Telegram Bot API hasta el 28 de Mayo de 2016. Está publicada bajo una licencia de software libre LGPL-3.

4.2.5 *WordNet*

WordNet [26] es una base de datos léxica que agrupa palabras en inglés en conjuntos de sinónimos (synsets), proporcionando definiciones generales y guardando relaciones semánticas entre diferentes conjuntos de sinónimos. El objetivo principal de WordNet es ofrecer un servicio entre diccionario y tesoro, y así poder realizar y procesar análisis de texto para aplicaciones basadas en inteligencia artificial. Tanto la base de datos como las herramientas software para utilizar el diccionario, poseen una licencia BSD, lo que permiten que puedan ser descargadas y utilizadas libremente.

En este proyecto se utilizará WordNet procesar la entrada del usuario y procesarla con la biblioteca NLTK (ver sección 4.2.6).

4.2.6 *Natural Language Toolkit - NLTK*

El Natural Language Toolkit [27] comúnmente conocido como NLTK, es un conjunto de librerías y programas para el procesamiento del lenguaje natural simbólico y estadístico para el lenguaje de programación Python [28]. NLTK está orientado a las áreas de investigación relacionadas con el PLN que incluyen técnicas de inteligencia artificial, extracción de la información de textos y aprendizaje automático. Incluye el diccionario WordNet y un conjunto de funcionalidades que nos permitirán realizar consultas de términos y sinónimos en diferentes idiomas. NLTK tiene publicado el código fuente en la plataforma GitHub con una licencia de software libre Apache 2.0.

4.2.7 Cloud9

Cloud9 [29] es un entorno de desarrollo integrado (IDE) online, que soporta diferentes lenguajes de programación como PHP, Ruby, Perl, Python, JavaScript... Funciona al igual que cualquier entorno de desarrollo de escritorio, incluyendo un editor de código, una consola de comandos para la ejecución del código y un depurador. Además Cloud9 permite alojar nuestros proyectos en la nube y dejarlos ejecutando, por lo que ofrece las mismas características de servidor web. Dispone de diferentes modalidades partiendo de una gratuita hasta otras de pago más ambiciosas. La plataforma está desarrollada en JavaScript y su desarrollo está publicado en GitHub con una licencia GNU GPLv3. El desarrollo y ejecución del código de @DemoCriticsBot quedará alojado en un workspace público de la plataforma Cloud9.

4.2.8 Git – GitHub

Git es un Sistema de Control de Versiones (VCS) distribuido y open source (bajo licencia pública GNU) diseñado para gestionar distintas versiones en una aplicación independientemente del tamaño y del número de personas que trabajan en ella. Creado por Linus Torvalds en 2005 para trabajar en el desarrollo del kernel de Linux.

GitHub [32] es una plataforma web lanzada en 2008 que permite alojar de forma gratuita y en sitios llamados repositorios, proyectos software que utilicen Git como Sistema de Control de Versiones. Los repositorios gratuitos son públicos y accesibles por todo el mundo, de manera que cualquiera puede participar en el desarrollo del código.

En este proyecto utilizaremos Git para el control de versiones en un entorno de trabajo local, para posteriormente subir el proyecto a un repositorio público en GitHub.

4.2.9 PHP

PHP es un lenguaje de propósito general del lado del servidor diseñado originalmente para crear aplicaciones web que generaran contenido dinámico. Diseñado en 1996 por Rasmus Lerdorf, es de código libre y se distribuye bajo licencia PHP. Tiene la ventaja de poder ser fácilmente incorporado dentro de los documentos HTML.

Para este proyecto se utilizará el lenguaje PHP para programar el comportamiento del Service REST que hace de intermediario entre las peticiones del bot desarrollado en Telegram y la base de datos MySQL de *DemoCritics*.

4.2.10 *OntoSpy's*

OntoSpy es una librería desarrollada para python y herramienta de línea de comandos, para inspeccionar y visualizar ontologías codificadas en RDF y OWL. OntoSpy será el componente para la arquitectura propuesta encargado de interactuar y realizar consultas a la ontología desarrollada en OWL.

4.3 Tecnologías para la integración con *DemoCritics*

4.3.1 *Laravel*

Laravel [30] es un framework que permite desarrollar aplicaciones y servicios web con PHP 5. Fue desarrollado por Taylor Otwell en 2011 y su versión actual 5.0 se distribuye en forma de código abierto (bajo licencia MIT) en su propio repositorio publico en GitHub. Su funcionalidad es extensible mediante módulos. Permite desarrollar un servicio back-end proporcionando una API REST para realizar peticiones HTTP desde cualquier plataforma.

Se utilizará la base desarrollada por el proyecto *DemoCritics* para realizar llamadas a la base de datos de conocimientos sobre los programas electorales. Además, ampliaremos la funcionalidad de llamadas y peticiones a la API para adaptarla a las necesidades del presente proyecto.

4.3.2 *MySQL*

MySQL es un Sistema Gestor de Base de Datos (SGBD) Relacionales que permite el almacenamiento, creación y modificación de dicho tipo de Bases de Datos. Desarrollado por Oracle, su versión actual (5.7.15) se distribuye tanto en forma de Software libre (bajo licencia GPL) o de uso comercial.

Al igual que en el Service REST desarrollado en Laravel (ver sección 4.3.1), adaptaremos la base de daos del proyecto *DemoCritics* añadiendo nuevas tablas y relaciones que necesitemos en el proyecto.

4.3.3 *OpenShift*

OpenShift [31] es una plataforma de computación en la nube que ofrece alojar servicios de forma gratuita en sus servidores mediante un modelo de arquitectura Software

as a Service (SaaS). Disponible desde 2011, se distribuye bajo licencia Apache 2.0. Dispone también de planes de pago que aumentan las prestaciones del servidor.

En este proyecto se utilizará OpenShift como copia del alojamiento para la base de datos y las llamadas a la API de la aplicación *DemoCritics*, siendo esta la base de conocimientos que de acceso a la información almacenada en la base de datos del sistema. El Servicio Web REST y la Base de Datos. Concretamente el servidor esta accesible desde la siguiente URL: <https://apirestpoliticalbot-servicerest.rhcloud.com>.

Capítulo 5 - Arquitectura del proyecto

En este capítulo se estudiarán los aspectos técnicos relacionados con la arquitectura y organización del Software desarrollado en el proyecto.

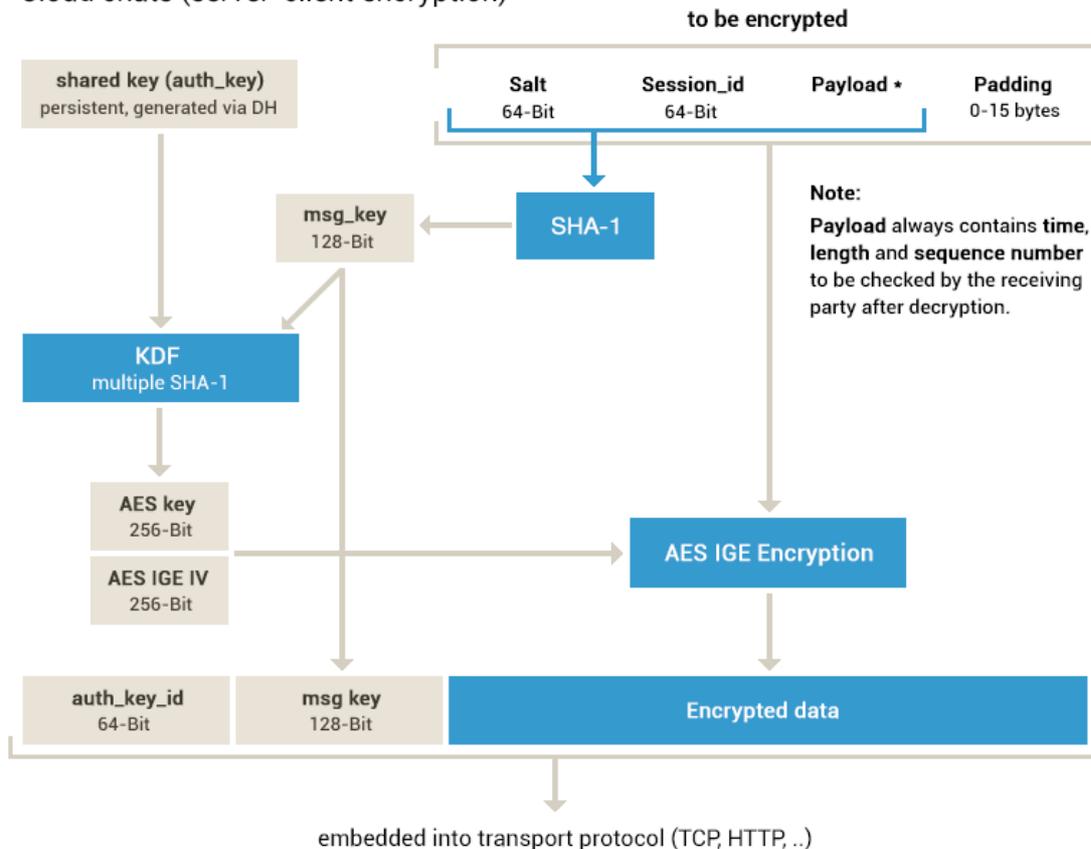
5.1 Telegram MTPROTO Mobile Protocolo

Telegram MTPROTO [54] son las siglas en inglés de *Mobile Transport Protocol (Protocolo de Transporte Móvil)*, es el protocolo para el envío de datos que utiliza Telegram Messenger. Está desarrollado bajo un estándar abierto a base de una API Java, desarrollado por el matemático Nikolai Durov y financiado por Pável Dúrov.

A diferencia de otros protocolos como XMPP [33] utilizados por otras aplicaciones de mensajería instantánea como Facebook, WhatsApp o Tuenti entre otras, MTPROTO está

MTPROTO, part I

Cloud chats (server-client encryption)



NB: after decryption, msg_key must be equal to **SHA-1** of data thus obtained.

Figura 8: Tráfico de los datos en los chats normales mediante MTPROTO [54].

enfocado en la multisesión multiplataforma, lo que nos permite mantener abiertas varias sesiones en diferentes dispositivos, y el transporte de archivos sin importar el formato o la capacidad.

El tráfico de los datos está compuesto por dos tipos de cifrados, ambos con AES de base, diferenciando entre chats normales y chats secretos, siendo este último el más seguro de utilizar contra la filtración de información sensible. Para los chats normales denominados *clouds chats*, se utiliza como cifrado cliente-servidor, lo que permite leerlos en diferentes sesiones y dispositivos de una misma cuenta. Los mensajes son codificados simétricamente con AES-256 y la clave compartida está formada en SHA-1. La figura 8 muestra la estructura del tráfico para los chats normales en *MTPProto*.

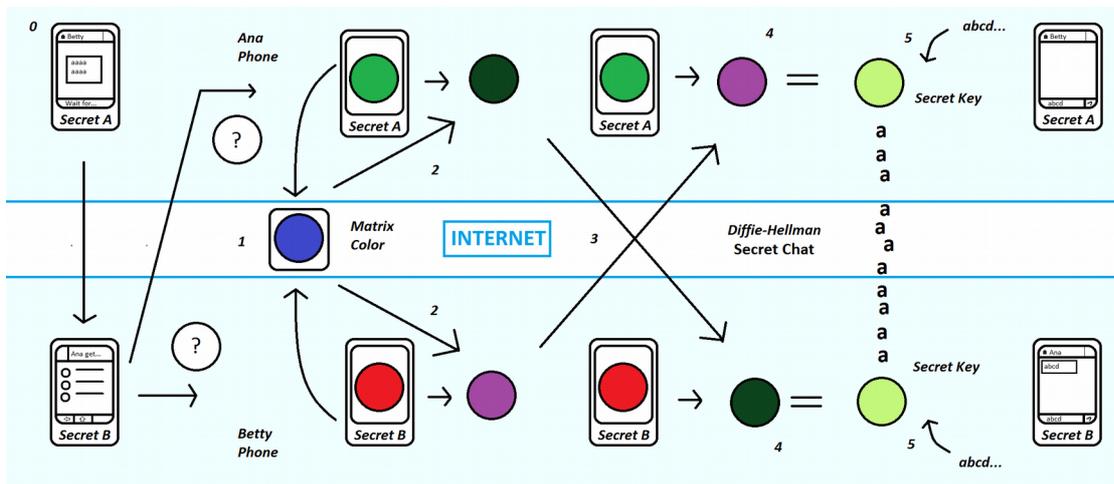


Figura 9: Tráfico de los datos en los chats normales mediante MTPProto [55].

Para los chats secretos se cifran de extremo a extremo, de forma que sólo se puede visualizar en un único dispositivo, aquel que inició la conversación. Utilizan el algoritmo SHA-1 y cifrado XOR de 128 bits para la forma digital. La figura 9 ilustra un esquema de funcionamiento en alto nivel para descifrar y acceder al contenido de los mensajes secretos a través de la clave secreta compartida.

5.2 Una arquitectura para bots moderadores-consultores basados en ontologías

En esta sección vamos a describir una arquitectura genérica que nos servirá para la implementación de bots moderadores-consultores basados en ontologías. Donde la ontología puede ser cualquier ontología genérica en cualquier contexto. Para ello partiremos implementando el bot en la plataforma Telegram Bot [44], y tomando una ontología diseñada en OWL. Necesitaremos algunos componentes o herramientas para realizar las llamadas a la Telegram Bot API [20], dar acceso al bot a la ontología, realizar consultas con un diccionario taxonómico como WordNet [26] e implementar la conexión con la base de datos donde mantenga el conocimiento. La figura 10 muestra un esquema de la arquitectura descrita en esta sección junto con los principales componentes que se detallan en la tabla 3.

Para implementar esta arquitectura del bot moderador-consultor basado en ontologías se propone el uso de los siguientes componentes:

Componente	Descripción	Sección
Bot-behaviour	Biblioteca desarrollada en Python para realizar todas las llamadas necesarias a la Telegram Bot API. Con ella definiremos todo el comportamiento base de nuestro bot, definiendo la interacción con los usuarios, el envío de mensajes e información, el procesamiento de los mensajes recibidos de los usuarios.	5.3.2
Ontology-connector	Biblioteca desarrollada para Python realizaremos la conexión del bot con la ontología. Con OntoSpy podremos extraer toda la información de ontologías desarrolladas en RDF, SKOS u OWL en nuestro caso. Cabe destacar que este componente se limita a extraer la información de las clases, relaciones y otra información relacionada con la ontología. Por tanto la utilizaremos como herramienta de consulta, no como edición o alteración de la ontología.	5.3.5
Domain-language-processing	Conjunto de herramientas desarrollado para Python será de gran utilidad para el procesamiento de las comunicaciones de los usuarios con el bot. Especialmente para la consulta del diccionario WordNet.	5.3.3
Knowledge-base-connector	Utilizar una API Rest para gestionar las llamadas a la base de datos donde guardemos el conocimiento del bot, puede ser una buena práctica si luego quisiéramos utilizar otros sistemas que pudieran dar utilidad a la misma API. En otro caso habría que implementar algún conector de base de datos como el paquete <i>mysql.connector</i> .	5.3.4

Tabla 3: Componentes principales para la arquitectura del bot-moderador basado en ontologías.

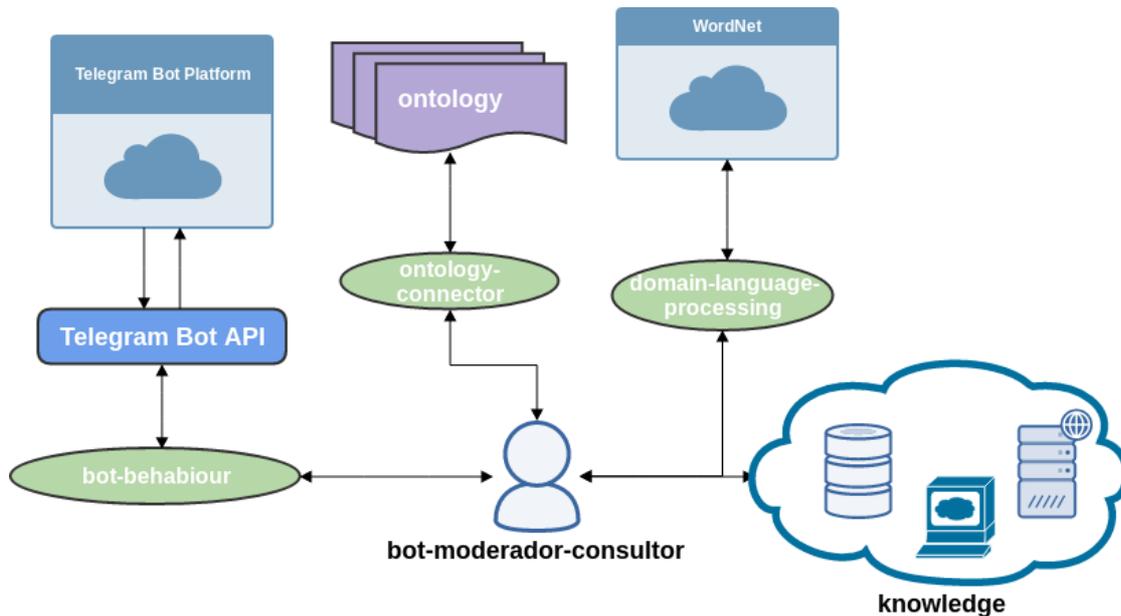


Figura 10: Arquitectura propuesta para el bot-moderador-consultor.

Como se puede comprobar la mayoría de los componentes parten del lenguaje Python, la elección de Python como lenguaje de programación base para la arquitectura se debe a varias razones. Por un lado, la flexibilidad que ofrece Python para desarrollar aplicaciones nos permite obtener grandes resultados en pocas líneas de código, algo muy favorable si queremos que nuestro sistema sea reutilizado. Mientras que por otro lado tenemos una gran cantidad de librerías y componentes para enriquecer nuestras aplicaciones en este caso con el Kit NLTK, la librería para conectar nuestro bot a la Telegram API o el caso de OntoSpy para extraer información de ontologías. Otros lenguajes como Java también disponen de herramientas similares, pero si bien la implementación y el código puede ser más complejo para exportarlo en otros usos. En cuanto al almacenamiento y consulta del conocimiento, se plantea la posibilidad de almacenarlo en bases de datos e incluso realizar consultas de Internet para poder extraer información en cualquier momento. Todos los componentes descritos contienen una licencia de software libre, por lo que se permite la reutilización de los mismos.

5.3 Arquitectura de la aplicación

La aplicación está formada por dos cuatro módulos principales en los que detallaremos en las siguientes sub-secciones. La arquitectura de la aplicación gira entorno al Bot alojado en el servidor Cloud9, quien se va a encargar de interactuar con los cuatro componentes de la aplicación. En primer lugar el bot realizará todas las comunicaciones con los usuarios a través de la aplicación Telegram Messenger, que nos ofrece dife-

rentes tipos de clientes tanto para iOS, Android, Windows Mobile, Escritorio o Web. La comunicación e interacción con los usuarios se realiza gracias a la Telegram Bot API conectada mediante la librería *python-telegram-bot*. Para la consulta de términos en el diccionario WordNet será facilitada por la interfaz NLTK. La consulta a ciertas partes de conocimiento se harán mediante peticiones HTTP a la API Rest desarrollada en Laravel, el cual nos dará la conexión con la base de datos. La figura 11 muestra un esquema en alto nivel de los componentes que forman el proyecto. Por último, la información de la ontología será extraída mediante la librería OntoSpy, que realizará las consultas cuando sean necesarias. Además cabe la posibilidad de que el bot pueda realizar consultas a otros servidores, bases de datos y/o páginas web ajenas al proyecto donde pueda extraer información de interés.

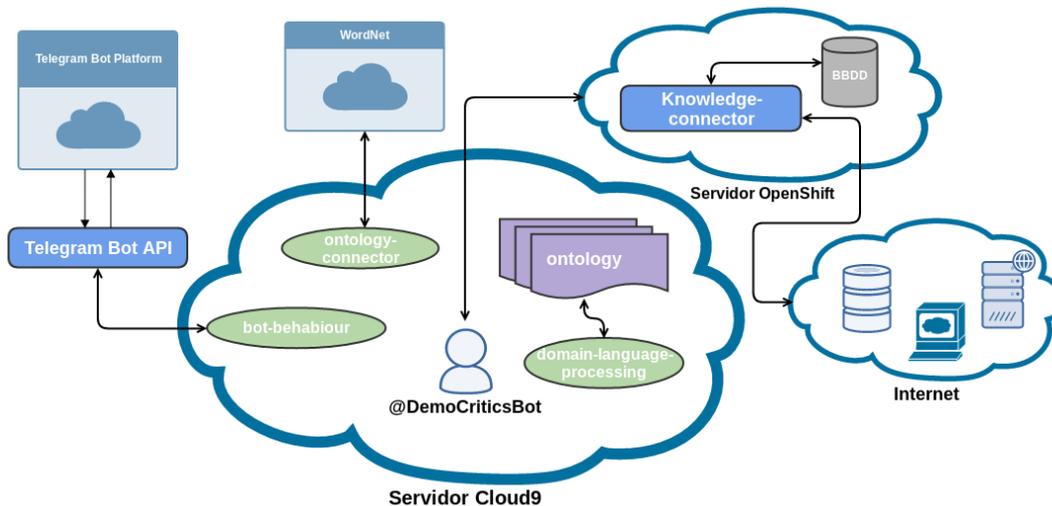


Figura 11: Estructura general de la aplicación.

5.3.1 Telegram Bot API

La API Bot de Telegram utiliza una interfaz basada en peticiones por GET y POST por HTTP. Para autenticar nuestro Bot deberemos solicitar nuestro identificador llamado *token* que tomará una forma similar a la siguiente cadena de texto: 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11. Para realizar las peticiones deberemos hacerla vía HTTPS ejecutando la siguiente consulta:

```
https://api.telegram.org/bot<token>/METHOD_NAME
```

Añadiendo nuestro token facilitado por Telegram ([@BotFather](#)), indicando el método de los disponibles por la API que queremos utilizar. Normalmente utilizaremos peticiones GET para métodos sin argumentos, y POST para aquellos métodos que necesiten

utilizar algún argumento como parámetro de la función. Todas las peticiones utilizarán la codificación UTF-8.

Desde la página oficial de la Telegram Bot API [20] se puede visualizar el listado completo de todos los métodos hasta la fecha.

5.3.2 python-telegram-bot

La librería python-telegram-bot nos ofrece una capa de abstracción entre lo que serían las llamadas a la Telegram Bot API mediante peticiones HTTP, y la interacción del Bot con los usuarios. Esta librería nos facilita la interacción de llamadas con la API mediante clases e interfaces que representan las funcionalidades de nuestro bot y nos permiten realizar acciones con los usuarios.

Las dos clases más destacadas que nos ofrece la librería son telegram.ext.Updater and telegram.ext.Dispatcher. La clase *Updater* es la encargada de recibir las actualizaciones y eventos de Telegram, y las comparte con la clase *Dispatcher*. La clase

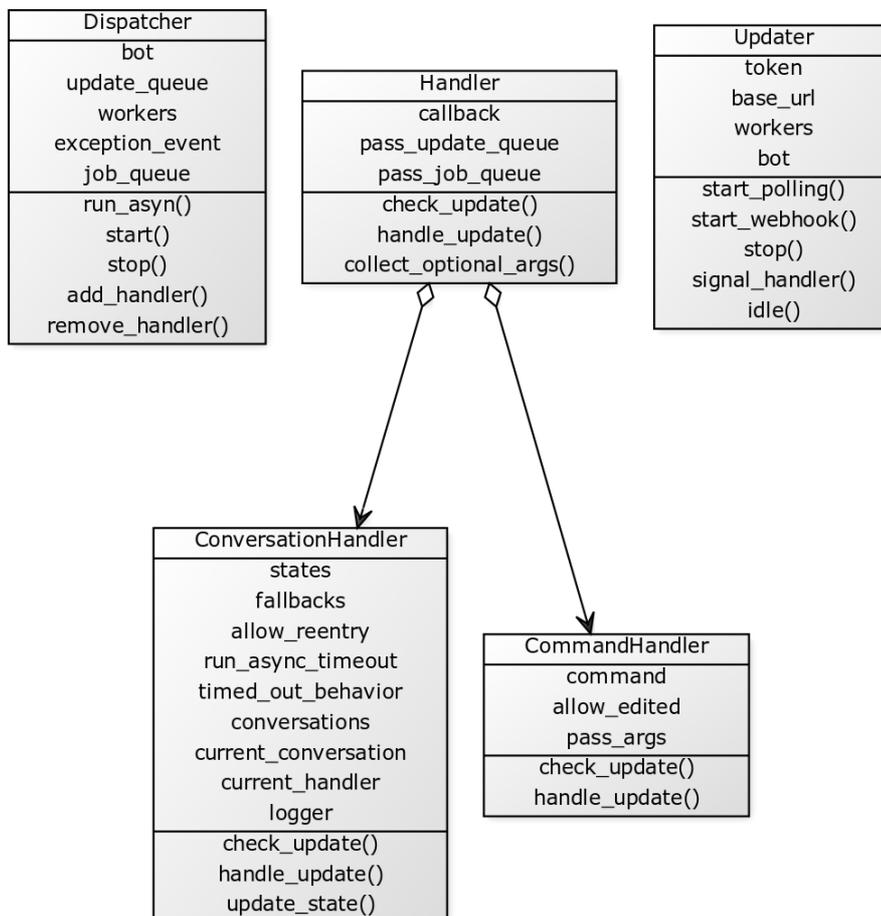


Figura 12: Relación de clases en la Telegram Bot API.

Updater será la encargada de almacenar el identificador del bot (token, ver sección 5.2.1), para poder realizar las peticiones y obtener las actualizaciones.

Para definir acciones, respuestas y comandos tendremos las clases llamadas *Handler*. Estas clases facilitan la programación al desarrollador ofreciéndole una sencilla interfaz para definir los métodos y registrar los comandos definidos en el Bot. Un ejemplo de uso sería la clase *ComandHandler*, que hereda de la clase *Handler*, el cual actúa como un *listener* o escuchador de los comandos definidos en el bot., de tal forma que ejecute el código o método definido cada vez que un usuario realice la llamada a ese comando.

Cada evento registrado como *Handler* recibirá dos parámetros; *bot* y *update*. El parámetro *bot* hará referencia a toda la funcionalidad disponible que podamos realizar con el bot, ya sea envío de mensajes, texto, vídeo... Toda la interacción con los usuarios. Mientras que el parámetro *update* hace referencia a la última actualización que ha producido este evento. Esto podría hacer referencia al comando de donde proviene, al mensaje escrito por un usuario u otro evento similar. La figura 12 contiene un diagrama a modo resumen con algunas de las clases y métodos citados.

Las funciones disponibles en la librería poseen el mismo nombre que la Telegram Bot API, al igual que los parámetros que requiere cada método. Los parámetros más comunes serán el *chat_id*, el cual representa el identificador único de la conversación donde está interactuando el Bot, que puede ser obtenido como atributo desde el parámetro *update*. Otro parámetro que se utilizará con frecuencia será *text*, el cual representa el texto, información u enlace que requiera enviar el Bot a los usuarios del chat. En caso de necesitar enviar una imagen, gráfico, documento o audio, este parámetro será sustituido por el objeto en cuestión. También tendremos la posibilidad de utilizar otros parámetros como *reply_to_message_id*, en caso de que queramos responder la respuesta de un usuario concreto.

Para registrar los eventos relacionados con las respuestas de los usuarios, utilizaremos la clase *MessageHandler*, cual tendremos que registrar en la clase *Dispatcher* junto a los comandos definidos al principio de la ejecución de la aplicación.

5.3.3 Domain-language-processing

Desde el conjunto de herramientas que proporciona NLTK, principalmente utilizaremos la interfaz que pone a nuestra disposición para WordNet. Para ello se utilizará el componente *wordnet* dentro de la librería *nltk.corpus*.

Para procesar la entrada del usuario desde el chat que interactúa con el bot, utilizaremos la llamada a *synset* para obtener un conjunto de sinónimos relacionada con esa palabra. Cabe destacar que por defecto WordNet tomará la entrada y la salida de la función con términos en inglés. Pero podemos modificar tanto la entrada como la salida especificándolo en la llamada a *synset*. De esta forma nos devolverá un array con el listado de posibles sinónimos relacionada con cada palabra que le hemos pasado como argumento.

También es posible analizar el significado de los términos obtenidos mediante la función *hypernyms()*, el cual nos ofrecerá un listado de posibles sinónimos en castellano obtenidos de la base de datos de WordNet. Desde la página web oficial de NLTK [35] podemos ver todas las posibilidades que nos ofrece con WordNet.

5.3.4 API Rest

Para establecer la conexión con la base de datos desarrollada por *DemoCritics* y ampliada para su uso en este proyecto, se ha utilizado Laravel como framework para de-

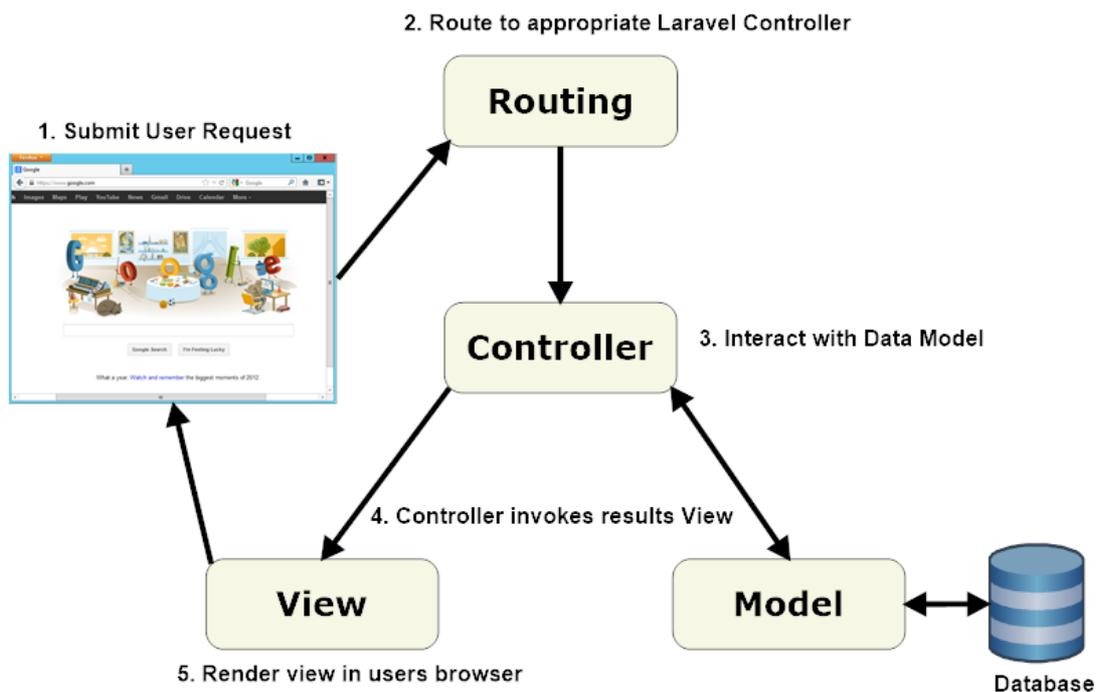


Figura 13: Arquitectura de Laravel, modelo MVC [56].

sarrollo de servicios Web. Laravel utiliza e implementa un funcionamiento basado en el patrón MVC (*Model-View-Controller*), separando el modelo de la vista, y delegando la gestión al controlador. En este caso tendríamos un modelo almacenado en las tablas de la base de datos, donde se guardará la información y base de conocimientos del Bot. El controlador será el encargado de gestionar las peticiones del cliente en Android en función del tipo de operación que requiera.

Por último, se tendría como vista el resultado devuelto por el servidor con los datos solicitados por el cliente, que sería adquirido y procesado por el Bot. Con esto se consigue elaborar un código más claro y sencillo, así como también evitar conflictos entre el modelo y la vista sin alterarlos.

5.3.5 *Ontology-connector*

OntoSpy permite extraer información de ontologías desarrolladas en RDF y OWL. Este componente nos será de gran utilidad para realizar todas las consultas necesarias a la ontología. Una vez cargado el archivo o dirección donde tengamos almacenada la ontología, OntoSpy nos mostrará un esquema con las clases, propiedades, relaciones e instancias encontradas en la ontología. Esta información será accesible desde el objeto que hayamos almacenado la ontología, desde el que podremos acceder a la información a través de sus atributos cargando la clase correspondiente.

El funcionamiento de OntoSpy se basa en RDFLib, una librería para Python que permite leer y representar información almacenada en RDF.

Capítulo 6 - Diseño de una Ontología política

A continuación se detalla el proceso y las motivaciones que llevaron a cabo en el desarrollo de una ontología política.

6.1 Entrevistas con expertos

Durante el desarrollo del proyecto *DemoCritics* [12] en 2015 tuve la oportunidad junto a mi compañero de poder realizar entrevistas con expertos en el ámbito de la ciencia política y el activismo político. Concretamente pudimos realizar dos entrevistas; una con dos de los miembros de Labodemo [36] y otra con el abogado, activista Javier de la Cueva [37].

La primera entrevista realizada con dos miembros de Labodemo, eran desarrolladores de la plataforma para la redacción colaborativa del programa electoral de la plataforma municipal ciudadana Ahora Madrid, puesta en práctica para las pasadas elecciones municipales de 2015. En esta conversación se planteo la idea de cómo clasificar aquellas categorías, temas, propuestas que diera lugar a una información más organizada y faci-

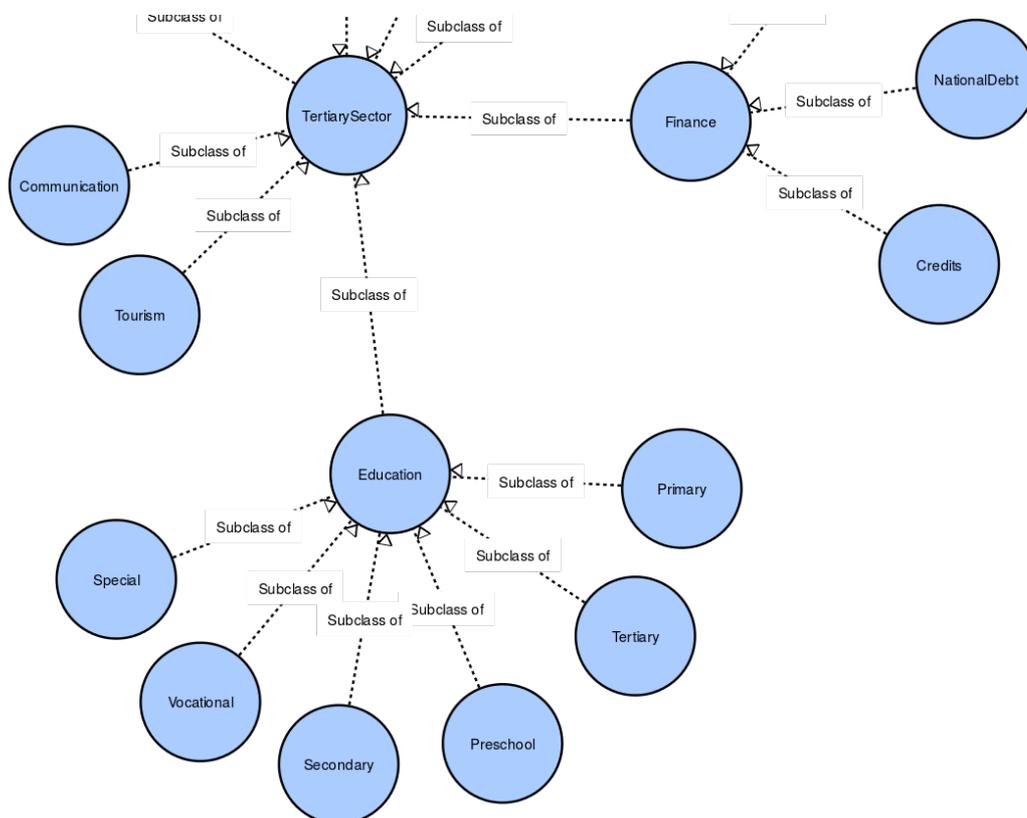


Figura 14: Algunas clases de la ontología pertenecientes al sector terciario.

litara el acceso de filtración para aquellos usuarios que no tienen un cierto dominio sobre el tema. Por ello se habló de la posibilidad de organizar la información en “hilos” como foros o sub-temas de un foro. De tal forma que los usuarios tuvieran libertad para organizar la información en base a unos hilos generales.

Sin embargo la entrevista con Javier de la Cueva fue más allá, pues destacó la importancia de categorizar el contenido de la información. Un usuario que no tenga conocimientos sobre diversos temas, se encontraría más cómodo si pudiera visualizar las diferentes partes de un programa o las propuestas ciudadanas por categorías o temas generales. Ya que dejar libertad a los usuarios para crear categorías personalizadas podría ser algo negativo para usuarios inexpertos o con pocos conocimientos sobre temas específicos.

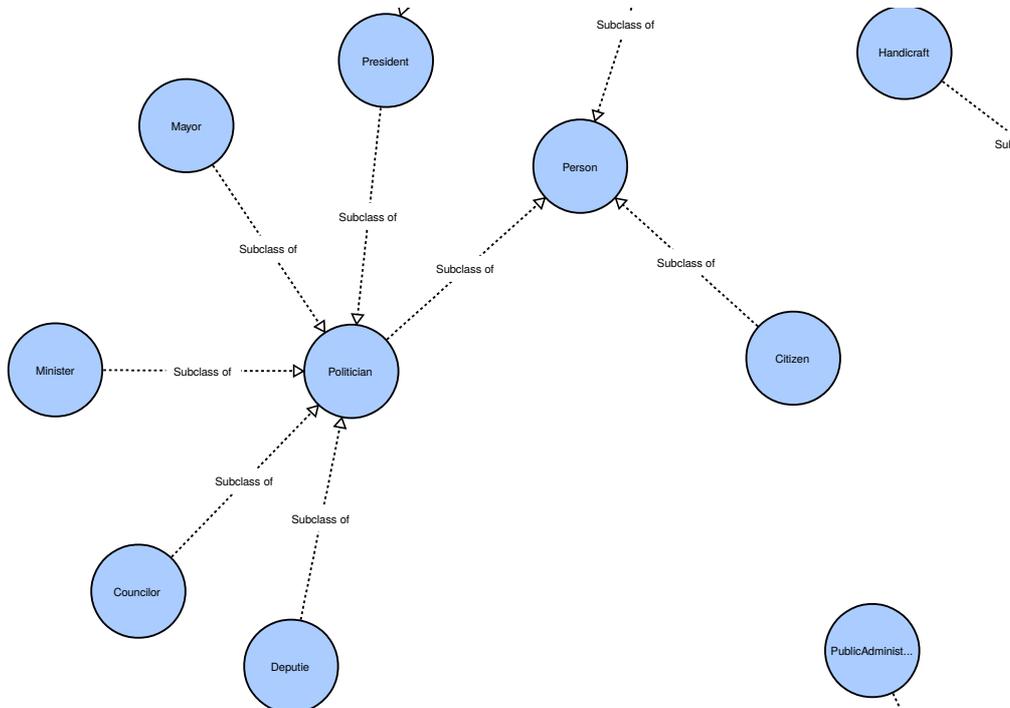


Figura 15: Diferentes cargos de un político.

Aunque el proyecto *DemoCritics* no pudo abordar ese problema, en este proyecto se plantea una de las soluciones para clasificar esta información. Abordando el problema desde la necesidad de tener que orientar al usuario a clasificar la información o saber dónde y cómo puede buscarla. Es por ello que al principio de este TFM se estudió la posibilidad de desarrollar una ontología política para que posteriormente una aplicación pudiera emplearla para facilitar el acceso a la información a los usuarios, garanti-

zar una correcta clasificación de los contenidos y orientar a los usuarios al uso correcto de la misma.

6.2 Diferentes perspectivas en una ontología común

Como se detalla en la sección 3.2 en la metodología para el desarrollo de la ontología se pueden diferenciar tres perspectivas a tener en cuenta:

- La perspectiva teórica-filosófica de la ciencia política.
- La perspectiva de la política común.
- La perspectiva del ciudadano medio.

La primera perspectiva, quizá la más compleja por guardar menos relación con las ciencias de la computación, pertenece al estudio de la visión teórica y filosófica de la ciencia política. En este caso podemos ver reflejados algunos artículos en el estado del arte que hacen referencia a la definición de los términos que podría englobar una ontología política.



Figura 16: Principales áreas o categorías del programa de Ahora Madrid.

La segunda perspectiva pertenece a la política definida en la práctica común, aquella que aparece en los programas electorales o la relacionada con los servicios públicos. De la primera poca información tenemos al respecto, ya que el único trabajo a considerar para extraer la información de los programas electorales más allá del clásico PDF es el proyecto *DemoCritics* [12].

Por último, la tercera perspectiva quizá la más delicada a tratar, pues el bot deberá recuperar información de textos basadas en las dos perspectivas anteriores, pero debe comunicarse con los ciudadanos que tienen la última perspectiva. Es decir saber guiar-educar al ciudadano con las categorías o conceptos precisos, como indicó la entrevista con Javier de la Cueva [37] (ver sección 6.1). Es por ello la necesidad de incorporar diccionarios de sinónimos que hagan de intermediario entre el conocimiento del bot (adquirida de las dos perspectivas anteriores) y el del ciudadano medio.

Esta visión acerca de las diferentes perspectivas que guardaban relación con el desarrollo de la ontología, fue gracias a la reunión con expertos, quienes me ayudaron a identificar los principales problemas de clasificación de la información y la forma de enfocarla al ciudadano medio.

Estas dificultades en relación con la consulta de expertos sobre la materia y documentos específicos sobre la misma, podrían servir de utilidad a la aportación para las metodologías del desarrollo en ontologías de otros contextos aún por explorar.

6.3 Identificación de los términos

Para familiarizarnos con los principales términos que podrían componer una ontología política, se exploraron diferentes documentos especializados. En particular, los programas electorales de las últimas elecciones. Pues en ellos se precisa gran parte de los términos, relaciones y conceptos que deberían aparecer en una ontología política. Para ello se analizaron como los diferentes partidos políticos estructuran sus programas electorales por temas, categorías, secciones, etcétera. Las figuras 14 y 15 muestran algunas términos pertenecientes a las clases de la ontología desarrollada.

Explorando los programas electorales de los principales partidos políticos que concurrieron a las últimas elecciones generales, municipales y autonómicas se generó una lista de términos y relaciones que podrían aparecer en una ontología política. Algunas partidos como *Ahora Madrid* optaron por una forma diferente de organizar sus programas electorales (ver figura 16, fuente: <http://programa.ahoramadrid.org/>).

Cabe destacar que la ontología hace referencias a términos y relaciones del Gobierno de España. Pues cada gobierno dispone de sus propias instituciones en función de su

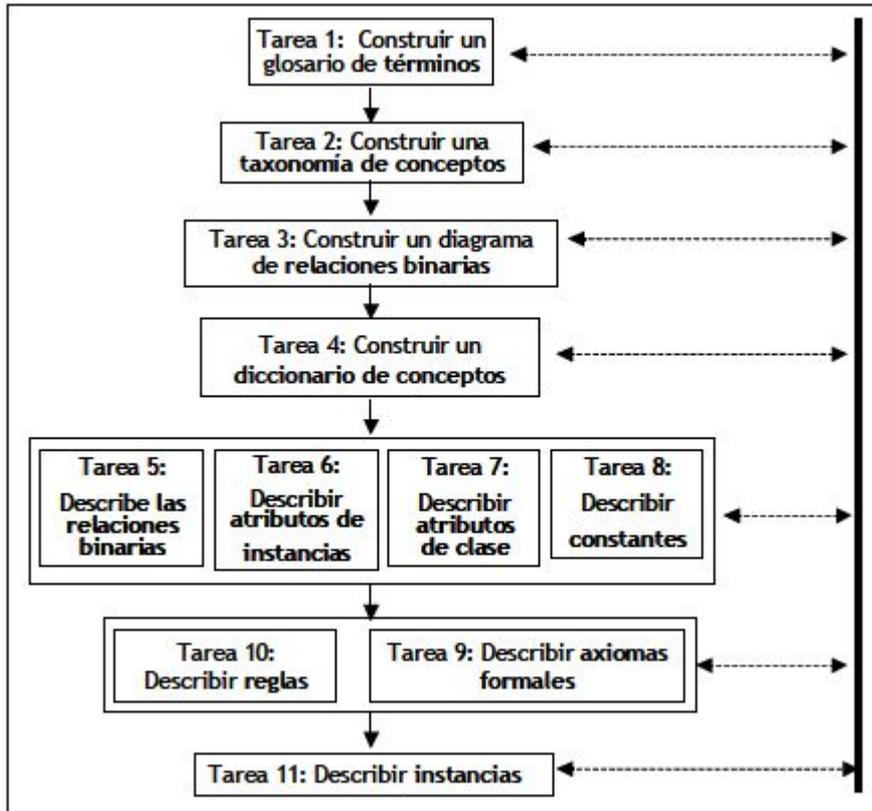


Figura 17: Tareas que componen la metodología Methontology [38].

organización, tipo de estado, cámara parlamentaria, etcétera. Por tanto la ontología puesta en práctica en el proyecto referenciaría al estado de España.

6.4 Metodologías para el desarrollo de ontologías

Para la metodología del desarrollo de la ontología política en primer lugar se exploraron las diferentes metodologías para la construcción de ontologías como *Methontology* [38], desarrollada por el Grupo de Ingeniería Ontológica de la Universidad Politécnica de Madrid. Esta metodología proporciona unas pautas de cómo llevar el proceso de desarrollo de una ontología a través de las actividades de especificación, formalización, implementación y mantenimiento. La figura 17 muestra las principales fases de la metodología. También se tuvieron en cuenta otros desarrollos relacionados como es el caso de *Un Modelo Ontológico para el Gobierno Electrónico* [10] de la Universidad de Misiones (Argentina), analizado en el capítulo 2 del presente documento.

Además de las referencias citadas anteriormente, se tuvo muy en cuenta el enfoque obtenido en las entrevistas por expertos (ver sección 6.1), que modificaron la percepción del enfoque particular en relación con la ciencia política. Por último, para la identificación de los términos, se estudiaron los programas electorales de diversas formaciones políticas. En estos documentos podíamos ver el enfoque y la relación de términos que utiliza el político o representante del partido, para dirigirse al ciudadano y mostrar su proyecto político.

6.5 Implementación en *protégé*

A continuación se procedió a implementar la ontología, basada en las orientaciones y captación de requisitos que había tenido en las entrevistas con expertos, la extracción de términos y relaciones de los programas electorales y la propuesta de una metodología que con la que desarrollar la ontología política.

Para ello se eligió OWL como lenguaje para definir la ontología política, por ser el estándar definido por la W3C y uno de los más utilizados en la web semántica. Como entorno de desarrollo se utilizó *protégé* (ver figura 14), un IDE desarrollado por la Universidad de Stanford que permite el desarrollo e implementación de ontologías en OWL.

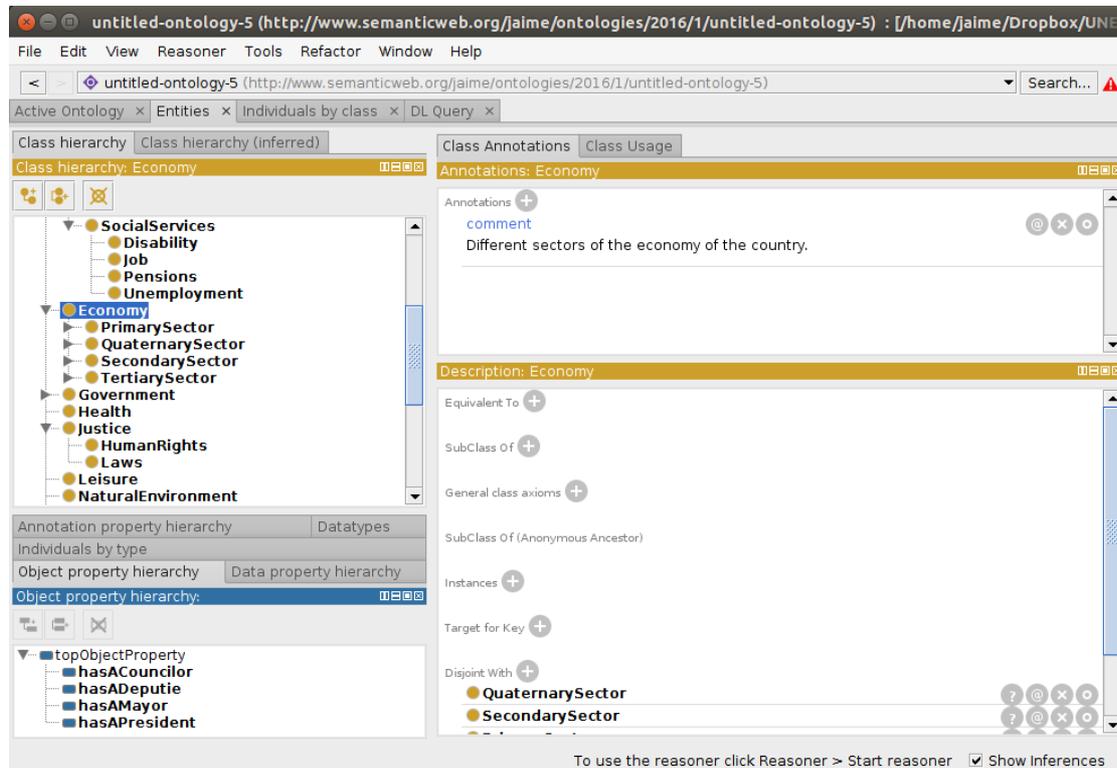


Figura 18: Ontología política implementada en *protégé*.

Capítulo 7 - Diseño de @DemoCriticsBot

En este capítulo detallaremos las características y los pasos seguidos para el desarrollo del bot, así como las principales funcionalidades e interacciones con los usuarios.

En primer lugar se necesitaba una plataforma donde poder desarrollar un bot que nos permitiera la interacción con los usuarios y definir alguna acción personalizada para que los usuarios pudieran solicitar tareas al bot. Se decidió la implementación del bot en la plataforma Telegram Bot [44] por dos motivos. El primero es el acceso que Telegram pone a su disposición con la Telegram Bot API [20], lo que nos da una gran libertad para llevar a cabo desarrollos de cualquier índole. La segunda, de cara más importante hacia el trabajo futuro, es asegurarnos de que la plataforma escogida garantizara la estabilidad de la plataforma con una comunidad de usuarios y seguridad en sus comunicaciones.

Hasta el momento Telegram es una de las aplicaciones de mensajería instantánea con una gran comunidad en constante crecimiento, registrando el pasado mes de Febrero de 2016 más de 100 millones de usuarios activos. Mientras que a la vez es una de las aplicaciones que se ha centrado en la seguridad de sus comunicaciones, respetando la privacidad de los usuarios desde el inicio.

El bot se encuentra publicado en la plataforma Telegram accesible con el identificador [@DemoCriticsBot](#). Actualmente existen dos formas de utilizarlo, bien en una conversación privada de uno a uno con el bot, o incluyendo el bot en el grupo donde queramos realizar el debate. Una vez incorporemos nuestro bot al grupo, por defecto se ejecutará el comando `/start`, el cual realizará una breve presentación de las competencias del bot.

7.0.1 Comandos personalizados

Como ya indicamos antes, en un bot conversacional necesitamos el uso de ciertos comandos para solicitar tareas, asistencia o realizar alguna configuración en las actividades de nuestro bot. Además desde el cliente de Telegram podemos visualizar los diferentes comandos que tenemos a nuestra disposición junto a una pequeña descripción de su funcionamiento. A continuación se describen los principales comandos que utilizará el bot:



Figura 19: Solicitando información a @DemoCriticsBot.

7.0.2 Comando /start

Es el comando de partida que pondrá en funcionamiento nuestro bot en cualquier conversación o grupo donde previamente se haya incorporado. El bot mostrará un pequeño mensaje de presentación para mostrar su funcionamiento, y sugerirá el uso del comando /help para que los usuarios puedan conocer los diferentes comandos y funcionalidades del bot.

7.0.3 Comando /help

Este comando cuenta con un funcionamiento muy simple. Cada vez que es solicitado por un usuario, el bot muestra la lista de comandos disponibles para interactuar con él, junto a una breve descripción de la utilidad de cada uno.

7.0.4 Comando /informacion

El comando información tiene el fin de poder solicitar información en cualquier momento del debate (bien esté relacionada o no con el tema a tratar). Se trata de dar la opción de que el usuario pueda solicitar algún dato sobre alguna ley, partido político, gráfico o estadísticas. Cuando un usuario ejecute este comando, dispondrá un menú desplegable para seleccionar la información que más le interese. Una vez seleccionado el tipo de información requerida, el bot sugerirá información relacionada con el debate que se esté tratando. Haciendo uso de la ontología política y base de conocimientos, proporcionado algún tipo de información básica en forma de enlaces, texto o gráficos que puedan ampliar los conocimientos de los integrantes del debate.

También dispone de una opción en la que el usuario puede realizar una búsqueda personalizada. En esta búsqueda el bot utilizando su diccionario de sinónimos buscará la información disponible en su base de conocimientos ofreciendo al usuario la información encontrada.

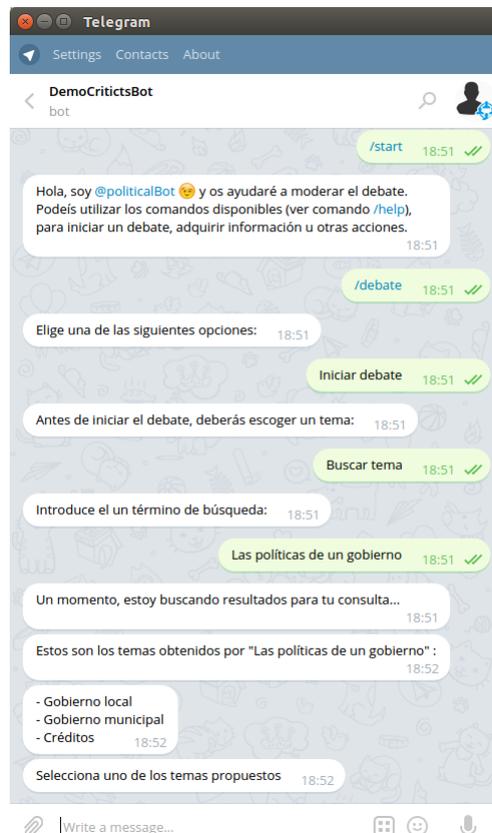


Figura 20: Captura de @DemoCriticsBot desde Telegram Desktop.

7.0.5 Comando /debate

Este será el comando principal de la aplicación para poner en marcha los debates. Al seleccionar este comando, el bot nos dará la opción de iniciar el debate, detenerlo o modificar otras características como el tiempo límite del debate o los tiempos de intervención. Cuando seleccionemos la opción de iniciar el debate, el bot nos dará la opción de escoger un tema en el que centrar el debate. Para seleccionar el tema dispondremos de las siguientes opciones:

- Seleccionar tema: el bot nos ofrece una lista de temas principales que extraiga de la ontología política desarrollada. El usuario tendrá la opción de seleccionar uno de los temas de la lista. Una vez seleccionado el tema, el bot utilizará las relaciones de la ontología para sugerir una lista de temas más específicos y centrar el tema del debate en caso de que así lo desee el usuario.
- Buscar tema: el bot ofrece al usuario que escriba un tema que pueda definir el tema principal del debate. Tras esto, el bot analizará la frase escrita por el usuario utilizando su diccionario de sinónimos para relacionar los términos con los objetos y relaciones que componen la ontología política. De esta forma el bot ofrecerá al usuario los temas que podrían estar relacionados con su búsqueda.
- Sugerir tema: en esta opción el bot seleccionará un tema al azar que pertenezca a la ontología política. En un futuro se podrían utilizar los temas más populares o más debatidos como opción de debate.

7.0.6 Comando /about

Este comando muestra un pequeño mensaje acerca del desarrollo del bot.

7.0.7 Otros comandos por implementar

En este primer prototipo hemos visto como las principales características de @DemoCriticsBot giran en torno a los comandos de /informacion y /debate. Más adelante se podrían añadir algunos otros comandos que pudieran darnos información y estadísticas de los debates moderados por el bot. Por ejemplo, sería interesante un pequeño porcentaje de las intervenciones de cada usuario, de esta forma podríamos conocer los usuarios más activos en el debate. Así como ofrecer una lista de los temas más debatidos,

populares o tendencias en el momento. Estos datos podrían enriquecer las habilidades del bot moderador para interactuar con los usuarios.

En cuanto a tareas de moderación, por el momento podemos configurar el tiempo limitado que puede durar el debate. Sería interesante implementar turnos de palabra por turnos o tiempos limitados. De esta forma él bot tendría más control sobre las intervenciones del debate.

7.1 Integración de @DemoCriticsBot con la base de datos de DemoCritics

Para enriquecer el conocimiento de cara a debates de carácter político y electoral, sería interesante que el bot contara con una base de conocimientos sobre el contenido de los programas electorales de los partidos políticos que se presentan a las elecciones. Hasta ahora en unas elecciones todos los partidos políticos suben su programa electoral en formato PDF, de forma que dificulta la extracción automática de la información.

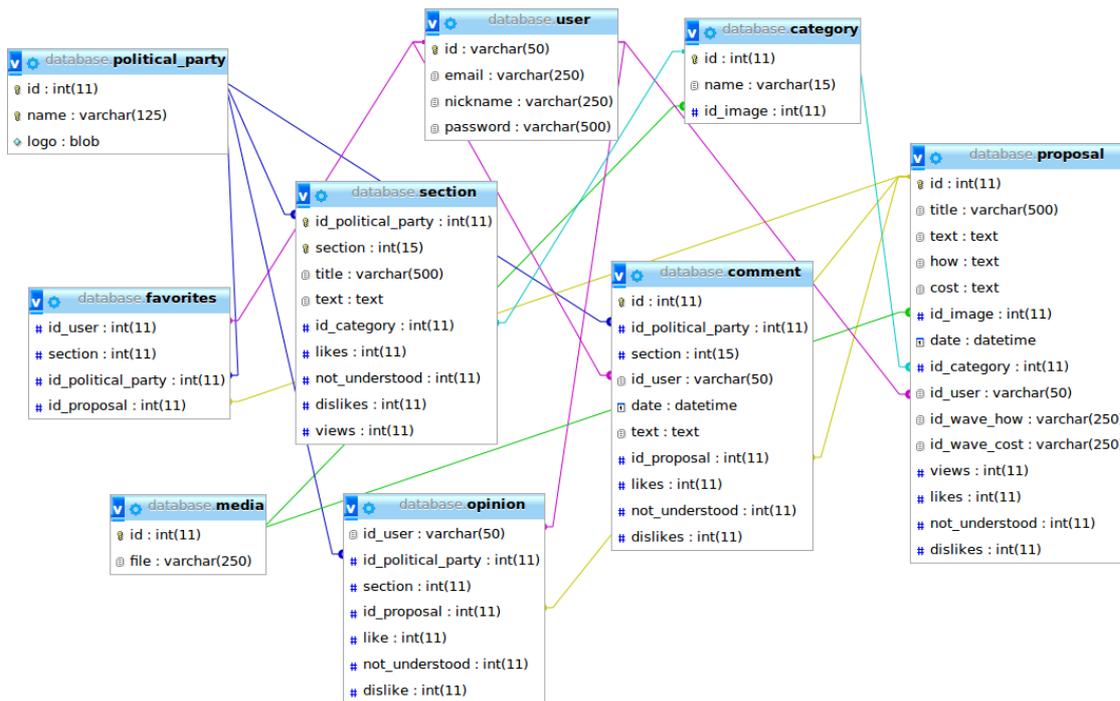


Figura 21: Estructura de la base de datos de DemoCritics [12].

Para ello el proyecto DemoCritics [12] desarrolló un sistema para clasificar los programas electorales de los diferentes partidos políticos en forma de varias secciones organizadas en una base de datos. De esta forma los programas electorales de cada partido quedarían clasificados en tablas para poder acceder directamente al contenido de los

programas. La figura 21 muestra la estructura de las tablas de la base de datos que almacena la información de los programas electorales.

Para que el bot desarrollado pudiera incorporar este conocimiento, se añadieron algunas rutas más al controlador de la API Rest que daba acceso a la base de datos del proyecto *DemoCritics* (ver sección 5.3.4). De esta forma @DemoCriticsBot puede acceder al conocimiento estructurado de los programas electorales, para así poder explorarlos y relacionarlos con la ontología política. Concretamente se han añadido nuevas rutas que dan acceso al contenido de los programas electorales mediante las consultas que realiza el bot para encontrar información o términos relacionados en los programas. De esta forma será mucho más fácil acceder a las diferentes secciones que tratan sobre un tema concreto e identificar al partido político que pertenecen.

Cuando el usuario ha elegido el tema que define el debate, el bot realizará búsquedas relacionada con la ontología política y los sinónimos que obtenga en el diccionario para encontrar aquellas secciones de los programas que puedan ser de utilidad para ofrecérselas a los usuarios. Además también cabe la posibilidad de que los usuarios realicen búsquedas personalizadas en las que el bot indagará en el contenido de los programas.

El acceso a la base de datos de DemoCritics junto a la ontología política desarrollada, enriquecen el funcionamiento del bot, partiendo de la ontología como base para organizar y clasificar el contenido de los programas electorales.

Además de las tablas definidas en la aplicación, se añade una tabla para almacenar los enlaces sobre partidos políticos y otras categorías de la ontología que contengan información relevante de actualidad para dar información al debate.

Como trabajo futuro cabe la posibilidad de reutilizar la base de conocimientos de *DemoCritics* para almacenar los usuarios registrados en la plataforma y guardar estadísticas de uso que puedan aportar nuevas funcionalidades al debate.

Capítulo 8 - Resultados y trabajo futuro

Como resultado del trabajo final en el desarrollo del presente Trabajo de Fin de Máster, se pone a disposición la propuesta de una arquitectura de bot conversacional basado en el uso de ontologías para la extracción de la información de textos y consulta de información compleja.

Este bot es capaz de interactuar con los usuarios y moderar debates de carácter político, ofreciendo apoyo a los usuarios utilizando su base de conocimientos basada en la ontología desarrollada. Además utiliza técnicas de inteligencia artificial para relacionar los mensajes de los usuarios con su base de conocimientos y la ontología política desarrollada.

Como trabajo futuro para la plataforma y arquitectura genérica propuesta para el bot-moderador de debates basado en ontologías, podemos destacar los siguientes puntos:

- Estudiar la posibilidad de utilizar una plataforma para el desarrollo de chatbot como *Pandorabots* o *ChatScript*.
- Validación de la arquitectura y plataforma de desarrollo desarrollando un bot asesor de debates en otro dominio.
- Incorporación en la arquitectura de un componente basado en técnicas de aprendizaje automático para la modificación dinámica de la ontología en base a la experiencia del bot en la moderación de debates.
- Enriquecimiento de las posibilidades de acceso a documentación diversa de interés incorporando asimismo diversas técnicas de recuperación de información en textos.

En cuanto a la aplicación específica desarrollada para el bot moderador de debates basado en una ontología política, destacamos las siguientes líneas de trabajo:

- Extensión de la ontología política actual (ontología ligera consistente en un conjunto de conceptos con sus propiedades y relaciones) para convertirla en una ontología "pesada", añadiendo axiomas y restricciones que permitan realizar chequeos de corrección e inferir nueva información.
- Incorporar técnicas de aprendizaje automático para que el bot pueda aprender a reconocer la terminología que utiliza el ciudadano medio.

- Modificar o enriquecer la ontología en función de los temas que se vayan proponiendo o sean los más populares a medida que se va utilizando el bot.
- Añadir nuevos comandos al bot que permitan realizar algunas consultas sobre las estadísticas, intervenciones o temas más debatidos.
- Automatizar la adquisición del conocimiento sobre programas electorales en tiempo real, accediendo a las páginas web de los partidos.
- La arquitectura y componentes desarrollados constituyen una plataforma de desarrollo de bots de amplia aplicación, en particular en contextos donde el debate pueda ser un recurso valioso, como el aprendizaje informal en e-learning [45][46].
- Experimentación de la aplicación en las próximas elecciones (españolas o de otro país) con el fin de validarla y analizar sus debilidades y fortalezas con vistas a una extensión futura.

Capítulo 9 - Conclusiones

Llevar a cabo este proyecto ha supuesto realizar una tarea de investigación en diferentes áreas que guardan poca relación en los trabajos publicados hasta la fecha. Destacando el hecho de poner en valor una aplicación de participación política y ciudadana que incorpore técnicas de inteligencia artificial por primera vez. Pues hasta la fecha ninguna aplicación explotaba estas técnicas para enriquecer su uso.

En el ámbito de desarrollo de bots inteligentes, se realiza una propuesta de arquitectura para el desarrollo de bots basados en ontologías. Esta arquitectura posee la implementación y uso de ciertos componentes para el uso de un bot-moderador-consultor que pueda dar soporte en debates y consultar información basada en una ontología y otras herramientas.

Aproximándonos al área de las ontologías y web semántica, se ha desarrollado una ontología de carácter político, cuyas etiquetas podrían servir para anotar el contenido de los diferentes programas políticos para su uso en diversas aplicaciones en el contexto de la web semántica. Añadiendo aportaciones al campo de las metodologías de desarrollo, haciendo énfasis al análisis y las peculiaridades del dominio.

Además, cabe la posibilidad de reutilizar el bot moderador en el ámbito de las tecnologías educativas inteligentes (e-learning). Este bot puede tener un alto valor de aprendizaje si se integra en aplicaciones educativas para diferentes disciplinas tanto científico-técnicas como humanísticas, así como en el ámbito del aprendizaje no-formal, facilitando el desarrollo de competencias de alto nivel.

Haciendo referencia a los objetivos y motivaciones iniciales, podemos decir que se han satisfecho en la medida de su planteamiento inicial y limitaciones del dominio. Por tanto podemos resumir los logros o aportaciones del proyecto en los siguientes puntos:

- Puesta en valor de una aplicación basada en participación política y ciudadana que integra técnicas de inteligencia artificial para la extracción de la información de textos basada en ontologías.
- Garantizar la seguridad y privacidad de las comunicaciones en lo que respecta al envío y recepción de mensajes o información sensible utilizando tecnologías basadas en técnicas de encriptación y autenticación.

- Aportaciones en el campo de las metodologías de desarrollo de ontologías, con énfasis en el análisis multiperspectiva en aquellos dominios en que existan diferentes aproximaciones a un marco conceptual, y en la entrevista con expertos a distintos niveles.
- Desarrollo de una ontología política para representar conceptos y relaciones, reflejando el sistema desde una perspectiva con un conocimiento del dominio medio.
- Puesta en práctica de una perspectiva diferente en torno a los chat-bot o bot conversacionales. Incrementando su protagonismo y utilidad implementando técnicas para la extracción de información y asesoramiento que modelan parte de su conocimiento.

Bibliografía

- [1] Dimitris, A., Ernst, P., Knut, Y. H., Nordwestschweiz, F. S., Fabian, P., & Archetypon, N. N. Ontology-enabled E-Government Service Configuration-the OntoGov Approach.
- [2] Hodgson R., Allemang D., (2006), Semantic Technology For e-Government, Semantic Web and Beyond Volume 3, 2006, pp 283-303, Springer US, ISBN 978-0-387-30239-3.
- [3] Klischewski, R., & Jeenicke, M. (2004, January). Semantic web technologies for information management within e-government services. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on* (pp. 10-pp). IEEE.
- [4] Hreño, J., Bednár, P., Furdík, K., & Sabol, T. (2011). Integration of government services using semantic technologies. *Journal of theoretical and applied electronic commerce research*, 6(1), 143-154.
- [5] Martínez, P., González-Cristobal, J. C., & Sandoval, A. M. (2011). MULTIMEDICA: Extracción de información multilingüe en Sanidad y su aplicación a documentación divulgativa y científica. *Procesamiento del lenguaje natural*, 47, 347-348.
- [6] Cotelo, J. M., Cruz, F., Ortega, F. J., & Troyano, J. A. (2015). Explorando Twitter mediante la integración de información estructurada y no estructurada. *Procesamiento del Lenguaje Natural*, 55, 75-82.
- [7] Maña, M., Mata, J., Domínguez, J. L., Vaquero, A., Alvarez, F., Gómez, J. M., ... & de Buenaga, M. (2006, September). Los proyectos SINAMED e ISIS: Mejoras en el Acceso a la Información Biomédica mediante la integración de Generación de Resúmenes, Categorización Automática de Textos y Ontologías. In *En Actas del XXII Congreso de la Sociedad Española de Procesamiento del Lenguaje (SEPLN)*.
- [8] Ortega, J. M. P., Valdivia, M. T. M., Ráez, A. M., & Galiano, M. C. D. (2008). Categorización de textos biomédicos usando UMLS. *Impresos de Inscripción*, 40, 121-127.
- [9] Ruiz Martínez, J. M. (2012). Metodología para la población automática de ontologías: aplicación en los dominios de medicina y turismo.
- [10] Brys, C. R., Aldana-Montes, J. F., & Martinez, D. L. L. R. Un Modelo Ontológico para el Gobierno Electrónico.
- [11] Portal de Datos Abiertos del Gobierno de Misiones
<http://www.datos.misiones.gov.ar/>
- [12] Ramos Romero, J., & Bastarrica Lacalle, J. (2015). DemoCritics: aplicación Android de participación política con edición colaborativa en tiempo real.

- [13] Poletika.org
<http://poletika.org/>
- [14] S.A. APPGREE. Appgree.
<http://www.appgree.com/appgree/que-es-appgree/>
- [15] Francés García, F. J. (2015). El salto virtual del debate colectivo: potencialidades y limitaciones de las nuevas plataformas para la deliberación y la toma de decisiones ciudadanas.
- [16] Clavell, G. G. (2015). Podemos y la política de la tecnología. *Teknokultura*, 12(1), 111-119.
- [17] Pizarro, M. A., & Labuske, E. (2015). El músculo deliberativo del algoritmo democrático: Podemos y la participación ciudadana. *Teknokultura*, 12(1), 93-109.
- [18] Politibot
<https://politibot.es/acerca/>
- [19] Al-Zubaidi, H., & Issa, A. A. (2011, November). Ontbot: Ontology based chatbot. In *Innovation in Information & Communication Technology (ISIICT), 2011 Fourth International Symposium on* (pp. 7-12). IEEE.
- [20] Telegram Bot API
<https://core.telegram.org/bots/api>
- [21] Augello, A., Pilato, G., Vassallo, G., & Gaglio, S. (2014). Chatbots as interface to ontologies. In *Advances onto the Internet of Things* (pp. 285-299). Springer International Publishing.
- [22] Smith, B. (2007). A quick guide to gplv3. *Free Software Foundation, Inc. Online*:
<http://www.gnu.org/licenses/quick-guide-gplv3.html>. Referred, 4, 2008
- [23] Bechhofer, S. (2009). OWL: Web ontology language. In *Encyclopedia of Database Systems* (pp. 2008-2009). Springer US.
- [24] *protégé* - A free, open-source ontology editor and framework for building intelligent systems
<http://protege.stanford.edu/>
- [25] Telegram Messenger LLP
<https://telegram.org/>
- [26] Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.

- [27] Bird, S. (2006, July). NLTK: the natural language toolkit. In *Proceedings of the CO-LING/ACL on Interactive presentation sessions* (pp. 69-72). Association for Computational Linguistics.
- [28] Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. "O'Reilly Media, Inc."
- [29] Cloud9 - Your development environment, in the cloud
<https://c9.io/>
- [30] Laravel - The PHP Framework For Web Artisans
<https://laravel.com/>
- [31] Inc. Red Hat. OpenShift by Red Hat.
<https://www.openshift.com/>
- [32] Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012, February). Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work* (pp. 1277-1286). ACM.
- [33] Saint-Andre, P. (2011). Extensible messaging and presence protocol (XMPP): Core.
- [34] Jakobsen, J., & Orlandi, C. (2015). *A practical cryptanalysis of the Telegram messaging protocol* (Doctoral dissertation, Master Thesis, Aarhus University (Available on request)).
- [35] NLTK - WordNet Interface
<http://www.nltk.org/howto/wordnet.html>
- [36] LaboDemo | Democratic Laboratory @labo_demo
<http://labodemo.net/>
- [37] Javier de la Cueva, abogado y doctor en Filosofía
<http://javierdelacueva.es/personas/javier-de-la-cueva/>
- [38] Fernández-López, M., & Gómez-Pérez, A. METHONTOLOGY. *Laboratorio de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo sn. Boadilla del Monte, 28660.*
- [39] ontospy 1.6.8.5 : Python Package Index
<https://pypi.python.org/pypi/ontospy>
- [40] WebVOWL - Web-based Visualization of Ontologies
<http://vowl.visualdataweb.org/webvowl.html>
- [41] Moobot, your Twitch chat moderator bot
<http://twitch.moobot.tv/>

- [42] Nightbot
<https://beta.nightbot.tv/>
- [43] Twitch
<https://www.twitch.tv/>
- [44] Telegram Bot Platform
<https://telegram.org/blog/bot-revolution>
- [45] Park, Y., Heo, G. M., & Lee, R. (2011). Blogging for Informal Learning: Analyzing Bloggers' Perceptions Using Learning Perspective. *Educational Technology & Society*, 14(2), 149-160.
- [46] García-Peñalvo, F. J., Johnson, M., Alves, G. R., Minović, M., & Conde-González, M. Á. (2014). Informal learning recognition through a cloud ecosystem. *Future Generation Computer Systems*, 32, 282-294.
- [47] Hay, C. (2006). Political ontology. *The oxford handbook of contextual political analysis*, 78-96.
- [48] Pettit, P. (2005). Rawls's political ontology. *politics, philosophy & economics*, 4(2), 157-174.
- [49] Kauppi, N. (2010). The political ontology of European integration. *Comparative European Politics*, 8(1), 19-36.
- [50] Marres, N. (2013). Why political ontology must be experimentalized: On ecoshowhomes as devices of participation. *Social studies of Science*, 0306312712475255.
- [51] Barón, M. J. S., Valencia, K. E. S., García, C. A. C., & Barón, J. C. S. (2014). Applying Ontologies for Web Text Mining Using Heterogeneous Information. *International Journal of Information Processing and Management*, 5(4), 18.
- [52] Wong, W., Liu, W., & Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, 44(4), 20.
- [53] Serra, I., Girardi, R., & Novais, P. (2014). Evaluating techniques for learning non-taxonomic relationships of ontologies from text. *Expert systems with applications*, 41(11), 5201-5211.
- [54] MTPProto Mobile Protocol
<https://core.telegram.org/mtproto>
- [55] Explanation of secret chats (Telegram).png - Diego Sanguinetti
[https://commons.wikimedia.org/wiki/File:Explanation_of_secret_chats_\(Telegram\).png](https://commons.wikimedia.org/wiki/File:Explanation_of_secret_chats_(Telegram).png)
- [56] Architecture of Laravel Applications - Laravel Book
<http://laravelbook.com/laravel-architecture/>