



UNIVERSITAT  
POLITÀCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# OCCVotingBOT

## Desarrollo de un sistema de votación telemático basado en un bot

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

*Autor:* Alejandro González Pérez

*Tutor:* Diego Álvarez Sánchez

Curso 2015-2016



*A mis padres, que lo son todo para mí.*



# Agradecimientos

---

Quiero agradecer muy especialmente a Diego Álvarez, tutor de este proyecto, su energía, inspiración y su capacidad para no rendirse pese a todo y pese a todos.

También al Ayuntamiento de Quart de Poblet, especialmente a Juan Medina, Concejal de Gobierno Abierto y Comunicación, Planificación Urbanística y Sostenibilidad y Empleo y Promoción Económica, por haber apostado por este proyecto desde el primer instante.

A Hub Cívico, por darme la oportunidad de contribuir a crear una sociedad mejor.



---

# Resum

En l'actualitat, el desenvolupament d'un nou paradigma de governança basat en el Govern Obert està introduint cotes majors de participació ciutadana. Així, trobem diferents processos promoguts des de diverses institucions, com els realitzats pels Ajuntaments de Madrid, Barcelona o València; des de la ciutadania, com el procés participatiu dinamitzat per l'organització sense ànim de lucre OpenKratio per a l'elaboració de l'Ordenança de Transparència de l'Ajuntament de Sevilla; o, promoguts des dels partits polítics, com els processos de primàries realitzats per partits com Equo o Podemos.

La característica principal de tots aquests processos és la incorporació de les tecnologies de la informació i la comunicació en el seu desenvolupament. En aquest context, trobem que avui dia existeixen diversos productes al mercat que permeten a qualsevol organització realitzar consultes o votacions. Mitjançant els quals, els votants emeten els seus vots a través d'un dispositiu remot (ordinador personal, telèfon mòbil, etc.) connectat a una urna o conjunt d'urnes mitjançant una xarxa telemàtica.

No obstant això, tot i la seva utilitat, persisteixen les dificultats associades al vot telemàtic, com són les relacionades amb la identificació dels votants, la dificultat d'accés o la complexitat dels processos.

D'altra banda, els sistemes de missatgeria instantània han demostrat causar veritable furor entre els usuaris de dispositius mòbils, que al seu torn representen a la majoria d'usuaris d'internet al món. Aquest èxit ve derivat de la comoditat, facilitat d'ús i avantatges d'utilitzar aquests sistemes de missatgeria en comparació amb el correu electrònic o altres mètodes de comunicació clàssics.

El projecte *OCCvotingBOT* té com a objectiu el desenvolupament d'un sistema de votacions telemàtiques mitjançant *bots* integrats en sistemes de missatgeria instantània. L'objectiu és aconseguir un sistema escalable que permeti als diferents municipis d'una província poder engegar consultes ciutadanes a través d'aquests *bots* de forma personalitzada, senzilla i fiable. Per aconseguir l'objectiu proposat es planteja el desenvolupament, implantació i avaluació d'un prototip per a un municipi concret i, a partir d'aquí, abordar el desenvolupament del sistema definitiu.

**Paraules clau:** votació, bot, participació, govern obert, empoderament, ciutadania, municipis

---

## Resumen

En la actualidad, el desarrollo de un nuevo paradigma de gobernanza basado en el Gobierno Abierto está introduciendo mayores cotas de participación de la ciudadanía. Así, encontramos diferentes procesos promovidos desde diversas instituciones, como los realizados por los Ayuntamientos de Madrid, Barcelona o Valencia; desde la ciudadanía, como el proceso participativo dinamizado por la organización sin ánimo de lucro OpenKratio para en la elaboración de la Ordenanza de Transparencia del Ayuntamiento de Sevilla; o, promovidos desde los partidos políticos, como los procesos de primarias realizados por partidos como Equo o Podemos.

La característica principal de todos estos procesos es la incorporación de las tecnologías de la información y la comunicación en su desarrollo. En este contexto, encontramos que hoy en día existen diversos productos en el mercado que permiten a cualquier organización realizar consultas o votaciones. Mediante estos, los votantes emiten sus votos a través de un dispositivo remoto (ordenador personal, teléfono móvil, etc.) conectado a una urna o conjunto de urnas mediante una red telemática.

Sin embargo, a pesar de su utilidad persisten las dificultades asociadas al voto telemático como son las relacionadas con la identificación de los votantes, la dificultad de acceso o la complejidad de los procesos.

Por otro lado, los sistemas de mensajería instantánea han demostrado causar verdadero furor entre los usuarios de dispositivos móviles, que a su vez representan a la mayoría de usuarios de internet en el mundo. Éste éxito viene derivado de la comodidad, facilidad de uso y ventajas de utilizar estos sistemas de mensajería en comparación con el correo electrónico u otros métodos de comunicación clásicos.

El proyecto *OCCvotingBOT* tiene como objetivo el desarrollo de un sistema de votaciones telemáticas mediante *bots* integrados en sistemas de mensajería instantánea. El objetivo es lograr un sistema escalable que permita a los diferentes municipios de una provincia poder poner en marcha consultas ciudadanas a través de estos *bots* de forma personalizada, sencilla y fiable. Para alcanzar el objetivo propuesto se plantea el desarrollo, implantación y evaluación de un prototipo para un municipio concreto y, a partir de ahí, abordar el desarrollo del sistema definitivo.

**Palabras clave:** votación, bot, participación, gobierno abierto, empoderamiento, ciudadanía, municipios

---

# Abstract

Nowadays, the development of a new paradigm of governance based on Open Government is introducing higher levels of citizen participation. Therefore, different processes are promoted by numerous institutions, such as those conducted by the City Councils of Madrid, Barcelona or Valencia; from citizens, as the participation process boosted by the non-profit organization OpenKratio for the arrangement of the Transparency Bylaw of the City of Seville; or promoted by political parties, as primaries performed by parties like Equo or Podemos.

The main feature of all these processes it is based on the incorporation of the information and communications technologies for their development. In this context, we currently find several products on the market that allow any organization to conduct polls or voting. Through these, voters cast their votes over a remote device (personal computer, phones, etc.) connected to an ballot box or boxes by means of a computer network.

Despite its usefulness, the difficulties associated with electronic voting (e.g.)those related to the identification of voters or the accessibility problems or process complexity, prevail.

On the other hand, instant messaging have been shown to cause real expectation among users of mobile devices, which in turn represent the majority of Internet users in the world. This success is derived from the comfort, ease of use and advantages of using these messaging systems compared with email or other classical communication methods.

The *OCCvotingBOT* project target is to develop a telematic voting system by means of *bots* integrated into messaging systems. The aim is to achieve a scalable system that allows different municipalities to launch public, custom, easy and reliably polls through these *bots*. To achieve this, we propose the development, implementation and evaluation of a prototype for a particular municipality and, from there, address the development of a final system.

**Key words:** voting, bot, participation, open government, empowerment, citizenship, municipalities

---



# Índice general

---

<b>Índice general</b>	XI
<b>Índice de figuras</b>	XIII
<b>Índice de tablas</b>	XIV
<b>Índice de códigos</b>	XIV
<hr/>	
<b>1 Introducción</b>	<b>1</b>
1.1 Justificación	2
1.2 Alcance	4
1.3 Contexto	5
1.3.1 Ayuntamiento de Quart de Poblet	5
1.3.2 Hub Cívico	6
1.4 Metodología	7
1.5 Estructura de la memoria	9
<b>2 Estado del arte</b>	<b>11</b>
2.1 El voto telemático	11
2.2 Sistemas de votación electrónica	12
2.2.1 Clasificación	12
2.2.2 Problemas asociados al voto electrónico	14
2.3 Votación en línea	17
2.4 Bots	18
2.5 Síntesis	19
<b>3 Solución propuesta</b>	<b>21</b>
3.1 OCCVotingBOT	21
3.2 Identificación de riesgos y limitaciones	23
3.2.1 Riesgos	23
3.2.2 Limitaciones	26
<b>4 Prototipado</b>	<b>27</b>
4.1 Especificación de Requerimientos	27
4.1.1 Requerimientos funcionales	27
4.1.2 Requerimientos no funcionales	29
4.2 Diseño y construcción	30
4.2.1 Diseño	30
4.2.2 Herramientas	40
4.2.3 Arquitectura	46
4.3 Implantación del prototipo	54
4.3.1 Fase 1	54
4.3.2 Fase 2	54
4.3.3 Fase 3	55
4.4 Evaluación	56

---

4.4.1	Prototipo final . . . . .	56
4.4.2	Análisis de uso . . . . .	59
4.4.3	Análisis de rendimiento . . . . .	60
<b>5</b>	<b>Conclusiones y trabajo futuro</b>	<b>63</b>
5.1	Conclusiones . . . . .	63
5.2	Trabajo futuro . . . . .	64
	<b>Bibliografía</b>	<b>67</b>

---

Apéndices

<b>A</b>	<b>Código implementado</b>	<b>69</b>
<b>B</b>	<b>Repercusión en medios</b>	<b>71</b>

# Índice de figuras

---

1.1	Nota de prensa sobre el proceso participativo de Quart de Poblet. . . . .	6
1.2	Esquema de la metodología implementada. . . . .	8
2.1	Sistema Votomatic. . . . .	13
3.1	Boceto del esquema de funcionamiento de <i>OCCVotingBOT</i> . . . . .	22
4.1	Flujo de usuario del <i>bot</i> . . . . .	35
4.2	Casos de uso. . . . .	36
4.3	Boceto del inicio de la conversación. . . . .	37
4.4	Boceto del mensaje de bienvenida. . . . .	37
4.5	Boceto del inicio de la autenticación. . . . .	38
4.6	Boceto de la primera fase de autenticación. . . . .	38
4.7	Boceto de la segunda fase de autenticación. . . . .	38
4.8	Boceto de la fase de emisión de voto. . . . .	38
4.9	Boceto de la fase de selección de opciones. . . . .	39
4.10	Boceto del inicio del proceso de verificación de voto. . . . .	39
4.11	Boceto del envío del código para verificación. . . . .	39
4.12	Boceto de confirmación de voto verificado. . . . .	39
4.13	Modelo de datos. . . . .	50
4.14	Esquema de integración. . . . .	51
4.15	Flujo del sistema. . . . .	53
4.16	Imagen de la pantalla principal del <i>bot</i> . . . . .	56
4.17	Imagen de la selección de idioma. . . . .	56
4.18	Imagen de la pantalla de bienvenida. . . . .	57
4.19	Imagen del inicio del proceso de registro. . . . .	57
4.20	Imagen de la primera fase de autenticación. . . . .	57
4.21	Imagen de la segunda fase de autenticación. . . . .	57
4.22	Imagen del inicio del proceso de votación. . . . .	58
4.23	Imagen de la emisión de voto. . . . .	58
4.24	Imagen del inicio del proceso de verificación de voto. . . . .	58
4.25	Imagen de la verificación de voto. . . . .	58
4.26	Mensajes recibidos por el <i>bot</i> , correctos vs. erróneos. . . . .	59
4.27	Relación de tiempo consumido por la aplicación. . . . .	60
4.28	Peticiones por minuto. . . . .	61
B.1	Nota de prensa sobre el proceso participativo publicada en el diario Levante. . . . .	71
B.2	<i>Tweet</i> publicado por Pavel Durov, creador de <i>Telegram</i> . . . . .	72
B.3	Nota de prensa sobre el proceso participativo publicada por Europa Press. . . . .	73

B.4	Publicación del grupo de desarrolladores de Telegram. . . . .	74
B.5	Nota de prensa sobre el proceso participativo el el diario Clarin. . .	75
B.6	Nota de prensa sobre el proceso participativo publicada por el me- dio francés <i>Housseniawriting</i> . . . . .	76
B.7	Nota de prensa sobre el proceso participativo publicada por el me- dio francés <i>sputniknews</i> . . . . .	77

## Índice de tablas

---

2.1	Elaboración própia: Comparativa de sistemas de votación en línea. . .	17
-----	---	----

## Índice de códigos

---

4.1	Esquema formato de los comandos. . . . .	31
4.2	Expresión regular del primer caso. . . . .	32
4.3	Expresión regular del segundo caso. . . . .	32
4.4	Expresión regular del tercer caso. . . . .	32
4.5	Expresión regular del cuarto caso. . . . .	32
4.6	Expresión regular del quinto caso. . . . .	32
4.7	Expresión regular del sexto caso. . . . .	33
4.8	Expresión regular del séptimo caso. . . . .	33
4.9	Función de clasificación de texto. . . . .	33

---

---

# CAPÍTULO 1

## Introducción

---

Pese a que algunos antropólogos apuntan la preexistencia de comportamientos democráticos en otros pueblos, (p.ej. el sistema timocrático establecido en la Constitución de Solón), el nacimiento de la democracia debe establecerse en Atenas, durante el siglo de Pericles: el V a.C., desde donde se extendió, con algunas peculiares variaciones, al resto de las ciudades-estado griegas.

Durante aquella época los atenienses no elegían a sus representantes, a diferencia de lo que hoy entendemos como democracia, sino que su participación era personal, con absoluta libertad para hablar en la *ecclesia* (la principal asamblea de la democracia ateniense en la Antigua Grecia), emitiendo su voto a mano alzada tanto en lo concerniente al desarrollo legislativo como para el ejercicio del poder ejecutivo, y la administración de justicia.

En los últimos años se ha venido produciendo una demanda que mucho tiene que ver con la forma en la que se entendía la democracia en la antigua Grecia: la ciudadanía reclama participar más activamente en los procesos de toma de decisiones gubernamentales.

Movimientos como el 15M, surgido en mayo de 2011, dieron énfasis a dichas exigencias. El debate del pueblo, las asambleas de barrio o las comisiones de trabajo son solo algunos de los ejemplos de implicación ciudadana en procesos democráticos que dejó como legado y que, desde entonces hasta hoy, algunos partidos políticos incluyen en sus programas electorales.

En mayor o menor medida la respuesta institucional ha sido positiva, reavivando el paradigma de gobernanza conocido como Gobierno abierto. La idea del gobierno abierto surgió en 1970. Sin embargo, no fue hasta 2009, momento en el que Barack Obama promulgó el Memorando sobre Transparencia y Gobierno Abierto, cuando el concepto se popularizó.

Desde la iniciativa internacional *Open Government Standards* [1] se han concretado tres pilares básicos del Gobierno abierto:

- **Transparencia.** Encargada de fomentar y promover la rendición de cuentas de la Administración ante la ciudadanía así como proporcionar información sobre lo que está realizando y sobre sus planes de actuación.
- **Participación.** Significa que el público puede involucrarse directamente en las discusiones sobre opiniones políticas y en la toma de decisiones, y que

también pueda contribuir con ideas y evidencias que lleven a políticas, leyes y decisiones que mejor sirvan a la sociedad y a los amplios intereses democráticos.

- **Rendición de cuentas.** Derecho ejercido por la ciudadanía para que políticos y funcionarios informen sobre sus decisiones y las justifiquen en público (*answerability*) y, además, se les sancione en caso de que hayan violado sus deberes públicos (*enforcement*) [2].

A estos pilares fundamentales, se empieza a unir un cuarto pilar, que implica a las tecnologías de la información y comunicación [3].

La idea de hacer partícipe a la ciudadanía en los procesos de toma de decisiones guarda cierta similitud con la filosofía del movimiento del software libre promovido por Richard Stallman quien, en 1983, anunció el proyecto *GNU*, con el claro objetivo de promocionar su utilización y desarrollo en todas las áreas de la computación, pero muy particularmente, ayudando a programar, de forma colaborativa, el sistema operativo *GNU* [4]. De la misma manera, el objetivo base del Gobierno abierto radica en que los ciudadanos participen en la creación y mejora de servicios públicos y en el fortalecimiento de la transparencia y el ejercicio de la rendición de cuentas.

De los tres pilares, el presente trabajo se centrará en el desarrollo de la participación directa de la ciudadanía en la toma de decisiones. La idea principal es dotar a los ciudadanos de voz y voto en las decisiones gubernamentales de manera independiente, sin ser necesario que formen parte de la administración pública o de un partido político.

El presente TFG se suma a la incorporación de las tecnologías de la información y la comunicación como un mecanismo para lograr que los ciudadanos participen en mayor número, y de manera más activa, en los procesos democráticos mediante la creación de nuevas formas de participación [5], [6], [7]. Y, de esta forma, reducir la actual pérdida de interés en los procedimientos democráticos y creciente alienación ciudadana de la sociedad y la democracia.

## 1.1 Justificación

---

La sociedad y, concretamente, las nuevas generaciones, vive actualmente una época en la que el tiempo es un bien muy preciado. La globalización y los nuevos entornos de trabajo hacen que resulte difícil disponer de tiempo libre y, por ende, de tiempo suficiente para interesarse por la situación democrática de un país o por los procesos participativos que este ponga en marcha. Un ejemplo de esta situación la tenemos en el pasado referéndum realizado en Reino Unido, donde participaron tres de cada cuatro mayores frente a uno de cada dos jóvenes [8].

Resulta necesario implementar nuevas soluciones tecnológicas que facilite y a la par simplifique las tareas que la ciudadanía debe realizar de cara a tomar parte activa en dichos procesos, permitiéndoles gestionar mejor su tiempo.

Innovar y mejorar los canales de comunicación con la ciudadanía es uno de los retos a los que los Gobiernos y Administraciones han de hacer frente. Hasta

ahora el modelo *web* era la única herramienta disponible, pero tal y como se ha citado en apartados anteriores, las tecnologías de la información hacen posible la mejora de los canales de comunicación, resultando más eficaces, intuitivos y agradables.

En la actualidad, las tecnologías de la información y la comunicación (TIC) han transformado el panorama social. Según datos del Instituto Nacional de Estadística el 64,3 % de la población comprendida entre los 16 y los 74 años de edad utiliza Internet a diario y el 96 % de hogares disponen de un terminal móvil [9]. Debemos, por tanto, aprovechar los medios que nos facilita esta importante revolución tecnológica para adaptarlos y que también forme parte de la transformación democrática.

Llegados a este punto se abre la cuestión de cómo se podría fusionar algo tan popular como las TIC con algo tan antiguo como la democracia. La respuesta no es, en absoluto, sencilla. Aunque es cierto que se ha intentado llevar a cabo en varias ocasiones, y de diversas maneras, los resultados obtenidos han sido, generalmente, poco satisfactorios. Un claro ejemplo son los sistemas de votación telemática, que pese a estar disponibles desde mediados del siglo XX, no han llegado a gozar de la popularidad que podría esperarse. Esto es debido a ciertos factores determinantes como la incomodidad del procedimiento que requiere demasiados pasos; el desinterés del ciudadano obligado a proporcionar un número excesivo de datos para su autenticación o la poca utilidad de los mismos, entre otros.

A todo ello se añaden los problemas derivados de estos sistemas, como son: la seguridad, garantizar el anonimato, trazabilidad del voto, etc.

La persistencia de estas dificultades mantienen vigente el reto de diseñar un sistema de votación cómodo, rápido, sencillo, seguro y ameno, ¿por qué no?, que facilite a la sociedad participar activamente y, por ende, contribuir en la mejora continua de la Democracia.

Tal y como se ha introducido al inicio, el uso de las TIC no resultaría un problema con vistas a ser la base sobre la que se construyan nuevas herramientas de participación: la gran mayoría de ciudadanos hacen ya uso de ellas. Además, la revolución móvil ha permitido que cualquier persona porte un terminal, casi permanentemente conectado a internet, en el bolsillo. Resulta evidente, por tanto, que el camino a seguir es el de aprovechar al máximo posible esta revolución tecnológica.

Cuando se habla de revolución móvil, se hace generalmente referencia al universo de las aplicaciones diseñadas para los *smartphones*, esto es, al software que, instalado en los terminales portables, facilita una funcionalidad concreta. Ejemplo de estas aplicaciones hay muchos, pero los que sin ninguna duda triunfan entre los usuarios son los clientes de mensajería instantánea, pues permiten una comunicación fluida, rápida, sencilla y gratuita con otras personas. Según datos de la Comisión Nacional de los Mercados y la Competencia, España es el país europeo en el que los usuarios de telefonía móvil usan más los servicios de mensajería instantánea independientes como *WhatsApp*, *Telegram*, *Line* u otros similares. Así, un 51,5 % de los usuarios de móvil los usan diariamente y el 83 % de los poseedores de teléfonos inteligentes lo emplean al menos una vez por semana, muy por

encima de otros países como Holanda (67%), Alemania (43%), Portugal (34%), Reino Unido (30%), Bélgica (20%) y Francia (15%).[10]

Sin embargo, las aplicaciones tienen diversos inconvenientes: han de ser instaladas en nuestros terminales y, en ocasiones, los usuarios se encuentran con un sinnúmero de aplicaciones almacenadas en sus dispositivos que únicamente resultan útiles para realizar una tarea muy concreta. Además, las aplicaciones suelen requerir recursos y permisos del dispositivo que, para muchos usuarios, pueden resultar excesivos. Esta combinación de factores se traduce en la desinstalación de las aplicaciones por parte de los usuarios. Este aspecto es importante tenerlo en cuenta a la hora de plantear la posibilidad de optar por la creación de una aplicación que sirva de punto de entrada para que un ciudadano emita su voto, pues obligará al usuario a descargarse la nueva aplicación, instalarla, emitir su voto y finalmente, al comprobar que la aplicación carece de mayor utilidad desinstalarla.

Cabe añadir que, para cada votación, sería necesario actualizar dicha aplicación o crear una completamente nueva que se ajustara a los requisitos del nuevo proceso. Algo bastante costoso, incómodo, y poco práctico, lo que ocasionará, en un corto plazo de tiempo, el desinterés de la mayoría de los usuarios.

Por tanto, se ha de procurar encontrar una solución que no implique los inconvenientes del desarrollo de aplicaciones específicas y a la vez resulte atractiva para el ciudadano sin subestimar todos los desafíos que la seguridad plantea. La idea podría derivar del aprovechamiento de una aplicación existente, que goce de una gran popularidad y garantice en la medida de lo posible que no supondrá un problema de seguridad. Como se ha visto, los clientes de mensajería instantánea son ampliamente usados, por lo que resultan un entorno óptimo para poder llegar al mayor número de personas. Además, en los últimos años la mayoría empiezan a ofrecer métodos que procuran garantizar la privacidad y seguridad del usuario.

Todo esto implicaría que ya no es necesario el desarrollo de una aplicación específica para un caso concreto, bastaría con la adaptación de un entorno consolidado como es el caso de las aplicaciones de mensajería instantánea.

## 1.2 Alcance

---

El objetivo del presente TFG es el diseño y desarrollo de un sistema de votación mediante un *bot* incorporado a la aplicación de mensajería instantánea de código abierto *Telegram*.

Este sistema de votación deberá ser escalable, y permitir realizar consultas ciudadanas a través de *Telegram* de forma personalizada, sencilla y fiable. Además deberá ser capaz de asistir y guiar al usuario durante todo el proceso de votación. Para alcanzar este objetivo propuesto se sugiere el desarrollo y evaluación de un prototipo para un municipio concreto con el fin de determinar los aspectos del sistema visibles para el usuario final así como la definición final de requisitos.

Este objetivo principal se desglosa a su vez en los siguientes objetivos secundarios:

- Estudiar y analizar las opciones de voto telemático existentes en el mercado.
- Valorar las distintas herramientas que permitirían crear un *bot*.
- Elaborar un prototipo usable de un sistema de votación mediante un *bot*.
- Evaluar el prototipo diseñado en un entorno real.
- Incorporar los requisitos derivados de la evaluación del prototipo.

Sin embargo ha de quedar claro que no es objetivo de este trabajo estudiar en profundidad los aspectos relacionados con la seguridad de las comunicaciones ni el posicionamiento a favor o en contra de los sistemas electrónicos de participación.

## 1.3 Contexto

---

En el momento de iniciar la apertura del gobierno encontramos el primero de los escollos: lograr la implicación activa de los ciudadanos en la toma de decisiones, derivando el problema hacia una doble vertiente: la tecnología y la política. Para solucionarlo será necesario que en el marco del presente TFG ambas disciplinas compartan el espacio y tengan el mismo peso específico en cuanto al diseño de la solución. Además, en cuanto a la utilidad efectiva de la solución, su desarrollo ha de estar basado en un caso real. Este es el motivo por el que el desarrollo del presente proyecto se enmarca en la actividad de Hub Cívico en colaboración con el Ayuntamiento de Quart de Poblet.

### 1.3.1. Ayuntamiento de Quart de Poblet

El Ayuntamiento de Quart de Poblet es el primer ayuntamiento del Estado Español en contar con una Concejalía de Gobierno Abierto creada en la legislatura municipal durante el periodo 2011-2015. Recientemente, esta concejalía ha resuelto en una Comisión integrada por todos los grupos políticos con representación en el pleno municipal y asesorada por historiadores e historiadoras locales, la convocatoria de una consulta online con el objetivo de cambiar el nombre actual de ocho calles del pueblo. La Figura 1.1 muestra la nota de prensa en la que se presenta dicha consulta.

Se trata de la primera experiencia del municipio en este ámbito, pues anteriormente nunca había llevado a cabo un referéndum en línea.

El principal reto era alcanzar la máxima participación posible, por lo que era condición necesaria disponer de todas las vías de participación que estuvieran al alcance. Con este fin se creó un grupo mixto de expertos, tanto en materia de participación como tecnológica, para poder estructurar una idea que permitiera hacer frente al reto planteado.

Una vez aprobado el proyecto desde el plano político, se construyó un grupo de trabajo para la construcción y puesta en marcha de dicha solución, conformado por el Jefe de servicio y el responsable de sistemas del área de tecnologías de

# LAS PROVINCIAS



VALENCIA | DEPORTES | ECONOMÍA | POLÍTICA | MUNDO | SUCESOS | SOCIEDAD | CULTURAS | TECNOLOGÍA | GENTE | PLANES



## OTRAS COMARCAS

VALENCIA | OTRAS COMARCAS | VALENCIA CIUDAD | LA COMUNITAT AL MINUTO

### Quart de Poblet votará para cambiar el nombre de las calles afectadas por la Ley de Memoria Histórica

Entre el lunes 6 de junio y el sábado 11 de junio, a las 14.00 horas, podrán acudir todas las personas censadas a partir de los 16 años

EFE | VALENCIA

31 mayo 2016  
14:23

Figura 1.1: Nota de prensa sobre el proceso participativo de Quart de Poblet.

la información del Ayuntamiento, un desarrollador de la empresa *IdiEikon* y un desarrollador de Hub Cívico: El que escribe.

### 1.3.2. Hub Cívico

Hub Cívico<sup>1</sup> es una iniciativa sin ánimo de lucro puesta en marcha por Grup Càlam<sup>2</sup> que tiene como objetivo facilitar el uso de la tecnología en procesos de empoderamiento de la ciudadanía. Lejos de grandes desarrollos la iniciativa se orienta inicialmente a proveer soluciones concretas a problemas específicos. Así, desde Hub Cívico trabajan con organizaciones de la sociedad civil e instituciones para lograr que los procesos de transparencia, participación/colaboración y rendición de cuentas sean más efectivos y lleguen a un mayor número de ciudadanos y ciudadanas.

Desde el inicio de sus actividades, en 2015, Hub Cívico ha colaborado con diferentes organizaciones e instituciones<sup>3</sup>. Por su relación con este proyecto, cabe destacar: La colaboración con la Diputación de Valencia en la incorporación de la ciudadanía al desarrollo de la Estrategia en materia de Gobierno Abierto a través de la implantación de la herramienta *DemocracyOS*<sup>4</sup>; y la colaboración con

<sup>1</sup>Puede consultarse información sobre sus integrantes en el sitio web de la iniciativa. Disponible en: <http://hubcivico.org>. [Acceso el 01.07.2016]

<sup>2</sup>Puede consultarse información sobre *Grup Càlam* en el sitio web de la iniciativa. Disponible en: <https://www.facebook.com/grupcalam> [Acceso el 01.07.2016]

<sup>3</sup>Puede consultarse información detallada sobre los diferentes proyectos en el sitio web de la iniciativa. Disponible en: <http://hubcivico.org/blog/> [Acceso el 01.07.2016]

<sup>4</sup>Puede consultarse información detallada sobre *DemocracyOS* en el sitio web. Disponible en: <http://democracyos.org/> [Acceso el 01.07.2016]

la organización *Fent Ciutat* en la implantación de la herramienta *consider.it*<sup>5</sup> en el proceso participativo de definición de un índice de empoderamiento de iniciativas ciudadanas. Igualmente, en el capítulo de desarrollos propios, Hub Cívico ha diseñado *civicBOT*<sup>6</sup>: un *bot* para *Telegram* que facilita a la ciudadanía fiscalizar la actuación de los partidos políticos y compartir, en abierto, los resultados.

Además, desde principios de 2016, Hub Cívico ha empezado a trabajar en la iniciativa *Open City Council (OCC)* un espacio para el desarrollo de proyectos que faciliten que la ciudadanía pueda colaborar en los procesos de toma de decisión llevados a cabo por los ayuntamientos. Dentro de esta iniciativa cabe destacar proyectos como *OCCDocs* y *OCCTv*. *OCCDocs* tiene como objetivo el desarrollo de una propuesta de estandarización, normalización y apertura de toda la información implicada en los plenos de un ayuntamiento. En este caso, el proyecto se realiza en colaboración con el Ayuntamiento de València. *OCCTv* tiene como objetivo diseñar un formato *transmedia* de televisión para la retransmisión de los plenos de los ayuntamientos de una forma atractiva para la ciudadanía.

## 1.4 Metodología

---

Dentro del amplio abanico de posibilidades existentes para abordar la realización de este proyecto, se ha optado por un modelo mixto entre el desarrollo orientado a prototipos y el desarrollo ágil.

Dado que interesa evaluar la idoneidad de la propuesta, además de garantizar que el proyecto cumple con los objetivos planteados, consideramos que el desarrollo del prototipo resulta adecuado. Este tipo de metodología permite mejorar la calidad de los requisitos y especificaciones propuestos inicialmente. Además, posibilita la reducción de tiempo y costes de desarrollo, ya que los cambios en el diseño o requisitos, que no hubieran sido tenidos en cuenta al inicio de la planificación y que se incorporen posteriormente, podrían aumentar su coste de forma exponencial a medida que se introducen en un proyecto que se va acercando al final de su ciclo de vida.

A su vez, el prototipo requiere de la participación de los potenciales usuarios finales, lo que les permite ver, e interactuar, con un prototipo funcional. Esto permite evitar los malentendidos y errores de comunicación que se producen cuando el binomio desarrolladores vs. usuarios creen, el uno respecto del otro, que entiende lo que se ha propuesto inicialmente. Dado que los usuarios conocen el problema, y el resultado que esperan para su solución incluso mejor que el equipo de desarrollo, una mayor interacción puede dar lugar a un producto final de mayor calidad. De esta forma, podemos tener una mayor seguridad de que el producto final satisfará las necesidades del usuario.

Igualmente, la incorporación de una metodología ágil para las tareas de desarrollo tanto de los prototipos como de versiones finales del producto, permite una mayor flexibilidad ante cambios en el planteamiento.

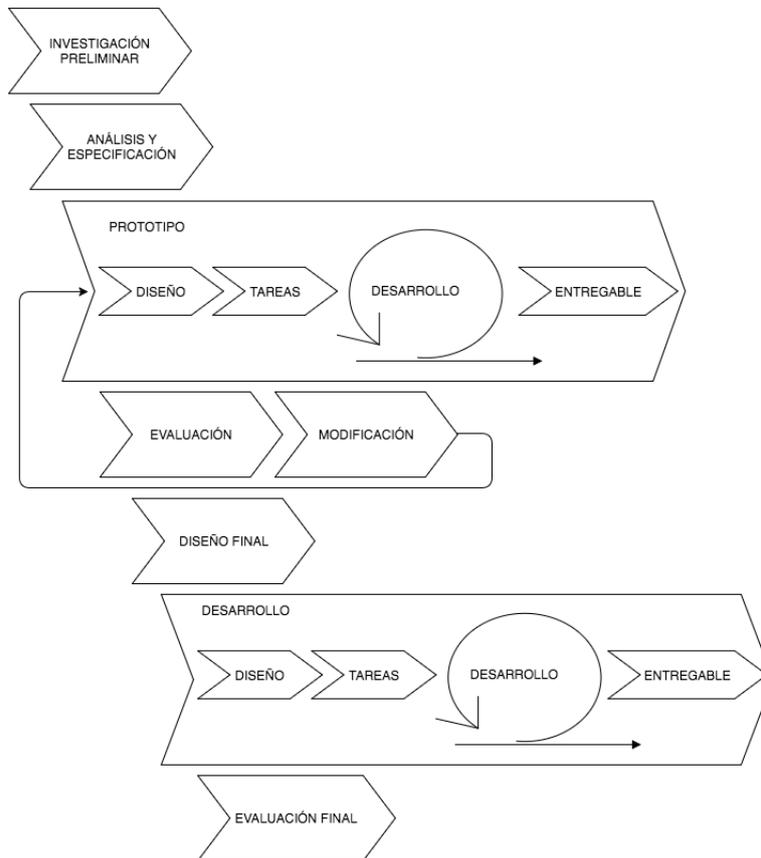
---

<sup>5</sup>Puede consultarse información detallada sobre *consider.it* en el sitio web. Disponible en: <https://consider.it/> [Acceso el 01.07.2016]

<sup>6</sup>Puede consultarse información en el sitio web del proyecto. Disponible en: <http://civicbot.hubcivico.org/> [Acceso el 01.07.2016]

El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, permite a la obtención de entregables de calidad. Además el hecho de desarrollar primero las funcionalidades de mayor valor, y de conocer la velocidad a la que el equipo avanza en el proyecto, logra despejar riesgos anticipadamente.

Tal y como se ilustra en la Figura 1.2, se han fusionado las dos metodologías anteriormente descritas.



**Figura 1.2:** Esquema de la metodología implementada.

Como puede observarse, en el esquema propuesto se han ajustado los requisitos que este trabajo plantea de cara a elaborar una metodología que permita alcanzar los objetivos planteados. Los elementos de la adaptación serán descritos a continuación:

### **Investigación preliminar**

Etapa en la que se determina el problema y su ámbito, se gradúan la importancia y los potenciales efectos que el planteamiento inicial pueda tener sobre el proyecto.

### **Especificación de requerimientos**

Etapa en la que se estudian y determinan los requerimientos que los futuros usuarios establezcan.

### **Diseño y construcción del prototipo**

Al final de esta etapa se dispone de un prototipo con la máxima funcionalidad con el objeto de ser evaluado para incorporar posibles cambios.

**Evaluación**

Etapa destinada a recoger todos los datos obtenidos durante la evaluación de prototipo, y estudiar su incorporación al desarrollo final del proyecto. Si la evaluación no fuera satisfactoria, se volvería a la etapa de diseño replanteando aquellos aspectos del prototipo que no hubiesen superado la evaluación, volviendo a construirse y a evaluarse.

**Diseño final**

Superada la evaluación, se plantea un diseño final basado en el inicial, incorporando todas las mejoras necesarias para disponer del producto final.

**Evaluación final**

Esta etapa se centra en la validación del correcto funcionamiento del desarrollo definitivo. Los cambios identificados en la evaluación del prototipo son probados para asegurar su correcto y completo funcionamiento respecto a los requerimientos. Estas pruebas se realizan tantas veces como sea necesario para verificar cualquier tipo de anomalía en el sistema.

## 1.5 Estructura de la memoria

---

El presente trabajo se ha estructurado en cinco capítulos. Además de estos capítulos, se encuentran la presente introducción, las referencias bibliográficas empleadas y los anexos.

**Capítulo 2**

El segundo capítulo está dedicado al análisis del estado del arte; concretamente al estudio del voto telemático. Además se realiza una descripción y comparación de los diferentes sistemas de votación telemática, incidiendo en la votación vía internet y los problemas asociados a esta. Por último, se realizará una introducción a los *bots* y sus aplicaciones.

**Capítulo 3**

El tercer capítulo contiene una descripción detallada de la idea propuesta como solución al problema, identificando además los riesgos y limitaciones de esta.

**Capítulo 4**

El cuarto capítulo describe el proceso de diseño y construcción del prototipo derivado de la solución propuesta. Concretamente se realizará una especificación de requerimientos, obtenidos de la solución propuesta junto con una descripción de los procesos de diseño y construcción seguidos para alcanzar un prototipo funcional que satisfaga los requerimientos especificados. Por último se realizará una evaluación y valoración del prototipo una vez este haya sido implantado y probado en un entorno real.

**Capítulo 5**

Finalmente, en el quinto capítulo, se recogerán las conclusiones obtenidas del desarrollo del presente trabajo, así como el trabajo futuro.



---

---

# CAPÍTULO 2

## Estado del arte

---

Antes de continuar, es necesario realizar el análisis de la situación actual de los sistemas de votación y sus características, y estudiarán todas las restricciones que un sistema de votación electrónica implica, tanto a nivel de seguridad como de garantía del proceso de votación. La documentación y análisis realizados servirán de base para el planteamiento posterior de la solución abordada.

### 2.1 El voto telemático

---

El voto telemático, conocido también como voto electrónico, se explica como la emisión de un voto por medios electrónicos, de tal forma que las tareas de emisión y conteo de votos se automaticen y no sea necesaria la intervención humana durante todo el proceso. Dependiendo de la implementación particular, el voto electrónico puede abarcar una amplia gama de servicios: desde una transmisión de votos básica, hasta toda la actuación necesaria para organizar y culminar un referéndum. [11] Del mismo modo, el grado de automatización puede variar entre tareas simples hasta una solución completa que incluye la composición del censo de votantes y la autenticación, la emisión del voto, escrutinio, el cifrado de los votos y la transmisión a los servidores recolectores de votos, conteo, ponderación y validación.

Un sistema de votación electrónico debe realizar la mayor parte de estas tareas, cumpliendo, al mismo tiempo, con un conjunto de normas establecidas por los organismos reguladores.

Los sistemas de votación telemática deberán también ser capaces de cumplir con los siguientes requisitos:

#### **Seguridad**

El sistema encargado de la recolección y conteo de los votos ha de ser capaz de resistir ataques malintencionados, con el objetivo tanto de proteger los resultados de la votación, como para prevenir una fuga de datos sensibles.

#### **Exactitud**

El resultado de la votación ha de ser exacto y sin margen de error.

#### **Integridad**

En todo momento, la integridad electoral ha de ser preservada.

**Rapidez**

El sistema ha de ser rápido, permitiendo así un proceso de votación fluido y sin retrasos.

**Privacidad**

Se ha de garantizar en todo momento la privacidad del votante.

**Auditabilidad**

Ha de ser posible examinar los procesos utilizados para recoger y contar los votos además de permitir volver a contar los votos con el fin de confirmar la exactitud de los resultados.

**Accesibilidad**

Se debe garantizar la accesibilidad a todos los colectivos de votantes.

**Rentabilidad**

El sistema deberá poder ajustarse a los requisitos económicos de cada votación.

**Escalabilidad**

Tanto si se trata de una pequeña consulta ciudadana como de una votación nacional, el sistema deberá poder atender a cualquier volumen de usuarios.

**Sostenibilidad**

Se deberá cuidar el consumo y gasto energético.

## 2.2 Sistemas de votación electrónica

---

### 2.2.1. Clasificación

Actualmente, según fuentes de *ACE (Electoral Knowledge Network)*, los sistemas de votación electrónica presentes en el mercado se pueden clasificar en cinco grandes grupos, a saber:[12]

**Los que emplean tarjetas perforadas:**

El voto se realiza mediante una tarjeta (o tarjetas) en la que los votantes perforan agujeros con un punzón, en la opción que deseen votar. Después de realizar el orificio u orificios, el votante puede depositar la papeleta perforada en una urna para y, una vez finalizada la votación, procesar automáticamente todas las papeletas o, directamente, puede introducir la papeleta en un sistema que contabiliza directamente el voto y lo almacena en un sistema de información. Los dos sistemas más famosos que emplean la tecnología de las tarjetas perforadas son *Votomatic* y *Datavote*.

El sistema *Votomatic* plantea la inclusión en la tarjeta a perforar de las opciones disponibles representadas únicamente mediante un número. A continuación, en un documento adjunto a la papeleta, aparece la nomenclatura de cada número y otras instrucciones. A diferencia de *Votomatic*, en *Datavote*, las opciones disponibles se plasman junto a la ubicación del agujero a ser perforado.



Figura 2.1: Sistema Votomatic.

#### Los basados en escáneres ópticos:

Utilizan un escáner o cámara para leer y procesar las papeletas que contienen el voto. Estos sistemas, a su vez, se clasifican en tres subtipos:

- Sistemas *marksense*. Bastante básico: las marcas realizadas por el votante en la papeleta son leídas y reconocidas por un escáner.
- Marcadores electrónicos de votación o *EBM* (*electronic ballot markers*), por sus siglas en inglés): permiten al votante emitir su voto desde una máquina que lo transcribe a un código legible por un escáner o cámara. Posteriormente, el votante acude a un lector de votos para depositarlo, quedando almacenado.
- Votación mediante lápiz digital: una pequeña cámara integrada en el lápiz reconoce lo que el votante ha marcado en una papeleta. Una vez recogido el voto, se volcará en el sistema de recuento de votos contabilizado y almacenado.

Como hemos visto, los sistemas de votación ópticos combinan papel con dispositivos electrónicos, esto quiere decir que en todos los casos se mantiene una papeleta tangible. Gracias a esto, los sistemas de votación ópticos permiten también el recuento manual de las papeletas. Además, el sistema es muy intuitivo: para el votante no cambia prácticamente nada entre un voto tradicional y éste, ya que continúa emitiendo su voto con una papeleta. De esta manera, en el caso de que por cualquier razón, el sistema de

escaneo no funciona o se produce un fallo, las papeletas se pueden contar manualmente.

### **Los de grabación directa o DRE (*Direct-recording electronic*)**

Para su funcionamiento se debe utilizar una máquina recolectora de votos y resulta bastante sencillo: el votante únicamente pulsa el botón que representa su elección. Una vez ha concluido el proceso de votación, la máquina realiza un recuento de los datos de la votación almacenados. Las máquinas de votación *DRE* se empezaron a utilizar masivamente en 1996 en Brasil. También fueron utilizados a gran escala en los EE.UU. después de la experiencia del estado de Florida en 2000. Las máquinas *DRE* se desplegaron también en Europa, por ejemplo, en los Países Bajos, donde la empresa *NE-DAP* proporciona sus propias máquinas *DRE* desde 1989 hasta 2006.

### **De voto electrónicamente verificado (*VVPAT*):**

En este caso no se trata de un sistema de votación electrónica en sí, sino que hace referencia a un componente que se puede integrar con varias formas de sistemas de votación electrónica. *VVPAT* está pensado como un sistema de verificación independiente para sistemas de votación con el objetivo de permitir a los votantes verificar que su voto fue almacenado correctamente, ya sea para detectar un posible fraude electoral o el mal funcionamiento, como para proporcionar un medio que audite los resultados electrónicamente almacenados.

### **Los de votación por Internet**

Tal y como el propio enunciado hace adivinar, el empleo de esta técnica requiere la utilización de la red para la emisión y/o transmisión del voto. Puede adoptar diversas formas, dependiendo de si se utiliza en ambientes incontrolados (no haciendo uso de un censo de electores) o controlados (identificando al electorado). Con el voto por Internet, a distancia, ni las máquinas cliente ni el entorno físico están bajo el control de los funcionarios electorales. Los votantes pueden emitir su voto desde, prácticamente, cualquier lugar (en casa, en el lugar de trabajo, en las terminales públicas de Internet, etc.). La votación se transmite entonces a través de Internet. Este método ofrece más ventajas para los electores activos, pero al mismo tiempo es más proclive a sufrir problemas de seguridad. Además, se generan dudas acerca de Internet como medio de transmisión de información confidencial, miedo a los ataques informáticos y la desconfianza ante la posibilidad de una influencia excesiva que se ejercería sobre el votante durante el proceso de votación.

## **2.2.2. Problemas asociados al voto electrónico**

La votación por este medio, como casi cualquier procedimiento electrónico, no está exenta de problemas. Principalmente, la documentación consultada distingue entre:

### Los relacionados con la programación:

Actualmente resulta casi imposible escribir software libre de errores, puesto que cuanto más complejo es el software, más complicado resulta mantenerlo, así como detectar y solventar errores.

La elección del software para un sistema de votación resulta, al menos, complicado: Cada votación suele ser única, con sus propios requisitos y características, por lo que encontrar un software que se ajuste completa y específicamente a cada caso es difícil y, muy probablemente, será necesaria una adaptación posterior del software, lo que conlleva una mayor probabilidad de que surjan errores que no han sido tenidos en cuenta por haber aparecido posteriormente.

Sirvan como ejemplo los siguientes casos reales en los que se han producido este tipo de problemas<sup>1</sup>:

- Noviembre de 2003: Una máquina MicroVote en el condado de Boone, (Carolina del Norte, EEUU) contaba más de 100.000 papeletas a pesar de que sólo había 5.000 electores.
- Diciembre de 1999: En el Condado de Onondaga, Nueva York, Bob Faulkner, nuevo en la política, terminó la jornada electoral seguro de que había ganado. Pero después los empleados de la Junta electoral del condado volvieron a comprobar los totales. Faulkner había perdido. Los funcionarios electorales descubrieron que un error de programación de software fue el culpable de este grave error.
- Noviembre de 1998: En unas elecciones en Salt Lake City (Utah, EEUU), 1.413 votos desaparecieron. Este error se atribuyó a un problema de programación que causó que un lote de votos no fuera guardado en la base de datos.

### Imposibilidad de recuentos verificables

Delegar el almacenamiento de votos a procesos completamente electrónicos de registro conlleva la inexistencia de copias físicas que permitan realizar un recuento de los votos emitidos en caso de fallo con pérdida de almacenamiento.

Los sistemas de votación basados en DRE permiten a los votantes verificar y corregir su voto antes de emitirlo de forma definitiva. Sin embargo, no permite una nueva verificación del voto realizado, por lo que la forma en la que este queda almacenado es totalmente opaca para el votante.

Sin registro de los votos en papel, que reflejen con precisión las decisiones de los electores, es simplemente imposible realizar un recuento significativo. Aunque la mayoría de las máquinas de votación tipo DRE pueden imprimir un registro en papel de los votos emitidos, este informe no se genera hasta después del cierre de las urnas y no es más que una copia impresa del contenido de los registros electrónicos. Si el registro electrónico no es correcto el informe impreso también será incorrecto.

---

<sup>1</sup>Puede consultarse más información en el sitio web de Ballotpedia. Disponible en: [https://ballotpedia.org/electronic\\_vote\\_fraud](https://ballotpedia.org/electronic_vote_fraud) [Acceso el 18.06.2016]

## Posibilidad de fraude

Las posibilidades de fraude electoral que pueden darse en los sistemas de votación electrónica se incrementan exponencialmente en comparación con las posibilidades de fraude en métodos electorales tradicionales. Por ejemplo: trabajadores de una empresa de distribución de sistemas de votación electrónica con intenciones deshonestas podrían ocultar código malicioso en el software. Esta empresa, a continuación, podría distribuirlo, sin ser consciente de su contenido malicioso, a miles de máquinas de votación en todo el mundo y, de esta manera, alterar los resultados de las elecciones en todos los países donde se utilice esas máquinas. Si bien existen procedimientos de auditoría y certificación del software, estos normalmente no se realizan de forma precisa ya que los funcionarios de la administración no disponen, por lo general, de conocimientos sobre seguridad informática.

Las preocupaciones sobre el fraude no son meras especulaciones. Un estudio de 2003, llevado a cabo por expertos en seguridad informática de la Universidad Johns Hopkins y la Universidad Rice [13], reveló cientos de fallos de seguridad en el software para sistemas de votación de un fabricante líder.

Un ejemplo de fraude electoral se dio en noviembre de 2002: En Texas (EEUU) se tuvieron sospechas de fraude tras la amplia ventaja que obtuvieron dos candidatos republicanos. Tras analizar los sistemas y recontar los votos manualmente se descubrió que los demócratas habían ganado, en realidad, por un amplio margen. Los funcionarios electorales atribuyeron y confirmaron su sospecha de que un chip trampeado.<sup>er</sup> a el culpable.

Por otro lado, el documental Hacking Democracy[14], filmado durante tres años, documenta la investigación de anomalías e irregularidades en los sistemas de votación electrónica que se produjeron durante las elecciones de 2000 y 2004 en Estados Unidos, especialmente en el condado de Volusia, Florida.

## Posibilidad de suplantación de identidad

Aunque existen muchos avances en el campo de la criptografía, la autenticación basada en certificados y la autenticación basada en biometría, sigue siendo un reto por resolver[15].

En el ámbito de los certificados como medio de autenticación de las personas en internet, estos no son infalibles, pues disponiendo del certificado y el código para activarlo sería posible suplantar la identidad del votante

La incapacidad para identificar fehacientemente al votante es uno de los mayores problemas de la votación basada en internet, pues al no ser presencial, es imposible determinar con absoluta certeza que el usuario que se encuentra tras la pantalla es el votante que dice ser.

Aplicación	Tipo	Código abierto	Cifrado	Tipos de votación diferentes	Verificación	Resultados auditables	Móvil	Censo
Demokratian	Gratis	Si	No	Si	Si	Si	Si	Si
Agora Voting	De pago	No	Si	Si	Si	Si	Si	Si
Ferendum	Gratis	No	No	No	No	No	No	No
App-Gree	Gratis	No	No	No	No	No	No	No

**Tabla 2.1:** Elaboración propia: Comparativa de sistemas de votación en línea.

## 2.3 Votación en línea

En el mercado existen actualmente diversas aplicaciones que implementan un sistema de votación en línea o basado en internet, que permiten la votación mediante una aplicación basada en web, a la que los votantes acceden y siguen un procedimiento que les guía indicándoles los pasos a seguir hasta poder emitir su voto. En la Tabla 2.1 se incluye una comparativa de algunos de estos sistemas en base a sus distintas características: Tipo, que establece si la aplicación es gratuita o de pago; Código abierto, que muestra si su código es auditable o no; Cifrado, que expresa si se implementa algún sistema de cifrado de los datos; Tipos de votación diferentes, que muestra si soporta la configuración de distintos tipos de votación como la multiselección, a favor y en contra, etc; Verificación, donde se expresa si existen métodos en los que los votantes puedan verificar su voto; Resultados auditables, que explica si los resultados del proceso de votación podrían auditarse para verificar su corrección; Móvil, que recoge si la aplicación está adaptada para poder ser utilizada en dispositivos móviles; Censo, que muestra si se podría implementar un sistema de comprobación de usuarios autorizados para emitir voto.

Las características de este tipo de votación varían en función del tipo de aplicación que las implementa. Como observamos en la Tabla 1.1, no todas permiten, por ejemplo, la posibilidad de establecer un censo de votantes con el que poder validar a los usuarios. Tampoco todas las opciones permiten una verificación posterior del voto, esto quiere decir que el usuario no podrá comprobar que su voto permanece inalterado en el sistema.

Otra característica interesante de algunas de las opciones analizadas es la posibilidad de disponer de un sistema que permita auditar el proceso, esto es, poder comprobar que los resultados de la votación no han sido alterados durante el su sustanciación.

---

## 2.4 Bots

---

La definición clásica los describe como aplicaciones de software capaces de ejecutar tareas simples y repetitivas de forma automática.

La virtud principal de un *bot* es la capacidad de ejecutar las tareas a una velocidad mucho más alta de lo que sería posible para un ser humano.

Durante años, una de las principales utilidades que han tenido los *bots* en el mundo de internet, son las relacionadas con el rastreo *web* o *crawling*; utilizando un *bot* para extraer, analizar y almacenar datos de páginas web indexándolo o utilizándolo para cualquier otra finalidad. Un ejemplo con estas características es el *Googlebot*, el *bot* rastreador de *Google*, comúnmente conocido como araña, utilizado por la empresa del conocido buscador para rastrear las páginas webs existentes y buscar otras nuevas, junto a contenidos capaces de ser añadidos al índice principal de *Google*.

Tal y como podemos deducir, este tipo de *bots* son completamente autónomos y, una vez ejecutados, realizarán las tareas que tengan definidas de forma repetitiva e independiente.

Durante el último año, sin embargo, han empezado a aparecer otro tipo de *bots*, basados en mensajería o *chatbots* que tienen la peculiaridad de simular una conversación con el usuario a través de texto o audio.

Los sistemas que tratan de imitar las interacciones humanas han existido desde la década de 1960. Fue entonces cuando el profesor Weizenbaum, del *MIT*, creó un *bot* llamado *ELIZA*, que tomó prestadas técnicas de psicoterapia para mantener conversaciones simples con los usuarios. Más tarde, aparecieron plataformas en los clientes de mensajería de escritorio como *AOL* e *Instant Messenger*, en el que los usuarios, por ejemplo, podían pedir el pronóstico del tiempo.

Estos *bots* se mantuvieron latentes durante la transición de la mensajería instantánea hacia los dispositivos móviles, dejando atrás los entornos de escritorio. Sin embargo, se puede constatar que durante este último año han empezado a despertar de su letargo debido, principalmente, a dos factores: En primer lugar, la inteligencia artificial y la computación en la nube, que ha experimentado un crecimiento muy acentuado, tanto en potencia como en competitividad. En segundo lugar, los *bots* son sistemas que pueden ahorrar mucho dinero a las empresas, pues el coste de uno de ellos es muy inferior al de un ser humano realizando el mismo trabajo [16].

Además, los *bots* basados en mensajería o *chatbots* suelen estar diseñados para simular de manera cuasi fidedigna el comportamiento humano, pretendiendo imitar, así, una conversación de forma natural.

Los *chatbots* se utilizan normalmente en sistemas cuyo objetivo es establecer un diálogo con otro interlocutor, ya sea como servicio de atención al cliente (contestadores automáticos) o para que el usuario adquiera información y/o reciba instrucciones.

Algunos de estos *chatbots* en la actualidad utilizan sistemas de procesamiento de lenguaje natural con el fin de poder entender o interpretar lo que el usuario intenta expresar. Sin embargo, y de cara a simplificar el proceso de desarrollo,

suelen ser sistemas que responden a órdenes preestablecidas o texto predefinido y reaccionan o contestan en consecuencia.

Actualmente, estos *bot* conversacionales empiezan a gozar de una gran popularidad en el entorno empresarial. Por una parte y, según un estudio de *ZenDesk* (una popular herramienta que concentra los servicios de atención al usuario de empresas como *Dropbox*, *Trivago* y *Vodafone*, entre otras muchas) más del 50 % de los usuarios piensan que es importante poder resolver sus propios problemas sin tener que depender de los agentes de atención al cliente[17]. Además, según otro estudio, realizado por la empresa *OneReach* (especializada en comercialización de *bots* para atención al cliente), el 64 % de los consumidores preferiría utilizar la mensajería instantánea, antes que la comunicación vía voz con otro interlocutor, como canal para ser atendido[18].

Hemos de tener en cuenta también que los *bot* introducen una nueva característica que resulta un factor determinante para la satisfacción de los usuarios: la inmediatez: permiten respuestas casi inmediatas a las entradas de un usuario, esto es, reaccionan en tiempo real a lo que el usuario introduce en una conversación.

Como podemos deducir, estamos ante lo que podría ser un nuevo canal de comunicación con los usuarios, pues abandona, en cierta forma, los clásicos como la comunicación vía voz, correo electrónico, aplicación *web*, etc. Además, la inmediatez de los *bot* aumenta la satisfacción de los usuarios pues se genera una sensación de atención constante, algo que la mayoría de ellos valora positivamente.

## 2.5 Síntesis

---

Hasta hace relativamente poco tiempo los medios que teníamos a nuestra disposición, no aprovechaban todo el potencial de los sistemas de votación telemática tal vez por falta de confianza de los usuarios, o tal vez por causas achacables a los propios sistemas.

Además, como ha quedado patente en anteriores párrafos, los disponibles resultaban ser poco eficientes, incómodos y poco convenientes en muchas ocasiones.

Otras veces aparentaban ser toscos y, hasta cierto punto, rudimentarios, pues la mayoría de los sistemas utilizados, exigía a los usuarios acudir físicamente a los colegios electorales para emitir el voto.

Se explicaba anteriormente que, en la mayoría de las ocasiones estos sistemas aparentaban ser poco fiables (véase el apartado 2.2.2 que trata de los problemas asociados al voto electrónico), pues se ha demostrado en algunas ocasiones que han existido tanto errores de programación como intentos de fraude, que han culminado ofreciendo unos resultados que nada han tenido que ver con lo elegido por los usuarios.

Estos problemas de seguridad incrementaron aún más, si cabe, el temor y el escepticismo de los ciudadanos a la hora de confiar su voto a través de estos entornos electrónicos, pues un sistema que no pueda ser objeto de auditoría no

es más que una caja negra, de la que no se conoce su funcionamiento interno y si este es el adecuado.

Sin embargo, en la última década el voto en línea y/o remoto empieza a ganar popularidad, pues este ya no requiere de la presencia física del votante. Lo que unido a los nuevos e incipientes canales de comunicación, nos permite atisbar la posibilidad de que, en un futuro muy cercano, podamos disponer del potencial necesario para actualizar y mejorar la votación telemática (p.ej. los *bot*)

Eso no significa que ya esté todo hecho. Parece más que evidente la cantidad de trabajo pendiente de hacer hasta obtener un sistema de votación electrónica fiable, seguro y cómodo. Los siguientes pasos se han de encaminar a obtener la seguridad, fiabilidad y auditabilidad de los sistemas de votación electrónica que esperan los electores: tanto los activos como los pasivos.

---

---

## CAPÍTULO 3

# Solución propuesta

---

En este trabajo se propone el análisis e implementación de un sistema de votación basado en un *bot*, que permita la emisión de votos de forma segura, anónima y cómoda para el ciudadano. Con lo que se pretende introducir este nuevo canal de comunicación en un proceso de gobierno abierto e innovación democrática, que permita incrementar los índices de participación y, por ende, involucrar en mayor medida a la ciudadanía.

El desarrollo abordado en el marco del presente TFG pretende incorporarse al conjunto de soluciones que integran la iniciativa *Open City Council (OCC)* de Hub Cívico que ha sido descrita en el apartado Contexto.

### 3.1 OCCVotingBOT

---

*OCCVotingBOT* es el nombre con el que se ha bautizado a la solución propuesta y corresponde a la siglas de la iniciativa (*OCC*) y al sistema de votación dentro de un *bot*. Se trata de un asistente que guiará al votante durante un proceso de votación. De esta manera, *OCCVotingBOT* se integrará dentro de cliente de mensajería instantánea que permita la incorporación de *bot*.

De forma general, el funcionamiento se basa en la interacción con el usuario, respondiendo en tiempo real a las acciones del mismo y proporcionando instrucciones precisas de los pasos que el usuario deberá ir siguiendo para poder emitir su voto.

El proceso de votación mediante la solución propuesta consta de cinco etapas. De forma general, podemos observar como en la etapa inicial el usuario elegirá el idioma en el que quiere ser atendido. En las siguientes etapas deberá proporcionar datos que permitan su identificación, para autorizarle, o no, a votar según sea el resultado de la validación a través de un censo establecido. A continuación se mostrarán las instrucciones para realizar la votación, así como las opciones disponibles de entre las que deberá elegir. Tras haber decidido las opciones, el usuario emitirá su voto en la siguiente etapa. Finalmente, en la última etapa el usuario podrá verificar que su voto se ha guardado correctamente.

De forma más detallada, según muestra la Figura 3.1, vemos como inicialmente el usuario deberá activar el *bot* enviando un comando identificado una barra (/) y la palabra reservada *start*, por lo tanto: */start*.

En este momento, el *bot* solicitará al usuario que indique el idioma en el que prefiere ser atendido, para lo que deberá pulsar uno de los botones que el *bot* mostrará y que identificarán a cada uno de los idiomas disponibles. Finalizado este paso, le dará la bienvenida al usuario y le proporcionará las instrucciones para pasar al siguiente paso. Junto con el saludo, preguntará al usuario si desea iniciar el proceso de autenticación para comprobar si este puede votar o no, dependiendo de si se encuentra registrado en el censo de votantes.

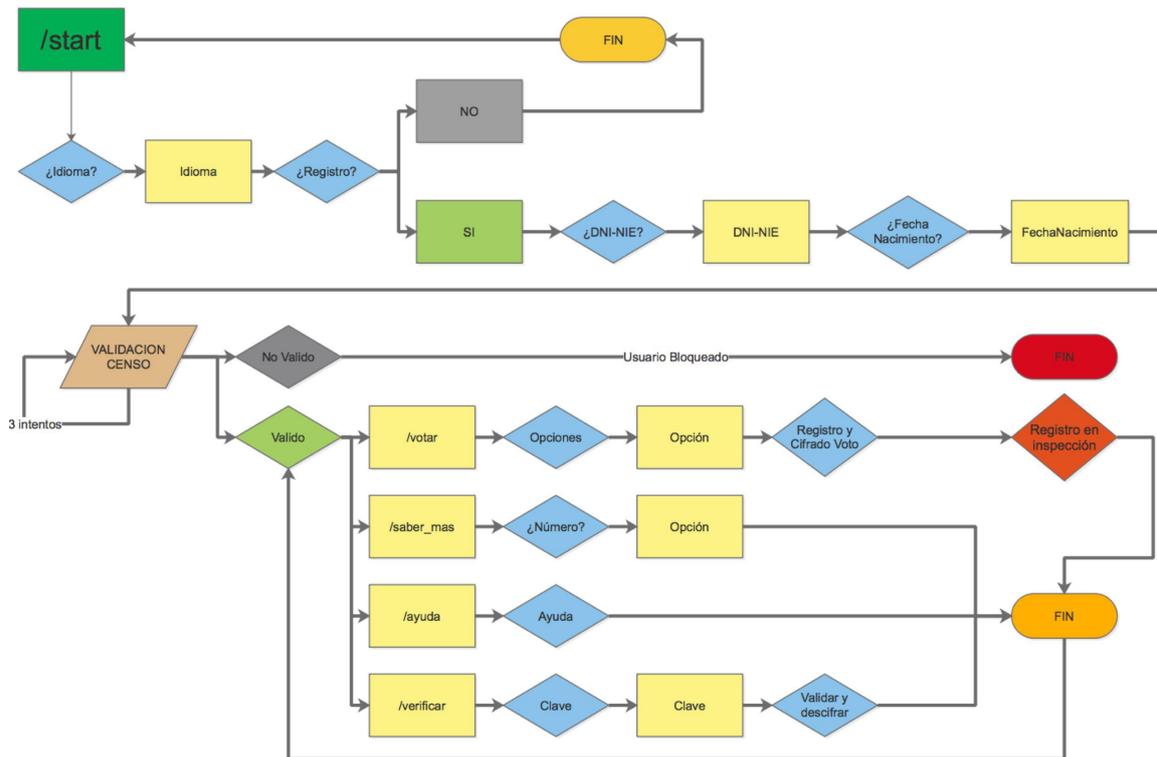


Figura 3.1: Boceto del esquema de funcionamiento de OCCVotingBOT.

Si el usuario decide iniciar el proceso de autenticación, se le pedirá que introduzca manualmente su número de documento nacional de identidad, o, en su caso, su Número de Identificación de Extranjero (NIE), así como su fecha de nacimiento. En este momento el *bot* comprobará si estos datos aparecen en el censo de votantes autorizados. En caso afirmativo, se le autorizará el acceso al siguiente paso. En caso contrario, el usuario contará con tres intentos para volver a identificarse, con el fin de no excluir a un usuario por el mero hecho de haber cometido un error en alguno de los dígitos que haya introducido. Si estos tres intentos fueran consumidos sin haber logrado una identificación satisfactoria, el usuario quedará bloqueado en el sistema y no podrá volver a interactuar con él.

Una vez completada la fase de identificación del votante, el *bot* procederá a mostrar las instrucciones para emitir el voto, junto con las opciones disponibles de entre las que el usuario deberá elegir.

Una vez el usuario haya decidido su voto y lo haya enviado al *bot*, este será validado por el sistema, para comprobar que cumple todas las restricciones establecidas en cuanto al formato que ha de tener dicho voto. Si la validación no fuera

satisfactoria, le comunicará el problema votante, quien podrá corregir su voto. Si el voto es válido, se realizarán las siguientes tareas:

1. En primer lugar, el voto se almacenará anonimizado en una base de datos, donde ocupará una posición aleatoria de cara a que no se pueda asociar a este con uno de los usuarios registrados.
2. En segundo lugar, el identificador del usuario que ha emitido el voto se encriptará de forma segura mediante una clave aleatoria generada en el momento previo al cifrado y que solo el usuario conocerá, y se almacenará junto con el voto.
3. A continuación se compondrá un correo electrónico dirigido a los interventores conteniendo el voto anonimizado. Con esto se garantizará, que, en el supuesto de producirse algún desajuste, la base de datos no ha sido alterada durante el proceso de votación o poder recontar los votos en caso de fallo del sistema con pérdida del almacenamiento en la base de datos.
4. Una vez se hayan completado todas estas tareas, se notificará al usuario del correcto registro de su voto y se le remitirá la clave con la que se ha cifrado su voto para que, posteriormente, en cualquier momento, pueda verificar que su voto sigue almacenado correctamente en el sistema.

## 3.2 Identificación de riesgos y limitaciones

---

### 3.2.1. Riesgos

#### Riesgos relacionados con la metodología

La elección del prototipado como metodología principal para el desarrollo del proyecto conlleva riesgos asociados que pudieran poner en peligro el correcto desarrollo del trabajo.

Por una parte, puede estimular un número de petición de cambios excesivo, ya que la única especificación de diseño es el propio prototipo, dando lugar a un ciclo interminable de modificación si no se establecen criterios de conclusión.

Por otro lado, los prototipos operativos pueden inducir a pensar a los interesados que el producto final está prácticamente dispuesto para su salida al mercado. Esto se debe a que los prototipos de alta fidelidad pueden ser confundidos con el producto real, cuando en realidad suelen albergar simplificaciones e imprecisiones.

También se podría dar la situación en la que los interesados podrían no estén dispuestos a participar en el ciclo de pruebas durante largos períodos de tiempo.

Además, se corre el riesgo de ignorar importantes características del sistema durante el desarrollo del prototipo, como por ejemplo, actuación, seguridad, robustez o fiabilidad.

Por último, el contexto de uso del prototipo puede diferir del contexto de uso final del sistema, dando lugar a problemas en la implantación final si esta se desarrolla en un contexto muy diferente al planteado.

### **Riesgos relacionados con la planificación**

En todos los proyectos, y este no tiene porqué ser una excepción, pueden aparecer riesgos dependientes de la planificación temporal del trabajo a realizar. Las fechas estimadas para la finalización de los entregables que se plantean en los siguientes capítulos pueden verse afectadas por errores de planificación. Además, al finalizar la fase de prototipado, podrían aparecer cambios de una envergadura considerable, no tenidos en cuenta durante la planificación general, y que requieran de una importante inversión de tiempo, lo que provocaría desajustes en la planificación inicial y retrasar todo el proyecto.

### **Riesgos relacionados con el equipo de trabajo**

Este proyecto plantea la implantación de su prototipo en un entorno real como es el caso del Ayuntamiento de Quart de Poblet. Esto conlleva la coordinación de diferentes equipos para su puesta en marcha, lo que podría generar una descoordinación que retrase o afecte al proyecto.

Por otro lado, este proyecto hará uso de tecnologías novedosas y no muy conocidas, por lo que la inexperiencia de algunos de los agentes implicados es otro factor a tener en cuenta a la hora de detectar riesgos relacionados con el equipo de trabajo.

### **Riesgos relacionados con la tecnología**

Las tecnologías que se van a utilizar en este proyecto son, en cierto grado, novedosas, incipientes y, en general, poco desarrolladas hasta el momento. Este hecho conlleva riesgos de cara al desarrollo de una solución estable y que garantice el cumplimiento de los requisitos que se establecerán, ya que la evolución de dicha tecnología durante la fase de desarrollo podría plantear serios problemas de compatibilidad e incluso provocar que el trabajo realizado, llegado el momento resultara inservible.

### **Riesgos relacionados con la seguridad**

En un sistema de votación electrónica, la seguridad es, sin duda un elemento imprescindible. Por este motivo, exponemos a continuación con mayor nivel de detalle las diferentes amenazas de seguridad a las que, con mayor probabilidad, deberá hacer frente la solución desarrollada:[19]

- Ataques ocasionados por inyección, tales como *SQL*, *OS* y *LDAP* se producen cuando los datos con origen no confiable se envían a un intérprete como parte de un comando o consulta. Los elementos o datos con carácter

hostil pueden engañar al intérprete para que ejecute comandos no deseados o accedan a datos sin la debida autorización.

- Los elementos de la aplicación relacionados con la autenticación y gestión de sesiones de usuario no se suelen aplicar correctamente, permitiendo a potenciales atacantes comprometer las contraseñas, claves o *tokens* de sesión, o para explotar otras debilidades de implementación que puedan asumir la identidad de otros usuarios.
- Los fallos de *XSS (Cross-Site-Scripting)* ocurren cuando una aplicación obtiene datos que no son de confianza y los envía a un navegador web sin la validación adecuada. *XSS* permite a los atacantes ejecutar secuencias de comandos en el navegador de la víctima. Estos, a su vez pueden secuestrar sesiones de usuario, modificar sitios web, o redirigir al usuario a sitios maliciosos.
- Si se expone una referencia a un objeto interno de la aplicación, tales como, un directorio o base de datos sin un control de accesos u otro tipo de protección, los atacantes pueden manipular estas referencias para acceder a datos no autorizados.
- Se requiere tener una configuración del entorno de ejecución segura, definida e implementada para la aplicación. Los servidores *web*, servidores de base de datos, y otras plataformas deberán superar un exhaustivo análisis de seguridad para prevenir posibles errores de configuración que puedan comprometer el sistema. Además, el software deberá mantenerse actualizado.
- Si no se protegen adecuadamente los datos sensibles, tales como tarjetas de crédito, números de identificación y credenciales de autenticación, los atacantes podrían robarlos o modificarlos para cometer fraude con las tarjetas de crédito, robo de identidad, u otros delitos. Los datos sensibles han de ser gozar de protección adicional, tales como el cifrado de almacenamiento o durante su transporte entre cliente y servidor o servidor y servidor.
- La mayoría de las aplicaciones web verifican los derechos de acceso del usuario antes de hacer visible funcionalidades en la interfaz. Sin embargo, es necesario realizar las mismas comprobaciones de control de acceso en el servidor cuando se entra a cada función. Si las solicitudes no se verifican, los atacantes podrían tener la capacidad de acceder a las funciones sin la debida autorización.
- Un ataque *CSRF (Cross-Site Request Forgery)* fuerza al navegador de la víctima que ha iniciado la sesión en una aplicación a enviar una petición *HTTP* que incluye los datos de sesión de la víctima y cualquier otra información de autenticación, a una aplicación web vulnerable. Esto permite al atacante utilizar el navegador de la víctima para generar solicitudes que la aplicación creerá que son legítimas, pudiendo tener acceso sin necesidad de haberse autenticado.

- Componentes, tales como librerías, *frameworks* y otros módulos de software, casi siempre se ejecutan con privilegios completos. Si un componente vulnerable es explotado, puede facilitar la fuga de datos importantes o toma de control del servidor. Las aplicaciones que utilizan componentes con vulnerabilidades conocidas pueden debilitar las defensas de aplicación y permitir una amplia gama de posibles ataques.
- Las aplicaciones *web* frecuentemente redirigen y reenvían a los usuarios a otras páginas y sitios. El uso de datos no confiables para determinar las páginas de destino y sin una validación adecuada pueden redirigir las víctimas a sitios de phishing o malware.

### 3.2.2. Limitaciones

En el desarrollo de la solución propuesta se han tenido en cuenta los riesgos identificados. Igualmente, se han tomado medidas para limitar su impacto. Aún así, hasta el mismo instante de la presentación del proyecto planteado, no existe una solución óptima y fiable para la autenticación de usuarios que permita evitar el riesgo de suplantación de identidad. Por este motivo no se podrá garantizar con total seguridad que el identificado como usuario de los elementos de este sistema propuesto es quien dice ser. Tampoco se puede establecer un método de cifrado, tanto de los datos como de las comunicaciones, que no pueda ser inutilizado o trampeado, por lo que no se podrá asegurar que los datos guardados en el sistema de almacenamiento propuesto puedan ser extraídos y utilizados de forma fraudulenta.

Por otro lado, al hacer uso de una plataforma de mensajería de un tercero, es posible que en cualquier momento se pueda experimentar una interrupción del servicio, dejando al sistema completamente inutilizable.

La solución propuesta al problema planteado al inicio de este trabajo no podrá garantizar ciertos aspectos recogidos en los requisitos que se expondrán en siguientes capítulos debido a limitaciones que no han sido objeto de profundización en el momento de su elaboración como la seguridad de la plataforma elegida para el desarrollo del *bot*.

---

# CAPÍTULO 4

## Prototipado

---

En este capítulo se recoge el proceso de diseño y desarrollo del prototipo de la solución propuesta en el capítulo anterior.

Primeramente, se recoge la especificación de los requerimientos que debe cumplir el prototipo en su versión inicial. A continuación, se realiza un estudio objetivo y justificado de las herramientas que se utilizan para su construcción. Tras todo ello, se describe el proceso de despliegue del prototipo en el Ayuntamiento de Quart de Poblet. También se describe el proceso de evaluación del mismo en un entorno real y, finalmente, su valoración una vez analizados todos los resultados obtenidos.

### 4.1 Especificación de Requerimientos

---

A continuación se recogen todos los requerimientos funcionales y no funcionales planteados. Los requerimientos funcionales se ocupan de lo que debe hacer el sistema o proporcionar a los usuarios. Estos incluyen la descripción de las funcionalidades requeridas. Por otro lado, los requerimientos no funcionales describen cómo, en qué nivel o qué estándar deben seguir las funcionalidades del sistema. Por ejemplo, los niveles de servicio requeridos, tales como tiempos de respuesta; requisitos de seguridad y de acceso; restricciones técnicas; si se requiere la interacción con los usuarios y otros sistemas; y las restricciones del proyecto.[20]

#### 4.1.1. Requerimientos funcionales

##### 1. RF-1: Autenticación

**Tipo:** Obligatorio.

**Descripción:** El sistema deberá implementar un sistema que permita la autenticación de los votantes contra un censo previamente establecido.

**Justificación:** Para que una votación sea válida, los votantes deben estar correctamente censados y el sistema debe de ser capaz de realizar esta comprobación.

##### 2. RF-2: Idioma

**Tipo:** Obligatorio.

**Descripción:** El sistema deberá poder utilizar diversos idiomas y darle opción al usuario para elegir en cual quiere utilizar el sistema. En un momento dado ha de ser posible incluir nuevos idiomas sin dificultad.

**Justificación:** El uso del sistema en lugares donde más de un idioma es oficial, hace necesario que pueda adaptarse a estos entornos.

### 3. RF-3: Validación

**Tipo:** Obligatorio.

**Descripción:** El sistema deberá poder validar la exactitud de un voto emitido, no aceptando aquellos votos que no cumplan las restricciones configuradas previamente y que podrán variar de una votación a otra.

**Justificación:** Cada tipo de votación requiere de unos requisitos específicos como, por ejemplo, el número de opciones que como máximo se pueden elegir en una votación de tipo multiselección.

### 4. RF-4: Verificación

**Tipo:** Obligatorio.

**Descripción:** El sistema deberá permitir al usuario la verificación de forma privada del voto emitido y validado por el sistema una vez haya sido correctamente guardado.

**Justificación:** De cara a garantizar la trazabilidad de un voto, el usuario podrá volver a consultar su voto directamente, garantizando en todo momento la privacidad.

### 5. RF-5: Asistencia

**Tipo:** Obligatorio.

**Descripción:** El sistema deberá guiar al usuario en cada uno de los procedimientos, ofreciendo instrucciones claras para cada uno de los pasos a ejecutar.

**Justificación:** Para facilitar las tareas que debe llevar a cabo el usuario y de cara a evitar problemas relacionados con la curva de aprendizaje de un sistema como el que se describe, será necesario implementar un sistema que proporcione soporte e instrucciones en cada momento de forma clara y concisa.

### 6. RF-6: Interoperabilidad

**Tipo:** Obligatorio.

**Descripción:** El sistema deberá poder integrarse fácilmente en entornos previos además de facilitar la incorporación de nuevas funcionalidades.

**Justificación:** De cara a permitir el crecimiento y adaptación del sistema, es necesario incluir mecanismos que faciliten la interoperabilidad así como pensar en un modelo de desarrollo modular que permita su extensión.

## 7. RF-7: Intervención y auditoría

**Tipo:** Obligatorio.

**Descripción:** Se debe implementar un mecanismo de auditoría del procedimiento. Para ello se permitirá que sendos interventores puedan validar el proceso.

**Justificación:** La intervención y control de una votación garantiza que el proceso no tenga fallos, ni errores, o proporcione resultados incoherentes y que además pueda recuperarse ante un fallo con pérdida de almacenamiento.

### 4.1.2. Requerimientos no funcionales

#### 1. RNF-1: Fiabilidad

**Tipo:** Obligatorio.

**Descripción:** El sistema implementado deberá garantizar la fiabilidad. Será requisito indispensable disponer de una segunda base de datos que mantenga una copia en tiempo real de, al menos, los votos emitidos así como todos aquellos usuarios que han emitido su voto.

**Justificación:** Un sistema de votación no puede, bajo ningún concepto, perder votos por fallos del mismo.

#### 2. RNF-2: Rendimiento

**Tipo:** Obligatorio.

**Descripción:** El rendimiento del sistema debe ser óptimo. Se debe asegurar en todo momento que el servidor nunca sufra pérdidas de memoria, saturación de la unidad de proceso, etc.

**Justificación:** Al tratarse de un sistema en tiempo real, es necesario que el rendimiento nunca caiga a márgenes no aceptables.

#### 3. RNF-3: Tiempo de respuesta

**Tipo:** Recomendable.

**Descripción:** Imitar la interacción humana requiere que el tiempo de respuesta entre cada interacción con el sistema sea de máximo 3 segundos.

**Justificación:** No es recomendable establecer un tiempo de respuesta que provoque la pérdida de inmediatez del sistema.

#### 4. RNF-4: Seguridad

**Tipo:** Obligatorio.

**Descripción:** Se deberán implementar protocolos seguros *HTTPS* en todas las comunicaciones que impliquen abandonar el entorno seguro del servidor.

**Justificación:** Para evitar en la medida de lo posible los diferentes ataques identificados en el punto Riesgos y limitaciones, será necesario implementar protocolos seguros para las transferencias de datos que salgan o entren hacia o desde el exterior (internet).

#### 5. RNF-5: Disponibilidad

**Tipo:** Obligatorio.

**Descripción:** Se deberá asegurar una disponibilidad del sistema de, al menos, un 90 %, asegurando que en caso de fallo del sistema, este se pueda recuperar en menos de un minuto.

**Justificación:** Para evitar problemas derivados de pérdida de acceso al sistema se debe garantizar una alta disponibilidad del sistema.

#### 6. RNF-6: Cifrado

**Tipo:** Obligatorio.

**Descripción:** Todos los elementos del sistema objeto de ser cifrados deberán cifrarse según el protocolo *AES-256*.

**Justificación:** Para garantizar un almacenamiento seguro de elementos sensibles se deberá realizar la cifrado mediante procedimientos estándares.

## 4.2 Diseño y construcción

---

### 4.2.1. Diseño

#### Etapas del sistema

##### Primera

Etapa inicial: Se requerirá al usuario que elija el idioma con el que desea interactuar con el *bot*.

##### Segunda

Etapa de bienvenida: El *bot*, tras saludar al usuario, le preguntará si desea autenticarse mediante el censo de votantes autorizados para votar.

##### Tercera

Etapa de autenticación: El *bot* solicitará el número del Documento Nacional de Identidad y lo cotejará con los datos censales. El usuario tendrá tres intentos para realizar este primer paso en la fase de autenticación.

##### Cuarta

Etapa de autenticación: En este caso el *bot* requerirá la fecha de nacimiento para cotejarla también con los datos censales. Dispondrá, también, de tres intentos para poder completar la autenticación con éxito.

### Quinta

Etapa de emisión de voto, en ella se informará al usuario con las instrucciones para emitir su voto, así como las opciones disponibles.

### Sexta

Etapa de verificación: Esta etapa es la última y tiene carácter opcional: el usuario podrá introducir el código de verificación proporcionado tras su emisión del voto, lo que le permitirá comprobar que su voto ha sido correctamente guardado.

## Comandos

Con el fin de facilitar la interacción del usuario con el *bot*, se han definido una serie de comandos reservados que el usuario podrá invocar en cualquier momento del proceso. A través de ellos se pretende que la comunicación entre el *bot* y el usuario sea más eficiente.

El formato de los comandos seguirá el siguiente esquema:

```
1 /<nombre_del_comando>
```

**código 4.1:** Esquema formato de los comandos.

El listado de los propuestos es el siguiente:

#### **/start**

En el caso de que el usuario se encuentre en la etapa inicial, o si este no ha entablado una conversación aún con el *bot*, le proporcionará instrucciones para la selección del idioma.

#### **/ayuda**

Este comando estará disponible durante las seis etapas del proceso y proporcionará información detallada al usuario de lo que debe hacer en cada momento durante su votación.

#### **/votar**

Con él iniciará el proceso de emisión del voto, únicamente estará disponible en la quinta etapa.

#### **/verificar**

Mediante el cual, el usuario podrá verificar un voto ya emitido. Este comando únicamente estará disponible en la sexta etapa.

#### **/acerca\_de**

Mostrará información sobre el *bot*.

## Procesador de mensajes

Para poder reaccionar ante los mensajes que el usuario envíe, se ha diseñado un sistema de procesamiento y clasificación de texto que permita identificar lo que el usuario quiere comunicar. Este sistema se basa en el uso de expresiones

regulares para reconocer patrones en dichos mensajes que pudieran contener una orden, un comando u otro tipo de elemento al que el *bot* debiera responder.

El algoritmo de procesamiento cuenta con la siguiente casuística:

### Caso 1

El mensaje contiene un comando reconocible. En este caso, el mensaje deberá contener una cadena de texto que obedezca a las restricciones del formato de comandos definidas (`/<nombre_del_comando>`). La expresión regular diseñada para tal objetivo es:

```
1 /(\\[a-z\_A-Z]+)/
```

**código 4.2:** Expresión regular del primer caso.

### Caso 2

El mensaje contiene la acción de pulsar un botón. Si el usuario presiona uno de los botones que el *bot* puede mostrar en la conversación, este mensaje contendrá dicha acción, representada por una cadena de caracteres que empieza como `butt_` e incluye un código identificador de dicho botón. La expresión regular planteada para poder detectar este tipo de mensaje es:

```
1 /(butt_)./
```

**código 4.3:** Expresión regular del segundo caso.

### Caso 3

El mensaje contiene un Documento Nacional de Identidad. En este caso, se detectará si el mensaje recibido corresponde con el formato de un DNI. La expresión regular que llevará a cabo dicha tarea será:

```
1 /^[0-9]{8}[TRWAGMYFPDXBNJZSQVHLCKET]{1}$/i
```

**código 4.4:** Expresión regular del tercer caso.

### Caso 4

El mensaje contiene un Número de Identificación de Extranjeros. Para detectar si el mensaje recibido se corresponde con un NIE, se utilizará la siguiente expresión regular:

```
1 /^[XYZ]{1}[0-9]{7}[TRWAGMYFPDXBNJZSQVHLCKET]{1}$/i
```

**código 4.5:** Expresión regular del cuarto caso.

### Caso 5

El mensaje contiene una fecha de nacimiento. En este caso no solo se deberá detectar que el mensaje recibido es una fecha de nacimiento, sino que también se realizará una validación de dicha fecha. La expresión regular diseñada para tal efecto es:

```
1 /^(?: (?: 31(\| - \| \.) (?: 0? [13578] | 1 [02])) \1 | (?: (?: 29 | 30)
  (\| - \| \.) (?: 0? [1, 3-9] | 1 [0-2]) \2) (?: (?: 1 [6-9] | [2-9] \d) ? \d
  {2}) $ | ^ (?: 29(\| - \| \.) 0? 2 \3 (?: (?: 1 [6-9] | [2-9] \d)
  ? (?: 0 [48] | [2468] [048] | [13579] [26])
  | (?: (?: 16 | [2468] [048] | [3579] [26]) 00))) $ | ^ (?: 0? [1-9] | 1 \d
  | 2 [0-8]) (\| - \| \.) (?: (?: 0? [1-9]) | (?: 1 [0-2]))
  \4 (?: (?: 1 [6-9] | [2-9] \d) ? \d {2}) $ /
```

---

**código 4.6:** Expresión regular del quinto caso.

### Caso 6

El mensaje contiene un voto. Para el caso de un voto, deberemos detectar si la cadena de caracteres es un conjunto de números enteros separados por comas. Para ello se utilizará la siguiente expresión regular:

```
1 /^(\d+(,\d+)*)?$/
```

**código 4.7:** Expresión regular del sexto caso.

### Caso 7

El mensaje contiene un código de verificación de voto. Finalmente, si el mensaje contiene una cadena de caracteres cuyo inicio es PASS la clasificaremos como código de verificación de voto. La expresión regular que cumple dicha tarea es:

```
1 /(PASS).+/?
```

**código 4.8:** Expresión regular del séptimo caso.

El algoritmo de clasificación estará obligado a comprobar cada uno de los mensajes, cotejarlo para ver si coincide con alguno de los casos descritos y proporcionar una interpretación. En el Código 4.9 se puede ver un extracto de dicho algoritmo.

```
1 function strip(text) {
2   var regex = /(\/[a-z\_A-Z]+)/;
3   var regex4 = /(butt_)./;
4   var regex5 = /^[0-9]{8}[TRWAGMYFPDXBNJZSQVHLCKET]{1}$/i;
5   var regex6 = /^[XYZ]{1}[0-9]{7}[TRWAGMYFPDXBNJZSQVHLCKET]{1}$/i;
6   var regex7 = /^(?:31(\/|-|\.) (?:0?[13578]|1[02]))
   \1|(?:29|30) (\/|-|\.) (?:0?[1,3-9]|1[0-2])\2)
   (?:1[6-9]|2-9\d)?\d{2})$|^ (?:29(\/|-|\.)
   0?2\3(?:1[6-9]|2-9\d)
   ?(?:0[48]|2468)[048]|13579[26])
   | (?:16|2468)[048]|3579[26])00))$|^ (?:0?[1-9]|1\d
   |2[0-8]) (\/|-|\.) (?:0?[1-9])|(?:1[0-2]))
   \4(?:1[6-9]|2-9\d)?\d{2})$/;
7   var regex8 = /(\d+(,\d+)*)?$/;
8   var regex9 = /(PASS).+/?;
9   var array = text.split(" ");
10  var vote = text.replace(/ /g, ',');
11
12
13  var matching = regex.test(array[0]);
14  var matching4 = regex4.test(array[0]);
15  var matching5 = regex5.test(array[0].toString().toUpperCase());
16  var matching6 = regex6.test(array[0].toString().toUpperCase());
17  var matching7 = regex7.test(array[0]);
18  var matching8 = regex8.test(vote);
19  var matching9 = regex9.test(array[0]);
20
```

```
21
22  if (matching) {
23      return {command: array[0], type: 1};
24  } else if (matching4) {
25      return {command: array[0], type: 4};
26  } else if (matching5 && validate(text)) {
27      return {command: array[0], type: 5};
28  } else if (matching6 && validate(text)) {
29      return {command: array[0], type: 6};
30  } else if (matching7) {
31      return {command: array[0], type: 7};
32  } else if (matching8){
33      return {command: vote, type: 8};
34  } else if (matching9){
35      return {command: array[0], type: 9};
36  }
37  else return false;
38
39 }
```

**código 4.9:** Función de clasificación de texto.

## Flujo de usuario

En la Figura 4.1 se puede ver el esquema básico que deberá seguir la interacción del usuario con el *bot*. De esta manera se pueden distinguir todas las fases anteriormente descritas, desde el punto de vista del votante.

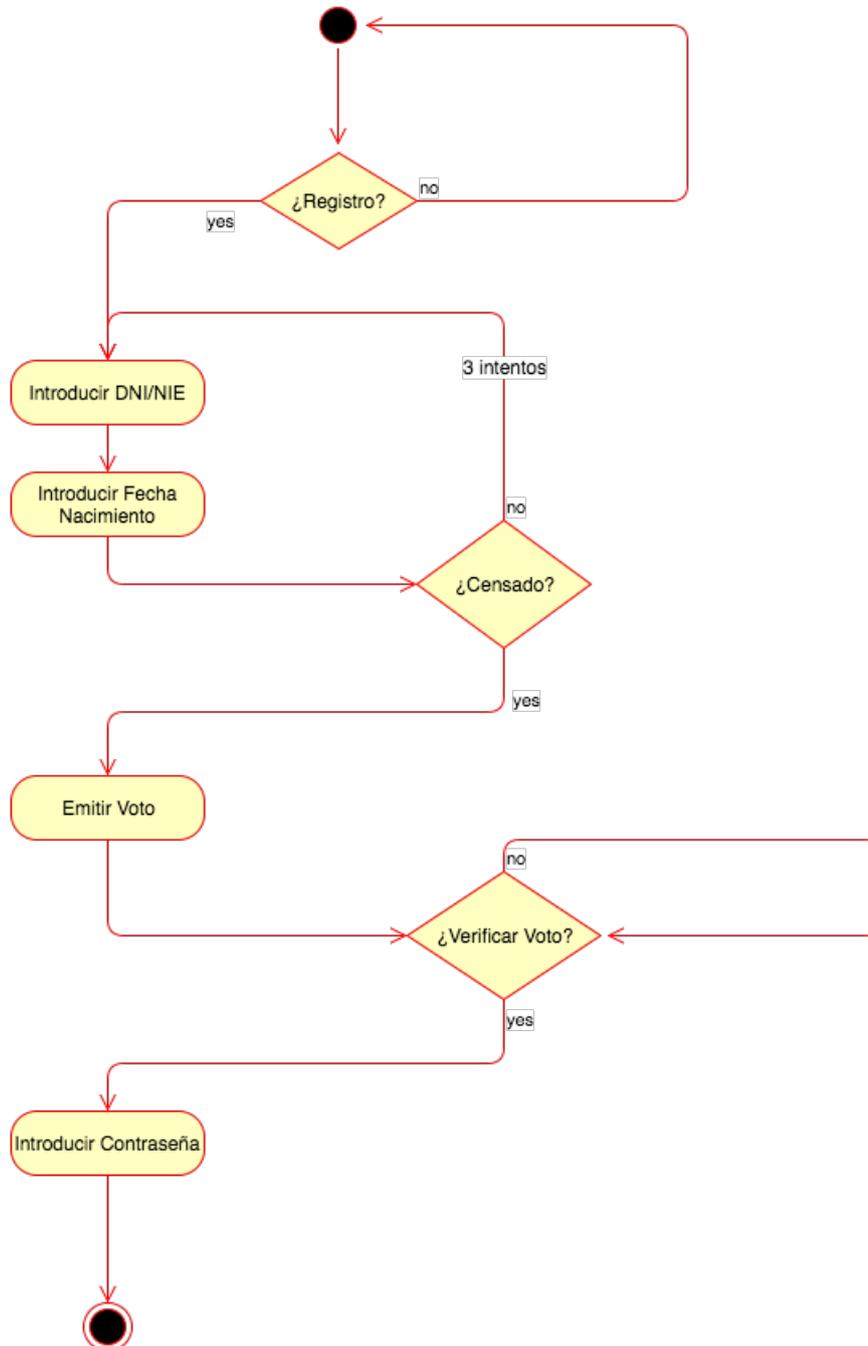


Figura 4.1: Flujo de usuario del *bot*.

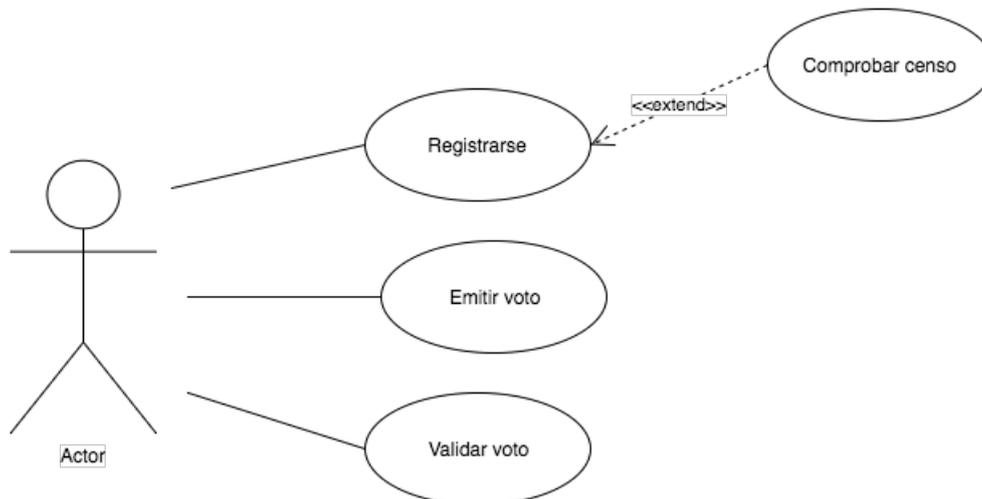
El primer elemento del esquema representa el árbol de decisión que se implementará para iniciar el registro del usuario. En caso de que este decida iniciar el registro, se procederá al siguiente paso. En caso contrario, se regresará al inicio, permitiendo así que el usuario pueda cambiar de opinión con respecto a este paso.

A continuación el usuario introducirá los datos requeridos para su autenticación, en este caso DNI/NI o Fecha de nacimiento. Dispondrá de tres intentos para cumplimentar correctamente esta fase, en caso contrario el usuario quedará bloqueado.

Una vez autenticado satisfactoriamente, el usuario podrá emitir su voto y, tras ello, verificarlo tantas veces como quiera.

### Casos de uso

En la Figura 4.2 se representan los casos de uso del sistema.

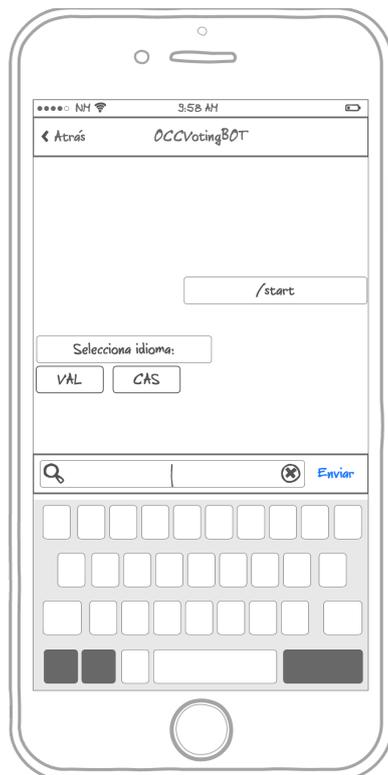


**Figura 4.2:** Casos de uso.

Como se puede ver, existen tres casos de uso: el registro, la emisión del voto y la validación de un voto emitido.

## Bocetos

A continuación, en las siguientes figuras, se muestran los bocetos que representan una idea básica de los mensajes y respuestas que el *bot* dará para cada etapa del proceso de votación.



**Figura 4.3:** Boceto del inicio de la conversación.



**Figura 4.4:** Boceto del mensaje de bienvenida.



Figura 4.5: Boceto del inicio de la autenticación.



Figura 4.6: Boceto de la primera fase de autenticación.



Figura 4.7: Boceto de la segunda fase de autenticación.



Figura 4.8: Boceto de la fase de emisión de voto.



Figura 4.9: Boceto de la fase de selección de opciones.



Figura 4.10: Boceto del inicio del proceso de verificación de voto.



Figura 4.11: Boceto del envío del código para verificación.

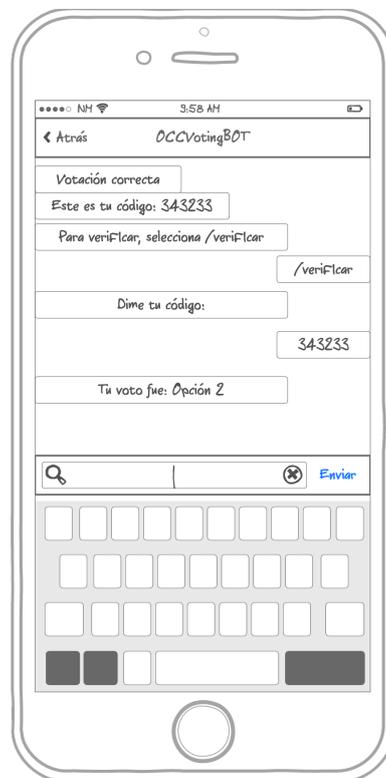


Figura 4.12: Boceto de confirmación de voto verificado.

## 4.2.2. Herramientas

Para implementar este proyecto se han tenido que utilizar diversas herramientas, cuidadosamente elegidas como en cualquier otra obra, junto a otras para darle soporte y que apoyen su correcto desarrollo. Todas ellas se describen a continuación, incluyendo la justificación de su elección.

### 1. Lenguaje de desarrollo

#### Descripción

El lenguaje elegido para el desarrollo de la solución propuesta es *NodeJS*<sup>1</sup>, un entorno de ejecución de código abierto y multiplataforma para el desarrollo de aplicaciones web en el lado servidor. Más concretamente, se hace uso de un *framework* basado en *NodeJS* llamado *SailsJS*<sup>2</sup>, diseñado para emular el patrón Modelo, Vista, Controlador, con una arquitectura escalable, orientada a servicios. *SailsJS*, además, incluye un motor de almacenamiento y recuperación de datos compatible con la mayoría de sistemas de gestión de bases de datos del mercado. Este sistema proporciona una API uniforme para acceder a elementos de datos, lo que quiere decir que se escribe el mismo código para obtener y almacenar datos, sea cual sea la tecnología de almacenamiento implementada.

Por otro lado, *SailsJS* ofrece seguridad y control de acceso basado en roles por defecto mediante el uso de políticas: funciones de middleware simples y reutilizables que se ejecutan antes que sus acciones.

#### Justificación

*NodeJS* es rápido y escalable, a altos niveles de concurrencia (miles de conexiones) el servidor tendrá que ejecutar instrucciones de forma asíncrona y sin bloqueos, por lo que no será posible ir creando un hilo de ejecución para cada conexión. Otros lenguajes como Java no reúnen este tipo de requisitos. Sin embargo *NodeJS* los gestiona muy eficientemente.

Además el hecho de haber elegido el *framework SailsJS* facilita en gran medida el proceso de desarrollo, pues la flexibilidad que éste tiene para la gestión de la persistencia y la seguridad es algo que reducirá la dificultad del trabajo a realizar.

### 2. Librerías

Las librerías y módulos <sup>3</sup> utilizados como apoyo al desarrollo del sistema se listan a continuación:

**crypto** El módulo *crypto* proporciona funcionalidad criptográfica que incluye un conjunto de *wrappers* para el hash de *OpenSSL*, *HMAC*, cifrado, descifrado

<sup>1</sup>Puede consultarse información sobre *NodeJS* en el sitio web del proyecto. Disponible en: <https://nodejs.org/> [Acceso el 28.06.2016]

<sup>2</sup>Puede consultarse información sobre *SailsJS* en el sitio web del proyecto. Disponible en: <http://sailsjs.org/> [Acceso el 28.06.2016]

<sup>3</sup>Puede consultarse información sobre todas las librerías en el sitio web del gestor de dependencias de *NodeJS* (npm). Disponible en: <https://www.npmjs.com/> [Acceso el 28.06.2016]

y funciones de firma y verificación. Este módulo será el encargado de las tareas de cifrado y descifrado de los elementos sensibles almacenados en la base de datos.

**form-data** Se trata de una librería para crear streams del tipo *multipart/form-data* legibles. Se encargará de componer y empaquetar los objetos enviados mediante peticiones *HTTP*.

**generate-password** Librería que permite generar contraseñas únicas de forma aleatoria. Estas contraseñas serán las usadas para el cifrado de los datos del votante que se almacenarán junto con el voto para permitir a éste la verificación de su voto en cualquier momento, preservando su anonimato.

**https** Librería para la gestión de las peticiones mediante protocolo seguro *HTTPS*.

**include-all** Módulo para la gestión de dependencias software del proyecto.

**moment** Librería para la gestión eficiente de fechas.

**newrelic** Librería para la recolección de datos relacionados con el rendimiento del sistema.

**node-emoji** Módulo que permite gestionar emoticonos.

**request** Cliente para el manejo de peticiones *HTTP*.

**sails-mongo** Módulo de integración del sistema de gestión de base de datos *MongoDB* en *SailsJS*.

### 3. Persistencia

#### Descripción

La implementación de la capa de persistencia se realiza mediante un sistema de bases de datos no relacionales, concretamente *MongoDB*<sup>4</sup>, un sistema de bases de datos *NoSQL* orientado a documentos y de código abierto. *MongoDB* permite una fragmentación transparente entre diferentes réplicas del servidor de datos, disminuyendo el tiempo de respuesta del servidor ante un número considerable de consultas a la base de datos.

#### Justificación

*MongoDB* almacena los datos utilizando un modelo de datos basado en documentos flexibles similares a *JSON*. Los documentos contienen uno o más campos, incluyendo matrices, datos binarios y sub-documentos. Los campos pueden variar de un documento a otro. Esta flexibilidad permite evolucionar el modelo de datos con rapidez a medida que cambien los requisitos de aplicación.

Además, puesto que se prevé que el sistema genere un número bastante considerable de accesos a la base de datos, se hace indispensable que esta sea capaz de manejar todo ese volumen de accesos con un tiempo de respuesta que no merme ante picos de trabajo.

---

<sup>4</sup>Puede consultarse información sobre *MongoDB* en el sitio web del proyecto. Disponible en: <https://www.mongodb.com/> [Acceso el 28.06.2016]

## 4. Entorno de ejecución

### Descripción

Como entorno en el que se despliega y ejecuta el sistema, se ha optado por una solución basada en la nube como es el caso de *Heroku*<sup>5</sup>, una plataforma de ejecución de aplicaciones orientada a servicios. Se trata de un entorno sencillo en el que no se requiere el control de la infraestructura, pues la escalabilidad se gestiona automáticamente. Además, permite el despliegue de nuevas versiones de la aplicación de forma casi instantánea.

### Justificación

Puesto que el sistema exige una alta disponibilidad (RNF-5) así como otros requisitos relacionados con el entorno de ejecución, se ha apostado por una solución basada en la nube para la ejecución del servidor. Este tipo de entornos nos permite garantizar que el sistema dispondrá de una capacidad de trabajo acorde a lo que se requiera en cada momento, ahorrando tiempo y costes de mantenimiento.

## 5. Comunicaciones y notificaciones

### Descripción

Para el envío de notificaciones y cualquier otro tipo de comunicaciones relacionadas con el sistema y, más concretamente, el envío de correo electrónico de forma masiva, se ha optado por *SendGrid*<sup>6</sup>, un servicio integrado de envío de correo transaccional que permite una alta fiabilidad a la hora de garantizar la entrega de dichos correos. Además incorpora una *API* que permite integrar el sistema desarrollado de una forma fácil y transparente.

### Justificación

Para garantizar el cumplimiento del requisito que establece la necesidad de auditoría de la votación (RF-7) se ha optado por remitir copia de cada uno de los votos al interventor o interventores.

Esto obliga a disponer de un sistema fiable de entrega de correos electrónicos capaz de manejar un alto número de solicitudes de envío simultáneas. Lo que no sería posible sin disponer de un servicio especializado, pues un servidor de envío local podría saturarse fácilmente.

## 6. Aseguramiento de la calidad

### Descripción

Para la implementación de test unitarios así como generar los informes de los mismos, se utilizarán *frameworks* de testing para *NodeJS*, que permitan asegurar la calidad del desarrollo en cada una de las modificaciones que se

---

<sup>5</sup>Puede consultarse información sobre *Heroku* en el sitio web del proyecto. Disponible en: <https://www.heroku.com/> [Acceso el 28.06.2016]

<sup>6</sup>Puede consultarse información sobre *SendGrid* en el sitio web del proyecto. Disponible en: <https://sendgrid.com/> [Acceso el 28.06.2016]

realicen durante el proceso de desarrollo e integración. Concretamente los *frameworks Mocha*<sup>7</sup> y *Supertest*<sup>8</sup>.

### Justificación

Mocha permite la creación de test unitarios así como la generación de informes detallados de cada uno de ellos, lo que permitirá poder mantener un código sin errores y capaz de evolucionar fácilmente. Por otro lado, Supertest permite probar de forma general, pero a la vez precisa, los puntos de las interfaces (API) desarrollados en el sistema. De esta manera, se podrá siempre asegurar que antes de entregar una versión del aplicativo, este responderá adecuadamente.

## 7. Integración Continua

### Descripción

Para facilitar la integración continua y el desarrollo ágil del sistema, se utiliza un servicio que automatice esta tarea. Concretamente se ha elegido *Travis CI*<sup>9</sup>, un entorno que permite la construcción y la ejecución de test, previamente al despliegue final, de aplicaciones. Además, este sistema permite el despliegue automático de nuevas versiones del software si este pasa todas las pruebas establecidas, disponiendo de forma rápida de una versión lista para ser probada.

### Justificación

*Travis CI* permite una configuración fácil y rápida de flujos de integración continua, permitiendo automatizar todo el proceso de construcción, pruebas y despliegue de cada una de las modificaciones que se vayan incluyendo en el proyecto.

## 8. Control de versiones

### Descripción

Para la tarea de control de versiones así como de repositorio de código se ha decidido utilizar *GitHub*<sup>10</sup>, una plataforma de desarrollo colaborativo de software que, a su vez, hace las funciones de repositorio utilizando el sistema de control de versiones *Git*.

### Justificación

La elección de *GitHub* como herramienta principal de gestión de versionado del código se hace atendiendo a la capacidad que tiene este entorno para poder no solo mantener de forma eficiente las actualizaciones y mejoras del código, si no también a las funcionalidades para crear un entorno de

---

<sup>7</sup>Puede consultarse información sobre Mocha en el sitio web del proyecto. Disponible en: <https://mochajs.org/> [Acceso el 28.06.2016]

<sup>8</sup>Puede consultarse información sobre Supertest en el sitio web del proyecto. Disponible en: <https://github.com/visionmedia/supertest> [Acceso el 28.06.2016]

<sup>9</sup>Puede consultarse información sobre Travis CI en el sitio web del proyecto. Disponible en: <https://travis-ci.org/> [Acceso el 28.06.2016]

<sup>10</sup>Puede consultarse información sobre GitHub en el sitio web del proyecto. Disponible en: <https://github.com/> [Acceso el 28.06.2016]

desarrollo colaborativo en la que cualquier persona pudiera tanto auditar como colaborar en la mejora continua del código fuente.

## 9. Gestión de tareas e incidencias

### Descripción

Para la gestión de tareas e incidencias se hace uso de la herramienta *JIRA*<sup>11</sup>, una aplicación basada en web para el seguimiento de errores e incidentes así como para la gestión operativa de proyectos. Este entorno permite operar con distintas aplicaciones como *Travis CI*, para la integración continua y *GitHub* para poder obtener información acerca de las versiones y cambios que ha habido en el código. También hará las funciones de gestión de proyectos, estableciendo las tareas planificadas y su duración para un correcto planteamiento del proyecto.

### Justificación

*JIRA* es ampliamente utilizada por la industria, habiendo demostrado sus cualidades como potente entorno de gestión de incidencias y proyectos. Sus integraciones con otras aplicaciones de gestión de proyectos y herramientas para el desarrollo permiten crear un entorno de organización optimizado para el trabajo en equipo, de forma deslocalizada y ágil.

## 10. Gestión documental

### Descripción

Para la gestión de todos los documentos que se generen durante el proceso de desarrollo del proyecto se utiliza *Confluence*<sup>12</sup>, un gestor de información integrado dentro de *JIRA* y que permite la creación colaborativa de documentos. Incluye además un buen número de plantillas prediseñadas para la generación de informes, documentos de trabajo, diario de reuniones, etc.

### Justificación

*Confluence* permite crear un repositorio documental en la nube, accesible en cualquier momento y con un entorno de edición amigable y cómodo. Además permite la creación de espacios de gestión documental asociados a proyectos, por lo que se puede mantener una correcta estructuración de la documentación generada.

## 11. Rendimiento

### Descripción

Para la medición del rendimiento del sistema implementado, se utiliza *NewRelic*<sup>13</sup>, un entorno de obtención de medidas de rendimiento muy comple-

---

<sup>11</sup>Puede consultarse información sobre *JIRA* en el sitio web del proyecto. Disponible en: <https://es.atlassian.com/software/jira> [Acceso el 28.06.2016]

<sup>12</sup>Puede consultarse información sobre *Confluence* en el sitio web del proyecto. Disponible en: <https://es.atlassian.com/software/confluence> [Acceso el 28.06.2016]

<sup>13</sup>Puede consultarse información sobre *NewRelic* en el sitio web del proyecto. Disponible en: <https://newrelic.com/> [Acceso el 28.06.2016]

to, que permite la obtención de información sobre el tiempo que el sistema dedica a procesar peticiones, a realizar consultas a la base de datos, etc.

**Justificación**

*NewRelic* es uno de los entornos más avanzados del mercado para la medición de rendimiento de aplicaciones web entre otras: no solo permite la obtención de analíticas en tiempo real, si no que detecta potenciales problemas que pudieran provocar una caída o fallo del sistema.

### 4.2.3. Arquitectura

Los componentes de la arquitectura del sistema son los que se detallan a continuación.

#### 1. Modelo de datos

La Figura 4.13 muestra el esquema del modelo de datos. En él se representan cada una de las colecciones de datos, así como los atributos de cada una de estas colecciones.

##### Colección Admin

La colección Admin representa a los usuarios con privilegios de administración sobre el sistema de votación. Los atributos de esta colección son:

- **id:** Representa el identificador único de cada uno de los elementos de la colección. Tendrá la función de clave primaria.
- **email:** Representa el correo electrónico utilizado para iniciar sesión como administrador en el sistema.
- **encrypted\_password:** Representa la contraseña del administrador, cifrada para evitar problemas de robo de identidad.
- **isLogged:** Representa el estado de la sesión del usuario, es decir, si se encuentra iniciada o no.
- **status:** Representa el estado actual del usuario, conectado, ausente o no disponible.
- **deleted:** Establece si el usuario ha sido borrado o no, de esta manera, al usar un borrado lógico, podrán recuperarse usuarios borrados por accidente.
- **project\_id:** Representa el código identificador del proyecto de votación que el usuario haya creado.

##### Colección Project

La colección Project representa una votación dentro del sistema mediante todos los datos necesarios para que esta pueda llevarse a cabo. Los atributos de esta colección son:

- **id:** Representa el identificador único del proyecto. Tendrá la función de clave primaria.
- **telegram\_token:** Representa el token de autorización para poder enviar peticiones a la API de *Telegram*.
- **server\_url:** Representa la url a la que el servidor de *Telegram* debe redirigir todos los nuevos mensajes que lleguen al *bot* desde un cliente de usuario.
- **voting\_type:** Representa el tipo de votación que se desea.
- **census\_activation:** Establece si la opción de comprobación del censo se encuentra activada o no.

- `intervention_email`: Es el correo al que se remite copia de cada uno de los votos emitidos para poder comprobar que no ha habido ninguna alteración de la base de datos.

### Colección Token

La colección Token incluye los datos necesarios para que un usuario administrador pueda realizar tareas dentro del sistema. Mediante esta colección se implementa el sistema de seguridad y control de accesos al servidor, que basa su razón de ser en el uso de tokens que permitirán (si el token es correcto) el acceso a funcionalidades restringidas del sistema. Los atributos de esta colección son:

- `user_id`: Representa el id de usuario de la tabla Admin.
- `token`: Representa el token de sesión del usuario para poder validarlo.
- `isValid`: Representa si el token es válido o no. En caso de no serlo el sistema no permitirá ejecutar las tareas restringidas a usuarios administradores.
- `ip`: Representa la dirección IP desde donde el usuario administrador ha iniciado la sesión.
- `agent`: Representa el agente (navegador, terminal móvil, etc.) desde el que el usuario administrador ha iniciado sesión.

### Colección Chats

La colección Chats se compone de todas y cada una de las conversaciones con las que el *bot* está manteniendo una conversación. Sus atributos son:

- `id`: Representa el identificador del chat en cuestión.
- `type`: Representa el tipo de conversación que se está manteniendo (grupo, chat privado, etc.).
- `title`: Representa el título de la conversación (solo para los casos en los que el tipo de conversación sea un chat grupal).
- `username`: Representa el nombre de usuario con el que se está manteniendo la conversación.
- `first_name`: Representa el nombre del usuario con el que se está manteniendo una conversación.
- `last_name`: Representa el apellido del usuario con el que se está manteniendo una conversación.
- `messages`: Representa la colección de mensajes que existen en la conversación en cuestión.
- 

### Colección Messages

La colección mensajes representa a cada uno de los mensajes intercambiados entre el *bot* y el usuario. Sus atributos son:

- `id`: Representa el identificador del mensaje.
- `from`: Representa el origen del mensaje (usuario).

- **date:** Representa la fecha en la que se originó el mensaje.
- **chat:** Representa el chat al que pertenece el mensaje en cuestión.
- **photo:** Representa si el mensaje en cuestión se trata de una foto.
- **update:** Representa el objeto entrante recibido desde los servidores de *Telegram* al que pertenece el mensaje en cuestión.

### Colección Update

La colección Update representa a todos y cada uno de los elementos que los servidores de *Telegram* remiten al servidor del *bot*. Sus atributos son:

- **id:** Representa el identificador único del elemento.
- **message:** Representa el mensaje incorporado dentro de este elemento.

### Colección Users

La colección Users representa a los usuarios que han iniciado una conversación con el *bot*. Sus atributos son:

- **id:** Representa el identificador único del usuario.
- **username:** Representa el nombre de usuario.
- **messages:** Representa a los mensajes remitidos al *bot* por el usuario en cuestión.
- **retry\_nid:** Representa el número de intentos que ha realizado el usuario para validar su número de identificación (DNI/NIF) contra el censo.
- **retry\_birth\_date:** Representa el número de intentos que ha realizado el usuario para validar su fecha de nacimiento contra el censo.
- **encrypted\_vote:** Representa el voto encriptado del usuario.

### Colección Stages

La colección Stages permite al *bot* llevar el seguimiento de las etapas en las que se encuentra cada uno de los usuarios. Sus atributos son:

- **user\_id:** Representa el identificador del usuario.
- **stage:** Representa el número de la etapa en la que el usuario se encuentra en un momento dado.
- **banned:** Representa si el usuario ha sido bloqueado tras haber agotado los intentos de validación contra el censo.
- **valid:** Representa si el usuario ha sido validado correctamente contra el censo.
- **has\_voted:** Representa si el usuario ha votado.
- **locale:** Representa el idioma elegido por el usuario.

### Colección Census

La colección Census representa el censo de votantes autorizados del proceso de votación. Sus atributos son:

- **nid:** Número de identificación, ya sea un Documento Nacional de Identidad o un Número de Identificación de Extranjeros.

- `birth_date`: Representa la fecha de nacimiento del votante.
- `sex`: Representa el sexo del votante.
- `address`: Representa la dirección del votante.
- `name`: Representa el nombre del votante.
- `surname1`: Representa el primer apellido del votante.
- `surname2`: Representa el segundo apellido del votante.

### Colección Status

La colección Status representa la información de todos los usuarios que están preparados para emitir un voto o si ya han emitido este. Sus atributos son:

- `nid`: Número de identificación (DNI/NIE) del votante.
- `telegram_id`: Identificador de usuario en *Telegram*.
- `has_voted`: Representa si el usuario ha votado.
- `user_type`: Representa el tipo de usuario.
- `encrypted_vote`: Representa el voto cifrado que el usuario ha emitido.

### Colección Votes

La colección Votes almacena todos los votos emitidos satisfactoriamente junto con el identificador de usuario, cifrado. Sus atributos son:

- `id`: Representa el identificador del voto en cuestión.
- `vote`: Representa el voto emitido.
- `encrypted_user_id`: Representa el identificador de usuario (DNI/NIE) cifrado.

### Colección Options

La colección Options representa todas las opciones disponibles que el votante tendrá para poder elegir su voto. Sus atributos son:

- `id`: Identificador único de la opción.
- `text`: Texto o descripción de la opción.

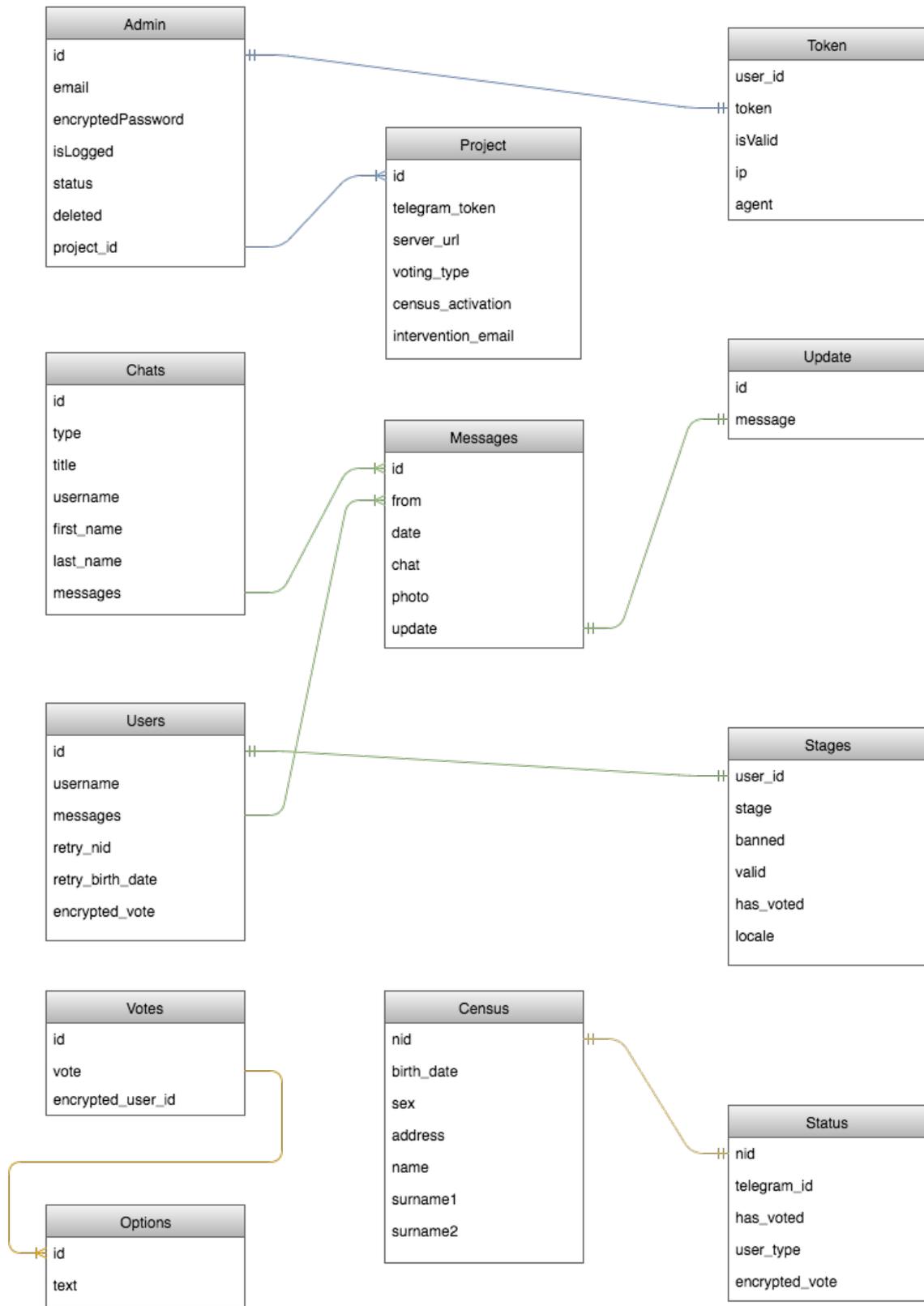


Figura 4.13: Modelo de datos.

## 2. Esquema de integración

La Figura 4.14 muestra el esquema general del sistema desde un punto de vista externo. En el se pueden distinguir todos los elementos que lo componen.

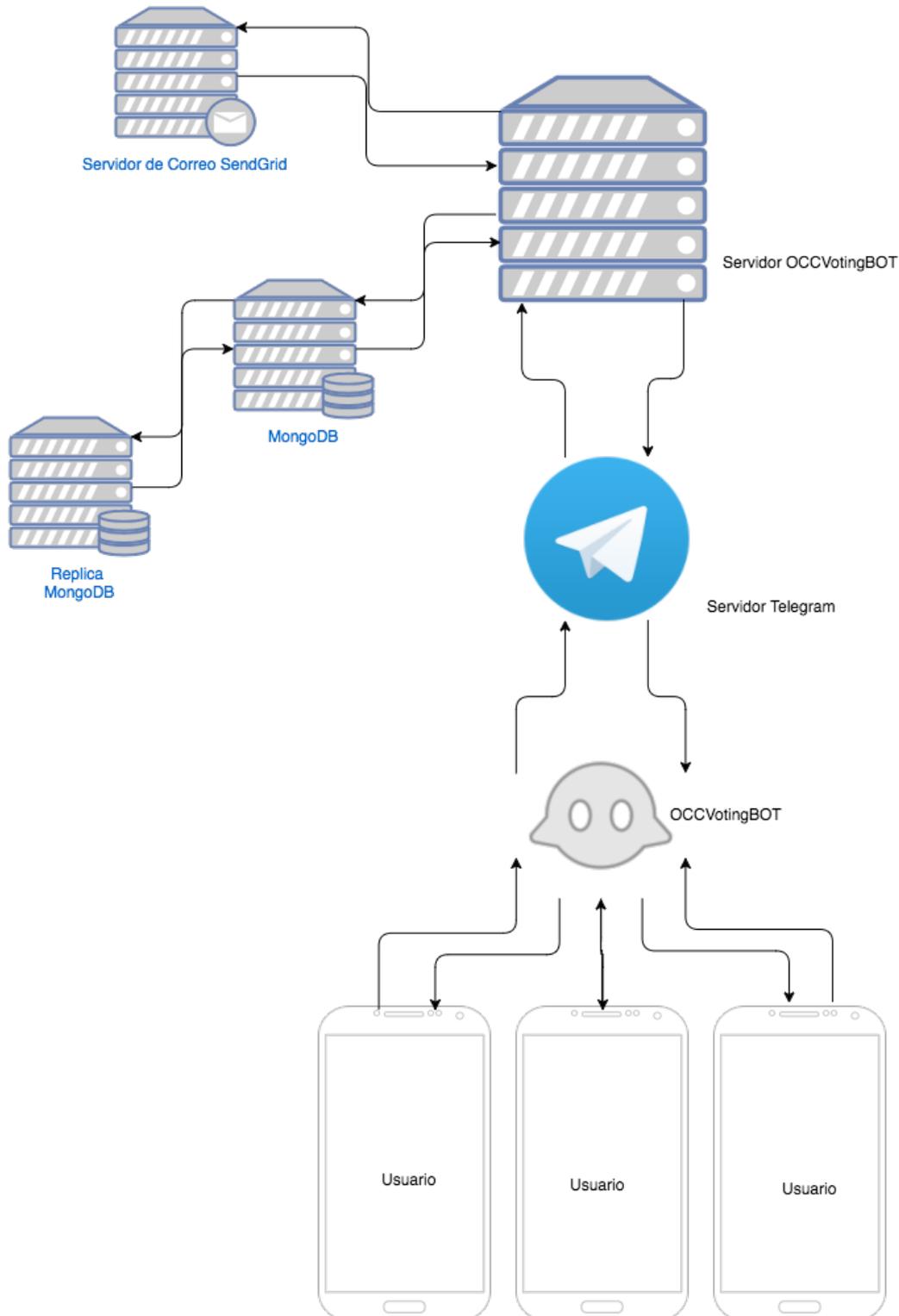


Figura 4.14: Esquema de integración.

### **OCCVotingBOT**

Es el 'contacto' dentro de la aplicación *Telegram* que nos permite interactuar con el *bot*

### **Servidor *Telegram***

Se trata de los servidores de la aplicación *Telegram*. Se encarga de reenviar todos los mensajes recibidos por *OCCVotingBOT* al servidor del sistema.

### **Servidor OCCVotingBOT**

Es el servidor del sistema, se encarga de toda la funcionalidad de la votación. Procesará todos los mensajes reenviados por los servidores de *Telegram* y responderá en función de lo recibido.

### **Servidor de correo SendGrid**

Se trata del servidor externo de correo electrónico, encargado de componer un correo electrónico por cada voto recibido y remitirlo a los interventores de la votación.

### **MongoDB**

Es el servidor de base de datos, en el se almacenará toda la información necesaria.

### **Replica MongoDB**

Se trata de una réplica en tiempo real del servidor de base de datos, localizado en otro entorno para evitar problemas de pérdida de información.

## **3. Flujo del sistema**

La Figura 4.15 representa el esquema interno del sistema. En ella se puede ver cómo el sistema procesa cada mensaje recibido por el usuario y genera una respuesta.

Cada vez que el servidor reciba un mensaje, este será procesado de la siguiente manera:

1. Se identificará al usuario que ha remitido el mensaje y, a continuación, se buscará en la base de datos para comprobar si este ya hubiera interactuado antes con el sistema. En caso negativo, se creará un nuevo usuario para futuras búsquedas.
2. Una vez identificado, se comprobará la etapa del proceso en la que el usuario se encuentre.
3. En función de la etapa en la que este se encuentre, el sistema activará unas funciones y otras. En todo caso, siempre se podrán ejecutar todos y cada uno de los comandos definidos. Sin embargo, estos comandos variarán su respuesta en función de la etapa anteriormente descrita.
4. Finalmente, ante cada etapa correctamente completada, el sistema incrementará el número de la etapa para que el usuario pueda progresar a la siguiente.

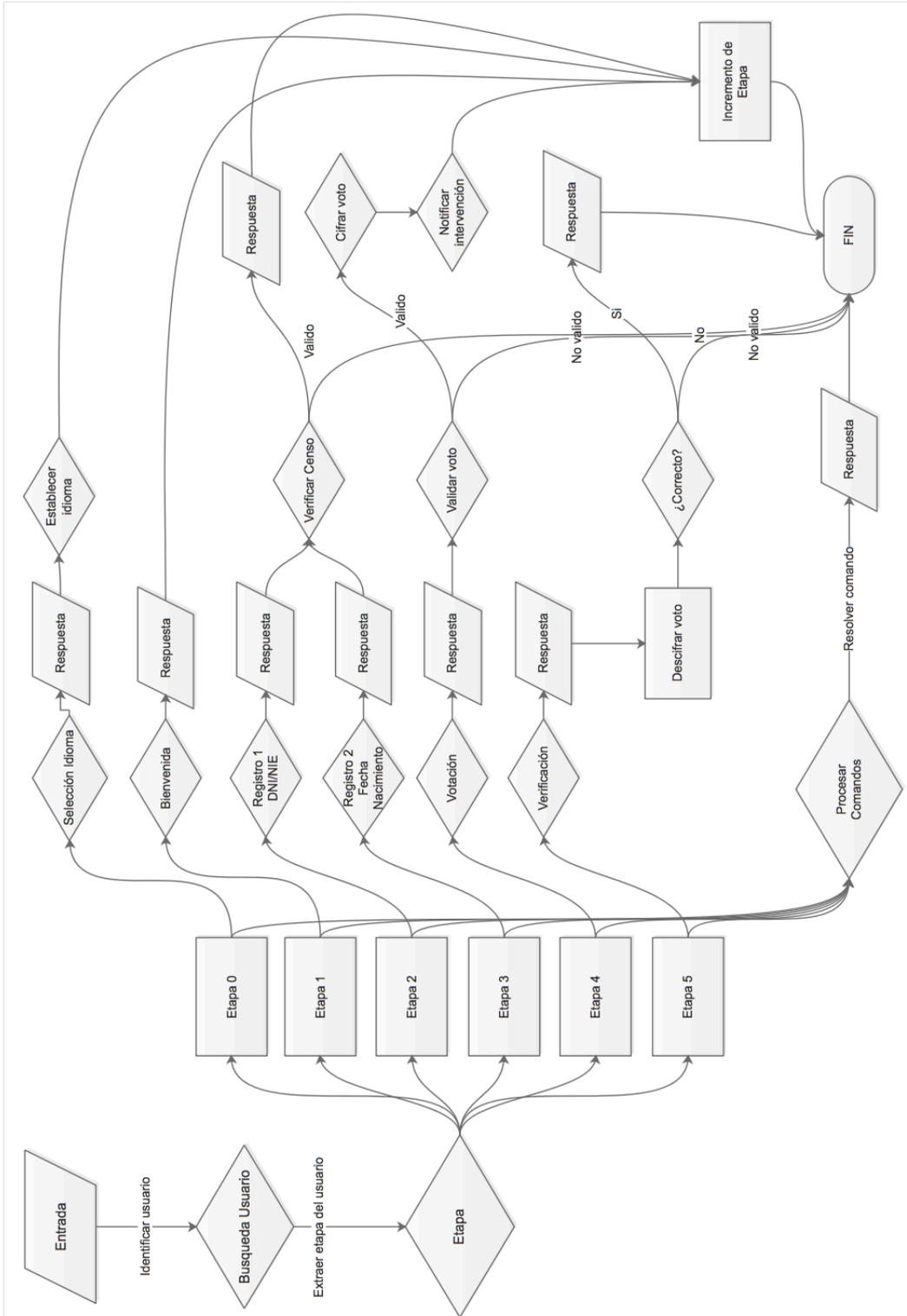


Figura 4.15: Flujo del sistema.

## 4.3 Implantación del prototipo

---

Una vez construido el prototipo, se ha procedido a implantarlo en un entorno real: el Ayuntamiento de Quart de Poblet (Valencia), llevándose a cabo en tres fases distintas:

### 4.3.1. Fase 1

La primera fase del proceso de implantación del prototipo tiene como objetivo el de organizar a todas las partes implicadas, pues es necesario acordar los aspectos técnicos a tener en cuenta durante dicha implantación.

Durante esta fase se especificaron los requisitos técnicos del entorno donde se iba a proceder a desplegar el aplicativo, así como la revisión de los requisitos no funcionales que el sistema requería para su correcto funcionamiento.

Además, se analizaron los potenciales problemas de seguridad y las medidas a tomar para evitarlos, prestando especial atención al cumplimiento de la Ley de Protección de Datos.

Los detalles acordados se listan a continuación:

1. La instalación de las bases de datos que contengan datos personales o que sean objeto de la Ley Orgánica sobre Protección de Datos se realizará en los servidores del Ayuntamiento de Quart de Poblet, el resto de bases de datos, así como la ejecución de la aplicación se realizará en la nube.
2. Cualquier colección de datos que pudiera contener información de carácter personal, no podrá abandonar bajo ningún concepto los servidores municipales.
3. El acceso a la colección de datos que almacena los votos emitidos, estará completamente restringida durante todo el proceso de votación.
4. Se establecerán procedimientos de monitorización del entorno para detectar cualquier anomalía.

### 4.3.2. Fase 2

Una vez coordinado todo el equipo y establecidos todos los parámetros técnicos, se inicia la implantación propiamente dicha. En primer lugar se llevaron a cabo las tareas de instalación de los requisitos software del sistema, junto con todas las dependencias requeridas por la aplicación para poder entrar en funcionamiento.

Se prepararon ambas bases de datos, la principal y una réplica en tiempo real, en distintas máquinas y con conexiones a internet distintas.

### **4.3.3. Fase 3**

Finalizada la fase de instalación y puesta en marcha del prototipo, durante la siguiente fase se llevaron a cabo las pruebas de estrés para garantizar que todo el sistema era estable y así evitar problemas durante la apertura al público del mismo. Concretamente se llevaron a cabo pruebas de rendimiento, enviando al servidor tres mil peticiones por minuto para comprobar que era capaz de soportar dicha carga y obteniendo un resultado satisfactorio.

Una vez confirmada la estabilidad del entorno, se cerró esta fase, habilitando el acceso al público.

## 4.4 Evaluación

### 4.4.1. Prototipo final

En las siguientes figuras se recoge el funcionamiento final del prototipo. Estas representan el *bot* tal y como lo puede ver los usuarios.



Figura 4.16: Imagen de la pantalla principal del *bot*.

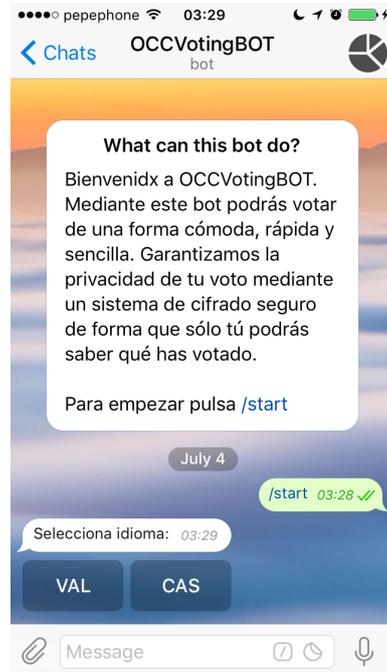


Figura 4.17: Imagen de la selección de idioma.



Figura 4.18: Imagen de la pantalla de bienvenida.



Figura 4.19: Imagen del inicio del proceso de registro.



Figura 4.20: Imagen de la primera fase de autenticación.



Figura 4.21: Imagen de la segunda fase de autenticación.

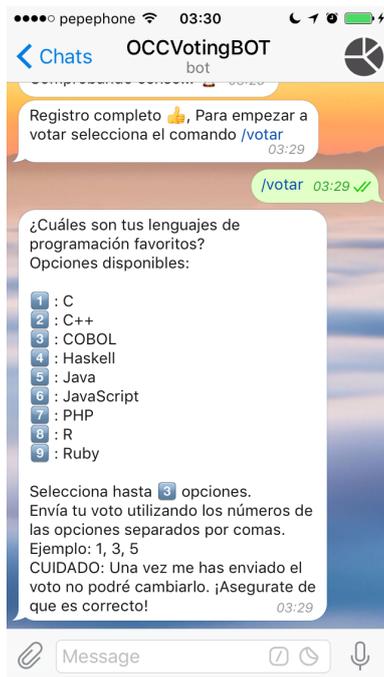


Figura 4.22: Imagen del inicio del proceso de votación.

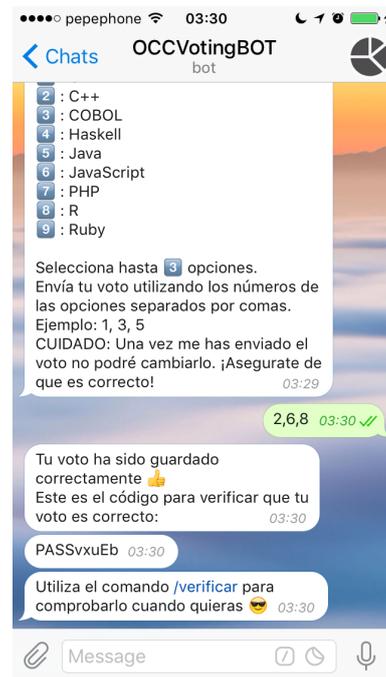


Figura 4.23: Imagen de la emisión de voto.

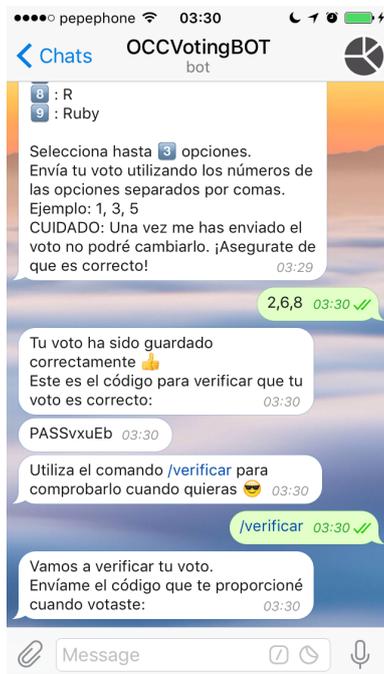


Figura 4.24: Imagen del inicio del proceso de verificación de voto.

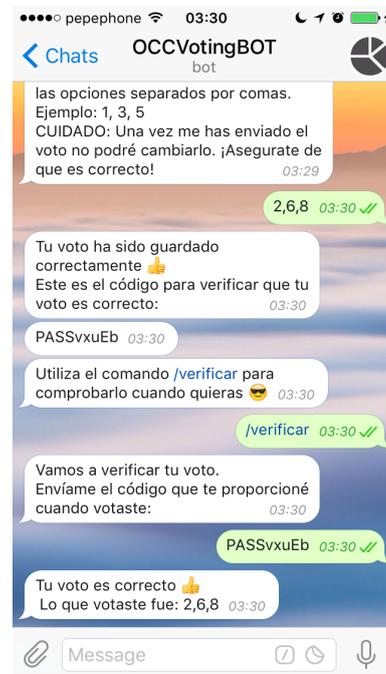


Figura 4.25: Imagen de la verificación de voto.

### 4.4.2. Análisis de uso

Una vez puesto en marcha el prototipo, se recogieron datos sobre el uso que se estaba haciendo de este en el entorno real de la votación en el Ayuntamiento de Quart de Poblet, durante los 5 días que duró el proceso participativo.

En concreto, se recogió información sobre los mensajes recibidos, distinguiendo entre mensajes correctos: Aquellos que son esperados o que forman parte de la casuística del sistema; y mensajes incorrectos: Aquellos que no son esperados y, por tanto, producen un mensaje que informa al usuario de dicho error.

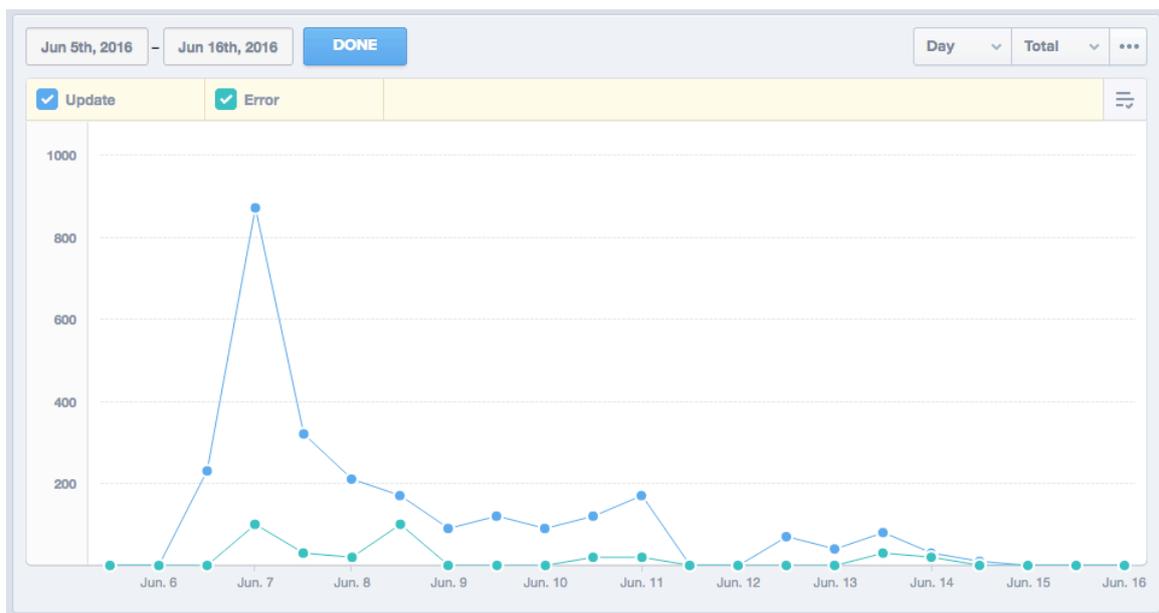


Figura 4.26: Mensajes recibidos por el bot, correctos vs. erróneos.

La gráfica de la Figura 4.26 muestra la evolución del número de mensajes de cada tipo. Se puede observar que, en general, el número de mensajes correctos supera al de mensajes incorrectos. Esto indica que los usuarios no parece que hayan cometido un gran número de errores durante el uso del bot.

De este análisis podemos deducir que, para tratarse de una tecnología bastante novedosa y a la que, en general, los usuarios no están acostumbrados, los resultados han sido muy satisfactorios en cuanto al uso que se ha hecho del sistema.

### 4.4.3. Análisis de rendimiento

Para el análisis del rendimiento del sistema, se tomaron medidas durante los treinta minutos en los que el *bot* tuvo una mayor carga de trabajo. En la Figura 4.27 se recoge una gráfica que representa este análisis. Concretamente se midieron cuatro tipos de dimensiones:

#### *Request Queuing*

Representada por el color verde en la gráfica, mide el tiempo que las peticiones recibidas por el servidor esperan a ser atendidas o resueltas.

#### *Node*

Representada por el color azul claro en la gráfica, mide el tiempo que tarda el servidor a nivel de ejecución de instrucciones en satisfacer una petición. Este sería el tiempo de ejecución del sistema.

#### *MongoDB*

Representada por el color amarillo en la gráfica, mide el tiempo de acceso a la base de datos.

#### *Response time*

Representada en color azul oscuro, mide el tiempo medio de respuesta, es decir, el tiempo que tarda el sistema en responder a una petición desde que esta es recibida.

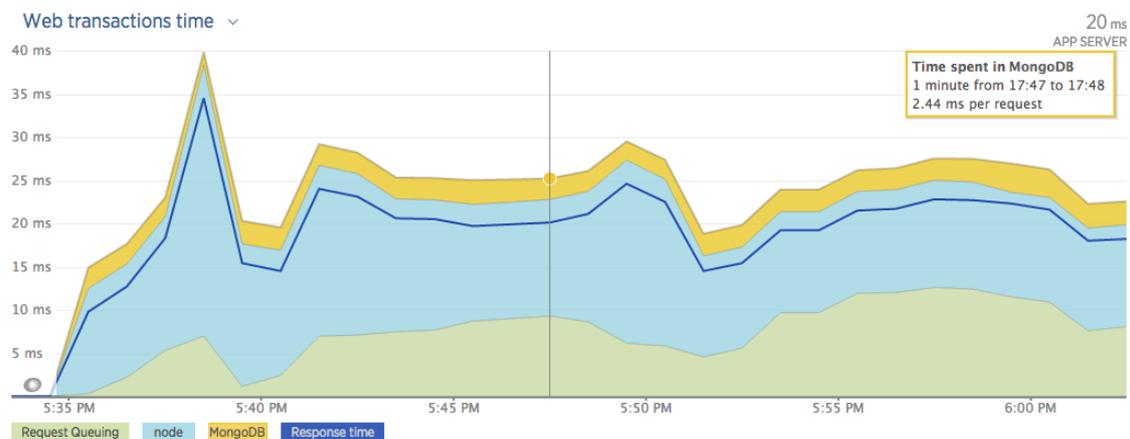


Figura 4.27: Relación de tiempo consumido por la aplicación.

Como se puede ver en la gráfica anterior, en general, para una carga de trabajo considerable tal y como muestra la Figura 4.28 que representa la evolución del número de peticiones recibidas por el servidor, el sistema nunca tarda más de 40ms en responder. De esta manera podemos concluir que satisface los requisitos establecidos en relación al tiempo de respuesta.

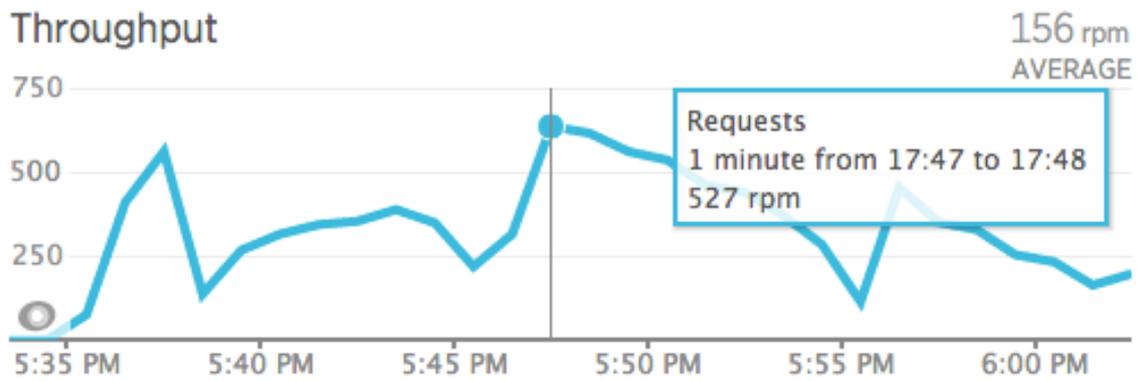


Figura 4.28: Peticiones por minuto.



---

---

## CAPÍTULO 5

# Conclusiones y trabajo futuro

---

### 5.1 Conclusiones

---

El presente TFG concluye habiendo alcanzado el objetivo establecido de desarrollar un sistema de votación basado en un *bot* e integrado en la plataforma de mensajería instantánea *Telegram*. El éxito del proyecto ha residido en gran parte a la metodología seleccionada. La idea de abordar el desarrollo de un prototipo que satisficiera los requisitos establecidos al inicio: escalabilidad, sencillez y fiabilidad, ha contribuido al diseño final de una solución adaptada a las necesidades reales. Asimismo, se ha conseguido implementar un sistema robusto, capaz de hacer frente a importantes exigencias de rendimiento una vez implantado y utilizado en un entorno real y, a su vez, crítico como fue el caso de la consulta realizada por el Ayuntamiento de Quart de Poblet.

Por otro lado, el uso de este novedoso sistema ha sentado un precedente en cuanto a consultas ciudadanas se refiere. De hecho, gracias a la repercusión mediática generada tras la iniciativa de Quart de Poblet, otros municipios han empezado a interesarse por este sistema de votación, hecho que ayudará a que este proyecto no quede en el olvido, sino que reforzará su continuidad, buscando la mejora del mismo. En este sentido, es plausible pensar que el proyecto ha contribuido a avanzar en el cambio de paradigma de la participación ciudadana. La solución aportada ha sentado un precedente sobre el que, desde ahora mismo, se puede continuar trabajando para reforzar sus puntos débiles, y mejorarlo de manera que, en un deseado y esperadamente cercano futuro, prospere hasta tal punto que se utilice por todos como opción para votar.

Ahora bien, de cara a cerrar el ciclo de aprendizaje iniciado con la realización del TFG, y con el ánimo de poner de manifiesto las lecciones aprendidas, creo necesario acabar la presente memoria con una reflexión crítica en torno al trabajo realizado.

Así, haber trabajado en un entorno real ha permitido poner de manifiesto las implicaciones y limitaciones del sistema de votación implementado. En primer lugar, las limitaciones relacionadas con la seguridad. En este sentido, el sistema desarrollado no cumple con las garantías requeridas para poder implantarse, por ejemplo, en consultas o elecciones vinculantes. Actualmente, no se dispone de los elementos necesarios para la autenticación de los votantes que sea completamente fiable. Además, el uso de *Telegram* como plataforma base del desarrollo

implica cierta dependencia, pues no se tiene el control sobre ella y, en caso de que esta dejara de funcionar por alguna razón, el proyecto se tornaría completamente inservible. Esto, junto con la brecha digital que el uso de una herramienta de nueva implantación implica, suponen una limitación más del trabajo realizado. Por otro lado, existen limitaciones inherentes al uso de *bots*, pues estos son sencillamente sistemas de envío y recepción de mensajes, sin capacidad para otras funcionalidades más avanzadas. Finalmente, cabe añadir que el ajuste de los plazos académicos a los plazos requeridos por el Ayuntamiento de Quart de Poblet han dificultado también el cumplimiento de la fecha objetivo de entrega.

Este trabajo, sin lugar a dudas, ha supuesto una experiencia personal que no puedo menos que calificar como extremadamente enriquecedora. La realización del TFG me ha permitido aplicar los conocimientos y habilidades adquiridas a lo largo de la titulación. A su vez, me ha situado ante el reto de tener que investigar para adquirir nuevos conocimientos y, de esta forma, poder obtener el resultado novedoso que se perseguía desde el inicio y, a la vista de la repercusión en medios, podemos decir que conseguido.

Igualmente, la utilización de tecnologías vanguardistas para la implementación del proyecto han supuesto un arduo, pero a la vez satisfactorio, reto, el cual me ha demostrado que la firme base de sabiduría adquirida a lo largo de la carrera ha resultado muy útil a la hora de poder aprender y utilizar tecnologías que nunca antes había visto. Base sin la cual nunca hubiera podido desarrollar la destreza para aplicarlas a la velocidad y el conocimiento necesarios.

También me ha resultado muy gratificante haber podido observar como un proyecto complicado, desarrollado íntegramente por mí mismo, se ha llevado a la práctica en un entorno real, donde ha sido utilizado por un gran número de personas, con el resultado esperado. Sin duda alguna, me ha hecho sentir orgulloso del trabajo realizado.

Finalmente, haber trabajado en un equipo multidisciplinar, compuesto, entre otros, por politólogos, políticos, informáticos, historiadores, administrativos, etc., ha supuesto una experiencia que, además, me ha demostrado que la informática no tiene límites.

Lo sentenció Edsger Dijkstra: *“Computer science is no more about computers than astronomy is about telescopes.”*

## 5.2 Trabajo futuro

---

La perspectiva de la solución desarrollada, ya implantada en el Ayuntamiento de Quart de Poblet, es bastante alentadora, pudiendo extenderse a otros municipios interesados, así como otras entidades que quisieran hacer uso de esta tecnología. Sin embargo, de cara a mejorar y ampliar el trabajo realizado, se proponen diversas líneas de trabajo de cara al futuro:

### **Interfaz de administración**

Una línea de trabajo en la que ya se está trabajando en la actualidad es la creación de una interfaz de administración que permita crear votaciones y

configurar el entorno del *bot* de forma dinámica y amigable para el usuario. Facilitando, así, todas las tareas relacionadas con la puesta en marcha de una nueva votación cada vez que se requiera.

### **Autenticación**

Pese a que la tecnología actual no hace posible la autenticación fiable de usuarios, garantizando que la persona que se encuentra detrás de un entorno digital es quien dice ser, se proponen diversas mejoras en esta línea de trabajo.

Una de ellas es la implementación de alternativas que añadan nuevas capas de verificación de usuarios, esto es, incluir nuevos requisitos a la hora de poder registrarse para proceder a la emisión de un voto.

### **Lenguaje natural**

Otra línea de trabajo muy interesante, y de la que se ha empezado a analizar su viabilidad, es la posibilidad de soportar el procesamiento de lenguaje natural para interactuar con el *bot*. Esto significaría que el usuario podría comunicarse de una forma más sencilla con el *bot*, pudiendo realizar preguntas o enviar mensajes directamente y sin seguir ningún tipo de formato restrictivo.

### **Tipos de votación**

Por otra parte, resultaría muy interesante que el sistema pudiera soportar diferentes tipologías de votación. De esta manera, se podría implantar para procesos que van más allá de una votación del tipo selección múltiple.

### **Diseño centrado en la ciudadanía**

Desde el punto de vista metodológico se ha empezado a trabajar en el diseño de una metodología de diseño centrada en la ciudadanía. A partir del Diseño Centrado en el Usuario se pretende la participación activa de la ciudadanía en las soluciones tecnológicas desarrolladas para facilitarle el ejercicio del poder en el marco de procesos de empoderamiento.



# Bibliografía

---

- [1] Open Government Standards. Los estándares de gobierno abierto. <http://www.opengovstandards.org/>, 2016. [Online; accedido el 06/06/2016].
- [2] Qué es la rendición de cuentas. *Cuadernos de transparencia. Instituto Federal de Acceso a la Información y Protección de Datos*, 3, 2004.
- [3] M. Villoria. Reflexiones sobre el gobierno abierto, qué es el gobierno abierto, la filosofía de un concepto en construcción. In *Fundación Manuel Giménez Abad de Estudios Parlamentarios y del Estado Autonómico.*, 2015.
- [4] Richard Stallman. About the gnu project. <https://www.gnu.org/gnu/thegnuproject.html>, 2015. [Online; accedido el 06/06/2016].
- [5] B. Bimber. The internet and citizen communication with government: Does the medium matter? *Political Communication*, 16(4):409–428, 1999.
- [6] B. S. Krueger. Assessing the potential of internet political participation in the united states. *American Politics Research*, 0(5):476–498, 2002.
- [7] Tolbert C. J. Mossberger, K. and R. S. McNeal. *Digital Citizenship. The Internet, Society, and Participation*. MIT Press, 2007.
- [8] BBC. The uk's eu referendum: All you need to know. <http://www.bbc.com/news/uk-politics-32810887>, junio 2016. [Online; accedido el 26/06/2016].
- [9] Instituto Nacional de Estadística. Encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares. *INEbase*, octubre 2015.
- [10] Comisión Nacional de los Mercados y la Competencia. Caracterización del uso de algunos servicios over the top en españa. *CNMC*, diciembre 2014.
- [11] The Electoral Knowledge Network. E-voting. <http://aceproject.org/ace-en/focus/e-voting/introducing-electronic-voting-considerations/>, 2015. [Online; accedido el 12/06/2016].
- [12] The Electoral Knowledge Network. Types of e-voting. <http://aceproject.org/ace-en/focus/e-voting/types-of-e-voting>, 2015. [Online; accedido el 12/06/2016].

- 
- [13] Aviel D. Rubin Tadayoshi Kohno, Adam Stubblefield and Dan S. Wallach. Analysis of an electronic voting system. Technical Report TR-2003-19, Johns Hopkins University Information Security Institute Technical Report, julio 2003.
- [14] Russell Michaels and Simon Ardizzone. Hacking democracy.
- [15] Tom Arnold. Internet identity theft. Technical report, Technology Working Group, eBusiness Division, SIIA, junio 2000.
- [16] Victor Luckerson. Chatbots are back and theyre about to take over. <http://time.com/4194063/chatbots-facebook-messenger-kik-wechat/>, enero 2016. [Online; accedido el 18/06/2016].
- [17] ZenDesk. The omnichannel customer service gap. Technical report, LoudHouse, noviembre 2013.
- [18] OneReach. The high demand for customer service via text message. <https://onereach.com/resources/high-demand-for-text-message-2014-report>, 2015. [Online; accedido el 19/06/2016].
- [19] The Open Web Application Security Project. Owasp top ten. [https://www.owasp.org/index.php?title=Top\\_10\\_2013-Top\\_10](https://www.owasp.org/index.php?title=Top_10_2013-Top_10), 2013. [Online; accedido el 20/06/2016].
- [20] Scottish Qualifications Authority. Functional and non-functional requirements. [http://www.sqa.org.uk/e-learning/SDM03CD/page\\_02.htm](http://www.sqa.org.uk/e-learning/SDM03CD/page_02.htm), 2007. [Online; accedido el 20/06/2016].

---

# APÉNDICE A

## Código implementado

---

Todo el código implementado puede consultarse en el perfil de *GitHub* del proyecto, accesible desde <https://github.com/hubcivico/OCCVotingBOT/tree/develop>



---

---

## APÉNDICE B

# Repercusión en medios

---

Quart

### El consistorio destaca la «repercusión» del cambio de las calles franquistas

08.06.2016 | 08:02

**LEVANTE-EMV | QUART DE POBLET** El Ayuntamiento de Quart de Poblet ha destacado a través de un comunicado la «repercusión» que está teniendo el proceso participativo abierto para la elección del nuevo nombre de ocho calles franquistas. «La iniciativa está teniendo repercusión internacional y ha despertado el interés de diferentes instituciones, comunidades, plataformas y personalidades ligadas al Gobierno Abierto, las TIC e internet», señalaba ayer el consistorio en un comunicado. «Entre ellos, el que fue responsable de Facebook para España y América Latina, Elies Campo, o novagob, la Red Social de la Administración Pública o uno de los fundadores de Telegram Messenger, Pavel Durov». El ayuntamiento informó de que los datos de participación en el primer día son de 272 votos, 79 por Telegram y 193 vía web. El proceso está abierto hasta el sábado 11 de junio a las 14 horas. «El uso de Telegram en una consulta a la ciudadanía es pionero y la Concejalía de Gobierno Abierto, que dirige Juan Medina, ha podido implementarlo gracias a una herramienta desarrollada por HUBcívico, una iniciativa sin ánimo de lucro puesta en marcha por el Grup Càlam con el objetivo de facilitar el uso de la tecnología en procesos de gobierno abierto».

**Figura B.1:** Nota de prensa sobre el proceso participativo publicada en el diario Levante.



The image shows a screenshot of a tweet from Pavel Durov (@durov). The tweet text reads: "A Spanish town will hold a referendum via Telegram to decide the name of 8 streets". Below the text is a link to an article with a thumbnail image of a woman at a computer. The article title is "Arranca el proceso para cambiar ocho calles franquistas de..." and the text says "Desde esta mañana los ciudadanos podrán escoger a ocho mujeres de una lista de 24 - Las votaciones concluirán el sábado levante-emv.com". The tweet has 138 retweets and 233 likes. It was posted at 12:42 AM on June 7, 2016, from Manhattan, NY.

**Pavel Durov** @durov Following

A Spanish town will hold a referendum via Telegram to decide the name of 8 streets

**Arranca el proceso para cambiar ocho calles franquistas de...**  
Desde esta mañana los ciudadanos podrán escoger a ocho mujeres de una lista de 24 - Las votaciones concluirán el sábado  
[levante-emv.com](http://levante-emv.com)

RETWEETS **138** LIKES **233**

12:42 AM - 7 Jun 2016

Manhattan, NY

138 233

**Figura B.2:** *Tweet* publicado por Pavel Durov, creador de *Telegram*.

## C. VALENCIANA

Alicante | Castellón | Valencia | Valencià | Innova | Cultura

### Quart de Poblet votará el nombre de ocho calles afectadas por la Ley de Memoria Histórica



Publicado 31/05/2016 13:54:06 CET

VALENCIA, 31 May. (EUROPA PRESS) -

El Ayuntamiento de Quart de Poblet ha convocado, por primera vez, una votación para que todos los vecinos censados y mayores de 16 años seleccionen el nombre de ocho de las 10 calles afectadas por Ley 52/2007, conocida como Ley de la Memoria Histórica, según han informado fuentes municipales en un comunicado.

Así, entre el lunes 6 de junio y el sábado 11 de junio, a las 14 horas, podrán votar los vecinos para eliminar el nombre de estas calles relacionadas con personas, entidades, hechos o señas distintivas del régimen franquista.

Una de las principales novedades es que se podrá participar a través del móvil, con la aplicación gratuita Telegram y gracias a una herramienta pionera desarrollada por HUBcívico, una iniciativa sin ánimo de lucro puesta en marcha por el Grupo Càlam con el objetivo de facilitar el uso de la tecnología en procesos de gobierno abierto, que colabora en este proceso.

**Figura B.3:** Nota de prensa sobre el proceso participativo publicada por Europa Press.



The image shows a screenshot of a Telegram post. At the top left is the Telegram logo (TG) and the name 'Telegram Geeks'. To the right, it shows '1K' views and the time '13:29'. Below the name is a header 'TELEGRAM | REFERENDUM' with a bar chart icon. The main text reads: 'A spanish town (Quart de Poblet, population: 25k) will hold a referendum via Telegram to decide the name of 8 streets with the bot @QuartVotaBOT.' Below this is a paragraph starting with a trophy icon: 'So maybe this is the first governmental referendum via Telegram of the history.' Underneath is a link 'More info (in spanish)'. There are two expandable sections: 'Arranca el proceso para cambiar ocho calles franquistas de Quart de Poblet' and 'Official Announcement'. The 'Official Announcement' section is expanded, showing the text: 'Arranca el proceso para cambiar ocho calles franquistas de Quart de Poblet' and 'Desde esta mañana los ciudadanos podrán escoger a ocho mujeres de una lista de 24 - Las votaciones concluirán el sábado'. To the right of this text is a small image of a woman at a computer.

**Telegram Geeks** 1K 13:29

**TELEGRAM | REFERENDUM**

A spanish town (*Quart de Poblet*, population: 25k) will hold a referendum via Telegram to decide the name of 8 streets with the bot [@QuartVotaBOT](#).

🏆 So maybe this is the first governmental referendum via Telegram of the history.

[More info \(in spanish\)](#)

➔ [Arranca el proceso para cambiar ocho calles franquistas de Quart de Poblet](#)

➔ [Official Announcement](#)

**Levante-Emv**  
**Arranca el proceso para cambiar ocho calles franquistas de Quart de Poblet**  
Desde esta mañana los ciudadanos podrán escoger a ocho mujeres de una lista de 24 - Las votaciones concluirán el sábado



Figura B.4: Publicación del grupo de desarrolladores de Telegram.



## DOS MUNICIPIOS DE ESPAÑA ATRAVESADOS POR LA "DEMOCRACIA DIGITAL"

En uno eligen nuevos nombres para unas calles desde la app de Telegram, mientras que en otro se economizan recursos gracias a Twitter.



**El municipio español de Jun conecta a su po**

TAGS [Telegram](#), [Twitter](#), [redes sociales](#)

La pujanza de las redes sociales, el uso masivo de los celulares inteligentes y la inmediatez de las comunicaciones modifican a diario la vida cotidiana de las personas. Pero no solo eso, ahora también los procesos democráticos y la simplificación de recursos, al menos en dos municipios de España.

El Ayuntamiento de Quart de Poblet, situado dentro de la Comunidad Valenciana, lleva adelante desde el lunes un proceso participativo para renombrar ocho calles afectadas

**Figura B.5:** Nota de prensa sobre el proceso participativo el el diario Clarin.

## Technologie

### UN RÉFÉRENDUM VIA L'APPLICATION TELEGRAM

Publié par : Houssen Moshinaly Date: 6 juin 2016 Heure :15:35 dans Technologie  
Laisser un commentaire

Affichage ultra-rapide !

Mot-clés: **Telegram**

**La petite commune de *Quart de Poblet* en Espagne a décidé d'utiliser des outils informatiques, notamment l'application *Telegram* pour collecter les votes d'un référendum pour nommer certaines de ses rues. C'est la première fois qu'on utilise Telegram pour organiser un référendum même si l'ampleur est modeste.**



La commune de *Quart de Poblet* est situé dans la province de Valence en Espagne. Elle veut renommer certaines de ses rues pour les remplacer par des noms de femme. La commune possède très peu de présence féminine dans les noms de ses rues.

La commune a mis plusieurs systèmes de vote pour les habitants. Le site officiel, des kiosques informatiques et l'application **Telegram** via un bot appelé **QuartVotaBot**. C'est un événement insignifiant dans une petite ville, mais cela prouve que les outils technologiques comme **Telegram** peuvent énormément faciliter des processus politiques et augmenter la participation citoyenne.

**Figura B.6:** Nota de prensa sobre el proceso participativo publicada por el medio francés *Housseniawriting*.



SOCIÉTÉ 09:04 08.06.2016 [URL courte](#)

0 447 9 3

**On n'arrête plus le progrès! Afin de faciliter le processus politique, les autorités d'une petite commune espagnole ont décidé d'utiliser l'application Telegram pour recueillir les votes d'un référendum pour nommer certaines des rues.**

C'est la première fois qu'une application est utilisée pour organiser un référendum, même si son ampleur reste modeste.

Quart de Poblet est une commune espagnole située dans la province de Valence et dont la population s'élève à près de 25.000 habitants. Une des particularités de cette ville réside dans le fait que le nom de ses rues est à dominante masculine. Les autorités ont donc décidé de se lancer sur la voie de la parité en proposant de rebaptiser huit d'entre elles. Les habitants de Quart de Poblet se sont donc vus proposer 24 noms.

**Figura B.7:** Nota de prensa sobre el proceso participativo publicada por el medio francés *sputniknews*.