



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo del estándar abierto Open Manifesto para los programas electorales

TRABAJO FIN DE MÁSTER

Máster en Gestión de la Información

Autor: Sebastián Pallarés Saltorio

Tutor: Diego Álvarez Sánchez

Curso 2018-2019

Resumen

Las elecciones libres son un proceso mediante el cual los ciudadanos ejercen su derecho a voto. Así se constituyen democráticamente las instituciones, mediante la elección de representantes que serán los encargados de tomar las decisiones políticas. El principal mecanismo por el que se transmite la información necesaria para que los ciudadanos puedan elegir a qué partido se dirigirá su voto son los programas electorales.

Sin embargo, las características de estos documentos dificulta el acceso, comprensión, interpretación y comparación entre las diferentes propuestas políticas por parte de la ciudadanía. Por ello, en este trabajo se ha desarrollado un estándar abierto para la elaboración y publicación de programas electorales, centrado en la ciudadanía, que estructura la información contenida en estos documentos y facilita la consulta automatizada de sus contenidos. De esta forma, favorece la interpretación de una información básica para el electorado.

Además, se ha implementado un asistente virtual, llamado Analista Político Personal, que permite consultar y llevar a cabo un seguimiento de las propuestas electorales de los partidos políticos en las distintas elecciones. Este asistente accede a la información contenida en los programas electorales definidos bajo el estándar *Open Manifesto*. Se ha desarrollado haciendo uso de tecnologías y herramientas de procesamiento del lenguaje natural.

Palabras clave: asistente virtual, programas electorales, estándar abierto, ciudadanía, Analista Político Personal

Resum

Les eleccions lliures són un procés mitjançant el qual els ciutadans exerceixen el seu dret a vot. Així es constitueixen democràticament les institucions, mitjançant l'elecció de representants que seran els encarregats de prendre les decisions polítiques. El principal mecanisme pel qual es transmet la informació necessària perquè els ciutadans puguin triar a quin partit es dirigirà el seu vot són els programes electorals.

No obstant això, les característiques d'aquests documents dificulta l'accés, comprensió, interpretació i comparació entre les diferents propostes polítiques per part de la ciutadania. Per això, en aquest treball s'ha desenvolupat un estàndard obert per a l'elaboració i publicació de programes electorals, centrat en la ciutadania, que estructura la informació continguda en aquests documents i facilita la consulta automatitzada dels seus continguts. D'aquesta forma, afavoreix la interpretació d'una informació bàsica per a l'electorat.

A més, s'ha implementat un assistent virtual, anomenat Analista Polític Personal, que permet consultar i dur a terme un seguiment de les propostes electorals dels partits polítics en les diferents eleccions. Aquest assistent accedeix a la informació continguda en els programes electorals definits sota l'estàndard *Open Manifesto*. S'ha desenvolupat fent ús de tecnologies i eines de processament del llenguatge natural.

Paraules clau: assistent virtual, programes electorals, estàndard obert, ciutadania, Analista Polític Personal

Abstract

Free elections are a process by which citizens exercise their right to vote. This is how institutions are democratically constituted, through the election of representatives who will be the political decision-makers. The main mechanism whereby the necessary information is transmitted for citizens to be able to make a choice which party your vote will go to are the election programs.

However, the characteristics of these documents make them difficult to access, understand, interpret and comparison between the different political proposals by the citizens. Therefore, in this work an open standard has been developed for the elaboration and publication of electoral programs, focused on citizenship, which structures the information contained in these documents and facilitates the automated consultation of their contents. In this way, this favors the interpretation of basic information for the electorate.

In addition, a virtual assistant has been implemented, called Personal Political Analyst, which allows to consult and carry out a follow up of the political parties electoral proposals in the different elections. This assistant accesses to the information contained in the electoral programs defined under the *Open Manifesto* standard. It has been developed using natural language processing technologies and tools.

Keywords: virtual assistant, electoral programs, open standard, citizenship, Personal Political Analyst

Índice general

Índice general	V
Índice de figuras	VII

1 Introducción	1
1.1 Contexto	1
1.2 Motivación	2
1.3 Objetivos	4
1.4 Metodología	4
1.5 Estructura de la memoria	5
2 Estado del arte	7
2.1 Estándares abiertos	7
2.1.1 Open Contracting Partnership	8
2.1.2 OpenStand	9
2.1.3 Open Source Definition	9
2.1.4 ITU-T	10
2.1.5 Open Data Institute	10
2.2 Asistentes virtuales	11
2.2.1 Definiciones	13
2.2.2 Asistente de Google	13
2.2.3 Siri	14
2.2.4 Alexa	14
2.2.5 Cortana	15
3 Diseño del estándar Open Manifesto	17
3.1 Metodología	17
3.2 Estructura de los programas electorales	18
3.3 Análisis y especificación	19
3.3.1 Jornadas de trabajo	19
3.3.2 Herramientas y tecnologías	20
3.4 Desarrollo	22
3.4.1 Estándar Open Manifesto	22
3.4.2 Proceso de estandarización	24
4 Implementación del asistente Analista Político Personal	27
4.1 Metodología	27
4.2 Herramientas y tecnologías utilizadas	28
4.2.1 Dialogflow	28
4.2.2 Actions on Google	30
4.2.3 Firebase	30
4.2.4 Google Cloud	31
4.2.5 Node.js	31
4.3 Análisis y especificación	32
4.3.1 Casos de uso	32
4.3.2 Sistema informacional	33

4.4	Desarrollo	34
4.4.1	Arquitectura	34
4.4.2	Implementación	35
4.4.3	Evaluación y seguimiento	41
5	Análisis de resultados	43
5.1	Resultados principales	43
5.2	Campaña social	44
6	Conclusiones	47
6.1	Posibles ampliaciones	48
6.2	Apariciones en prensa y reconocimientos	48

Apéndices

A	Estándar Open Manifesto	51
B	Talleres	55
B.1	Taller en Madrid	55
B.2	Taller en Valencia	56
B.3	Taller en Barcelona	56

Índice de figuras

1.1	Tendencia de la preocupación por la política	2
1.2	Metodología de desarrollo del proyecto	5
2.1	Etapas para el desarrollo de un estándar abierto	11
2.2	Asistentes más utilizados por los españoles	12
2.3	Altavoces inteligentes más usados por los españoles	13
3.1	Repositorio del estándar <i>Open Manifesto</i> en Github	21
3.2	Ejemplo de JSON Schema	21
3.3	Definición del estándar <i>Open Manifesto</i>	24
3.4	Programa electoral del PSOE estandarizado	25
4.1	Diagrama del funcionamiento de Dialogflow	29
4.2	Interfaz web de Actions on Google	30
4.3	Ejemplo de implementación en Firebase CLI	31
4.4	Arquitectura de la librería Actions on Google de Node.js	32
4.5	Atributos de un programa electoral en el directorio	34
4.6	Componentes de <i>Analista Político Personal</i>	34
4.7	<i>Entities</i> del agente Dialogflow	36
4.8	Ejemplo de la entidad <i>Región</i>	36
4.9	<i>Intents</i> del asistente virtual	38
4.10	Frases de entrenamiento del <i>IntentEnunciativaGeneralesEuropeas</i>	38
4.11	Parámetros de las frases de entrenamiento	39
4.12	Diagrama conversacional del asistente virtual	40
4.13	Entrenamiento del asistente virtual	42
4.14	Estadísticas de uso de <i>Analista Político Personal</i>	42
5.1	<i>Analista Político Personal</i> en el directorio de Actions	44
5.2	Resultados de la campaña por Twitter	44
5.3	Resultados de la campaña por Instagram	45
B.1	Fotografía del taller de Madrid	55
B.2	Fotografía del taller de Valencia	56
B.3	Fotografía del taller de Barcelona	57

CAPÍTULO 1

Introducción

En este primer capítulo se pretende exponer el contexto en que desarrolla este trabajo final del máster en gestión de la información. Posteriormente, se describen las principales razones que han motivado la realización del proyecto y también se enumeran los objetivos, tanto principales como específicos, que se desean cumplir al finalizar el trabajo. Por último, se presenta la metodología seguida y la estructura de la memoria.

1.1 Contexto

Las elecciones son la base de las democracias representativas. En los sistemas democráticos actuales, los ciudadanos eligen a sus representantes ejerciendo el derecho a voto y, en consecuencia, ejerciendo también el derecho a la participación política. De esta forma se constituyen democráticamente las instituciones mediante la elección de representantes que serán los encargados de tomar las decisiones políticas. Estos procesos electorales son el punto en el que los ciudadanos y los partidos establecen un diálogo social para abordar cuestiones de interés común.

En este contexto electoral, los partidos políticos intentan hacer llegar su mensaje al mayor número posible de votantes, por distintos canales y formatos, para conseguir el voto del electorado. Se podría decir que el mensaje electoral de un partido se transmite típicamente por la propia ideología del partido, por su candidato y por el programa electoral, siendo este último un elemento clave de ese mensaje.

Mediante los programas electorales, los partidos reflejan cómo interpretan e identifican los problemas actuales y qué proponen para solucionarlos. Por tanto, son el mecanismo por el que se transmite la información necesaria para que los ciudadanos puedan elegir a qué partido se dirigirá su voto.

Por otro lado, durante estos últimos años los procesos electorales han sido un tema de actualidad. Concretamente en España han habido tres elecciones generales en los últimos cuatro años; en 2015, en 2016 y en 2019. Además, los ciudadanos han sido llamados a las urnas para las elecciones municipales, autonómicas y europeas de 2019. Todo esto ha generado que para la ciudadanía española los políticos, los partidos y la política en general, sea el segundo tema que más preocupa, después del paro, según los datos del CIS en Julio de 2019 [1][2], consiguiendo así el máximo porcentaje de preocupación en este tema desde que se realiza la encuesta. En la Figura 1.1 se puede observar la evolución que ha tenido desde 1985 la preocupación de los ciudadanos encuestados sobre la política.



Figura 1.1: Tendencia de la preocupación por la política. Fuente: EpData y CIS

Se puede comprobar que, no solo se trata de un tema de actualidad, sino que genera preocupación en los ciudadanos todo lo relativo a la política actual. Por tanto, mejorar la forma en que se genera o transmite el mensaje electoral, parece necesario.

Como se ha mencionado anteriormente, los programas electorales son la clave de dicho mensaje para comprender la información que quieren transmitir los partidos. Pero las características que tienen estos documentos actualmente dificulta el acceso, comprensión e interpretación por parte de la ciudadanía. Suelen ser documentos en un único formato (típicamente PDF), de un gran volumen de páginas, y que no tienen una estructura común, entre otras dificultades para el lector.

Por tanto, se puede decir que, proveer la información contenida en los programas electorales en un nuevo formato estandarizado, facilitaría a los ciudadanos comprender la posición de los partidos, identificar los temas abordados, analizar su contenido o simplificar la comparación con los programas de otras formaciones. En definitiva, la estandarización sería un mecanismo para facilitar la comprensión, interpretación y accesibilidad a una información básica para el electorado.

1.2 Motivación

Se podría decir que los pilares en que se basa la calidad democrática y el gobierno abierto son la transparencia, la participación y la rendición de cuentas o *accountability*. En un trabajo publicado por el Banco Mundial llamado *Worldwide Governance Indicators* [3], se enumeran unos indicadores para medir la calidad de las instituciones que facilitan el buen gobierno. Para ello, seleccionaron tres áreas que definen la gobernanza, y crearon dos indicadores de calidad por cada una de las áreas, en suma un total de seis indicadores.

En esta sección se hará referencia a los indicadores relativos al proceso por el cual los gobiernos son seleccionados, monitoreados y reemplazados. Y estos indicadores no

son otros que: "La participación y la rendición de cuentas: capturar las percepciones de la medida en que los ciudadanos de un país son capaces de participar en la selección de su gobierno, así como la libertad de expresión, libertad de asociación, y libertad de prensa."[3]

Así mismo, se puede ver que la rendición cuentas es un elemento esencial en los sistemas democráticos eficaces y "uno de los principios básicos del buen gobierno"[4]. Como indica Andreas Schedler [5], para alcanzar la rendición de cuentas es necesario que políticos y funcionarios informen sobre sus decisiones y las justifiquen en público (*answerability*), y además, puedan crearse agencias capaces de sancionar a políticos y funcionarios en los casos en que violen sus deberes públicos (*enforcement*).

Esta noción del autor[6] sobre la rendición de cuentas se divide en tres aspectos: información, justificación y castigo. Así pues, el concepto comprende tres dimensiones:

- Dimensión informativa: Derecho a recibir información y obligación de darla. Se pregunta sobre los hechos o decisiones.
- Dimensión argumentativa: Derecho a recibir explicaciones y el deber de justificar el ejercicio del poder. Se pregunta sobre las razones que han llevado a tomar una decisión.
- Dimensión punitiva: Se reconoce lo correcto y se castiga lo erróneo mediante la penalización o castigo.

En este trabajo se empleará una ampliación de estas dimensiones, definidas por Andreas Schedler, para que se incluya junto a los hechos, razones y sanciones, las propuestas realizadas por los partidos políticos. Este nuevo concepto, definido por Diego Álvarez en [7], se denomina *dimensión propositiva* y se define como sigue: derecho a percibir políticas claras, comprensibles y detalladas.

Por tanto, sumado a la importancia del contexto expuesto en la sección anterior, los procesos de rendición de cuentas de los partidos políticos se consideran una herramienta fundamental para avanzar en la justicia social y política. Hasta hoy, la rendición de cuentas de los partidos políticos se limita a sus actividades económico-financieras, llevada a cabo por el Tribunal de Cuentas[8]. Pero se cree necesario que los programas electorales y las propuestas, no sean un mero documento difícil de acceder, sino que permitan valorar la coherencia de lo propuesto con las futuras políticas adoptadas y, así también, facilitar un seguimiento tras los procesos electorales.

En definitiva, la estandarización de los programas electorales y su uso como mecanismo para la rendición de cuentas desde la dimensión propositiva, son las principales razones que han motivado el desarrollo de este trabajo.

1.3 Objetivos

El objetivo principal de este trabajo es desarrollar un estándar abierto para los programas electorales centrado en la ciudadanía e implementar un asistente virtual que facilite el acceso a la información. El estándar permitirá crear un esquema común para los programas electorales que facilitará la identificación de los temas abordados por los partidos

políticos, al igual que también simplificará el análisis de su contenido y la comparación con otros programas, puesto que será un formato legible por máquina.

Por su parte, el asistente virtual facilitará el acceso a la información contenida en los programas electorales siguiendo el estándar definido.

Los objetivos específicos de este trabajo son los siguientes:

- Analizar la estructura y las características de los programas electorales.
- Especificar los requisitos del estándar siguiendo un diseño centrado en la ciudadanía.
- Evaluar las distintas opciones para estructurar y publicar el estándar.
- Establecer las bases para la explotación futura del estándar por parte de la ciudadanía, facilitando su accesibilidad con un formato abierto y público.
- Analizar las distintas herramientas de reconocimiento de lenguaje natural y seleccionar la idónea para la implementación del asistente virtual.
- Análisis de los dispositivos de acceso al asistente virtual.
- Desarrollar un prototipo funcional de asistente virtual que sea modular y facilite el testeo.

1.4 Metodología

En primer lugar, destacar que este trabajo final de máster se ha desarrollado en el marco de la realización de unas prácticas en la Cátedra de Tecnologías Cívicas y Empoderamiento (TCE). Esta cátedra es promovida por la Universitat Politècnica de València junto a la Delegació de Transparència i Govern Obert de la Diputació de València.

Además, este proyecto denominado *Open Manifesto Project*[9] se ha realizado en colaboración con CIECODE¹. Se trata del centro de investigación responsable de la plataforma *Political Watch* que cuenta con un conjunto de proyectos pioneros en materia de acceso a información, vigilancia ciudadana y rendición de cuentas.

En referencia a la metodología seguida, se divide en tres fases. Una primera fase llamada *Investigación*, en la cual se realiza una búsqueda y revisión de la literatura para establecer el marco conceptual del trabajo. Para ello, en primer lugar se investiga sobre los estándares abiertos y se analiza la estructura de los programas electorales. En segundo lugar, se valoran las distintas tecnologías para el desarrollo de asistentes virtuales enfocados al acceso a la información.

En la segunda fase, llamada *Análisis y especificación*, entra en juego el diseño centrado en la ciudadanía, puesto que para el desarrollo del estándar se realizan talleres donde se definen las necesidades de información de los ciudadanos con respecto a los programas electorales. Además, se evalúan las distintas propuestas para estructurar y publicar los programas. De esta manera, en esta fase se establece la especificación del estándar y, por

¹Más información sobre CIECODE en su página web: <https://www.ciecode.es>

consiguiente, del sistema informacional del asistente virtual.

En la tercera y última fase, denominada *Desarrollo*, se diseña el estándar para programas electorales, a partir de la información recopilada en las fases anteriores, y se implementa el asistente virtual para acceder a los programas electorales ya estandarizados. Cabe destacar que para el desarrollo del asistente virtual se emplea un modelo de prototipado evolutivo.

En la Figura 1.2 se pueden observar las tres fases aplicadas a los dos elementos principales a desarrollar en el proyecto, que son el **estándar** y el **asistente virtual**. En posteriores capítulos se detallarán las actividades realizadas siguiendo la metodología planteada.

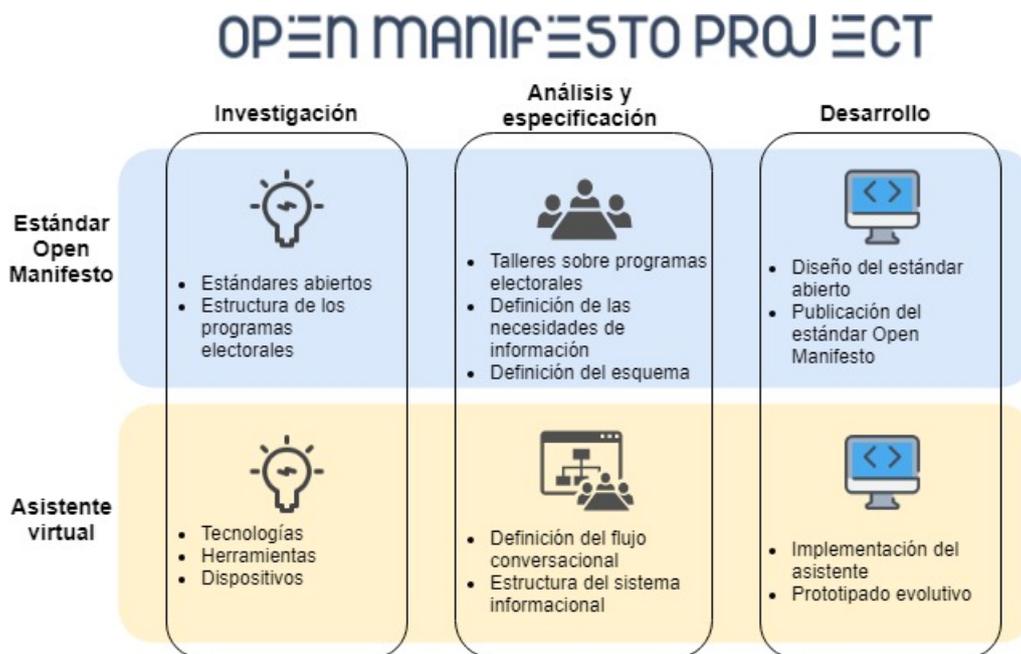


Figura 1.2: Metodología de desarrollo del proyecto. Fuente: elaboración propia

1.5 Estructura de la memoria

Esta memoria está compuesta por seis capítulos, incluida la introducción, que se estructuran de la siguiente forma:

- En el Capítulo 2, se establece el marco conceptual en que se desarrolla el trabajo. En concreto, se realiza un análisis de estándares abiertos, desde definiciones a metodologías, y se analiza el estado actual de los de asistentes virtuales.
- En el Capítulo 3, se presenta la metodología utilizada para realizar el diseño del estándar de programas electorales llamado *Open Manifesto*. También se detallan las fases del desarrollo, desde el análisis y especificación hasta el diseño final.
- En el Capítulo 4, se presenta la metodología utilizada para implementar el asistente virtual llamado *Analista Político Personal*. También se describen las tecnologías y herramientas utilizadas, al igual que se detallan las tareas realizadas durante la implementación.

- En el Capítulo 5, se analizan los resultados obtenidos tras la implementación del estándar y del asistente virtual.
- Por último, el Capítulo 6 recoge las conclusiones extraídas y las posibles líneas de trabajo futuras o ampliaciones. También se enumeran las apariciones en prensa y los reconocimientos obtenidos en relación al proyecto realizado.

CAPÍTULO 2

Estado del arte

En este capítulo se exponen los conceptos teóricos que sirven de base para el desarrollo del trabajo final de máster. Para ello, se dedican dos secciones a las principales temáticas tratadas en el proyecto. Por un lado, en la sección 2.1 se presentan definiciones y ejemplos de estándares abiertos. Por otro lado, en la sección 2.2 se realiza un análisis de los asistentes virtuales y el impacto que tienen en la actualidad.

2.1 Estándares abiertos

Según la Organización Internacional de Estandarización (ISO)¹ y la Comisión Internacional Electrotécnica (IEC)², un estándar es un "documento, establecido por consenso y aprobado por un organismo reconocido, que proporciona para uso común y repetido, reglas, pautas o características para actividades o sus resultados, dirigido al logro del grado óptimo en un contexto dado"[10]. Esta definición formal de ISO va asociada a la implementación o el uso de estándares para la evaluación de la conformidad, que es el proceso para evaluar o medir si en cierto contexto se cumplen con los requisitos establecidos.

En referencia al nivel de apertura, un estándar puede ser cerrado (también llamado propietario) o abierto. El estándar cerrado está disponible bajo términos restrictivos establecidos por un contrato con la organización que posee el *copyright*³ de la especificación.

En cuanto a los estándares abiertos, existen varias definiciones. En el Diccionario del español jurídico se define como un "modelo estándar que reúne las siguientes condiciones: a) que sea público y su utilización sea disponible de manera gratuita o a un coste que no suponga una dificultad de acceso, y b) que su uso y aplicación no estén condicionados al pago de un derecho de propiedad intelectual o industrial". Esta definición[11] se centra en el derecho mercantil del estándar.

La Fundación Europea de Software Libre (FSFE por sus siglas en inglés)[12][13] define un estándar abierto como un formato o protocolo que está sujeto a evaluación pública, que se puede usar sin restricciones y está disponible por igual para todas las partes, libre de cláusulas legales o técnicas que limiten su uso y que no tenga ninguna dependencia

¹Página web de ISO: <https://www.iso.org/home.html>

²Página web de IEC: <https://www.iec.ch>

³También conocido como derechos de autor, es el derecho exclusivo de un autor, editor o concesionario para explotar una obra.

con formatos, protocolos o entidades propietarias.

Desde las administraciones públicas se está regulando e integrando el uso de estándares abiertos. Por ejemplo, la definición anterior de la FSFE está basada en el Marco Europeo de Interoperabilidad[14] de la Comisión Europea. Además, en el gobierno de Reino Unido se ha creado un trabajo llamado *Open Standard Principles*(Principios de los Estándares Abiertos)[15] publicado originalmente en 2012 y revisado en 2015, en el que destacan el compromiso con la modernización digital, el aumento de la accesibilidad, al igual que presentan mejoras en la forma en que los documentos se generan, editan y almacenan adoptando formatos de documento abierto (ODF)⁴.

A nivel español, en el Real Decreto 4/2010, de 8 de enero, por el que se regula el Esquema Nacional de Interoperabilidad en el ámbito de la Administración Electrónica[16], y siguiendo las recomendaciones de la Unión Europea, se enuncia que "las Administraciones públicas usarán estándares abiertos, así como, en su caso y de forma complementaria, estándares que sean de uso generalizado por los ciudadanos, al objeto de garantizar la independencia en la elección de alternativas tecnológicas por los ciudadanos y las Administraciones públicas y la adaptabilidad al progreso de la tecnología. Las administraciones podrán usar los estándares que estén incluidos en el Catálogo de Estándares"[17].

Aunque la norma sea de 2010, el Esquema Nacional de Interoperabilidad entró en vigor en 2014. Pero el grado de implantación desde entonces parece que no ha sido el deseado puesto que, como se recoge en esta [18] noticia, la mayoría de documentos públicos están publicado en formato PDF o en formatos propietarios.

En los siguientes apartados se exponen iniciativas que promueven la estandarización y metodologías que establecen criterios para los procesos de estandarización abierta.

2.1.1. Open Contracting Partnership

La Alianza para las Contrataciones Abiertas [19] nació en el 2012 con el objetivo de promover una mayor transparencia en los procesos de contrataciones públicas, a través de un estándar de Datos para la Contratación Abierta creado en 2014.

Este estándar de datos abiertos ha sido creado para publicar información estructurada en todas las etapas de un proceso de contratación, desde la planificación hasta la implementación. Además, pretende que los datos publicados siguiendo este modelo permitan una mayor transparencia en la contratación pública y el respaldo de un análisis accesible de la eficiencia, efectividad, equidad e integridad de los sistemas de contratación pública.

A partir del desarrollo del estándar (*Open Contracting Data Standard*)[20] y de los pasos que realizaron para su implementación, crearon una metodología[21] formada por siete pasos para la apertura de datos de contratación abierta.

2.1.2. OpenStand

OpenStand es un movimiento dedicado a promover un conjunto de principios que establecen un paradigma moderno para los estándares[22]. Fue creado en 2012 y apo-

⁴Más información en <http://www.opendocumentformat.org>

yado por organizaciones como IEEE⁵, la *Internet Society*(ISOC)⁶, *Internet Engineering Task Force*(IETF)⁷ o la *World Wide Web*(W3C)⁸.

Los principios que establecen en OpenStand se basan en muchos de los estándares usados por las organizaciones que han apoyado el movimiento. De forma resumida, los principios son los siguientes:

- Cooperación entre organizaciones de estandarización.
- Adhesión al proceso, consenso, transparencia, equilibrio y apertura en el desarrollo de estándares.
- Compromiso con el mérito técnico, la interoperabilidad, la competencia, la innovación y el beneficio para la humanidad.
- Disponibilidad global de los estándares.
- Adopción voluntaria de los estándares.

2.1.3. Open Source Definition

Bruce Perens, el creador de la *Open Source Definition*[23] [24], también describió seis criterios que debe cumplir un estándar abierto y se enumeran a continuación:

1. Disponibilidad para todos, tanto para leerlos como para implementarlos.
2. Crean un mercado justo y competitivo para las implementaciones del estándar. No encierran al cliente en un proveedor o grupo en particular.
3. Son gratuitos para que todos los implementen, sin tasas ni impuestos.
4. Sin discriminación: los estándares abiertos y las organizaciones que los administran no favorecen una implementación sobre otra por ningún motivo que no sea el cumplimiento de los estándares técnicos de la implementación de una compañía.
5. Las implementaciones de estándares abiertos pueden extenderse u ofrecerse en forma de subconjunto.
6. Prácticas predatorias: Los estándares abiertos pueden emplear licencias que los protejan de las subversiones del tipo “aprovechar y extender”. Las licencias aplicadas al estándar pueden requerir la publicación de la información de referencia para las extensiones, y una licencia para que los demás puedan crear, distribuir y vender software compatible con las extensiones. No obstante, un estándar abierto no debe prohibir las extensiones.

2.1.4. ITU-T

ITU-T es uno de los tres sectores de la Unión Internacional de Telecomunicaciones (ITU)⁹ que se dedica a el desarrollo de estándares abiertos. Este grupo realizó una definición de los estándares abiertos en 2005[25] y es la siguiente: *son estándares puestos a*

⁵Página web de IEEE: <https://www.ieee.org>

⁶Página web de ISOC: <https://www.internetsociety.org/es/>

⁷Página web de IETF: <https://www.ietf.org>

⁸Página web de W3C: <https://www.w3.org>

⁹Más información en su página web: <https://www.itu.int/en/Pages/default.aspx>

disposición del público y se desarrollan (o aprueban) y se mantienen a través de un proceso colaborativo e impulsado por consenso. Los estándares abiertos facilitan la interoperabilidad y el intercambio de datos entre diferentes productos o servicios y están destinados a una adopción generalizada.

También establecen una serie de criterios que deben tener estos estándares y se resumen como sigue:

- Desarrollo o aprobación siguiendo un proceso transparente y abierto para todas las partes.
- Garantía que el proceso no esté dominado por ningún grupo de interés.
- Consideración y respuesta a las partes interesadas durante el proceso de desarrollo.
- Los derechos de propiedad intelectual son esenciales al implementar un estándar.
- Calidad y nivel de detalle para permitir el desarrollo de variantes interoperables.
- De fácil acceso para su uso e implementación.
- Mantenido y apoyado durante un largo período de tiempo.

2.1.5. Open Data Institute

El Instituto de Datos Abiertos (ODI) fue creado en 2012 por Tim Berners-Lee¹⁰ y por Nigel Shadbolt¹¹ para abogar por el uso innovador de datos abiertos para generar un cambio positivo en todo el mundo.

Para ello, han creado una guía[26] para ayudar a personas y organizaciones a crear, desarrollar y adoptar estándares abiertos para los datos. Así pretenden que los estándares abiertos para datos sean acuerdos reutilizables que facilite a las personas y organizaciones publicar, acceder, compartir y usar datos de calidad.

En esta guía se definen un conjunto de criterios y metodologías para crear estándares abiertos de datos. En la Figura 2.1 se pueden ver las etapas para el desarrollo de un estándar abierto. De forma resumida, tendría una etapa de inicio, donde se define el alcance, una etapa de desarrollo, una etapa de lanzamiento y adopción del estándar y, por último, una etapa de mantenimiento o actualización. Aunque parece un ciclo lineal, algunas de las etapas son repetidas o trabajadas simultáneamente.

¹⁰Más información en el directorio de ODI: <https://theodi.org/person/sir-tim-berners-lee/>

¹¹Más información en el directorio de ODI: <https://theodi.org/person/sir-nigel-shadbolt/>

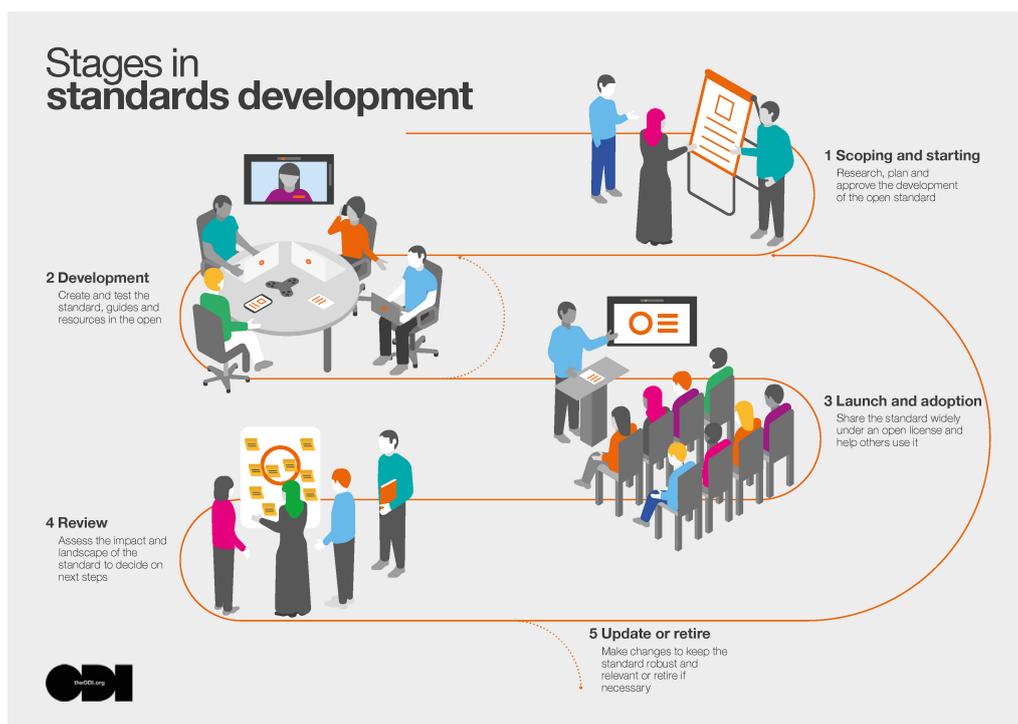


Figura 2.1: Etapas para el desarrollo de un estándar abierto. Fuente: [27]

2.2 Asistentes virtuales

El uso de asistentes virtuales está creciendo de forma significativa los últimos años. El auge de estos asistentes activados por voz se debe a que están presentes en muchos dispositivos, especialmente en los teléfonos inteligentes (*smartphones*), que ya forman parte de la vida diaria de los usuarios y que, en su mayoría, tienen un asistente incorporado en su sistema.

La historia de los asistentes virtuales se remonta a la década de los sesenta, donde IBM ya trabajaba en herramientas para el reconocimiento de voz, con IBM Shobox. Tras muchos años, donde hubieron varios proyectos de computadores capaces de realizar alguna tarea de reconocimiento del habla (como contestadores automáticos), fue en el año 2011 donde salió al mercado el primer asistente integrado en un teléfono móvil llamado Siri.

Desde entonces, ha habido una gran evolución y el número de asistentes virtuales se ha multiplicado. Las grandes empresas tecnológicas tienen sus propios asistentes como Siri de Apple, *Google Assistant* de Google, Cortana de Microsoft o Alexa de Amazon. Pero no solo las grandes tecnológicas tienen sus asistentes, otras empresas están empezando a crear sus propios asistentes, como es el caso de Aura de Movistar¹².

Estas nuevas tecnologías están suponiendo un cambio importante en el acceso a la información puesto que, no solo están incorporados en los teléfonos móviles o altavoces inteligentes, sino que se irán integrando cada vez más a componentes de la vida cotidiana como automóviles, frigoríficos o televisores. [28][29]

¹²Más información en: <http://www.movistar.es/particulares/Aura/>

La Asociación para la Investigación de Medios de Comunicación (AIMC)¹³ ha realizado un análisis llamado Estudio General de Medios (EGM) [30], realizado el primer trimestre de 2019, que por primera vez incorpora preguntas sobre nuevos dispositivos tecnológicos. En concreto, han incluido preguntas sobre el uso de asistentes virtuales de voz. Los datos obtenidos muestran que un 10.7% de la población española utiliza este tipo de dispositivos, es decir, más de cuatro millones de individuos. Los resultados en cuanto a qué asistente es más utilizado, un 86% de los encuestados afirman utilizar Siri. Lejos queda el segundo más utilizado que es el asistente de Google. En la Figura 2.2 se pueden comprobar estos datos:

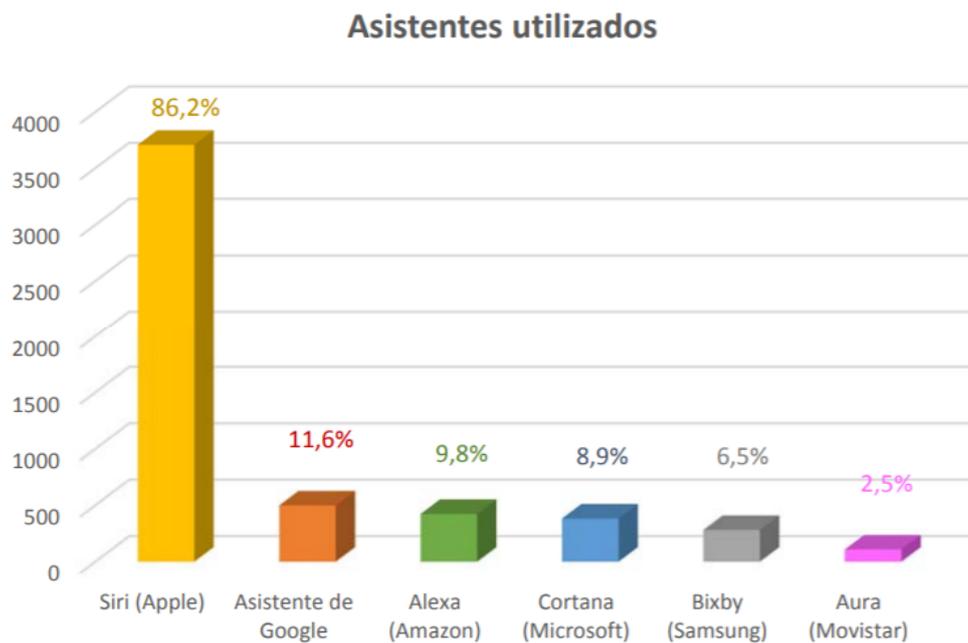


Figura 2.2: Asistentes más utilizados por los españoles. Fuente: [30]

Otro dato a destacar del estudio anterior hace referencia a la penetración en los hogares de los altavoces inteligentes. En este caso, un 2.6% de la población afirma tener uno de estos dispositivos, lo que significa casi medio millón de personas. El altavoz más usado es el Google Home, como se puede ver en la Figura 2.3.

¹³Más información en su página web: <https://www.aimc.es/>

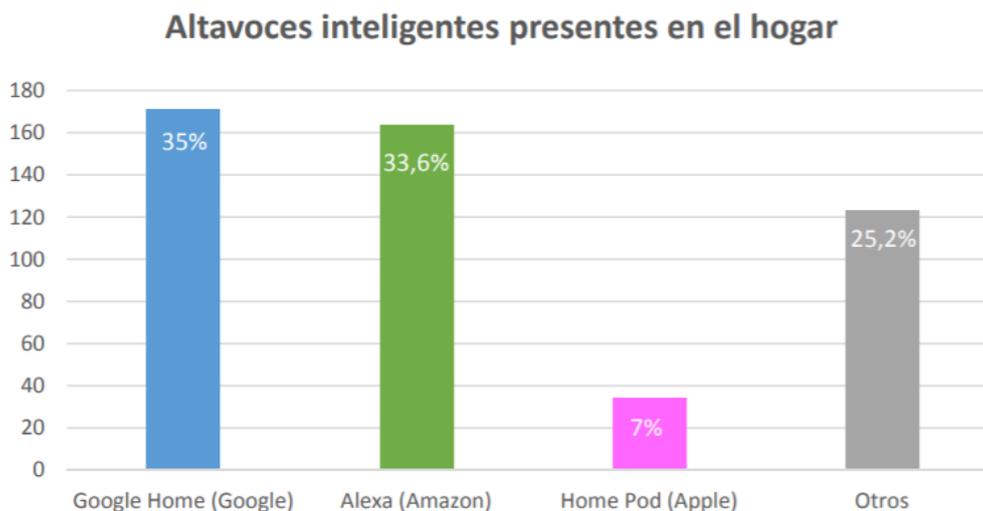


Figura 2.3: Altavoces inteligentes más usados por los españoles. Fuente: [30]

Por último, en las siguientes secciones se define el concepto asistente virtual y se analizan los principales asistentes virtuales y los dispositivos de acceso a ellos.

2.2.1. Definiciones

En la literatura no existe una terminología común puesto que los asistentes virtuales[31] también se conocen como asistentes de voz, asistentes inteligentes[32], *chatbots*, agentes de inteligencia artificial[33], y otros varios sinónimos que se alejan de los utilizados comúnmente. Aunque se utilicen muchos términos, todos ellos se refieren al mismo concepto.

En este trabajo, se utiliza el término asistente virtual como un programa informático capaz de reconocer el lenguaje natural de los usuarios y de realizar respuestas inteligentes. Se llama respuesta inteligente porque el sistema debe ser capaz de identificar qué es lo que desea el usuario y efectuar una respuesta acorde a esa petición.

Así pues, el asistente virtual tiene dos componentes claros que son la interfaz externa e interna. La externa es la encargada de la recepción de la información y de la emisión de la respuesta. La interfaz interna es la encargada de procesar la petición del usuario y seleccionar la respuesta o la acción correcta en función de la entrada. Para ello hacen uso de herramientas de procesamiento e interpretación del lenguaje natural o de herramientas de inteligencia artificial. Los asistentes virtuales no solo se basan en la entrada eventual del usuario, sino en el aprendizaje de muchas entradas para dar la respuesta correcta.

2.2.2. Asistente de Google

El asistente de Google ¹⁴ fue lanzado en mayo de 2016. Nació como una evolución de la búsqueda por voz que ya estaba integrada en los móviles con sistema operativo *Android*.

¹⁴Más información en: https://assistant.google.com/intl/es_es/

Se ha consagrado como uno de los asistentes de voz más avanzados tanto por la capacidad de reconocimiento del lenguaje como por la variedad de dispositivos desde los que se puede acceder. *Google Assistant* está integrado de forma nativa en todos los dispositivos *Android*, también es accesible desde el sistema *iOS* (*Apple*). Google ha lanzado dispositivos propios para hacer uso de su asistente como los altavoces inteligentes *Google Home* o *Google Home Mini*.

Otro aspecto que destaca es la posibilidad de ejecutar acciones por voz en el hogar haciendo uso de dispositivos domóticos compatibles con el asistente como luces, persianas, *Chromecast* (receptor multimedia) o directamente controlar televisores y electrodomésticos.

Por último, donde reside la potencia real de este asistente es en su directorio de aplicaciones¹⁵ puesto que cualquier desarrollador puede crear nuevas funcionalidades llamadas acciones para el asistente. Además, Google proporciona documentación y herramientas para los desarrolladores.

2.2.3. Siri

Siri¹⁶ de Apple es el primer asistente virtual integrado en teléfonos móviles, puesto que fue lanzado en octubre de 2011.

Es uno de los asistentes más conocidos y utilizados, pero solamente es compatible con los dispositivos iOS. A nivel de reconocimiento e interpretación del lenguaje, está por debajo de otros asistentes. Tiene un ecosistema cerrado y controlado que le dota de facilidad de configuración y uso, siempre dentro de el entorno iOS.

2.2.4. Alexa

Alexa¹⁷ de Amazon fue lanzado en noviembre de 2014 junto al altavoz inteligente Amazon Echo. Es el primer asistente integrado en un dispositivo independiente (el altavoz) a los que ya estaban en el mercado.

Al poco tiempo del lanzamiento crearon el *Alexa Skill Kit*, que permite a los desarrolladores crear nuevas funcionalidades para el asistente llamadas *skills*, facilitando herramientas, API, y documentación. Esto, al igual que el asistente de Google, permite que el usuario tenga variedad de funcionalidades a elegir en el directorio de aplicaciones de Amazon.

En principio solamente era accesible por los altavoces inteligentes creados por Amazon, pero hoy en día es compatible con gran variedad de dispositivos. Amazon ofrece una gran gama de altavoces integrados con Alexa de forma nativa, al mercado español han llegado hasta seis tipos de *Amazon Echo*. Este asistente también permite el control de elementos del hogar que sean compatibles como enchufes, bombillas, interruptores, entre otros.

¹⁵Directorio de aplicaciones de Google Assistant: <https://assistant.google.com/explore?hl=es>

¹⁶Más información en: <https://www.apple.com/es/siri/>

¹⁷Más información en: <https://developer.amazon.com/es/alexa>

Por último, destacar que es uno de los asistentes que mejor interpreta el lenguaje natural junto al asistente de Google.

2.2.5. Cortana

Cortana¹⁸ fue lanzado en abril de 2014 por Microsoft. La propia empresa lo define como un asistente digital personal que ha sido diseñado para ayudar a los usuarios en tareas básicas y proporcionar respuesta a sus preguntas.

Funciona solamente en dispositivos con el sistema operativo Windows, en concreto en los ordenadores con dicho sistema. Está restringido a realizar funciones dentro del sistema(alarmas, tareas...), aunque también ofrece la posibilidad de resolver dudas consultando en la web.

La desventaja en el mercado frente a los otros asistentes virtuales con mayor presencia como Alexa y Google Assistant, ha llevado a Microsoft a integrarlo como una aplicación más dentro de ambos asistentes¹⁹.

¹⁸Más información en: <https://www.microsoft.com/es-es/windows/cortana>

¹⁹Noticia relacionada:<https://planetachatbot.com/microsfot-cortana-integrara-en-alex-google-assistant-22d4281>

CAPÍTULO 3

Diseño del estándar Open Manifesto

En este capítulo se presenta el desarrollo del estándar para los programas electorales *Open Manifesto*. Para ello, en la primera sección se detalla la metodología empleada para desarrollar el estándar. En las siguientes secciones, se describen las distintas fases del proceso de desarrollo, desde el análisis y especificación al diseño final.

3.1 Metodología

Siguiendo las tres fases establecidas en la Figura 1.2, en la sección 2.1 se han analizado ejemplos y metodologías para el desarrollo de estándares abiertos, tarea que forma parte de la fase de Investigación. Por tanto, en este capítulo se completarán las tareas del proceso de desarrollo del estándar *Open Manifesto*.

En concreto, se desarrollarán las siguientes actividades de cada fase de la metodología establecida:

- Fase de Investigación:
 - Estudio de la estructura de los programas electorales y de los contenidos para valorar las similitudes y divergencias que existen entre ellos.
- Fase de Análisis y especificación:
 - Realización de jornadas de trabajo (talleres) para conocer las necesidades de información de los ciudadanos en referencia a los programas y las propuestas electorales siguiendo así un diseño centrado en la ciudadanía. Las conclusiones extraídas servirán de guía para la creación del estándar.
 - Evaluación de las distintas propuestas para estructurar y publicar el estándar.
- Fase de Desarrollo:
 - Creación del estándar para programas electorales que define un conjunto de restricciones de integridad y reglas relacionadas con su estructura y contenido.

3.2 Estructura de los programas electorales

Para el diseño de un estándar de programas electorales es necesario conocer cómo se estructuran dichos documentos. Por ello, se realiza un análisis sistemático que permita definir la estructura actual de los programas y determinar las similitudes y diferencias existentes entre ellos.

En la fase de investigación se han analizado algunos de los programas de los procesos electorales de ámbito nacional, local, autonómico y europeo que en concreto son los siguientes:

- Programas de las elecciones generales del 20 de diciembre de 2015 de PSOE, PP, Ciudadanos y Podemos.
- Programas de las elecciones generales del 26 de junio de 2016 de PSOE, PP, Ciudadanos y Podemos.
- Programas de las elecciones generales del 28 de abril de 2019 de PSOE, PP, Ciudadanos, Unidas Podemos y VOX.
- Programas de las elecciones europeas del 26 de mayo de 2019 de PSOE, PP, Ciudadanos, Unidas Podemos y VOX.
- Programas de las elecciones autonómicas de la Comunidad Valenciana de 2015 y 2019 de los principales partidos: PSPV, PP, Ciudadanos, Compromís, VOX y Unides Podem.

De estos programas no solo se ha analizado la estructura sino también los formatos en los que se publican, el volumen de los documentos, la dificultad para encontrar propuestas concretas, entre otras. Las conclusiones extraídas del análisis se describen a continuación:

1. El formato en el que se publican la mayoría de los programas es en PDF. Existen algunas excepciones que permiten leer el programa en una página web o descargarlo en formato CSV.
2. El volumen de páginas de los programas de las elecciones generales de media supera las doscientas. Sin embargo, el volumen de los programas presentados para las elecciones europeas no supera las cien páginas. Por último, en el caso de los presentados para las elecciones autonómicas de la Comunidad Valenciana, también se trata de documentos extensos que de media superan las cien páginas.
3. Todos los programas electorales tienen un índice que estructura la información contenida en ellos.
4. La mayoría de los programas tienen una introducción que presenta en líneas generales la ideología del partido y el camino que seguirá el documento.
5. Todos los programas presentan propuestas electorales concretas, también llamadas promesas, aunque en algunos casos son difíciles de encontrar debido a la extensión del programa y a la retórica presente en ellos, puesto que es no es fácil determinar dónde comienza y dónde termina el texto de la propuesta, sobre todo en los casos en los que no se enumeran. Existen excepciones en las que el programa electoral solamente está compuesto por las propuestas electorales enumeradas. También hay

partidos que presentan la versión larga del programa, que presenta mayor argumentación, y la corta, donde solo se encuentran las propuestas específicas.

6. El número de propuestas electorales en el caso de las elecciones generales es superior o igual a cien en todos los casos. En los programas de las elecciones europeas y autonómicas es variable.
7. La memoria económica, que se trata de una previsión de gastos e ingresos en función de las medidas de cada partido, no está incluida en la mayoría de los casos. Hay casos que sí la presentan de forma detallada.

En resumen, se trata de documentos extensos, con cantidad de información retórica y que no presentan, en la mayoría de los casos, la viabilidad financiera de las medidas descritas en el programa¹. No existen grandes diferencias entre los programas electorales de las elecciones generales de los partidos mayoritarios en cuanto a estructura, a excepción de los programas que solamente incluyen las medidas concretas. En cuanto a los programas de las elecciones europeas son menos voluminosos y se centran en las propuestas concretas. Por último, los programas de las elecciones autonómicas analizados son similares a los de las elecciones generales. Cabe destacar que la unidad mínima de expresión en un programa electoral son las propuestas, y esto es algo que se cumple en todos los casos.

3.3 Análisis y especificación

En esta sección se describen las actividades realizadas en la fase de análisis y especificación del estándar *Open Manifesto*. Para ello, siguiendo un diseño centrado en la ciudadanía, en el primer apartado se determinan y catalogan las necesidades de información en relación a las propuestas electorales a través de jornadas de trabajo. En el segundo apartado, se analizan las tecnologías y herramientas para el diseño y publicación del estándar.

3.3.1. Jornadas de trabajo

Como se ha mencionado anteriormente, para conocer las necesidades de información de los ciudadanos al respecto de los programas y propuestas electorales se han organizado unas jornadas de trabajo en Madrid, Valencia y Barcelona. La dinámica empleada en las jornadas de trabajo estaba centrada en especificar los requisitos formales para crear un estándar abierto de programas electorales que facilite la consulta automatizada de sus contenidos, centrado tanto en la globalidad del documento como en las características de las propuestas.

En cuanto a los asistentes a dichas jornadas destacar que tienen varios perfiles. Estos perfiles están en la línea de los usuarios potenciales del estándar y son los siguientes: agentes sociales (organizaciones, activistas...), partidos políticos y ciudadanía en general de edades comprendidas entre los veinte y los sesenta años.

En primera instancia, se enumeran las necesidades de información extraídas en referencia a la globalidad del programa, es decir, las que referencian al documento en general:

¹Artículo relacionado: https://elpais.com/politica/2019/04/27/actualidad/1556394057_350548.html

1. Gestión del programa: conocer la metodología, actores o entidades que han participado en la realización del programa electoral.
2. Inclusión de memoria económica: viabilidad financiera de las medidas del programa, ya sea a nivel global o a nivel concreto de cada medida.
3. Datos cuantitativos concretos: cuántas medidas o propuestas contiene el programa.

En referencia a las propuestas concretas de los programas, las necesidades de información son las siguientes:

1. Temáticas concretas: especificar las temáticas concretas tratadas en cada propuesta.
2. Inclusión de memoria económica: viabilidad financiera de las medidas o propuestas.
3. Prioridad de la propuesta: conocer qué prioridad tiene una propuesta para el partido.
4. "Líneas rojas": saber si la propuesta es negociable o no para el partido en caso de negociación programática con otro u otros partidos.
5. Temporalidad: conocer qué límite temporal estima el partido para el cumplimiento de una propuesta o medida concreta.

Por último, en el Anexo B se pueden ver más detalles sobre las convocatorias de las jornadas de trabajo (también llamadas talleres) y fotografías tomadas durante las dinámicas empleadas para la especificación de los requisitos formales enumerados anteriormente.

3.3.2. Herramientas y tecnologías

Para el diseño del estándar se han valorado distintas propuestas, tanto para la estructura como para la publicación puesto que, la definición del estándar está sujeta a evaluación pública y se administra y desarrolla a través de un proceso transparente y abierto a las partes interesadas.

Es por ello que, a partir de los ejemplos estudiados en la sección 2.1, se ha desarrollado el estándar con un formato JSON Schema y se ha publicado en el repositorio del proyecto en Github. En los siguientes apartados se describen ambas tecnologías.

Github

Github [34] es una herramienta de desarrollo colaborativo que permite publicar, documentar, revisar y colaborar en proyectos. Estos proyectos, generalmente de *software*, son publicados en repositorios mediante el sistema de control de versiones Git [35]. Por tanto, Github es un servicio web de alojamiento de repositorios que dispone de una interfaz gráfica.

Por ello que se ha seleccionado esta herramienta para publicar el estándar *Open Manifiesto*, los programas estandarizados, y todos los proyectos relacionados con esta iniciativa, como las distintas aplicaciones del estándar entre las que se encuentra el asistente virtual *Analista Político Personal*.

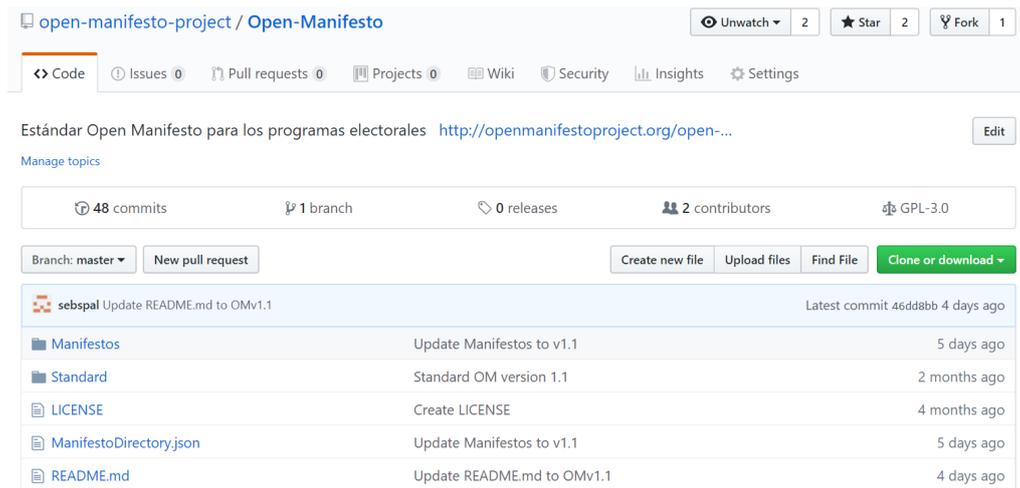


Figura 3.1: Repositorio del estándar *Open Manifesto* en Github

En la Figura 3.1 se puede observar el repositorio donde se han publicado las distintas versiones del estándar y los programas estandarizados de los distintos procesos electorales. Las distintas versiones del estándar se encuentran en la carpeta *Standard* y los programas estandarizados en la carpeta *Manifestos*.

JSON Schema

JSON Schema [36] es un vocabulario que permite describir y validar la estructura de documentos JSON [37]. Este esquema para la descripción de datos en JSON también proporciona documentación clara y legible tanto por humano como por máquina. Otro aspecto a destacar es la posibilidad de la validación automatizada de documentos JSON mediante el uso de librerías de lenguajes de programación que hagan uso del JSON Schema.

En la Figura 3.2 se puede observar un ejemplo de JSON Schema, en el cual se define que un objeto *Persona* tiene dos atributos *nombre* y *edad*. En algunos atributos se puede especificar también características para ser válidos como máximos o mínimos, en caso de atributos numéricos, entre otros.



Figura 3.2: Ejemplo de JSON Schema. Fuente: [38]

Los atributos principales para definir un JSON Schema son los siguientes:

- **\$schema**: define la versión del esquema JSON en el cual se especificará la definición. Las distintas versiones son llamadas *draft*.
- **title**: indica el título para identificar el esquema.
- **description**: se emplea para describir la descripción de un atributo.
- **type**: para indicar el tipo que va a representar el esquema o el atributo.
- **properties**: es una lista no ordenada de propiedades de un objeto que define el conjunto de valores estáticos de cada instancia del objeto en cuestión.
- **required**: indica qué propiedades del objeto son obligatorias.
- **\$ref**: URI de un esquema que contiene la representación para una propiedad.

En resumen, se ha seleccionado este lenguaje para la definición del estándar por las ventajas que presenta y también siguiendo el ejemplo de otros estándares maduros como *Open Contracting Partnership* [20] que también está definido en JSON Schema.

3.4 Desarrollo

En esta sección se define la estructura y los atributos del estándar abierto *Open Manifiesto* a partir de la información generada en las fases anteriores. También se dedica una sección a la descripción del proceso de estandarización de programas electorales.

3.4.1. Estándar Open Manifiesto

El estándar *Open Manifiesto* que se describe en este apartado corresponde a la versión 1.1 publicada el 12 de agosto de 2019. Las versiones anteriores se pueden consultar en el repositorio del proyecto [39].

En primer lugar, la definición del estándar se divide en dos partes. Una parte que referencia a los atributos correspondientes a la globalidad del programa electoral y, la otra parte, hace referencia a las características de las propuestas concretas contenidas en el programa. Esta definición se basa en las conclusiones extraídas de las jornadas de trabajo y del estudio de la estructura de los programas electorales.

Por tanto, los atributos definidos en referencia a la globalidad del programa son los siguientes:

- **politicalParty**: nombre del partido político que ha publicado el programa electoral.
- **title**: título del programa electoral.
- **publicationDate**: fecha de publicación del programa electoral.
- **electionDate**: fecha de las elecciones electorales en las cuales se presenta el programa.

- ***electionsType***: el tipo de elecciones a las que se presenta el programa. Pueden ser elecciones generales, autonómicas, europeas o municipales.
- ***geographicalArea***: ámbito geográfico en el que se presenta el programa ya sea país, comunidad autónoma o municipio.
- ***standardVersion***: versión del estándar que se ha utilizado para la estandarización del programa en cuestión.
- ***URI***: identificador de recursos uniforme que indica el alojamiento del programa.
- ***createdBy***: nombre del equipo que haya generado la versión estandarizada.
- ***pages***: número de páginas que tiene el programa electoral que ha sido estandarizado.
- ***numProposals***: número de propuestas electorales que contiene el programa.

Los atributos definidos en referencia a las propuestas electorales son los siguientes:

- ***id***: identificador único de la propuesta de tipo numérico e incremental.
- ***body***: texto de la propuesta.
- ***topics***: temas principales sobre los que versa la propuesta siguiendo un conjunto de términos definido por el grupo *Open Manifesto Project*.
- ***tags***: palabras clave que referencian a los temas contenidos en el texto de la propuesta.
- ***priority***: prioridad de la propuesta que es asignada por el partido. Puede tener cuatro valores que son prioridad baja, media, alta o NA.
- ***budget***: indica si la propuesta tiene definido el coste que implica su realización. Es de tipo *boolean* (verdadero o falso).
- ***nonNegotiable***: indica si para el partido se trata de una propuesta innegociable, también llamadas "líneas rojas". Es de tipo *boolean* (verdadero o falso).
- ***agents***: lista de agentes o entidades incluidos o mencionados en la propuesta.

En cuanto a esta versión del esquema de programas electorales hay varios aspectos a destacar. El primero de ellos es que los atributos han sido definidos siguiendo una convención de escritura llamada *camelCase* que es utilizado en los esquemas para estructurar datos en internet de *schema.org* [40], entre otros. El objeto de esto es la futura adaptación de este estándar definido en JSON Schema a formatos para la publicación de datos estructurados en internet como puede ser *JSON for Linking Data* [41] o RDF (*Resource Description Framework*) [42]. También destacar que para que *Open Manifesto* sea un formato de datos estructurados en web debería ser admitido por *schema.org* y de esta forma identificable por los motores de búsqueda.

Otro aspecto importante es cómo se estructura y describe el JSON Schema que sirve para validar los programas electorales en formato JSON. Este documento describe un conjunto de reglas de integridad que se deben cumplir necesariamente para que el JSON del programa electoral sea válido y también sirve de guía para la definición de estos. Los atributos descritos anteriormente son requeridos todos, es decir, ninguno es opcional

aunque pueden estar vacíos en caso que la información requerida no exista. Además, los atributos relativos a las propuestas sirven para cada una de ellas puesto que el programa está compuesto por un listado de propuestas.

Por otra parte, la descripción y el nombre de los atributos ha sido descrita en inglés para ofrecer la posibilidad de extender el estándar fuera del ámbito nacional y hacerlo accesible, que es otra de las características esenciales de los estándares abiertos.

```
{
  "$schema": "http://json-schema.org/draft-06/schema#",
  "$ref": "#/definitions/ManifiestoSchema",
  "definitions": {
    "ManifiestoSchema": {
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "politicalParty": {
          "description": "Name of the political party",
          "type": "string"
        },
        "title": {
          "description": "Title of the electoral program",
          "type": "string"
        }
      }
    }
  }
}
```

Figura 3.3: Definición del estándar *Open Manifiesto*

En la Figura 3.3 se puede observar la primera parte del estándar abierto para programas electorales donde se define bajo el nombre de *ManifiestoSchema* y siguiendo la especificación del *draft-06* de JSON Schema. La especificación seguida por el estándar está sujeta a los futuros cambios o actualizaciones de JSON Schema.

Por último, la definición completa en formato JSON Schema del estándar se puede consultar en el Anexo A y en el repositorio del proyecto citado anteriormente. Ha sido publicado con una licencia abierta², sin dependencias a otros estándares que no sean abiertos y libre de cláusulas legales o técnicas que limiten su uso.

3.4.2. Proceso de estandarización

El proceso de estandarización de programas electorales a la versión 1.1 descrita en la sección anterior sigue una secuencia lineal que tiene un conjunto de etapas y tareas a desarrollar que se enumeran a continuación:

1. Identificación de los partidos y coaliciones que hayan obtenido representación. En caso de coaliciones, identificar los partidos que la conforman.
2. Búsqueda o solicitud de los programas electorales de los partidos identificados.
3. Elaboración de un listado con los atributos referentes a la globalidad del programa.
4. Identificación de las propuestas electorales dentro del programa.
5. Definición de los campos relativos a las propuestas concretas.

²<https://github.com/open-manifesto-project/Open-Manifiesto/blob/master/LICENSE>

6. Automatización el proceso de extracción de las palabras clave o *tags*.
7. Automatización la generación de los JSON para la validación del programa bajo las directrices del estándar.

La etapa cinco, que es la definición de lo campos relativos a las propuestas, tiene varios aspectos a remarcar. El primero de ellos es la definición del campo *topic*. Para el ámbito nacional se ha definido junto a CIECODE un conjunto de términos para clasificar las propuestas electorales de manera objetiva, pertinente y exhaustiva. Este sistema de categorías sería válido para el ámbito nacional porque se basa en un análisis que es relativo a la realidad sociopolítica de cada país. Las categorías definidas se pueden observar en la siguiente tabla:

Empleo	Legislación y justicia	Medio ambiente	Urbanismo e infraestructuras
Hacienda	Sociedad y bienestar	Cultura y Ocio	Industria
Economía	Igualdad de género	Ciencia y tecnología	Medio rural
Energía	Vivienda	Seguridad	Deporte
Salud	Migración	Transporte	Sector Público
Educación	Política Exterior	Turismo	Comercio

Además, destacar que el equipo de CIECODE y *Open Manifesto Project* ha efectuado las tareas de automatización con *scripts*(programas) definidos en el lenguaje de programación Python [43], tanto para la extracción de palabras clave como para la generación de los JSON con los programas estandarizados.

En la Figura 3.4 se puede observar parte del resultado de un programa estandarizado, en concreto del programa electoral del Partido Socialista para las elecciones generales del 28 de abril de 2019 siguiendo la versión 1.1 del estándar *Open Manifesto*. Los programas estandarizados por el equipo de *Open Mnaifesto Prroject* se pueden consultar en el repositorio del proyecto [44] dentro de la carpeta *Manifestos* y la versión actual y las anteriores del estándar en la carpeta *Standard*.

```
{
  "politicalParty": "PSOE",
  "title": "PROGRAMA ELECTORAL PSOE ELECCIONES GENERALES 2019 - Haz que pase",
  "publicationDate": "",
  "electionDate": "28/04/19",
  "electionType": "Generales",
  "geographicalArea": "Estado Español",
  "standardVersion": "1.1",
  "URI": "",
  "createdBy": "Open Manifiesto Project",
  "pages": "300",
  "numProposals": "741",
  "proposals": [
    {
      "id": 1,
      "body": "El Gobierno ha aprobado ya la transposición de varias directivas europeas que permiten reforzar la superv",
      "topics": [
        "Legislación y justicia"
      ],
      "tags": [
        "gobierno",
        "aprobado",
        "transposición",
        "varias",
        ...
      ]
    }
  ]
}
```

Figura 3.4: Programa electoral del PSOE estandarizado

Por último, destacar que uno de los objetivos de un estándar abierto en su fase final es la adopción del mismo. En este caso sería la adopción del estándar *Open Manifesto* por parte de los partidos políticos para definir los programas electorales.

Implementación del asistente Analista Político Personal

En este capítulo se presentan las fases y detalles del desarrollo del asistente llamado *Analista Político Personal*. Para ello, se dedica la primera sección para detallar la metodología empleada. También se dedica una sección para describir las herramientas y tecnologías elegidas para la implementación, donde se describen conceptos iniciales necesarios para su desarrollo. Posteriormente, en la sección 4.3 se especifican los requerimientos y se definen los casos de uso del asistente. Por último, se realiza la implementación del asistente.

4.1 Metodología

Al igual que en el estándar, la metodología sigue las tres fases establecidas en la Figura 1.2. En la sección 2.2 se ha analizado el estado actual de los asistentes virtuales, tarea que forma parte de la fase de Investigación. En este capítulo se completarán las tareas del proceso de desarrollo del asistente virtual.

En concreto, se desarrollarán las siguientes actividades de cada fase de la metodología establecida:

- Fase de Investigación:
 - Estudio de las tecnologías que se utilizarán para desarrollar el asistente y definición de los conceptos generales.
- Fase de Análisis y especificación:
 - Casos de uso del asistente virtual definidos a partir de las conclusiones extraídas de las jornadas de trabajo. Se definen las preguntas a las que se desea dar respuesta con el asistente.
 - Análisis de la estructura del sistema informacional, que es el conjunto de programas electorales estandarizados.
- Fase de Desarrollo:
 - Definición de la arquitectura del sistema.
 - Diseño del flujo conversacional a partir de los casos de uso.

- Implementación de la funcionalidad del asistente haciendo uso de las tecnologías y herramientas seleccionadas en la primera fase.
- Evaluación y seguimiento del asistente desarrollado. Como fase final del proceso de prototipado evolutivo, se debe realizar un aprendizaje en base a la interacción de los usuarios con el asistente.

4.2 Herramientas y tecnologías utilizadas

En esta sección se definen las tecnologías y conceptos básicos que sirven de base para el desarrollo del asistente. Para ello, se dedica un apartado a cada una de las herramientas utilizadas, introduciendo así los conceptos tecnológicos que se utilizarán en las siguientes fases.

4.2.1. Dialogflow

Dialogflow [45] es una herramienta adquirida por Google en 2016, que permite crear aplicaciones conversacionales (*chatbots*, asistentes virtuales) capaces de entender el lenguaje natural. Para ello, ofrecen una interfaz web que simplifica la tarea de comprender la entrada de voz de los usuarios, extraer palabras clave y devolver la respuesta correcta a dicha petición, esto se conoce como flujo conversacional. Además este servicio permite integrar las aplicaciones creadas con varias plataformas como el Asistente de Google, Slack, Telegram, entre otras.

Los componentes principales para la creación de estas aplicaciones son los siguientes:

- **Agents:** los agentes son los proyectos o módulos de comprensión del lenguaje natural. Traducen solicitudes de texto o de voz en datos procesables. La traducción se produce cuando la expresión de un usuario coincide con una de las intenciones registradas dentro del agente. La intención que coincida con los datos recibidos del usuario, entrega una respuesta. La respuesta puede ser texto simple, voz o una respuesta *Webhook* que obtiene la información de un sistema externo, como puede ser un JSON alojado en la nube.
- **Intents:** Los *Intents* o intenciones se crean en el agente con el objetivo de definir expresiones que representen las posibles entradas del usuario. Si la entrada de datos coincide con alguna de las expresiones definidas, se activa un *Intent* y este desencadena la extracción de la expresión para crear la respuesta correspondiente. Además, cada *Intent* tiene cuatro componentes principales:
 - **Nombre:** identifica la intención o *Intent* de la expresión de entrada.
 - **Frases de entrenamiento:** frases de ejemplo de lo que pueden decir los usuarios que coincida con una intención (*Intent*) concreta. Dialogflow expande estas frases para asociarlas con expresiones de usuario similares.
 - **Acción y parámetros:** con los parámetros se define cómo extraer la información relevante de las frases de entrada.
 - **Respuesta:** la posible respuesta creada en función de la frase de entrada.
- **Entities:** se usan para seleccionar la información específica de la entrada. Cada dato importante que se desee obtener de la petición de un usuario, tendrá una entidad correspondiente. Se pueden definir varios tipos de entidades, y estas pueden estar compuestas por un conjunto de palabras o frases que se consideran equivalentes.

- **Contexts:** representan el estado actual de la solicitud de usuario y permite al agente Dialogflow pasar información de un *Intent* a otro. En concreto, controla la ruta que toma la conversación con el usuario.
- **Events:** representan los eventos recibidos de la plataforma que se haya integrado con el agente, como puede ser *Google Assistant* o *Telegram*, basado en las frases de entrada de los usuarios. También se pueden crear eventos vía programación a través de *detectIntent API*.
- **Fulfillment:** es código fuente implementado como un *Webhook* que le permite a un agente Dialogflow llamar a la lógica de negocio en base a un *Intent*. Con esto se generan respuestas dinámicas o se desencadenan acciones dentro del sistema informacional o base de datos(*backend*).

Otro concepto a destacar es el *Webhook*. Es un método que permite comunicar en tiempo real, a través de HTTP, dos aplicaciones o servicios web. Interviene cuando Dialogflow genera una solicitud HTTP POST con un objeto JSON que contiene la información sobre el *Intent* coincidente. El *Webhook* realiza una tarea en un servicio externo, como puede ser una base de datos, y responde en el mismo formato.

Por otra parte, en la Figura 4.1 se muestra un diagrama general del funcionamiento de Dialogflow y del flujo de interacción con el usuario y entre los componentes del agente.

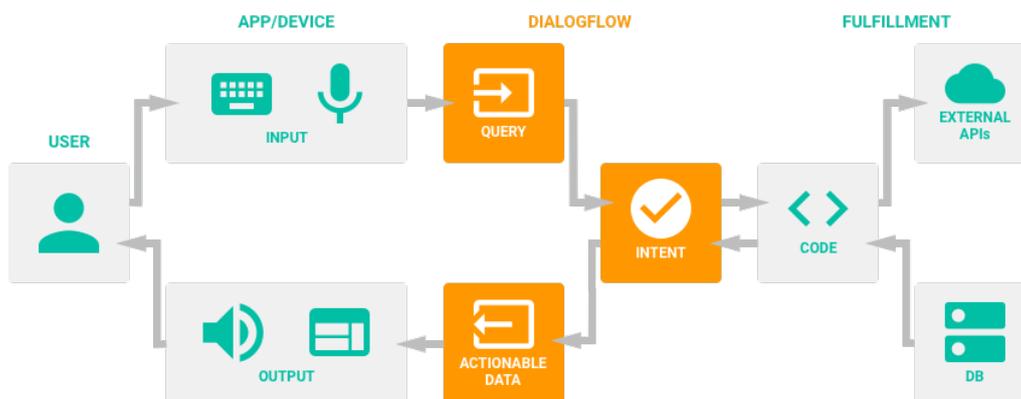


Figura 4.1: Diagrama del funcionamiento de Dialogflow. Fuente: [46]

Cuando un usuario interactúa con una aplicación conversacional de cualquier plataforma, como un Action de Google, el agente Dialogflow asociado a dicha aplicación, se encarga de reconocer el lenguaje natural y extraer la información necesaria, para identificar qué quiere el usuario. Cuando el agente ha identificado la intención (*Intent*), pueden suceder dos cosas: a) el *Intent* y, por consiguiente, el agente, tiene la respuesta a la petición y la devuelve, o b) el agente no tiene la respuesta y necesita datos externos. Por tanto, intervendrá el *Fulfillment* procesando y devolviendo la respuesta, accediendo al servicio externo que se haya implementado.

De esta manera, se puede decir que Dialogflow es una herramienta que actúa de intermediaria y que proporciona funcionalidades para el reconocimiento del lenguaje natural y aprendizaje automático. En este trabajo, se ha utilizado como plataforma base para la implementación del asistente virtual y el desarrollo del flujo conversacional.

4.2.2. Actions on Google

Actions on Google [47] es una plataforma que permite a los desarrolladores crear aplicaciones para *Google Assistant*. Para ello, dispone de una consola que es una herramienta web para crear, mantener, testear y publicar Actions. Actions es el nombre que reciben las aplicaciones desarrolladas para el asistente de Google y se publican en el directorio [48], donde los usuarios pueden descubrir e invocar las aplicaciones a través del asistente.

En este trabajo se ha utilizado para crear el proyecto *Analista Político Personal*, para integrarlo al ecosistema de *Google Assistant* y hacerlo accesible para su testeo por los diferentes dispositivos que pueden acceder a dicho asistente. En la siguiente Figura 4.2 se puede ver la interfaz de testeo que ofrece. También tiene varias funcionalidades sobre analítica de uso de los Actions publicados en el directorio, y permite cambiar las opciones del Action que se publique y controlar la versiones de este.

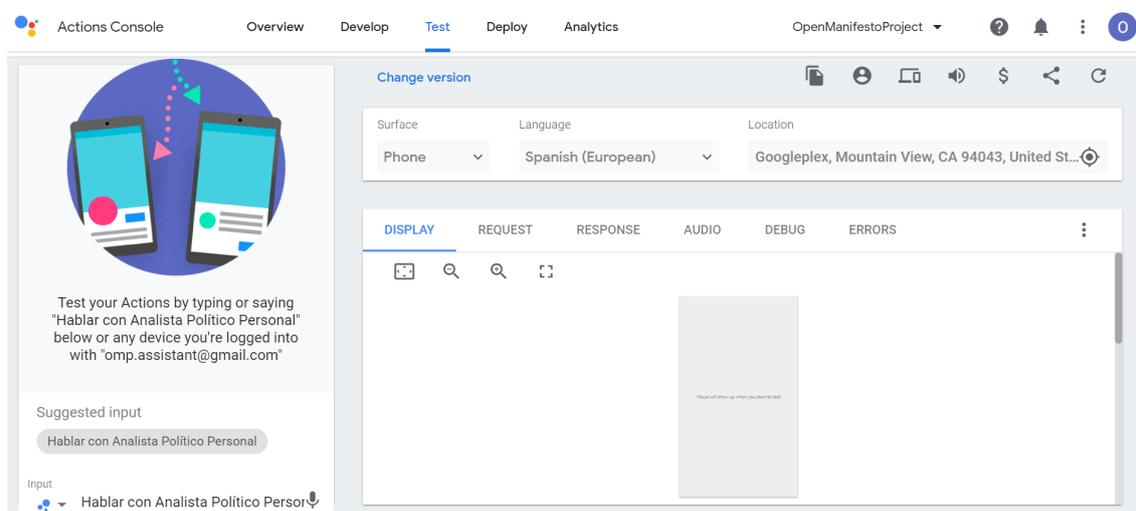


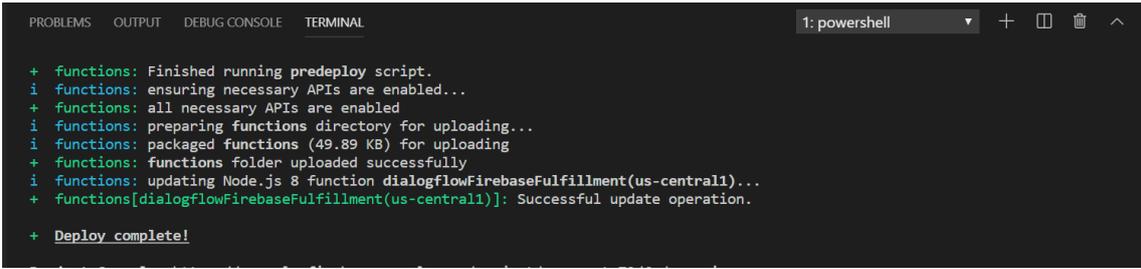
Figura 4.2: Interfaz web de Actions on Google. Fuente: elaboración propia

4.2.3. Firebase

Firebase [49] es una plataforma creada por Google, cuya principal función es facilitar la creación de aplicaciones. Para ello, ofrecen un conjunto de funcionalidades para el desarrollo como base de datos en tiempo real, alojamiento en la nube, *Cloud Functions for Firebase*, monetización con AdMob, entre otras.

En este proyecto se ha utilizado *Cloud Functions for Firebase*, que es un servicio que permite ejecutar el código del *Webhook* del asistente en un entorno almacenado y administrado en la nube. De esta forma, no es necesario administrar y escalar los servidores. Este paradigma de abstracción de la gestión de servidores se conoce como *serverless*.

Para hacer uso de esta funcionalidad es necesario trabajar con la interfaz de línea de comandos (CLI) de Firebase [50], que es donde se implementan las funciones del proyecto. La función Firebase almacenada en la nube contiene la lógica de negocio del asistente *Analista Político Personal*. En la siguiente Figura 4.3 se puede observar la consola tras la ejecución de la función llamada *dialogflowFirebaseFulfillment*.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell
+ functions: Finished running predeploy script.
i functions: ensuring necessary APIs are enabled...
+ functions: all necessary APIs are enabled
i functions: preparing functions directory for uploading...
i functions: packaged functions (49.89 KB) for uploading
+ functions: functions folder uploaded successfully
i functions: updating Node.js 8 function dialogflowFirebaseFulfillment(us-central1)...
+ functions[dialogflowFirebaseFulfillment(us-central1)]: Successful update operation.
+ Deploy complete!
```

Figura 4.3: Ejemplo de implementación en Firebase CLI. Fuente: elaboración propia

Por último, destacar que desde la interfaz web de la consola de Firebase se puede hacer seguimiento de la función creada con la lógica del asistente, tanto para detectar errores como para ver y buscar registros de ejecución.

4.2.4. Google Cloud

Google Cloud [51] es una plataforma utilizada para crear soluciones de la tecnología almacenada en la nube. También dispone de muchas herramientas para los desarrolladores de aplicaciones.

Google Cloud Functions es la solución de procesamiento sin servidor para crear aplicaciones controladas por eventos, es un producto conjunto de Google Cloud y Firebase. De esta forma, en este trabajo se ha usado las soluciones que ofrece para controlar la función *Webhook* del asistente virtual. En concreto, se ha utilizado las opciones que ofrece para controlar los errores de la función, llamado *Error Reporting*, entre otras.

4.2.5. Node.js

Node.js [52] es un motor de ejecución para JavaScript¹, de código abierto y que es utilizado, entre otras cosas, para desarrollar aplicaciones web.

En el contexto de este proyecto, se ha utilizado para crear el *Webhook* del asistente, que hace utiliza la librería de Node.js para desarrollar Actions on Google. De esta forma, la lógica de negocio del asistente ha sido implementada haciendo uso de esta librería puesto que ofrece facilidades para procesar las respuestas de los *Intents*, implementar el *Webhook* en una plataforma *serverless* y hacer peticiones a sistemas externos para enriquecer las respuestas dadas por el asistente. En la Figura 4.4 se puede ver la arquitectura de esta librería.

¹<https://www.javascript.com/>

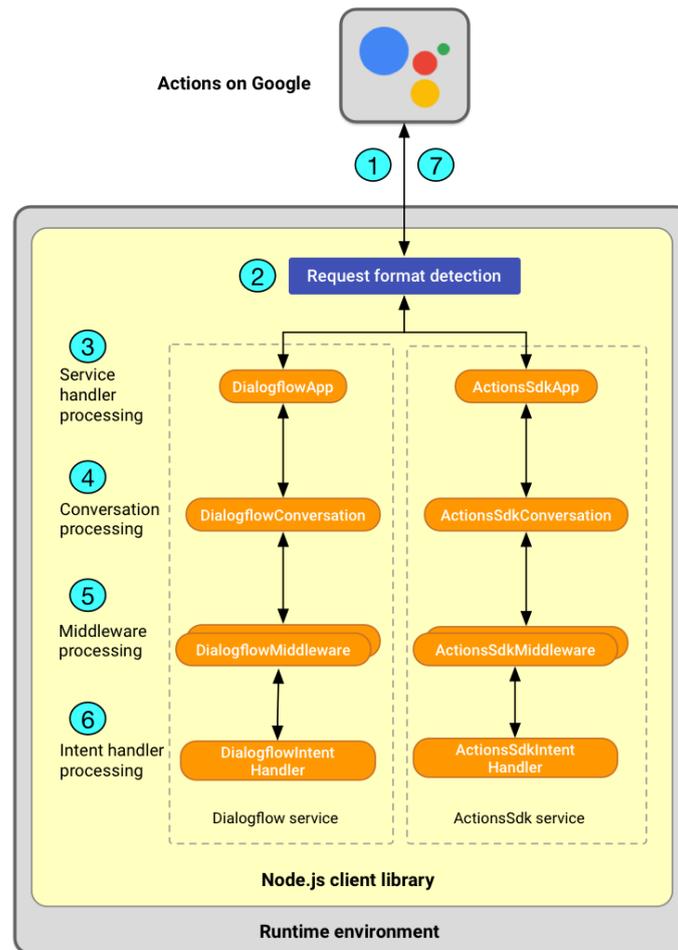


Figura 4.4: Arquitectura de la librería Actions on Google de Node.js. Fuente: [53]

4.3 Análisis y especificación

En esta sección, se realiza la especificación formal de los requerimientos para la implementación del asistente virtual. Para ello, se dedica una sección a la definición de los casos de uso. Por último, se analiza el sistema informacional al que accederá *Analista Político Personal*.

4.3.1. Casos de uso

En las jornadas de trabajo realizadas (más información en el apartado 3.3.1), no solo se definieron las necesidades de información de los ciudadanos con respecto a los programas electorales. También se extrajeron las posibles preguntas que se realizarían al asistente virtual para consultar los programas electorales y cómo se clasificarían dichas preguntas.

De esta forma, tras el análisis de las preguntas recogidas de las jornadas de trabajo, se definen el conjunto de casos de uso en los cuales solamente hay un actor, que es el usuario que interactúa con el asistente. Así pues, se listan las peticiones que realizaría el usuario al asistente *Analista Político Personal*:

- Consultar las propuestas concretas del programa electoral por la temática tratada.

- Consultar el número de propuestas concretas del programa electoral por temática tratada.
- Consultar las propuestas electorales de un partido en los distintos ámbitos electorales (elecciones generales, autonómicas, europeas y municipales).
- Poder consultar sobre el funcionamiento del asistente virtual.

En cuanto al formato del tipo de pregunta, en este trabajo se han categorizado como sigue:

- Preguntas **enunciativas**: son las preguntas realizadas para conocer las medidas del programa electoral referentes a un tema (empleo, vivienda...), y que reciben como respuesta el enunciado de la propuesta.
- Preguntas **enumerativas**: realizadas para conocer el número de medidas del programa electoral con respecto a un tema.
- Preguntas **funcionamiento**: relativas al asistente virtual. La respuesta debe ser cómo interactuar con el asistente.

4.3.2. Sistema informacional

El sistema informacional, es decir, el sistema al que accederá el asistente para dar respuesta a las peticiones de los usuarios, es el conjunto de programas electorales estandarizados en formato JSON alojados en un repositorio Github.

La estructura y parámetros de estos archivos JSON, definida en la sección 3.4, será básica para la consulta y también para la implementación de la lógica de negocio del asistente virtual.

En función de los tipos de preguntas definidas en el apartado anterior, se puede inferir que los parámetros básicos para realizar la búsqueda de propuestas en los programas son los siguientes:

- **Tema**: para dar respuesta a las preguntas enunciativas devolviendo una respuesta coincidente con la entrada del usuario, la temática de la propuesta es básica. De esta forma, este parámetro se hará referencia a el atributo *tags* de los programas estandarizados.
- **Partido**: para realizar la búsqueda del programa electoral correspondiente a la petición del usuario, es necesario conocer al partido al que se refiere. En los JSON de los programas estandarizados este dato está en el atributo *politicalParty*.
- **Elecciones**: otro dato necesario para realizar la búsqueda entre el conjunto de programas estandarizados es al tipo de elecciones a las que se refiere el usuario. En los programas estandarizados este dato se encuentra en el atributo *electionsType*.
- **Región**: en los casos en que el usuario haga consultas sobre programas presentados en elecciones locales o autonómicas, será necesario para la búsqueda conocer a qué región se refiere. Este dato se encuentra en el atributo *gographicalArea*.

Con estos datos, ya se puede establecer un nexo de unión entre las posibles entradas o peticiones del usuario y el sistema informacional. Quien tratará este nexo es el intérprete del lenguaje natural (agente Dialogflow) y la lógica de negocio (*Webhook*). Para facilitar la tarea de búsqueda, se ha generado un directorio del sistema informacional en formato JSON, disponible en el repositorio del proyecto², que contiene un listado con los parámetros básicos para la búsqueda de los programas electorales. Cada entrada de este directorio referencia a un programa electoral y especifica la URI donde se encuentra el recurso. En la Figura 4.5 se puede ver un ejemplo.

```
{
  "politicalParty": "PSOE",
  "type_of_elections": "Generales",
  "geographical_area": "Estado Español",
  "election_date": "28/04/19",
  "URI": "https://raw.githubusercontent.com/open-manifesto-project/Open-Manifesto/master/Manifestos/PSOE_GENERALES_ESTADO-ESPAÑOL_28-04-19_OMv1.1.json"
}
```

Figura 4.5: Atributos de un programa electoral en el directorio

4.4 Desarrollo

En esta sección, se detalla el proceso de desarrollo del asistente *Analista Político Personal*. En primer lugar, se proporciona una visión general de su arquitectura, los componentes que la conforman y la relación entre ellos. Posteriormente, se detallan los pasos realizados en la implementación, como la definición del flujo conversacional. Por último, se expone cómo se realiza la evaluación y seguimiento del asistente haciendo uso de las herramientas utilizadas.

4.4.1. Arquitectura

La arquitectura del asistente virtual está formada por distintos componentes. Estos componentes se corresponden a cada una de las tecnologías que intervienen en la interpretación del lenguaje natural y en el procesamiento de la respuesta.

En la Figura 4.6 se puede ver el diagrama de componentes que conforman la arquitectura del asistente.

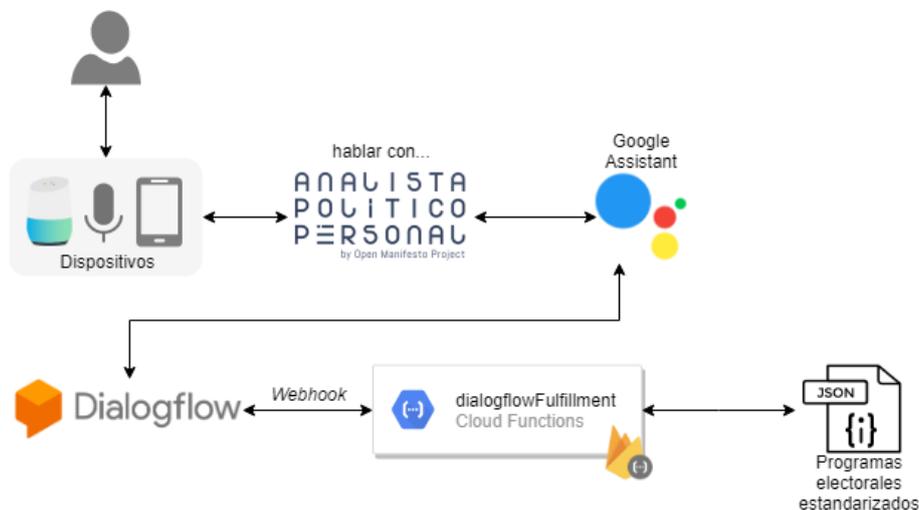


Figura 4.6: Componentes de *Analista Político Personal*. Fuente: elaboración propia

²<https://github.com/open-manifesto-project/Open-Manifesto>

Vista la arquitectura, se detalla la interacción entre los distintos componentes. Como se puede observar, el asistente virtual se ha desarrollado con las tecnologías y herramientas estudiadas. Por tanto, *Analista Político Personal* es un Action de Google accesible desde dispositivos que tengan integrado *Google Assistant*, como puede ser un dispositivo móvil o un altavoz inteligente. Esta parte es la encargada de recibir la petición del usuario y devolver la respuesta.

El componente encargado del procesamiento del lenguaje natural, es el agente Dialogflow, que es quien procesa las peticiones y las interpreta. El agente selecciona si la intención del usuario es coincidente con la funcionalidad implementada (*Intents*) y genera una solicitud. Esta solicitud HTTP POST, que contiene la información del *Intent* que ha coincidido con la petición del usuario, es recibida por el *Webhook*, que contiene la lógica para construir dinámicamente las respuestas a los usuarios.

El código del *Webhook* está alojado en una *Cloud Function* de Firebase (llamada *dialogflowFirebaseFulfillment*), y se activa cuando recibe una solicitud HTTP. Por tanto, cuando recibe el evento generado por el agente Dialogflow, se invoca el método que consulta el repositorio de programas estandarizados en formato JSON.

Por último, se genera la respuesta y es devuelta por el *Webhook* con el mismo protocolo en formato JSON. El agente Dialogflow la traduce a lenguaje natural y es recibida por el usuario.

4.4.2. Implementación

En base a los casos de uso definidos y a la estructura del sistema informacional, se crean los *Intents* que deberán cubrir las posibles preguntas de los usuarios al asistente virtual, y que deberán devolver las respuestas en base a los parámetros de las frases entrantes. Estos parámetros se definen en las *Entities*, que se utilizarán para procesar las respuesta, extrayendo los datos necesarios de la petición.

En los siguientes apartados, se describen las *Entities*, los *Intents* y el flujo conversacional del asistente. También se describen las funcionalidades que se han desarrollado en el *Webhook*.

Entities

Como se ha definido anteriormente, para seleccionar la información específica de las frases de entrada del asistente, se definen entidades para *Tema*, *Partidos*, *Elecciones* y *Región*, como se puede observar en la siguiente Figura 4.7.

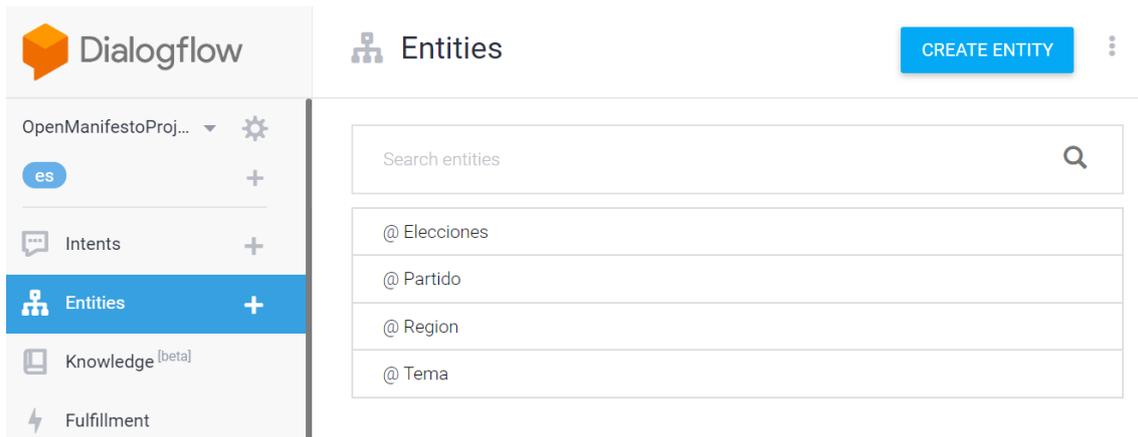


Figura 4.7: *Entities* del agente Dialogflow

Estas entidades deben cubrir toda la información que se considere necesaria, para posteriormente generar la respuesta en la lógica de negocio. Por ejemplo, en el caso de la entidad *Partidos*, se puede definir PSOE y sus sinónimos, que pueden ser Partido Socialista o Partido Socialista Obrero Español. De esta forma, se cubren todos los casos. En la siguiente Figura 4.8 se puede observar las entidades *Región*, donde se define la entidad principal y los posibles sinónimos de dicha entidad.

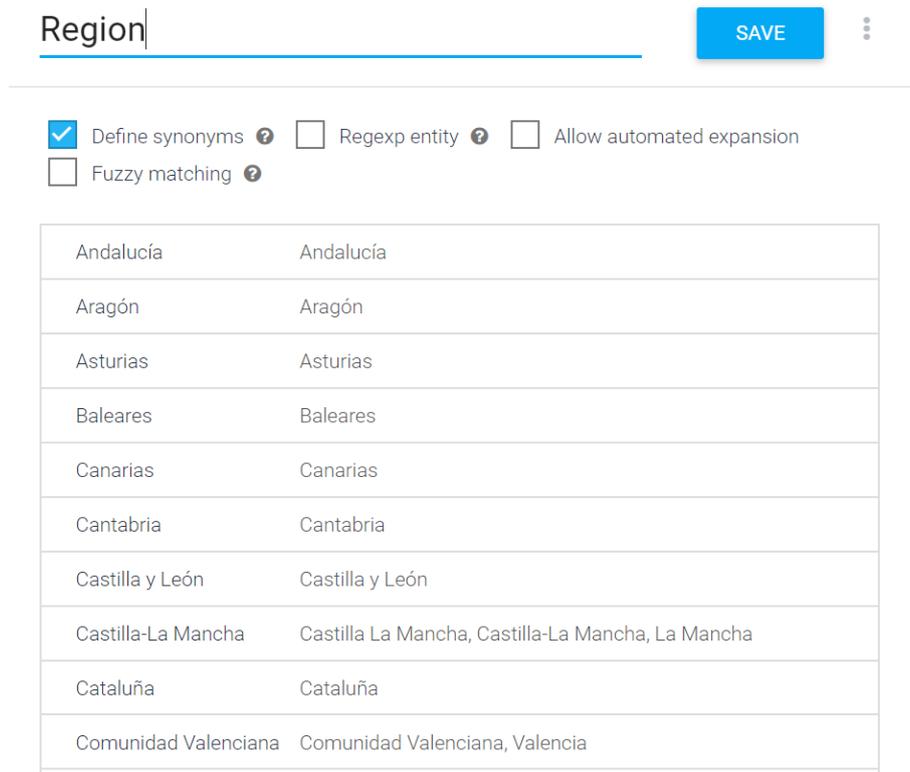


Figura 4.8: Ejemplo de la entidad *Región*

En casos como la entidad *Tema*, se dificulta la tarea de cubrir todos los casos, puesto que las temáticas tratadas en las propuestas electorales son muy amplias. Pero esto se soluciona en base al entrenamiento del agente, que lo veremos en detalle en próximos apartados.

Intents

Los *Intents* creados, están compuestos por las frases de entrenamiento que sirven para interpretar la intencionalidad del usuario y devolverle una respuesta. De esta manera, se han definido *Intents* que cubren las preguntas de tipo enunciativo referentes a las distintas elecciones.

A continuación, se dan ejemplos de la estructura de las preguntas enunciativas:

- En caso de elecciones generales o europeas: ¿Qué propuestas llevaba el *Partido* relacionada con *Tema* para las *Elecciones*?
- En caso de elecciones autonómicas o municipales: ¿Qué propuestas llevaba el *Partido* relacionada con *Tema* para las *Elecciones en Región*?

Como es lógico, en base al aprendizaje existen muchas variantes, pero la estructura genérica de las frases de entrenamiento es similar a los ejemplos expuestos. Se ha destacado en cursiva las *Entities*, que son básicas para la búsqueda de la respuesta.

Los *Intents* principales del agente son los siguientes:

- *Default Welcome Intent*: da la bienvenida al usuario cuando el asistente virtual es invocado a través de *Google Assistant* pronunciando la frase "hablar con Analista Político Personal".
- *IntentEnunciativaAutonomicasMunicipales*: contiene las frases de entrenamiento que coinciden con las preguntas sobre las propuestas de las elecciones autonómicas y municipales. Se ha decidido juntarlas porque la estructura sintáctica es la misma, con la variante del tipo de elección que se maneja a nivel de lógica de negocio.
- *IntentEnunciativaGeneralesEuropeas*: contiene las frases de entrenamiento que coinciden con las preguntas de las propuestas electorales de los programas presentados para las elecciones generales y autonómicas. Al igual que en el caso anterior, se ha decidido unirlas puesto que la estructura es la misma y solamente cambia el tipo de elecciones.
- *funcionamiento*: da respuesta a la pregunta "¿Cómo funciona?", dando ejemplos sobre cómo realizar las preguntas al asistente.
- *fallback*: da respuesta a las entradas erróneas y está implementado a nivel lógico de negocio.
- *salida*: para finalizar la conversación se pronuncia "salir", y finaliza la sesión con el asistente virtual.

En la Figura 4.9 se pueden ver los *Intents* desarrollados en el agente Dialogflow del asistente virtual. Como se puede comprobar, los *Intents* definidos anteriormente, se desglosan en otros que se llaman *followup Intent*. Esto se verá con mayor detalle en el siguiente apartado, donde se detalla el flujo conversacional.

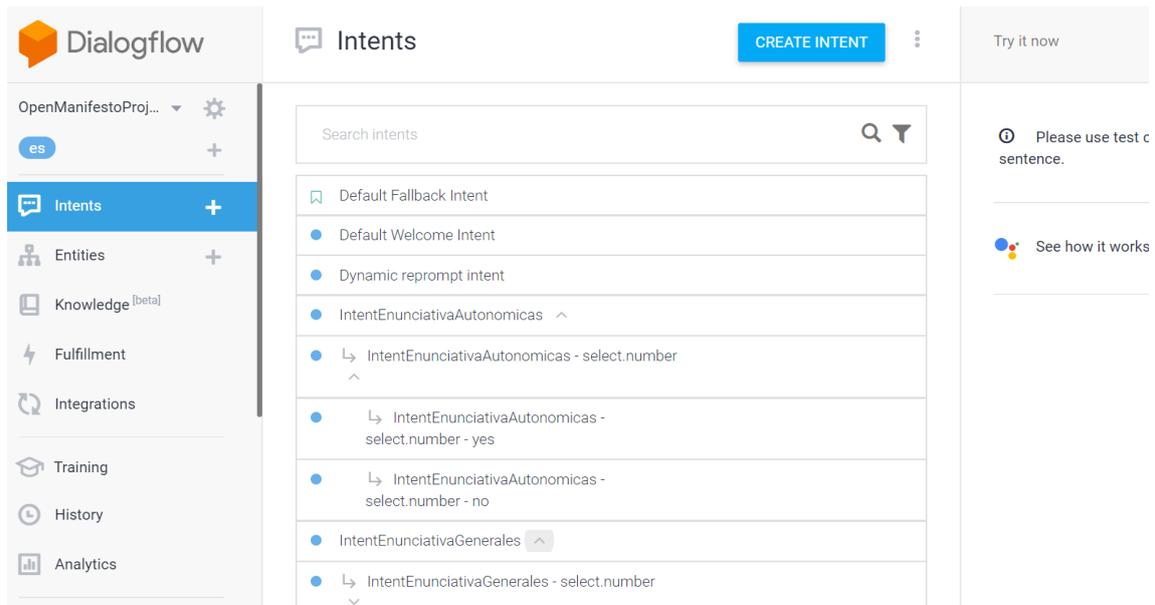


Figura 4.9: *Intents* del asistente virtual

La interfaz web de Dialogflow permite gestionar las frase de entrenamiento de cada *Intent*, introduciendo nuevas de forma manual o a partir del entrenamiento. En la Figura 4.10 se pueden ver algunas de las frases correspondientes al *IntentEnunciativaGeneralesEuropeas*.



Figura 4.10: Frases de entrenamiento del *IntentEnunciativaGeneralesEuropeas*

Por último, destacar que desde los *Intents* también se pueden especificar los parámetros que se desean obtener de las frases de entrada, y a partir de esto el agente Dialogflow construye las estructuras de las frases y las coincidencias con las solicitudes de los usuarios. En la siguiente Figura 4.11 se puede ver un ejemplo de esto.

Action and parameters ^

Enter action name

REQUIRED ?	PARAMETER NAME ?	ENTITY ?	VALUE	IS LIST ?
<input type="checkbox"/>	Partido	@Partido	\$Partido	<input type="checkbox"/>
<input type="checkbox"/>	Tema	@Tema	\$Tema	<input type="checkbox"/>
<input type="checkbox"/>	Elecciones	@Elecciones	\$Elecciones	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

[+ New parameter](#)

Figura 4.11: Parámetros de las frases de entrenamiento

Flujo conversacional

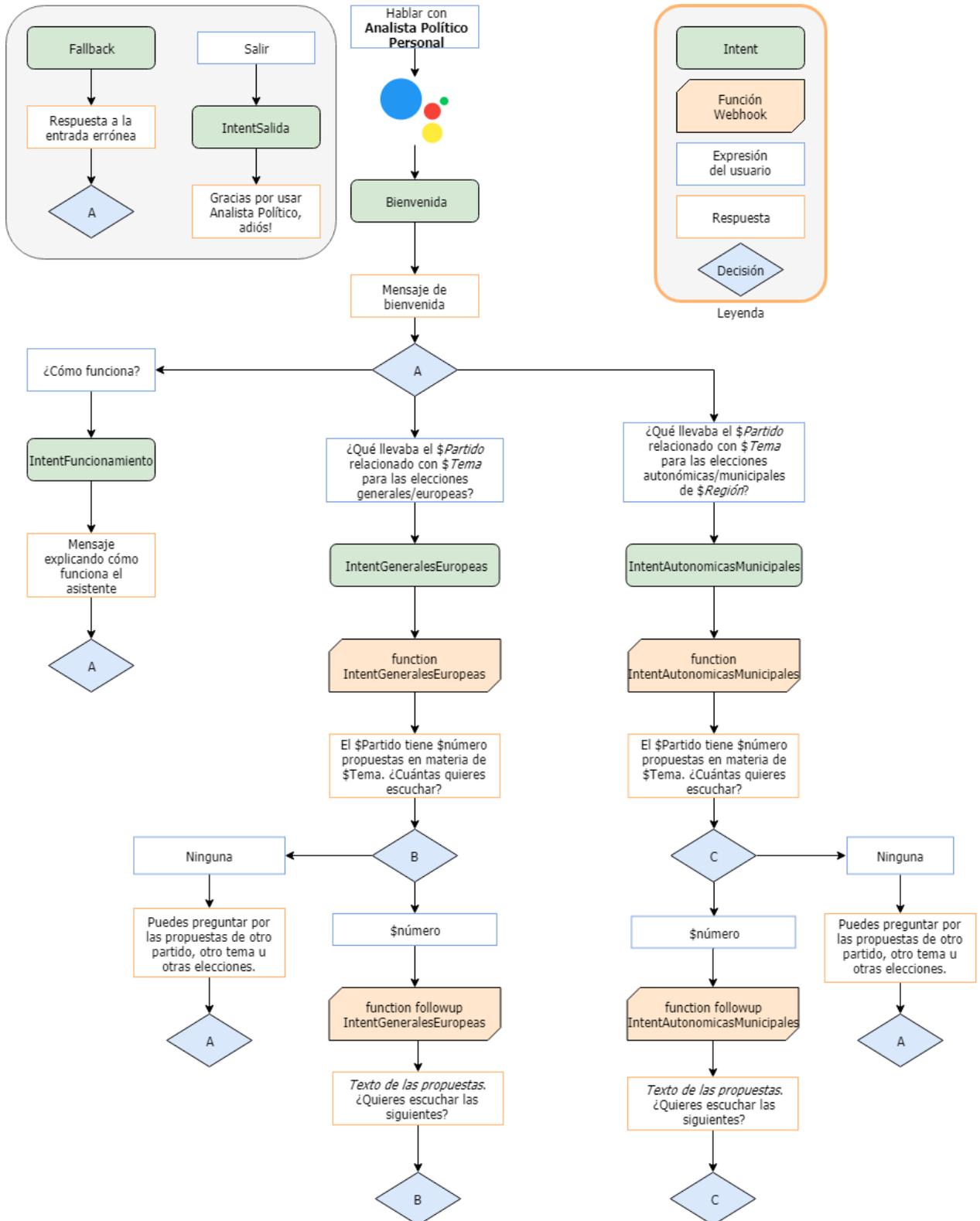


Figura 4.12: Diagrama conversacional del asistente virtual. Fuente: elaboración propia

El diagrama de flujo de la Figura 4.12 refleja de forma genérica todas las posibles interacciones que puede efectuar un usuario del asistente *Analista Político Personal*.

Además de los *Intents* descritos en el apartado anterior, vemos que existen otros llamados *followup Intent*, se podrían traducir como intenciones anidadas. Estas intenciones se crean para generar una conversación algo más natural y para establecer un contexto.

En este trabajo se emplean de la siguiente forma:

1. El usuario realiza una pregunta enunciativa como: "¿Qué propuestas llevaba el *Partido* relacionada con *Tema* para las *Elecciones*?"
2. El asistente responde con el número de propuestas que tiene el partido, relacionada con la temática y correspondiente a las elecciones X. Y pregunta: "¿Cuántas quieres escuchar?"
3. El usuario responde un número o ninguna.
4. El asistente enuncia como máximo dos propuestas del número solicitado, y vuelve a preguntar "¿Quieres oír las siguientes?"
5. El usuario responde sí o no.
6. El asistente enuncia las siguientes propuestas en caso afirmativo o pregunta si quiere oír otras propuestas en caso negativo.

De esta forma se establece un contexto, donde el asistente solicita al usuario si quiere oír las propuestas solicitadas. De esta forma, el usuario conoce con una pregunta enunciativa el número de propuestas, así también se resuelven las preguntas enumerativas, y conociendo el número elige si quiere escuchar las propuestas o no. Se ha tomado esta decisión puesto que las propuestas electorales pueden ser textos largos y es posible que el usuario al escuchar una o dos, ya no desee oír las siguientes.

A nivel lógica de negocio (*Webhook*), es necesario crear un contexto común a la sesión, donde se guardan los valores correspondientes a la primera pregunta efectuada por el usuario, que generalmente son *Tema*, *Partido* y *Elecciones*, y se devuelve el número de propuestas que coinciden con dichos parámetros al consultar en los JSON de los programas estandarizados.

4.4.3. Evaluación y seguimiento

Las herramientas utilizadas para implementar el asistente ofrecen varias opciones para realizar un seguimiento continuo y una evaluación del estado en tiempo real.

Desde el agente Dialogflow del asistente virtual, se pueden ver las frases que efectúan los usuarios al invocar el asistente e interactuar con él. Esto permite realizar lo que se conoce como fase de entrenamiento. Con el entrenamiento del agente se pueden realizar varias acciones, como incluir nuevas frases en los *Intents*, incluir nuevas entidades o sinónimos, eliminar frases de entrada que no aporten al aprendizaje o incluir nuevos *Intents* si se desea dar respuesta a frases que no han coincidido con ninguna intención ya creada. En la Figura 4.13 se puede comprobar cómo automáticamente Dialogflow le asigna un *Intent* a las entradas coincidentes con alguno de los creados. Si la intención asignada no es la correcta, también permite cambiarlo desde el entrenamiento e incluir dichas frases como nuevas frases de entrenamiento del *Intent* correcto.



Figura 4.13: Entrenamiento del asistente virtual

Desde la consola de Actions on Google, también se puede realizar un seguimiento del uso del asistente, si este está publicado en el directorio de Actions de *Google Assistant*. La consola ofrece varias estadísticas, como las conversaciones que ha establecido en un período de tiempo, los errores que ha tenido, las interacciones que ha establecido o las reseñas que ha recibido el asistente en el directorio. En la Figura 4.14 se puede observar un gráfico de las conversaciones que han establecido los usuarios con *Analista Político Personal* desde su publicación en el directorio.

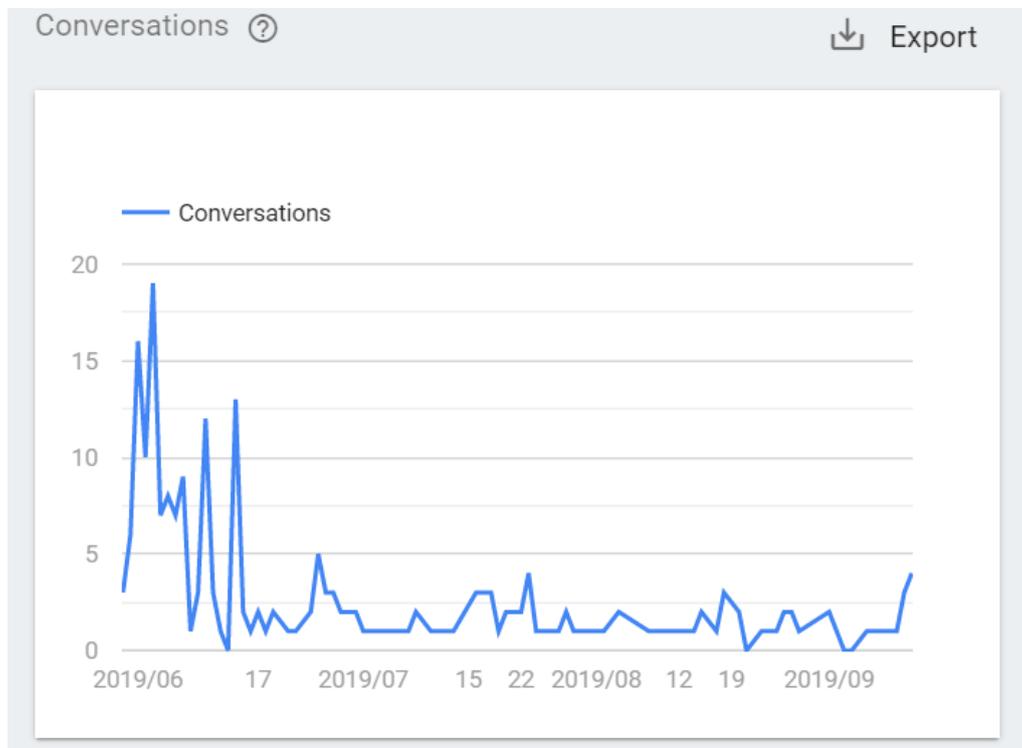


Figura 4.14: Estadísticas de uso de *Analista Político Personal*

CAPÍTULO 5

Análisis de resultados

En este capítulo se exponen los resultados principales obtenidos al realizar este proyecto. Además, se dedica una sección a la campaña realizada en las redes sociales para dar a conocer la iniciativa *Open Manifesto Project* y el asistente virtual.

5.1 Resultados principales

Durante la realización de este trabajo fin de máster, se han obtenido varios resultados. El primero de ellos es el diseño del estándar abierto para programas electorales, llamado *Open Manifesto* y definido en JSON Schema. Este estándar está publicado en el repositorio público del proyecto [39]. Está sujeto a la evaluación pública y abierto a la participación de partidos políticos, agentes sociales o ciudadanía en general.

Además, el grupo *Open Manifesto Project* junto a CIECODE, ha estandarizado 76 programas electorales de los partidos que obtuvieron representación en las elecciones generales, europeas y de algunas comunidades autónomas. Estos programas se puede encontrar también en el repositorio del proyecto. De esta forma, los ciudadanos y ciudadanas podrán consultar, analizar y comparar los programas estandarizados o desarrollar sus aplicaciones haciendo uso del formato *Open Manifesto*.

Otro de los resultados obtenidos es el apoyo de organizaciones a la iniciativa *Open Manifesto Project*, donde ha recibido el respaldo público de organizaciones como OXFAM Intermón¹ o Greenpeace². Se pueden ver las organizaciones que han mostrado su apoyo a la iniciativa en la página web [9].

Por otra parte, lo que comenzó siendo un prototipo ha pasado a producción. El asistente virtual *Analista Político Personal* desarrollado en este trabajo, ha sido publicado en el directorio de Actions de Google para ser accedido por los ciudadanos y ciudadanas que deseen consultar las propuestas electorales de los partidos políticos. En la Figura 5.1 se puede ver el proyecto en el directorio de acciones accesible desde [54]

¹<https://www.oxfamintermon.org/>

²<https://es.greenpeace.org/es/>



Figura 5.1: *Analista Político Personal* en el directorio de Actions

5.2 Campaña social

Para promover y dar a conocer la iniciativa, se realizó una campaña en las redes sociales por Twitter³ e Instagram⁴. También se realizó un vídeo promocional del asistente *Analista Político Personal* que se puede ver en [55].

Los resultados de la campaña, realizada por Odisei⁵ del 3 al 16 de junio de 2019, se pueden ver en las figuras 5.2 y 5.3.

TWITTER

- **71 tweets/retweets**
- 203 seguidores
- **180.447 reproducciones** del vídeo promocionado.

Figura 5.2: Resultados de la campaña por Twitter

³Cuenta de Twitter del proyecto: https://twitter.com/Open_Manifesto

⁴Cuenta de Instagram del proyecto: https://www.instagram.com/open_manifesto/

⁵<https://www.odisei.es/proyectos/open-manifesto-project/>

INSTAGRAM

- **10 publicaciones**
- 20 stories
- 183 seguidores
- **18.802 reproducciones** del vídeo promocionado.

Figura 5.3: Resultados de la campaña por Instagram

Los objetivos principales que tenía esta campaña eran:

- Promover el conocimiento del estándar *Open Manifesto* al público general.
- Generar en los partidos políticos la necesidad de publicar sus programas electorales de manera estandarizada.
- Situar *Analista Político Personal* como herramienta básica para la rendición de cuentas, de manera que la ciudadanía asocie este tipo de solución a su ejercicio.

CAPÍTULO 6

Conclusiones

En este trabajo se ha desarrollado un estándar abierto para los programas electorales, llamado *Open Manifesto*, que permite estructurar la información contenida en estos documentos y facilita la consulta automatizada. Para ello, se ha realizado un estudio de estándares abiertos y de programas electorales. Este proceso de aprendizaje ha supuesto la base del desarrollo del estándar.

Además, se ha realizado un diseño centrado en la ciudadanía mediante el cual se han recopilado las necesidades de información de los ciudadanos con respecto a los programas electorales y a las medidas que lo componen. Esta recopilación de datos se realizó en jornadas de trabajo que tuvieron lugar en Madrid, Valencia y Barcelona.

Con esta información, y en base a las metodologías de diseño de estándares abiertos estudiadas, se ha creado y publicado el estándar bajo licencia abierta. Destacar que, una vez creado el estándar, se han estandarizado programas electorales de las formaciones que han conseguido representación política en las distintas elecciones, probando así que el estándar cumple con la especificación definida antes de su desarrollo. Así pues, uno de los objetivos principales del proyecto se ha cumplido.

Otro de los objetivos principales, que era el desarrollo de un prototipo de asistente virtual que facilitara el acceso a la información contenida en los programas electorales estandarizados con *Open Manifesto*, también se ha cumplido. Sin embargo, durante las fases del prototipado evolutivo, se contempló la posibilidad que este pasase a producción, vista la buena acogida que tuvo en las demostraciones realizadas en las jornadas de trabajo. Por este motivo, se publicó el asistente virtual en el directorio de Actions de Google y es accesible por todos los usuarios y usuarias a través de los dispositivos que integren *Google Assistant*. Con lo cual con el asistente, llamado *Analista Político Personal*, se han cumplido más objetivos de los planteados al principio del trabajo.

Para su implementación, también fue necesaria una etapa de estudio de las tecnologías y herramientas que se emplearon para el procesamiento del lenguaje natural y para el diseño de los distintos componentes del asistente virtual. Tras el estudio, se especificaron las preguntas a las que debería dar respuesta y se desarrollaron los distintos componentes formados por el agente Dialogflow, el *Webhook* implementado en una *Google Cloud Function for Firebase* y el sistema informacional, formado por el conjunto de programas estandarizados.

Desde el punto de vista personal, el desarrollo de este trabajo ha supuesto una gran experiencia de aprendizaje en todos los aspectos. En primer lugar, por haber formado parte de un equipo muy implicado en la iniciativa *Open Manifesto Project*, impulsada por la Cátedra de Tecnología Cívica y Empoderamiento en colaboración con CIECODE, y en el desarrollo de soluciones tecnológicas que empoderen a la ciudadanía, como en este caso, que facilita el voto informado y la rendición de cuentas. En segundo lugar, porque durante todo el proceso de desarrollo se ha aprendido a utilizar nuevas tecnologías y se han afianzado conocimientos ya adquiridos en el máster.

En resumen, se han cumplido todos los objetivos planteados al principio del trabajo, tanto los principales como específicos, y se ha desarrollado el estándar abierto y el asistente virtual.

6.1 Posibles ampliaciones

Tras realizar el diseño del estándar *Open Manifesto* e implementar *Analista Político Personal*, se han identificado las siguientes mejoras o ampliaciones:

- Adaptación del estándar a un formato para estructurar la información en Internet y trabajar por su inclusión como un esquema de schema.org.
- Trabajar para que el estándar sea adoptado por algún partido político en próximas elecciones.
- Ampliar la funcionalidad del asistente virtual para permitir realizar preguntas sobre propuestas electorales que cumplan algún atributo de los definidos en el estándar. Destacar que para implementar esta mejora también es necesario disponer de dichos atributos y eso solo será posible si el estándar es adoptado.
- Integrar el asistente virtual a otras plataformas como Alexa, Telegram o Twitter.

6.2 Apariciones en prensa y reconocimientos

La iniciativa *Open Manifesto* y el asistente *Analista Político Personal* han sido referenciados en varios artículos y noticias, que se enumeran a continuación:

- Artículo de Europa Press publicado el 4 de junio de 2019: [56]
- Noticia de El Periódico de Aquí publicada el 4 de junio de 2019: [57]
- Artículo de ExtraDigital.es publicado el 12 de junio de 2019: [58]
- Noticia del periódico Valencia Plaza publicada el 3 de junio de 2019: [59]
- Noticia de Las Provincias publicada el 4 de junio de 2019: [60]
- Noticia de El Periódico publicada el 26 de junio de 2019 y actualizada el 23 de julio: [61]
- Artículo de Periodismo Ciudadano publicado el 27 de junio de 2019: [62]
- Artículo en la revista Compromiso Empresarial publicado el 27 de agosto de 2019: [63]

Además, el asistente implementado en este proyecto ha recibido el primer premio en el concurso de iniciativas europeas *You4EU* llamado *Citizen Participation 2.0*. Se puede ver el resultado de dicho concurso en [64].

Por último, *Open Manifesto Project* ha sido incluido en la *appteca* [65] de la plataforma *apps4citizens*, puesto que dicha plataforma promueve el uso de las aplicaciones como un instrumento tecnológico al servicio de la ciudadanía.

Bibliografía

- [1] EpData. Principales Problemas Para Los Españoles, dirección: <https://www.epdata.es/datos/principales-problemas-espanoles-cis/45> (visitado 11 de ago. de 2019).
- [2] Centro de Investigaciones Sociológicas. Three Principal Problems, dirección: http://www.cis.es/cis/export/sites/default/-Archivos/Indicadores/documentos_html/TresProblemas.html (visitado 11 de ago. de 2019).
- [3] Daniel Kaufmann, Aart Kraay y Massimo Mastruzzi, «The Worldwide Governance Indicators: Methodology and Analytical Issues», pág. 31,
- [4] Rosa Nonell, *Transparencia y buen gobierno: la rendición de cuentas (accountability) en una sociedad avanzada : el posicionamiento de las universidades*. Icaria, 2002, ISBN: 978-84-7426-616-0. dirección: <https://dialnet.unirioja.es/servlet/libro?codigo=138470> (visitado 10 de ago. de 2019).
- [5] Andreas Schedler, Larry Jay Diamond y Marc F. Plattner, *The Self-Restraining State: Power and Accountability in New Democracies*. Lynne Rienner Publishers, 1999, 412 págs., ISBN: 978-1-55587-774-3.
- [6] Andreas Schedler, «¿Qué es la rendición de cuentas?», pág. 47,
- [7] Diego Álvarez Sánchez, «Análisis comparativo y propuesta metodológica para el análisis del contenido de programas electorales para estimar la posición política en relación al Desarrollo Sostenible», 22 de jun. de 2015. dirección: <https://riunet.upv.es/handle/10251/52032> (visitado 4 de ago. de 2019).
- [8] Tribunal de Cuentas. Rendición de Cuentas de Los Partidos Políticos, dirección: <https://sede.tcu.es/tribunal-de-cuentas/es/sede-electronica/GRCuentas/PartidosPoliticicos/> (visitado 10 de ago. de 2019).
- [9] Open Manifesto Project. Open Manifesto Project, dirección: <https://openmanifestoproject.org/> (visitado 4 de ago. de 2019).
- [10] ISO. Standards in Our World, dirección: https://www.iso.org/sites/ConsumersStandards/1_standards.html#section1_1 (visitado 18 de ago. de 2019).
- [11] RAE. Definición de estándar abierto - Diccionario del español jurídico - RAE, dirección: <https://dej.rae.es/lema/est%C3%A1ndar-abierto> (visitado 18 de ago. de 2019).
- [12] FSFE. Estándares Abiertos, dirección: <https://fsfe.org/activities/os/os.html> (visitado 18 de ago. de 2019).
- [13] —, Estándares Abiertos - Definición, dirección: <https://fsfe.org/activities/os/def.html> (visitado 18 de ago. de 2019).
- [14] Comisión Europea IDABC. Documentation on the European Interoperability Framework, dirección: <https://ec.europa.eu/idabc/en/document/3473/5585.html#finalEIF> (visitado 18 de ago. de 2019).

- [15] UK Government. Open Standards principles, dirección: <https://www.gov.uk/government/publications/open-standards-principles/open-standards-principles> (visitado 18 de ago. de 2019).
- [16] Boletín Oficial del Estado. Regulación El Esquema Nacional de Interoperabilidad En El Ámbito de La Administración Electrónica - Documento BOE-A-2010-1331, dirección: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2010-1331 (visitado 18 de ago. de 2019).
- [17] —, Catálogo de Estándares-2012-13501, dirección: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2012-13501 (visitado 18 de ago. de 2019).
- [18] Juan Jesús Velasco. España, El País Que Apuesta Por Los Estándares Abiertos Pero Sólo Usa PDF, dirección: https://www.eldiario.es/turing/software_libre/estandar-documentos-Administraciones-Publicas-Espana_0_311469935.html (visitado 19 de ago. de 2019).
- [19] Open Contracting Partnership. Open Contracting Partnership, dirección: <https://www.open-contracting.org/?lang=es> (visitado 19 de ago. de 2019).
- [20] —, Getting Started — Open Contracting Data Standard 1.1.4 Documentation, dirección: <https://standard.open-contracting.org/latest/en/getting-started/> (visitado 19 de ago. de 2019).
- [21] —, The Open Contracting Journey, dirección: <https://www.open-contracting.org/implement/#/> (visitado 19 de ago. de 2019).
- [22] OpenStand. OpenStand: The Modern Paradigm for Standards, dirección: <https://open-stand.org/> (visitado 18 de ago. de 2019).
- [23] Open Source Initiative. The Open Source Definition (Annotated), dirección: <https://opensource.org/osd-annotated> (visitado 19 de ago. de 2019).
- [24] Opensource. What are open standards?, dirección: <https://opensource.com/resources/what-are-open-standards> (visitado 18 de ago. de 2019).
- [25] International Telecommunications Union (ITU). Definition of "Open Standards", dirección: <https://www.itu.int/en/ITU-T/ipr/Pages/open.aspx> (visitado 18 de ago. de 2019).
- [26] Open Data Institute. Creating impact with open standards, dirección: <https://standards.theodi.org/creating-impact/getting-started/> (visitado 10 de ago. de 2019).
- [27] —, How open standards are developed, dirección: <https://standards.theodi.org/introduction/how-open-standards-are-developed/> (visitado 20 de ago. de 2019).
- [28] Irene Benavent. Este año haremos más preguntas a nuestros asistentes de voz virtuales, dirección: https://cronicaglobal.elespanol.com/vida/haremos-preguntas-asistentes-de-voz-virtuales_212445_102.html (visitado 13 de ago. de 2019).
- [29] Juan F. Samaniego. ¿OK, Google? La carrera de los asistentes de voz se pondrá interesante en 2019, dirección: <https://www.nobbot.com/futuro/asistentes-virtuales-2019/> (visitado 13 de ago. de 2019).
- [30] Asociación para la Investigación de Medios de Comunicación. La 1ª Ola del EGM se estrena con un universo más amplio y novedades en su cuestionario, dirección: <https://www.aimc.es/blog/la-1a-ola-del-egm-se-estrena-universo-mas-amplio-novedades-cuestionario/> (visitado 13 de ago. de 2019).

- [31] Pedro Jiménez Martín y Jesús Sánchez Allende, «De Eliza a Siri: La evolución», *Tecnología y desarrollo*, vol. 13, n.º 0, pág. 28, 5 de feb. de 2015, ISSN: 1696-8085. dirección: https://revistas.uax.es/index.php/tec_des/article/view/616 (visitado 31 de ago. de 2019).
- [32] J. Kiseleva, K. Williams, J. Jiang, A. Hassan Awadallah, A. C. Crook, I. Zitouni y T. Anastasakos, «Understanding User Satisfaction with Intelligent Assistants», en *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, (Carrboro, North Carolina, USA), ép. CHIIR '16, New York, NY, USA: ACM, 2016, págs. 121-130, ISBN: 978-1-4503-3751-9. DOI: [10.1145/2854946.2854961](https://doi.org/10.1145/2854946.2854961). dirección: <http://doi.acm.org/10.1145/2854946.2854961> (visitado 31 de ago. de 2019).
- [33] Cristiano Castelfranchi, «Modelling Social Action for AI Agents», *Artificial Intelligence, Artificial Intelligence 40 Years Later*, vol. 103, n.º 1, págs. 157-182, 1 de ago. de 1998, ISSN: 0004-3702. DOI: [10.1016/S0004-3702\(98\)00056-3](https://doi.org/10.1016/S0004-3702(98)00056-3). dirección: <http://www.sciencedirect.com/science/article/pii/S0004370298000563> (visitado 31 de ago. de 2019).
- [34] GitHub. Build software better, together, dirección: <https://github.com> (visitado 4 de ago. de 2019).
- [35] Git. Git, dirección: <https://git-scm.com/> (visitado 1 de sep. de 2019).
- [36] JSON Schema. JSON Schema, dirección: <http://json-schema.org/> (visitado 4 de ago. de 2019).
- [37] JSON. JSON, dirección: <https://www.json.org/> (visitado 10 de sep. de 2019).
- [38] REST API Tutorial. JSON Schema, dirección: <https://restfulapi.net/json-schema/> (visitado 12 de ago. de 2019).
- [39] Open Manifesto Project. Open Manifesto Project, dirección: <https://github.com/open-manifesto-project> (visitado 4 de ago. de 2019).
- [40] Schema.org. Schema.Org, dirección: <https://schema.org/> (visitado 4 de ago. de 2019).
- [41] JSON-LD. JSON for Linking Data, dirección: <https://json-ld.org/> (visitado 12 de ago. de 2019).
- [42] World Wide Web Consortium. RDF - Semantic Web Standards, dirección: <https://www.w3.org/RDF/> (visitado 12 de sep. de 2019).
- [43] Python.org. Welcome to Python.org, dirección: <https://www.python.org/> (visitado 12 de ago. de 2019).
- [44] Sebastián Pallarés, Diego Álvarez y Pablo Martín, *Estándar Open Manifesto Para Los Programas Electorales: Open-Manifesto-Project/Open-Manifesto*, Open Manifesto Project, 21 de jul. de 2019. dirección: <https://github.com/open-manifesto-project/Open-Manifesto> (visitado 4 de ago. de 2019).
- [45] Dialogflow. Dialogflow, dirección: <https://dialogflow.com/> (visitado 4 de ago. de 2019).
- [46] Google Cloud. Dialogflow, dirección: <https://cloud.google.com/dialogflow/> (visitado 14 de ago. de 2019).
- [47] Google Developers. Actions on Google, dirección: <https://developers.google.com/actions/> (visitado 14 de ago. de 2019).
- [48] Google. Asistente de Google, dirección: <https://assistant.google.com/explore/> (visitado 14 de ago. de 2019).

- [49] Firebase. Firebase, dirección: <https://firebase.google.com/> (visitado 14 de ago. de 2019).
- [50] —, Referencia de Firebase CLI, dirección: <https://firebase.google.com/docs/cli> (visitado 14 de ago. de 2019).
- [51] Google Cloud. Servicios de cloud computing, dirección: <https://cloud.google.com/?hl=es> (visitado 14 de ago. de 2019).
- [52] F. de Node.js. Node.js, dirección: <https://nodejs.org/es/> (visitado 14 de ago. de 2019).
- [53] Google Developers. Build fulfillment with the Actions on Google Node.js client library | Actions on Google, dirección: <https://developers.google.com/actions/reference/nodejsv2/overview> (visitado 15 de ago. de 2019).
- [54] Open Manifiesto Project. Analista Político Personal | Asistente de Google, dirección: <https://assistant.google.com/services/a/uid/000000b80ee7bf8d?hl=es-ES> (visitado 4 de ago. de 2019).
- [55] O. M. Project. Analista Político Personal by Open Manifiesto Project - YouTube, dirección: https://www.youtube.com/watch?v=pB1XV_OF6MI (visitado 4 de ago. de 2019).
- [56] Europa Press. Idean una solución para evitar que las promesas electorales caigan en el olvido, dirección: <https://www.europapress.es/comunitat-valenciana/noticia-idean-solucion-evitar-promesas-electorales-caigan-olvido-20190604122602.html> (visitado 14 de ago. de 2019).
- [57] EFE. Idean una solución para comparar promesas electorales con su cumplimiento, dirección: <https://www.elperiodicodeaqui.com/epda-noticias/idean-una-solucion-para-comparar-promesas-electorales-con-su-cumplimiento/183761> (visitado 14 de ago. de 2019).
- [58] Extra Digital. La Tecnología Open Manifiesto Revoluciona La Política, dirección: <http://www.extradigital.es/la-tecnologia-open-manifiesto-revoluciona-la-politica/> (visitado 14 de ago. de 2019).
- [59] Valencia Plaza. La UPV y la Diputación crean un modelo de programa electoral para detectar los incumplimientos, dirección: <https://valenciaplaza.com/la-upv-diputacion-crea-un-modelo-de-programa-electoral-para-detectar-los-incumplimientos> (visitado 4 de ago. de 2019).
- [60] Las Provincias. Open Manifiesto | La Fórmula Valenciana Para Comparar Promesas Electorales Con Su Cumplimiento, dirección: <https://www.lasprovincias.es/politica/open-manifiesto-promesas-electorales-comprara-20190604133819-nt.html> (visitado 14 de ago. de 2019).
- [61] Gabriela Coba. Open Manifiesto Project, Una Iniciativa Para El Empoderamiento, dirección: <https://www.elperiodico.com/es/apps-para-el-ciudadano-comprometido/20190626/open-manifiesto-project-una-iniciativa-para-el-empoderamiento-7523836> (visitado 4 de ago. de 2019).
- [62] Paula Gonzalo. Open Manifiesto: participación ciudadana en favor del voto informado, dirección: <https://www.periodismociudadano.com/open-manifiesto-participacion-ciudadana-en-favor-del-voto-informado/> (visitado 4 de ago. de 2019).
- [63] Rafa Camacho. Transparencia, el juguete roto de las elecciones (y los pactos), dirección: <https://www.compromisoempresarial.com/transparencia/2019/08/transparencia-el-juguete-roto-de-las-elecciones-y-los-pactos/> (visitado 12 de sep. de 2019).

-
- [64] Institute Alternative. Results of the Contest You4EU – Citizen Participation 2.0, dirección: <https://institut-alternativa.org/en/results-of-the-contest-you4eu-citizen-participation-2-0/> (visitado 14 de ago. de 2019).
- [65] apps4citizens. Open Manifesto Project, empoderamiento electoral por y para los ciudadanos, dirección: <http://apoteca.apps4citizens.org/open-manifesto-project-empoderamiento-electoral-por-y-para-los-ciudadanos/> (visitado 4 de ago. de 2019).

APÉNDICE A

Estándar Open Manifesto

El estándar ha sido diseñado utilizando JSON Schema y está abierto en el repositorio público del proyecto¹. Se trata de la versión 1.1 publicada el 12/08/2019.

```
1 {
2   "$schema": "http://json-schema.org/draft-06/schema#",
3   "$ref": "#/definitions/ManifestoSchema",
4   "definitions": {
5     "ManifestoSchema": {
6       "type": "object",
7       "additionalProperties": false,
8       "properties": {
9         "politicalParty": {
10          "description": "Name of the political party",
11          "type": "string"
12        },
13        "title": {
14          "description": "Title of the electoral program",
15          "type": "string"
16        },
17        "publicationDate": {
18          "description": "Publication date of the electoral program",
19          "type": "string"
20        },
21        "electionDate": {
22          "description": "Election date of the electoral program",
23          "type": "string"
24        },
25        "electionsType": {
26          "description": "Type of election",
27          "type": "string"
28        },
29        "geographicalArea": {
30          "description": "Geographic area that covers the election
31          process",
32          "type": "string"
33        },
34        "standardVersion": {
35          "description": "Version of the standard used to structure
36          the manifesto",
37          "type": "string"
38        },
39        "URI": {
```

¹<https://github.com/open-manifesto-project/Open-Manifesto/blob/master/Standard/OpenManifestoV1.1.schema.json>

```

38         "description": "URI or URL that links to the electoral
39             program",
40         "type": "string",
41         "format": "uri"
42     },
43     "createdBy": {
44         "type": "string"
45     },
46     "pages": {
47         "type": "integer"
48     },
49     "numProposals": {
50         "description": "Number of proposals included in the
51             manifesto",
52         "type": "integer"
53     },
54     "proposals": {
55         "type": "array",
56         "items": {
57             "$ref": "#/definitions/Proposal"
58         }
59     },
60     "required": [
61         "politicalParty",
62         "title",
63         "publicationDate",
64         "electionDate",
65         "electionsType",
66         "geographicalArea",
67         "standardVersion",
68         "URI",
69         "createdBy",
70         "pages",
71         "numProposals",
72         "proposals"
73     ],
74     "title": "ManifiestoSchema"
75 },
76 "Proposal": {
77     "type": "object",
78     "additionalProperties": false,
79     "properties": {
80         "id": {
81             "type": "integer"
82         },
83         "body": {
84             "description": "Body of the proposal",
85             "type": "string"
86         },
87         "topics": {
88             "description": "Set of topics defined by the group Open Manifiesto Project
89             .",
90             "type": "array",
91             "items": {
92                 "type": "string"
93             }
94         },
95         "tags": {
96             "description": "Set of proposal tags/labels",
97             "type": "array",
98             "items": {
99                 "type": "string"
100             }
101         }
102     }
103 }

```

```
99     },
100     "priority": {
101     "decription": "Priority of the proposal Low/Medium/High/NA",
102     "type": "string"
103     },
104     },
105     "budget": {
106     "decription": "If the proposal includes economic value estimation",
107     "type": "boolean"
108     },
109     "nonNegotiable": {
110     "decription": "If the proposal is not negotiable",
111     "type": "boolean"
112     },
113     "agents": {
114     "decription": "Set of subjects or organizations that are included or
115     mentioned in the proposal",
116     "type": "array",
117     "items": {
118     "type": "string"
119     }
120     },
121     "required": [
122     "id",
123     "body",
124     "topic",
125     "tags",
126     "priority",
127     "budget",
128     "nonNegotiable",
129     "agents"
130     ],
131     "title": "Proposal"
132 }
133 }
134 }
```

APÉNDICE B

Talleres

Como se ha especificado anteriormente, para el desarrollo del estándar se ha seguido un diseño centrado en la ciudadanía. Para ello, se realizaron talleres en Madrid, Valencia y Barcelona. Estas dinámicas tenían como objetivo determinar las necesidades de información relacionadas con las propuestas electorales y establecer las características formales que han de satisfacer los programas electorales para favorecer el ejercicio del derecho a la rendición de cuentas.

B.1 Taller en Madrid

La jornada de trabajo tenía como nombre **‘Asistentes virtuales y programas electorales’**. Fue organizada por CIECODE y la Cátedra de Tecnología Cívica y Empoderamiento UPV-Diputación de Valencia y tuvo lugar en el edificio Impact Hub Alameda (Sala San Francisco) el jueves 28 de marzo de 2019.



Figura B.1: Fotografía del taller de Madrid

B.2 Taller en Valencia

La jornada de trabajo también tuvo como nombre **‘Asistentes virtuales y programas electorales’** y se realizó el sábado 6 de Abril de 2019. Fue organizada por CIECODE y la Cátedra de Tecnología Cívica y Empoderamiento UPV-Diputación de Valencia y tuvo lugar en el edificio Las Naves de Valencia.

Se puede consultar la convocatoria al taller en <https://www.lasnaves.com/estrategias-ciudad/asistentes-virtuales-y-programas-electorales/?lang=es> (visitado 9 de ago. de 2019)

En la Figura B.2 se puede ver una fotografía de la jornada.



Figura B.2: Fotografía del taller de Valencia. Fuente: Valencia Plaza[59]

B.3 Taller en Barcelona

La jornada de trabajo tuvo como nombre **‘Jornada de Presentación y Debate del Open Manifesto Project’** y se realizó el jueves 11 de Abril de 2019. Fue organizada por CIECODE y Cátedra Tecnología Cívica y Empoderamiento UPV-Diputación de Valencia y tuvo lugar en el Espai Societat Oberta.

Se puede consultar la convocatoria al taller en la siguiente dirección <https://www.espaisocietatoberta.org/en/agenda/presentacion-open-manifesto-project/> (visitado 9 de ago. de 2019)

En la Figura B.3 se puede ver una fotografía de la jornada tomada durante la demostración de Analista Político Personal.



Figura B.3: Fotografía del taller de Barcelona