

UNIVERSIDAD SAN PABLO - CEU  
ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA EN INFORMÁTICA



PROYECTO FINAL DE CARRERA

**SISTEMA PARA ELECCIONES A LA JUNTA DE  
ESCUELA CON VOTO POR INTERNET E  
IDENTIFICACIÓN CON DNIe 3.0**

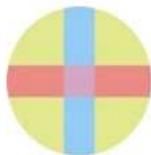
Autor: José Carlos Jiménez Gómez

Director: Raúl García García

---

junio de 2017





UNIVERSIDAD SAN PABLO-CEU

ESCUELA POLITÉCNICA SUPERIOR

División de Ingeniería Informática y de Telecomunicación

## Calificación del Proyecto Fin de Carrera

### Datos del alumno

NOMBRE:

### Datos del Proyecto

TÍTULO DEL PROYECTO:

### Tribunal calificador

PRESIDENTE:

FDO.: \_\_\_\_\_

SECRETARIO:

FDO.: \_\_\_\_\_

VOCAL:

FDO.: \_\_\_\_\_

Reunido este tribunal el \_\_\_\_/\_\_\_\_/\_\_\_\_\_, acuerda otorgar al Proyecto de Fin de Carrera presentado por Don \_\_\_\_\_ la calificación de \_\_\_\_\_.



# Resumen

El presente proyecto desarrolla una solución tecnológica de voto por Internet con el que realizar las Elecciones a la Junta de Escuela de la Escuela Politécnica Superior<sup>1</sup> de la Universidad San Pablo CEU<sup>2</sup>, sita en Madrid, España.

Trata de ofrecer una prueba de concepto de un sistema seguro de voto por Internet que hace uso del nuevo DNIe 3.0 como herramienta de identificación remota del votante.

Hasta el momento de la redacción de esta memoria, existen soluciones implementadas para este tipo de problemas, pero ningún sistema actual permite utilizar el nuevo documento de identidad español para identificar de forma remota al votante.

La finalidad de este proyecto, además de la implementación de una solución, es que esta sirva como prueba y punto de comienzo de futuros desarrollos con el DNIe con chip NFC.

Por tanto, podemos considerar que con este desarrollo se establece el primer sistema de voto por Internet que utiliza del DNIe 3.0 para identificar al votante con tecnología NFC.

Para llevar a cabo el desarrollo se ha decidido adaptar una solución ya existente, Helios Voting. Este proyecto, creado por Ben Adida y nacido en el MIT, es considerado un estándar de facto en votación electrónica basada en protocolos de Verificación Punto a Punto y en un esquema criptográfico homomórfico. Es el proyecto libre más completo para aquellos procesos electorales con riesgo bajo de coacción.

No obstante, para poder cumplir con los objetivos del PFC, que contiene el uso del DNIe 3.0 como documento digital de identificación de usuario, ha sido necesario realizar una integración de sistemas. El proyecto Helios Voting, pese a proporcionar un gran número de opciones de login, no soporta por defecto identificación con certificados digitales, lo cual es básico para poder utilizar los que contiene el DNIe. Esto provoca que se haya diseñado un módulo de identificación alternativo, basado en protocolo OAuth 2.0 con un servidor web configurado para aceptar estos certificados.

---

<sup>1</sup><http://www.uspceu.com/es/alumnos/escuela-politecnica-superior/conoce/index.aspx>

<sup>2</sup><http://www.uspceu.com/>

---

Para facilitar el uso de este documento, se ha integrado también una app de Android desarrollada por la Policía. Esta app requiere una adaptación para las necesidades del proyecto, pero permite a los votantes usar sus dispositivos móviles para votar utilizando el sensor NFC de los mismos y su propio DNIe 3.0, sin necesidad de requerir de hardware externo como los lectores de chips con contacto. Esta aproximación posibilita que realmente se pueda votar desde cualquier lugar con conexión a Internet.

# Abstract

This document shows the development of an Internet voting solution to be used to hold the Elections for the Junta de Escuela in the Escuela Politécnica Superior of the Universidad San Pablo CEU, placed in Madrid, Spain.

It tries to offer a proof of concept of a secure Internet voting system which uses the new DNIe 3.0 as a tool for the remote voter identification.

At the moment of writing this document, there are several solutions implemented to solve this kind of problems, but there is none system which allow the use of the new Spanish ID card to identify the voter remotely.

The goal of this project, along with the implementation of the solution, is that it will serve as a start point for future developments with voting and the new DNIe with NFC chip.

So then, we can consider that this project establishes the first Internet voting system that uses the DNIe 3.0 to identify the voter using its NFC chip.

To accomplish the solution, it has been decided to adapt an existing solution, Helios Voting. This project, created by Ben Adida in the MIT, is considered as a de-facto standard in electronic voting based in End-to-end Verifiability protocols and a homomorphic cryptography. It is the more complete free software project for low coercion risk elections.

Nevertheless, to fulfill the goals of this PFC, which includes the use of the DNIe as digital user identification, it has been necessary a system integration. Helios Voting, though it offers several login options, it does not support digital certificates login by default, which is basic for using the ones included in the DNIe card. This causes the need to design an alternative identification module, based in OAuth 2.0 protocol with a web server configured to accept these digital certificates.

To ease the use of this document, an Android app developed by Spanish Police Department has been integrated. This app has needed to be adapted in order to fulfill the requirements of the project, but it allows the voters to use their mobile devices to cast a

---

vote using their NFC sensor and their own DNIe 3.0 cards, with no needs of external hardware as contact chip readers. This approach makes possible to cast votes from anywhere with an Internet connection.

# Agradecimientos

Hay demasiadas personas a las que me gustaría agradecer porque son artífices de que este PFC se haya redactado y haya completado así mis estudios de Ingeniería Informática Superior. Quisiera dar las gracias a todas y cada una de ellas, pero espero que, por falta de espacio, me permitan la licencia de hacerlo aquí a algunas de ellas.

En primer lugar quiero dar las gracias a mis padres y a mi hermana Marta, mi familia. Sin ellos, sin su ayuda, sin su apoyo, este proyecto no sería posible. Para ellos es cada una de las páginas e ideas aquí plasmadas.

A Begoña. Ella es la que ha tenido que soportar todo lo que este proyecto ha conllevado. Gracias por aguantarme, sobre todo mi carácter, mi genio, mis divagaciones filosóficas acerca del voto por Internet. Y, sobre todo, gracias por el tiempo que me has dado y todo aquél que no hemos aprovechado juntos por dedicarlo a este PFC. Espero poder devolvértelo a partir de ahora.

A la gente del *Sanagus*, en especial a mis compañeros de piso Jordi, Dj y Molinete, que me han acompañado durante todos mis años universitarios. Gracias a Ana, que también tuvo que soportar bastante mi carácter al inicio de este proyecto y su ayuda ha sido importantísima. Y a Felipe, que me ha dado su confianza y la oportunidad de ayudarle en una nueva aventura.

A mis compañeros y amigos de Indra, nombrando a David, Mariano, Jesús, Fernando, Sara, Pedro, Elena, Luis, Rodrigo, Lorena, Julián, Zule, Cari, Eduardo, Belén... y podría seguir durante muchas líneas. De prácticamente todos he aprendido muchísimo más de lo que he enseñado. Ha sido un placer compartir tiempo y conocimiento con el grupo de ingenieros más experto en elecciones que hay en este país.

A mis compañeros en la carrera, incluidos los miembros del *Gang of Five*. Carfesán, Juaca, Weil, Marta, Chacón, Kapi, Vendrell, Juanolo, Charly... también me faltan hojas para nombrarlos a todos. Fue un placer compartir estos años con vosotros.

A mis profesores, que son los que emplearon su tiempo y energía en que pudiese obtener

los conocimientos necesarios para formarme como Ingeniero.

A mis primos, tíos y abuelos. Mi familia es grande, y siempre habéis estado ahí. Y los que faltáis, espero que estéis orgullosos con mi forma de ser y en lo que me he convertido.

Al Betis. Por el sufrimiento de cada semana. Por su bandera, que me ha dado un hobby viajero. Y por aquel junio de 2005. ;) Y a Iniesta, por aquel julio de 2010.

A Ramón, por demasiadas cosas, ya lo sabes. Hace un tiempo aquí habría puesto que gracias por ser *mi hermano mayor* y compañero de prácticas, estudio, fiestas y vida. Pero, junto con María, me habéis *presentado* a mis *sobrinos* Ángel e Irene y todo ha cambiado. Ahora tengo que agradecerlos todo a los cuatro, no a ti solo. Menos mal que siempre has estado ahí, que estáis ahí.

A Raúl, mi tutor, por TODO. Literalmente, te debo que este PFC se haya llegado a presentar. Después de tanto tiempo de charlas y emails creo que no he llegado a agradecerte suficientemente lo que has hecho por mí. Espero que no se quede corto resumirlo en dos simples palabras: Muchas gracias.

A todos los nombrados aquí y a los que no, pero aún así deberían estar, muchas gracias. Espero poder aportar a los demás todo lo que he recibido de vosotros.

Gracias.

# Índice general

<b>Calificación</b>	<b>III</b>
<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VII</b>
<b>Agradecimientos</b>	<b>IX</b>
<b>Índice General</b>	<b>xv</b>
<b>Índice de Figuras</b>	<b>xix</b>
<b>Índice de Tablas</b>	<b>xxi</b>
<b>1. Introducción</b>	<b>23</b>
1.1. Motivación del Proyecto . . . . .	25
1.2. Antecedentes . . . . .	26
1.3. Organización de la memoria del PFC . . . . .	28
1.4. Metodología . . . . .	29
1.4.1. Documentación . . . . .	29
1.4.2. Metodología de desarrollo . . . . .	29
<b>2. Estado de la cuestión</b>	<b>31</b>
2.1. Voto electrónico . . . . .	31
2.1.1. Niveles . . . . .	32
2.1.2. Verificabilidad vs. Secreto . . . . .	34
2.1.3. Requisitos del voto electrónico . . . . .	36
2.1.3.1. Fujioka, Okamoto y Ohta . . . . .	36
2.1.3.2. Universidad de Extremadura . . . . .	37
2.1.3.3. Bokslag y Vries . . . . .	39
2.1.3.4. Resumen . . . . .	40
2.2. Experiencias de Voto por Internet . . . . .	41
2.2.1. Estonia . . . . .	42
2.2.2. Noruega . . . . .	47

2.2.3.	Suiza . . . . .	48
2.2.4.	Universidades . . . . .	49
2.2.5.	Escuela Politécnica Superior - Universidad San Pablo CEU . . . . .	50
2.3.	Proyectos varios de voto por Internet . . . . .	51
2.3.1.	Votescript . . . . .	51
2.3.2.	SEVI . . . . .	53
2.4.	Proyectos de Sistemas de Votación Auditables Punto a Punto . . . . .	54
2.4.1.	ADDER . . . . .	56
2.4.2.	Ágora Ciudadana - Ágora Voting . . . . .	59
2.4.3.	Helios . . . . .	60
2.4.4.	Comparación Ágora - Helios . . . . .	62
2.5.	Mecanismos criptográficos . . . . .	63
2.5.1.	Primitivas criptográficas . . . . .	64
2.5.1.1.	Firma ciega . . . . .	65
2.5.1.2.	Secreto compartido . . . . .	65
2.5.1.3.	Pruebas de conocimiento nulo . . . . .	65
2.5.1.4.	Mixnets . . . . .	68
2.5.1.5.	Cifrado homomórfico . . . . .	69
2.5.1.5.1.	ElGamal . . . . .	70
2.5.1.5.2.	ElGamal Exponencial . . . . .	73
2.5.1.2.	Esquemas de Voto Electrónico . . . . .	75
2.6.	Estado actual de las tecnologías . . . . .	77
2.6.1.	Certificados Digitales . . . . .	77
2.6.2.	DNIe . . . . .	79
2.6.3.	NFC . . . . .	81
<b>3. Planteamiento</b>		<b>85</b>
3.1.	Objetivos finales del proyecto . . . . .	85
3.2.	Descripción del sistema actual . . . . .	87
3.2.1.	Elecciones a la Junta de Escuela de la EPS . . . . .	87
3.2.1.1.	Definición de la Junta de Escuela . . . . .	87
3.2.1.2.	Plazos del Proceso Electoral . . . . .	88
3.3.	Fases del proceso electoral . . . . .	88
3.3.1.	Fase preelectoral . . . . .	91
3.3.1.1.	Definición de los límites o reglas de la elección . . . . .	91
3.3.1.2.	Elaboración del censo . . . . .	91
3.3.1.3.	Registro de votantes . . . . .	92
3.3.1.4.	Presentación de las candidaturas . . . . .	93
3.3.1.5.	Generación de claves de encriptado . . . . .	94
3.3.2.	Fase electoral . . . . .	94
3.3.2.1.	Identificación del votante . . . . .	94

3.3.2.2. Votación . . . . .	100
3.3.2.3. Totalización de resultados . . . . .	100
3.3.3. Fase postelectoral . . . . .	101
3.3.3.1. Difusión de resultados . . . . .	101
3.3.3.2. Verificación de resultados . . . . .	102
3.4. Descripción . . . . .	102
<b>4. Riesgos</b>	<b>105</b>
4.1. Identificación y gestión de riesgos . . . . .	105
<b>5. Análisis del sistema</b>	<b>109</b>
5.1. Especificación de requisitos . . . . .	109
5.1.1. Introducción . . . . .	109
5.1.2. Ámbito del sistema . . . . .	110
5.1.3. Requisitos funcionales . . . . .	111
5.1.4. Requisitos propios del voto electrónico . . . . .	111
5.1.5. Requisitos del proceso electoral . . . . .	112
5.1.6. Requisitos no funcionales . . . . .	113
5.1.7. Requisitos específicos . . . . .	114
5.2. Actores . . . . .	114
5.3. Integración . . . . .	117
5.3.1. Sistema central de votación . . . . .	118
5.3.2. Sistema de autenticación / identificación . . . . .	122
5.3.3. Cliente web + app Android . . . . .	122
5.4. Esquema de Voto Electrónico . . . . .	123
5.5. Helios Voting . . . . .	125
5.5.1. Fases de la elección en Helios . . . . .	125
5.5.2. Auditorías . . . . .	129
5.5.3. Protocolo criptográfico . . . . .	130
5.5.4. Identificación . . . . .	131
5.5.4.1. Evolución . . . . .	131
5.5.5. Adaptación a dispositivos . . . . .	137
5.5.6. Votación . . . . .	138
5.5.7. Difusión de resultados . . . . .	139
5.5.8. Verificación del voto . . . . .	141
5.5.9. Verificación de la totalización e integridad de la elección . . . . .	141
<b>6. Solución</b>	<b>143</b>
6.1. Arquitectura del sistema . . . . .	143
6.2. Sistema de autenticación . . . . .	144
6.2.1. Servidor web: . . . . .	145
6.2.2. Aplicación de autenticación . . . . .	149

6.2.3. OCSP . . . . .	150
6.2.4. Esquema de la base de datos . . . . .	151
6.3. Sistema de votación . . . . .	151
6.3.1. Esquema de la base de datos . . . . .	153
6.3.2. Estructura . . . . .	155
6.4. Cliente Android . . . . .	159
6.4.1. Código fuente . . . . .	160
6.4.2. App Android de Autenticación . . . . .	161
6.5. Topología de red . . . . .	168
6.6. Esquema de votación . . . . .	171
6.6.1. Registro . . . . .	171
6.6.2. Identificación / Autenticación . . . . .	172
6.6.3. Definición de la papeleta . . . . .	174
6.6.4. Votación . . . . .	174
6.6.5. Totalización . . . . .	181
6.6.6. Difusión de resultados . . . . .	183
6.7. Diseño de la interfaz de usuario . . . . .	184
6.7.1. Estructura de la página web . . . . .	184
6.7.2. Estructura de la aplicación móvil . . . . .	184
6.7.3. Accesibilidad . . . . .	185
6.8. Prototipo . . . . .	185
6.8.1. Arquitectura física . . . . .	185
6.8.2. Arquitectura de red . . . . .	186
<b>7. Líneas futuras</b>	<b>191</b>
7.1. Auditoría de seguridad / criptografía . . . . .	191
7.2. Blockchain . . . . .	192
7.3. Procurar escalabilidad del sistema . . . . .	192
7.4. Pruebas . . . . .	192
7.5. Sistema de identificación del votante . . . . .	193
<b>8. Conclusiones</b>	<b>195</b>
<b>Glosario de términos</b>	<b>199</b>
<b>Bibliografía</b>	<b>201</b>
<b>Anexos</b>	<b>205</b>
A. Instalación de Helios . . . . .	205
B. Configuración del Servidor Web del Sistema de Votación . . . . .	207

C.	Exportar los certificados a la App Android . . . . .	210
D.	Ejemplo de un voto en JSON . . . . .	211
E.	Verificar un voto individualmente . . . . .	216
F.	Auditoría de un voto . . . . .	218
G.	Verificar la Totalización de una Elección . . . . .	219



# Índice de figuras

1.1.	El Voto por Internet es un subconjunto del Voto Electrónico. . . . .	23
1.2.	U.S. Patent 0,090,646 – Electrographic Vote-Recorder: Primera patente de Thomas A. Edison. Permitía un voto de tipo 'A favor' o 'En contra' a través de dos interruptores. (1869). Fuente: Wikipedia . . . . .	27
1.3.	Electrographic Vote-Recorder: Fotografía del invento de Thomas A. Edison. Fuente: Rutgers.edu . . . . .	27
2.1.	Categorización de los sistemas de voto . . . . .	31
2.2.	Tipos de e-Voting [16,53] . . . . .	32
2.3.	Participación histórica de elecciones con i-voting en Estonia. . . . .	44
2.4.	Histórico: Porcentaje de voto presencial comparado con el i-Voting en Estonia. . . . .	44
2.5.	Histórico: Porcentaje de voto por Internet frente al total de votos en Estonia. . . . .	45
2.6.	Primera aproximación de funcionalidad de SEVI . . . . .	53
2.7.	Diagrama de secuencia del procedimiento para una elección con ADDER [32]. . . . .	57
2.8.	Arquitectura de ADDER. . . . .	58
2.9.	Comparativa Ágora - Helios en cuanto a características de Proyecto, por Codina Lligoña [16] . . . . .	62
2.10.	Comparativa Ágora - Helios en cuanto a requisitos del Voto por Internet de Proyecto, por Codina Lligoña [16] . . . . .	62
2.11.	Diagrama mezclado mixnet . . . . .	69
2.12.	Diagrama de mixnet con descifrado de mensajes . . . . .	69
2.13.	Ejemplo de homomorfismo basado en la operación multiplicación . . . . .	71
2.14.	Anverso de un especimen del DNIe 3.0 . . . . .	79
2.15.	Reverso de un especimen del DNIe 3.0 . . . . .	80
2.16.	Especificaciones más relevantes del DNIe 3.0 . . . . .	80
2.17.	Logo de RFID . . . . .	81
2.18.	Arquitectura de un chip NFC . . . . .	83
3.1.	Facebook es uno de los proveedores de servicio de Single Sign-In . . . . .	97
5.1.	Actores/roles que interactúan con el sistema. . . . .	115
5.2.	Elementos integradores del sistema . . . . .	118
5.3.	Fases de una elección en Helios Voting . . . . .	126

5.4. Fase de Login en Helios Voting . . . . .	127
5.5. Fase de creación de una elección en Helios Voting . . . . .	127
5.6. Mensaje en la web de la Sede Electrónica de la Agencia Tributaria que indica que no se han encontrado certificados en el dispositivo. . . . .	133
6.1. Arquitectura general del sistema . . . . .	144
6.2. Diagrama ER del Servidor de Autenticación . . . . .	152
6.3. Arquitectura del sistema . . . . .	153
6.4. Diagrama ER del Servidor de Votación . . . . .	154
6.5. Clases del módulo central de Helios. . . . .	156
6.6. Carpeta auth_system que contiene los módulos de autenticación del sistema.	157
6.7. Arquitectura lógica de la App sobre DNIeDroid [18] . . . . .	161
6.8. Página de inicio en navegador de dispositivo Android. . . . .	162
6.9. Menú de aplicaciones de dispositivo Android . . . . .	163
6.10. App Autenticación Android: Selección del CAN cuando no hay ningún documento guardado previamente. . . . .	163
6.11. App Autenticación Android: Pantalla para introducir el CAN. . . . .	164
6.12. App Autenticación Android: Selección del CAN. . . . .	164
6.13. App Autenticación Android: Error al introducir un CAN incorrecto. . . . .	165
6.14. App Autenticación Android: Esperando que se aproxime el DNIe 3.0. . . . .	165
6.15. App Autenticación Android: Cargando certificados. . . . .	166
6.16. App Autenticación Android: Introducción del PIN del DNIe 3.0 . . . . .	166
6.17. App Autenticación Android: Lanzando conexión segura con los certificados.	167
6.18. App Autenticación Android: Recepción de código oAuth. . . . .	167
6.19. App Autenticación Android: Los certificados autofirmados no son seguros, pero la comunicación va cifrada de todos modos. . . . .	168
6.20. App Autenticación Android: Acceso al sistema de voto con login con el DNIe.	168
6.21. Topología de la red con los servidores compartiendo red. . . . .	169
6.22. Topología de la red con los servidores en redes diferentes. . . . .	170
6.23. Topología de la red con los servidores y los dispositivos en red privada. . . . .	171
6.24. Flujo oAuth para servicio de autenticación . . . . .	173
6.25. Esquema de la Cabina de Votación . . . . .	175
6.26. Diagrama de estados del proceso de votación . . . . .	176
6.27. Carga de la cabina de votación. . . . .	176
6.28. Inicio de la cabina de votación. . . . .	177
6.29. Elección de candidatos . . . . .	177
6.30. Pantalla de espera durante el proceso de cifrado del voto. . . . .	179
6.31. Pantalla de revisión del voto. . . . .	179
6.32. Pantalla de emisión del voto. . . . .	180
6.33. Pantalla de confirmación de voto emitido. . . . .	180
6.34. Pantalla con información para auditar el voto y publicarlo. . . . .	180

6.35. Pantalla del verificador de voto individual. . . . .	181
6.36. Tabla de resultados del escrutinio de una elección simple. . . . .	183
6.37. Raspberry Pi Model 3 B, utilizada para albergar el Servidor de votación . .	186
6.38. Raspberry Pi Model B (Rev. 2.0, 512Mb), utilizada para albergar el Servidor de identificación y autenticación . . . . .	186
6.39. Router TP-LINK TL-WR702N utilizado para la construcción del prototipo.	187



# Índice de tablas

2.1. Principales razones para considerar la adopción del Voto por Internet . . . . .	43
2.2. Evolución del voto por Internet en Estonia . . . . .	43
2.3. Comparativa de la evaluación de los sistemas de Votos por Internet de Estonia y Noruega conforme al criterio de Seguridad de la Información . . . . .	48
2.4. Comparativa de la evaluación de los sistemas de Votos por Internet de Estonia y Noruega conforme a los criterios de verificación, auditoría y procedimiento . . . . .	48
2.5. Comparativa Ágora - Helios, por Codina Lligoña [16] . . . . .	63
2.6. Resumen de ventajas y desventajas de los esquemas de voto electrónico según Morales Rocha [40] (p. 109) . . . . .	78
2.7. Comparativa DNIe - DNIe 3.0. . . . .	82
5.1. Principales motivos de elección de Helios Voting como base del Sistema . . .	121
6.1. Detalles técnicos de las Raspberry Pi utilizadas en el prototipo. . . . .	188
6.2. Port forwarding en la red de desarrollo. . . . .	189



A man without a vote is a man  
without protection.

---

Lyndon B. Johnson<sup>1</sup>

## Capítulo 1

# Introducción

Un sistema de voto por Internet, como la mayoría de los proyectos IT, dispone de un marco de posibilidades de desarrollo enorme. Se puede diseñar un sistema basado en sufragio desde cabinas electorales gestionadas electrónicamente que envían el conteo a través de la red (ejemplo en 2.3.1). También es factible optar por un sistema completamente distribuido en el cual el votante puede ejercer su derecho al voto desde cualquier dispositivo electrónico y cualquier lugar del planeta, enviando el contenido de su voto a través de Internet a la autoridad de recuento electoral (ver experiencias en 2.2).

Ambas visiones son ejemplos diferentes de lo que se puede entender como voto por Internet (i-voting), denominado en muchos ámbitos también como voto electrónico (e-voting). Este término no es erróneo, sino incorrecto, ya que peca de inexactitud al llamar a un subconjunto con el nombre del conjunto que lo contiene. El voto por Internet, es un tipo de voto electrónico, pero no todos los sistemas de voto electrónico se realizan a través de Internet (ver figura 1.1).

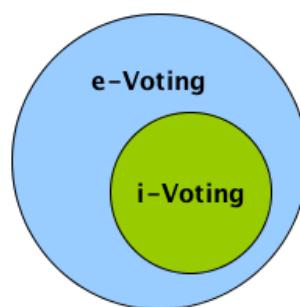


Figura 1.1: *El Voto por Internet es un subconjunto del Voto Electrónico.*

Este proyecto trata de entrar en la problemática del voto electrónico remoto frente al presencial, de las reticencias sociales y tecnológicas que influyen en su reducida implan-

---

<sup>1</sup>Presidente de los Estados Unidos de América (1963-1969) [https://es.wikipedia.org/wiki/Lyndon\\_B.\\_Johnson](https://es.wikipedia.org/wiki/Lyndon_B._Johnson)

tación en procesos electorales de gran importancia y alto número de electores. Para ello, vamos a reproducir la situación a escala reducida. Plantearemos una posible solución para llevar a cabo las Elecciones a la Junta de Escuela de la Escuela Politécnica Superior de la Universidad San Pablo - CEU.

Con este planteamiento es obvio que no vamos a solucionar las trabas técnicas y sociales del voto por Internet a nivel de unas elecciones legislativas a niveles territoriales. Es un tema que se escapa del objetivo de este PFC. Aunque sí que se va a tratar de identificar algunos de los agentes que influyen en los procesos electorales y buscar una posible solución, aplicándola a menor escala a la elección a la Junta de Escuela.

Así, perseguiremos dos objetivos. Por un lado, estudiar la dificultad existente para la implantación del voto por Internet en las elecciones nacionales. Por otro, un soporte electrónico al proceso electoral de la Universidad, buscando aportarle una mejora significativa respecto a comicios anteriores.

Para poder acometer el proyecto, previamente habrá que definir el tipo de votación que se quiere implementar.

De nuevo, hay que tener en cuenta que, aunque se hable de voto electrónico, la intención de este Proyecto de Fin de Carrera es desarrollar una solución de voto por Internet. Un sistema que permita a los votantes la libertad de poder ejercer el voto desde cualquier lugar y dispositivo, con el único de requisito de que éste posea una conexión a Internet, sin necesidad de presencia física en un local de votación.

Este detalle, tratar de separar el concepto del voto por Internet del concepto de voto electrónico, aunque puede parecer trivial, en realidad es fundamental. En el capítulo 2.1 se ahondará en ello, pero como avance se puede destacar que una de las grandes diferencias a tener en cuenta es que con el voto electrónico remoto se pueden utilizar máquinas de votación (que también podrían emitir el voto por Internet), las cuales pueden generar un recibo con el voto emitido por el votante, al estilo de las papeletas que llenan la urna electoral, mientras que con el voto por Internet puro, esto no es tan obvio. Con este mecanismo, la auditoría es más simple para el voto electrónico con máquinas en el centro de votación, pues se podrían contar las papeletas generadas.

Si no se generan estos recibos ni hay una urna física donde se depositan, ¿cómo podemos estar seguros de que hemos votado? ¿Qué ocurre si el sistema tiene fallas y no se contabilizan (o lo hacen de forma incorrecta) los votos, teniendo en cuenta que puede ser imposible un conteo físico de papeletas al no existir estas? Como estas, hay muchas cuestiones a las que el voto por Internet debe dar respuesta de forma fiable antes de poder llevar a cabo su implantación en procesos electorales tradicionales.

La forma de llegar a la solución buscada debe comenzar identificando los factores que afectan a un proceso electoral general y, a continuación, personalizar los que se encuentran

en el que vamos a estudiar.

Una vez identificados estos agentes, definiremos las fases que comportan unas elecciones y estudiaremos cómo podrían ser apoyadas tecnológicamente, evaluando cómo llegar al punto óptimo de integración con el sistema tradicional para mejorar el proceso.

La primera fase se concentrará en desarrollar los sistemas asociados a la fase preelectoral. En ella, se recoge el censo electoral y se identifican tanto los candidatos como los diferentes cargos que se votan.

La segunda fase, la electoral, la identificamos con los procesos que se requieren durante el periodo que dura la elección (ya sea un día o varios). Esta consistirá en desarrollar los sistemas de identificación y validación de votantes, el sistema de votación y el de totalización de los votos.

Para la última fase será necesario proveer de mecanismos para la difusión de los resultados electorales, así como de una auditoría fiable de la integridad de la elección y el resultado de la misma.

Dentro del objetivo de implantar el voto por Internet, será necesario encontrar una forma de identificar remotamente al votante sin que este se presente ante los miembros de la mesa electoral para que den fe de su identidad y le permitan el voto. Por contra, tratamos de reducir a cero la necesidad de que el votante tenga que acudir a ningún local de votación. Será interesante estudiar qué tecnologías nos podrían permitir resolver este problema, introduciendo el uso del DNIe 3.0, con el que existe la posibilidad de identificarse digitalmente sin necesidad de poseer un lector de tarjetas. Es una oportunidad para el voto desde dispositivos móviles. Y una puerta de entrada para el uso de este documento en comicios electorales digitales, pues *este sería el primer desarrollo de estas características hasta la fecha*. Una seria prueba de concepto.

## 1.1. Motivación del Proyecto

El Proyecto Fin de Carrera que presento consiste en el diseño e implementación de un sistema de voto por internet para las Elecciones a la Junta de Escuela de la Escuela Politécnica Superior de la Universidad San Pablo CEU.

Desde que salí de la Universidad y me incorporé al mundo laboral, el sector en el que me he desempeñado profesionalmente ha sido el de los Procesos Electorales.

Tras varios años en el sector, he participado en conversaciones acerca de las elecciones, los sistemas de información que se usan, los que podrían usarse, los que se han denostado y los que se vislumbran para el futuro.

Una de estas conversaciones fue con un profesor de esta misma Escuela, unos días

después de unas Elecciones Generales, en las que me comentó que había probado un sistema piloto implementado para aquellos comicios para el municipio de Madrid, el cual tenía por nombre Mesa Administrada Electrónicamente. Hablamos de que usando su DNIe, se identificó ante el Presidente de la Mesa Electoral y, a través de un portátil, se le encontró en el censo, se le marcó como votado e, incluso, se le imprimió un justificante de votación. Todo esto en poco tiempo. Tras indagar lo que opinaba del sistema, me dijo que lo que realmente esperaba es que para las siguientes elecciones, ni siquiera tuviese que salir de casa, que pudiese votar desde allí, por internet.

Ese fue el germe de este PFC, la idea de encontrar una solución viable al voto por Internet. No obstante, realizar este estudio con un ámbito nacional puede realizar muy ambicioso. Por ello, la idea viró hacia la realización de un sistema más pequeño, como son unas elecciones en la Escuela, pero ambicioso y escalable, buscando construir una prueba de concepto para un sistema robusto que pueda cubrir las necesidades funcionales, de seguridad y de confianza necesarias para unas elecciones a nivel estatal.

Siempre me ha interesado el debate que surge cuando se observa que pese al ritmo de evolución y progreso de los sistemas de información y las nuevas tecnologías en todos los ámbitos, en el electoral se seguía ejerciendo el voto como cuando voté por primera vez a los 18 años. Otra motivación, pues, es tratar de entender los argumentos que hacen que el desarrollo tecnológico en este campo no parezca evolucionar, a simple vista, a la misma velocidad que en el resto de ámbitos cotidianos.

Un elemento clave para la elaboración de este PFC, además del desarrollo web y la aparición de los dispositivos móviles como smartphones y tablets ha sido la difusión del DNIe y, en especial, la aparición de la versión 3.0. Con esta nueva versión del documento se abría la posibilidad de identificar a un votante con un móvil sin necesidad de elementos de software externos. La posibilidad de votación remota segura. Era una buena oportunidad para realizar este PFC e implementar el primer prototipo de sistema de votación electrónica con este tipo de tecnologías.

## 1.2. Antecedentes

En la actualidad, son muchos los proyectos que tratan de incluir el voto electrónico en los procesos electorales por todo el planeta. Estos intentos, de hecho, no se limitan a pequeños sufragios de entidades privadas o gobiernos locales, ya que se han propuesto múltiples paradigmas diseñados para ser implementados en elecciones a nivel estatal, como se puede ver en las experiencias de Estonia, país en el que se desarrolla la votación electrónica remota vinculante con mayor censo activo.

El voto electrónico se lleva tratando de desarrollar e implementar desde hace bastante tiempo. Concretamente, podríamos datar el comienzo en el año 1868, cuando el inventor

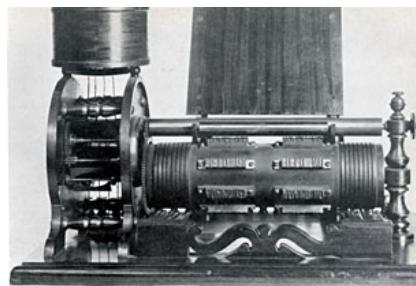
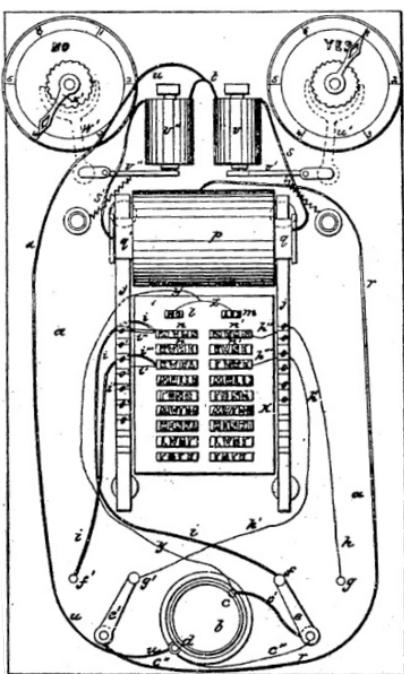


Figura 1.3: *Electrographic Vote-Recorder: Fotografía del invento de Thomas A. Edison. Fuente: Rutgers.edu*

Figura 1.2: U.S. Patent 0,090,646 – Electro-graphic Vote-Recorder: Primera patente de Thomas A. Edison. Permitía un voto de tipo 'A favor' o 'En contra' a través de dos interruptores. (1869). Fuente: Wikipedia

estadounidense Thomas Alva Edison<sup>2</sup> registró su primera patente, consistente en un instrumento simple para el recuento mecánico de votos. El instrumento se podía colocar en la mesa delante de cada congresista y tenía dos botones, uno para el voto a favor y otro para el voto en contra. Pese a considerarlo un avance, no consiguió ser aceptado en el Congreso de Washington, donde le dieron el siguiente motivo para argumentar el rechazo de los representantes a esta nueva tecnología:

*"Si hay en la tierra algún invento que no queremos aquí, es exactamente el suyo.*

*Uno de nuestros principales intereses es evitar fraudes en las votaciones, y su aparato no haría otra cosa que favorecerlos”.*

A partir de este intento, el voto electrónico ha avanzado tecnológicamente y socialmente, logrando herramientas más sofisticadas y seguras en conjunción con un entendimiento, comprensión y, en algunos casos, aceptación de su uso. Estos factores han hecho posible que se hayan podido implementar soluciones e integrarlas en procesos electorales reales, ya sea a nivel nacional o de entidades o estamentos.

Se considera que el inicio del desarrollo del voto electrónico moderno está datado en 1964, año en el que siete condados de EEUU utilizaron un sistema de voto electrónico para las Elecciones Presidenciales.

<sup>2</sup>Inventor estadounidense (1847-1931) [https://es.wikipedia.org/wiki/Thomas\\_Alva\\_Edison](https://es.wikipedia.org/wiki/Thomas_Alva_Edison)

Tras el paso por las máquinas DRE<sup>3</sup> y las tarjetas perforadas, en el año 1981 se produce un hito en la historia del voto electrónico. El criptógrafo David Chaum<sup>4</sup> publica [13] la primera propuesta de un sistema criptográfico diseñado para proteger la privacidad del votante. Entre otras propuestas, en esta publicación se recoge su primera idea de un sistema verificable *end-to-end*, basado en el uso de redes mixtas (2.5.1.4) (mix networks o mixnets).

En la época actual, prácticamente la totalidad de los procesos electorales implementan soluciones tecnológicas para algunos pasos de la fase de recuento de votos, con el fin de un recuento más rápido de votos y una difusión de resultados sólo varias horas después del cierre de los colegios electorales. Es el caso de, por ejemplo, España, donde tanto el Ministerio del Interior como las Comunidades Autónomas requieren que el conteo provisional de los votos se realice en un tiempo corto durante la misma noche electoral.

En este caso, las empresas adjudicatarias implementan soluciones tecnológicas para llevar a cabo el proceso. Así, se puede destacar la tecnología electoral de empresas como Indra<sup>5</sup> - pionera en uso de tablets para enviar información de recuento o la Mesa Administrada Electrónicamente que agiliza y da soporte a la labor de los miembros de mesa, desde gestión del censo a generación de actas de escrutinio - o Scytel<sup>6</sup>, referente a nivel mundial en proyectos de voto electrónico, tanto presencial como remoto.

### 1.3. Organización de la memoria del PFC

En el Capítulo 2 (Estado de la cuestión), se presenta el estado actual de las tecnologías utilizadas para el desarrollo de este Proyecto. Se plantean las características del voto electrónico y mecanismos criptográficos, junto con experiencias de proyectos reales de Voto por Internet, tanto a nivel legislativo como organizacional.

En el Capítulo 3 (Planteamiento), se describen los objetivos y alcance del Proyecto, junto con las distintas fases que definen un proceso electoral.

En el Capítulo 4 (Riesgos), se incluye una lista de riesgos que pueden comprometer el éxito del proyecto al llevarlo a cabo en un proceso electoral real.

En el Capítulo 5 (Análisis del sistema) se plantean elementos necesarios para el estudio de la solución a implementar. Se incluye la especificación de requisitos, actores que intervienen, así como los motivos que dirigen el desarrollo hacia un proyecto de integración, incluyendo el esquema de voto electrónico a utilizar y el sistema sobre el cual se basa esta integración, Helios Voting.

En el Capítulo 6 (Solución) se describen las soluciones implementadas para cumplir

---

<sup>3</sup>[https://en.wikipedia.org/wiki/DRE\\_voting\\_machine](https://en.wikipedia.org/wiki/DRE_voting_machine)

<sup>4</sup>[https://es.wikipedia.org/wiki/David\\_Chaum](https://es.wikipedia.org/wiki/David_Chaum)

<sup>5</sup><http://www.indracompany.com/es/procesos-electorales>

<sup>6</sup><http://www.scytl.com/>

con las necesidades identificadas. Incluye la descripción de un prototipo funcional.

En el Capítulo 7 (Líneas futuras) se recogen una serie de ideas que podría resultar interesante estudiar y llevar a cabo usando como base este proyecto, con el objetivo de mejorarlo o de encontrar otras posibles implementaciones.

En el Capítulo 8 (Conclusiones) se exponen algunas consideraciones tanto objetivas como de libre opinión acerca del voto por Internet, tecnologías y resultados observados durante el desarrollo de este PFC.

## 1.4. Metodología

### 1.4.1. Documentación

Para la redacción del documento de la memoria de este PFC se ha utilizado el lenguaje LATEX.

Tanto la memoria como el código fuente del proyecto se ha versionado con la herramienta Git usando la web [github.com](https://github.com). Se puede visitar el proyecto, proponer pull requests o realizar un fork en el repositorio público <https://github.com/Betisman/pfc-carlosjg>.

El enlace a la última versión de la memoria es <https://github.com/Betisman/pfc-carlosjg/raw/master/doc/memoria/pfc.pdf>.

### 1.4.2. Metodología de desarrollo

En el desarrollo de proyectos de software existen múltiples de metodologías. Una de ellas basa el desarrollo en 5+2 etapas:

- |                   |                   |                  |
|-------------------|-------------------|------------------|
| 1. Análisis       | 4. Implementación | 7. Mantenimiento |
| 2. Especificación | 5. Prueba         |                  |
| 3. Diseño         | 6. Documentación  |                  |

Aunque para este proyecto, la metodología a emplear se basa en estas fases, por las características el mismo habrá diferencias, sobre todo en cuanto a la distribución de recursos y tiempo entre ellas.

A diferencia de los proyectos de software evolutivos o mantenidos en el tiempo, este proyecto está diseñado para ser puesto en producción durante un relativamente corto espacio de tiempo. Por esto, la fase de Mantenimiento no merece disponer de recursos abundantes, puesto que se limitaría al tiempo en el que el sistema está en producción, lo cual será desde

un día a unos pocos, lo que la Autoridad Electoral considere oportuno que estén las urnas abiertas junto con el tiempo empleado para el conteo de resultados y su difusión.

Las fases de análisis y diseño, como en todos los proyectos son realmente importantes, en este caso, además, porque tratan datos muy sensibles, como es la elección de representantes y tienen que lidiar con problemas como el anonimato del votante rompiendo la asociación voto-votante, la autenticidad del votante, verificabilidad del voto, etc. Además, es la fase de desarrollo donde se recopilan los requisitos del proyecto, en este caso incluyendo aquellos intrínsecos al voto electrónico.

La fase de implementación, si el diseño ha sido bien desarrollado debería llevar el tiempo necesario para realizar el software y la integración de los diferentes módulos y sistemas. Sin embargo, la fase de Pruebas cobra una importancia capital en este tipo de proyectos. Al tener una vida tan corta y una importancia en cuanto a datos tan alta, el margen de error del sistema durante el breve período que estará en producción debe ser residual. Este tipo de sistemas deben tener una tolerancia a fallos de prácticamente el 100 %. No hay opción a realizar sistemas evolutivos, por lo que hay que tratar de que no lleguen errores a producción, ya que los que se encuentren durante la jornada electoral se tendrían que arreglar en el momento, con las implicaciones que esto acarrea en cuanto a riesgo de una mala solución y compromiso con los datos o, incluso, con la transparencia del proceso (técnicos modificando código fuente durante la jornada electoral no es una buena práctica de cara a auditorías externas en un sistema electrónico de voto).

Es indispensable, por tanto, desarrollar un sistema de testing serio y automatizado. Deberían implementarse, por una parte, un protocolo de pruebas que abarque desde las unitarias a las de sistema, teniendo en cuenta tanto datos, como software, hardware y redes. Además, es muy importante que estas sean automáticas y que se ejecuten cada vez que el software sea modificado, asegurando así la integridad del mismo.

La fase de documentación es importante para dejar constancia de cómo se ha desarrollado el proyecto. Junto a los textos que expliquen la implementación del mismo, se debe documentar el protocolo de pruebas, manual de usuario personalizado para cada uno de los actores que interactúan con el sistema, protocolos de mantenimiento, etc. En el caso de este PFC, se incluiría aquí la redacción de esta memoria.

La fase de mantenimiento es muy importante en proyectos evolutivos. Un proyecto que se ejecuta, se implanta en producción y tiene una vida objetiva relativamente longeva, requiere un plan de mantenimiento que asegure que los posibles *bugs* sean corregidos y las posibles nuevas funcionalidades requeridas sean desarrolladas. En el caso de un proyecto para un proceso electoral, esta fase puede no tener esta importancia, pues el proyecto tiene una vida muy corta, normalmente asociada a la noche electoral, por lo que no tiene sentido proveer un mantenimiento activo, sino un protocolo de contingencia ante posibles problemas que haya que arreglar *en directo*.

Las elecciones no resuelven por sí mismas los problemas, aunque son el paso previo y necesario para su solución.

---

Adolfo Suárez<sup>1</sup>

## Capítulo 2

# Estado de la cuestión

### 2.1. Voto electrónico

Cuando se habla de voto electrónico, una primera acepción del término se refiere a los procesos electorales cuyas fases pueden llevarse a cabo haciendo uso de tecnologías de la información. Dentro de estas fases susceptibles de ser implementadas con protocolos informáticos se incluyen el registro de votantes, diseño de mapas de distritos o circunscripciones electorales, y la gestión, administración y logística electoral; así como el escrutinio provisional o definitivo, transmisión de resultados y difusión de los mismos, o el sufragio del voto en sí mismo.

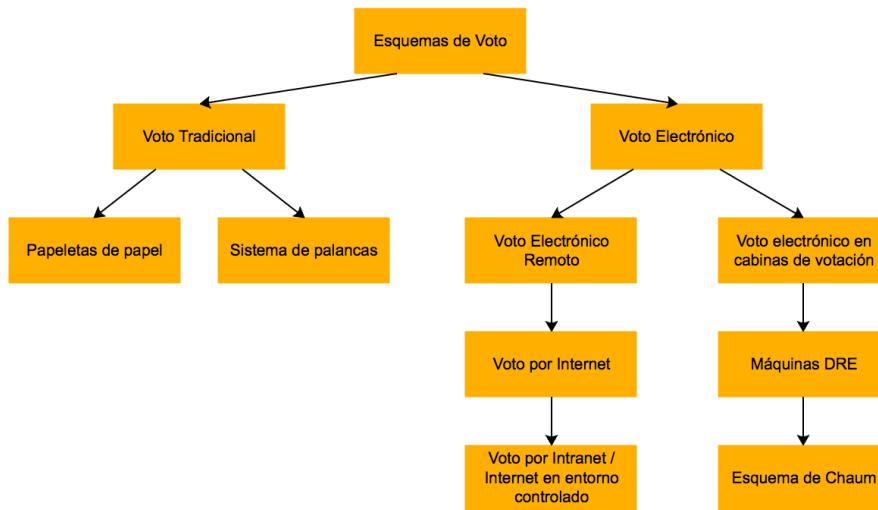


Figura 2.1: *Categorización de los sistemas de voto*

No obstante, una definición más simple del voto electrónico se refiere únicamente a este último acto de votar, ya sea a través Internet o simplemente utilizando sistemas que no

---

<sup>1</sup>Presidente del Gobierno de España (1976-1981) [https://es.wikipedia.org/wiki/Adolfo\\_Su%C3%A1rez](https://es.wikipedia.org/wiki/Adolfo_Su%C3%A1rez)  
Cita del discurso de cierre de campaña [14/06/1977]

estén intercomunicados a través de Internet, aunque sí con un servidor receptor del voto.

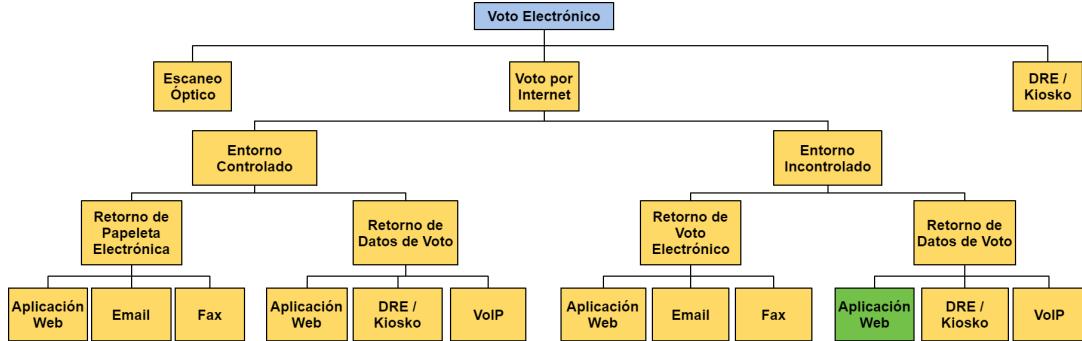


Figura 2.2: *Tipos de e-Voting* [16, 53]

### 2.1.1. Niveles

Podemos estudiar el voto electrónico separándolo en varios niveles, dependiendo de su implantación en el proceso.

- Nivel 0
- Voto electrónico sustitutivo
- Voto electrónico remoto
  - Voto telemático en local de votación
  - Voto por Internet
- Nivel 0

Es el sistema de voto tradicional, sin hacer uso de elementos electrónicos para llevar a cabo ninguna fase del proceso. Es el sistema que se ha venido utilizando desde las primeras votaciones hasta bien entrado el siglo XX y todavía en uso en muchos territorios del planeta.

- Voto electrónico sustitutivo

En este nivel, se sustituyen algunos procedimientos manuales o elementos utilizados en el voto tradicional por sistemas electrónicos determinados. Lo que se intenta es que el proceso de votación sea lo más parecido al que se ha venido llevando a cabo, pero pudiendo utilizar avances técnicos que mejoren el procedimiento en algunos de los puntos del mismo. Así, dependiendo de la legislación, el nivel democrático y social y la aceptación de la innovación tecnológica, se han adoptado procesos electorales en los que se hace uso de algunos elementos tales como tarjetas magnéticas o documento de identidad electrónico (para identificar al votante o incluso para emitir el voto),

urnas de votación electrónica que recuentan los votos de forma automática (RFID, lector código de barras, etc.), pantallas de votación para selección de candidaturas (en EEUU es una de las formas en las que se elige la opción a votar), sistemas de totalización y consolidación de resultados (para evitar el escrutinio manual), e incluso sistemas para guiar el recuento definitivo pasados unos días de la jornada electoral. Así podemos encontrar muchos más ejemplos.

Como se puede observar, todos los sistemas que se tienen en cuenta en este nivel están orientados a sustituir un elemento del proceso tradicional de votación. Todos están pensados para tener una función en el local electoral, ya sea para la identificación del votante, emisión del voto, escrutinio o (en otro tipo de local electoral) recuento definitivo. Aquí podemos observar, de paso, diferentes fases del proceso electoral, que son fácilmente reconocibles.

- Voto electrónico remoto

En este nivel, el concepto del voto traspasa el local electoral común. Se trata de que el voto se transmita desde un punto de votación a una "urna remota". Dependiendo del punto de origen, podemos dividir este grupo en dos subgrupos, uno en el que los diferentes colegios electorales están interconectados entre si y otro en el que el voto se emite desde cualquier punto con conexión a Internet.

- Voto telemático en local de votación

En esta primera aproximación al voto telemático sigue pensándose en el sistema de voto tradicional en cuanto a que el votante ha de acudir a un local de votación acondicionado para ejercer su derecho al voto. En este local, encontraría una serie de sistemas de identificación (tanto personal frente a los miembros de mesa - como en el sistema tradicional - como telemático frente a una autoridad certificadora remota a través de una identificación digital) para superar el primer paso del proceso. Una vez cerrada la votación, se conectarían los diferentes colegios electorales para comunicar cada uno sus escrutinios y pasar los resultados para la fase de totalización.

- Voto por Internet

La aproximación del voto por Internet es la más ambiciosa en términos tecnológicos y de seguridad. En esta, el votante puede ejercer su derecho al voto desde cualquier punto conectado a Internet, como puede ser su propia casa o el lugar en el que se encuentre de viaje. La identificación del votante debe ser digital y remota. El voto emitido tiene que ser transmitido a la urna electrónica remota que corresponda. No obstante, desde un punto de vista sociológico, este sistema tiene todavía una serie de retos que debe cumplir, como es el acceso universal al proceso de votación, ya que es complicado asegurar que la totalidad de la población podría hacer uso de un sistema informático de este tipo. Además, encontramos dificultades en cuanto a fraude electoral, ataques al sistema,

tolerabilidad al fallo, etc.

El término voto electrónico se utiliza también para referirse a los sistemas electorales que tratan de automatizar algunas fases del proceso, como son la autenticación de votantes, la votación y el escrutinio de los votos y/o difusión de los resultados. En el nivel 3 de la clasificación anterior se pueden encuadrar todos estos sistemas que, además de hacer uso de tecnologías de la información para la automatización de estos procesos, se basen en una comunicación de redes telemáticas para interconectar votantes con mesas electorales - urnas digitales - y estas con los centros de procesamiento de resultados.

Son estos sistemas los que están en auge para los investigadores de protocolos electrónicos electorales. Con el aumento de la participación ciudadana en Internet, en la sociedad digital, los usuarios realizan todo tipo de procesos cotidianos a través de la red de forma remota, ya sea interactuar con las entidades estatales o municipales con trámites burocráticos, multas o pagando impuestos, gestionando los recursos familiares o de la empresa con el banco desde casa o el despacho, o incluso compras por Internet o consumición de ocio digital. Con este panorama es cuestión de tiempo que cierto sector de la población demande una actualización de los procesos de votación, igualándolos a las posibilidades de ubicuidad de los que disfrutan el resto de servicios comentados. He aquí donde el voto electrónico remoto tiene que estudiarse si es el candidato ideal para cubrir este nicho o si, por el contrario, los riesgos de seguridad y procedimiento que sus detractores le achacan realmente imposibilitarán este cambio en un corto período de tiempo.

La introducción de este paradigma en el mundo de los procesos electorales plantea para los expertos, por tanto, un conjunto de retos muy importante, tanto desde el punto de vista tecnológico - sobre todo a nivel de seguridad del sistema y privacidad del votante - como a nivel social, ya que estos sistemas electrónicos deben garantizar al votante al menos la misma confianza que la que le proporciona el sistema de voto tradicional, para que se pueda plantear aceptar el cambio.

### 2.1.2. Verificabilidad vs. Secreto

El voto electrónico es uno de los retos más importantes y complejos del mundo tecnológico hoy en día. Aunque pueda parecer que en otras áreas más modernas hay desarrollos mucho más complejos, el problema del voto electrónico sigue teniendo factores de difícil resolución.

Uno de los retos más complicado de resolver para los desarrolladores de estos protocolos y sistemas es el de la *dualidad Verificabilidad-Secreto*.

Aquí se enfrentan dos requisitos básicos de un sistema de voto electrónico: la posibilidad de un elector de **verificar** que el voto que ha emitido ha sido correctamente incluido en el escrutinio y el derecho fundamental del **secreto** de voto del propio elector.

Con el fin de visualizar el problema, planteo un ejemplo típico.

Para comenzar con el ejemplo, introducimos una serie de actores:

**Marta y Begoña:** dos votantes.

**Rafael:** quiere influenciar el voto de Marta.

**Alphas y Betas:** las dos opciones entre las que los votantes han de elegir en la Elección.

Marta tiene intención de votar por los Alphas, mientras que Begoña desea votar a los Betas. Por su parte, Rafael tiene un gran interés en que Marta vote por los Betas.

Como hemos avanzado, en cualquier proceso electoral hay un conflicto entre la verificabilidad y el secreto. Cualquier votante querría verificar que el proceso de su voto ocurre correctamente, desde su inclusión en la urna hasta el preciso conteo en el escrutinio. De forma particular, Marta quiere verificar que su voto es apropiadamente escrutado como Alphas. No obstante, si Marta consigue suficiente información como para poder demostrar a Rafael de que votó por los Betas, aparece la amenaza de la compraventa de votos. Sabiendo que el voto emitido es demostrable a un tercero, antes de que Marta vote, Rafael podría ofrecerle dinero u otros recursos a cambio de que su voto sea para los Betas en vez de para los Alphas.

Naturalmente, junto con la amenaza de la compraventa de votos, aparece también la de la coerción. Como Marta puede mostrar el contenido de su voto a Rafael, puede ser coaccionada por éste para que vote por Betas como él quiere, en lugar de los Alphas, como ella pretendía, por miedo a represalias.

Lo que se debe buscar con el voto electrónico es que Marta consiga la información suficiente para verificar personalmente que su voto a Carnívoros ha sido efectivamente emitido y escrutado como Alphas, pero que no consiga la información necesaria como para probar a Juan el contenido de su voto.

Si Marta vota a Alphas y Begoña a Betas, ambas deberían tener la seguridad de que sus votos han sido correctamente incluidos en el escrutinio según sus opciones elegidas. Las dos pueden decirle a Rafael que han votado a Betas, con lo que Marta estará mintiendo y Begoña dirá la verdad, pero Rafael no notará la diferencia. Quizá, al ver Rafael que no tiene seguridad para incentivar o coartar a Marta, desista de tratar de comprar su voto (o de coartarlo) y se acabe así con la amenaza.

Resumiendo, es muy complicado conseguir que los Sistemas de Voto sean confiables ya que presentan requisitos de seguridad que chocan entre ellos:

- El sistema debe asegurar la integridad de la elección para que todos los votantes

estén convencidos de que los votos se cuentan correctamente.

- El sistema debe asegurar la confidencialidad de los votos para proteger la privacidad del votante, prevenir la venta de votos y para defender a los votantes de ser coartados.

La integridad del sistema es fácil de obtener por medio de un muestreo público de los votos emitidos. Pero esto destruiría automáticamente la confidencialidad del voto.

La confidencialidad se puede obtener a través del voto secreto, pero con ello es muy difícil asegurar la integridad del sistema.

Debido a la naturaleza de unas elecciones, la violación de cualquiera de estos requisitos del voto electrónico puede derivar en consecuencias dramáticas. Por tanto, el reto del diseño de un sistema electoral digital reside en encontrar el punto medio en el que asegurar la integridad del sistema no comprometa el derecho del votante al secreto de su voto.

### 2.1.3. Requisitos del voto electrónico

Un dogma que deben cumplir los sistemas de voto electrónico es la consigna de aportar al proceso al menos las mismas garantías de seguridad que el sistema tradicional al que está sustituyendo / complementando.

El voto presencial tradicional permite un recuento de la votación una vez acabado el proceso y abierta la urna que contiene los votos físicos.

En el siguiente nivel, lo mismo le ocurre a ciertos sistemas de voto electrónico que hacen uso de urnas digitales pero generan un recibo o papeleta física, que almacenan también como una urna física.

En cuanto al último nivel, el correspondiente al voto electrónico remoto, esto no está tan claro, pues la mayoría de estos sistemas no generan un resguardo físico de los votos electrónicos emitidos, por lo que es complicado pensar en un recuento en caso de fallo o de duda de la autoridad electoral o del propio electorado.

#### 2.1.3.1. Fujioka, Okamoto y Ohta

Según publican *Fujioka, Okamoto y Ohta* [27], un sistema de voto secreto es *seguro* si cumple con los siguientes requisitos:

- **Completitud (Completeness):** Todos los votos válidos son contados correctamente.
- **Solidez (Soundness):** Un votante deshonesto no puede interrumpir la votación.
- **Privacidad (Privacy):** Todos los votos deben ser secretos.

- **Unicidad (Unreusability):** Ningún votante puede votar dos veces.
- **Elegibilidad (Elegibility):** Nadie que no tenga permitido el voto puede votar.
- **Fiabilidad (Fairness):** Nada debe afectar la votación.
- **Verificabilidad (Verifiability):** Nadie puede falsificar el resultado de la votación.

A estos requisitos básicos, el equipo de Fujioka añade otros seis que considera importantes para la correcta implementación de un sistema de voto electrónico:

- **Robustez (Robustness):** El sistema debe ser capaz de tolerar una cierta cantidad de condiciones de fallas, a la vez que debe ser capaz de manejar y responder a estas situaciones.
- **Verificabilidad Universal (Universal Verifiability):** Cualquier actor debe poder verificar el resultado de las votaciones.
- **Sin recibo (Receipt Freeness):** El votante no necesita una prueba del voto realizado, debe ser incapaz de probar a un tercero el contenido de su voto
- **Incoercibilidad (Incoercibility):** El votante no puede ser coartado por un tercero para que vote por una opción en concreto. Se debe asegurar la libertad del voto.
- **Sin duplicados (Non-Duplication):** Nadie puede duplicar el voto de otra persona.
- **Participación Pública (Public Participation):** La lista de quiénes votaron o quiénes no lo hicieron ha de ser pública.
- **Corrección Privada de Errores (Private Error Correction):** El votante tiene la capacidad de probar que su voto no fue contado correctamente sin tener que revelar qué opción votó.

### 2.1.3.2. Universidad de Extremadura

A partir de esta primera definición de los requisitos del voto electrónico, muchos equipos de desarrolladores o teóricos de infraestructuras para el voto electrónico han redactado sus propias interpretaciones, aunque suelen ser análogas a las ofrecidas por Fujioka. Por ejemplo, estas son las propiedades que debe tener un sistema de voto electrónico a través de Internet, según publican desde la Universidad de Extremadura [17] son las siguientes:

- **Universal** Todas las personas que tienen derecho al voto deben poder hacerlo usando el sistema telemático.

Por ello, debe cumplir con los requisitos para acceso al proceso a través de Internet si es el caso.

- **Libre** Las personas con derecho a voto tienen la libertad para escoger si emitir su voto a través del sistema por Internet o de alguna otra forma que se haya implementado para llevar a cabo el proceso. No se les puede imponer un sistema por encima de otro. Además, los votantes tienen la libertad de elección en la forma y contenido del voto, incluso en si desean abstenerse.

Los votantes han de ser agnósticos en cuanto a la tecnología del sistema, por lo que este debe permitirles el voto sin importar el sistema operativo, navegador o dispositivo móvil que utilicen. De la misma forma, debe superar las barreras de accesibilidad física o social.

- **Directo** Los votantes son los encargados de realizar su voto, sin posibilidad de delegarlo en otra persona.

Este requisito plantea retos en cuanto a la identificación y autenticación del votante, requiriendo la aplicación de protocolos criptográficos en el proceso.

- **Igual** Todos los votantes han de ser iguales frente al sistema de votación.

Este requisito admite algunas salvedades, como pueden ser votantes con necesidades especiales.

- **Secreto** Cada votante es la única persona que puede conocer el contenido de su voto. Además, ningún voto debe poder ser asociado al votante que lo ha emitido.

Hasta aquí están las características inherentes a un sistema de votación tradicional, el realizado hasta ahora en cualquier proceso electoral que haya habido en España, por ejemplo. A continuación, añaden una serie de propiedades ligadas al voto electrónico:

- **Autenticación** Sólo se permite el voto a los votantes que hayan sido correctamente autorizados.

En caso de procesos con votación mixta, por Internet y de forma presencial, el censo electrónico ha de ir actualizándose para contener los votantes de ambas posibilidades con el fin de evitar que un votante vote en más de una ocasión, como podría ser presencialmente y por Internet.

- **Unicidad** Un votante con derecho al voto sólo debe poder votar una única vez.

Para esto es esencial tener un censo electrónico actualizado con los votantes que emiten su voto tanto presencialmente como por Internet, o cualquier otro canal que se proporcione.

- **Integridad** Todo voto, una vez se ha registrado en el sistema, no puede ser modificado o eliminado de éste.

El sistema debe aportar herramientas de seguridad de los datos, como *backups*, así como para realizar y almacenar auditorías que corroboren la invariabilidad de los votos emitidos.

- **Confidencialidad** El contenido de un voto registrado en el sistema sólo lo conoce el votante que lo emitió. Ninguna entidad puede llegar a averiguar el contenido de algún voto.
- **Fiabilidad** Una vez un voto es almacenado en el sistema, éste debe asegurar que no puede perderlo. Por ello el sistema debe ser tolerante a todo tipo de fallos en dispositivos y conexiones.
- **Flexibilidad** El sistema debe permitir que los votantes puedan emitir su voto a través de Internet con independencia tecnológica respecto a sistema operativo, dispositivo móvil, navegador web, etc.
- **Comodidad** El sistema debe proporcionar a los votantes una buena experiencia de usuario a la hora de votar, con independencia de las habilidades o conocimientos técnicos que posean.
- **Ergonomía** El sistema debe proporcionar ayudas técnicas a los votantes para facilitarles el proceso de voto. Esto incluye ayuda interactiva y gestión de errores.

#### 2.1.3.3. Bokslag y Vries

Otro ejemplo que indica que, pese a las variaciones, hay requisitos fundamentales reconocidos por la mayoría de los autores, nos lleva a una visión muy actual de estos requisitos inherentes al voto electrónico. En el artículo [5], Bokslag y Vries, dos autores holandeses, profundizan acerca del voto electrónico y por Internet en este país. Este artículo enumera una serie de principios fundamentales para cualquier solución de e-voting basándose en los acordados por el Consejo de Europa [39]. Según estos principios, se debe asegurar que el **sufragio** sea:

- **Universal:** Cualquier persona tiene derecho a votar y a postularse para la Elección (entendiendo la existencia de límites basados en condiciones como la edad o la nacionalidad).
- **Igualitario:** Cada votante puede emitir el mismo número de votos.
- **Libre:** Todo votante tiene el derecho de formarse y expresar su propia opinión libremente, sin ser coaccionado por ninguna influencia exterior.
- **Secreto:** Todo votante tiene el derecho de poder votar de forma individual y secreta.
- **Directo:** Los votos emitidos por los votantes deben ser los que directamente determinen la/s persona/s elegida/s en la votación.

Según el mismo artículo, estos principios fundamentales se traducen en los siguientes requisitos:

- **Transparencia/Integridad:** Asegurar que el público general y los asociados al proceso electoral tienen confianza en la solución implementada.
- **Voto secreto/privacidad:** Proteger el secreto del voto en todas las etapas del proceso de votación.
- **Unicidad:** Asegurar que cada voto emitido es contado y que cada uno es contado tan sólo una vez.
- **Derecho al voto:** Asegurar que sólo aquellos votantes con derecho de voto pueden votar en el proceso.
- **Verificabilidad/auditoría:** El votante debería poder comprobar que su voto ha sido correctamente contado en el escrutinio. Si no lo puede comprobar, al menos auditores independientes deberían poder comprobar la integridad de los resultados de la elección.
- **Accesibilidad:** Garantizar la accesibilidad al proceso del mayor número de personas, especialmente aquellas con algún tipo de discapacidad.
- **Libertad del votante/resistencia a la coacción:** Mantener el derecho del votante a expresar su opinión mediante el voto sin coacción o una influencia externa.
- **Disponibilidad:** Asegurar la disponibilidad del sistema durante el tiempo que tiene lugar la votación.

En el artículo mencionado se realiza una valoración muy interesante de varios sistemas de votación electrónica y por Internet con resultados basados en esta clasificación.

#### 2.1.3.4. Resumen

Se han presentado tres definiciones de requisitos del voto electrónico/por Internet.

En primer lugar se han introducido los requisitos "clásicos", definidos por el importante equipo de Fujioka, Okamoto y Ohta en el año 1993. Prácticamente establecieron el punto de partida para las reglas que debían cumplir los sistemas de voto electrónico. A partir de esta definición, me ha parecido interesante aportar otras dos, de entre los incontables ejemplos que se pueden encontrar en la bibliografía especializada, que mostrases las variaciones a partir de estas originales. Para ello, se han recogido las de una Universidad española puntera en proyectos de software libre como es la Universidad de Extremadura, para poner en esta memoria un ejemplo de nuestro país. Y se ha incluido un artículo holandés por ser interesante entre los más recientes publicados, intentando dar una definición de vanguardia.

La conclusión, revisando las tres definiciones, es que los requisitos son interpretables y cada autor propone los que cree convenientes basándose en su propia idea o experiencia.

Pero es claramente visible que pese a esta interpretación libre, existe un conjunto de requisitos que parece que todos estos autores consideran inherentes al voto electrónico, pues, de una forma u otra, los recogen la mayoría de las recopilaciones estudiadas.

Así entre los requisitos básicos vemos que los autores aquí recogidos siempre hablan de *integridad, privacidad, verificabilidad, unicidad, fiabilidad, disponibilidad, derecho de voto* ... Aspectos que debe cumplir todo sistema de votación electrónica para que el proceso pueda ser llevado a cabo con un nivel aceptable de seguridad y confianza por parte de los intervenientes y observadores de la Elección.

## 2.2. Experiencias de Voto por Internet

En lo referente al voto electrónico hay numerosos proyectos llevados a cabo, tanto desde el mundo empresarial como estatal o universitario.

Muchos de ellos se utilizan hoy en día. Se pueden destacar a niveles estatales todas las elecciones en la que se usan urnas electrónicas o máquinas cuenta-votos, como ocurre en países como Venezuela, Estados Unidos, India y varios más. Hay otros estados cuya prioridad es el estudio de la implantación de este tipo de herramientas para sus procesos electorales, como es el caso actual de Argentina. Lo mismo ocurre con muchos proyectos surgidos desde ámbitos académicos o empresariales, donde se desarrollan sistemas que permiten el uso de tecnología para la fase de votación. Incluso algunos permiten el volcado de información de los votos de la urna en el sistema de recuento de voto, aunque una vez la urna ha sido cerrada, no en el momento en el que el votante introduce su voto en el Sistema.

No obstante, la naturaleza de este proyecto implica que nos centremos en aquellos procesos que utilizan la variante del voto electrónico consistente en el voto por Internet, siendo éste transmitido al sistema de recuento en el momento en el que se introduce el voto.

En este sentido, en cuanto al estado de la cuestión del voto por Internet, como hemos destacado, la experiencia más ambiciosa es, sin duda, las elecciones que se llevan a cabo en Estonia (2.2.1) donde, desde el año 2005, se utiliza un sistema de voto por Internet accesible por la totalidad del censo que desee hacer uso de él.

En clave nacional podemos destacar el desempeño de empresas como Scytl<sup>2</sup>, que ha implementado sistemas de voto por Internet para voto desde el extranjero para algunos condados de Estados Unidos, ciertos cantones de Suiza y varias provincias de India, la mayor democracia del mundo (en número de votantes). Otra empresa española, Indra, también tiene soluciones tecnológicas de voto por Internet utilizadas para elegir las cúpulas

---

<sup>2</sup><https://www.scytl.com/es/clientes/>

directivas de organismos como la Guardia Civil, universidades como la UAH o la UNED e incluso de partidos políticos, como es el caso de la dirección de UPyD.

Dentro de las soluciones de voto electrónico telemático, es importante el desarrollo que se ha hecho en el voto por Internet.

Según un estudio<sup>3</sup> de un ente holandés, once países han desarrollado pruebas piloto para elecciones a través de voto electrónico por Internet a nivel nacional, aunque cuatro de ellos ya habían abandonado sus proyectos. Los países que continuaban probando de distinta forma soluciones de voto por Internet y que recoge dicho estudio serían Australia, Canadá, Estonia, Francia, India, Noruega y Suiza. Las motivaciones de los países en desarrollar herramientas de votación remota difieren en cada uno de ellos, dependiendo del tipo de votantes al que va dirigido este tipo de votación. Indica como ejemplo el de Francia, motivado por la necesidad de incrementar la participación electoral de los expatriados o el de Estonia, país que ha apostado por un desarrollo tecnológico en la mayoría de entornos gubernamentales, buscando incrementar la participación tanto de los votantes ocasionales como de acercar a los que se suelen abstener.

Según otro estudio presentado por la municipalidad de Guelph, en Ontario, con vistas a sus propias Elecciones Municipales con voto por Internet, se destacan una serie de razones que diferentes territorios han tenido en cuenta para invertir en el desarrollo e implantación del iVoting. Estas razones se resumen en la tabla 2.1.

Observando esta tabla, se descubre que hay un motivo en el que coinciden por unanimidad los territorios citados, que es la Accesibilidad de los votantes al proceso electoral, permitiendo que votantes con discapacidades o que se encuentren fuera de su circunscripción electoral, por ejemplo, puedan ejercer su voto. Los siguientes motivos tenidos en cuenta para la inversión en Voto por Internet son la búsqueda de un aumento de la participación y el posicionamiento de liderazgo en cuanto a Gobierno Electrónico.

Es interesante destacar de esta tabla que aunque la mayoría apuesta por el Voto por Internet como una herramienta para incrementar la participación electoral, tan sólo dos territorios se centran en la participación de los jóvenes. De hecho, Noruega, que es uno de esos dos territorios, ni siquiera esgrime el motivo de la participación electoral como uno que le lleve a invertir en iVoting por encima de otros.

### 2.2.1. Estonia

Estonia es quizás el ejemplo más destacado en cuanto a la utilización del voto por Internet en elecciones a nivel estatal. Desde el año 2005 lleva usando una solución de voto electrónico remoto no presencial complementando al voto tradicional.

---

<sup>3</sup><https://www.jbisa.nl/download/?id=17700076&download=1>

Territorio	Razones para adopción del Voto por Internet						
	Participación electoral	Liderazgo en Gobierno Electrónico	Accesibilidad	Conveniencia	Foco en servicio centrado en el ciudadano	Incrementar la participación de los jóvenes	Eficiencia en el escrutinio
Estonia	•	•	•				
Suiza	•	•	•	•			
Noruega			•			•	•
Edmonton		•	•	•	•		
Markham	•	•	•	•	•		
Halifax	•		•				
Cape Breton	•	•	•				
Truro	•	•	•	•		•	
Resumen	6/8 75 %	6/8 75 %	8/8 100 %	4/8 50 %	2/8 25 %	2/8 25 %	1/8 12,5 %

<sup>a</sup> Fuente: Presentación del Voto Online para las Elecciones Municipales de Guelph (Ontario, Canadá) en 2014<sup>b</sup>

<sup>b</sup> <https://www.youtube.com/watch?v=FJ2rHI8NNBk>

Tabla 2.1: Principales razones para considerar la adopción del Voto por Internet

El impacto del voto electrónico sobre el electorado estonio ha ido evolucionando en cada comicio. En el 2005, el primer año en que se comenzó a utilizar, no llegó al 2 % de los votantes los que se decantaron por hacerlo por Internet, mientras que en el 2014 y 2015, este porcentaje superó el 30 % de los sufragistas (ver tabla ??).

Elección	Tipo	IV <sup>a</sup>	% IV-TV <sup>b</sup>
2005	Elecciones Locales	9.317	1,90 %
2007	Elecciones Parlamentarias	30.275	5,50 %
2009	Elecciones Parlamento Europeo	58.669	14,70 %
2009	Elecciones Locales	104.413	15,80 %
2011	Elecciones Parlamentarias	140.846	24,30 %
2013	Elecciones Locales	133.808	21,20 %
2014	Elecciones Parlamento Europeo	103.151	31,30 %
2015	Elecciones Parlamentarias	176.491	30,05 %

<sup>a</sup> IV: Votantes que votaron a través de Internet.

<sup>b</sup> %IV-TV: % de votantes que votaron a través de Internet sobre el total de votantes.

c Fuente: Comisión Nacional Electoral de Estonia / Vabariigi Valimiskomisjon<sup>d</sup>

d <http://www.vvk.ee/voting-methods-in-estonia/engindex/statistics>

Tabla 2.2: Evolución del voto por Internet en Estonia<sup>c</sup>

Estonia es el primer estado que utiliza, oficialmente, el voto electrónico remoto por Internet de forma vinculante. Este sistema puesto en práctica en el año 2005 es una parte de un plan de modernización del país báltico. De hecho, previamente a la puesta en producción del sistema electoral, se comenzó a desarrollar en el año 2000 un despliegue técnico importante para la implantación del documento de identidad electrónico, junto con mecanismos de comunicación con la Administración para facilitar los trámites con la misma por parte de los ciudadanos de forma electrónica y remota.

La ley electoral estonia permite a los votantes ejercer su derecho al voto de tres formas:

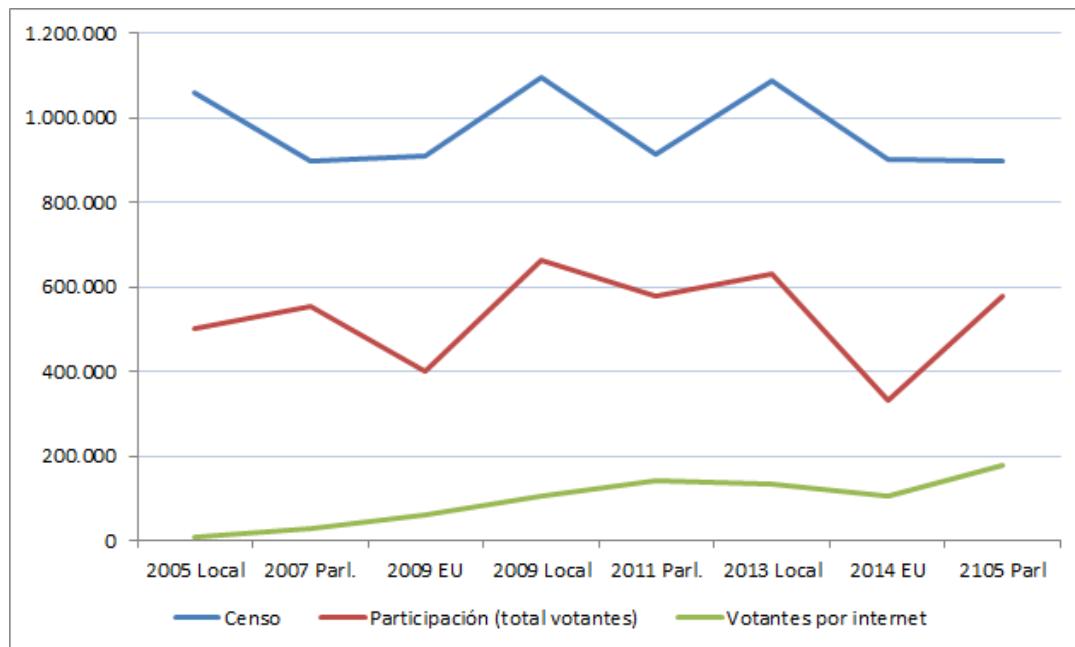


Figura 2.3: Participación histórica de elecciones con i-voting en Estonia.

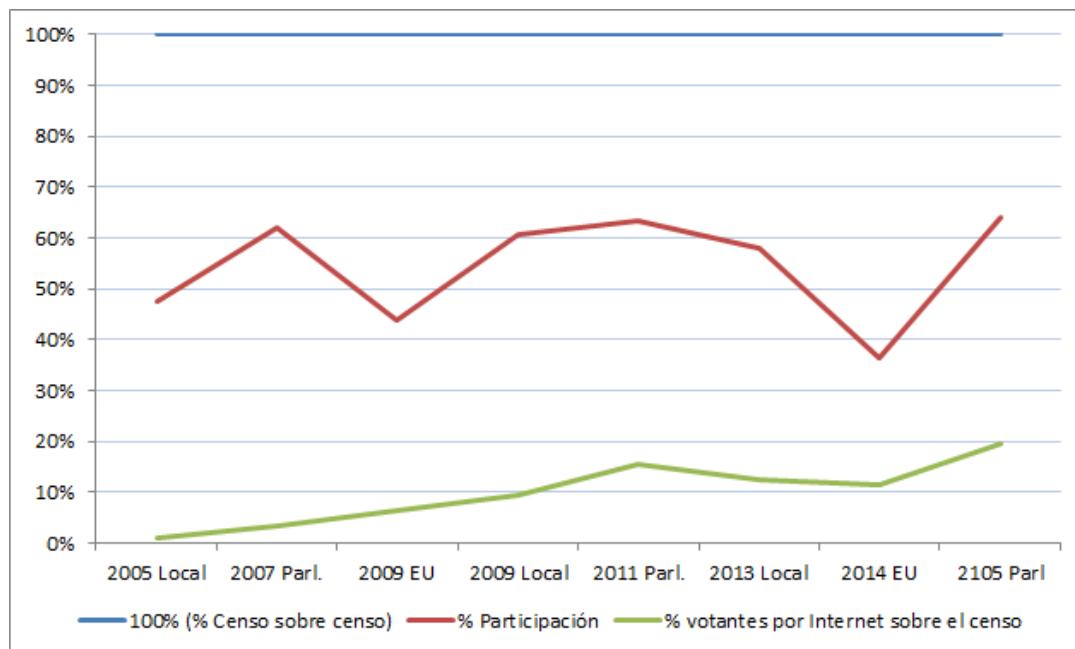


Figura 2.4: Histórico: Porcentaje de voto presencial comparado con el i-Voting en Estonia.

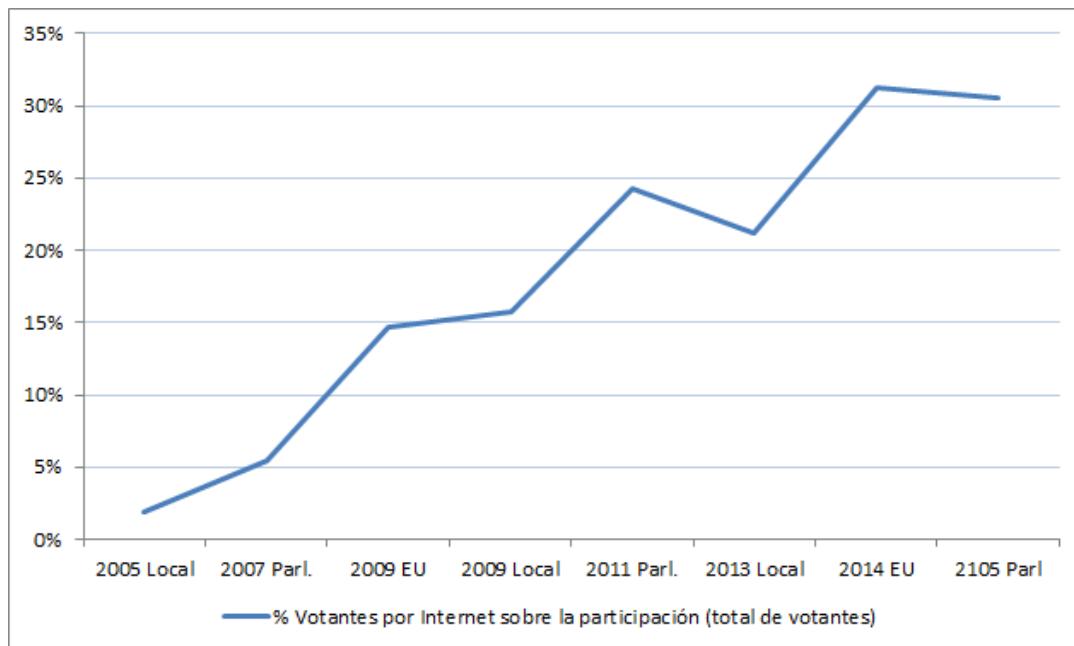


Figura 2.5: Histórico: Porcentaje de voto por Internet frente al total de votos en Estonia.

- a) **Voto tradicional** Los votantes pueden acudir a los colegios electorales e introducir su voto en la urna previa identificación del votante por parte de los miembros de la mesa.
- b) **Voto postal** Los votantes estonios tienen la posibilidad de acudir en unas fechas determinadas anteriores al día electoral a unas Estaciones de Votación, que funcionan de forma análoga a Correos en España, donde pueden entregar el voto en papel y una acreditación que le identifique. Esta Estación se encarga de hacer llegar el voto y la identificación a la mesa o Distrito Electoral donde el votante esté censado.
- c) **Voto por Internet** Durante un período de tiempo anterior al día electoral, los votantes tienen la posibilidad de entregar el voto por medio de Internet.

Aunque el votante haya emitido su voto de forma electrónica, la Ley Electoral estonia permite al mismo ejercer su voto de cualquiera de las otras dos formas invalidando su voto electrónico. Es decir, que si una vez votado por Internet, el votante decide votar por correo, éste voto anulará el emitido por Internet. Lo mismo pasaría si decidiese votar presencialmente el día electoral, que su voto emitido por Internet quedaría anulado y fuera del escrutinio. Este hecho es una medida de la Autoridad Electoral para proteger a los votantes frente a la coerción, proveyendo de un mecanismo por el cual un votante que haya elegido una formación determinada por presiones de terceros podría libremente cambiar la dirección de su voto una vez emitido el primero.

Son requisitos fundamentales de este sistema de voto electrónico remoto la seguridad,

confiabilidad y la precisión, así como proveer de mecanismos eficaces contra la coerción. Otra necesidad importante del sistema es su acceso, que debe ser prácticamente universal, lo cual implica que ha de ser accesible y sencillo de entender para los usuarios, además de que debe funcionar en la mayoría de las plataformas tecnológicas.

Hay una serie de puntos, recogidos en [42], que consiguen que el sistema satisfaga tales requisitos:

1. Uso de ID-cards o Mobile ID para la identificación de los votantes.
2. Un votante puede emitir cualquier número de votos durante el periodo habilitado para la votación electrónica. El último voto enviado será el único que cuente en el escrutinio. No obstante, si el votante se encuentra bajo algún tipo de coerción, siempre podrá volver a votar más adelante (cuando no ejerzan presión sobre su decisión) y este último será el que cuente. Así se intenta minimizar el riesgo de la coacción.
3. Prioridad del voto tradicional. Si el votante ejerce su derecho al voto de forma presencial, cualquier voto que hubiese emitido de forma electrónica será cancelado y no se contará en el escrutinio.
4. Todos los servidores en el sistema de voto son seguros y siempre estarán bajo monitorización durante el periodo de la votación.
5. El servidor de almacenamiento de voto está detrás de un firewall. Nadie puede acceder a este servidor desde Internet.
6. El servidor de conteo de votos está offline, sin conexión a Internet y asegurado por medio de clave privada compartida.
7. Todas las comunicaciones a través de Internet usan cifrado SSL.
8. El cifrado y la firma digital usan un mecanismo de cifrado RSA.

En las conclusiones de la tesis [49] se destaca de este proceso:

- Interesante por ser una elección a nivel nacional y vinculante.
- Número de votantes en tendencia creciente a nivel nacional.
- Debilidades:
  - No se hace uso de mecanismos seguros que garanticen la protección de la privacidad del voto.
  - El voto no está protegido por primitivas de firma ciega, anonimizadores (el voto se almacena entre 4 y 10 días junto a la identificación del votante) ni mecanismos equivalentes, sino que se traslada a este sistema de voto por Internet las debilidades ya existentes en el voto tradicional.

La importancia que tiene el sistema de voto utilizado en Estonia radica en el hecho de que provee un mecanismo de voto por Internet a un potencial de votantes consistente en el 100 % de la población con derecho de voto de un estado democrático. Hasta el momento de su implantación, esto no ocurría. Se daban casos en los que se proporcionaba un sistema electrónico remoto a diversos espectros de la población, como podían ser los residentes en el extranjero, los militares en misiones activas o los focos de posibles proyectos pilotos.

Es destacable que, con el objetivo de alcanzar a la totalidad de la población, incluso, el propio estado estonio aumentara el desarrollo de infraestructura tecnológica en el país para intentar reducir la brecha digital de sus habitantes, tratando de proveer el acceso a Internet a la mayor parte del país. Igualmente importante fueron los esfuerzos por la certificación digital, teniendo como punto esencial la implantación del documento nacional de identidad electrónico para la totalidad de la población.

La mayoría de los estados no pueden implantar unas elecciones como las llevadas a cabo en Estonia por diversos motivos, véase el miedo a la falta de transparencia, fraude, logística o, en muchos casos, ilegalidad con respecto a las leyes electorales actuales.

En diversos países se ha logrado implementar sistemas reales de votación deslocalizada por Internet en varios de sus territorios, alternativamente y al mismo nivel que el sistema tradicional presencial. Un ejemplo veremos que es Suiza con los proyectos de varios de sus cantones. No es comparable a la experiencia estonia, pues cada cantón se rige de forma diferente y son diferentes empresas las que realizan los desarrollos del sistema independientemente del resto, además de que no todos los cantones han implementado estos sistemas remotos.

### **2.2.2. Noruega**

La implantación del voto electrónico en Noruega empieza en 1993 con una experiencia piloto utilizando máquinas de lectura óptica en la capital, Oslo.

Ya en 2011, en las elecciones municipales se realiza una primera prueba de voto por Internet en la participaron diez municipios (de un total de 429). Los votantes tenían la posibilidad de votar por Internet durante un período de voto anticipado. No obstante, podían votar mediante el tradicional voto en papel el día de las elecciones, prevaleciendo este voto físico sobre el emitido previamente por Internet.

Dos años más tarde, en las elecciones parlamentarias de 2013 se realiza una segunda prueba de voto por Internet en 12 municipios. Más de 250.000 votantes tenían la posibilidad de utilizar este canal de votación.

En 2014, no obstante, el Gobierno noruego decide dar por concluidas las pruebas de voto por Internet debido a la controversia política existente y a que los ensayos realizados

no dieron muestras de impulsar la participación entre los ciudadanos.

	SISTEMA DE VOTACIÓN	
	ESTONIA	NORUEGA
Identificación digital de votantes	•••	••
Protección frente a la suplantación de votantes	••	••
Usabilidad de la interfaz de votante	•	•
Seguridad criptográfica de la información intercambiada	•••	•••
Protección frente a ruptura del secreto del voto por una sola entidad	•••	••
Protección frente a ruptura del secreto del voto por colusión entre entidades	•	•
Protección frente a contabilización indebida de votos	••	••
Protección frente a denegación arbitraria de derecho a voto	•••	•••

<sup>a</sup> Fuente: Ponencia *Posibilidades del voto telemático en la democracia digital*, por Justo Carracedo<sup>b</sup>

<sup>b</sup> <http://www.criptored.upm.es/descarga/ConferenciaJustoCarracedoTASSI2014.pdf>

Tabla 2.3: *Comparativa de la evaluación de los sistemas de Votos por Internet de Estonia y Noruega conforme al criterio de Seguridad de la Información*<sup>a</sup>

	SISTEMA DE VOTACIÓN	
	ESTONIA	NORUEGA
Identificación robusta de gestores del sistema	•••	•••
Verificación individual de voto	•	•
Verificabilidad de resultados	•	••
Protección frente a la coacción	•	•
Protección del sistema frente a falsas acusaciones	••	••
Capacidad de supervisión de los interventores	••	••
Uso de software público	•	•••
Auditabilidad del sistema	••	••

<sup>a</sup> Fuente: Ponencia *Posibilidades del voto telemático en la democracia digital*, por Justo Carracedo<sup>b</sup>

<sup>b</sup> <http://www.criptored.upm.es/descarga/ConferenciaJustoCarracedoTASSI2014.pdf>

Tabla 2.4: *Comparativa de la evaluación de los sistemas de Votos por Internet de Estonia y Noruega conforme a los criterios de verificación, auditoría y procedimiento*<sup>a</sup>

### 2.2.3. Suiza

El estado suizo se divide administrativamente en cantones. Estos cantones son los responsables de la celebración de procesos electorales en sus territorios. Con esto, varios de ellos, impulsados por el Estado, han dedicado mucho esfuerzo al estudio y desarrollo del voto por Internet para poder implementarlo de forma general.

Suiza es un país especial, pues tiene una gran costumbre en la realización de referendos. Entre 1789 y 2012 se contabilizaron 577 comicios en el país, lo que da una idea de la necesidad de acometer soluciones para combatir el "cansancio" democrático que puede suponer votar unas seis veces al año.

Con el objetivo de la implantación del voto electrónico, el estado suizo marcó tres fases de actuación como línea a seguir para la resolución de estudios y pruebas pilotos previas a una futura utilización de este sistema con garantías de viabilidad y seguridad. [7]

En una primera fase, de 2000 a 2002, se realizaron una serie de estudios e investigaciones que derivaron en la creación de programas piloto de voto electrónico.

De 2002 a 2006, tres cantones - Neuchatel, Ginebra y Zurich, comenzaron a realizar pruebas piloto que mostraron que era posible implantar el voto electrónico remoto en Suiza, lo cual acentuó el apoyo del Gobierno en el proyecto.

A partir de 2006, las pruebas piloto se expandieron a otros cantones, utilizando estos los sistemas desarrollados por el cantón de Zurich o el de Ginebra.

En 2010 ya eran 12 los cantones que realizaron pruebas pilotos en los comicios del 28 de noviembre. El número de votantes que podían votar de forma electrónica ascendía a 193.236 personas aunque, sin embargo, tan sólo 28.192, no llegó al 15 %, lo hicieron de esta forma.

En 2017, Swiss Post, ente organizador de comicios electorales en Suiza publicó una demo<sup>4</sup> de votación por Internet para que los ciudadanos puedan tener contacto con este tipo de sistemas y aclarar así dudas de funcionamiento y tecnología.

La tecnología utilizada para este sistema se basa en un protocolo E2E verificable. Se puede utilizar en cualquier navegador web. Los votos se cifran en el servidor y se almacenan de forma segura desacoplándolos de la información del votante.

#### 2.2.4. Universidades

A nivel de Juntas de Gobierno o de Elecciones de Rector a nivel universitario, merece la pena introducir que existen experiencias de voto por Internet en varias de ellas, destacando las de la UNED o las de la UPV/EHU.

La UNED fue pionera en el voto electrónico para elegir su claustro dentro del mundo universitario español. En el año 2010, se alió con la multinacional española Indra para adaptar su plataforma Net-vote<sup>5</sup> al proceso electoral y generar un caso de éxito.



Para llevar a cabo estos comicios fue necesario adaptar su Reglamento Electoral General.

El proceso consistía en cuatro fases:

---

<sup>4</sup> [https://www.evoting.ch/index\\_en.php](https://www.evoting.ch/index_en.php)

<sup>5</sup> [http://www.indracompany.com/sites/default/files/indra\\_pe\\_netvote\\_baja.pdf](http://www.indracompany.com/sites/default/files/indra_pe_netvote_baja.pdf)

1. **Preparación.** Preparar las necesidades a cubrir del resto de fases.
2. **Fase Preelectoral.** En esta fase, se precarga el sistema con el censo de la elección, los parámetros que la definen y se generan las claves de cifrado de los votos.
3. **Votación.** Fase en la que el votante introduce su voto en el sistema y éste lo guarda cifrado.
4. **Recuento.** Se desencriptan los votos para ser contados. A continuación, se publican los resultados.

Es interesante destacar el apartado de la Identidad Digital de esta primera elección. Se permitieron tres tokens digitales para poder identificar al votante contra el sistema:

- Certificado electrónico incluido en el **DNIe**
- Certificado **CERES-FNMT**, incluido en la tarjeta universitaria inteligente de la UNED.
- **Clave concertada** (usuario y contraseña) de uso exclusivo en este proceso electoral.

Ya en esta elección se apostaba por la identificación con los certificados del DNIe 2.0, el que posee chips de contacto. Dentro de las conclusiones de esta experiencia, se observa que el tipo de incidencia más común en el proceso fue la de *Instalación de la máquina virtual de JAVA*, necesaria para poder ejecutar el applet de comunicación entre el navegador y los drivers de lectura de los certificados del documento. Se empieza a vislumbrar que la Web no está realmente preparada para este mecanismo de identificación y resulta ser un cuello de botella muy importante para la identificación remota en Internet.

Para consultar de forma rápida datos sobre esta primera elección, un documento muy resumido y recomendables es [43].

A partir de 2013, los diferentes comicios de esta universidad fueron adjudicados a la empresa española Scytl. En los de 2013, correspondientes a la elección de Rector, pusieron a disposición del proceso su plataforma de voto por Internet Pnyx. Para conocer más acerca de la adaptación de esta plataforma a estas elecciones y las diferencias con la anterior, adjunto en la bibliografía un documento divulgativo de fácil lectura. [51]



Este mismo sistema Pnyx es el que lleva varios años implementando Scytl como plataforma de voto para las diferentes elecciones de la Universidad del País Vasco, además de otras Universidades y organizaciones públicas y privadas.

## 2.2.5. Escuela Politécnica Superior - Universidad San Pablo CEU

En la Escuela Politécnica Superior de la Universidad San Pablo-CEU ya se realizó una elección por medio de voto electrónico.



Sucedió en 2005, cuando en una colaboración entre la Universidad y la multinacional española Indra se celebró la primera elección de delegados de clase a través de voto electrónico con motivo del Día de Internet, celebrado el 25 de octubre del mismo año.

En esta experiencia, más de 600 alumnos de los últimos cursos de la Escuela Politécnica eligieron a sus delegados de clase a través de este sistema.

En la fecha de la elección, cada alumno emitió su voto a través de un nombre de usuario y una clave personal. Por motivos divulgativos, los organizadores de la elección determinaron que una parte del alumnado censado realizara la votación desde un aula de votación concreta, perteneciente al centro y adecuada para ello; mientras que el resto del alumnado debía elegir sus representantes desde algún equipo personal fuera del dominio de la Universidad.

Para que estas elecciones a través de Internet pudiesen llevarse a cabo la Universidad San Pablo-CEU tuvo que adaptar su normativa de régimen interno, pues la que tenía originalmente establecía únicamente la posibilidad de un sistema de voto presencial.

## 2.3. Proyectos varios de voto por Internet

En esta sección introducimos algunos proyectos de voto por Internet que se han estudiado para la elaboración de este PFC y que han resultado ser interesantes y con propuestas a tener en cuenta para llevar a cabo proyectos de índole electoral.

No obstante, una vez que se decidió que el sistema debe cumplir con la característica de ser un sistema de votación auditabile punto a punto (E2E Auditabile Voting System), estos proyectos fueron descartados como códigos base de la adaptación al sistema a implementar para la EPS.

### 2.3.1. Votescript

Votescript es un proyecto de español, gestado en la Universidad Politécnica de Madrid, de voto electrónico remoto basado en locales de votación. Pese a que no responde completamente al objetivo de este PFC, pues requiere de un entorno controlado como los locales de votación, su desarrollo telemático sí que se basa en una comunicación de los votos a través

de Internet. Se podría corresponder con el escenario en el que se deba dar la posibilidad de voto presencial de forma complementaria al voto remoto (es uno de los requisitos que se definirán en el capítulo 5.1.3).

El esquema de votación telemática Votescript tiene su origen en el proyecto de investigación *Votación Electrónica Segura basada en criptografía avanzada* [10], denominación de la cual adquiere el nombre, Votescript. Este proyecto es una colaboración entre el grupo de la Universidad Politécnica y la Fábrica Nacional de Moneda y Timbre - Real Casa de la Moneda (una de las principales entidades emisoras de certificados digitales de España).

A partir de este proyecto de investigación, los autores publican diversos artículos sobre el funcionamiento y alcance de los resultados obtenidos. En este apartado, nos basamos en la versión más actual del proyecto, desarrollado en su tesis doctoral [49] por una de las autoras del original, la Dra. Emilia Pérez Belleboni.

En esta tesis, además de analizar el estado del voto telemático, teniendo en consideración, esquemas, problemas y riesgos, realiza un estudio de varias implementaciones reales a nivel nacional, como por ejemplo un extenso análisis del procedimiento electoral electrónico de Estonia (2.2.1). No obstante, a partir de estos análisis, desarrolla el esquema que proponen, con base en el Votescript original, evolucionándolo para solucionar las debilidades del resto de sistemas y para su aplicación en la elección de representantes para el Parlamento Europeo.

En contraposición a los sistemas que estudia en la tesis, el sistema Votescript centra sus esfuerzos en la superación de debilidades identificadas en los anteriores, en especial en la fase de identificación del votante. En elecciones como las del Parlamento Europeo, una entidad supranacional, es muy importante que la identificación de los votantes se pueda realizar electrónicamente de una forma altamente confiable, pues deben ser válidas no sólo en el país del propio votante, sino en el resto de países europeos.

El esquema que propone Votescript define la necesidad de unos puntos específicos de votación, centros donde han de acudir los votantes a votar telemáticamente. En estos centros se implantarían los medios y equipamientos tecnológicos para que el votante emita su voto en un entorno controlado.

Según su documentación, las bases del sistema se pueden adecuar sin problemas a un *sistema abierto* (voto por Internet), pero el precio que implica la comodidad de los votantes de poder votar sin necesidad de trasladarse a locales oficiales incurre en un incremento de los riesgos de coacción.

### 2.3.2. SEVI

El Sistema Electrónico de Votación por Internet (SEVI) [33] es una propuesta de sistema software de voto electrónico para reemplazar el canal que constituye el correo postal certificado en el proceso electoral de México.

La idea del sistema es que los ciudadanos con derecho a voto que no puedan hacer uso del mismo el día del proceso tengan un canal de votación disponible a través de Internet. Este canal sustituiría al proporcionado por el correo postal, por lo que debe asegurar, como mínimo, los mismos servicios que ya proporcionaba éste en procesos electorales anteriores.

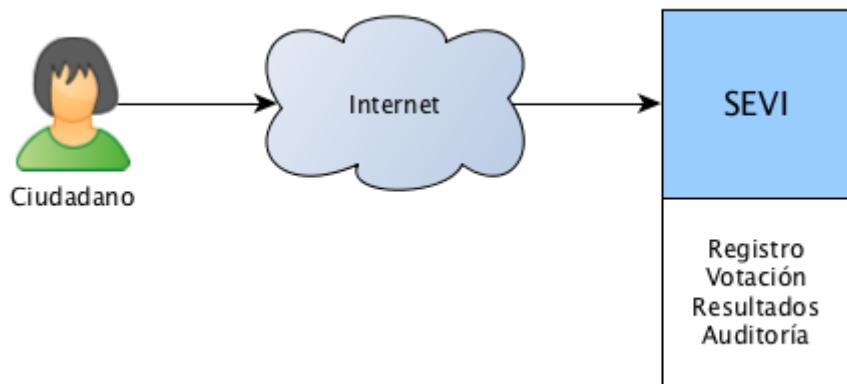


Figura 2.6: Primera aproximación de funcionalidad de SEVI

El esquema de votación utilizado en SEVI divide el proceso electoral en cuatro fases:

- Registro
- Votación
- Resultados
- Auditoría

Al igual que en la tesis del proyecto SELES [29], el protocolo de seguridad en el que se basa SEVI es una variante del de Lin-Hwang-Chang, el cual se compone de tres fases que cubren la seguridad del esquema en los módulos de votación y generación de resultados. También con este protocolo, el acuse de recibo dado a los votantes sirve para cumplir con los requisitos de auditoría. Otra de las necesidades resueltas es la de la democracia del proceso, pues el hecho de que se puedan identificar los votantes no honestos al emitir el voto aumenta la confianza en el proceso.

No obstante, el protocolo de seguridad que respalda la mayor parte de la interacción con el sistema no cubre la fase de registro. Precisamente esta es la fase con la que comienza

el sistema y donde se basa parte de la suposición de honestidad esperada por el protocolo para el resto del proceso. Para resolver el problema, en SEVI optan por establecer un canal seguro entre la máquina cliente y la máquina servidor a través de un protocolo de transferencia segura SSL.

## 2.4. Proyectos de Sistemas de Votación Auditables Punto a Punto

Una propiedad bastante útil para el desarrollo de esquemas para votación por Internet es la llamada *verificabilidad punto a punto* (o punta a cabo según otros autores hispanohablantes) (en la literatura inglesa: end-to-end verifiability; E2E-verifiability). Con ella se puede solucionar el problema de la necesidad de confiar en el proceso que recoge, almacena y cuenta los votos.

Se considera a Josh Benaloh<sup>6</sup> como el precursor del concepto de verificabilidad E2E. Según su propia definición del término, se considera que los requisitos de un sistema completamente verificable E2E son [4, 22, 52]:

1. **Verificabilidad individual:** Los votantes pueden comprobar que sus votos se han registrado con la opción que han elegido.
2. **Verificabilidad universal:** Cualquiera puede comprobar que todos los votos han sido escrutados con precisión.

Con estas premisas, en [22] se recoge un resumen sobre sistemas E2E del cual se puede extraer que los sistemas utilizados para las votaciones en Estonia (2.2.1) y en Noruega (2.2.2) son casi completamente E2E verificables.

Para cumplir el requisito de la verificabilidad individual, el sistema estonio (2.2.1) se apoya en una app para smartphones que gestiona la verificación del voto, mientras que en la solución noruega (2.2.2) apuestan por un sistema de retorno de código por SMS. A través de ambos protocolos, se facilita al votante una herramienta con la cual asegurarse de que su voto ha sido correctamente incluido en la votación.

Los problemas para satisfacer los requisitos de Benaloh aparecen cuando se estudian las herramientas para cumplir con la verificabilidad universal.

El método tradicional para garantizar este requisito es utilizar las pruebas de conocimiento zero (2.5.1.3). Con estas pruebas se trata de convencer a un verificador de que el proceso se ha llevado a cabo correctamente con una alta probabilidad, sin necesidad de que el verificador tenga conocimiento del contenido de los votos incluidos en la elección.

---

<sup>6</sup><http://research.microsoft.com/en-us/um/people/benaloh/>

En los sistemas implementados en Estonia y Noruega, aunque parecen tener herramientas que incluyen estas pruebas de conocimiento nulo en todas las etapas del proceso, según algunos informes postelectorales [11], esto no ocurría en la totalidad del mismo.

Según estos mismos informes [11], un ejemplo de sistema electoral por Internet E2E verificable es Helios Voting (2.4.3), del que dicen que es un reconocido “*estándar en verificabilidad de voto por Internet*”. Las bonanzas de este sistema es que proporciona al proceso todas las garantías que aporta la verificabilidad E2E, como la capacidad de observar pruebas de conocimiento cero no interactivas que verifican que cada voto fue incluido correctamente y que el escrutinio completo fue computado con precisión. No obstante, hay que remarcar que, como su propio desarrollador indica, Helios es un sistema pensado en elecciones con bajo riesgo de coacción, lo cual no es válido para elecciones nacionales como el modelo estonio o noruego.

Las garantías que ofrecen los esquemas de verificabilidad E2E eliminan la necesidad de los votantes de confiar tanto en los propios clientes (dispositivos, navegadores, sistemas operativos) que utilizan para votar, como los servidores y los trabajadores oficiales que administran los sistemas asociados a la recepción, descifrado y conteo de votos, que son los pilares críticos en los que se basa un sistema de voto por Internet. Esto es así porque los esquemas basados en verificabilidad E2E aseguran la inviolabilidad del voto emitido hasta ser contado, eliminando la amenaza de interceptación del mismo en cliente, canal de comunicación o servidor de escrutinio.

Con esta perspectiva de relajada confianza en los dispositivos clientes, se torna como un buen protocolo para el voto por Internet sin locales habilitados. Se debe a que en este marco altamente distribuido, la votación se realiza desde entornos no controlados por el sistema, con lo que hay un alto riesgo de amenazas, pero la verificabilidad E2E se encarga de demostrar la invariabilidad del voto emitido frente al escrutinio, por lo que minimiza varios de los riesgos inherentes a los clientes no controlados. (Obvia comentar que sigue habiendo bastantes amenazas que este medio no va a tratar y tendrán que ser enfrentadas con otras herramientas.)

En esta sección, vamos a recoger e introducir brevemente, tres ejemplos de proyectos desarrollados con el objetivo de cumplir con las premisas de la verificabilidad punto a punto (E2E-verifiability).

De los tres sistemas, ADDER (2.4.1) es un desarrollo académico, Ágora Voting(2.4.2) es un sistema desarrollado en España con un amplio historial de procesos electorales llevados a cabo, al igual que Helios Voting(2.4.3), considerado un estándar de facto de votación basada en verificabilidad E2E.

### 2.4.1. ADDER

ADDER [32] se define como un sistema de voto electrónico basado en Internet, libre y de código abierto. Desarrollado por la Universidad de Connecticut (EEUU) en 2006, supone una plataforma de eVoting completamente funcional con una serie de características de seguridad como robustez, privacidad del voto, auditabilidad y verificabilidad.

Los desarrolladores de ADDER dividen el voto por Internet en 3 escenarios:

**Remoto** En el escenario del voto por Internet remoto, un actor diferente a la autoridad electoral, ya sea el votante o un tercero, es el que tiene el control sobre el cliente de voto y el entorno operativo.

**Kiosko** En el escenario del voto por Internet en modo kiosko, el cliente de voto puede ser instalado por las autoridades de la elección, pero entorno de la votación está fuera de su control.

**Cabina de voto** En este escenario, las autoridades electorales tienen control tanto del cliente de voto como del entorno en el que se lleva a cabo.

ADDER fue diseñado para el primero de los escenarios, el voto telemático remoto, pero dependiendo de los requisitos de seguridad, es adaptable a los otros dos. Además, es un sistema capaz de llevar a cabo elecciones tanto a pequeña como a gran escala.

Procedimiento de la elección:

- El primer paso en el procedimiento del proceso electoral con este sistema es que el administrador ha de alimentar al sistema con la lista de usuarios (votantes y autoridades) y candidatos.
- Las autoridades acceden al sistema para participar en la generación de claves criptográficas. Se genera una clave pública para el sistema y privadas para cada una de las autoridades.
- En el período de votación, cada votante ingresa en el sistema y descarga la clave pública de éste. Con esta clave cifra su voto. El voto cifrado se guarda en un área reservada al votante.
- Al finalizar el período de votación, el servidor cuenta los votos y publica el resultado cifrado.
- Las autoridades proporcionan información basada en el resultado cifrado y sus claves privadas para descifrar el resultado. El sistema combina estas claves individuales y los descifrados parciales del resultado para componer el resultado electoral, que se publica.

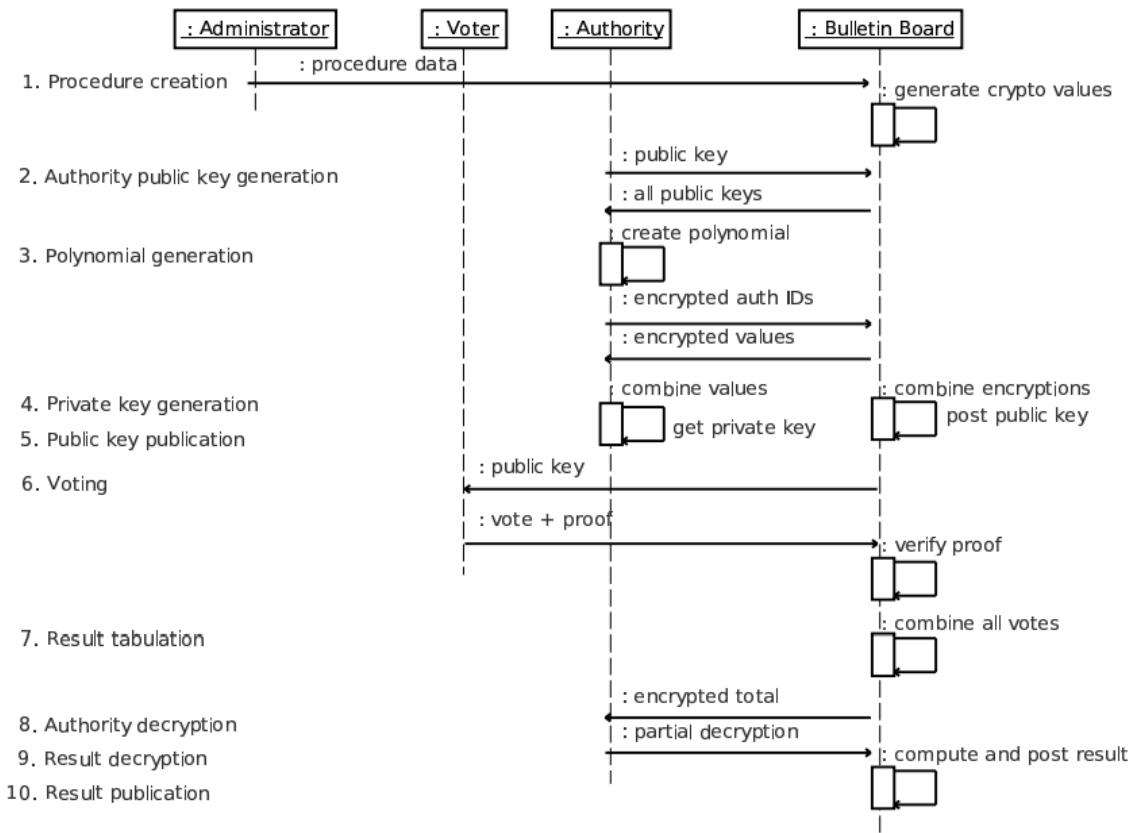


Figura 2.7: Diagrama de secuencia del procedimiento para una elección con ADDER [32].

- El sistema, por tanto, se implementa alrededor de un tablón de anuncios, un servidor de autenticación (*gatekeeper*) y un cliente. Haciendo uso de secreto compartido a la hora de repartir trozos de la clave privada del sistema y del resultado cifrado entre las diferentes autoridades con responsabilidad en el sistema.

Los objetivos que persigue el sistema ADDER son:

**Transparencia** Toda la información del tablón de anuncios es pública y puede ser consultada por cualquier observador. Aquí se incluyen votos cifrados, claves públicas y escrutinios.

**Verificabilidad universal** Cualquier resultado electoral obtenido por el sistema debería ser verificado por cualquier observador. A través de los logs y trazas del sistema se puede realizar una auditoría de cualquiera de los procesos.

**Privacidad** Todos los votantes pueden confiar en que sus votos se mantienen secretos. Sólo el recuento es accesible al público.

**Confianza distribuida** Cada procedimiento del proceso electoral está supervisado por varias autoridades. El recuento no puede llevarse a cabo sin la cooperación de un determinado número de autoridades.

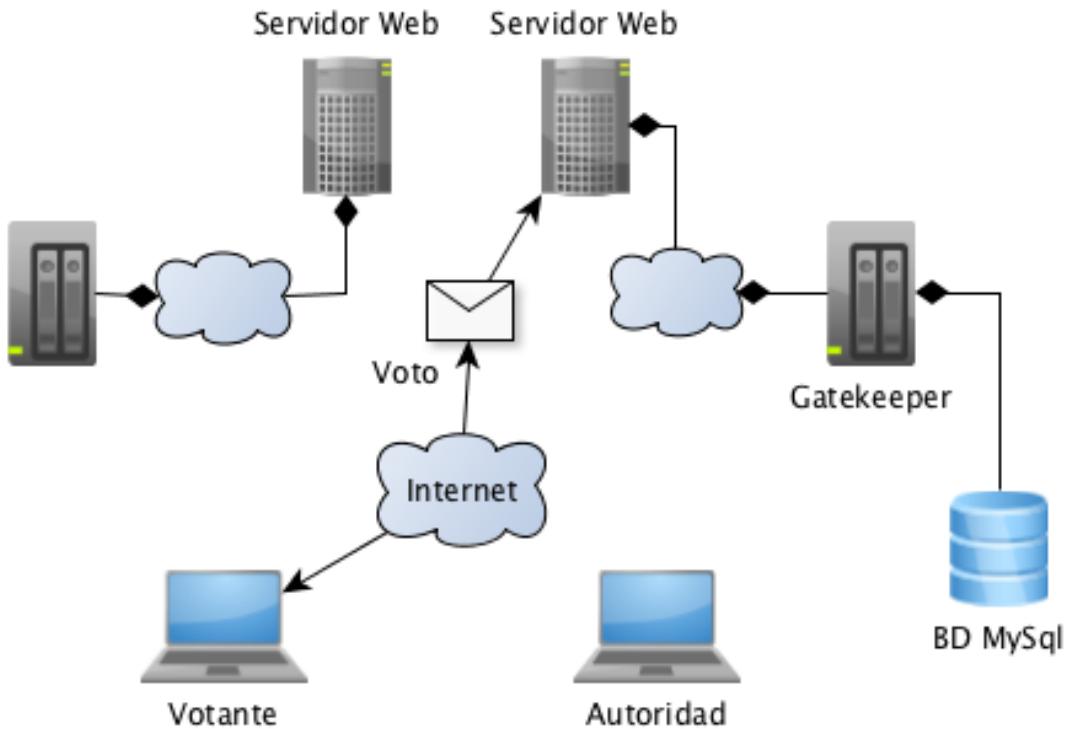


Figura 2.8: Arquitectura de ADDER.

Soluciones propuestas por ADDER:

1. *Autenticación de usuarios.* Para la autenticación de usuarios, ADDER emplea un sistema análogo a Kerberos que denominan *gatekeeper*.
2. *Privacidad del voto.* Para contrarrestar el conflicto entre privacidad del voto y verificabilidad universal y la necesidad de acceso al contenido del voto para asegurar el correcto conteo, el sistema utiliza técnicas de cifrado homomórfico.
3. *Verificabilidad universal.* Para cumplir con este requisito, el sistema se apoya en un tablón de anuncios, en el que se publica información relevante al proceso. Junto al sistema, se han implementado una serie de herramientas libres y de código abierto que permiten hacer uso de los datos publicados en el tablón y realizar una serie de tareas:
  - a) *Recuento de los votos encriptados.* Gracias a las propiedades del cifrado homomórfico, no necesita usar claves privadas para contar los votos, pues no es necesario que los descifre para realizar el recuento. El programa puede repetir el proceso ejecutado en el servidor.
  - b) *Verificación de todas las pruebas.* Cada votante puede comprobar las pruebas de validación de su voto.

- c) *Descifrado del recuento final.* Una vez que todas las autoridades han terminado sus descifrados parciales, la suite de verificación recalcula los coeficientes de Lagrange y desencripta la suma final.
  - d) *Verificación del hash.*
4. Verificabilidad del votante.

#### 2.4.2. Ágora Ciudadana - Ágora Voting



Ágora Voting se inició como un proyecto de software libre cuyo objetivo era el de proporcionar una plataforma de democracia líquida, incluyendo un protocolo de voto criptográficamente seguro.

Este sistema se compone de varios elementos:

##### Registro

Plataforma web que sirve para la verificación de votantes.

- API web y AngularJS *single web application*.
- Base de datos con acceso al padrón.
- Servidor con certificado TLS.
- Fail2ban<sup>7</sup> y Cloudfare<sup>8</sup> para protección contra DDoS.
- Redundancia de Hardware.

##### Cabina de voto

Plataforma web a través de la cual los votantes emiten sus votos.

- Servidor con validación TLS.
- Cabina de voto Javascript para votar o auditar.
- Librerías de cifrado Javascript (Helios, SJCL).
- Base de datos PostgreSQL replicada para la urna electrónica.
- Fail2ban para protección ante DOS y ataques de fuerza bruta.
- Cloudfare para protección ante ataques DOS.
- Autenticación de cliente para votantes registrados y validados.

##### Servicios de la Autoridad de la Elección

Servicios web http que sirven para generar claves públicas, mezclar, descifrar y totalizar.

- Cola HTTP para orquestación asíncrona.

---

<sup>7</sup> <https://www.fail2ban.org>

<sup>8</sup> <http://www.cloudflare.com>

- Validación TLS en cliente y servidor.
- Librería Verificatum<sup>9</sup> para mixnets.
- Librería OpenSTV<sup>10</sup> para totalización.

### Verificador de la Elección

Una aplicación Python/Java *stand alone* que verifica los datos publicados de la elección.

Las primitivas criptográficas que se emplean en este sistema son:

- Esquema de cifrado homomórfico ElGamal
- Esquema de cifrado de umbral Pedersen.
- Mixnet verificable universal con pruebas de conocimiento cero.
- Heurística Fiat-Shamir para transformar pruebas de conocimiento cero de verificabilidad en prueba verificables no interactivas.

Comentar que Ágora Voting dejó de ser un proyecto activo y pasó de software libre a ser una plataforma comercial de voto por Internet llamada nVote. Está implementada en Python y tiene bastantes funcionalidades comunes con Helios Voting.

En Ágora Voting se observa que hay una gran influencia de Helios Voting, especialmente en el inicio de ambos proyectos.

La diferencia fundamental en la evolución de sendos proyectos se encuentra en puntos como que Ágora continuó utilizando mixnets para desacoplar votos de los votantes mientras que Helios dejó de utilizar este protocolo, virando hacia el homomorfismo en la totalización.

Como se ha adelantado, Ágora Voting dejó de ser un proyecto de software libre. Todavía se puede encontrar en Github<sup>11</sup>, pero sus creadores han decidido desarrollar un producto comercial de nombre nVotes<sup>12</sup>.

Este proyecto ha sido utilizado en varios procesos electorales reales. Varios de ellos han sido votaciones interna de partidos políticos, como Podemos o sus partidos asociados.

La última gran elección llevada a cabo por nVotes ha sido la de Decide Madrid<sup>13</sup> en 2017.

### 2.4.3. Helios

---

<sup>9</sup><http://www.verificatum.com>

<sup>10</sup><http://www.opavote.com/openstv>

<sup>11</sup><https://github.com/agoravoting>

<sup>12</sup><https://nvotes.com>

<sup>13</sup><https://decide.madrid.es>



Helios Voting<sup>14</sup> es un sistema de voto por Internet cuyos desarrolladores consideran que es el primer sistema de voto con auditoría abierta (open-audit) basado en web. Actualmente es un proyecto en desarrollo, funcional y públicamente accesible. Cualquier organismo interesado puede descargar el código fuente, configurar un proceso electoral y llevar a término la elección, junto con que cualquier observador puede auditar todo el proceso.

El creador del proyecto es Ben Adida<sup>15</sup>, doctor en Ingeniería Informática en Criptografía y Seguridad de la Información por el MIT<sup>16</sup>. Para su doctorado, base de este proyecto, fue asesorado<sup>17</sup> por otro Doctor como Ron Rivest<sup>18</sup>, experto en Criptografía y Voto Electrónico.

Para ver el rigor académico del proyecto desde sus inicios y durante su desarrollo, Adida también ha sido asesorado por profesionales de reputada experiencia en la criptografía y el Voto Electrónico como son Lawrence Lessig<sup>19</sup>, de la Universidad de Stanford<sup>20</sup> Y Josh Benaloh<sup>21</sup>, doctor por el MIT y Yale<sup>22</sup>, considerado el precursor de los sistemas de verificabilidad punto a punto (2.4).

Este proyecto es apropiado para realizar procesos electorales para organismos que necesiten que estos sean confiables y con voto secreto, aunque eso sí, siempre que los comicios se celebren en un ambiente en el que la coacción del voto no sea una amenaza. Este detalle es importante, ya que muestra una de las *debilidades* de esta implementación con respecto a un proceso electoral de gran escala.

Helios trata de ser un sistema de voto por Internet simple comparado con otros protocolos criptográficos, centrándose en la *auditabilidad pública* como elemento diferencial. Con esta propiedad, cualquier organismo puede apoyarse en Helios para llevar a cabo la elección y, aunque resultase que Helios estuviera corrupto, la integridad de la elección puede ser verificada por los observadores.

El proyecto Helios ha sufrido una evolución desde sus primeras versiones a la actual, cambiando protocolos y esquemas criptográficos y superando vulnerabilidades que se le iban encontrando. Un buen documento para consultar esta evolución desde la primera versión de Helios hasta la última de 2016 es [52].

En este documento [52] se indica que las versiones de Helios 2.0 y 3.1.4, sufrían vulnerabilidades que permitían violar el secreto de voto y la verificabilidad. La versión de

---

<sup>14</sup><https://github.com/benadida/helios>

<sup>15</sup><http://ben.adida.net/>

<sup>16</sup><http://mit.edu/>

<sup>17</sup><https://vote.heliosvoting.org/about>

<sup>18</sup><http://people.csail.mit.edu/rivest/>

<sup>19</sup>[https://es.wikipedia.org/wiki/Lawrence\\_Lessig](https://es.wikipedia.org/wiki/Lawrence_Lessig)

<sup>20</sup><https://www.stanford.edu/>

<sup>21</sup><https://www.microsoft.com/en-us/research/people/benaloh/>

<sup>22</sup><https://www.yale.edu/>

2012 redujo esta vulnerabilidad, aunque siguió siendo insuficiente para asegurar la verificabilidad. La versión 2016 implementa un sistema de autenticación externa que permite satisfacer el peligro de violación de la verificabilidad.

Las primitivas criptográficas que encontramos en este sistema de votación por Internet son [35]:

- Cifrado homomórfico multiplicativo: ElGamal (2.5.1.5.1)
- Cifrado homomórfico aditivo: ElGamal Exponencial (2.5.1.5.2)
- Cifrado de umbral (+ Secreto Compartido) (2.5.1.2)

A lo largo de esta memoria se desgranarán las particularidades de este sistema (5.5, 6.3), sus ventajas e inconvenientes y el por qué ha sido el elegido como base del Sistema (5.3.1).

#### 2.4.4. Comparación Ágora - Helios

En la tesis de Codina Lligoña [16], el autor realiza un estudio de diversos sistemas de votación, entre ellos Ágora Voting (en su versión agora-ciudadana 2.0) y Helios 4.

Entre las múltiples características que valora, recoge información acerca de la interfaz, licencia, requisitos de voto electrónico, antecedentes y comunidad de soporte del proyecto.

Los resultados de la comparativa los resume en la tabla 2.5, en la cual los valores van desde el 0 (valoración negativa) al 2 (valoración positiva).

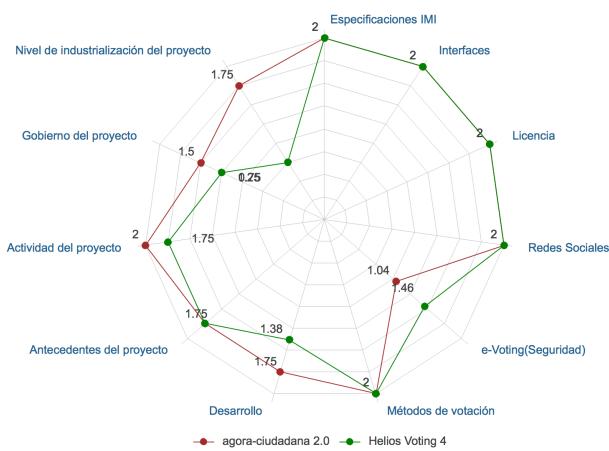


Figura 2.9: Comparativa Ágora - Helios en cuanto a características de Proyecto, por Codina Lligoña [16]

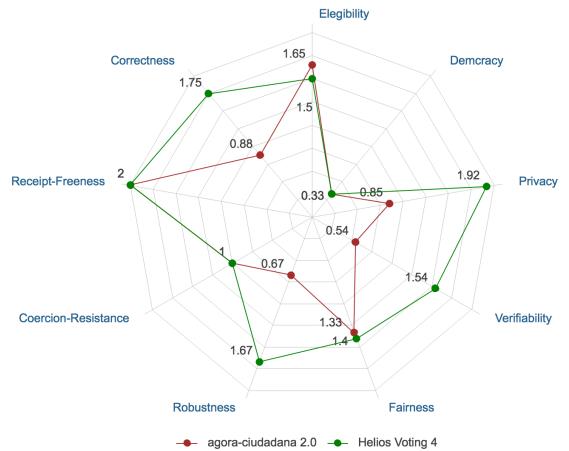


Figura 2.10: Comparativa Ágora - Helios en cuanto a requisitos del Voto por Internet de Proyecto, por Codina Lligoña [16]

En el estudio, el autor divide las valoraciones en dos, las correspondientes al nivel de desarrollo de los proyectos y a las características de seguridad.

Criterio	agora-ciudadana 2.0	Helios 4
Especificaciones IMI	2	2
Interfaces	2	2
Licencia	2	2
Redes Sociales	2	2
e-Voting (Seguridad)	1,04	1,46
Elegibility	1,75	1,5
Democracy	0,33	0,33
Privacy	0,85	1,92
Verifiability	0,54	1,54
Fairness	1,33	1,4
Robustness	0,67	1,67
Coercion-Resistance	1	1
Receipt-Freeness	2	2
Correctness	0,88	1,75
Métodos de votación	2	2
Desarrollo	1,75	1,38
Antecedentes del proyecto	1,75	1,75
Actividad del proyecto	2	1,75
Gobierno del proyecto	1,5	1,25
Nivel de industrialización del proyecto	1,75	0,75

Tabla 2.5: *Comparativa Ágora - Helios, por Codina Lligoña [16]*

En las primeras, da una mejor valoración al proyecto Ágora Voting. Sin embargo, en el segundo grupo, el que se correspondería con la mayoría de los requisitos inherentes al voto electrónico enunciadas por diferentes autores, la mejor nota se la lleva el proyecto Helios Voting, más correcto desde el punto de vista académico y de la seguridad en el voto remoto.

Según observa, Helios supera a Ágora Voting en estas características porque fue directamente diseñado teniéndolas en cuenta, por lo que desde el origen toda la información se almacena cifrada y existen procedimientos de auditoría, cumpliendo con los requisitos de Privacidad y Verificabilidad del voto.

Destaca que en ambos sistemas es posible emitir más de un voto por votante, aunque sólo uno de ellos será contado.

## 2.5. Mecanismos criptográficos

Los sistemas de votación electrónica son sistemas críticos en cuanto a que la información que tratan es muy sensible. Han de satisfacer requisitos de anonimato, secreto, verificabilidad, completitud, etc. los cuales requieren que la protección de votante, voto y sistema sea absoluta.

Para conseguir cumplir estos requerimientos básicos de seguridad, este tipo de sistemas han de apoyarse en las matemáticas, concretamente en la criptografía. Por ello, existen una serie de mecanismos criptográficos que aportan las herramientas, diseños y soluciones necesarias para tratar este problema y minimizarlo.

Una primera clasificación de mecanismos criptográficos distingue entre primitivas, esquemas y protocolos criptográficos:

- **Primitivas** Operaciones matemáticas, usadas como bloques constructores en la realización de esquemas. Su caracterización depende de los problemas matemáticos que sustentan su uso criptográfico. Ej: DES, RSA.
- **Esquemas** Combinación de primitivas y métodos adicionales para la realización de tareas criptográficas como la firma y el cifrado digital. Ej: DES-CBC-PKCS5Padding; RSA-OAEP-MGF1-SHA1
- **Protocolos** Secuencias de operaciones, a realizar por dos o más entidades, que contienen esquemas y primitivas con el propósito de dotar a una aplicación de características de seguridad. Ej: TLS

### 2.5.1. Primitivas criptográficas

Dentro de los retos tecnológicos que propone el voto electrónico, uno de los más importantes es la seguridad. Para poder implementar un sistema seguro que pueda soportar toda la infraestructura necesaria para poder poner en marcha un sistema de voto electrónico confiable hay que hacer uso de herramientas que sean capaces de asegurar las comunicaciones y el secreto de estas. Es en este escenario donde la criptografía es el núcleo de la solución.

Los requerimientos que se tratan de satisfacer con el uso de la criptografía son [40]:

- Privacidad del voto
- Autenticación del votante
- Integridad de la elección

En la bibliografía sobre criptografía se establece que existen tres aproximaciones generales para diseñar sistemas de voto electrónico basándose en primitivas criptográficas robustas:

**Basadas en mixnets** introducidas por David Chaum [13]

**Basadas en encriptado homomórfico** introducido por Josh Benaloh [3]

**Basadas en firma ciega** introducida por Fujioka et al. [27]

Antes de entrar en los diferentes esquemas de voto electrónico (2.5.2), introducimos una serie de primitivas criptográficas que se utilizan en ellos.

### 2.5.1.1. Firma ciega

Los protocolos criptográficos de firma ciega se dan lugar entre dos agentes, un usuario U y un firmante F de forma que F firma digitalmente una serie de datos comunicados por U sin conocer el contenido de estos.

El objetivo de este tipo de protocolos es proporcionar una serie de datos firmados cuyo contenido solamente sea conocido por el actor que envía, siendo completamente desconocidos para el actor que los firma.

Los protocolos de firma ciega se basan en dos componentes [9]:

1. Un protocolo de firma digital. Desarrollado por el actor F, quien es el prestador del servicio de firma. De tal forma,  $S(m)$  es la notación de la firma digital del mensaje  $m$ .
2. Dos funciones  $f$  y  $g$ , conocidas únicamente por el usuario U, de forma que

$$g(S(f(m))) = S(m)$$

La función  $f$  se denomina *función de ocultación o de opacidad*. La función  $g$  es la *función de recuperación*.

### 2.5.1.2. Secreto compartido

Los protocolos criptográficos de secreto compartido dividen un mensaje (secreto) determinado en diferentes fragmentos que se reparten entre los participantes de la comunicación. El reparto de información consiste en los siguientes preceptos:

1. El mensaje (secreto) original únicamente puede ser reconstruido por un cierto grupo de participantes autorizados.
2. Los participantes no autorizados no pueden obtener información sobre el contenido del mensaje original.

### 2.5.1.3. Pruebas de conocimiento nulo

Los protocolos basados en pruebas de conocimiento cero o nulo son protocolos criptográficos que se basan en la necesidad de una de las partes en poder demostrar a otra que

un enunciado es cierto sin revelar nada más que la veracidad del mismo.

Goldwasser, Micali y Rackoff propusieron tres propiedades que deben satisfacer todos los protocolos basados en pruebas de conocimiento cero. De forma coloquial, vamos a comentarlas aquí:

**Completeness** Si el emisor dice la verdad, en algún momento, convencerá al receptor de ello.\*

**Soundness** El emisor sólo puede convencer al receptor si realmente está diciendo la verdad.

**Zero-knowledgeness** El receptor no conoce nada acerca de la solución real del emisor.

\* A base de repetición de una solución, alguien con reputación es capaz de acabar convenciendo al receptor, con hechos, de que lo que demuestra es cierto.

Un sencillo ejemplo para entender el concepto de este tipo de protocolos lo encontramos en una publicación en su blog personal [21] por parte de Pablo Della Paolera, astrónomo de la Universidad Nacional de La Plata (Argentina). En este texto, el científico explica que se puede demostrar *algo* correctamente sin necesidad de demostrar *por qué*.

Basándonos en el ejemplo publicado, supongamos dos actores *Marta* y *Carmen*.

*Carmen* quiere demostrar que *Marta tiene (o no) la misma cantidad de monedas en su bolsillo izquierdo que en el derecho.*

Para ello, la forma más simple de hacerlo sería que *Carmen* le pidiese a *Marta* las monedas de cada bolsillo y las contase. Por tanto, para demostrar la afirmación de que *Marta realmente tiene (o no) las mismas monedas en cada bolsillo* basta con que las enseñe.

En este caso, el problema es que *Carmen* está violando la privacidad de *Marta* pues no debería ser necesario que *Carmen* conozca cuántas monedas tiene *Marta* y, mucho menos, tener que enseñarlas a la audiencia para demostrar la veracidad de sus investigaciones.

Esta violación de la privacidad se puede superar aplicando de forma simple una solución basada en prueba de conocimiento cero.

Supongamos que *Marta* tiene  $X$  monedas en el bolsillo derecho e  $Y$  monedas en el izquierdo. Para no conocer el número de monedas que posee *Marta*, *Carmen* le pide que piense en un número  $Z$  entero y mayor que  $X$  e  $Y$ . A continuación le pide que le diga la diferencia entre  $Z$  y  $X$  y entre  $Z$  e  $Y$ . Si ambas diferencias son iguales, *Marta* tiene el mismo número de monedas en ambos bolsillos, del mismo modo que si las diferencias no coinciden, se puede afirmar lo contrario, incluso sabiendo en qué bolsillo hay un mayor número de monedas. Y todo esto **sin que Carmen llegue a conocer en ningún momento cuántas monedas posee Marta**.

Matemáticamente, se trataría de un sistema de 2 ecuaciones con 3 incógnitas, por lo que no se puede resolver y no se pueden obtener los valores de  $X$  e  $Y$ :

$$Z - X = C$$

$$Z - Y = D$$

Pese a que no es resoluble, sabiendo los valores de  $C$  y  $D$  se puede demostrar en qué bolsillo tiene más monedas ( $C > D$  ó  $D > C$ ) o si se tienen las mismas ( $C = D$ ), sin necesidad de conocer los valores reales.

La importancia de este esquema criptográfico es tal que se ha implementado de forma satisfactoria en campos tan esenciales como la verificación de armas nucleares, permitiendo a los observadores internacionales medir la veracidad de una nación al cuantificar su fuerza nuclear sin necesidad de conocer la tecnología o cabezas u otros detalles clasificados que no quieren que sean sacados a la luz.

El ejemplo expuesto, pese a ser extremadamente simple, da una idea de lo que luego, matemáticamente se encarga de demostrar la criptografía.

Recomiendo estos otros ejemplos sobre pruebas de conocimiento cero. En este artículo<sup>23</sup>, se recomiendan tres ejemplos de distinto nivel, muy interesantes: Uno sobre un ganador de lotería<sup>24</sup>, otro para convencer a un niño de que no le estás haciendo trampa<sup>25</sup> y otro sobre

<sup>23</sup> <http://jorgegarciaherrero.com/auditar-algoritmo/>

<sup>24</sup> <http://elprofedefisica.naukas.com/2014/11/12/desafio-del-millon-de-james-randi-2/>

<sup>25</sup> <http://www.wisdom.weizmann.ac.il/~naor/PAPERS/waldo.pdf>

Google y sombreros<sup>26</sup>.

#### 2.5.1.4. Mixnets

Los sistemas de mixnets, o redes mixtas, se basan en una técnica enunciada por David Chaum en 1981 en [13] que permite establecer un canal anónimo entre un emisor y un receptor.

El funcionamiento básico consiste en que cada servidor, cuando recibe una serie de mensajes, los mezcla y entrega al siguiente servidor. Y así en todos los servidores hasta llegar al definitivo.

La relación entre una entrada y su correspondiente salida sólo es conocida por el servidor que realiza el mezclado. De este modo, al final del proceso de mezclado no se puede determinar qué entrada corresponde con qué salida. A no ser que haya una conspiración coordinada entre todos los servidores del proceso.

Esta idea de mezclado es aplicable al voto electrónico en busca de poder desacoplar el voto de su votante, protegiendo así la privacidad de éste.

Para una mixnet se utiliza un esquema de cifrado de clave pública  $(G, E, D)$  donde  $(pubK, secK) \leftarrow G()$  genera un par de claves pública y secreta.  $c \leftarrow E_{pubK}(m)$  cifra el mensaje  $v$  utilizando la clave pública  $pubK$ .  $D_{secK}(c)$  descifra el mensaje cifrado  $c$  utilizando la clave secreta  $secK$ .

El esquema ha de ser correcto y seguro. Para todo mensaje  $v$ ,  $D_{secK}(E_{pubK}(m)) = m$  y debe ser imposible conocer la información de  $m$  a partir de  $c$ .

La mixnet consiste en  $n$  servidores, cada uno de los cuales genera un par de claves públicas y privadas  $(pubK_1, secK_1), (pubK_2, secK_2), \dots, (pubK_n, secK_n)$ . Supongamos  $i$  usuarios que quieren enviar mensajes  $m_1, m_2, \dots, m_i$  a través de la red. Cada usuario prepara un mensaje cifrado de forma que  $c_i = E_{pubK_1}(E_{pubK_2}(\dots(E_{pubK_n}(m_i))\dots))$  y lo publica en un tablón público.

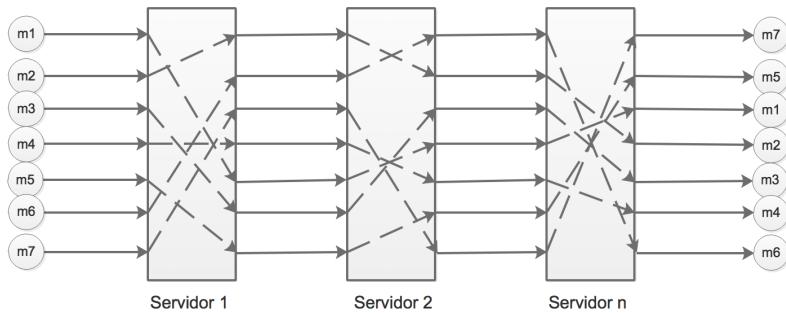
El primer servidor descifra la primera capa de cada mensaje cifrado mediante  $D_{secK_1}(c_i)$  para obtener  $c'_i = E_{pubK_2}(\dots(E_{pubK_1}(m_i))\dots)$ . Así, reordena  $c'_i$  y escribe el resultado en un tablón público. En este punto, sólo el primer servidor, que conoce la clave privada  $secK_1$ , es quien conoce qué  $c_i$  se corresponde con  $c'_i$ .

Cada uno de los servidores realiza el mismo proceso, de forma que al final el último servidor es el que obtiene los mensajes originales  $m_1, m_2, \dots, m_i$  y los publica.

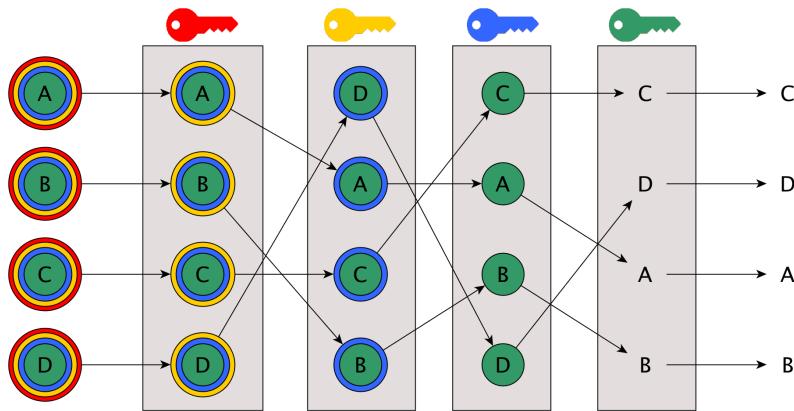
Este es el modelo enunciado por Chaum, en el que cada servidor descifra un mensaje cifrado. Existen otras

---

<sup>26</sup><https://blog.cryptographyengineering.com/2014/11/27/zero-knowledge-proofs-illustrated-primer/>

Figura 2.11: *Diagrama mezclado mixnet*

De todos modos, como se indica en [8], no basta con realizar un mezclado de votos para obtener un sistema de votación electrónica seguro, sino que se necesita un esquema con mecanismos para que los votantes certifiquen su identidad, así como para que puedan firmar digitalmente los votos de forma anónima, asegurando la integridad del voto y la privacidad del votante.

Figura 2.12: *Diagrama de mixnet con descifrado de mensajes*

### 2.5.1.5. Cifrado homomórfico

El cifrado homomórfico es un tipo de cifrado basado en algoritmos criptográficos que poseen la propiedad de que si se aplica una función ( $f$ ) a un mensaje cifrado, el resultado de descifrarlo es igual a la aplicación de otra función ( $g$ ) al mensaje sin cifrar.

Definamos  $D$  como la función de descifrado y  $C$  como la función de cifrado.

$$D(C(m, k), k) = m$$

$$D(f(C(m, k)), k) = g(m)$$

Hay algoritmos en los que las funciones que se aplican ( $f$  y  $g$ ) son idénticas, de forma que ( $f = g$ ), lo cual resulta muy interesante ya que aplicando la misma función a un mensaje cifrado, al descifrar se obtiene el mismo resultado que resultaría de aplicar la

función al mensaje en claro.

$$D(f(C(m, k)), k) = f(m)$$

En la figura 2.13 se muestra un diagrama con el siguiente ejemplo de cifrado homomórfico basado en la aplicación de la operación multiplicación:

( $m$  es el mensaje a cifrar,  $m'$  es el mensaje cifrado)

$$m = [11, 7] \Rightarrow m[0] = 11, m[1] = 7$$

$$f(m) = m[0] \cdot m[1] \Rightarrow$$

$$f(m) = 11 \cdot 7 = 77$$

$$C(m, k) = m'$$

$$C(11|7, k) = [15, 12] = m'$$

$$f(m') = m'[0] \cdot m'[1] \Rightarrow$$

$$f(m') = 15 \cdot 12 = 180$$

$$D(180, k) = 77$$

$$D(f(m')) = f(m)$$

$$D(f(C(m, k)), k) = f(m)$$

Entre los algoritmos parcialmente homomórficos, podemos destacar los siguientes:

**RSA**  $C(m_1) \cdot C(m_2) = C(m_1 \cdot m_2)$

**ElGamal**  $C(m_1) \cdot C(m_2) = C(m_1 \cdot m_2)$

**Benaloh**  $C(m_1) \cdot C(m_2) = C(m_1 + m_2 \pmod{c})$

**Goldwasser-Micali**  $C(x_1) \cdot C(x_2) = C(x_1 \oplus x_2)$

**Paillier**  $C(m_1) \cdot C(m_2) = C((m_1 + m_2) \pmod{c})$

### 2.5.1.5.1. ElGamal

Uno de los algoritmos más utilizados para realizar un cifrado homomórfico es el de ElGamal [25].

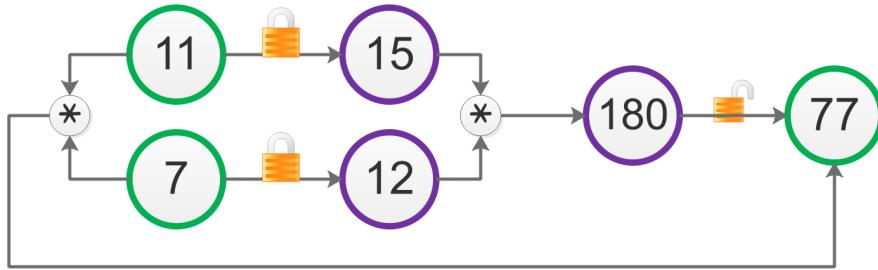


Figura 2.13: Ejemplo de homomorfismo basado en la operación multiplicación

ElGamal es un algoritmo de criptografía asimétrica basado en Diffie-Hellman de cifrado de clave pública que se usa tanto para cifrado como para firma digital. Se apoya en el problema de logaritmos discretos.

El creador de este algoritmo fue Taher Elgamal<sup>27</sup>, en 1984. Es un algoritmo de uso libre.

#### Clave:

El algoritmo de ElGamal necesita un par de claves pública - privada para poder ser utilizado. [38, 47] Para ello, el cifrador del mensaje ha de escoger tres números:

- Número primo  $p$  cualquiera, tal que el logaritmo discreto no soluble en un tiempo asumible en  $Z_p^*$  (grupo multiplicativo módulo un primo  $p$ ). Esto quiere decir que  $p-1$  ha de tener un factor primo grande, lo cual provoca que la resolución del problema de logaritmo discreto sea difícil).
- Número aleatorio  $g$ , que será el generador del grupo cíclico  $Z_p^*$ .
- Número aleatorio  $a$  tal que  $a \in 0, \dots, p-1$ , que será la clave privada.

La clave pública será  $(g, p, K)$  donde  $K = g^a \pmod{p}$ . De este modo, la clave privada  $a$  se mantendrá en secreto.

Para el **cifrado** de un mensaje:

Supongamos que queremos cifrar el mensaje  $m$  tal que  $1 < m < p-1$  (es decir que  $m \in Z_p$ ).

El cifrador ha de escoger un número aleatorio  $b$  tal que  $a \in 2, \dots, p-1$  que mantendrá en secreto.

El mensaje cifrado se corresponde con la tupla  $C_b(m, b) = (y_1, y_2)$ , donde:

$$y_1 = g^b \pmod{p}$$

$$y_2 = K^b \cdot m \pmod{p}$$

<sup>27</sup>[https://es.wikipedia.org/wiki/Taher\\_Elgamal](https://es.wikipedia.org/wiki/Taher_Elgamal)

Para **descifrar** el mensaje  $m$  se utiliza el pequeño teorema de Fermat<sup>28</sup>, por el cual podemos inferir que:

$$\begin{aligned} y_1^{-a} \cdot y_2 \pmod{p} &= (g^b)^{-a} \cdot K^b \cdot m \pmod{p} = g^{-ab} \cdot (g^a)^b \cdot m \pmod{p} = (g^a)^{-b} \cdot (g^a)^b \cdot m \\ &\pmod{p} = (g^a)^{b-b} \cdot m \pmod{p} = (g^a)^0 \cdot m \pmod{p} = 1 \cdot m \pmod{p} = m \pmod{p} \end{aligned}$$

Por tanto:

$$m = D(C_b(m, b)) = D((y_1, y_2)) = y_1^{p-1-a} \cdot y_2 \pmod{p}$$

Un ejemplo de este algoritmo es el siguiente:

Begoña escoge los siguientes valores:

- $p = 101$  (primo aleatorio. Supongamos, aunque no es cierto en este ejemplo, que  $p - 1 = 100$  tiene un factor primo grande.)
- $g = 5$
- $a = 16$  (clave privada)

Con estos valores, se calcula la clave pública.

$$K = g^a \pmod{p} = 5^{16} \pmod{101} = 31 \Rightarrow K = 31$$

$$pubK = (g, p, K) = (5, 101, 31)$$

Marta quiere cifrar el mensaje  $m = 6$  (tal que  $1 < m < p - 1 = 100$ , para lo que escoge un número  $b$  aleatorio  $b = 7$  de forma que  $2 < b < p - 1 = 2 < 7 < 100$ .

Para cifrar el texto, ha de utilizar la clave pública generada por Begoña y calcular:

$$y_1 = g^b \pmod{p} = 5^7 \pmod{101} = 52$$

$$y_2 = K^b \cdot m \pmod{p} = 31^7 \cdot 6 \pmod{101} = 47$$

Así el mensaje  $m = 9$  cifrado será:

$$C_b(m, b) = C_7(9, 7) = (y_1, y_2) = (52, 47) \Rightarrow C(9) = (52, 47)$$

Con este mensaje cifrado, Begoña debe utilizar su clave privada para poder obtener el texto en claro:

$$D(C_b(m, b)) = D(C_7(9, 7)) = D((y_1, y_2)) = D((52, 47)) = y_1^{p-1-a} \cdot y_2 \pmod{p} = 52^{101-1-16} \cdot 47 \pmod{101} = 52^8 \cdot 47 \pmod{101} = 6$$

Este algoritmo es un sistema de cifrado homomórfico respecto de la operación multiplicación. Con esto, se obtiene un proceso en el que el producto de dos mensajes cifrados equivale al cifrado del producto de ambos mensajes.

---

<sup>28</sup>[https://es.wikipedia.org/wiki/Peque%C3%B1o\\_teorema\\_de\\_Fermat](https://es.wikipedia.org/wiki/Peque%C3%B1o_teorema_de_Fermat)

$$C(x_1) \cdot C(x_2) = C(x_1 \cdot x_2)$$

Definamos:

$a$ : clave secreta

$(g, p, K)$ : clave pública

$g$ : generador

$m$ : mensaje

$b$ : aleatoriedad

$$C(m) = (y_1, y_2) = (g^b \pmod{p}, K^b \cdot m \pmod{p})$$

$$C(m_1) \cdot C(m_2) = (g^{b_1}, m_1 \cdot K^{b_1})(g^{b_2}, m_2 \cdot K^{b_2}) = (g^{r_1+r_2}, (m_1 \cdot m_2) \cdot K^{r_1+r_2}) = (g^t, (m_1 \cdot m_2) \cdot h^t) = C(m_1 \cdot m_2)$$

Un ejemplo práctico de este homomorfismo:

$$C(6) = (5^7 \pmod{101}, 31^7 \cdot 6 \pmod{101}) = (52, 47)$$

$$C(8) = (5^7 \pmod{101}, 31^7 \cdot 8 \pmod{101}) = (52, 29)$$

Para descifrar ambos mensajes, utilizaríamos el siguiente proceso:

$$m_1 = y_1^{p-1-a} \cdot y_2 \pmod{p} = 52^{101-1-16} \cdot 47 \pmod{101} = 6$$

$$m_2 = y_1^{p-1-a} \cdot y_2 \pmod{p} = 52^{101-1-16} \cdot 29 \pmod{101} = 8$$

Si multiplicamos los dos mensajes cifrados:

$$C(6) \cdot C(8) = (52, 47) \cdot (52, 29) = (52^2, 47 \cdot 29)$$

$$D(C(6) \cdot C(8)) = D((52^2, 47 \cdot 29)) = [y_1^{p-1-a} \cdot y_2 \pmod{p}] = (52^2)^{101-1-16} \cdot (47 \cdot 29) \pmod{101} = 48 = 6 \cdot 8$$

Así, se observa que si el resultado de descifrar el producto de dos mensajes cifrados es igual al producto de los mensajes sin cifrar.

#### 2.5.1.5.2. ElGamal Exponencial

La propiedad de ElGamal en cuanto a homomorfismo es muy interesante, pero de cara al voto por Internet, presenta ciertos problemas. El principal es que la operación en la que se basa es el producto. Para un sistema orientado a procesos electorales, sería mucho

más útil que el homomorfismo fuera sobre la suma, ya que el escrutinio no es otra cosa que la suma (totalización) de votos. [38, 47] Así, existe una variante de ElGamal que se denomina ElGamal Exponencial. En esta variante, en lugar de encriptar el mensaje  $m$  como en ElGamal tradicional, se cifra  $g^m$ , donde  $g$  es un generador (normalmente se suele reutilizar el mismo que se utiliza para generar la clave pública), transformando el sistema en un homomorfismo aditivo, es decir, sobre la operación suma.

$$C(g^{x_1}) \cdot C(g^{x_2}) = C(g^{x_1+x_2})$$

Para esta variante de ElGamal, definamos:

$a$ : clave secreta

$(g, p, K)$ : clave pública

$g$ : generador

$m$ : mensaje

$b$ : aleatoriedad

y modifiquemos [36] la función del ElGamal multiplicativo de modo que

$$C(m) = (y_1, y_2) = (g^b \pmod{p}, g^m \cdot K^b \pmod{p})$$

Desarrollando:

$$\begin{aligned} C(m_1) \cdot C(m_2) &= (g^{b_1}, g^{m_1} \cdot K^{b_1})(g^{b_2}, g^{m_2} \cdot K^{b_2}) = (g^{b_1} \cdot g^{b_2}, g^{m_1} \cdot K^{b_1} \cdot g^{m_2} \cdot K^{b_2}) = \\ &= (g^{b_1+b_2}, g^{m_1+m_2} \cdot K^{b_1+b_2}) = (g^t, g^{m_1+m_2} \cdot K^{b_1+b_2}) = C(g^{m_1+m_2}) \end{aligned}$$

El descifrado en este caso se obtiene de este modo:

$$y_1^{-a} \cdot y_2 \pmod{p} = (g^b)^{-a} \cdot g^m \cdot K^b \pmod{p} = g^{-ab} \cdot g^m \cdot g^{ab} \pmod{p} = g^m \pmod{p}$$

$$D(C(y_1, y_2)) = y_1^{-a} \cdot y_2 \pmod{p} = g^m \pmod{p}$$

$$m = \log_b \frac{y_2}{y_1^a}$$

Un ejemplo práctico de este homomorfismo aditivo:

$$C(6) = (5^7 \pmod{101}, 31^7 \cdot 5^6 \pmod{101}) = (52, 68)$$

$$C(8) = (5^7 \pmod{101}, 31^7 \cdot 5^8 \pmod{101}) = (52, 84)$$

Para descifrar ambos mensajes, utilizaríamos el siguiente proceso:

$$D(C(6)) \Rightarrow y_1^{p-1-a} \cdot y_2 \pmod{p} = 52^{101-1-16} \cdot 68 \pmod{101} = 71 \Rightarrow 71 = g^m \pmod{101}$$

Para obtener el descifrado, estoy haciendo uso de una herramienta online que resuelve el logaritmo discreto<sup>29</sup> de forma que obtiene el exponente de la ecuación  $Base^{Exponent} = Power \pmod{Modulus}$

Para ello, se pasan 3 parámetros (Base, Power y Modulus) y la aplicación calcula el exponente.

Así, Base = 5, Power = 71, Modulus = 101  $\Rightarrow m = 6$

$$D(C(8)) \Rightarrow y_1^{p-1-a} \cdot y_2 \pmod{p} = 52^{101-1-16} \cdot 84 \pmod{101} = 58 \Rightarrow 58 = g^m \pmod{101}$$

Base = 5, Power = 58, Modulus = 101  $\Rightarrow m = 8$

Si multiplicamos los dos mensajes cifrados:

$$C(6) \cdot C(8) = (52, 68) \cdot (52, 84) = (52^2, 68 \cdot 84) = (2704, 5712)$$

$$D(C(6) \cdot C(8)) = D((2704, 5712)) \Rightarrow [y_1^{p-1-a} \cdot y_2 \pmod{p}] \Rightarrow (2704)^{101-1-16} \cdot 5712 \pmod{101} = 78$$

Base = 5, Power = 78, Modulus = 101  $\Rightarrow m = 14 = 8 + 6$

Así, se observa que, con este cambio en el algoritmo, si el resultado de descifrar el producto de dos mensajes cifrados es igual a la suma de los mensajes sin cifrar.

### 2.5.2. Esquemas de Voto Electrónico

Los sistemas de voto electrónico están formados por un diseño conceptual y el llamado esquema o paradigma de voto electrónico (E-Voting Schemes - EVS). El esquema es el núcleo del sistema, lo que asegura que los requisitos se cumplan.

La práctica totalidad de estos esquemas usan mecanismos y principios criptográficos.

Los esquemas de voto electrónico se basan en una primitiva criptográfica o en un conjunto de ellas. Por eso, hay una serie de esquemas publicados apoyados sobre alguna de las primitivas introducidas en el apartado anterior.

Podemos clasificar varios tipos de esquemas de voto electrónico entre los más usados según las publicaciones de una serie de expertos en el campo del voto electrónico criptográfico:

- Esquema de Voto Electrónico basado en **Cifrado Homomórfico**

El votante emite su voto codificado y el recuento se realiza sin descodificar los votos.

De esta forma se consigue que no se vulnere el secreto del voto. Para poder reali-

---

<sup>29</sup><https://www.alpertron.com.ar/DILOG.HTM>

zar esta descodificación, el elector debe instalar algún software desarrollado por la autoridad electoral para realizar las operaciones criptográficas.

- Esquema de Voto Electrónico basado en **Canales Anónimos**

Se trata de un esquema bastante seguro, aunque complejo al mismo tiempo. Se trata el anonimato del votante ocultando el origen de los votos que recibe el sistema.

- Esquema de Voto Electrónico basado en **Mixnets**

El esquema basado en mixnets (redes mixtas) define la existencia de una serie de servidores enlazados. Cada uno de estos servidores recibe un grupo de mensajes encriptados, los reordena, los vuelve a encriptar de forma aleatoria y los envía al siguiente servidor. Con este proceso se consigue que no sea posible asociar la información de los mensajes de entrada con los de salida, rompiendo la relación votante-voto del sistema.

La desencriptación de los votos se puede realizar tanto en cada servidor (por medio de su propia clave) como al finalizar el proceso utilizando una clave distribuida entre varios de los servidores.

- Esquema de Voto Electrónico basado en **Secreto Compartido**

En el esquema de voto electrónico el votante comparte su voto entre varias autoridades electorales. Una vez finalizado el proceso de votación, cada autoridad computa los votos que ha recibido y los pone en común con el resto de autoridades electorales que toman parte en la elección. Así se obtiene el resultado total del proceso.

- Esquema de Voto Electrónico basado en **Pruebas de Conocimiento Nulo**

- Esquema de Voto Electrónico basado en **Firma Ciega**

En un Esquema de Firma Ciega, el firmante no conoce el contenido del mensaje que firma, ya que el emisor del mismo realiza un proceso previo para ocultar su contenido, lo que se conoce por *cegar* el mensaje.

Se caracteriza porque la entidad firmante no adquiere ningún conocimiento sobre el contenido del mensaje que está firmando, aunque, con posterioridad, la firma obtenida puede ser verificada como válida tanto por esta entidad firmante como cualquier otra entidad que disponga de la información necesaria.

Se caracteriza porque la entidad firmante no adquiere ningún conocimiento sobre el contenido del mensaje que está firmando, aunque, con posterioridad, la firma obtenida puede ser verificada como válida tanto por esta entidad firmante como cualquier otra entidad que disponga de la información necesaria.

Los esquemas que se basan en protocolos con firma ciega suelen usar canales anónimos para enviar tanto la firma como el voto cifrado a la autoridad electoral, con lo que protege el anonimato del votante.

Podemos encontrar este esquema en soluciones como la propuesta en 1992 por Fujioka en [27], la cual sirvió de base a Cranor<sup>30</sup> para la implementación de un prototipo (Sensus<sup>31</sup><sup>32</sup>).

El esquema desarrollado en Sensus divide el proceso en cuatro etapas: *inicialización, registro, votación y recuento*.

- Esquema de Voto Electrónico basado en **papeletas precifradas**

Este esquema de voto electrónico aparece en la tesis de Morales Rocha [40].

Aprovechando que en el último esquema se cita la tesis de Morales Rocha [40], vamos a introducir una alternativa en cuanto al tipo de esquemas de voto electrónico existentes. En esta tesis, el autor define cuatro grupos de esquemas de voto electrónico remoto. Estos se diferencian en la forma en la que usan los elementos criptográficos para tratar de resolver los requisitos de seguridad de un sistema electoral:

- Esquemas basados en **firma ciega**
- Esquemas basados en **mixnets**
- Esquemas basados en **cifrado homomórfico**
- Esquemas basados en **papeletas precifradas**

El propio autor de la tesis citada [40], incorpora (en la página 109) un resumen con las ventajas y desventajas que ofrece cada uno de estos esquemas (tabla 2.6).

Está fuera del alcance de este proyecto el estudio de estos esquemas y sus evoluciones, pero nos basamos en esta información para el desarrollo del sistema que se implementa. Para ahondar en ellos, recomiendo la lectura de la citada tesis de Morales Rocha [40], así como el capítulo 4 de la tesis de la Dra. Emilia Pérez Belleboni [49], en la cual se expone una recopilación de información muy concisa sobre multitud de esquemas y sistemas que los implementan, según las necesidades que se necesiten cubrir.

## 2.6. Estado actual de las tecnologías

### 2.6.1. Certificados Digitales

La web de la Fábrica Nacional de Moneda y Timbre [12] indica que *un certificado digital es un documento electrónico que asocia una clave pública con la identidad de su propietario*.

---

<sup>30</sup><http://lorrie.cranor.org/>

<sup>31</sup><http://lorrie.cranor.org/voting/sensus/>

<sup>32</sup><http://lorrie.cranor.org/pubs/hicss/hicss.html>

Clasificación	Ventajas	Desventajas
Esquemas basados en firma ciega	<ul style="list-style-type: none"> <li>Protegen la privacidad del votante al separar los procesos de autenticación y voto.</li> </ul>	<ul style="list-style-type: none"> <li>La protección del anonimato puede verse afectada si un atacante monitorea el canal de comunicación.</li> <li>Con el conocimiento de la clave privada de la autoridad de autenticación se pueden añadir votos no legítimos.</li> </ul>
Esquemas basados en mixnets	<ul style="list-style-type: none"> <li>Protegen la privacidad del votante a través de las permutaciones llevadas a cabo.</li> </ul>	<ul style="list-style-type: none"> <li>Difícil verificación de que los servidores mix han actuado correctamente.</li> <li>En el caso de mixnets de descifrado, el terminal de votación requiere de alta capacidad de cómputo.</li> </ul>
Esquemas basados en cifrado homomórfico	<ul style="list-style-type: none"> <li>Protegen la privacidad del votante al no tener que desencriptar los votos individualmente para llevar a cabo el escrutinio.</li> </ul>	<ul style="list-style-type: none"> <li>No soportan todo tipo de elecciones.</li> <li>Son susceptibles a ataques en donde votantes deshonestos pueden enviar un mensaje que represente más de un voto para un candidato.</li> </ul>
Esquemas basados en papeletas precifradas	<ul style="list-style-type: none"> <li>Protegen la privacidad del votante ya que este envía como voto un código cuya relación con el candidato es desconocida para el servidor de votación.</li> <li>Evitan ataques de código malicioso que trate de alterar o conocer el contenido del voto.</li> <li>El voto puede ser enviado desde un dispositivo con baja capacidad de cómputo.</li> <li>Permiten al votante verificar que su voto se ha recibido correctamente en el servidor de votación.</li> </ul>	<ul style="list-style-type: none"> <li>Posibles alteraciones en las papeletas precifradas sin detección, lo cual ocasionaría que el votante envíe un voto diferente al deseado.</li> <li>Se pueden presentar problemas de logística en la distribución de las papeletas a los votantes.</li> <li>Votantes no pueden verificar que su voto fue incluido en el escrutinio sin arriesgar un ataque de coerción masiva.</li> <li>Poca usabilidad al tener que teclear códigos de votación.</li> </ul>

Tabla 2.6: Resumen de ventajas y desventajas de los esquemas de voto electrónico según Morales Rocha [40] (p. 109)

Complementa la definición añadiendo que *”además de la clave pública y la identidad de su propietario, un certificado digital puede contener otros atributos para, por ejemplo, concretar el ámbito de utilización de la clave pública, las fechas de inicio y fin de la validez del certificado, etc. El usuario que haga uso del certificado podrá, gracias a los distintos atributos que posee, conocer más detalles sobre las características del mismo”*.

La utilidad de los certificados digitales, simplificando el contexto, se resume en *asegurar que una determinada clave pública pertenece a un usuario en concreto*.

Con las tecnologías actuales, la economía ha virado su desarrollo hacia el comercio electrónico y las relaciones remotas. Muchas transacciones que antes se realizaban en persona han evolucionado al mundo digital, por lo que, para la mayoría de ellas es indispensable poseer mecanismos que puedan demostrar que los sujetos intervenientes en la comunicación están únicamente identificados y con la seguridad de que no se produce suplantación.

Una herramienta fundamental para cumplir con este propósito ha sido el desarrollo de las certificaciones digitales.

Los certificados digitales permiten cifrar las comunicaciones, permitiendo tan sólo al destinatario de estas acceder al contenido.

Los certificados electrónicos están expedidos por una Autoridad de Certificación e identifica a una persona con un par de claves criptográficas, una pública y otra privada, generadas mediante un algoritmo matemático. Ambas son complementarias, de forma que lo que cifra una sólo lo puede descifrar la otra, y viceversa. La diferencia entre ellas es que la clave privada está pensada para que nunca salga del certificado y permanezca siempre bajo control del firmante. La clave pública se puede enviar a otros usuarios.

Tienen como objetivo validar y certificar que una firma electrónica se corresponde con una persona concreta. Por esta razón, para dar fe de que el certificado se corresponde con una persona concreta, es por lo que los certificados están firmados, a su vez por la Autoridad de Certificación.

## 2.6.2. DNIe

El DNIe es un documento con una antigüedad de más de setenta años. Tras los primeros, ideados en 1944 y estrenados en 1961, entre 2006 y 2015 se añadió al documento un chip con certificados válidos para la identificación y firma digital. Era el llamado DNIe 2.0. A partir de 2015 y sobre todo ya en 2016, el documento fue evolucionado a una tercera versión, la cual, como elemento más novedoso, añadía al chip de contacto otro chip sin contacto con tecnología RFID, más concretamente,



Figura 2.14: Anverso de un ejemplar del DNIe 3.0

NFC.

Para utilizar el nuevo chip sin contacto, el usuario necesita un dispositivo (smartphone o tablet) con tecnología NFC y una app que dé el servicio que requiera de identificación conectándose al documento.

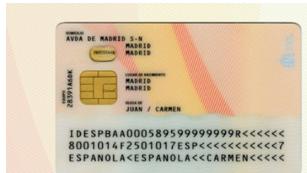


Figura 2.15: Reverso de un ejemplar del DNIe 3.0

Hay que tener en cuenta que no basta con acercar el DNIe al dispositivo móvil con NFC para que éste lea la información contenida en aquél. El usuario deberá introducir dos códigos, uno es el PACE, que encontrará en el anverso del documento físico, y el otro es su PIN personal que establece en las máquinas habilitadas para ello en comisarías.

El chip electrónico almacena los datos personales y la fotografía del titular del documento, su firma manuscrita digitalizada y el patrón de su huella dactilar. Junto a estos datos personales, encontramos los certificados digitales de autenticación y de firma electrónica.

Este conjunto de información y certificados permite al titular del documento identificarse tanto de forma física como remota, así como para realizar trámites electrónicos en los que se requiera la identificación unívoca de la persona.

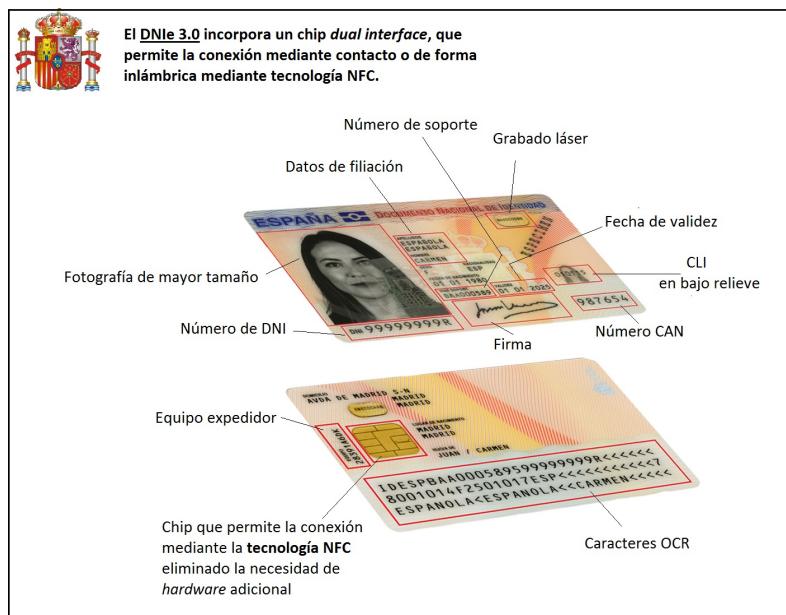


Figura 2.16: Especificaciones más relevantes del DNIe 3.0

La firma electrónica cuenta con una clave pública y otra privada. Al realizar una operación electrónica, el receptor utiliza la clave pública para asegurarse de la identificación del emisor, mientras que éste utiliza su clave privada para firmar la operación.

El certificado de autenticación asegura que la persona que está utilizando el documento realmente es quien dice ser.

Una buena fuente de información para consultar las funcionalidades del DNIe 3.0 es la ofrecida en el propio portal<sup>33</sup> del DNIe por la Policía Nacional de España en [19].

De esta fuente se obtienen los datos comparativos que se muestran en la tabla 2.7 acerca de las características de los chips de las versiones 2.0 y 3.0 del DNIe.

El proceso a seguir cuando se está en posesión todos los certificados necesarios para operar con el DNIe es (para cada uno de los certificados) [20]:

- Verificar que el certificado fue firmado usando la clave privada que corresponde a la clave pública de su emisor. Este paso no es necesario para el certificado de la CA raíz.
- Verificar la validez del certificado, es decir, no ha caducado.
- Realizar la validación OCSP contra el servidor de la FNMT.

Durante esta memoria se desglosa más información acerca del DNIe 3.0, ya que es uno de los elementos claves del PFC. Por ello, para mayor información, recomiendo visitar aquellos puntos en los que se explican ventajas y desventajas, motivos de adopción tecnológica, implementación, integración. (3.3.2.1, 5.3.2, 6.2.2)

### 2.6.3. NFC

NFC (Near Field Communication - Comunicación de Campo Cercano) se trata de una tecnología de comunicación inalámbrica de corto alcance.

El origen de esta tecnología está en el RFID, tecnología ya bastante utilizada en múltiples campos de nuestra vida cotidiana. Un buen ejemplo son las etiquetas en prendas en tiendas de ropa o en productos de supermercado. Si tratas de llevarte algo sin pagar, al pasar por unos arcos a la salida preparados con cierto campo magnético, las etiquetas RFID activan una alarma, la cual se desactiva una vez se ha abonado el producto.

RFID es una tecnología de comunicación inalámbrica centrada en la transmisión de un identificador mediante ondas de radio. Así, a cualquier producto se le puede asociar un identificador único a través de una etiqueta RFID y así poder ser detectado y leído por un lector apropiado de forma inalámbrica.

En el caso de NFC, se trata de un concepto similar, pero buscando la transmisión de datos más complejos que un simple identificador.



Figura 2.17: Logo de RFID

<sup>33</sup> <https://www.dnielectronico.es/PortalDNIe/>

Característica	DNIe	DNIe 3.0
<b>Chip</b>	ST19WL34	SLE78CLFX408AP de Infineon Technologies
<b>Sistema operativo</b>	DNIe v1.1	DNIe v4.0
<b>Capacidad</b>	32Kb	400KB memoria Flash, 8KB memoria RAM
<b>Interfaz</b>	Single (con contacto)	Dual (con contacto y sin contacto)
<b>Criptolibrería</b>		RSA
<b>Certificación</b>		CC EAL5+
<b>Zona pública:</b> Accesible en lectura sin restricciones	<ul style="list-style-type: none"> <li>• Certificado CA intermedio emisora</li> <li>• Claves Diffie-Hellman</li> <li>• Certificado x509 de componente</li> </ul>	
<b>Zona privada:</b> Accesible en lectura por el ciudadano, mediante el uso de su PIN	<ul style="list-style-type: none"> <li>• Certificado de Firma (no repudio)</li> <li>• Certificado de Autenticación</li> </ul>	
<b>Zona de seguridad:</b> Accesible en lectura por el ciudadano, en los Puntos de Actualización del DNI.	<ul style="list-style-type: none"> <li>• Datos de filiación del ciudadano (los mismos que están impresos en el soporte físico del DNI)</li> <li>• Imagen de la fotografía</li> <li>• Imagen de la firma manuscrita</li> </ul>	
<b>Datos criptográficos:</b> Claves de ciudadano	<ul style="list-style-type: none"> <li>• Clave RSA pública de autenticación (Digital Signature)</li> <li>• Clave RSA pública de no repudio (ContentCommitment)</li> <li>• Clave RSA privada de autenticación (Digital Signature)</li> <li>• Clave RSA privada de firma (ContentCommitment)</li> <li>• Patrón de impresión dactilar</li> <li>• Clave Pública de root CA para certificados card-verificables</li> <li>• Claves Diffie-Hellman</li> </ul>	

Fuente: Portal del DNI electrónico. Ministerio del Interior. Dirección General de la Policía. Cuerpo Nacional de Policía. [https://www.dnielectronico.es/PortalDNIe/PRF1\\_Cons02.action?pag=REF\\_1078&id\\_menu=\[26\\_%2030\]](https://www.dnielectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_1078&id_menu=[26_%2030]).

Tabla 2.7: Comparativa DNIe - DNIe 3.0.

NFC, se basa en la norma ISO 14443 (RFID), un estándar internacional relacionado con tarjetas de identificación electrónica, especialmente tarjetas de proximidad. Como establece esta norma estándar, NFC se comunica mediante inducción en un campo magnético. Por tanto cada tarjeta NFC cuenta con una antena en espiral, con lo que cuando se ambas (origen-destino) se colocan dentro de un campo electromagnético cercano, se puede realizar la transmisión de datos. Cumpliendo igualmente con la norma, esta tecnología trabaja en una banda de frecuencia de 13,56 MHz.

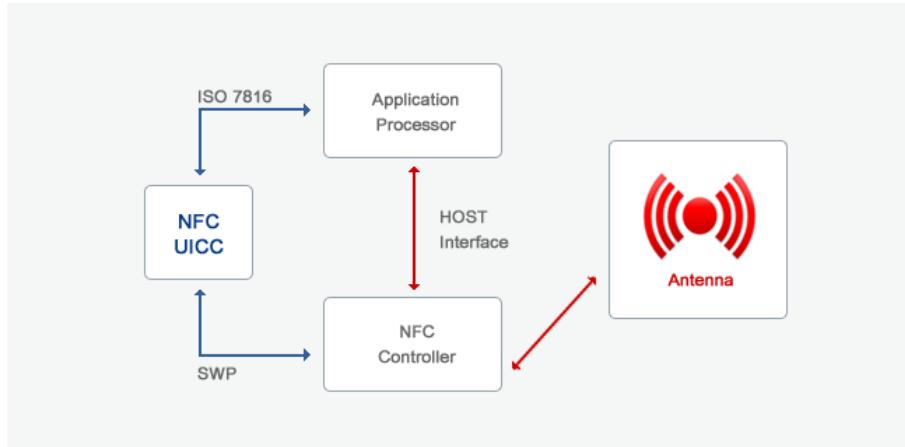


Figura 2.18: Arquitectura de un chip NFC

Este campo electromagnético debe ser producido por uno de los dispositivos o por ambos al mismo tiempo. Esto provoca que los dispositivos que implementan el estándar NFCIP-1 tienen la capacidad de funcionar de dos modos:

**Modo Activo:** En el modo activo, ambos dispositivos generan su propio campo electromagnético.

**Modo Pasivo:** En el modo pasivo, uno de los dispositivos genera el campo electromagnético y el otro se aprovecha de la modulación de la carga para poder transferir de datos. Es el dispositivo que inicia la comunicación el que genera el campo.

Este protocolo de comunicaciones puede trabajar a diferentes velocidades: 106, 212, 424 ó 848 Kbits/s. La velocidad de la transmisión se negocia al comienzo de la comunicación entre los dispositivos, aunque puede regularse según sea necesario variando ciertos parámetros.

Según NFC Forum, organización fundada en 2004 por Philips, Nokia y Sony y que hoy reúne alrededor de 170 miembros, la conexión entre dos dispositivos se realiza de forma automática cuando estos se encuentran cerca el uno del otro, digamos, a unos 5cm de distancia. De todos modos, esta distancia es variable en cuanto a que se puede ver afectada por ciertos factores como pueden ser el tipo de emisor/receptor, temperatura, aislantes, etc. Este foro fija la máxima distancia para operar con este canal de transmisión en 20cm,

con el propósito de cubrir la seguridad de la información, que esta no pueda ser accedida de forma remota por un atacante a distancia.

Dentro de los modos de comunicación activo y pasivo también se definen 3 modos de comunicación:

**Modo Lectura / Escritura** Con este modo, las aplicaciones pueden transferir datos en un formato definido por el NFC Forum, aunque es un modo no considerado seguro.

**Modo emulación NFC Card** Este modo permite al dispositivo actuar como una Smart-card estándar, es decir, la transmisión de datos se encuentra securizada.

**Modo Peer to Peer** Este modo permite la comunicación directa entre dos dispositivos a nivel de enlace.

Hoy en día, el uso del NFC está bastante extendido en cuanto a que muchos dispositivos entre smartphone y tablets lo implementa de serie. Además, está desarrollándose el pago con tarjetas de débito/crédito contactless, por lo que muchos comercios ya poseen TPVs para tarjetas sin contacto, o incluso con pago a través del móvil por NFC.

Igualmente, desde hace un tiempo, existen tarjetas NFC con las que se pueden realizar tareas al poner en contacto con el dispositivo móvil. Un ejemplo de funcionamiento sería colocar una etiqueta NFC junto a la puerta y que al pasar el móvil por ella se encienda la luz de la habitación por medio de una bombilla inteligente. O colocar una tarjeta junto a la puerta de casa que indique al móvil que desconecte el WiFi al salir de casa para disminuir el consumo de batería.

También se utiliza el NFC en marketing. Hay anuncios que incorporan etiquetas NFC en ellos. Concretamente, en junio de 2010, una empresa de Motril diseñó el primer anuncio en España que incorporaba una etiqueta NFC situada bajo pegatinas colocadas en motos que realizaban una ruta deportiva.

Ya en 2015, con la salida del DNIe 3.0, la tecnología NFC llegó a la identificación de los ciudadanos españoles. Esta nueva versión del carnet de identidad complementa el chip con contacto que incluía la primera versión del DNIe con un chip sin contacto que realiza las mismas funciones sin necesidad de un lector de chips del primer tipo.

Always vote for principle, though you may vote alone, and you may cherish the sweetest reflection that your vote is never lost.

---

*John Quincy Adams<sup>1</sup>*

## Capítulo 3

# Planteamiento

### 3.1. Objetivos finales del proyecto

El objetivo principal de este Proyecto de Fin de Carrera es la implementación de un sistema de votación electrónica remota (i-voting) diseñada de forma ad-hoc para dos de los procesos electorales que se llevan a cabo en la Escuela Politécnica Superior de la Universidad San Pablo CEU.

Estos procesos están definidos en las *Normas de Organización y Funcionamiento de la Universidad San Pablo-CEU* [44] y son:

- Elecciones de Miembros de la Junta Electoral
- Elecciones de Delegados y Subdelegados

El sistema que se propone en esta memoria es un sistema **robusto, fiable, verificable y auditabile**, buscando satisfacer las exigencias de seguridad de procesos electorales más ambiciosos que los que tratamos en este proyecto, los cuales abarcan el ámbito universitario. Por tanto, la intención de este PFC, no es la de simplemente realizar estos comicios de forma electrónica, sino tratar de diseñar un sistema con idea de que pudiera ser escalable para niveles superiores al ámbito de la Escuela o la Universidad.

Como se ha avanzado al inicio de esta sección, el sistema propuesto busca tener estas características:

- **Robusto** El sistema debe ser tolerante tanto a fallos como ataques externos e internos.
- **Fiable** El sistema cumple con requisitos de seguridad que satisfacen la privacidad y la precisión de votos y votantes.

---

<sup>1</sup>6º Presidente de los Estados Unidos de América (1825-1829) [6] <https://goo.gl/bf8zDP> [https://es.wikipedia.org/wiki/John\\_Quincy\\_Adams](https://es.wikipedia.org/wiki/John_Quincy_Adams)

- **Verifiable** Se puede verificar que los votos han sido contados y forman parte del resultado del escrutinio.
- **Auditabile** El sistema debe proporcionar mecanismos para que pueda llevarse una auditoría del mismo antes, durante y después del proceso.

El germen de la idea del proyecto, como se comenta en las Motivaciones del proyecto (1.1) era la búsqueda de soluciones para elecciones generales, autonómicas, municipales, etc. Junto a la idea de desarrollar soluciones para este tipo de comicios, también está el estudio de las circunstancias por las que todavía no se han implementado sistemas de este tipo en España de carácter general.

Al realizar el estudio del estado del arte, se puede recopilar una ingente documentación teórica sobre diferentes sistemas, paradigmas y esquemas de todo tipo de votación electrónica. Hay muchos artículos y tesis muy importantes que tratan de desarrollar sistemas de este tipo, desde el punto de vista teórico hasta el prototipo práctico. Incluso tenemos estados que han implementado una solución con carácter vinculante. En este PFC, lejos de tratar de encontrar una solución novedosa y revolucionaria, vamos a tratar de plasmar un conjunto de ideas y proposiciones de diferentes autores para implementar un sistema propio para la escuela que cumpla con el mayor número de requisitos básicos y deseables para el voto electrónico, teniendo en cuenta el factor remoto, es decir, que cumpla - además de con los requisitos del voto electrónico - con requisitos de control, confiabilidad y seguridad que hagan viable el voto a través de Internet.

Así pues, el objetivo general del PFC será:

Diseñar un esquema y un sistema de voto electrónico remoto a través de Internet que sea robusto, fiable, verificable y auditabile para que se puedan llevar a cabo las Elecciones a Miembro de la Junta de Escuela de la Escuela Politécnica Superior de la Universidad San Pablo CEU.

Teniendo en cuenta el objetivo general, se definen una serie de objetivos intermedios:

OBJETIVO 1 Definir un esquema de voto electrónico que soporte la implementación del sistema en base a los requisitos del mismo.

OBJETIVO 2 Especificar el mecanismo y los protocolos para la identificación de votantes en el sistema y la emisión de los votos con bajo riesgo de coacción.

OBJETIVO 3 Especificar los mecanismos y protocolos para una segura recepción de los votos, así como un correcto escrutinio y una veraz publicación de resultados.

OBJETIVO 4 Plantear un sistema de voto existente que cumpla con los requisitos del voto electrónico y sea adaptable a las necesidades de los procesos electorales a acometer.

OBJETIVO 5 Poner a disposición de votantes y observadores un mecanismo que demuestre que los votos han sido correctamente emitidos y contabilizados y que la integridad de la elección no ha sido comprometida.

OBJETIVO 6 Construir un prototipo de este sistema cumpliendo con el diseño de este PFC.

## 3.2. Descripción del sistema actual

En estos momentos, las elecciones en la Escuela Politécnica Superior se realizan con el método tradicional de voto en urna, sin componentes electrónicos que gestionen el proceso.

### 3.2.1. Elecciones a la Junta de Escuela de la EPS

#### 3.2.1.1. Definición de la Junta de Escuela

Según el documento **NORMAS DE ORGANIZACIÓN Y FUNCIONAMIENTO DE LA UNIVERSIDAD SAN PABLO-CEU** [44], en su Artículo 9, *"Las Facultades, Escuelas y Centros integrados o adscritos son las instancias responsables de la organización de la enseñanza e investigación, de acuerdo con las directrices emanadas de los órganos superiores de la Universidad, y de los procesos académicos, administrativos y de gestión conducentes a la obtención de títulos de carácter oficial y validez en todo el territorio nacional, así como de aquellas otras funciones que determinen las presentes Normas de Organización y Funcionamiento y los restantes reglamentos universitarios."*

A partir de esta definición, en el *Capítulo II. De los órganos académicos*, encontramos el Artículo 22, *Tipos de órganos*, donde se establece *"(1c) que las Juntas de Facultad, Escuela o Centro son órganos colegiados"*. Y encontramos su definición en el Artículo 31, *Las Juntas de Centros*, donde podemos leer que *"La Junta de Facultad, Escuela o Centro es el órgano colegiado de gobierno del mismo, que ejerce sus funciones con vinculación a los acuerdos del Patronato, Consejo de Gobierno y resoluciones del Rector."*

También podemos destacar los artículos 32 y 33, donde se establece la composición y funciones de las Juntas de Facultad, Centro o Escuela:

- *Artículo 32: Composición de las Juntas*

*La Junta de Facultad, Escuela o Centro estará compuesta por miembros natos y electos.*

*Son miembros natos: El Decano o Director, que presidirá sus reuniones; los Vicedecanos o Subdirectores, el Secretario académico, que levantará acta de sus sesiones y los Directores de los Departamentos integrados en la Facultad o Escuela.*

*Son miembros electos: Quienes resulten elegidos en representación del profesorado y de los alumnos de acuerdo con la normativa que reglamentariamente se establezca.*

- *Artículo 33: Funciones de las Juntas Las competencias de la Junta de Facultad, Escuela o Centro son:*
  - a) *Colaborar con el Decano o Director en la gestión de la Facultad, Escuela o Centro.*
  - b) *Promover el perfeccionamiento de los planes de estudio y de la metodología docente, así como el establecimiento de nuevos títulos tanto propios como oficiales.*
  - c) *Participar en la programación de las actividades de extensión universitaria.*
  - d) *Velar por la adecuada dotación de los servicios necesarios para su correcto funcionamiento.*
  - e) *Cualquier otra competencia que le pueda ser atribuida en el desarrollo de estas Normas de Organización y Funcionamiento.*

### **3.2.1.2. Plazos del Proceso Electoral**

- Convocatoria
- Presentación de candidaturas
- Publicación del censo
- Constitución de la Junta Electoral
- Designación de las mesas electorales

## **3.3. Fases del proceso electoral**

El Consejo Europeo, dentro de si definición de estándares, recomienda la aceptación para utilizar el EML. Desarrollado por la OASIS, su objetivo es el de permitir el "intercambio de información entre hardware, software y proveedores de servicios implicados en cualquier aspecto relacionado con el desarrollo de elecciones o servicios a votantes tanto para organizaciones públicas como privadas." [16]

Según el estándar EML, se distinguen cuatro actores implicados y tres fases en el proceso electoral.

### **Actores**

- Autoridad

- Administrador
- Votante
- Auditor/observador

### Fases

- Fase Pre-Voting
- Fase Voting
- Fase Post-Voting

Teniendo en cuenta el EML, identificamos las tres fases que dirigen el proceso electoral como fases preelectoral, electoral y postelectoral, adecuándolas al idioma español. Cada una de estas fases contiene una serie de procesos o tareas determinadas.

#### ● Fase Preelectoral

- **Definición de los límites o reglas de la elección:** Deben definirse de forma que no den lugar a ambigüedades. Qué se vota, a quién se vota, de qué forma, cómo se cuentan los votos o se asignan los cargos. Quiénes pueden votar, cuándo comienza y finaliza el sufragio.
- **Elaboración del censo:** Las autoridades de la Elección deben realizar un proceso de elaboración del censo electoral, para identificar qué votantes tienen derecho a ejercer el voto y dónde (con qué opciones de voto).
- **Registro de votantes:** Puede ser necesario que, según los mecanismos de identificación a utilizar, el votante deba registrarse previamente a la elección frente a la Autoridad Electoral, con el fin de, si no existe censo electoral formalizado, introducirse en el censo de la elección o, si existe ese censo previo, obtener la acreditación identificativa necesaria para poder votar de forma remota con las garantías avaladas por la autoridad electoral.
- **Presentación de candidaturas:** A efectos del sistema informático que desarrollamos es el proceso en el que la autoridad electoral define qué candidaturas pueden ser elegidas por cada votante en cada circunscripción lógica.

#### ● Fase Electoral (Votación)

- **Identificación:** El primer paso del proceso de votación es el de la identificación del votante. Como ya se ha planteado, la identificación del votante es uno de los procesos críticos de una elección, pues, el sistema debe cumplir con varios

requisitos básicos del voto electrónico, como puede ser el principio de autenticidad (en el que sólo los votantes autorizados pueden votar) o el democrático (por el cual el votante que tiene derecho a votar es sólo para hacerlo una vez).

- **Votación:** El momento en el que el votante, ya identificado, visualiza las opciones entre las que puede elegir y ejerce su voto a una o varias de ellas (dependiendo del tipo de elección).
- **Totalización de resultados:** La totalización de resultados es la fase del proceso electoral en la que se agregan los resultados contados en cada una de las mesas electorales en los diferentes niveles que componen la división territorial electoral. Por ejemplo, se suman todas las mesas de un colegio y se obtienen los resultados de ese colegio. Se suman estos para obtener los resultados de un distrito. Estos se sumarían para obtener los de una localidad, estos para los de una provincia, los de la Comunidad Autónoma, los de nivel nacional. Etc. Además, en cada elección, dependiendo de cómo sea, el nivel básico de circunscripción variará, siendo la localidad, por ejemplo, la circunscripción en elecciones Municipales o la provincia (en casi todas las comunidades autónomas) en caso de elecciones autonómicas o nacionales.

En elecciones de voto electrónico dependerá de si son de urna electrónica centralizada o virtualmente distribuida para realizar la totalización simulando mesas electorales o no. Igualmente, las circunscripciones en este tipo de elecciones pueden ser cambiadas de forma lógica, pensando en qué votantes han de elegir diferentes candidatos, en lugar de tomar como base la segregación territorial.

- **Fase postelectoral**

- **Difusión de resultados:** La difusión de resultados es la fase que tiene la responsabilidad de publicar los resultados de forma oficial u oficiosa. La velocidad de difusión de los resultados es importante para afianzar la transparencia del proceso. En los conteos tradicionales se muestra según van siendo totalizados los resultados de cada mesa, mostrando los resultados con pequeñas variaciones en el porcentaje de censo escrutado, en lugar de esperar a recibir toda la información y sumarla, lo que supondría difundir tan sólo una vez, aunque los resultados al 100 % del censo escrutado. Esto, en elecciones de voto electrónico o por Internet, es más complicado que ocurra si la topología en la que se basa el sistema es de urna electrónica centralizada en lugar de distribuida.
- **Auditoría:** No es una fase propiamente dicha en el sentido cronológico en el que se han definido las anteriores. La fase de auditoría abarca todas las etapas del proce-

so, en mayor o menor medida, puesto que debe permitir la vigilancia del correcto funcionamiento del sistema en todas ellas.

### 3.3.1. Fase preelectoral

#### 3.3.1.1. Definición de los límites o reglas de la elección

Para ejercer la democracia de forma correcta las "reglas del juego" deben estar bien definidas, de forma clara y concisa, estableciendo los límites, los mecanismos, las fechas y todo lo necesario para una correcta interpretación, sin lugar a ambigüedades.

Estas reglas de la elección son responsabilidad de la Autoridad Electoral encargada de la organización de los comicios, así como del organismo que los convoca. De cara al sistema informático, esta fase preelectoral es la que sienta las bases de la lógica de negocio del sistema. Ya que define las reglas que el sistema deberá cumplir para llevar a cabo correctamente la elección.

Estas son algunas reglas que son definidas en esta fase:

- Qué cargos se eligen.
- Quiénes tienen autorización para votar.
- Fechas del proceso: registro de votantes, inicio y fin de la votación, publicación de resultados.
- Definición de circunscripciones lógicas.
- Tipo de papeletas.
- Tipo de voto permitido. Incluyendo el número de opciones que un votante puede votar (sólo una o un número máximo) y número de veces que lo puede hacer.
- Asignación de roles a usuarios encargados de administrar la elección o custodiar los trozos de la clave de descifrado compartida.

#### 3.3.1.2. Elaboración del censo

Uno de los cometidos de la Autoridad Electoral previamente a la celebración de unos comicios es la elaboración de un censo electoral completo y fiable que les permita tener un control de cuánta gente y quiénes disfrutan del derecho a votar. Además, este censo debe recoger a qué circunscripción pertenece cada votante y la mesa/urna donde debe realizar su voto.

Una circunscripción es una división electoral. Pensando en elecciones legislativas de España, por ejemplo, casi todas las provincias son unicircunscripcionales, excepto el Principado de Asturias, que se conforma con 3 circunscripciones y la Región de Murcia, compuesta por 5 circunscripciones. Sin embargo, para las Elecciones al Parlamento Europeo, España registra sus votos como una única circunscripción.

Al asignar cargos basándose en circunscripciones, es básico que en el censo esté definido en cuál de ellas vota cada votante. Además, en cada circunscripción, los candidatos varían, por lo que las papeletas entre las que cada votante puede elegir no serán iguales de unas circunscripciones a otras.

Extrapolando a las Elecciones a la Junta de Escuela de la EPS, podemos identificar varias de estas circunscripciones, a saber:

- Alumnos, por titulación: Arquitectura, Ingeniería Informática, Ingeniería de Telecomunicaciones e Ingeniería de la Edificación
- Profesores, por categoría: colaboradores, adjuntos, agregados y catedráticos.

Podemos asumir, entonces que hay 8 circunscripciones. Por las normas de estas elecciones, para cada circunscripción se eligen 2 representantes que serán los que acaben formando la Junta de Escuela, con 16 cargos electos.

La Universidad deberá elaborar un censo con los alumnos y profesores que tienen derecho a votar en las Elecciones, así como definir en qué circunscripción lo harán, para que tengan conocimiento de entre qué candidatos pueden elegir a sus representantes. De cara al sistema, es importante conocer estas divisiones, tanto para el conteo de los votos, como para la gestión de los candidatos en el momento en el que se presentan al votante.

Por tanto, es necesario tener un sistema que cargue el censo electoral elaborado por la Universidad, así como la definición de las circunscripciones y la relación entre estas y el propio censo de votantes.

### **3.3.1.3. Registro de votantes**

En muchos procesos electorales no existe un censo oficial elaborado por la Autoridad Electoral o alguna otra institución relacionada (como puede ser el INE en España). En ese caso, en multitud de estados se procede a una fase de registro en la cual se permite (en algunos casos, se obliga) a los ciudadanos a que se registren en un listado de votantes. Es el paso previo para poder votar. Una vez finalizada esta fase de registro, la Autoridad Electoral posee un censo *oficial* de votantes.

En el caso de las Elecciones dentro de la Escuela Politécnica, el censo lo proveerá la propia Autoridad Electoral a través de los datos de profesores, alumnos y empleados del

Centro. Al realizar la carga de estos datos para conformar el censo electrónico, el sistema tendrá conocimiento de qué potenciales votantes tendrán permiso o no para votar y qué opciones de voto deberá presentarles para que conformen su boleta electrónica.

Dependiendo del mecanismo digital de identificación y votación que se adopte para el sistema, la fase de registro puede ser tan simple como la correcta carga del censo en el sistema o puede aumentar ligeramente su complejidad. Si se decide utilizar una identificación basada en base de datos, habría que asignar a cada votante un nombre de usuario y una contraseña, al menos para ingresar en el sistema, ya que podría generarse otro par para la votación. En cualquier caso, en una situación como esta, además de la carga del censo resulta necesario una asociación de cada votante incluido en este con los nombres de usuarios y contraseñas generados para cada uno.

La Universidad proporciona a cada alumno, profesor y empleado un carnet universitario para su identificación. Entre otros servicios, estos carnets poseen la funcionalidad de identidad y firma digitales. Puede una buena opción hacer uso de estos servicios y evitar la asociación anterior, la cual, además del paso extra, no proporciona un nivel de seguridad aceptable. El servicio de esta tarjeta (TUI) o del DNIe español permite una identificación digital unívoca y confiable entre el votante y el sistema. Haciendo uso del servicio de firma digital, también se varía el esquema de votación del sistema, pues no requeriría de un módulo de generación de firmas electrónicas para votantes, pues cada uno llevaría el suyo propio en su TUI personal.

### **3.3.1.4. Presentación de las candidaturas**

Una vez definido tanto el censo como las divisiones electorales, tiene que afrontarse el problema de la presentación de las candidaturas que van a competir en la elección.

Para esta fase se pueden plantear diferentes soluciones en cuanto a la necesidad del proceso electoral. Desde un punto de vista simple, la propia Autoridad Electoral define y carga las diferentes candidaturas al sistema. Desde una visión más compleja, puede ser necesario la implementación de un sistema de gestión de candidaturas, como ya existe en elecciones de voto tanto electrónico como presencial, para que sean las propias candidaturas o partidos los que inscriban las listas o candidatos que les representen.

Para una elección como las que se quieren afrontar de la EPS, la primera visión, más simple, tiene muchas posibilidades de ser una solución adecuada. Sin embargo, pensando en un sistema para implantar en cualquier tipo de proceso electoral o, por ejemplo, pensando en elecciones a nivel municipal gestionadas por una autoridad a nivel estatal, donde concurren multitud de candidaturas, se hace imperioso desarrollar un sistema externo que realice la gestión de la inscripción de todas estas listas, estableciendo un plazo y mecanismos para que sean los propios partidos quienes lo hagan.

Esto último es lo que se realiza en las elecciones que se celebran en España, donde la Junta Electoral provee a los partidos de un sistema donde inscribir sus candidaturas, que la Junta puede gestionar, en caso necesario, dan posibilidades a los partidos a reclamar y generan el listado de candidaturas participantes en los formatos necesarios tanto para publicar en el Boletín Oficial del Estado o la Comunidad Autónoma como para los diferentes sistemas de escrutinio o postelectoral que se vayan a utilizar con respecto a la Jornada Electoral.

### **3.3.1.5. Generación de claves de encriptado**

Uno de los requisitos principales del voto electrónico se centra en la privacidad del mismo. El voto es secreto. Para satisfacer este requisito es necesario contar con herramientas que protejan el secreto de voto. Hay muchas esquemas de voto electrónico diferentes que pueden utilizarse para desarrollar un sistema de voto telemático, pero en la práctica totalidad de ellas será necesario generar al menos una clave criptográfica de cifrado.

En algunos esquemas, además de clave de cifrado hará falta clave de descifrado. Se puede considerar la generación de claves públicas y claves privadas si el esquema criptográfico del protocolo así lo requiere. En esquemas que se basan en el secreto compartido, se han de generar claves para los actores encargados de juntar la clave privada de la elección.

Por ejemplo, en Helios Voting (2.4.3), no se descifra cada voto, pero se requiere una clave para descifrar el resultado totalizado. Para ello, existe la figura de los trustees, que se reparten (comparten) esta clave (compartida) y, una vez totalizado el resultado de los votos cifrados, aportan su trozo para juntarla y utilizarla para descifrar el resultado.

A parte, habrá que generar claves para los certificados de servidor y cliente como herramientas seguras en las que basar la infraestructura del sistema.

## **3.3.2. Fase electoral**

### **3.3.2.1. Identificación del votante**

El primer paso de un votante a la hora de emitir su voto, en el sistema de voto tradicional, es identificarse ante los miembros de la mesa electoral. Para ello, en elecciones como las que organizan el Ministerio de Interior en España o las diferentes Comunidades Autónomas, el votante hace uso de un documento que verifique su identidad. En España, este documento es el DNI, aunque también se puede hacer uso del Pasaporte o carnet de conducir. En otros países en los que se carece de un documento oficial de identidad expedido por las autoridades del Estado, se realiza un registro biométrico de los votantes con, por ejemplo, las huellas dactilares de los mismos.

## Identificación

En el caso de las Elecciones a la Junta de Escuela de la EPS CEU, la identificación de los votantes se realiza de forma presencial, mostrando el documento acreditativo correspondiente a la mesa electoral.

Una vez identificado al votante, se le tiene que cotejar con el censo de la elección o de la mesa en la que ha sido identificado.

## Censo

En países como España, la elaboración del censo corre a cargo del INE y reparte a los votantes en diferentes mesas repartidas en locales electorales. En otros estados, este censo no existe y se requiere que sea la ciudadanía la que se registre en un Registro de Votantes, con lo que si no se ha acudido a tiempo de realizar este trámite, la persona pierde su derecho al voto.

En el caso de estudio de las elecciones de la EPS, este censo debe ser proporcionado por la propia Escuela.

## Votación

Para dejar constancia de que un votante ya ha ejercido su derecho al voto, en países como España es tan simple como que los miembros de la mesa electoral lo reflejen en una lista con el censo de su mesa. En otros territorios, sin embargo, la costumbre es marcar de alguna forma a aquellas personas que han votado, como puede ser manchar algún dedo de la mano con tinta indeleble, para que, si el votante intenta votar en otra mesa, se pueda comprobar que ya lo había hecho previamente y no pueda repetir.

En un sistema de voto por Internet no hay una interacción directa entre el votante y la autoridad electoral, que es quien debe permitirle votar. Por ello, es muy importante que los mecanismos para identificar al votante sean precisos y confiables. Con este objetivo, hay que valorar qué método de identificación es el mejor para cumplir con los requisitos de la elección y del voto electrónico telemático y remoto.

### ○ Usuario / contraseña

Para las elecciones de la Junta de Escuela de la EPS, el método de usar un par usuario / contraseña sería una solución sencilla. El censo está bastante acotado y, al ser todos los potenciales votantes miembros de la Universidad, poseen una cuenta de correo electrónico corporativa proporcionada por ésta. El proceso sería tan fácil como, por ejemplo, usar la dirección de correo electrónico de cada alumno / profesor / trabajador de la Escuela como nombre de usuario y enviarles un email a cada uno con una clave aleatoria generada por la autoridad electoral.

Esta solución, no obstante, sería inviable para elecciones más ambiciosas, como lo

son las legislativas estatales o autonómicas, ya que carecemos de elementos como direcciones de correo electrónico de todo el censo. Se podría utilizar el correo ordinario como método para hacer llegar estas credenciales, de la misma forma en que los partidos políticos hacen llegar la propaganda electoral o la Junta Electoral hace llegar la información del censo electoral a cada votante. Considero que sería un gasto extra de recursos económicos, humanos y medioambientales que no se sostiene para la utilización de este servicio. Tampoco se asegura la recepción del correo si aprovechamos el envío de la información del censo electoral, pues el envío, al contrario que cuando hemos solicitado el voto por correo y nos hacen llegar las papeletas, no es certificado. Realizar este envío de credenciales con garantía de recibo, resultaría muy costoso y lenta.

Otro motivo que desaconseja el envío de credenciales por correo es que éstas podrían ser interceptadas por otra persona distinta a quien identifican de forma no muy complicada, lo cual supone una brecha de seguridad bastante importante.

#### ○ Social Login

Ahora está muy extendido el concepto de login por Social Sign-in, que es un tipo de single sign-on que hace uso de la información de un usuario registrado en una red social para acceder a una web o servicio de terceros en vez de que estos implementen un sistema de cuentas de usuarios propia.

Este sistema delega el registro del usuario en la red social, simplificando el servicio de login de la web interesada, a la vez que puede obtener de la propia red social bastante información demográfica y personal, siempre que el usuario acceda a compartirlo.

Se trata de un servicio de autenticación y autorización. La red social autentica al usuario y

Redes sociales que permiten este servicio de login hay bastantes, destacando Google<sup>2</sup>, Facebook<sup>3</sup>, Twitter<sup>4</sup>, LinkedIn<sup>5</sup> o Yahoo!<sup>6</sup>, entre otros servicios.

En los enlaces de cada uno de los proveedores de este servicio que adjunto viene una explicación detallada para desarrolladores donde se puede consultar cómo funcionan y cómo se pueden implementar en las aplicaciones web de terceros.

#### ○ Mobile ID

El Gobierno de Estonia, para sus comicios por Internet está desarrollando una tecnología en la que el propio smartphone es la herramienta que sirve para identificarnos.

---

<sup>2</sup>Google Identity Platform <https://developers.google.com/identity/>

<sup>3</sup><https://developers.facebook.com/docs/facebook-login>

<sup>4</sup><https://dev.twitter.com/web/sign-in>

<sup>5</sup><https://developer.linkedin.com/docs/oauth2>

<sup>6</sup><https://developer.yahoo.com/auth/>

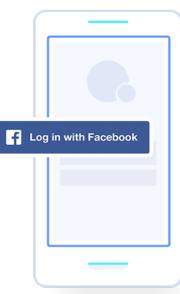


Figura 3.1: *Facebook es uno de los proveedores de servicio de Single Sign-In*

Mobiil-ID<sup>7</sup> es, según la web de la policía estonia<sup>8</sup>, una solución de identificación y firma digital para teléfonos móviles.

El principio de esta tecnología es asociar la identificación digital del ciudadano con una tarjeta SIM, utilizada para tener línea de teléfono o datos en teléfonos y dispositivos móviles. Esta SIM contiene los certificados digitales de autenticación y firma apropiados para asegurar la identidad digital del ciudadano de la misma forma que lo haría su carnet de identidad digital.

Parece una buena opción, pues hoy por hoy, es bastante común que llevemos el smartphone con nosotros de la misma forma que llevamos el DNI. De cara a un proceso electoral, es un dispositivo muy personal, bastante asociado al día a día de su dueño, por lo que es una herramienta con un gran potencial para proporcionar un servicio de identificación única entre el usuario-votante y su registro en el censo electoral o contra los administradores del proceso.

Otra ventaja es que, al estar los certificados en la SIM, nos ahorraremos el hardware necesario para la lectura de certificados externos, como son el DNIe 2.0 español o las SmartCards tradicionales. Incluso no haría falta hardware de lectura por radiofrecuencia como necesitan algunas SmartCards modernas o el propio DNIe 3.0, con lo que se puede ahorrar en hardware interno del dispositivo.

Lamentablemente, es una tecnología que en estos momentos no se implementa a gran escala en España y no hay visos de que ocurra en un futuro próximo. Motivo suficiente para descartar esta opción.

- **Smartcard**

Otra opción posible es el uso de una smartcard que contenga certificados emitidos por la Autoridad Electoral para cada votante. Los inconvenientes de este método son varios:

<sup>7</sup> <http://www.id.ee/index.php?id=36881>

<sup>8</sup> <https://www.politsei.ee/en/teenused/isikut-toendavad-dokumendid/mobiil-id/>

- Por un lado, requiere un registro previo de los votantes, pues hay que generarles los certificados.
- Un problema logístico ya que, una vez generados los certificados e introducidos en las tarjetas, éstas deben hacerse llegar a los votantes que las van a utilizar. Este paso, en unas elecciones a gran escala puede suponer un esfuerzo injustificado.

- **DNIe**

Lo ideal para una elección por el sistema de voto por Internet es implementar un proceso que resulte sencillo al votante, ya que si resulta ser más complicado que el voto tradicional, el votante no le verá sentido y no hará uso de él.

Con este planteamiento, parece que el uso del DNIe es una buena idea. Por un lado, es un documento oficial que llevamos normalmente con nosotros en todo momento. Además es el mismo documento que nos identifica en las elecciones tradicionales, con lo que para el votante no debería suponer ningún trauma, al estar completamente insertado en la sociedad su uso para este cometido (asumimos en este supuesto que la implantación del DNIe en España es casi completa, que el votante ya no necesita acudir a una comisaría a solicitarlo y que los certificados no están caducados).

Ventajas del uso del DNIe como identificador del votante:

- Documento expedido por las propias Autoridades del Estado, quienes lo avalan.
- Seguridad.
- La gente lo lleva consigo constantemente y está acostumbrada a usarlo para identificarse o, incluso, para realizar otro tipo de actividades en Internet, como obtener certificados de Organismos Públicos, banca por Internet, etc.
- Es el mismo documento que ya se utiliza para identificarse en las elecciones presenciales tradicionales.

Inconvenientes del DNIe:

- Extranjeros con derecho a voto pueden no tener DNIe, pero deberían poder votar con el pasaporte.
- Certificados caducados. Los certificados que incluye el DNIe tienen una fecha de caducidad a partir de la cual son revocados y no tienen validez legal. Esta fecha no se corresponde con la caducidad del documento físico, que está impresa en el anverso del documento. Esta discordancia de fechas considero que es un punto débil del DNIe, pues es muy posible que el usuario no renueve los certificados simplemente porque no sepa que han caducado, dado que el propio documento físico no lo ha hecho.

- Rotura del chip que contiene los certificados.
- Limitaciones técnicas para las aplicaciones web. En el estado actual de la tecnología, es necesario hacer uso de un applet de Java para poder firmar con el DNIe. De cara a la identificación, ya hay software Javascript que se salta este paso, aunque no a la hora de firmar, para lo cual, hoy por hoy, no hay alternativa. Este detalle es una limitación importante, quizás no para el voto electrónico, pero sí para el voto ubicuo por Internet, ya que requiere de más tecnología que simplemente un dispositivo conectado a Internet y un lector. Además, el uso de applets está cada vez más limitado en Internet, en desuso, y se recomienda implementar alternativas basadas en el estándar W3C. Por desgracia, este organismo todavía no tiene definido de una manera versátil cómo afrontar el problema de la criptografía con certificados digitales personales en los nuevos estándares web.
- Necesidad de HW externo, como son los lectores de Smartcard. Para poder utilizar el DNIe como identificador, el sistema tiene que poder leer los datos que le indica. Si hacemos uso de los certificados que contiene, necesitamos un lector externo, lo cual quizás no sea un problema si usamos un PC que tenemos en casa, pero sí que puede serlo cuando queremos votar desde otro ordenador o incluso desde un dispositivo móvil, donde ya no es tan simple que tengamos este lector y que sea compatible. Ciento es que podríamos hacer uso de la banda MRZ del documento escaneándola pero solamente con este dato no se puede mantener un nivel aceptable de seguridad. Por ejemplo, se podría identificar un usuario con una fotocopia de un DNI, lo cual en unas elecciones oficiales en España no está permitido.

A partir de enero de 2015, el Ministerio del Interior de España, a través de la Dirección General de la Policía, empezó a emitir una nueva versión del DNIe, denominada 3.0, la cual, entre varios avances, posee, junto al chip electrónico, otro chip de radiofrecuencia basado en tecnología NFC.

Sin entrar en los posibles problemas de seguridad que implica el uso de chips de radiofrecuencia, se observa interesante este avance en cuanto a una aplicación de voto por Internet. Hoy en día son muchos los dispositivos móviles (smartphones y tablets) que disponen de un lector de chips por NFC, con lo que el inconveniente de necesitar lectores externos para identificación del votante y firma digital del voto ya no existiría. El votante podría votar desde una app instalada en su smartphone sin necesidad de elementos externos, ya que este chip NFC contiene los mismos certificados que el chip por contacto que existía hasta ahora en la primera versión del DNIe.

Para este proyecto vamos a utilizar el login con DNIe pues creemos que es una buena prueba de concepto para demostrar la viabilidad de esta tecnología como futuro del voto por Internet en España.

Posee certificados de identificación y firma. El primero es esencial a la hora de identificar a un votante de forma unívoca. El certificado de firma es fundamental en procesos electorales que incluyan, por ejemplo, procesos basados en protocolos de firma ciega, pues pueden firmar con seguridad documentos digitales, manteniendo secreto e identidad.

### **3.3.2.2. Votación**

En el sistema tradicional, el momento de la votación es aquel en el que el votante deposita su voto en la urna tras haber escogido la papeleta o marcado la boleta de candidatos y haber sido identificado correctamente por los miembros de la mesa electoral.

Este proceso es al que estamos habituados en los territorios con una cierta historia democrática. En principio, parece bastante transparente, en cuanto a que el votante puede confirmar sin ninguna duda que su voto, efectivamente, se encuentra dentro de la urna sellada, junto con el resto de votos de la mesa.

Aquí encontramos el primer detalle controvertido con respecto al voto por Internet. El votante no tiene constancia física de que su voto se ha depositado en la urna correcta, ni siquiera de si está en alguna urna. No "se ve".

Es más, sabe que ha introducido en la urna la papeleta que tenía en su mano, que sabe cuál es porque él mismo la ha elegido. Pero en el sistema informático, no sabe si ocurre lo mismo. Puede pensar que aunque haya seleccionado un candidato y el sistema le diga que ha contabilizado su voto por éste, realmente, por detrás esté cambiando el voto y registrando a otro candidato diferente.

Es misión del sistema informático proveer al votante de mecanismos que le permitan verificar todas estas cuestiones. Hay que diseñar el sistema para que haya confianza en él. Quizá esta sea la mayor de las barreras existentes en la actualidad para la implantación del voto por Internet, la falta de confianza.

No es por falta de métodos seguros o carencia de medios criptográficos. El problema es que no es fácil que el elector, opinión pública u organismos de control o auditoría confíen en el proceso, ya que, a priori, parece una gran caja negra, ante la cual es complicado asegurar una verificación de datos de forma transparente.

### **3.3.2.3. Totalización de resultados**

La totalización es el proceso en el que se suman los resultados recogidos en cada mesa electoral y se agrupan estas agregaciones de votos en los diferentes niveles que existen "por encima".

En los procesos electorales tradicionales, esto resulta ser, por ejemplo, que se suman

los resultados de cada una de las mesas de un local y se obtiene el resultado de ese colegio. A continuación se totalizan los resultados de todos los colegios de un distrito para obtener los resultados de este. Los resultados del distrito se totalizan para obtener los de una localidad, estos para llegar a los de una comarca, a continuación provincia o circunscripción, comunidad autónoma y los estatales.

En nuestro sistema de votación remota, la totalización consistirá en sumar, para cada “circunscripción lógica” los votos emitidos por cada votante.

La idea es utilizar un protocolo de cifrado homomórfico aditivo, con lo que la totalización se realizaría sobre votos cifrados y sería el resultado de esta el que se descifre. Así se asegura la privacidad del votante a la vez que se le permite votar tantas veces como desee, para reducir el peligro de la coacción.

### **3.3.3. Fase postelectoral**

#### **3.3.3.1. Difusión de resultados**

La difusión de resultados trata de dar visibilidad a los resultados del escrutinio y totalización de los votos de la elección.

En los procesos electorales tradicionales es un proceso esencial en cuanto a la transparencia del mismo. Se intenta que la difusión sea casi en tiempo real. Básicamente, cuanto antes se muestre el resultado de escrutar una mesa, menor será la probabilidad de que los interesados en la elección tengan la sensación de que haya podido haber alguna manipulación de los resultados. Igualmente, en España, estos resultados de la noche electoral son provisionales, por lo que carece de sentido una exhausta demostración de su correctitud, ya que los fallos que haya en el conteo se resolverán unas horas o días más tarde en el recuento definitivo, llevado a cabo públicamente por un juez de paz.

En el caso de nuestro sistema de voto remoto, lo normal será que la difusión se haga con el 100 % escrutado, ya que se totalizarán todos los votos al mismo tiempo, por lo que este efecto de transparencia no podrá tenerse en cuenta. Para ello, habrá que confiar en los procesos criptográficos diseñados para salvaguardar la integridad del escrutinio de los votos y a las auditorías que lo demuestren.

La difusión de resultados es la forma en la que se pone en conocimiento de los interesados en el proceso electoral el resultado de éste. Por tanto, es muy importante que se diseñe con el objetivo de alcanzar el mayor número de personas.

Por eso, es interesante que el número de canales de acceso a esta información sea amplio, a la vez que cumpla con estándares de accesibilidad.

Además de publicar en web o en dispositivos móviles a través de aplicaciones nativas,

resulta muy interesante pensar en la publicación de resultados a través de una API para que terceros puedan realizar su propia difusión. Esto es interesante para medios de comunicación, por ejemplo, las televisiones o periódicos online, que pueden conectarse a la API y realizar sus propios gráficos multimedia para cubrir en directo la noticia del avance del escrutinio. Es lo que ocurre ahora mismo con las elecciones en España.

### 3.3.3.2. Verificación de resultados

Tras el cierre de la elección, totalización y difusión de resultados, es necesario implementar un sistema que permita a los votantes verificar la integridad de su voto y a los observadores verificar la integración de la elección.

Hay que hacer uso de protocolos criptográficos de pruebas de conocimiento cero para poder llevar a cabo estas verificaciones de forma efectiva y segura.

## 3.4. Descripción

Tras el estudio de cómo debe ser un sistema de voto por Internet, aparecen cuestiones importantes como puede ser el esquema criptográfico a utilizar, el flujo de procesos, actores intervenientes, fases, problemas asociados al voto electrónico, seguridad, confiabilidad, etc.

Teniendo en cuenta estos problemas, se ha valorado el uso de un sistema desarrollado desde cero, adecuado a las necesidades de la Elección de la Junta de Escuela que se quiere realizar. Frente a este escenario, otra opción es la de basarse en alguna herramienta existente y, si es necesario, realizar una adaptación de la misma para cuadrar con los requisitos de este proceso electoral.

El hecho de que este tipo de sistemas deban tener una base criptográfica muy importante es uno de los motivos principales para optar por la adaptación de alguna solución ya existente en lugar del desarrollo. En el estudio de soluciones en el mercado encontramos que, aunque no existen realmente muchas opciones libres que reutilizar, sí que existe un verdadero estudio teórico del problema del voto digital. Entre las opciones que han decidido desarrollar las propuestas teóricas hay algunos que merece tener en cuenta, como son Helios Voting (2.4.3), Agora Voting (2.4.2), Adder (2.4.1), el sistema electoral de Estonia (2.2.1), junto con otras opciones que no son E2E verificables o no tratan el voto remoto por completo, como pueden ser Votescript, Punch&Vote u otros sistemas publicados.

De entre todas estas opciones, las más atractivas para el tipo de elecciones que se quieren desarrollar en este proyecto resultan ser las propuestas por Ágora Voting y Helios Voting.

Helios Voting es un sistema de voto E2E verificable, basado en un esquema criptográfico

homomórfico, utilizando pruebas de conocimiento nulo para verificar tanto el voto individual de cada votante como el correcto conteo de la elección. En el sistema hemos visto que no es necesario desencriptar los votos para realizar el escrutinio, sino que se realiza con los votos encriptados y, posteriormente, lo que se descifra es el cifrado del resultado del mismo. El desarrollador de Helios Voting se ha apoyado en gente bastante reputada en el campo de la seguridad informática, la criptografía y el voto digital por lo que el apartado criptográfico del sistema está suficientemente probado y documentado que cumple con los requisitos del voto electrónico.

Aunque el peso de la criptografía la estamos delegando en el sistema elegido, en este se observan una serie de elementos que no se ajustan a la primera idea sobre el sistema desde cero.

El esquema criptográfico utilizado por Helios es homomórfico. En una primera versión apostaban por un esquema basado en mixnets, análogo al utilizado en el desarrollo del sistema estonio. El uso de una mixnet resultaba bastante interesante al pensar en la identificación del votante con un soporte que posee certificados digitales, como el DNIe, que permitiría identificar al votante y que este pudiese firmar su voto. Así, utilizando un protocolo de firma ciega, podría asegurarse la identificación unívoca del votante, su elegibilidad para votar y si ya ha ejercido su voto con anterioridad. A partir del momento en el que se cierra la votación y comienza el escrutinio, sería muy importante desacoplar el dato del votante del voto emitido, momento en el que entra en juego el esquema de mixnets, ideal para llevar a cabo este proceso de desacoplamiento votante-voto.

Aunque en la primera versión de Helios el esquema utilizado fue el de mixnets, a partir de la siguiente versión se cambió al homomorfismo. El futuro a partir de la versión 3 era implementar ambas soluciones en el sistema, pero al final se decidió no continuar con la integración de la mixnet.

El sistema de identificación actual de Helios es el de usuario/contraseña o el uso de protocolos OAuth contra entidades externas, como Google, Facebook, Yahoo! o Twitter. En el caso de las Elecciones a la Junta de Escuela, parece muy interesante la introducción de los certificados digitales, aprovechando el esfuerzo de la Administración al desarrollar una nueva versión del DNI electrónico, con chips sin contacto NFC, lo que puede suponer una oportunidad para el uso de certificados digitales desde dispositivos móviles. No obstante, eliminando el elemento innovador del nuevo chip del DNIe, observando casos de éxito del sistema Helios en elecciones reales, los organismos en los que se ha utilizado han desarrollado sus propios módulos de identificación de votantes. En concreto, en las elecciones de la Universidad de Louvain, se desarrolló el subsistema de identificación para que fuese compatible con las credenciales que los alumnos utilizaban para acceder a los servicios de la propia Universidad.

Por tanto, para las Elecciones de la EPS, sería una buena solución el desacoplar el

módulo de identificación de votante y desarrollar uno nuevo que se comunique con la base de datos del censo de la Universidad. Para esto, se puede implementar, como en las elecciones de la Universidad de Louvain, un servicio de usuario/password y continuar con el sistema Helios existente. También se podría desarrollar un servicio de autenticación OAuth siendo la propia Universidad la que proporciona los tokens. Pero, como se ha avanzado, se va a implementar un sistema de autenticación basado en las credenciales proporcionadas por el DNIe.

Un primer cambio importante que se propone para el sistema Helios es que la gestión del censo electoral pueda externalizarse. Actualmente, el sistema guarda en su propia base de datos la lista de votantes que pueden votar. De hecho, por el carácter general del sistema, esto es así para que el administrador de la elección pueda cargar el censo de votantes a través de ficheros CSV, separados por comas. Para el cometido del proyecto, sería necesario modificar este sistema censal.

Modificaciones sobre Helios:

- Capar el sistema para que no permita el voto a cualquier votante, sino solamente a aquellos votantes que aparezcan en el censo.
- Externalizar el listado de votantes para que no lo gestione Helios, sino el organizador de la Elección.
- Cambiar los métodos de autenticación para que el permitido sea el DNIe.
- Modificar la interfaz web del sistema para adecuarla a las Elecciones a la Junta de Escuela de la EPS.

The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards - and even then I have my doubts.

---

Gene Spafford<sup>1</sup>

## Capítulo 4

# Riesgos

### 4.1. Identificación y gestión de riesgos

Miembros de una conocida empresa española líder en procesos de voto electrónico publicaron un artículo de buenas prácticas al implementar un sistema de voto electrónico por Internet. En el mismo exponen una lista de riesgos generales de seguridad inherentes al voto electrónico. Su intención era usarlos como referencia para poder comparar diferentes sistemas de voto sin tener en cuenta la tecnología que los implementen.

En este PFC van a tenerse en cuenta de cara al diseño de un sistema robusto de voto por Internet.

- **Votos por parte de votantes sin autorización** El sistema de voto debe poseer un mecanismo robusto y confiable para identificar correctamente de forma remota a los votantes, ya que personas sin autorización podrían intentar emitir su voto.
- **Suplantación del voto** Un votante o un atacante podrían intentar suplantar la identidad de un votante autorizado para votar en su lugar. El sistema debe proporcionar un mecanismo que detecte este tipo de intentos de suplantación.
- **Inyección de votos** El sistema debe prevenir la aceptación de votos *inyectados*. Un atacante puede intentar introducir en la urna votos de votantes que no han participado en el proceso electoral (por ejemplo, por abstención) y que, por tanto, no deberían contabilizarse.
- **Privacidad del voto comprometida** : Un atacante podría intentar quebrar la privacidad del voto de un votante, identificando al mismo con su opción elegida, con lo que se pierde el requisito del derecho al voto secreto. El sistema debe implementar

---

<sup>1</sup>Profesor en Purdue University y experto en seguridad informática [https://en.wikipedia.org/wiki/Gene\\_Spafford](https://en.wikipedia.org/wiki/Gene_Spafford)

mecanismos que eviten completamente que, durante cualquier fase del proceso, la intención de voto de cualquier votante pueda dejar de ser secreta.

- **Coerción y compra de votos** Una persona u organización puede comprar a un votante u obligarle a votar por una candidatura específica. El sistema de voto debe evitar que un votante pueda probar a un tercero su intención de voto de forma irrefutable.
- **Modificación del voto** Los votos emitidos pueden ser modificados para cambiar el resultado de la elección. El sistema debe detectar cualquier manipulación en los votos válidos ya emitidos.
- **Borrado de votos** Relacionado con el anterior, un atacante podría intentar borrar votos que ya han sido emitidos. La urna debe estar protegida ante cambios no autorizados, como puede ser un intento de borrado.
- **Publicación de resultados intermedios no autorizados** Los resultados intermedios podrían ser divulgados antes del cierre de la elección, con lo que se puede influir en los votantes que todavía no hayan emitido su voto. El sistema debe preservar el secreto de los votos sufragados hasta el proceso de escrutinio y evitar la difusión de resultados parciales antes de la finalización del periodo de votación.
- **Desconfianza del votante** Un votante puede no tener ningún medio para verificar la correcta recepción y cuenta de su voto por parte del sistema. Debido a esto, el votante podría desconfiar del proceso. El sistema debe permitir al votante verificar si su voto ha sido correctamente recibido por el sistema y si ha sido incluido en el proceso de escrutinio con la opción con la que fue emitido.
- **Ataque DoS** Un atacante podría interrumpir la disponibilidad del canal de votación realizando un ataque DoS (*Denial of Service - Denegación de Servicio*). El sistema debe detectar una eventual congestión de los servicios de votación para poder reaccionar tan pronto como sea posible y evitar una caída de los mismos que no permita a los electores sufragar su voto.
- **Auditoría no fiable** Una insuficiente trazabilidad de los eventos de la elección o una manifiesta facilidad para modificar los datos auditables puede permitir a un atacante esconder cualquier comportamiento no autorizado en el sistema. El sistema debe proporcionar medios para implementar un proceso de auditoría que permita detectar cualquier manipulación de estos datos.
- **Fecha límite** Un problema de este proyecto es el tiempo. No se trata de un desarrollo continuo en el cual se pueden ir desarrollando versiones que, una vez puestas en producción, pueden ser actualizadas para corregir errores o añadir funcionalidades. En este caso, el sistema tiene un barrera temporal claramente definida y que, en ningún caso, puede ser traspasada. El proceso electoral tiene unos tiempos establecidos a

base de hitos predefinidos. A pesar de la existencia de hitos en la fase preelectoral, la importancia del sistema radica en que el día (o durante el período definido) de la Jornada Electoral debe estar en producción, totalmente funcional y lo más depurado posible para evitar prácticamente todos los fallos que puedan ser estimados. Por tanto, la variable temporal de este proyecto conlleva un riesgo extremadamente importante, ya que no tenerlo en cuenta y no cumplir los plazos establecidos conduce a la imposibilidad de uso del sistema y, por tanto, al fracaso del proyecto.

- **Errores en software / Caída del sistema** Los errores de software para un sistema de voto electrónico pueden resultar catastróficos. Dependiendo del tipo de aplicación, un error software puede ser más o menos grave, más o menos subsanable. Dependiendo del diseño del sistema, puede suponer desde un conteo incorrecto hasta la imposibilidad de realizar el escrutinio. Por ejemplo, los sistemas que se utilizan para realizar el escrutinio provisional de las elecciones legislativas en España se basan en un conteo manual de las papeletas sufragadas, una comunicación digital de los datos contados en cada mesa y, a continuación, el escrutinio de estos datos. En caso de un error de software o caída del sistema y podría resultar imposible realizar el escrutinio provisional. Aún siendo un fracaso del proyecto, puede recuperarse la información, que está en las actas físicas de cada mesa, para realizar el escrutinio definitivo de forma manual, por lo que esta incidencia no afecta a la elección como para que no pueda ser llevada a cabo. Si en el diseño del sistema para las elecciones en la EPS, si no se usa ningún tipo de urna, el voto sería completamente digital, puede ocurrir que, en caso de caída del sistema o de fallo general, sin posibilidad de recuperación de la contingencia, sea imposible realizar un escrutinio manual alternativo. Aquí es donde se puede valorar el uso de otros sistemas y la conveniencia de realizar un voto por Internet *puro*, ya que el riesgo que conlleva en cuanto a la falla grave del sistema es bastante importante y puede llevar al fracaso de la elección y/o a su repetición.



Una nación sin elecciones libres es una nación sin voz, sin ojos y sin brazos.

---

Octavio Paz<sup>1</sup>

## Capítulo 5

# Análisis del sistema

Una vez ha sido presentado el proyecto, planteados los objetivos y estudiado el estado de la cuestión, la siguiente fase en el desarrollo es el Análisis. En esta etapa se analiza el problema con el cliente, llegando a un acuerdo en el alcance del proyecto y los requisitos que deben ser satisfechos.

### 5.1. Especificación de requisitos

#### 5.1.1. Introducción

En esta sección de la memoria vamos a desarrollar la especificación de requisitos de software. Con esta técnica lo que se consigue es una descripción completa del sistema que se va a desarrollar.

Los requisitos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan las necesidades de los clientes para resolver un problema.

En este proyecto, los requisitos se han dividido en cuatro grupos:

- **Funcionales** : Son los requisitos que el sistema debe cumplir para su correcto funcionamiento. Son requisitos fundamentales de cara al usuario, ya que responden a la pregunta *¿qué hace?*, por lo que implican directamente en la funcionalidad que el sistema proporciona al usuario.
- **No funcionales** : Usualmente son los requisitos que responden a la pregunta *¿cómo lo hace?*. Definen las necesidades de recursos para el funcionamiento del sistema, como protocolos, infraestructura, tecnología...

---

<sup>1</sup>Poeta, ensayista y diplomático mexicano, Premio Nobel de Literatura en 1990 [https://es.wikipedia.org/wiki/Octavio\\_Paz](https://es.wikipedia.org/wiki/Octavio_Paz)

- **Del proceso electoral :** En diferentes bibliografías se corresponden con los requisitos organizacionales. Recogen las necesidades que el sistema debe cumplir con el marco contextual, en este caso, las propias del proceso electoral.
- **Del voto electrónico :** Son los requisitos implícitos de los sistemas de voto electrónico, introducidos en el capítulo 2.1.3.

### 5.1.2. Ámbito del sistema

El sistema que se va a desarrollar tiene como finalidad que los alumnos, profesores y empleados de la Escuela Politécnica Superior de la Universidad San Pablo CEU puedan votar remotamente en las Elecciones a la Junta de Escuela.

Para ello, el sistema debe ser distribuido y dirigido hacia un entorno web, permitiendo que los votantes puedan emitir su voto desde cualquier lugar donde tengan conexión a Internet. Con el auge de los teléfonos móviles inteligentes y de las redes móviles (GPRS, 3G, 4G, HSDPA...), se puede permitir la votación desde cualquier lugar en la que el proveedor de telefonía móvil provea de cobertura al votante.

Este enfoque ubicuo del voto debe asegurar, no obstante, las mismas garantías que proporciona un sistema de voto tradicional o uno de voto electrónico presencial.

El sistema dispone, por un lado, del servidor web que permite la interacción tanto con el votante como la difusión de los resultados electorales. Este servidor web tendrá comunicación con Internet y debe tener una seguridad acorde con el nivel de peligro ante ataques que se espere para este tipo de elección.

Por otro lado, la parte de la lógica de negocio del sistema se basa en subsistemas independientes, aunque modulares, dependiendo del servicio que tengan que ofrecer. La independencia de estos subsistemas responde a cuestiones de seguridad y para proporcionar mayor transparencia del proceso en cuanto al *flujo del voto* a través del software.

Adicionalmente, el sistema debe proporcionar funcionalidades externas al proceso de voto.

Debe tener un sistema de carga y gestión de los datos electorales (reglas de la elección, urnas, circunscripciones, candidatos, censo).

Debe proporcionar herramientas que permitan la monitorización del proceso para comprobar el estado del mismo, además de herramientas de transparencia para que pueda ser auditado en tiempo real por las autoridades competentes.

### 5.1.3. Requisitos funcionales

- **Votación por Internet** El sistema de votación debe funcionar de forma remota en sus fases de registro, identificación, votación y consulta de resultados. Cualquier votante puede acceder a las funcionalidades del sistema a las que tiene autorización desde cualquier punto conectado a Internet.
- **Permitir votación presencial** El sistema debe proporcionar los mecanismos necesarios para permitir el voto a aquellos votantes con derecho al mismo que quieran emitirlo de forma presencial en el periodo habilitado para ello.
- **Disponibilidad total** El sistema debe estar disponible para proporcionar servicio de voto durante todo el periodo estipulado en las normas que se fijen para la elección.
- **Identificación remota** El sistema debe implementar los mecanismos y adoptar las tecnologías necesarias para asegurar la identificación de un votante en el sistema de forma digital y remota.
- **Autenticación remota** El sistema debe poder autenticar a los votantes que tratan de usar su identificación digital para ingresar en el sistema de forma remota. El sistema no debe errar en esta autenticación, permitiendo la entrada de los votantes autorizados y revocando el acceso a los atacantes, suplantadores o desautorizados.
- **Papeleta/boleta digital** El sistema debe mostrar al votante la papeleta o boleta (dependiendo del tipo de elección) correspondiente a la elección y el censo que le corresponda. Debe contener las opciones correctas por las que puede optar y mantener correctamente la/s opción/es seleccionadas.
- **Voto anónimo** El sistema debe poder romper la relación existente entre el voto y el votante. Deben desarrollarse los protocolos criptográficos y de infraestructura necesarios para que nadie pueda vincular el contenido del voto a un votante determinado.

### 5.1.4. Requisitos propios del voto electrónico

A partir de los requisitos implícitos al voto electrónico presentados en el capítulo 2.1.3, consideramos los siguientes requisitos como los implícitos al voto electrónico:

- **Autenticidad** : Sólo los votantes autorizados pueden votar. La autorización de un votante para ejercer su derecho al voto está expresada en el censo electoral conformado por la Autoridad Electoral competente. El sistema debe comprobar que el votante que quiere realizar un voto debe estar inscrito correctamente en el censo electoral y que en éste no se indique que tiene vetada su participación en el proceso.

- **Anonimato** : El voto es secreto. Ningún votante, observador o manipulador del sistema puede tener la habilidad o herramienta de poder conocer el voto que ha sufragado otro votante en ningún momento del proceso electoral.
- **Verificabilidad** : El votante puede asegurarse de que su voto se ha contado adecuadamente. El sistema tiene que proporcionar una herramienta que permita a un votante poder verificar que la opción por la que ha votado ha sido correctamente añadida a los resultados consolidados del proceso electoral, sin que por ello pueda violarse el requisito de Anonimato, asegurando que ningún actor del sistema pueda tener acceso al contenido de dicho voto.
- **Imposibilidad de coacción** : El sistema debe evitar que el voto emitido pueda ser mostrado a un tercer actor con el fin de evitar la coacción al votante o la venta del voto por/a un tercero.
- **Fiabilidad** : El sistema debe asegurar que no se producen alteraciones de los resultados. Es esencial que el sistema asegure que, aunque existan riesgos inherentes a cualquier sistema informático, estos no van a afectar los resultados del proceso electoral.
- **Auditabilidad** : Se debe poder comprobar que el funcionamiento de los elementos que intervienen en el proceso es correcto. Para favorecer la transparencia del proceso, es muy importante que el sistema proporcione unas herramientas que permitan tanto la monitorización del proceso como auditorías del mismo. Estas herramientas deben ser fiables y demostrar tanto su correcto funcionamiento como el del proceso electoral en si mismo a una serie de actores designados (operadores, auditores, observadores).
- **Usabilidad** : Cualquier votante debe ser capaz de emitir un voto en un tiempo razonable. El sistema debe ser usable y accesible, debe facilitar el proceso de emisión de voto a prácticamente la totalidad del electorado.

#### 5.1.5. Requisitos del proceso electoral

- **Fechas límite** : El proceso electoral marca un requisito crucial en el funcionamiento del proyecto, las fechas límite operacionales. Desde la autoridad electoral, se deben definir en el protocolo o las Normativa Electoral de la Universidad una serie de fechas en las que se deben cumplir hitos del proceso electoral.
  - \* **Fechas de inicio y fin de la carga del censo**
  - \* **Fecha de apertura de mesas** : Una vez convocada la elección, habrá un día definido para llevar a cabo la misma. La autoridad electoral debe fijar un día y una hora en la que el sistema ha de estar abierto y disponible para que la totalidad del censo pueda votar.

- \* **Fecha de cierre de mesas :** Igualmente, el sistema debe cerrar las mesas en la fecha y hora estipuladas. A partir de este momento, se ha de asegurar que ningún votante puede votar o modificar ningún voto ya emitido.
- \* **Fecha de inicio de escrutinio :** Normalmente, el escrutinio comenzará a continuación del cierre de mesas. El sistema debe estar preparado para que los miembros de la Autoridad Electoral den comienzo al escrutinio a través de la interfaz del sistema.
- \* **Fecha de publicación de resultados :** El sistema debe permitir que los resultados sean publicados y difundidos en el momento en el que la Autoridad Electoral lo requiera, ya sea al término del escrutinio o con posterioridad.
- **Acreditación :** Con el carácter de prototipo de este proyecto, el sistema permitirá el voto solamente con la acreditación a través del DNI español, ya sea 2.0 ó 3.0, no estando permitido el voto remoto a través de este sistema con ningún otro tipo de acreditación, ya sea digital o no.

#### 5.1.6. Requisitos no funcionales

- **Bajo coste :** El coste del sistema debe ser relativamente bajo. Teniendo en cuenta las características del cliente, una Escuela o Universidad y los potenciales desarrolladores del proyecto, estudiantes, es importante procurar que el espíritu del sistema a implementar se base en el bajo coste. Con este principio, se pueden ahorrar recursos económicos a la institución que podría destinar a otras necesidades. Por ello, se considera importante promover la utilización de software libre o sin licencia, así como reducir el número de responsables en la gestión del proceso electoral a nivel de control del sistema informático y la utilización de tecnologías hardware que reduzcan el desembolso de capital, como es la utilización de software de virtualización frente a la inversión en máquinas, mucho más costosas.
- **Soporte a usuarios :** El sistema debe proporcionar unas herramientas de soporte a usuarios de cualquier rol que hagan uso del mismo. Al ser un sistema de votación nuevo, diferente al tradicional y con una característica tecnológica por medio, es fundamental que se ofrezca soporte, tanto técnico como de procedimiento a los usuarios del sistema en la jornada electoral, ya sean votantes activos como responsables de la autoridad electoral, auditores u observadores del proceso. Hay que procurar que puedan disponer de la información y el soporte necesario para que todos puedan realizar sus funciones durante la jornada electoral sin problemas o, al menos, minimizándolos.
- **Accesibilidad :** El sistema debe ser accesible a la mayor cantidad de votantes potenciales posible. Debe cumplir con las especificaciones de accesibilidad marcadas por los organismos reconocidos (W3C, WCAG...) y tratar de que los votantes con alguna

discapacidad física no tengan problemas funcionales y de comprensión a la hora de elegir, emitir y verificar su voto.

### 5.1.7. Requisitos específicos

**RE.1** La votación debe permitir al votante el voto en blanco.

**RE.2** La difusión de resultados debe añadir la información de votos en blanco, participación y censo. Los dos últimos datos de este requisito se pueden mostrar asumiendo que las elecciones sean de censo cerrado.

## 5.2. Actores

Considerando el flujo del votante en el sistema, identificamos seis actores/roles en el sistema que deben ser tenidos en cuenta de cara a las funcionalidades, privilegios y responsabilidades que tienen que encontrar en el uso del mismo.

En este sistema identificamos 6 actores que interactúan con el mismo en algún momento del proceso (figura 5.1):

- Votante
- Administrador
- Miembro de la Junta Electoral
- Auditor
- Autoridad Certificadora
- Trustee

### ● Votante

El votante es el actor principal del sistema, pues es al que va dirigido el proceso de votación. Es el único rol que tiene la capacidad de votar. Las acciones que puede realizar son:

- Consultar su presencia en el censo.
- Identificarse unívocamente en el sistema.
- Elegir la papeleta con su voto.
- Capacidad de poder emitir un voto nulo.

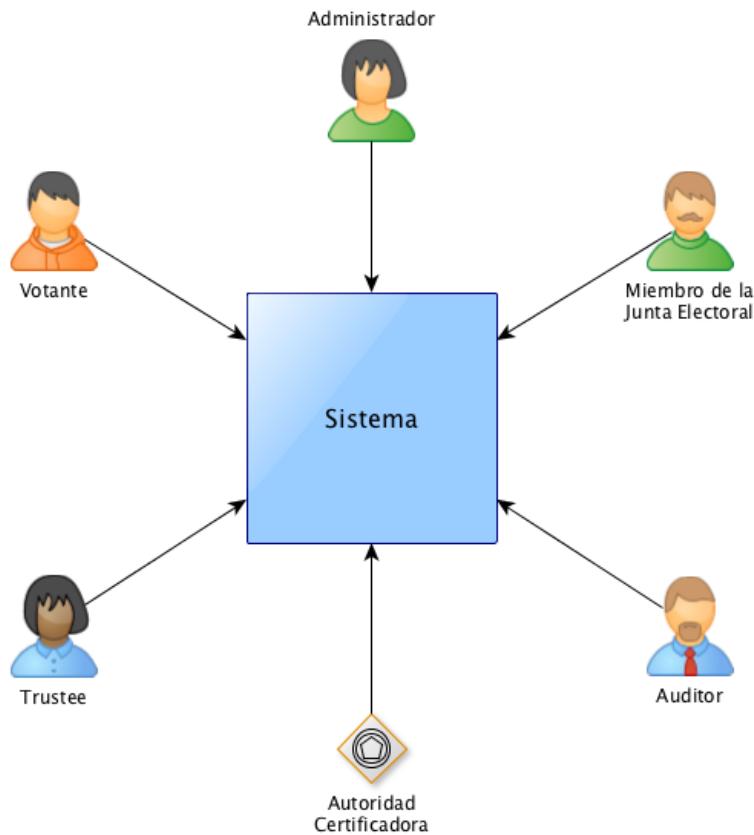


Figura 5.1: Actores/roles que interactúan con el sistema.

- Firmar el voto.
- Votar preservando el carácter anónimo del voto.
- Consultar que su voto ha sido contabilizado.
- Audituar el correcto escrutinio de la elección.

#### • Administrador

El administrador es el rol encargado de gestión de las fases electorales. Tiene responsabilidad y potestad de:

- Iniciar el proceso electoral.
- Iniciar el proceso de votación.
- Terminar el proceso de votación.
- Apertura de la urna
- Inicio del escrutinio
- Apertura de los canales de difusión de resultados.
- Finalizar el proceso electoral.
- Designar los trustees de la elección.

Los usuarios con este rol no puede votar. El administrador de la elección no forma parte del censo de votantes acreditados para votar en las elecciones, por lo que no puede tener acceso al módulo de votación.

- **Miembro de la Junta Electoral**

Una vez el administrador de la elección dé por finalizado el proceso electoral, se requerirá que varios miembros de la Junta Electoral proporcionen unas claves personales que, juntando varias de ellas, servirán como llave lógica para la apertura de la urna que contiene los votos.

Los usuarios con este rol pueden votar. El miembro de la Junta Electoral puede formar parte del censo de votantes acreditados para votar en las elecciones, con lo que, dependiendo de este detalle, puede tener acceso al módulo de votación.

- **Auditor**

El auditor debe tener acceso a una serie de funcionalidades del sistema. Su función es velar porque el desarrollo del proceso electoral se realiza sin ningún tipo de fallo o de interferencia por parte de algún atacante.

Los usuarios con este rol no pueden votar. El auditor no forma parte del censo de votantes acreditados para votar en las elecciones, por lo que no puede tener acceso al módulo de votación.

Su misión es de control, por lo que ninguna acción que realice en el sistema puede afectar al desarrollo de la elección.

- **Autoridad Certificadora**

Junto con los cuatro roles expuestos, encontramos un quinto actor en la figura de la Autoridad Certificadora. Esta entidad es la encargada de generar, administrar, validar y verificar las credenciales que han de usar cada uno de los votantes para emitir el voto, así como los de cada uno de los actores del proceso (administradores, miembros de la Junta Electoral, auditores o incluso los sistemas y sus comunicaciones).

Al hacer uso del DNIe para la identificación y firma del votante, en estas fases, es la Dirección General de la Policía, dependiente del Ministerio del Interior, la que actúa como Autoridad de Certificación, combinando dos pares de claves con un ciudadano concreto a través de la emisión de sendos Certificados de conformidad con los términos de la Declaración de Prácticas y Políticas de Certificación (DPC) [23] [24] que rige el funcionamiento y operaciones de la Infraestructura de Clave Pública de los Certificados de identidad pública y firma electrónica del Documento Nacional de Identidad (DNIe).

- **Trustee**

Es necesario una serie de actores en el proceso, los llamados Trustees por Helios Voting. Estos actores, que también pueden formar parte de los anteriores grupos

de actores, en especial los miembros de la Junta Electoral, son los encargados de custodiar un trozo de la clave secreta que el sistema necesita para la fase de escrutinio. Helios necesita una clave que sirve para descifrar la totalización de los votos cifrados. Es en este punto donde se aprecia el protocolo de secreto compartido. La clave se trocea y se reparte entre los trustees del sistema. Es necesario que estos colaboren para descifrar el resultado de la elección. Para que el sistema sea comprometido, sería necesario que todos ellos se pusieran de acuerdo para manipular el sistema, lo cual es poco probable, pues se supone que son autoridades confiables o de signos políticos contrarios.

Los usuarios cuyo rol sea *Votante* son los únicos que tienen capacidad para poder votar en la elección.

Hay usuarios que pueden ejercer varios roles en el sistema. Tanto un *Administrador* como, sobre todo, un *Miembro de la Autoridad Electoral*, perteneciendo a la Universidad pueden tener derecho al voto, por lo que deberán también poseer el rol de *Votante* para tener permiso de sufragio.

Por ello, en estos casos de usuarios con múltiples roles asociados, para mantener la transparencia y fiabilidad del proceso, estos votantes, a la hora de votar deberán hacerlo accediendo al sistema con un usuario distinto, con un rol asignado de *Votante*, garantizando que no pueden acceder con éste a ningún otro módulo del sistema.

Se entiende que el rol de *Auditor* está destinado a personas ajenas a la votación que tienen como objetivo vigilar y asegurar que la elección se lleva a cabo de forma limpia y correcta. Por ende, se entiende que las personas con usuarios asociados a estos roles no deben acceder al sistema de votación en ningún caso, además de que no deberían aparecer acreditados en el censo electoral.

### 5.3. Integración

Una vez identificados tanto requisitos como actores intervenientes en el proceso electoral, es el momento de abordar la integración de sistemas para obtener la implementación necesaria para realizar la elección.

- El sistema ha de solucionar las necesidades de cada una de las fases de una elección (3.3).
- Del mismo modo, ha de cumplir con los requisitos definidos (5.1.1).
- Ha de satisfacer también las necesidades y facilitar el cumplimiento de responsabilidades de los actores identificados (5.2).

Para llevar a cabo estos objetivos es necesario identificar los subsistemas que se requieren para implementar la integración de sistemas.

Tras un estudio de posibilidades y alternativas, se ha decidido implementar el sistema en base a tres módulos integrados (figura 5.2):

- Servidor central de votación ⇒ **Helios Voting**
- Servidor de autenticación / identificación de usuario ⇒ **oAuth 2.0 Server**
- Cliente web + app Android ⇒ **Web / App Android Policía**

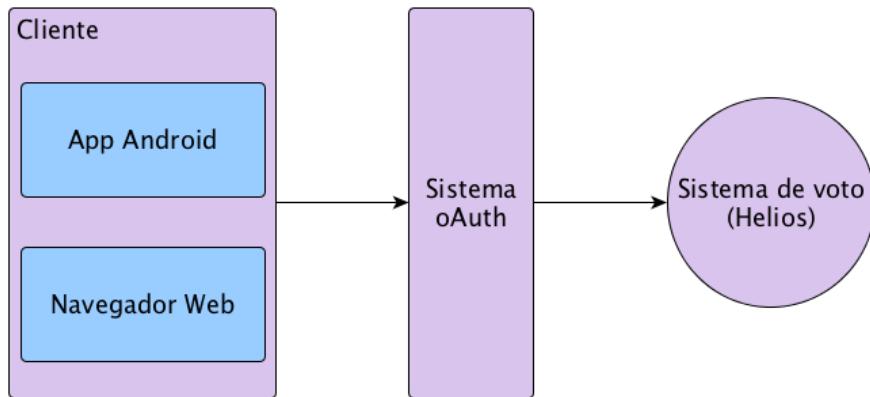


Figura 5.2: *Elementos integradores del sistema*

### 5.3.1. Sistema central de votación

El sistema orquestador del protocolo de votación elegido será Helios Voting<sup>2</sup>, de Ben Adida<sup>3</sup>. Este proyecto cumple perfectamente con requisitos tales como el bajo coste, al ser software libre y correr sobre sistemas como Unix, que no requiere pago de licencias. También, gracias a la forma en la que está implementado, las acciones que requieren mayores recursos de cómputo se realizan o en el cliente (cifrado del voto) o tan sólo una vez al finalizar una elección (totalización y su descifrado), por lo que durante el resto del tiempo no necesita un sistema con gran capacidad de cálculo; en su caso, dependiendo del número de elecciones y de votantes, requiera una plataforma fácilmente escalable basándose en el número de usuarios concurrentes que puedan acceder al sistema.

Tras un estudio de diferentes opciones para este módulo del sistema a implementar, estos son algunos motivos por los que se ha decidido adaptar el sistema Helios Voting como:

**Base criptográfica** : Para mantener la fiabilidad de un sistema de voto por Internet es fundamental una buena base criptográfica y unos protocolos cimentados en sólidos

<sup>2</sup><https://github.com/benadida/helios>

<sup>3</sup><http://ben.adida.net/>

procesos matemáticos. El mundo de la criptografía es muy complejo y en un sistema de votación en la que el votante debe tener su secreto y anonimato protegidos, se vuelve crucial.

Por ello, se ha estimado buscar la solución que implemente los protocolos criptográficos más robustos, desarrollada y mantenida por grandes expertos en la materia, reconocidos internacionalmente en este campo.

En este caso, encontramos al desarrollador de Helios Voting, el norteamericano Ben Adida<sup>4</sup>, doctor en Ingeniería Informática en Criptografía y Seguridad de la Información por el MIT<sup>5</sup> (Massachusetts Institute of Technology), siendo asesorado<sup>6</sup> en el doctorado por otro Doctor como Ron Rivest<sup>7</sup>, experto en Criptografía y Voto Electrónico.

Además, para el desarrollo de Helios, se apoyó en la asesoría de profesionales con muchísima experiencia acreditada en conceptos tales como el voto electrónico y criptografía como son Lessig o Benaloh<sup>8</sup> (considerado el precursor de la verificabilidad punto a punto (ver 2.4)), entre otros. Con esta carta de presentación, se observa que el nivel criptográfico del sistema tiene un amplio sustento académico detrás. Igualmente, existen multitud de artículos en los que se presentan, estudian y discuten los protocolos criptográficos del sistema Helios Voting en sus diferentes versiones. Incluso en muchos de estos artículos se ofrecen alternativas a distintos elementos para aumentar la seguridad del sistema o para introducir alternativas a sistemas de voto pero manteniendo la máxima seguridad para que el sistema siga siendo E2E verifiable.

En estas bibliografías se explican las bondades de Helios Voting respecto a los protocolos criptográficos de votación, siendo considerado por muchos expertos como el mejor sistema de votación por Internet auditabile E2E existente para elecciones con un bajo riesgo de coacción.

Sin duda alguna, está más que justificada la decisión de delegar la seguridad criptográfica al propio sistema.

**Software Libre:** Helios es un sistema de voto por Internet basado en desarrollo de código abierto. Está escrito en Python, utilizando Django como framework de desarrollo. Esta filosofía de software libre permite que el código pueda ser reutilizado y modificado para adaptarlo a las necesidades de la Escuela. Además, el lenguaje utilizado es bastante potente y la facilidad de integración con los recursos que provee la Escuela es un punto importante que se ha tenido en cuenta.

**Flexibilidad:** El proyecto Helios Voting<sup>9</sup> está activo, con desarrolladores implicados, in-

---

<sup>4</sup><http://ben.adida.net/>

<sup>5</sup><http://mit.edu/>

<sup>6</sup><https://vote.heliosvoting.org/about>

<sup>7</sup><http://people.csail.mit.edu/rivest/>

<sup>8</sup><http://research.microsoft.com/~benaloh/>

<sup>9</sup><https://github.com/benadida/helios-server>

cluso con forks en Github<sup>10</sup> proponiendo alternativas o soluciones. Por ejemplo, proyectos como Ágora Voting (2.4.2) nacieron a partir de un fork de Helios Voting, aunque actualmente han virado hacia otros protocolos criptográficos como base de su solución. Existen también algunos forks de este proyecto en el que lo que se realiza es un cambio del protocolo criptográfico, por ejemplo, hay uno (helios-server-mixnet<sup>11</sup>, del desarrollador RunasSudo) en el que se da soporte a mixnets y a criptografía de umbral. Así, se puede observar que los desarrolladores hacen uso de Helios como base y lo adaptan a sus necesidades criptográficas o funcionales.

**Verificabilidad E2E:** Según algunos informes [11] desarrollados estudiando una experiencia real del sistema, Helios Voting es un ejemplo excepcional de sistema electoral por Internet E2E verificable y lo reconocen como un “*estándar en verificabilidad de voto por Internet*”. Proporciona al proceso electoral todas las garantías que aporta la verificabilidad E2E, como la capacidad de observar pruebas de conocimiento cero no interactivas que verifican que cada voto fue incluido correctamente y que el escrutinio completo fue computado con precisión.

**Sistema para elecciones con bajo riesgo de coacción:** Una de las debilidades de Helios Voting es que el protocolo en el que se basa no es efectivo ante la coacción. Por ello, su propio desarrollador indica que es un sistema pensado para elecciones con bajo riesgo de coacción. Las elecciones a llevar a cabo en la Escuela Politécnica Superior cumplen con esta propiedad, pues el riesgo de que se intente coartar a algún votante es mínimo. Por ello mismo, una de las debilidades de Helios no incide en gran medida en el sistema a implementar.

**Los votos cifrados no se descifran:** Gracias a que su protocolo criptográfico se basa en el **cifrado homomórfico aditivo** (ver 2.5.1.5.1), no es necesario que sean descriptados para realizar el escrutinio, por lo que protegen la privacidad, secreto y anonimidad de cada votante, cumpliendo con varios de los requisitos fundamentales del voto electrónico/por Internet.

Buscando alguna solución para implementar identificación de usuarios de la EPS, ya provee de mecanismos como OAuth, que puede ser modificado para utilizar el DNIe como herramienta de acceso seguro a la web, o sistemas como el CAS utilizado en las elecciones de la Universidad Católica de Lovaina, que podría servir de base a un sistema de acceso basado en las cuentas de usuario de los estudiantes de la EPS. Son frameworks que no valen tal cual están implementados, pero que permiten una posibilidad de modificación muy interesante para implementar nuevos sistemas de acceso basados en las necesidades de la EPS.

---

<sup>10</sup> <http://www.github.com>

<sup>11</sup> <https://github.com/RunasSudo/helios-server-mixnet>

<b>Base criptográfica</b>	La reputación de la comunidad universitaria que está detrás de Helios Voting es suficiente como para garantizar la calidad y funcionalidad de los protocolos criptográficos implementados en el sistema.
<b>Software Libre</b>	Helios Voting es un proyecto de Software Libre, lo que permite que pueda ser modificado. Esto lo convierte en un buen candidato para utilizarlo como base de un sistema integrado, adaptándolo a las necesidades propias de nuestro desarrollo.
<b>Flexibilidad</b>	Helios Voting es un sistema maduro, con usuarios activos en su desarrollo en Github y con varios sistemas desarrollándose como <i>forks</i> de éste en la misma plataforma, por lo que hay una buena comunidad detrás de él.
<b>Verificabilidad E2E</b>	Helios es un estándar de facto en cuanto a verificabilidad E2E, el protocolo que considero que mejor se amolda al tipo de sistema que se desea implementar en este PFC.
<b>Bajo riesgo de coacción</b>	Helios define como una de sus principales vulnerabilidades que está dirigido a elecciones con bajo riesgo de coacción, como es el caso de la que se afronta en este PFC, por lo que no es un factor determinante para no utilizar este sistema.
<b>Votos cifrados no se descifran</b>	Protocolo criptográfico basado en cifrado homomórfico aditivo, que permite totalizar sin necesidad de descifrar votos individualmente.
<b>Capacidad de integración del DNIe</b>	Helios proporciona sistemas de identificación / login de terceros, incluyendo la posibilidad de poder escribir uno propio, en este caso para incluir el DNIe 3.0 como método válido.

Tabla 5.1: *Principales motivos de elección de Helios Voting como base del Sistema*

Con la intención de utilizar el DNIe como herramienta de seguridad para el login, ha sido necesario adaptar el sistema de login propio de Helios Voting. El sistema ya de por sí ofrecía alternativas, tales como login con terceros como Google, Facebook, Yahoo! o Twitter. También ofrece login por usuario/contraseña. Y se desarrolló un CAS para el login en las elecciones de la Universidad de Lovaina. Cualquiera de estos sistemas de login está bien para el tipo de elección que queremos desarrollar - incluso el del CAS, asociándolo a la cuenta de alumnado y profesorado de la Universidad me resulta muy interesante - pero no cumplen con los objetivos iniciales del proyecto. Por ello ha sido necesaria la integración de un sistema que permita la utilización del documento oficial español.

### 5.3.2. Sistema de autenticación / identificación

Se ha utilizado un sistema basado en el protocolo OAuth de tres patas como base para el sistema de login. Este protocolo está implementado en el proyecto django-oauth2-server<sup>12</sup> desarrollado por Richard Knop. De todos modos, ha sido necesario modificarlo para adaptarlo a las necesidades del DNIe.

Dentro del objetivo de utilizar el DNIe como herramienta criptográfica para la identificación del usuario, estaba el de emplear el nuevo DNIe 3.0 para poder realizar esta identificación segura desde dispositivos móviles. El objetivo final del proyecto se basa en que un votante pueda votar remotamente, haciendo uso de su documento nacional de identidad (obligatorio, y por tanto completamente extendido) para ser identificado, desde cualquier lugar en el que tenga conexión a Internet. Con este documento se puede conseguir esta última premisa. Para ello, había que encontrar la forma de adaptar este protocolo al de Helios. La forma de realizarlo es a través del login contra el sistema que se ha presentado anteriormente.

### 5.3.3. Cliente web + app Android

El escenario que se busca es uno en el que desde los dispositivos móviles se pueda acceder a los certificados digitales del DNIe y presentarlos al sistema de identificación para permitir el acceso al sistema.

El entorno web está preparado para establecer protocolos seguros basados en certificados digitales, pero es hostil en cuanto estos certificados se encuentran en medios externos al Sistema Operativo.

Los navegadores web actuales tienen una buena integración con los Sistemas Operativos y hacen uso sin problemas del almacén de certificados digitales de estos. Así pues, no tienen problema a la hora de establecer canales HTTPS contra servidores que se consi-

---

<sup>12</sup><https://github.com/RichardKnop/django-oauth2-server>

deran seguros según el SO, ya que posee certificados y la confiabilidad de las autoridades certificadoras que los han generado.

El problema viene cuando el servidor requiere un certificado de cliente y éste se encuentra en un dispositivo externo al Sistema Operativo. Normalmente una tarjeta criptográfica, como en nuestro caso podemos considerar el DNIe físico. En este caso, se requiere un driver que permita al SO acceder a los certificados, lo cual no es un problema con el DNIe, pues desde la web de la Policía se pueden obtener e instalar. Una vez el SO tiene acceso a estos certificados, nos encontramos el problema de que ha de pasarlos al navegador y éste enviarlos al servidor. Aquí es donde afirmamos que la Web no está preparada para este proceso. Desde que apareció el DNIe 2.0, se han ido desarrollando diferentes módulos que permitan realizar una comunicación fiable desde la Web con los certificados del documento. Así, lo más utilizado han sido applets de Java, que se descargan y ejecutan siendo la herramienta que comunica al navegador con los drivers que se comunican con el documento.

Hay cierto desarrollo enfocado a este problema, tratando de desligar el DNIe de applets de Java, considerados peligrosos por ser un agujero de seguridad, además de alejarse de las recomendaciones del W3C respecto a cómo implementar la Web. Así, hay implementaciones puramente Javascript, que obvian estos programas Java y se basan sólo en lenguaje cliente ejecutado en el navegador. Pero no son la solución.

Con este problema en mente, aparece el DNIe 3.0 y aporta un nuevo enfoque, los certificados, además de por contacto como hasta ahora, son accesibles sin contacto, a través de un lector NFC. Los nuevos dispositivos móviles ya implementan esta tecnología y empieza a ser prácticamente de serie según van saliendo al mercado nuevos modelos. Se ha vuelto una tecnología accesible. Es el momento de hacer uso de ella.

La Policía ha puesto a disposición de los desarrolladores tres aplicaciones de Android como ejemplo de cómo se pueden acceder a los certificados a través de estos dispositivos por NFC<sup>13</sup>.

Se ha decidido utilizar una de estas aplicaciones, concretamente la que muestra cómo acceder al chip de identificación, como base de un aplicación adaptada que permita acceder a este certificado, comunicarse con el servidor de autenticación y logar al usuario con el servidor de votación, permitiéndole a partir de ese momento utilizar la aplicación web desde el navegador web de su dispositivo.

## 5.4. Esquema de Voto Electrónico

El esquema de voto electrónico que se va a implementar en el sistema de voto de este PFC se corresponde con el que ya implementa el sistema Helios Voting.

---

<sup>13</sup> [https://www.dnielectronico.es/PortalDNIe/PRF1\\_Cons02.action?pag=REF\\_037](https://www.dnielectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_037)

El esquema se basa en cifrado homomórfico (utilizando el algoritmo de ElGamal) y verificación con pruebas de conocimiento cero de Chaum-Pedersen.

Algo en lo que se desenvuelve muy bien el homomorfismo es en la totalización, por lo que es una gran herramienta para un sistema de voto. Un protocolo que es capaz de totalizar votos cifrados sin necesidad de descifrarlos es una gran opción para mantener el secreto del voto.

No obstante, no es la mejor solución para la confianza en el sistema, pues ningún votante u observador es capaz de asegurar que un voto haya sido almacenado en la urna digital o correctamente totalizado en el recuento, así como si ha sido manipulado previamente.

Para paliar este problema de confianza es donde aparecen las pruebas de conocimiento cero. A partir de ellas, gracias a pruebas criptográficas se puede asegurar que tanto votos como la suma de ellos no han sido manipulados y fueron correctamente cifrados, así como asegurar su correcta inserción y descifrado en la totalización.

El proceso empieza con el votante cifrando su voto. Este voto se publica cifrado en un tablón público en el que se acumulan todas las relaciones (votante)-(voto cifrado) y donde cualquier votante puede comprobar que su voto ha sido correctamente tenido en cuenta o que ningún votante ha votado dos veces (que el sistema tenga en cuenta dos votos de un mismo votante, ya que cada votante sí que puede votar más de una vez, pero sólo el último voto es el que el sistema debería contar).

Cuando se acaba el período de votación y se tienen todos los votos almacenados, se aplica el homomorfismo para obtener el cifrado de la suma de los votos. Esta suma es descifrada a continuación. Para realizar el descifrado de la totalización, se utiliza una clave privada generada antes de comenzar la votación, que ha sido dividida y repartida entre un número determinado de autoridades confiables y honestas que la han custodiado. Estas partes forman parte del protocolo de secreto compartido que se aplica en el sistema. Si una autoridad no es honesta, no se pone en riesgo la integridad de la elección, pues no será posible que altere el resultado ni violará el principio de anonimato del voto, ya que no será capaz de descifrar ni votos individuales ni el resultado de la suma de estos.

Las autoridades que poseen parte de la clave privada la ponen en común y con ella se descifra la suma de los votos, con lo que se obtiene el resultado final de la elección.

Otra cuestión importante de este esquema es cómo podemos asegurar que un voto ha sido cifrado con la opción de voto que realmente ha seleccionado el votante. Puede ser que en pantalla elija la opción B, pero cómo puede asegurarse que en el sistema es esta elección, y no la opción A o C, por ejemplo, la que se ha almacenado cifrada. Es común que los protocolos de voto electrónico que se basan en el cifrado homomórfico se apoyen en pruebas no interactivas de conocimiento cero para ello. Consisten en proponer la suficiente base matemática como para que usando algún tipo de algoritmo de cifrado asimétrico y con

la colaboración del votante, se puede probar que un valor cifrado es realmente el cifrado de la opción elegida y no otra diferente.

## 5.5. Helios Voting

Tras el estudio de diferentes opciones para la implementación del *core* del sistema de votación, se ha decidido, teniendo en cuenta los motivos planteados en 5.3.1, basar el desarrollo de este módulo en la adaptación del proyecto de software libre Helios Voting.

Por ello, se plantea aquí una breve descripción de varios elementos técnicos y funcionales de este proyecto, así como de las actuaciones que se deben realizar sobre cada módulo para poder realizar una integración satisfactoria y desarrollar un sistema que cumpla con los requisitos y necesidades de la Elección a la Junta de Escuela.

### 5.5.1. Fases de la elección en Helios

Desde el punto de vista del administrador, el actor que más acciones puede realizar en el sistema, una elección con el sistema Helios Voting se desarrolla a través de una serie de fases o acciones.

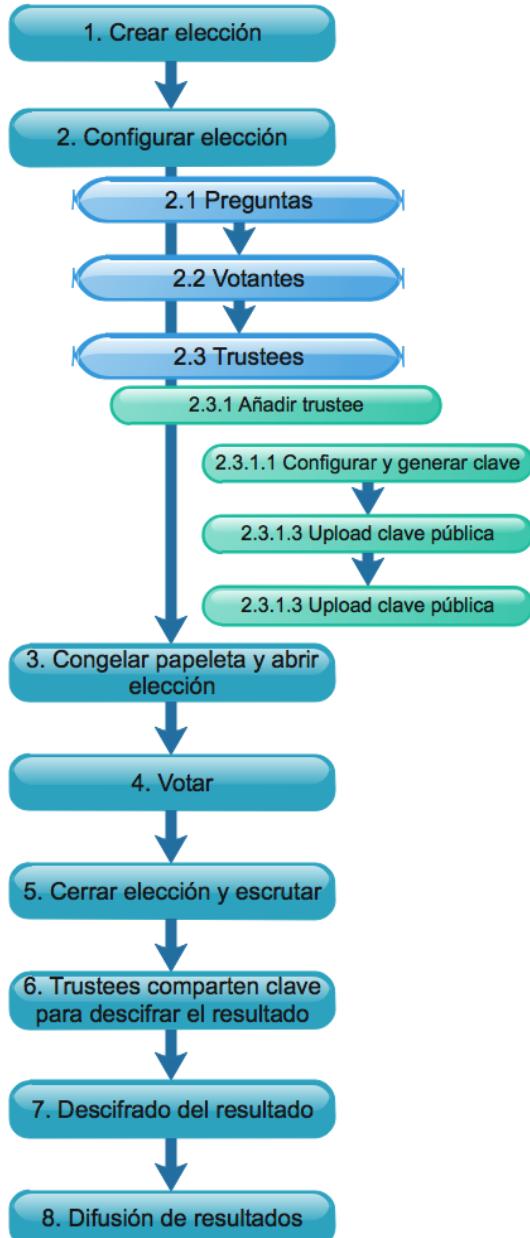
**Identificación usuario / login** El primer paso del protocolo es la identificación del usuario para ingresar al sistema. Helios propone varios tipos de servicios de login. Desde el basado en usuario/contraseña contra base de datos, hasta los que utilizan servicios OAuth contra servidores de autenticación de terceros, como Facebook, Google o Yahoo!.

**Crear elección** El primer paso funcional es la creación de la Elección. Corresponde al administrador crear la elección. Para ello, puede poner el nombre y descripción del Proceso, seleccionar si es una elección al uso o un referéndum, si se usan los nombres de los votantes o pseudónimos, si se presentan los nombres de los candidatos a cada votante en orden aleatorio y si la elección es privada para los votantes registrados o abierta a cualquier votante.

**Preparar la elección** Antes de abrir el proceso de votación hay que preparar la elección. Para ello hay tres elementos que requieren ser inicializados.

**Preguntas** Helios funciona a base de preguntas al electorado. Con una elección creada, el primer paso debería ser formular dichas preguntas y sus posibles respuestas, así como si es multiselección o de elección simple.

**Censo de votantes** Es necesario incluir un censo de votantes o permitir una votación abierta. En el caso del censo, Helios permite hacer cargas de ficheros

Figura 5.3: *Fases de una elección en Helios Voting*

CSV con la información del votante (id, email y nombre). También permite que cualquiera pueda emitir un voto.

**Añadir Trustees** En voto electrónico, es necesario tener entidades de las que poder fiarse para el correcto funcionamiento de la elección. Como es complicado que exista una entidad externa que cumpla con los requisitos, esto se emula utilizando varios trustees. Básicamente, los trustees son entidades fiables que es difícil que se puedan poner de acuerdo entre sí para falsear la elección. Estas entidades combinan sus claves públicas y privadas para formar la clave pública y privada de la elección a utilizar en la encriptación de los votos y la desencriptación del

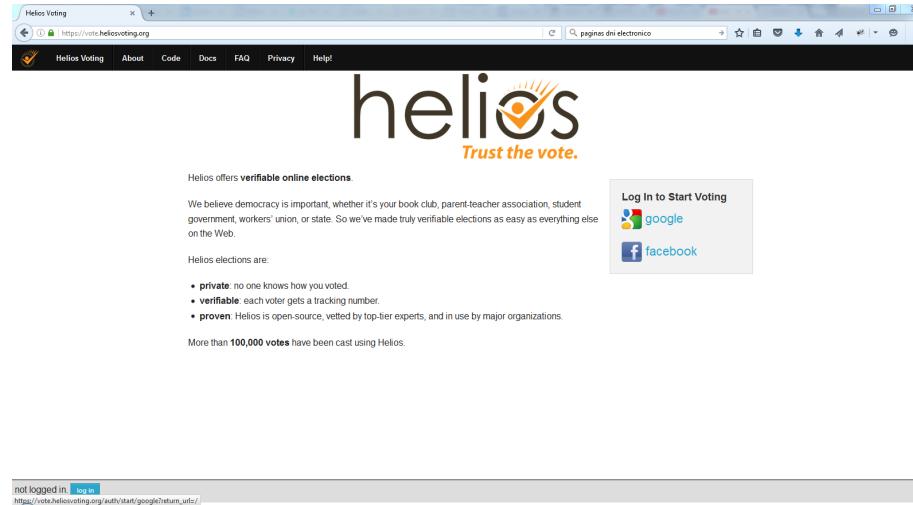


Figura 5.4: Fase de Login en Helios Voting

The screenshot shows the 'Create a New Election' page. The form fields include:

- Short name:** Election\_de\_prueba
- Name:** Elección de prueba
- Description:** Una elección para probar la funcionalidad de Helios Voting.
- Type:** Election
- Use voter aliases:**  If selected, voter identities will be replaced with aliases, e.g. "V12", in the ballot tracking center.
- Randomize answer order:**  enable this if you want the answers to questions to appear in random order for each voter.
- Private?**  A private election is only visible to registered voters.
- Help Email Address:** carlosjimenezgomez@gmail.com

At the bottom, there's a 'Next >>' button and a status bar indicating 'logged in as Carlos Jiménez Gómez'.

Figura 5.5: Fase de creación de una elección en Helios Voting

resultado del escrutinio. En Helios, el propio sistema es un trustee, suficiente para llevar a cabo los procesos criptográficos en caso de que no se defina ninguna entidad o usuario confiable adicional.

#### Configuración y generación de la clave

#### Guardar la clave secreta

#### Subir la clave pública (verificar la clave privada)

**Congelar papeleta y abrir la elección** Una vez se ha definido la papeleta (diferentes preguntas y opciones a elegir), es tarea de los administradores proceder a su congelamiento, con lo cual se asegura que la misma no va a ser modificada durante el periodo de votación. Una vez congelada, se acomete la apertura de urnas, con lo que el sistema permite a los votantes emitir sus votos.

**Votar** Durante el tiempo fijado al definir las características de la elección, los votantes

pueden logarse en el sistema y emitir sus votos.

**Calcular recuento cifrado** (cerrar votación y computar) Una vez superada la fecha fijada como cierre de urnas, los administradores proceden a iniciar el escrutinio. Para ello, lo primero que se produce es un cierre virtual de las urnas, con lo que los votantes no podrán emitir ningún voto a partir de este momento. A su vez, se inicia el procedimiento de cálculo de votos, el cual, debido al esquema homomórfico de la elección, realizará el recuento sumando los votos encriptados. El resultado de esta suma de votos cifrados, necesitará ser descifrado para poder publicar el resultado.

**Combinar trustees de descifrado y totalizar los resultados** Para poder realizar el descifrado de la totalización de los votos, se requiere que todos los trustees de la elección combinen su secreto compartido de la clave privada de descifrado de la elección.

**Computar resultados** Una vez se han combinado las claves necesarias de los trustees, el sistema procede al descifrado de la suma de los votos. Una vez realizado este proceso es cuando se tienen los resultados en claro.

**Difusión de resultados** La difusión de los resultados en el sistema Helios Voting es simplemente una página donde se recogen cada una de las preguntas de la papeleta, junto con sus opciones propuestas y el resultado de los votos que cada una ha obtenido. Esta difusión se realiza en dos pasos. Primero, el sistema muestra los resultados al administrador. El segundo paso requiere que el administrador de su aprobación para liberar los resultados, momento en el cual cualquier usuario del sistema, votante, auditor o un tercero externo tendrá la posibilidad de ver los resultados y verificar la integridad de los mismos.

**Auditorías** El sistema Helios Voting implementa dos mecanismos de auditoría activa (que puede realizar el votante y/o auditores o visitantes).

Por un lado, el votante tiene la posibilidad de auditar un voto de forma individual. Puede realizar el paso de votar una opción y auditar este voto. El voto será descifrado y presentado en claro, para que compruebe la correctitud del proceso. Este voto, no obstante, será desechado y el votante deberá volver a realizar el proceso de emisión de voto.

Por otra parte, Helios Voting implementa un mecanismo de auditoría por el que, tanto los votantes como cualquier persona interesada en el proceso, tiene la posibilidad de auditar el escrutinio una vez ha sido publicado para probar la integridad del mismo.

Ambos procedimientos se explican en 5.5.2.

### 5.5.2. Auditorías

Las auditorías son procesos fundamentales para la fiabilidad de un sistema de voto por Internet. En este caso vamos a introducir aquellas que, usando pruebas de conocimiento nulo (2.5.1.3), permiten comprobar el correcto funcionamiento de cada proceso de la elección.

Durante el período de votación, tanto el votante como entidades externas pueden realizar una serie de pruebas o auditorías para confirmar el correcto estado del proceso. Estas auditorías se realizan a distintos niveles dependiendo del actor que desee realizarlas, así los votantes pueden realizar auditorías de correctitud de su voto emitido, mientras que auditores externos no pueden llevar a cabo estas comprobaciones. Sin embargo, tanto los primeros como los segundos pueden realizar una auditoría al escrutinio.

La interfaz de usuario de Helios posee una zona dedicada a las pruebas de auditoría. En ellas, según se va avanzando en la Elección, van apareciendo enlaces e instrucciones según se pueden comenzar a realizar pruebas a diferentes elementos del sistema.

Las auditorías de procesos que permite Helios Voting durante el proceso de votación:

**Verificación de la papeleta (antes de votar)** El sistema permite a un votante verificar su papeleta cifrada antes de utilizarla para votar. No obstante, como medida para combatir la coacción, una vez la papeleta ha sido verificada se descarta y no se utiliza como voto. El votante deberá volver a encriptarla para poder emitir su voto.

Una vez el votante está en la cabina de voto virtual y ha encriptado sus opciones de voto, el sistema permite realizar una verificación para asegurarse de que la encriptación de estas ha sido correcta. Para ello, el sistema presenta al votante la información para auditar su papeleta. Este texto es el que debe copiar el votante para utilizar en la herramienta de verificación individual de papeletas. En ésta, se incluye dicho texto y la URL de la elección datos suficientes para realizar la verificación.

Una vez hemos auditado nuestro voto, tenemos la posibilidad de publicar la papeleta auditada en el *Helios Tracking Center*, permitiendo a otros observadores realizar una doble comprobación de la verificación de la papeleta.

**Lista de papeletas auditadas** Como hemos avanzado en el punto anterior, en el momento de realizar la votación, el votante puede realizar una verificación del contenido encriptado de su papeleta. El sistema le permite, una vez realizada la comprobación, publicar esta *papeleta abierta* en un tablón con una lista de papeletas auditadas. En cualquier momento de la elección, cualquier usuario, sea votante o no, puede acceder a esta lista y realizar la misma verificación sobre las papeletas ahí publicadas. Estas papeletas, no obstante, no son las que se han utilizado para emitir el voto, pues ya se ha comentado que una vez el votante la *abre* para su verificación, ha de reencriptar su voto para *meterlo en la urna* definitivamente.

**Auditoría de la elección ya escrutada** Una vez se ha realizado el escrutinio, el sistema permite a los observadores interesados, realizar una prueba para verificar la veracidad del resultado.

En el caso de Helios, esto se corresponde con una herramienta de verificación, Helios Election Verifier, a la cual se le pasa la URL de la elección que se quiere verificar. Una explicación y ejemplo de este proceso se puede ver en 5.5.9 (p. 142).

### 5.5.3. Protocolo criptográfico

Otra de las piedras angulares en las que se debe basar un sistema de votación por Internet es en su protocolo criptográfico y en las primitivas que utiliza, base para poder argumentar la fiabilidad de los procesos que se realizan.

Helios Voting basa su seguridad criptográfica en un protocolo de cifrado homomórfico (2.5.1.5).

El voto se encripta en el navegador del cliente utilizando una implementación del algoritmo de ElGamal Exponencial (2.5.1.5.2) realizada en Javascript.

Como se ha explicado en 2.5.1.5.1 las propiedades de un cifrado homomórfico implican que el cifrado del producto de dos mensajes (números) se corresponda con el producto de ambos mensajes cifrados.

El algoritmo de ElGamal es un algoritmo de cifrado homomórfico. Este algoritmo define que "*el cifrado de la multiplicación de dos elementos es igual a la multiplicación de ambos elementos cifrados*". A partir de este algoritmo, se realiza una evolución importante para el voto electrónico, con el el cifrado homomórfico exponencial. De este tipo es el algoritmo de ElGamal Exponencial, que define que *.el cifrado de la suma de dos elementos es igual que la suma de los dos elementos cifrados*". Esto se obtiene variando ElGamal tradicional hacia la exponenciación de sus elementos. Se basa en la elección de un número grande usado como generador, el cual usa el mensaje como exponente, con el objetivo de utilizar la problemática de la resolución del logaritmo discreto. El producto de dos generadores elevados exponencialmente cada uno a un mensaje y cifrados es igual que el cifrado del generador elevado a la suma de los dos mensajes.

Aplicando estas propiedades al voto electrónico podemos inferir que cifrando los votos de cada votante, se puede obtener el la suma de los votos cifrados. Cuando se descifra esta suma de votos cifrados, se obtiene el resultado totalizado de los votos. Es decir, que se pueden cifrar los votos y conseguir obtener la suma de ellos sin violar el secreto de cada voto individual. En ningún momento el voto está en plano, desde que el votante lo cifra al votar permanece encriptado. Ni siquiera al totalizar se descifran los votos, sólo se descifra la suma de ellos.

### 5.5.4. Identificación

Una vez visto el sistema sobre el que basar los procesos electorales que hemos de llevar a cabo en la EPS, es el momento de implementar modificaciones en el mismo.

En primer lugar, el cambio más importante es el que se refiere a la identificación de votantes.

En Helios, de forma nativa, los sistemas que permiten la entrada de votantes al sistema son usuario/password y una serie de servicios OAuth de terceros, incluyendo Facebook, Twitter, Google y Yahoo!.

Existe una modificación especial para las elecciones de la Universidad de Lovaina en la que se implementó un sistema CAS contra la base de datos de alumnos de la Universidad. Con este sistema, los usuarios se podían logar en el sistema con la identificación de la Universidad y contra un servicio que ésta brindaba, es decir, que el control de entrada de usuarios se delegaba en la Universidad y se confiaba en su sistema a la hora de proporcionar roles y permisos a los usuarios del sistema.

Para las Elecciones en la EPS, podría llegar a valorarse este sistema CAS, modificándolo para adecuarlo a la infraestructura digital de la Universidad. Pero no es uno de los objetivos de este PFC, teniendo en cuenta la idea de utilización del DNIe 3.0 y la necesidad de limitar la interacción con la Universidad por el acceso a sus sistemas.

#### 5.5.4.1. Evolución

La primera idea para integrar el DNIe 3.0 en el sistema Helios consistía en realizar un acceso seguro, por SSL, que requiriese autenticación por el DNIe, del mismo modo que la mayoría de las webs de la Administración pública.

Para llegar a esta implementación, era necesario modificar el sistema de login de Helios, así como el servidor sobre el que corre.

Hubo que configurar el servidor para que éste requiriese el establecimiento de un canal seguro SSL a través de los certificados del DNIe. Para ello, hay que modificar la configuración del servidor Apache sobre el que corremos el proyecto Django sobre el que se implementa Helios Voting.

Una vez configurado el canal HTTPS, había que implementar un módulo de login que pidiese al usuario el DNIe. Para esto existe un applet de java que proporciona la Dirección General de la Policía que cumple con el cometido.

Con estas implementaciones somos capaces de que un usuario intente acceder al sistema de votación, este le pedirá las credenciales del DNIe para validarlas y establecer un canal

SSL seguro con el servidor. A través del canal seguro establecido, el applet presenta las credenciales del usuario por medio del certificado privado del DNIe, previa consulta al usuario del PIN correspondiente. Con las credenciales que recibe el sistema de login, se le puede conceder o denegar acceso al usuario identificado

El Ministerio del Interior de España<sup>14</sup>, a través de la Dirección General de la Policía, es el órgano encargado de la gestión y desarrollo del DNI electrónico<sup>15</sup>. Por tanto, además de ser los responsables del control y la emisión de estas credenciales, son los responsables de desarrollar las APIs y documentación para que los desarrolladores puedan utilizar los certificados digitales que llevan en aplicaciones o servicios prestados por terceros.

En el caso del DNIe 3.0, en el área de descargas<sup>16</sup> de la web de la DGP han publicado el código fuente de tres aplicaciones Android y su respectiva documentación para dar una muestra de cómo implementar apps que hagan uso del certificado NFC que trae como novedad este documento.

Estas tres aplicaciones compartidas se centran en las tres funcionalidades que aporta este chip:

**Aplicación Android de Lectura:** Código fuente de una aplicación que permite a un dispositivo Android con un sensor NFC el leer los datos públicos del chip NFC del DNI electrónico.

**Aplicación Android Autenticación:** Código fuente de una aplicación que permite a través de un dispositivo Android el poder autenticarse contra un servidor con las credenciales digitales que integra el chip NFC en el DNI electrónico.

**Aplicación Android Firma:** Código fuente de una aplicación que permite la firma de documentos u otro tipo de objetos digitales con el certificado digital personal privado que ofrece el DNIe a través de software en dispositivos Android gracias a la tecnología sin contacto NFC.

El sistema Helios Voting permite votar en unas elecciones desde un navegador web. El objetivo es que también podamos hacerlo desde un smartphone a través de una app que nos permita votar desde cualquier punto con conexión a Internet, por lo que hay que valorar la mejor opción para ello.

La primera opción es utilizar, como se suele hacer desde un computador de escritorio, un navegador web.

Una primera prueba que se realizó fue tratar de autenticarse en una web a través del DNIe.

---

<sup>14</sup> <http://www.interior.gob.es/>

<sup>15</sup> <http://www.dnielectronico.es/PortalDNIe/>

<sup>16</sup> [http://www.dnielectronico.es/PortalDNIe/PRF1\\_Cons02.action?pag=REF\\_1120](http://www.dnielectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_1120)

Al entrar en la Sede Electrónica de la Agencia Tributaria, para consultar los expedientes es necesario facilitar certificados digitales para poder ser identificado y autenticado contra el sistema. Al intentar abrir la página, el navegador nos muestra un aviso en el que nos indica que no se han encontrado los certificados (ver figura 5.6). Incluso nos ofrece la posibilidad de instalarlos. Podríamos dirigirnos a la web de la FNMT<sup>17</sup> y gestionar la emisión de un certificado digital, que podríamos descargar y utilizar como identificación digital de la misma forma que lo podemos hacer con el DNIe. De todos modos, la obtención de este certificado por parte de la FNMT requiere de un paso en el que el usuario ha de personarse físicamente en una Oficina de Registro para acreditar su identidad. Con este certificado instalado en el almacén de certificados de Google Chrome se podría gestionar la identificación contra la web a la que tratamos de acceder, pero hemos decidido no continuar por este camino porque es necesario realizar trámites extras que no consideramos necesarios frente a la facilidad de obtener el DNIe.

Implementando algún tipo de soporte de autenticación por DNIe en Helios y tratando de acceder desde el móvil a través del navegador, nos encontraríamos este problema. Chrome, o el navegador que utilicemos, no tiene los certificados digitales cargados en su almacén de certificados.

## App Android

Otro camino de investigación se basa en la app de Autenticación publicada por el Ministerio del Interior a través de la Policía.

En esta, se implementa un ejemplo a través del cual se pueden realizar varios trámites de consultas contra la web del Ayuntamiento de Madrid. Concretamente, permite las consultas:

- Tributos
- Gestión de multas
- Datos censales
- Expedientes

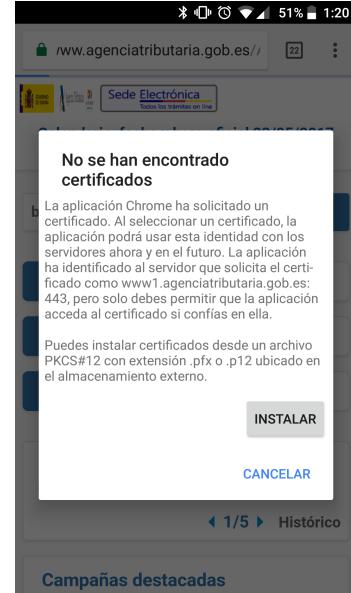


Figura 5.6: Mensaje en la web de la Sede Electrónica de la Agencia Tributaria que indica que no se han encontrado certificados en el dispositivo.

<sup>17</sup> <https://www.sede.fnmt.gob.es/certificados/persona-fisica/obtener-certificado-software>

El funcionamiento es sencillo. Una vez se abre la app, aparece un menú con las gestiones que se pueden realizar.

Por ejemplo, pulsamos sobre Consulta de los datos censales.

La siguiente pantalla que nos aparece es una en la que se nos requiere que introduzcamos el CAN (Card Access Number) que podemos ver en el anverso del DNIe físico en la zona inferior derecha. Este identificador es un número de 6 dígitos que se utiliza para establecer un canal cifrado PACE entre el DNIe y el dispositivo. Como indica la documentación proporcionada por los desarrolladores el canal PACE (Password Authenticated Connection Establishment) es obligatorio para establecer esta conexión cifrada, pero en ningún caso reemplaza al código PIN de acceso a los certificados ni da permisos de uso de las claves de firma y/o autenticación del ciudadano. Se trata de una medida de seguridad extra para evitar que un tercero pueda realizar lecturas de datos públicos sin conocimiento del dueño del DNI.

En caso de haber introducido previamente el código CAN en la app, éste se puede visualizar en un listado seleccionable para que no haga falta que el usuario tenga que introducirlo cada vez que quiera utilizar los certificados.

Una vez introducido el CAN, la siguiente pantalla nos pide que acerquemos el DNIe 3.0 al dispositivo.

Es en este momento cuando se produce la comunicación entre dispositivo y el chip de radiofrecuencia, permitiendo el paso de respuestas y comandos entre ambos.

Una vez se realizan las operaciones básicas de lectura de certificados, la app solicita el PIN del DNIe al usuario. Con este PIN, se autentica al usuario contra el DNIe y concede permisos de acceso a las claves. A partir de este momento, todas las comunicaciones entre DNIe y dispositivo van cifradas.

El usuario puede a partir de aquí realizar las operaciones que requieren sus claves, como puede ser la firma de documentos o, la que buscamos para el sistema Helios, establecer una conexión segura con el servidor prestador de servicios, entre otras.

Siguiendo con la app de ejemplo, a continuación nos conecta con el servicio de datos censales del Ayuntamiento de Madrid y nos muestra nuestros datos censales en caso de una conexión correcta.

En principio es una muy buena opción para implementar para el sistema de votación. Permite establecer un canal seguro con una autenticación fiable a través de los certificados del DNIe.

Al final lo que obtenemos es la web empotrada dentro de un WebView<sup>18</sup> creado por la aplicación.

---

<sup>18</sup> <https://developer.android.com/reference/android/webkit/WebView.html>

## Problema

Después multitud de pruebas y de bucear en foros sobre seguridad digital, certificados digitales, DNIe y Android, resultó haber un problema bastante importante con esta aproximación.

Partamos de la base de la correcta configuración del servidor y login a través de certificados del DNIe. La prueba de la correcta configuración residía en que era posible acceder al servicio, e incluso llegar a emitir un voto, utilizando el DNIe (tanto 2.0 como 3.0) en un PC, utilizando el chip de contacto y utilizando un navegador como el Mozilla Firefox.

Demostrado que era factible esta arquitectura, se intentó desde una versión modificada de la app original de la Policía. En ella, se ajustó el certificado que espera el servidor, así como el link al que tiene que dirigirse tras el login.

Al realizar las pruebas, se observaba que era posible logarse en el sistema, la app incluso lo mostraba por pantalla en el WebView anteriormente citado. Pero el problema es que realmente la lógica de la app de la Policía no funciona como la de un navegador al uso.

El funcionamiento es el de peticiones simples estilo REST. Es decir, establece la conexión con el servidor autenticándose con los certificados del DNIe y realizando una llamada a una URL concreta. Esta le devuelve la información requerida y la app construye un WebView y *pinta* el HTML recibido en el WebView.

Esto, para las funciones de la aplicación de ejemplo cumple con su cometido, pero en nuestro caso no es así, pues si la página que devuelve el servidor contiene links (que tendrá que ser así para poder interactuar con la web) la gestión de los certificados no se mantienen. En este caso, no podemos navegar con la conexión HTTPS establecida.

El elemento WebView de Android todavía no es capaz de soportar este tipo de funcionamiento. No es posible, en la primera conexión con el DNIe almacenar las claves en el almacén del navegador para poder reutilizarlas en siguientes peticiones.

Ante este problema, no es posible continuar con la forma de trabajar de la app de la Policía.

## REST

Ante este problema aparece una alternativa. Plantear la web del sistema como un servicio REST.

Helios funciona casi en su mayoría tratando mensajes JSON como paso de información entre sus distintas fases, por lo que se podría modificar el *Front End* para responder a peticiones REST. Con este nuevo enfoque, se podría implementar una app Android nativa

que tuviese todas las opciones que aparecen en la web y que para su funcionamiento realizara peticiones a la web. La información que recuperaría, en lugar de ser HTML que muestra en un *WebView* embebido, como ocurre en la app de la Policía, serían objetos JSON con la información requerida. Obviamente, para cada llamada habría que reutilizar la técnica que se ha descrito en el paso anterior, por la cual se puede realizar una llamada HTTP acoplando las claves del certificado para autenticarse contra el servidor.

El problema de esta solución es que pasamos de un sistema web ya existente a una implementación del sistema para Android, junto con la reimplementación de la web para que soporte tanto votación navegando por ella como por llamadas REST.

Además, Helios realiza el cifrado de los votos con un módulo de Javascript, delegando en el cliente la carga de las operaciones criptográficas. Esta propuesta requiere también reescribir los módulos criptográficos en Java, con el aumento del riesgo y la necesidad de pruebas intensivas para asegurar el correcto funcionamiento de un módulo que ya está implementado y probado.

Quizá es el futuro a seguir por el sistema, pero requiere tiempo y recursos no planificados.

## Solución

En la búsqueda de una solución que no comprometiese la viabilidad del proyecto, se pensó en una intermedia a las anteriores.

Las llamadas a Helios, teniendo que mantener el canal HTTPS cifrado con el DNIe, deben ser contra un servicio que ha de ser *renderizado* en un *WebView*. En caso de querer navegar una vez establecida la conexión, hay que volver a realizar la llamada con los certificados y volver a crear (y *renderizar*) el *WebView*. Es decir, no se obtiene la navegación natural que se busca.

Tampoco es viable implementar una app de voto. Teniendo ya implementada una aplicación web, es un gasto de recursos tener que desarrollar y mantener ambas soluciones, cuando la web puede ser adaptada y utilizada tanto desde navegadores de escritorio como desde dispositivos móviles.

Es en este momento en el que aparece el concepto de autenticación delegada. Los desarrolladores de Helios han implementado módulos de OAuth para ingresar al sistema autenticándose con sus cuentas de Facebook, Google o Yahoo!, o un módulo CAS para hacer lo propio con las credenciales de la Universidad de Lovaina. Del mismo modo, se podría implementar un sistema de autenticación delegada basado en el DNIe.

La idea es la de tener dos servidores diferenciados. Por un lado el del sistema de votación y por otro el servidor de autenticación.

El usuario, al intentar entrar en el sistema de votación es redirigido al de autenticación. Contra este ha de identificarse con su DNIe. Si la autenticación es correcta, el servicio de autenticación otorgará permisos de acceso al usuario en el servicio de votación y será redirigido de nuevo a éste. A partir de este momento, el usuario, con permisos de accesos, podrá operar en el sistema de votación autenticado de forma segura.

Con este planteamiento, para votar desde el móvil o desde el PC no tiene mucha diferencia. En ambos casos se accede desde el navegador a la página principal de la Elección. Aquí se pulsa el enlace de autenticación por DNIe. La diferencia principal se observa en este momento:

**PC:** Desde el PC, el usuario será redirigido al servicio de autenticación y, si el PC tiene correctamente configurados los drivers del DNIe, aparecerá un applet a través del cual se le solicita que introduzca el documento en el lector y, posteriormente, la clave.

**Smartphone:** Por su parte, desde Android, el enlace de login en vez de redirigir a otra web, lanzará un *intent* de Android en con el que se abre una aplicación Android (si no está instalada, se redirige a la web o *market* desde la cual puede instalarse). Con esta aplicación se realizan las llamadas necesarias al servicio de autenticación y se accede a los certificados del DNIe a través del protocolo NFC. Una vez que el servicio de autenticación gestiona entre el dispositivo y el servidor de votación la validez del usuario, la aplicación se cierra y devuelve al usuario al navegador pero ya logado en el sistema y con una conexión HTTPS segura.

A partir de este momento, tanto en PC como en Smartphone, el usuario estaría ya logado con seguridad en el sistema y podría comenzar a operar con el sistema de votación.

#### 5.5.5. Adaptación a dispositivos

Helios es un sistema que se comenzó a desarrollar alrededor de 2008. En aquel momento, el *boom* de los dispositivos móviles no estaba en un punto tan álgido como en los últimos años. El desarrollo se dirigió, por tanto, al mundo de la web del momento, centrada en los dispositivos con pantallas de cierto tamaño.

Un vistazo a la web estándar de Helios Voting<sup>19</sup> muestra que es *responsive* en cierta medida. Da la sensación de serlo, aunque realmente no está diseñada para ello.

Los textos se observan muy apelmazados y ajustados a los bordes de la pantalla de un dispositivo del tamaño de un smartphone.

Más aún, si accedemos a la urna virtual, ya sea para votar o para auditar un voto, se puede ver cómo la web pierde su resolución y deja de cumplir con las características de

---

<sup>19</sup> <http://www.heliosvoting.com>

una web *responsive*, con la consecuente dificultad de navegación.

Con el objetivo del voto por Internet en el horizonte, esta incapacidad de adaptar la web a dispositivos portátiles pasa a ser un detalle a tener muy en cuenta. Lo ideal es que la web del sistema al completo sea *responsive*. Además sería muy valorable que cumpliese los estándares<sup>20</sup> que marca el consorcio W3C<sup>21</sup>, así como el mayor número posible de reglas de accesibilidad sin comprometer la funcionalidad ni la simplicidad de interacción con el usuario.

Para ello, se incluye una serie de librerías que facilitan la transformación de la web para ser adaptada a la visualización desde pantallas de dispositivos móviles, destacando Bootstrap<sup>22</sup> y Modernizr<sup>23</sup>.

Con estas librerías y ajustes planos de CSS y Javascript se consigue que la visualización de las páginas se adapten a los diferentes dispositivos desde los que van a ser visualizadas.

#### 5.5.6. Votación

La fase de votación, en la que en un sistema tradicional un votante introduce su voto en la urna junto a los votos del resto de votantes, es uno de los derechos fundamentales que un ciudadano tiene en un estado de derecho. Es, además, el derecho que más iguala a los ciudadanos, pues (interpretaciones de la Ley D'Hont<sup>24</sup> o del peso de cada circunscripción en la Ley Electoral vigente aparte) cualquier persona, independientemente de su condición sexual, social, económica, religiosa, etc. se corresponde con un voto. Una vez hemos votado, todos los votos tienen el mismo valor. El famoso lema de "una persona, un voto".

Obviamente, es el elemento central de cualquier sistema de votación.

Teniendo en cuenta los criterios sobre los que se basa el voto electrónico/por internet, esta fase debería cumplir una serie de propiedades inherentes al mismo.

A la hora de diseñar el protocolo de un sistema de votación, uno de los puntos más importantes es el de decidir si el voto debe ser único. Hay que valorar qué es lo más adecuado para el protocolo, si un votante puede emitir tantos votos como quiera, validando tan sólo uno de ellos y rechazando el resto. O si, por el contrario, el votante debería ejercer su derecho al voto tan sólo una vez y deberíamos asegurar que no hay posibilidad de que pueda volver a hacerlo.

La posibilidad de permitir al votante emitir su voto más de una vez proviene, en parte, de la necesidad de proteger al votante frente a la coacción por parte de un tercero.

---

<sup>20</sup><https://www.w3.org/standards/webdesign/mobilweb>

<sup>21</sup><https://www.w3.org/>

<sup>22</sup><http://getbootstrap.com/>

<sup>23</sup><https://modernizr.com/>

<sup>24</sup>[https://es.wikipedia.org/wiki/Sistema\\_d'Hondt](https://es.wikipedia.org/wiki/Sistema_d'Hondt)

En caso de ser coaccionado a votar por un candidato, siempre tendrá la posibilidad de volver a emitir un nuevo voto que invalide el anterior mientras esté abierto el período electoral.

Este mecanismo, unido a que no se emite ningún recibo que demuestre el contenido del voto, hace muy complicado que un sujeto coaccionador pueda llevar a cabo un fraude electoral, ya que resultaría muy complicado que pudiese conseguir que el voto por el que coacciona sea el último emitido por todos y cada uno de los votantes que necesita para cambiar el resultado electoral.

Sin embargo, otras corrientes entienden que para que un votante pueda votar varias veces, invalidando por el camino los votos intermedios, es necesario que, durante un período del proceso electoral, estén almacenados en el sistema una tupla voto-votante. Esto ocurre porque si un votante emite un voto, cuando quiera volver a votar, el primero ha de ser descartado. Por este motivo es necesario que sea posible encontrar el voto previo, por lo que debe almacenarse en el sistema con alguna asociación al propio votante.

En principio, esta situación provoca una violación del anonimato del votante, característica esencial en muchos procesos electorales. Sin ir más lejos, este detalle descartaría, basándonos en la Constitución Española, a un sistema con esta propiedad, pues en el Texto se recoge que el voto ha de ser anónimo y no puede ser trazado. Manteniendo la dualidad votante-voto, esta relación podría ser trazada.

Es por ello que estas corrientes defienden que, de la misma forma que el voto tradicional, es más seguro para el votante el que vote solamente una vez, siendo este el único voto que debe ser introducido en la urna. Además, de este modo, no hace falta guardar la dupla voto-votante en el sistema, basta con almacenar el voto en la urna digital y marcar en el censo que el votante ya ha votado.

En el caso del protocolo criptográfico de Helios, en el que se apoya este desarrollo, se puede implementar una elección con un mecanismo de voto múltiple a la vez que se protege el anonimato de cada votante. Esto se consigue utilizando el cifrado homomórfico. Los votos, al estar cifrados, pueden almacenarse junto a la identidad de su votante. En caso de un nuevo voto, se puede localizar y descartar el anterior. Al final, se une su último voto al del resto de los votantes y se totalizan, para, en el último paso, descifrar el resultado de la suma. La protección del anonimato y la imposibilidad de trazabilidad del voto es posible porque una vez que un voto ha sido cifrado, nunca será descifrado, por lo que nunca estará almacenado un voto en plano junto con su votante.

### 5.5.7. Difusión de resultados

La difusión de resultados por parte de Helios consiste en una web muy simple que indica las distintas opciones de voto y cuántos votos ha recibido cada una. Ni siquiera ordena los

resultados por orden de votos contabilizados.

### **Abstención**

Un problema que tiene la web de Helios con respecto a la información que queremos mostrar en estas Elecciones es que no se muestra la abstención, ya que es un dato para el que no está diseñado el sistema original, ya que en principio permite votaciones con censo cerrado y abierto. En este último caso no tiene sentido, por tanto, el concepto de abstención, pues no tenemos constancia del censo de potenciales votantes del proceso.

Sin embargo, para el tipo de elección que desarrollamos para la EPS, con censo cerrado, se ha creído oportuno implementar un mecanismo para que la abstención sea mostrada en la difusión de resultados. Para ello hay que retocar el módulo de presentación de resultados e incluir una consulta del censo para la elección y contrastar el número resultante con la suma de los votos (contando los blancos) que se han emitido.

### **Falta información electoral**

Junto con el problema de la abstención, podemos incluir que falta información acerca de la propia elección. Muestra las votaciones por "preguntas", pero no aporta datos como el comentado de la abstención, censo, participación y diversos valores que deben estar presentes para dar legitimidad a un proceso electoral.

Sería correcto incrementar el valor de la información en la medida de lo posible añadiendo este tipo de datos a la web de difusión de resultados. Incluso, resultaría muy interesante acompañar estos datos en crudo con algún tipo de elaboración de datos para dar mayor contexto al resultado del Proceso.

En el caso de las Elecciones a Delegado o a la Junta de la USP CEU esto podría consistir en ofrecer en la web contenido extra como informes de votación agregada o comparativas entre diferentes grupos de votantes o candidatos, es decir una serie de estadísticas que den mayor información, siempre objetiva, del desarrollo y resultado del Proceso Electoral.

Un ejemplo podría ser en el caso de la elección de delegado. Aunque un alumno votante sólo esté interesado en quién ha resultado elegido en su clase, se le debería proporcionar la opción de conocer resultados en otras clases. Del mismo modo, resulta interesante complementar la información del ámbito de cada votante comparándola con el resto de ámbitos, mostrando qué datos ofrece su circunscripción en cuanto a participación, disgragación del voto, rangos horarios, etc.

Las estadísticas, si son las adecuadas, son herramientas que aportan valor añadido a este tipo de difusión de información.

### 5.5.8. Verificación del voto

Un problema que surge en un sistema de voto electrónico o por Internet es el de la veracidad de los resultados.

La difusión del resultado de la votación debería incluir dos herramientas para poder limitar la duda en dos aspectos.

- Tratar de asegurar a la persona que consulta los resultados la veracidad de los mismos, que la consolidación de los votos emitidos ha sido correcta, sin equivocación por parte del sistema o fraude por parte de los organizadores del Proceso. (*Verificabilidad Universal*)
- Asegurar al votante que ha participado en la elección una herramienta que le permita verificar que en el resultado que está viendo su voto ha sido contado y, además, correctamente. (*Verificabilidad Individual*)

Es la base de los sistemas de voto E2E verificables.

En procesos electorales tradicionales este punto no tiene tanta trascendencia porque, normalmente, confiamos en el protocolo de la elección. Nos aporta seguridad ver cómo nuestro voto lo introducimos nosotros mismos en la urna transparente y sabemos que el conteo de los votos, además de ser público, se realiza entre un grupo de vocales designados aleatoriamente y apoyados por interventores de diferentes fuerzas políticas que, difícilmente puedan llegar a un acuerdo para falsificar el conteo de la mesa electoral en cuestión.

Pero como ya se ha comentado ampliamente, esto no ocurre en procesos electorales por Internet, ya que, en principio carecemos de urna física y de ciudadanos anónimos que cuenten los votos. Hay que fiarse del software que realiza el proceso. Y esta confianza suele ponerse en duda en fases anteriores del proceso, pero sobre todo aquí, cuando se observa el resultado de la totalización de los votos.

El sistema de voto de este proyecto, basándose en Helios, cumple con las premisas de verificabilidad universal e individual de los sistema de verificabilidad punto a punto. En este caso, se observa que en el momento del voto, el votante puede auditarlo y publicarlo en un tablón para que otros también lo auditén y demostrar así la no manipulación del contenido del mismo.

### 5.5.9. Verificación de la totalización e integridad de la elección

Al finalizar el proceso electoral, el resultado de la totalización de los votos puede ser auditada por cualquier observador.

Helios provee una herramienta, llamada *Helios Election Verifier*, la cual, dada la URL de una elección que haya finalizado se encarga de ejecutar las pruebas criptográficas necesarias para, sin comprometer la integridad de la elección y el secreto de los votantes, verificar la correctitud de la misma. Así, la herramienta comprueba la huella digital de la elección, el número de votantes, cada una de las opciones de voto de cada votante, así como los Trustees y la totalización.

El informe que emite la herramienta tiene contenido análogo al siguiente ejemplo:

```

1 -- Election --
2
3   loading election...
4   loaded election: Elección de prueba
5   election fingerprint: CpiW3hBXawR5kiZyq8p914wPExx01gsaUbOnuWrRzeY
6   loading list of voters...
7   loaded voter list, now loading ballots for each..
8   loading ballot for voter #1
9
10  === FOUND a ballot for voter #1 ===
11
12 -- Ballots --
13
14   Voter #1
15   -- UUID: bb872ace-94fd-4029-93c5-6fd61db6f635
16   -- Ballot Tracking Number: Go5SKpKeXNxvwoXhSeQrPXMIeNa0B8gLp6F/OMaDLwE
17   Question #1, Option #1 -- VERIFIED
18   Question #1, Option #2 -- VERIFIED
19   Question #1, Option #3 -- VERIFIED
20
21  === Question #1 OVERALL -- VERIFIED ===
22
23 -- Trustees --
24
25   Trustee #1: undefined
26   -- PK ICGGnxHLDJKoU1W5/aiPKVf2zBJLjM5FLQj9s3Y+x1E -- VERIFIED.
27
28  === election public key CORRECTLY FORMED ===
29
30 -- Tally --
31
32   Question #1: Pregunta de prueba
33   Answer #1: Respuesta 1 - COUNT = 0
34   -- Trustee undefined: decryption factor verifies
35   -VERIFIED
36   Answer #2: Respuesta 2 - COUNT = 1
37   -- Trustee undefined: decryption factor verifies
38   -VERIFIED
39   Answer #3: Respuesta 3 - COUNT = 0
40   -- Trustee undefined: decryption factor verifies
41   -VERIFIED
42
43  === FINAL RESULT ===
44
45  === ELECTION FULLY VERIFIED -- SUCCESS! ===

```

Do not mistake that the ballot is  
stronger than the bullet.

---

Abraham Lincoln<sup>1</sup>

## Capítulo 6

# Solución

En este capítulo se expone la solución desarrollada para llevar a cabo el sistema analizado en los capítulos anteriores.

Durante el capítulo se expondrán la arquitectura, topología de red, software y decisiones de diseño que se hacen posible este proyecto de integración de sistemas.

El capítulo finaliza con la exposición de una prueba de concepto implementada en base a estas decisiones tomadas a lo largo del capítulo.

Este prototipo trata de demostrar que se puede implementar un sistema de voto seguro que utilice el DNIe 3.0 como herramienta de identificación digital del votante. De hecho, es la primera implementación que utiliza esta tecnología para un proyecto electoral.

Además, el prototipo muestra que la versatilidad del proyecto, pues se utiliza software libre, con herramientas libres y usando como servidores dos Raspberry Pi, unos dispositivos que destacan por una potencia destacada a un coste realmente bajo.

### 6.1. Arquitectura del sistema

Tal como se determina en el análisis (5), la solución que se propone para este Proyecto consiste en la integración de tres componentes, adaptados a las necesidades de una elección para la EPS.

**Cliente:** Para permitir el acceso del usuario al sistema utilizando el DNIe 3.0, se ha implementado una solución para poder acceder a las credenciales del chip sin contacto del documento. Este desarrollo es una aplicación Android, basada en una de ejemplo compartida por la Policía. Además de la app cliente, tanto el sistema de autenticación

---

<sup>1</sup>16º Presidente de los Estados Unidos de América (1861-1865) [https://es.wikipedia.org/wiki/Abraham\\_Lincoln](https://es.wikipedia.org/wiki/Abraham_Lincoln) Fuente: <http://www.bartleby.com/73/1903.html>

como el de votación ofrecen una interfaz web para interactuar con el usuario a través de navegadores web.

**Sistema de autenticación:** Para permitir el acceso al sistema con las credenciales del DNIe, se utiliza un sistema de autenticación basado en el protocolo oAuth2. El usuario accede a través de un navegador web o de la app Android desarrollada por HTTPS de forma segura al servidor web. Se establece un canal seguro utilizando el certificado del propio servidor y como certificado de cliente, el de identificación del DNIe. Este sistema es el que autoriza al usuario comprobando su identidad y consultando su presencia en el censo de la elección.

**Sistema de votación:** El sistema de votación es una versión adaptada del sistema Helios Voting. Se comunica con el cliente (y con el sistema de autenticación) por HTTPS y el usuario, autorizado previamente, tiene la posibilidad de, dependiendo de su rol, crear y gestionar una elección, votar en ella y/o, simplemente, consultar los resultados.

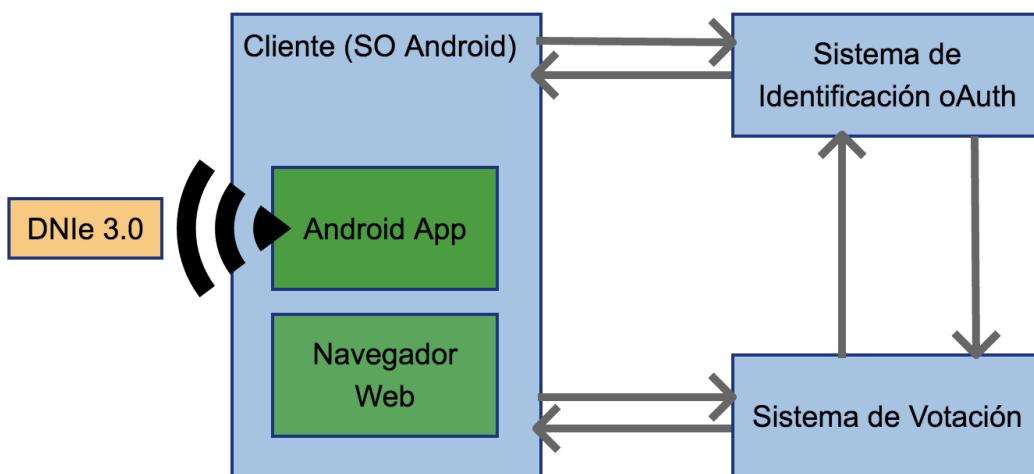


Figura 6.1: Arquitectura general del sistema

## 6.2. Sistema de autenticación

El Sistema de Autenticación implementa el protocolo de autenticación oAuth 2.0 entre el usuario y el sistema. En concreto, se ha realizado un fork del proyecto *django-oauth2-server*<sup>2</sup> desarrollado por Richard Knop. El objetivo era modificarlo para adaptarlo a diferentes necesidades teniendo en cuenta si el acceso es por la web o a través de la app Android. Además, era necesario que la autorización del usuario se realizara consultando el censo de usuarios y/o votantes de la Elección.

El sistema de autenticación se compone de un servidor web, el sistema de oAuth basado en Django que hemos comentado y una base de datos que contiene tanto permisos de

<sup>2</sup><https://github.com/RichardKnop/django-oauth2-server>

aplicaciones y usuario como los propios usuarios con permiso de acceso al sistema de votación.

El servidor web se configura para exponer un endpoint de entrada seguro, permitiendo el acceso por HTTPS. En dicha configuración es muy importante activar diversas características:

#### 6.2.1. Servidor web:

El canal de comunicación entre el usuario y el sistema de autenticación es la web, el protocolo es HTTPS y la herramienta que gestiona estos accesos es el servidor web.

Con el fin de establecer un canal HTTPS seguro entre usuario y servidor, es necesario que en su configuración se contemplen tres certificados, dos de ellos en posesión del servidor y el tercero de obligada presentación por parte del cliente.

**Certificado de servidor:** Es el certificado que permite al navegador negociar el establecimiento de la conexión segura. Si se ha obtenido de algún proveedor confiable, el navegador lo gestiona de forma casi transparente al usuario. Sin embargo, si la fuente no es confiable y no se encuentra en el almacén de certificados del sistema operativo o el navegador, se muestra una advertencia al usuario aconsejándole que no continúe la navegación, aunque le permite seguir adelante a riesgo del usuario. Es lo que ocurre también cuando se utiliza un certificado autofirmado.

En nuestro caso, se ha decidido utilizar un certificado autofirmado para el prototipo implementado. Los pasos necesarios tanto para crear y firmar el certificado como para configurar el servidor web para que lo utilice se pueden consultar en el anexo B.

**Certificado AC Raíz DGP:** Es el certificado que permite al Sistema Operativo y al navegador web reconocer la Autoridad de Certificación de los certificados de sede electrónica, la AC Raíz DGP, que son los encargados de generar los certificados que se incluyen en el DNIe 3.0.

**Certificado de cliente:** Para utilizar el DNIe como fuente de identificación del usuario, es preciso configurar el servidor web para que requiera en la conexión la presencia del certificado de autenticación del DNIe del usuario. Para ello, el servidor debe poseer el certificado raíz que provee la Dirección General de la Policía. Este certificado se configura en el servidor. Se puede consultar el anexo B para ver cómo realizarlo.

Además de configurar el servidor para que requiera el certificado de cliente SSL, es necesario aplicar la configuración necesaria para pasar la información a la aplicación que la necesita.

Para ello, el servidor web ha de exportar los datos del certificado SSL a variables de entorno.

Cuando se recibe una petición al servidor, éste inyecta estas variables en la request que hace llegar al código. Concretamente, en Python (Django) son accesibles desde el META de la request.

En el siguiente código podemos ver una función que atiende una petición GET a la vista de autorización para mostrar una lista de variables asociadas a la seguridad del servidor que recibe de éste:

```
1 class AuthorizeView(View):
2     form_class = AuthorizeForm
3     initial = {}
4     template_name = 'web/authorize.html'
5
6     @method_decorator(validate_request)
7     def dispatch(self, *args, **kwargs):
8         return super(AuthorizeView, self).dispatch(*args, **kwargs)
9
10    def get(self, request, *args, **kwargs):
11        form = self.form_class(initial=self.initial)
12        try:
13            claves = [
14                'SSL_PROTOCOL',
15                'SSL_SESSION_RESUMED',
16                'SSL_SECURE_RENEG',
17                'SSL_CIPHER',
18                'SSL_CIPHER_EXPORT',
19                'SSL_CIPHER_USEKEYSIZE',
20                'SSL_CIPHER_ALGKEYSIZE',
21                'SSL_COMPRESS_METHOD',
22                'SSL_VERSION_INTERFACE',
23                'SSL_VERSION_LIBRARY',
24                'SSL_CLIENT_M_VERSION',
25                'SSL_CLIENT_M_SERIAL',
26                'SSL_CLIENT_S_DN',
27                'SSL_CLIENT_I_DN',
28                'SSL_CLIENT_V_START',
29                'SSL_CLIENT_V_END',
30                'SSL_CLIENT_V_REMAIN',
31                'SSL_CLIENT_A_SIG',
32                'SSL_CLIENT_A_KEY',
33                'SSL_CLIENT_CERT',
34                'SSL_CLIENT_VERIFY',
35                'SSL_SERVER_M_VERSION',
36                'SSL_SERVER_M_SERIAL',
37                'SSL_SERVER_S_DN',
38                'SSL_SERVER_I_DN',
39                'SSL_SERVER_V_START',
40                'SSL_SERVER_V_END',
41                'SSL_SERVER_A_SIG',
42                'SSL_SERVER_A_KEY',
43                'SSL_SERVER_CERT'
44            ]
45            for k in claves:
```

```

46     try:
47         logger.warn(k+': '+str(request.META[k]))
48     except Exception as ex:
49         logger.warn('El objeto no tiene la clave *' + k + '* --- ' + ex)
50
51     \gls{DNIe} = get_dni_info_from_ssl(request)
52 except Exception as e:
53     logger.warn('get_No vienen las credenciales del \gls{DNIe}')
54
55 return self._render(request=request, form=form, \gls{DNIe}=dnie)

```

Listing 6.1: Variables SSL que recibe el servidor web de autenticación.

En la lista se observa una gran cantidad de variables de entorno relacionadas con la seguridad del servidor.

A continuación se muestra la ejecución de ese código para ver los valores que se devuelven y recogen del certificado, como ejemplo:

```

1  2017-01-09 18:23:10,163 DEBUG prueba: *CN=JIMÉNEZ GÓMEZ\, JOSÉ CARLOS (AUTENTICACIÓN),
   ↵ GN=JOSÉ CARLOS,SN=JIMÉNEZ,serialNumber=53159931P,C=ES*
2  2017-01-09 18:23:10,165 DEBUG SSL_PROTOCOL: TLSv1.2
3  2017-01-09 18:23:10,167 DEBUG SSL_SESSION_RESUMED: Initial
4  2017-01-09 18:23:10,170 DEBUG SSL_SECURE_RENEG: true
5  2017-01-09 18:23:10,172 DEBUG SSL_CIPHER: ECDHE-RSA-AES128-GCM-SHA256
6  2017-01-09 18:23:10,174 DEBUG SSL_CIPHER_EXPORT: false
7  2017-01-09 18:23:10,176 DEBUG SSL_CIPHER_USEKEYSIZE: 128
8  2017-01-09 18:23:10,178 DEBUG SSL_CIPHER_ALGKEYSIZE: 128
9  2017-01-09 18:23:10,180 DEBUG SSL_COMPRESS_METHOD: NULL
10 2017-01-09 18:23:10,182 DEBUG SSL_VERSION_INTERFACE: mod_ssl/2.4.10
11 2017-01-09 18:23:10,184 DEBUG SSL_VERSION_LIBRARY: OpenSSL/1.0.1t
12 2017-01-09 18:23:10,186 DEBUG SSL_CLIENT_M_VERSION: 3
13 2017-01-09 18:23:10,188 DEBUG SSL_CLIENT_M_SERIAL: 1753616228212A66BD1AB7E719BC5613
14 2017-01-09 18:23:10,191 DEBUG SSL_CLIENT_S_DN: CN=GORDILLO CARDEÑOSA\, ALFONSO (
   ↵ AUTENTICACIÓN),GN=ALFONSO,SN=GORDILLO,serialNumber=11062005B,C=ES
15 2017-01-09 18:23:10,193 DEBUG SSL_CLIENT_I_DN: CN=AC \gls{DNIe} 001,OU=DNIE,O=
   ↵ DIRECCION GENERAL DE LA POLICIA,C=ES
16 2017-01-09 18:23:10,195 DEBUG SSL_CLIENT_V_START: Jun 11 12:26:58 2016 GMT
17 2017-01-09 18:23:10,197 DEBUG SSL_CLIENT_V_END: Mar 15 22:59:59 2021 GMT
18 2017-01-09 18:23:10,199 DEBUG SSL_CLIENT_V_REMAIN: 1510
19 2017-01-09 18:23:10,201 DEBUG SSL_CLIENT_A_SIG: sha256WithRSAEncryption
20 2017-01-09 18:23:10,203 DEBUG SSL_CLIENT_A_KEY: rsaEncryption
21 2017-01-09 18:23:10,205 DEBUG SSL_CLIENT_CERT: -----BEGIN CERTIFICATE-----
22 TUIJRjh6Q0NCTnVnQXdJQkFnSVFJc1hvYW54bE5oQ1lORG1TSU85a1BUQU5CZ2tx
23 aGtpRz13MEJBUXNGQURCYwpNUXN3Q1FZRFZRUUDFd0pGVXpFb01DWUdBMVVFQ2d3
24 ZlJFbFNSVU5EU1U5T01FZEZua1ZTUVV3Z1JFVWdURUVhClVFOU1TVU5KUVRTk1B
25 c0dBMVVFQ3d3RVJFNUpSVEVVTTUJJR0ExVUVBd3dMUVVNz1JFNUpSU0F3TURFd0ho
26 Y04KTVRZeE1USX1NVE15TmpVNFdoY05NakV3TWpJMK1qSTFPVFU1V2pDQmhERUxN
27 QWtHQTFVRUJoTUNSVk14RWpBUQpCZ05WQkFVVENUVXpNVFU1T1RNefFVERVJNQThH
28 QTFVRUJBd01Ta2xOdzRsT1JWb3hGVEFUQmdOVkJDb01ERXBQC1U4T0pJRU5CVwt4
29 UFV6RTNNRFVHQTFRVRUF3d3Vta2xOdzRsT1JWb2dSOE9UVFVWYUxDQktUMVBEaVNC
30 RFFWSk0KVDFNZ0tFR1ZWRVZPVkVsRFFVTkp3NU5PS1RDQ0FTSxdEUV1KS29aSWh2
31 Y05BUUVCQ1FBRGdnRVBBRENDQVFvQwpnZ0VCQUpYcmVxMTV6R0xnMmcwbW5hcmlN
32 U1pUU05ZRTBDVn1haEw1R3ZydGZjRH1xSUY2SHgvWkxtY2R1ck5yCkd4cU1XYktz
33 amk1Z01pTFVueFowVWdpGvQn3RyRvp4NDBObW40RmxqZ2lzcmsgzMWtpNWtBQ2Rj
34 elBYaE9sN20KVThNK2ZTeXZMVUd1TmN1SUhUYkJGcWV3UwtJNTVydk1IZ2x5MjVQ
35 WjZoakczSUJQRThEckZOd0ZUbkN3NXFmYgp6TCs2ZFdxAfpj2JaZk5rV3FsdUN0

```

```

36 emhFRG5PaWJMNkp1c0s3VnhSNE1zc2dRYWZoUnVSdmddyHJsNGFINXNsCj1Gc310
37 Z1IvNmNpVHhGMmtremhDZ05GN1luNit4TGo0T3NoVXh3dVo4NRsM1A5NndrcTRH
38 eGx5V21QSWJhTDMKdFMzQmcwaGdVS3Q4cmFVNy8rQW4rc2hMaDhFQ0F3RUFBYU9D
39 QW9zd2dnS0NNQXdHQTFVZEV3RUIvd1FDTUFBdwpEZ11EV1wUEFRSC9CQVFEQWd1
40 QU1CMEdBMVVkRGdRV0JCUWlwbU1VRTJwYnh1NG5xN0tUbWhUaU9MTDFRVEFmCkJn
41 T1ZIU01FR0RBV2dCUWFpYWPgn285M1hWVnhpZk03TmIycUJRQ1ZiekFpQmdnckJn
42 RUZCUWNQXQdRV01CUXcKQ0FZR0JBQ09S20VCTUFnR0JnUUUfqa11CQkRCZ0JnZ3JC
43 Z0VGQ1FjQKFRL1VNRk13SHdZSUt3WUJCUVVITUFHrpFMmgwZEHBNkx5OZMZ053
44 TG1SdWFVXVxVaWE13THdZSUt3WUJCUVVITUFRL0kyADBkSEE2THk5M2QzY3VaRzVw
45 ClpTNWxjeT1qWlhKMGN5OUJRMUpoYVhvdVkzSjBNRHNTQFVZE1BUTBNRE13TUFZ
46 SV1JV1VBUU1DQWdRd0pE0QWkKQmdnckJnRUZCUWNDQVJZV2FIUjBjRG92TDNkM2R5
47 NWTibWxsTG1WekeyUndZekNCOEFZSUt3WUJCUVVIQVFJRKpNZU13Z2VBd01nSUJB
48 VEFQMqmdsZ2hrZ0JaUU1FQWdFRU10QUF5cDY1VThZT3M3bGxXSnlvZXRBRLzVvV01
49 TTR3CnNaNk14VnJ0OD1QV01ESUNBUUF3Q3dZS11JWk1BV1VEQkFJQkJDQuwvdS9R
50 NzZbit0RWoyRmNIQn1aN01CRVkJY2xaTmRld1pDUVNnNnVLdmNEQTZCZ2xnaFZR
51 QkFnSUVBZ0V3Q3dZS11JWk1BV1VEQkFJQkJDQ1d6T1d3cm0vOQo5d2YrVHNEZXd5
52 RnNER21kR0tNS0M2U2I0ck1zS1U5UwxqQTZCZ2xnaFZRQkFnSUVBZ113Q3dZS11J
53 WklBV1VECkJBsUJCQ0JXek5Xd3JtLzk5d2YrVHNEZXd5RnNER21k
54 -----END CERTIFICATE-----
55
56 2017-01-09 18:23:10,207 DEBUG SSL_CLIENT_VERIFY: SUCCESS
57 2017-01-09 18:23:10,210 DEBUG SSL_SERVER_M_VERSION: 1
58 2017-01-09 18:23:10,212 DEBUG SSL_SERVER_M_SERIAL: 69B5F037577EDB1B
59 2017-01-09 18:23:10,214 DEBUG SSL_SERVER_S_DN: emailAddress=pepe.mel@miservidor.rbb,CN
   ↗ =eleccionesuspceu.com,OU=EPS,O=USP CEU,L=Madrid,ST=Madrid,C=ES
60 2017-01-09 18:23:10,216 DEBUG SSL_SERVER_I_DN: emailAddress=pepe.mel@miservidor.rbb,CN
   ↗ =eleccionesuspceu.com,OU=EPS,O=USP CEU,L=Madrid,ST=Madrid,C=ES
61 2017-01-09 18:23:10,218 DEBUG SSL_SERVER_V_START: Nov 25 19:47:17 2016 GMT
62 2017-01-09 18:23:10,220 DEBUG SSL_SERVER_V_END: Nov 25 19:47:17 2017 GMT
63 2017-01-09 18:23:10,222 DEBUG SSL_SERVER_A_SIG: sha256WithRSAEncryption
64 2017-01-09 18:23:10,224 DEBUG SSL_SERVER_A_KEY: rsaEncryption
65 2017-01-09 18:23:10,226 DEBUG SSL_SERVER_CERT: -----BEGIN CERTIFICATE-----
66 wr9RdcOpIGVzIHVuIGNlcnPzmljYWRvIGVsZWN0cs0zbmljBz8KCKvzIHVuIGRv
67 Y3VtZW50byBkaWdpdGFsIHf1ZSBjb250aWVuZSwgZW50cmUgb3RyYSBpbmZvcmlh
68 Y2nDs24sIGxvcyBkYXRvcyBpZGVudGlmaWNhdG12b3MgcGVyc29uYWx1cy4gUGVv
69 bWl0ZSBpZGVudGlmaWNhcN1lIGVuIGludGVybmoIGUgaW50ZXJjYW1iaWFyIGlu
70 Zm9ybWFjacOzb1jb24gb3RyYXMgcGVyc29uYXMgY29uIGxhIGdhcmFudM0tYSBk
71 ZSBxdWUgc8OzbG8gZWwdgBl0dWxhciBwdWVkZSBhY2N1ZGVyIGEgZWxsYS4KCsk/
72 UGFyYSBxdOpIHNpcnZlIHVuIGNlcnPzmljYWRvIGVsZWN0cs0zbmljBz8KCKvs
73 IGNlcnPzmljYWRvIGVsZXryw7NuaWNhIGdhcmFudG16YSBsYSBpZGVudGlkYWQg
74 ZGVsIHVzdWFyaW8sIGxvIHf1ZSBwZXJtaXR1IHZJ1YWxpmFyIGxvcyB0csOhbW10
75 ZXMcXV1IHJ1cXVpZXJhbiBpZGVudGlmaWNhY2nDs24gc2VndXjhIHBvcIBwYXJ0
76 ZSBkZWwdgXN1YXJpby4gUGVybWl0ZSB0YW1iacOpbiBsYSBmaXjtYSB1bGVjdHLD
77 s25pY2EgZGUGz9ybXsYXJpb3MgeSBkb2N1bWVudG9zIGVsZWN0cs0zbmljB3Ms
78 IHf1ZSB0aWVuZSBsYSBtaXNtYSB2YXpZGV6IGp1cs0tZgljYSBxdWUgbGEgZmly
79 bWEgbWFudXN1cml0YSB1b1bCBkb2N1bWVudG8gZW4gcGFwZWWuCgpQcm9ibGVt
80 YXMgZGUgbG9zIGN1cnRpZmljYWRvcyBjb24gc3UgbmF2ZwdhZG9yCgpFcyBwb3Np
81 Ymx1IHf1ZSBhbCbhY2N1ZGVyIGEgbGFzIHDDoWdpbmFzIGR1IGxhIFN1ZGUGRWx1
82 Y3Ryw7NuaWNhLCBzdSBuYXZ1Z2Fkb3IgcHJ1c2VudGugdW4gbWVuc2FqZSBtw6Fz
83 IG8gbWVub3MgYXhcm1hbnR1LgoKRXN0ZSBhdmlzbyBpbmRpY2EgcXV1IGVsIHNp
84 dGlvIHf1ZSBzZSB2aNpdGEgbm8gZXMgZGUgY29uZmlhbnpHLCBhbCBubyByZWNv
85 bm9jZXIgc3UgbmF2ZwdhZG9yIGxhIEF1dG9yaWRhZCBkZSBdzXJ
86 -----END CERTIFICATE-----

```

Aquellas que comienzan por SSL\_SERVER se refieren a la información del certificado del servidor.

Por su parte, aquellas que comienzan por SSL\_CLIENT son las que contienen la información proporcionada por el certificado presentado por el cliente. En el caso de este proyecto, contendrán los datos proporcionados por el certificado de autenticación del DNIe.

De estas variables, algunas son las que se utilizan para recoger la información del usuario, contrastarlo con el censo de usuarios y poder comprobar qué permisos posee, incluido el de acceso al sistema.

**SSL\_CLIENT\_S\_DN:** Contiene la información personal del sujeto que identifica el certificado. En el ejemplo:

```
1 SSL\_CLIENT\_S\_DN: CN=GORDILLO CARDEÑOSA\,, ALFONSO (AUTENTICACIÓN), GN=ALFONSO,
   ↳ SN=GORDILLO, serialNumber=11062005B, C=ES
```

Así, del certificado de cliente obtenemos:

**CN (commonName):** GORDILLO CARDEÑOSA

, ALFONSO (AUTENTICACIÓN) ⇐ Junto con el nombre de la persona identificada por el certificado, se encuentra el tipo de certificado que estamos recibiendo, en este caso, el de AUTENTICACIÓN.

**GN (givenName):** ALFONSO

**SN (surName):** GORDILLO

**serialNumber:** 11062005B ⇐ En el caso del DNIe, en serialNumber encontraremos el número del DNI de la persona identificada por el documento.

**C (country):** ES ⇐ Código del país de expedición del documento según el estándar ISO 3166<sup>3</sup>. En este caso, ES<sup>4</sup> = ESPAÑA.

**SSL\_CLIENT\_V\_START:** Contiene la fecha de expedición del certificado.

```
1 SSL\_CLIENT\_V\_START: Jun 11 12:26:58 2016 GMT
```

**SSL\_CLIENT\_V\_END:** Contiene la fecha de caducidad del certificado.

```
1 SSL\_CLIENT\_V\_END: Mar 15 22:59:59 2021 GMT
```

### 6.2.2. Aplicación de autenticación

La aplicación recibe del servidor web la información del certificado del cliente a través de los METAS injectados en el *header* de la *request* por parte del servidor.

La aplicación implementa el flujo oAuth2 en modo Authorization Code. Con la información del usuario que recibe del certificado de cliente, lo identifica y valida los permisos que tiene.

<sup>3</sup> [https://es.wikipedia.org/wiki/ISO\\_3166](https://es.wikipedia.org/wiki/ISO_3166)

<sup>4</sup> [https://es.wikipedia.org/wiki/ISO\\_3166-2:ES](https://es.wikipedia.org/wiki/ISO_3166-2:ES)

En caso de que la validación sea correcta, genera un token que almacena en la base de datos para comunicar al sistema de votación que el votante tiene permisos de acceso. Además, genera un código de autorización que devuelve al cliente.

Este código es el que ha de presentar el cliente al sistema para confirmar su autenticación contra éste e interactuar con él.

Una vez el sistema de votación recibe el código, lo utiliza para obtener del sistema de autenticación el token de acceso que identifica la aplicación que solicita la interacción, junto con los permisos que dispone el usuario y la información contenida en la base de datos del censo y en el certificado de autenticación de su DNIe.

Sigue, por tanto un modelo del flujo de OAuth2, apoyado en una herramienta escrita por Richard Knop en Python sobre un framework Django. No obstante, este código ha servido de base, pues ha debido ser modificado para adaptarlo a las necesidades particulares de este sistema:

- Se ha modificado el proceso de obtención de datos del usuario, pues, además de obtenerlos de la base de datos, hay que extraer información del usuario de los certificados de cliente que presenta para su identificación.
- Se ha modificado el sistema de validación del usuario para que contrasta esta información del certificado de autenticación con el censo de la base de datos.
- Hay que contrastar la validez del certificado, principalmente si no ha caducado. Además, aunque al final se haya configurado en el servidor, hay que realizar una comprobación contra un servidor OCSP para ver si ha sido revocado por la autoridad certificadora.

### 6.2.3. OCSP

OCSP es un servicio de revocación que ofrece la autoridad que crea el certificado raíz, en nuestro caso la DGP y la FNMT. Estas organizaciones proveen de un servicio al cual conectarse y consultar la validez de los certificados que se estén tratando de autenticar.

El sistema funciona validando contra la OCSP de la DGP. La implementación de este sistema se realiza por medio de configuración del servidor web. Es el propio servidor quien realiza la comprobación OCSP del certificado cliente y recibe la validez o no del mismo por parte del servicio externo.

Cuando el sistema esté configurado para operar sobre una red interna será necesario desactivar esta configuración del servidor web, pues sin acceso a Internet no será posible realizar la comprobación contra el servicio de la autoridad certificadora.

Una alternativa al OCSP es la de generar una CRL. Las ventajas de las listas de revocación consisten en que son más simples que el OCSP y pueden ser consultadas sin conexión a la red, lo cual es importante en caso de que el sistema esté configurado de modo que corra sobre una red privada, sin acceso a Internet (ver figura 6.23 en 6.5).

#### 6.2.4. Esquema de la base de datos

En la figura 6.2 se presenta el diagrama entidad-relación correspondiente al sistema de autenticación.

En este diagrama se observan dos ámbitos de entidades diferenciados:

- Entidades para gestión de tokens.
- Entidades para gestión de autenticación de usuario.

### 6.3. Sistema de votación

El Sistema de Votación es el *core* del sistema. Es el que soporta toda la lógica de la elección.

De cara a los administradores, permite gestionar una Elección desde su creación y definición hasta la publicación de resultados tras el escrutinio.

De cara al votante, le permite acceder a la elección de las alternativas que se le presentan para votar y emitir su voto de forma segura, secreta y anónima, con la seguridad de que no es manipulado y que será correctamente escrutado.

Junto al módulo central de votación, este sistema presenta otros cuatro módulos específicos.

**Cabina de votación:** Es el módulo que permite a los votantes elegir el voto y cifrarlo.

Lo separamos del módulo principal porque es un módulo escrito principalmente en javascript y se ejecuta en el navegador del cliente. Se delega en la máquina del cliente la potencia necesaria para el cifrado de su voto.

**Verificador de la elección:** Es un módulo de acceso externo que permite, pasándole la huella digital de una elección que ya haya terminado, la verificación de la integridad de la misma. Así, por medio de funciones criptográficas, asegura que la elección ha sido correctamente escrutada y que todos los votos fueron correctamente incluidos en el escrutinio.

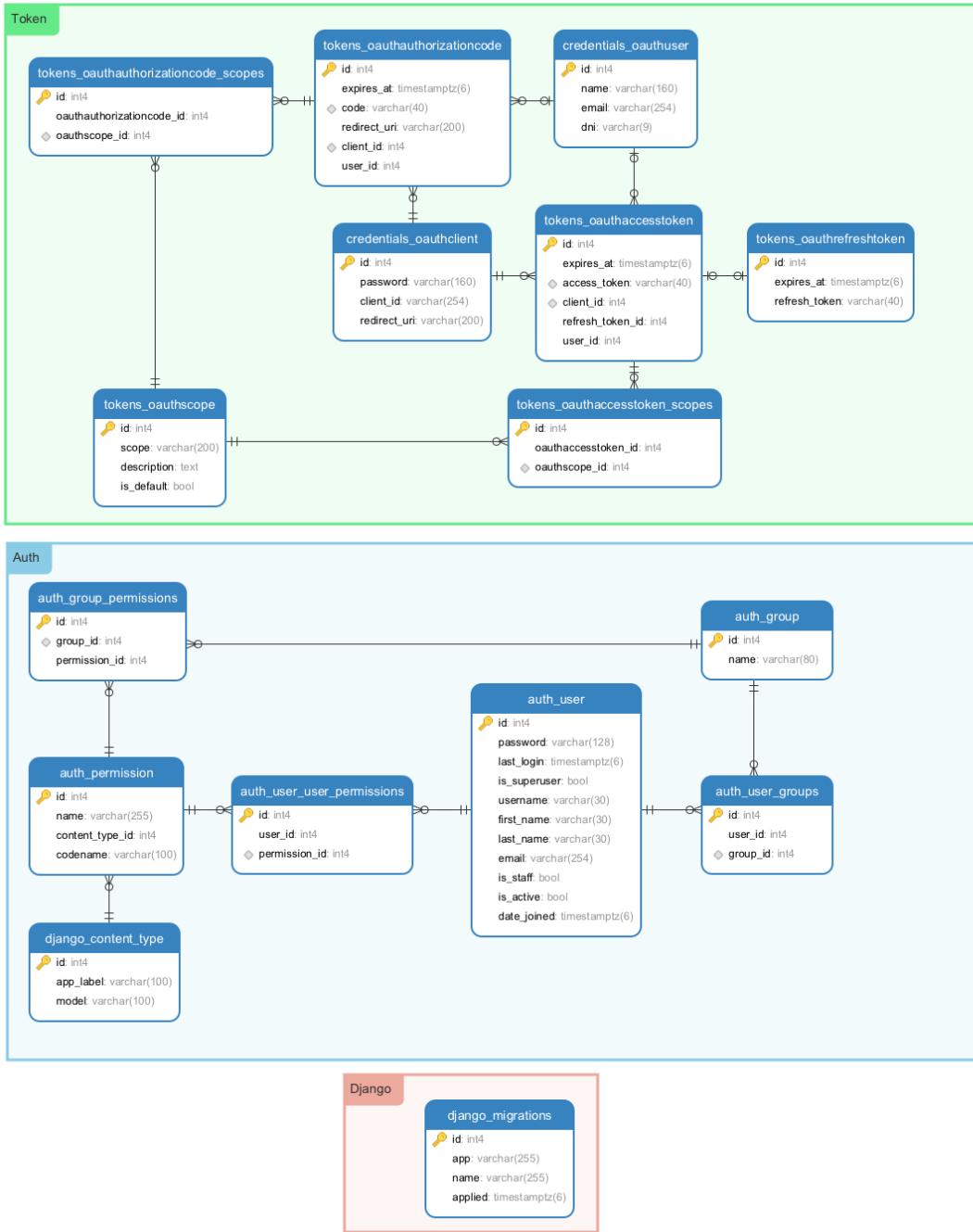


Figura 6.2: Diagrama ER del Servidor de Autenticación

**Gestor de tareas:** Utilizando las librerías celery y kombu, Helios implementa un gestor de tareas basado en una cola. En el sistema se definen varias tareas que, cuando toca llevarlas a cabo, en vez de encargarse de realizarlas de forma síncrona, las incluye en una cola y sigue con el flujo de funcionamiento normal. Celery se encarga de consultar esta cola y extraer las tareas que hay que realizar.

Es el responsable de realizar tareas como la carga de ficheros de votantes del censo, iniciar el escrutinio o el envío masivo de emails a usuarios del sistema, entre otras.

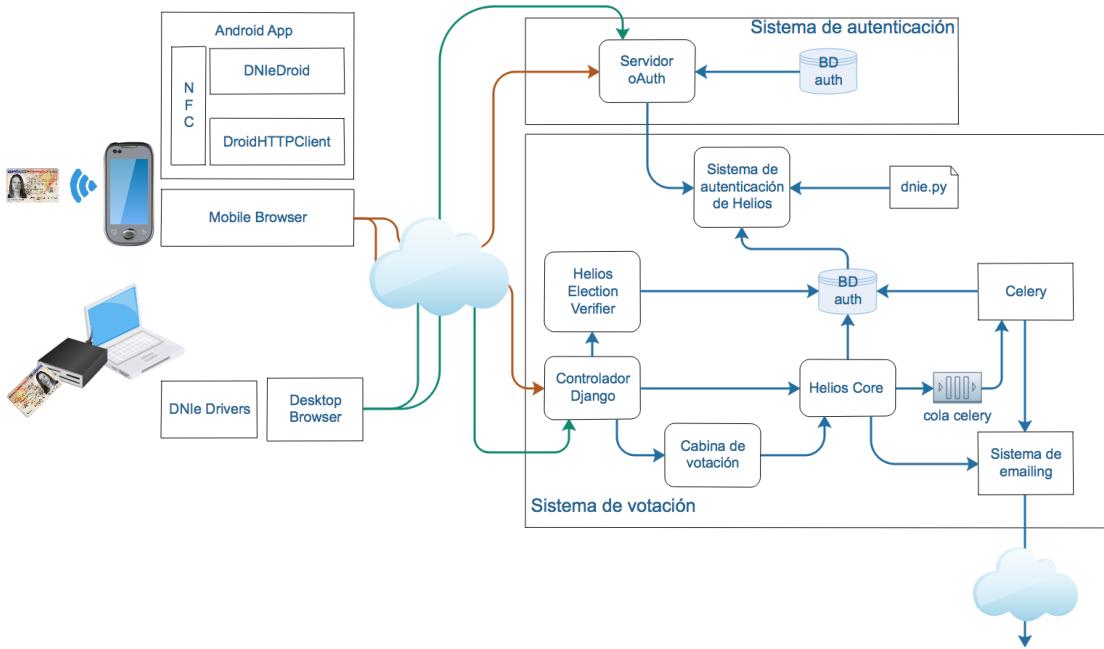


Figura 6.3: Arquitectura del sistema

**Sistema de emails:** Hay un subsistema encargado del envío de emails a usuarios. Lo utiliza tanto el módulo central como celery para enviar emails avisando de los estados de la elección, envío del trozo correspondiente de la clave privada de la elección a los trustees, etc.

### 6.3.1. Esquema de la base de datos

En la figura 6.4 se muestra el diagrama entidad-relación del sistema Helios Voting, con las modificaciones realizadas en la implementación del sistema de este PFC.

En el esquema se distinguen tres grupos de entidades:

- Las propias del core de Helios, cuyos nombres comienzan por *helios\_*.
- Las entidades que utilizan la herramienta Celery (las que comienzan por *djcelery\_*).
- Las entidades que utiliza la herramienta Kombu (las que comienzan por *djkombu\_*).

Con respecto al primer grupo, las que forman la capa de negocio del sistema Helios, se observa un diseño bastante simple, aunque adecuado para las necesidades del sistema:

**helios\_election** Es la tabla que almacena las diferentes elecciones que crean los administradores.

**helios\_auditedballot** En esta tabla se almacenan los votos que los votantes deciden verificar y publicar en el tablón para que puedan ser posteriormente auditados.

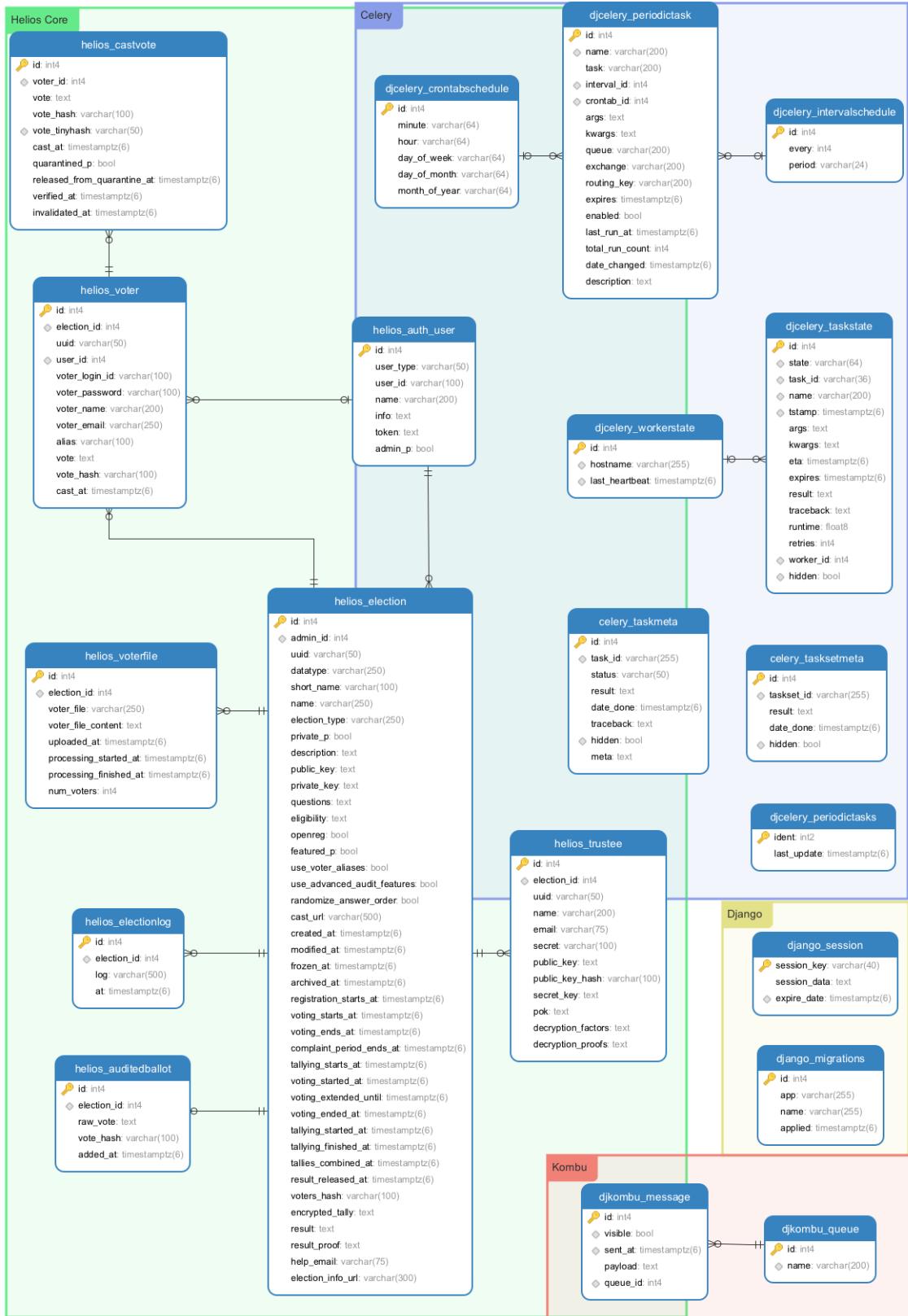


Figura 6.4: Diagrama ER del Servidor de Votación

**helios\_electionlog** Esta entidad registra los eventos que suceden en torno a una elección, ya sea su creación, congelamiento de la papeleta, apertura de la urna, escrutinio, etc.

**helios\_trustee** Es la entidad que recoge la información de los usuarios que actúan como trustees de la elección. Entre la información de estos que se registra se incluyen las claves públicas que se almacena automáticamente al dar de alta a un trustee. También la clave secreta, que el trustee habrá de subir al servidor una vez se requiera para poder realizar el escrutinio.

**helios\_voterfile** Esta entidad almacena los ficheros de votantes que el administrador sube al servidor desde la interfaz web del sistema. Desde estos ficheros se carga el listado de votantes.

**helios\_user** En esta tabla se almacenan los usuarios / votantes del sistema, relacionados cada uno con las elecciones a las que tienen derecho a voto.

**helios\_auth\_user** Para cada elección, esta tabla asocia usuarios del sistema (con derecho a voto o sin él) con las capacidades de administración.

### 6.3.2. Estructura

El desarrollo del sistema de votación distribuye el código del sistema en una serie de módulos:

#### Core

Dentro del paquete *helios* se encuentra el código fuente que implementa el núcleo del sistema de votación.

Sigue el protocolo de un proyecto Django sobre el patrón MVC. Contiene las vistas, el modelo y el controlador que necesita para su funcionamiento.

En el fichero *urls.py* se definen los *endpoints* expuestos para la interacción con el sistema. Este fichero establece el punto de comienzo de la API REST que proporciona el sistema, indicando dónde deben ser realizadas las llamadas HTTP para inicio de procesos y obtención de resultados.

Este fichero funciona como un enrutador. Dirige la llamada a un endpoint hacia la vista que va a tratar esta llamada.

Estas vistas se encuentran agrupadas en el sistema en el módulo contenido en el fichero *views.py*.

Dentro del fichero *models.py* es donde se gestiona el modelo de datos.

Estas son las clases que se definen en este módulo (figura 6.5):

- **ELECTION** Clase que gestiona las elecciones que pueden ser creadas en el sistema.
- **ELECTIONLOG** Clase para gestión de los logs del sistema.
- **VOTERFILE** Clase diseñada para almacenar los ficheros de votantes utilizados para la carga del censo.
- **VOTER** Clase que define los votantes.
- **CASTVOTE** Clase necesaria para almacenar los votos emitidos por los votantes.
- **AUDITEDBALLOT** Clase que recoge los votos que han sido auditados.
- **TRUSTEE** Clase que expone los Trustees del sistema.

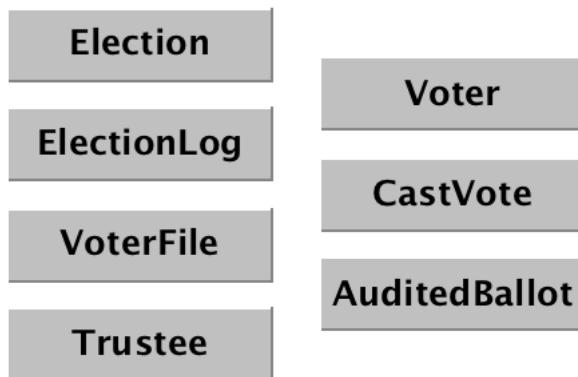


Figura 6.5: Clases del módulo central de Helios.

Se observa que no es un mapeo exacto del modelo Entidad-Relación visto en 6.3.1. Define las clases principales que interactúan para llevar a cabo la elección.

## Autenticación

El sistema de votación posee un módulo de autenticación propio que es el que gestiona las identidades y permisos de un usuario o votante frente al sistema. Este subsistema recoge la identificación del votante, sea cual sea el medio por el que se haya *logado* el usuario.

En el caso del login por DNI, este subsistema es el que maneja la comunicación con el Sistema de Autenticación (6.2).

Por tanto, bajo la carpeta *helios\_auth* se encuentra el módulo de autenticación del sistema de voto.

Aún siendo un subsistema, sigue también el patrón MVC de Django. Contiene los ficheros *urls.py*, *views.py* y *models.py*, tan característicos de este patrón en este framework.

Lo más peculiar de este subsistema es que contiene adaptadores para los diferentes sistemas de login que expone. Dentro de la carpeta *auth\_system* implementa una serie de módulos que gestionan estos diferentes métodos de login.

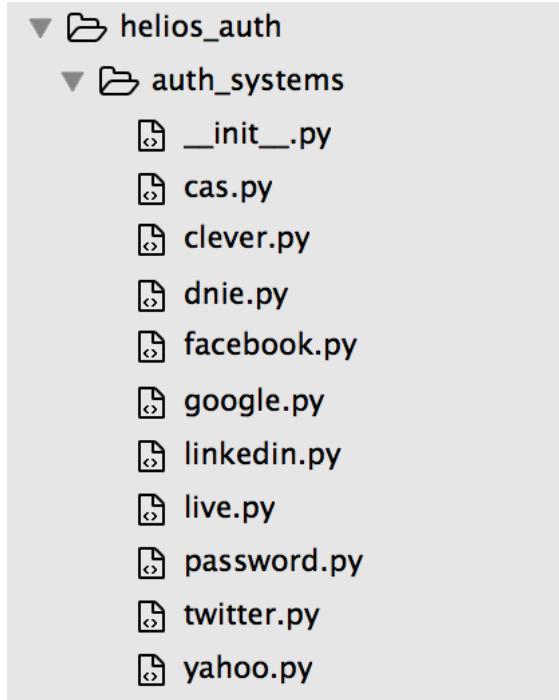


Figura 6.6: Carpeta *auth\_system* que contiene los módulos de autenticación del sistema.

Por defecto, Helios ha implementado ya los siguientes módulos para permitir la identificación federada:

- Facebook ⇌ helios\_auth/auth\_system/facebook.py
- CAS ⇌ helios\_auth/auth\_system/cas.py
- Clever.com ⇌ helios\_auth/auth\_system/clever.py
- Google ⇌ helios\_auth/auth\_system/google.py
- LinkedIn ⇌ helios\_auth/auth\_system/linkedin.py
- Twitter ⇌ helios\_auth/auth\_system/twitter.py
- Yahoo! ⇌ helios\_auth/auth\_system/yahoo.py

Además, incluye un módulo *password.py* utilizado para la identificación de usuario por medio del par {usuario, contraseña}.

- User/password ⇌ helios\_auth/auth\_system/password.py

Para este PFC ha sido necesario, pues, la implementación de un módulo nuevo para gestionar la identificación a través del DNIe.

Este módulo es *dnie.py*.

- DNIe ⇐ helios\_auth/auth\_system/dnie.py

Define una clase **DNIe** con una serie de miembros que almacenan la información del usuario recibida desde el Sistema de Autenticación, del certificado digital de autenticación del DNIe.

```

1  class DNIe:
2      def __init__(self, commonName, givenName, surname, serialNumber, c):
3          self.commonName = commonName
4          self.givenName = givenName
5          self.surname = surname
6          self.serialNumber = serialNumber
7          self.c = c
8

```

Este módulo, junto con esta clase DNIe para la gestión de la información recibida del documento, implementa las interfaces que permiten la comunicación del flujo OAuth con el Sistema de Autenticación.

```

1  def dnie_url(request, url, params)
2
3  def dnie_url_step2(request, url, params)
4
5  def dnie_get(request, url, params)
6
7  def dnie_post(request, url, params)
8
9  def dnie_post_step2(request, url, params)
10
11 def can_create_election(user_id, user_info)
12
13 def get_auth_url(request, redirect_url = None)
14
15 def get_user_info_after_auth(request)
16
17 def get_user_info_after_auth_androidClient(request)
18
19 def get_dni_info_from_ssl(request)
20
21 def do_auth(request)
22
23 def do_logout(request)
24

```

Las funciones de este módulo se alimentan de la información que se recibe en el request de la petición HTTP.

Se puede observar en la definición de estas funciones que son las que llaman al Sistema

de Autenticación OAuth o acaban siendo llamadas por los callbacks que a aquél se pasan en el flujo de comunicación que implementa este protocolo.

Un aspecto técnico a tener en cuenta es que se ha tenido que variar la forma de tratar la recogida de información si accedemos al sistema desde un dispositivo móvil o desde un navegador de escritorio. Esto es así porque en este último, la gestión de las cabeceras HTTP la realiza el navegador, mientras que en el dispositivo Android es preciso que la app que realiza el baile de obtención de certificados del documento y token del sistema de autenticación es la que debe pasar la autorización y la información al navegador web del dispositivo.

### Cabina de votación

En la carpeta *helios\_booth* se encuentra el componente de la Cabina digital de Voto.

Es un módulo autocontenido cuya funcionalidad se desarrolla en tan sólo un elemento HTML, apoyado en código Javascript para controlar la interacción con el usuario, el cifrado de los votos y el envío del voto al Sistema de Votación en servidor. La lógica de este módulo se ejecuta en cliente, por eso es importante la comunicación final con el servidor.

Su estructura y funcionamiento se explica en 6.6.4.

### Verificador de voto

El verificador de voto es una herramienta para que un votante u observador pueda verificar que el sistema cifra y descifra correctamente un voto.

### Elementos estáticos

Esta carpeta contiene los ficheros estáticos necesarios para la interfaz web, incluyendo HTML, Javascript y hojas de estilo CSS.

## 6.4. Cliente Android

El cliente del sistema se apoya en el navegador web que utilice el usuario para interactuar con el sistema, principalmente contra el subsistema de votación, pero también contra el de autenticación.

Ambos sistemas son aplicaciones web, por lo que esta interacción se realiza a través de páginas web que permiten al usuario comunicarse con los servidores.

Como cualquier aplicación web, ambos sistemas sirven sus propias páginas o APIs para que puedan ser renderizadas por el navegador web del usuario y que éste pueda realizar las acciones oportunas.

No obstante, debido a la necesidad de uso de certificados digitales de cliente como herramienta para la identificación digital, se hace necesario incluir una capa de comunicación entre el navegador y estos certificados.

En el caso de la navegación desde un PC, esta capa intermedia son los drivers del DNIe, que gestiona el propio Sistema Operativo, con los que, a través de un lector del chip de contacto, accede a los certificados y los presenta, a través de su Almacén de Certificados, al navegador para que los tenga accesibles.

Esto no es posible, sin embargo, en dispositivos Android. No existen drivers liberados para acceder a los certificados del DNIe 3.0 por NFC desde el Sistema Operativo Android y comunicarlo con el navegador web, al contrario de los que existen para Windows, Linux o MacOS. No es posible por ahora, por tanto que un navegador web en Android (ni iOS) tenga acceso a los certificados como ocurre en un navegador de un equipo de escritorio.

La idea original era acceder al sistema de votación con un login con los certificados digitales directamente contra el servidor de votación, que sería el que estaría protegido y esperaría estos certificados de cliente.

En el capítulo 5.5.4 se explica la evolución experimentada para la solución de este problema, pues no era posible llevar a cabo la primera idea y hubo que estudiar diversas opciones hasta encontrar con la que finalmente se implementó. Esta consiste en la implementación de una app Android que haga de intermediaria entre el navegador web del dispositivo y, por un lado, la lectura por NFC del certificado de autenticación del DNIe 3.0 y, por otro lado, con su uso contra el Sistema de Autenticación, para, al final del proceso, dirigir al usuario al propio navegador web con el acceso al Sistema de Votación ya validado.

#### 6.4.1. Código fuente

El código fuente de la app original de la DGP ha debido ser modificado ya que la funcionalidad que se busca que tenga la aplicación no es la misma que la que ofrece la original.

En la app original, al entrar se nos ofrece un menú radial con diferentes opciones. Esto no ocurre en la versión de este proyecto. La app en nuestro caso está pensada para realizar tan sólo una función, realizar la identificación y autenticación de usuario contra un servidor OAuth y transmitir al sistema de votación la aceptación o revocación de acceso del mismo contra el sistema.

Por ello, la app va a carecer de un menú inicial, ya que no es necesario.

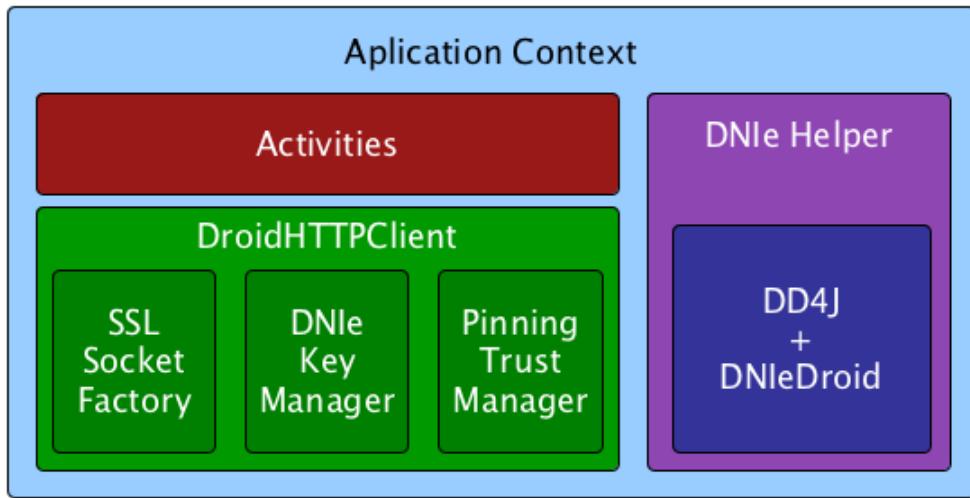


Figura 6.7: Arquitectura lógica de la App sobre DNIeDroid [18]

Una vez se abra la app, ya sea desde el menú del dispositivo o desde el botón de login de la página inicial de la web del sistema de votación, aparecerá el menú de selección del CAN del DNIe que queremos utilizar, o, si es la primera vez que cierto usuario utiliza la app, la posibilidad de añadir un nuevo DNIe con su CAN correspondiente.

#### 6.4.2. App Android de Autenticación

Ya se ha explicado que para realizar el acceso al sistema de voto desde dispositivos móviles, se ha optado por modificar una de las apps de ejemplo publicadas por la Policía española.

La app original mostraba un ejemplo de funcionamiento en el que se accedía a varias webs que necesitaban acceso con identificación por DNIe. Las modificaciones que se han realizado a esta app han consistido en:

- Eliminar el menú inicial con el que se podía elegir el servicio al que se pretendía acceder.
- La app ahora sólo tiene una funcionalidad, conectarse contra el servidor de autenticación de nuestro sistema de votación.
- Para ello, se ha modificado el software cambiando el flujo de llamadas. Originalmente sólo se hacía una llamada a una web para obtener información. Para cumplir con el protocolo OAuth, se ha incrementado el número de llamadas HTTPS de la app, con una mayor comunicación bidireccional entre el servicio y el dispositivo.
- La app embebe keystore con los certificados de servidor y el raíz de la Policía para cada uno de los servicios que utiliza. Se han eliminado y se ha incorporado una

keystore con los certificados del servidor de autenticación y el de la AC raíz de la Policía.

- La app posee una lista de servidores en los que confía. En esta lista se incluyen los hashes de los certificados de servidor de los servicios que consulta. Se han eliminado los elementos de esta lista y se ha añadido el hash del certificado del servidor de autenticación, que es contra el que ahora se va a comunicar la app.
- En la app original, se realiza una llamada HTTPS, se obtiene la información y se muestra en una página estática dentro de un WebView. Los WebViews en Android no soportan navegación manteniendo certificado de cliente, por lo que no es útil para la nueva funcionalidad de nuestra app. Por ello, se ha eliminado esta forma de mostrar información. Ahora se realizan las comunicaciones del flujo OAuth y cuando el usuario ha sido identificado y autenticado, se redirige a la web del sistema de voto por medio de un navegador externo a la app.

La comunicación puede comenzar de dos formas:

1. El usuario entra con un navegador móvil en la web de la elección (figura 6.8). Esta le muestra un link para realizar el login seguro a la misma. El sistema de voto identifica con Javascript que el usuario accede desde un dispositivo Android, con lo que modifica el link para que se ejecute un intent, la intención de abrir una app en el dispositivo, en lugar de redirigirse a una web. Este link modificado se corresponde con

```
1  intent://scan/#Intent;scheme=dnie;package=com.dnieadmin;end
```

En este enlace, *scheme=dnie* y *package=com.dnieadmin* son valores definidos en el *AndroidManifest.xml* del proyecto. Al utilizar el protocolo *intent://* se indica al dispositivo que se está *intentando* utilizar una aplicación.

Si la app se encuentra instalada, se abrirá. En caso contrario, se dirige al usuario a la Play Store de Google para que se descargue la app.



Figura 6.8: Página de inicio en navegador de dispositivo Android.

2. Directamente, el usuario entra en la app desde el menú de aplicaciones (figura 6.9) del Launcher del Sistema Operativo.

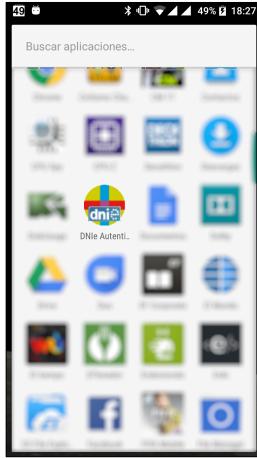


Figura 6.9: Menú de aplicaciones de dispositivo Android

En este momento, cualquiera que sea el camino que se ha seguido, el usuario entra en la app. Con respecto a la app original, se ha eliminado el menú principal y todas las funcionalidades que aportaba. Directamente, se muestra la pantalla de introducción o selección del CAN (figura 6.13). El CAN es un identificador de 6 dígitos que aparece en el anverso del documento físico del DNIe. Se utiliza para establecer un canal cifrado PACE entre el documento y el dispositivo.



Figura 6.10: App Autenticación Android: Selección del CAN cuando no hay ningún documento guardado previamente.

Este canal PACE es obligatorio a la hora de establecer una conexión inalámbrica cifrada, pero no sustituye al PIN en ningún caso, igual que no da permiso de uso de las claves de autenticación y/o firma del ciudadano. Es una herramienta más de seguridad para evitar que un tercero pueda leer información del DNIe sin conocimiento del ciudadano dueño del documento.



Figura 6.11: *App Autenticación Android: Pantalla para introducir el CAN.*

Una vez introducido el CAN por primera vez, ya no será necesario volver a hacerlo, ya que se almacena en el sistema y puede seleccionarse el apropiado cada vez que el usuario hace uso de la aplicación. Al igual que se pueden guardar más documentos, es posible eliminar el CAN de documentos ya introducidos. (figura 6.12)

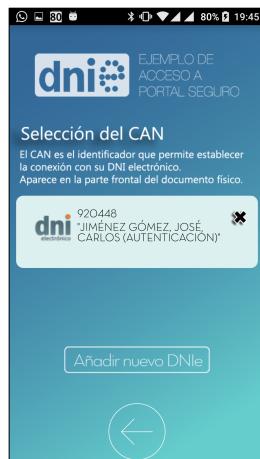


Figura 6.12: *App Autenticación Android: Selección del CAN.*

En caso de intentar una lectura de la información de un DNIe introduciendo un CAN incorrecto se produciría un error al establecer el canal PACE y no sería posible la comunicación entre documento y dispositivo (figura 6.13).



Figura 6.13: App Autenticación Android: Error al introducir un CAN incorrecto.

Cuando se ha indicado el CAN correcto del documento que se va a utilizar para logarse en el sistema, la aplicación solicita al usuario que acerque su documento físico al dispositivo para que este pueda acceder a los certificados (figura 6.14).



Figura 6.14: App Autenticación Android: Esperando que se aproxime el DNIe 3.0.

La comunicación entre documento y dispositivo se realiza mediante NFC (2.6.3). Según el documento [18], los desarrolladores de la app de ejemplo indican que *"por motivos de hardware y de seguridad, la distancia máxima soportada por NFC es, dependiendo del dispositivo, de alrededor de un centímetro"*. De todos modos, es una comunicación ciertamente sensible, que se ve afectada por la distancia y el movimiento, necesitando para un correcto funcionamiento cercanía y una notable estabilidad.

Una vez estén cerca documento y dispositivo, se detectarán y establecerán la comunicación cifrada entre ambos (figura 6.15), a través de la cual podrán intercambiarse tanto comandos como respuestas a estos, accediendo a la información alojada en los certificados y permitiendo a estos firmar información presentada por el dispositivo (aunque en nuestra app no hará falta utilizar el certificado de firma digital).

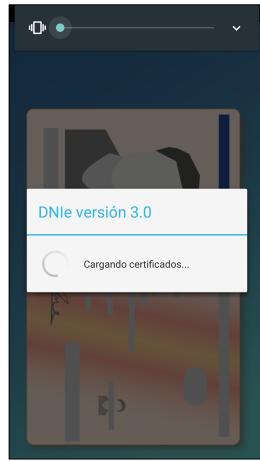


Figura 6.15: App Autenticación Android: Cargando certificados.

Para poder llegar a utilizar los certificados, no obstante, primero se solicitará el PIN al usuario (figura 6.16), con el cual se autentica al mismo contra el DNIE y se le conceden permiso de acceso a las claves. A partir de este momento, la comunicación entre documento y dispositivo va cifrada y se pueden acceder a los certificados.

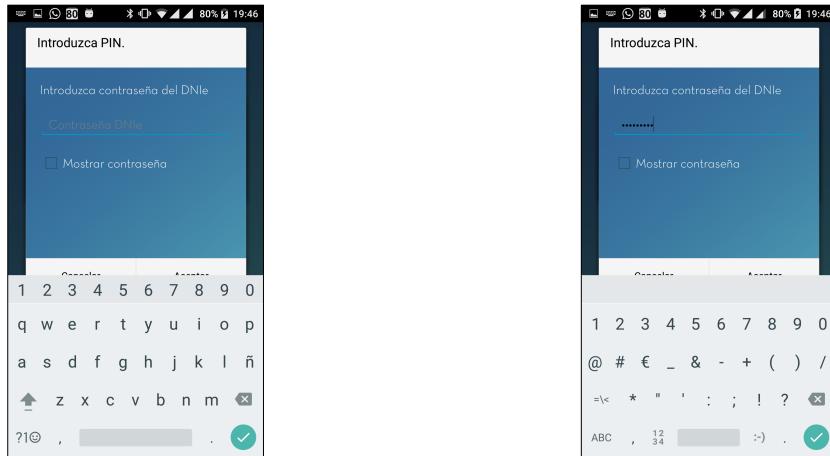


Figura 6.16: App Autenticación Android: Introducción del PIN del DNIE 3.0

Es en este momento cuando la app utiliza el certificado de autenticación del DNIE realizar una petición HTTPS contra el servidor de autenticación del sistema de voto. Cada una de las llamadas que le exige el protocolo de comunicación OAuth2 se realizará de forma segura, con los certificados del servidor y el de autenticación del documento (figuras 6.17 y 6.18).

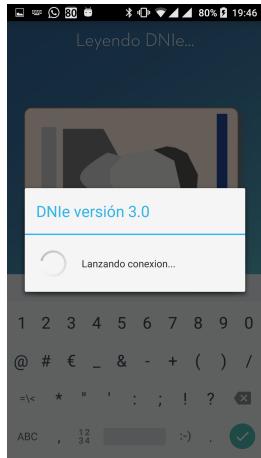


Figura 6.17: App Autenticación Android: Lanzando conexión segura con los certificados.

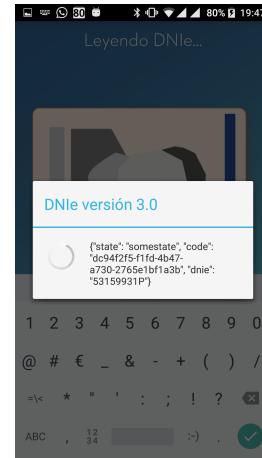


Figura 6.18: App Autenticación Android: Recepción de código oAuth.

Para poder establecer un canal seguro contra el servidor de autenticación, es necesario, según la implementación de la app original de ejemplo, que el certificado del servidor y la AC raíz vayan embebidos en el código de la app. Así mismo, es necesario que se incluya el hash del mismo para que el objeto DroidHTTPClient que establece la conexión al servidor sea capaz de establecer este camino seguro.

Esta keystore con los dos certificados se debe incluir en el proyecto Android con un formato BKS, lo cual se consigue realizando los pasos expuestos en el anexo C.

Una vez obtenido el fichero keystore que contiene los dos certificados, ha de colocarse en la carpeta del proyecto:

```
1 clienteAndroidDNIEoAuth2/app/src/main/res/raw
```

Para añadir el hash del certificado del servidor habrá que editar también el fichero:

```
1 clienteAndroidDNIEoAuth2/app/src/main/res/values/trusted_hosts.xml
```

Este hash se obtiene en el servidor con el comando que se indica igualmente en el anexo C.

Así, el fichero que indica qué servidores son confiables y que utiliza la app para poder establecer una conexión segura con el servidor quedaría parecido a este:

```
1 <resources xmlns:android="http://schemas.android.com/apk/res/android">
2   <array name="trusted_hosts">
3     <!-- En el servidor: openssl x509 -fingerprint -noout -in server.crt -->
4     <item>D5E6E3759023DE9B881399DC034E8FE1AE7D9D87</item>
5   </array>
6 </resources>
```

El servidor de autenticación establece un canal seguro con su certificado (en el prototipo será autofirmado) pero valida el certificado de autenticación con el certificado AC Raíz de la Policía que posee y que se ha configurado para ser utilizado. Por esta razón aparece un error en el navegador cuando se entra por primera vez a la web (figura 6.19). Avisa al usuario de que no encuentra en su almacén de certificados uno que sea confiable con el servidor. En nuestro caso, para el prototipo, bastará con hacer click en *CANCELAR*

ya que no vamos a instalar certificados adicionales. En caso de implementar el sistema en producción, será necesario obtener e instalar en el servidor un certificado confiable, emitido por una autoridad de confianza.

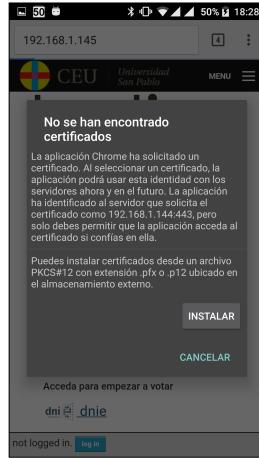


Figura 6.19: App Autenticación Android: Los certificados autofirmados no son seguros, pero la comunicación va cifrada de todos modos.

Una vez asumimos el certificado autofirmado, por fin accedemos a la web del sistema de voto logados con la información contenida en el certificado de autenticación de nuestro DNIe 3.0 (figura 6.20).

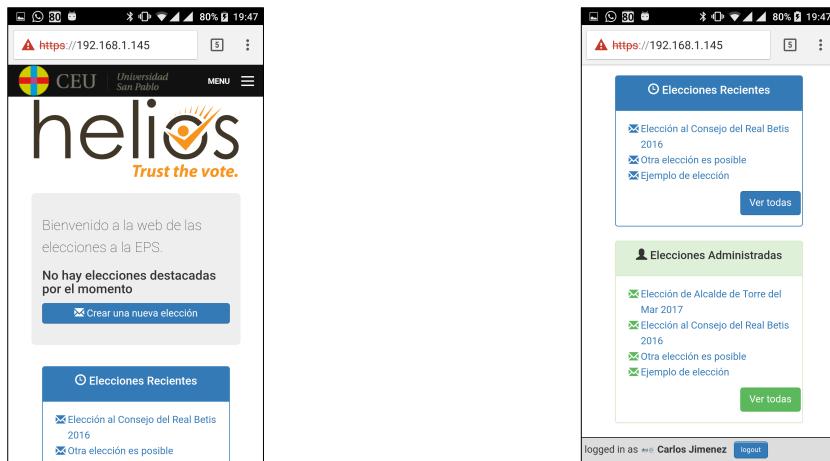


Figura 6.20: App Autenticación Android: Acceso al sistema de voto con login con el DNIe.

## 6.5. Topología de red

De cara a la topología de la red del sistema, se ha diseñado la integración para abordar tres configuraciones diferentes dependiendo de las necesidades de la elección:

1. **Servidor de autenticación y Servidor de votación en la misma red. Acceso de los votantes desde Internet.** Figura 6.21

Esta topología requiere port forwarding del router que comunica la red de los dos servidores con Internet para que, a través del mismo endpoint, los dispositivos se puedan conectar con los servicios de ambos servidores.

Las consultas OCSP para validación del DNIe se realizan a través de Internet.



Figura 6.21: Topología de la red con los servidores compartiendo red.

2. **Servidor de autenticación y Servidor de Votación en redes diferentes. Acceso de los votantes desde Internet.** Figura 6.22

Esta topología no requiere port forwarding, pues cada servidor tiene su propio acceso a/por Internet. Los dispositivos accederían directamente a la IP/URL de cada servidor y utilizarían sus diferentes servicios utilizando el puerto correspondiente.

Las consultas OCSP para validación del DNIe se realizan a través de Internet.

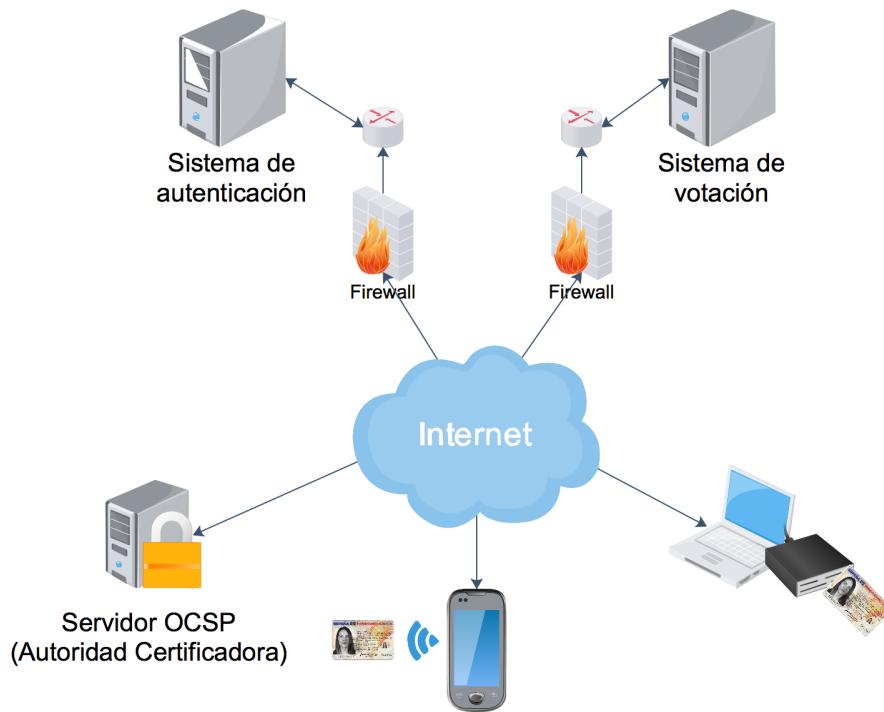


Figura 6.22: Topología de la red con los servidores en redes diferentes.

3. Tanto el Servidor de autenticación, como el Servidor de Votación y los dispositivos se comunican por la misma red, sin necesitar acceder a través de Internet. Figura 6.23

Haría falta un acceso a Internet para poder realizar consultas OCSP de validación del DNIe.

Esta topología de red interna no cumple con los requisitos de voto e identificación remotos a través de Internet definidos en 5.1.3, pero es una buena solución para elecciones internas o para la presentación de un prototipo.

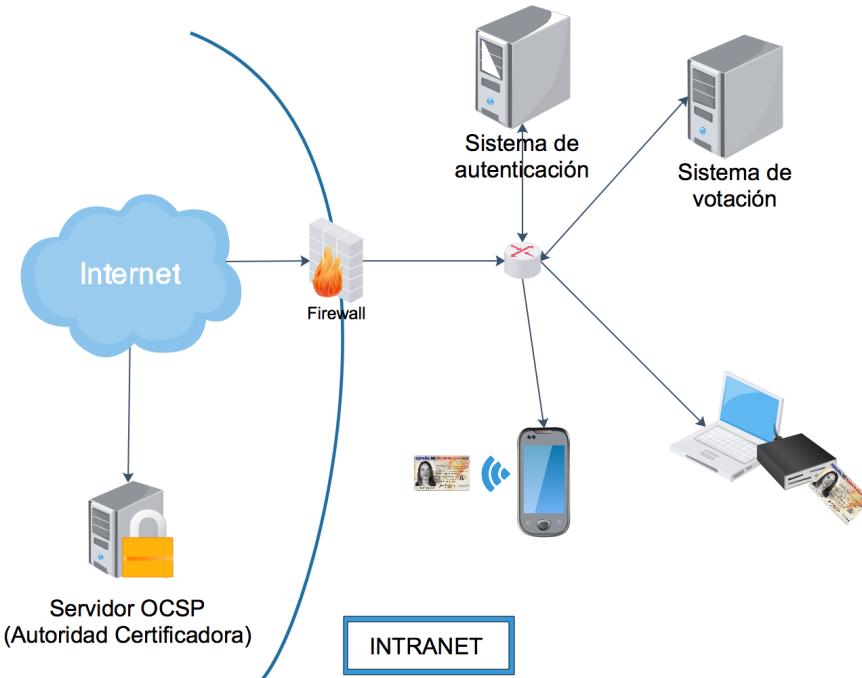


Figura 6.23: Topología de la red con los servidores y los dispositivos en red privada.

## 6.6. Esquema de votación

### 6.6.1. Registro

La fase de registro de votantes en el sistema no será interactiva en cuanto a que no es el propio votante el que debe inscribirse para poder votar en las elecciones, sino que es la Autoridad Electoral quien lo registra en el censo. En esta fase, pues, se trata de establecer el censo de votantes que tienen autoridad para votar en el proceso electoral.

Como se advierte en el análisis se ha tomado en consideración que sean los administradores del sistema quienes tengan la responsabilidad sobre el tratamiento del censo, por lo que se ha de cargar en el sistema y éste es el que lo va a tratar.

El censo ha de cargarse en dos servicios del sistema, tanto en el servidor de autenticación como en el sistema de votación.

El censo del subsistema de autenticación se utilizará para llevar el control de los votantes que tienen derecho de acceso al sistema, por lo que a los votantes se les pueden añadir otros usuarios necesarios para llevar a término la votación, como pueden ser administradores o auditores, aunque estos no tengan derecho de voto. Así se conformaría la base de usuarios activos del sistema.

En el subsistema de voto también se vuelca el censo para cada uno de los diferentes procesos de voto que conformen la elección.

El procedimiento a seguir consistirá en que la Autoridad Organizadora del Proceso Electoral, la Universidad, proveerá una lista del censo a los administradores del sistema. El administrador utilizará la función de carga de votantes con la lista proporcionada para realizar la carga inicial de votantes para cada una de las subelecciones que se configuren.

La lista proporcionada por la Universidad debe contener la siguiente información de cada uno de los votantes:

- Nombre y apellidos
- DNI
- Clase / grupo de votantes
- E-mail

La carga de los votantes a través de su aplicación se realiza subiendo un fichero csv con la información requerida. Este fichero se pone a disposición de la cola de procesos, la cual, llegado el momento volcará cada uno de los registros en la base de datos del sistema de votación.

### 6.6.2. Identificación / Autenticación

El servicio de identificación es un subsistema clave en el proceso electoral. En él recae parte de la responsabilidad de la robustez del sistema, en cuanto a que debe asegurar varios de los requisitos básicos que definen el voto electrónico en concreto:

**Solidez:** Debe asegurar que un votante deshonesto no tenga capacidad de acceder al sistema e interrumpir la votación. Es decir, que sólo debe dar acceso a los votantes que realmente deben ingresar al sistema de votación.

**Elegibilidad:** Este requisito implica que el sistema debe controlar que ningún votante que no tenga permitido el voto pueda votar. Aunque es el proceso de votación el que debe controlar esta circunstancia cuando un usuario trata de emitir un voto, el sistema de votación, de forma análoga al requisito anterior, también debe proteger el sistema evitando el acceso a aquellos que, directamente, no tengan permisos para votar.

**Sin duplicados:** El sistema debe evitar que un votante duplique o reemplace el voto de otro. Igualmente, aunque es el sistema de votación el que debe tener mecanismos que controlen esta situación, la primera barrera debe ser la servicio de autenticación del votante.

El sistema de identificación del votante se apoya en el protocolo OAuth2.

Como ya se ha comentado, el sistema de identificación/autenticación federada que se va a utilizar se basa en un proyecto publicado como software libre por el desarrollador Richard Knop con nombre django-oauth2-server.

Este servidor está escrito en Python y se ha desarrollado sobre el framework Django, el mismo sobre el que está desarrollado el propio sistema de votación, Helios Voting.

El proyecto original no se adapta a las necesidades del sistema que se quiere integrar, pues no soporta certificados digitales. Esta es, pues, una de las funcionalidades que hay que agregar para poder integrar ambos sistemas.

El primer paso para la implementación del servidor consiste en configurar el framework Django para que se apoye en un servidor web externo. Para este proyecto se ha utilizado el servidor web Apache.

Es necesario instalar dos módulos específicos:

- **mod\_wsgi** : Módulo que permite que una aplicación basada en Django corra sobre un servidor web externo, como Apache.

- **MOD\_SSL** : Módulo que permite a ciertos servidores web utilizar certificados digitales como forma de autenticación. En el caso de nuestro proyecto, servirán tanto para establecer los canales SSL/TLS como para incluir el certificado digital de autenticación del DNIe en el canal seguro, recuperar la información de usuario del certificado y pasarla al código del servidor.

Es imprescindible que el servidor web esté configurado para aceptar tan sólo comunicaciones seguras.

Otra configuración básica del servidor será que requiera que las comunicaciones con el votante, para su autenticación, deban realizarse presentando éste los certificados de su DNIe.

Se puede ver la forma en que ambas configuraciones se han realizado en el prototipo (6.8) en el anexo B.

### Flujo de comunicación oAuth

El flujo del proceso de identificación del usuario a través del protocolo oAuth2 modificado se puede observar en la figura 6.24, donde se esquematizan los siguientes pasos:

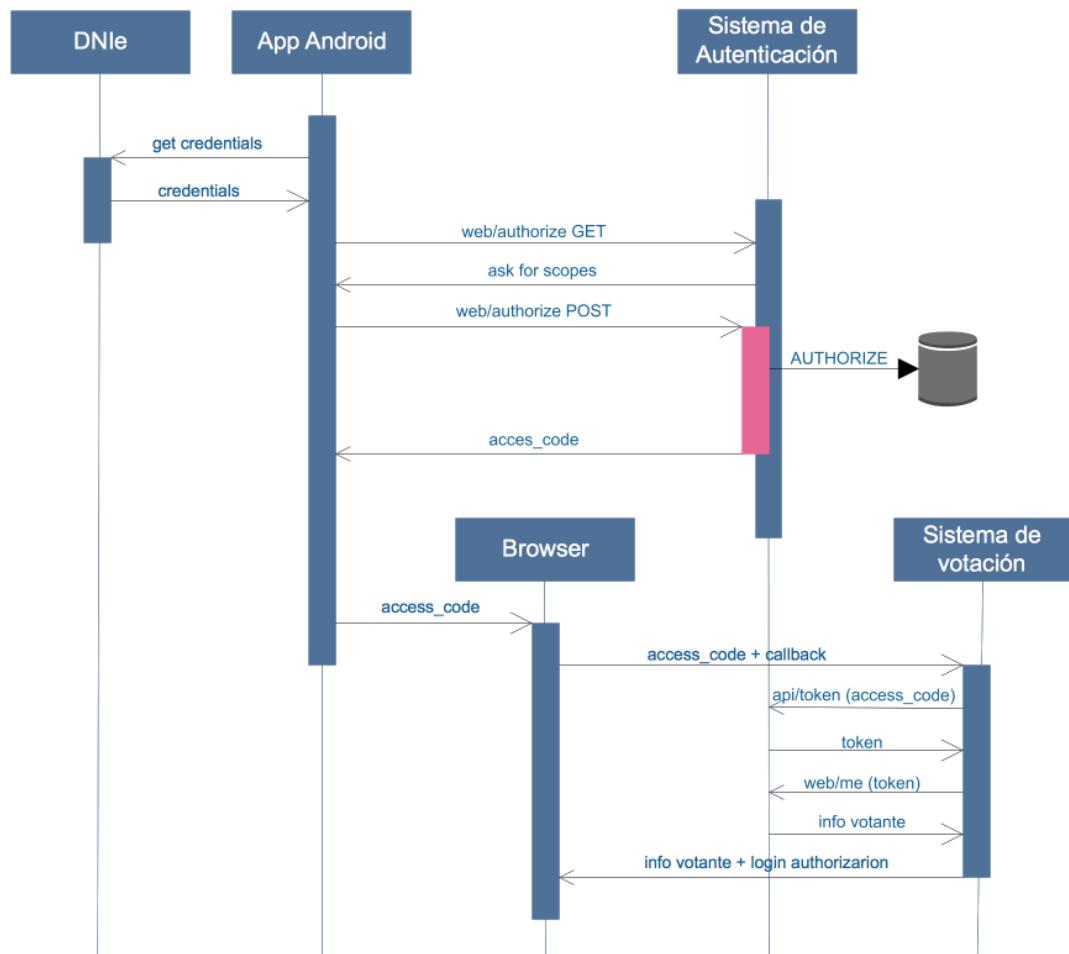


Figura 6.24: Flujo oAuth para servicio de autenticación

1. La app del DNIe obtiene las credenciales del chip de identificación del documento por NFC.

2. La app realiza una petición GET a la web del Sistema de Autenticación.
3. El Sistema de Autenticación devuelve un formulario de aceptación con la información de los *scopes* de autorización.
4. La app Android realiza una petición POST con los *scopes* aceptados por el usuario, dando aquí su conformidad a que la aplicación web de votación tenga acceso a información sobre el usuario.
5. El Sistema de Autenticación consulta la Base de Datos de censo de votantes y usuarios y, si permite el acceso al usuario, devuelve un *access code*.
6. La app del DNIe lanza un *intent* de Android para abrir el navegador web por defecto del dispositivo en la página de inicio del Sistema de Votación, pasándole además el *access code*. Aquí el entorno del dispositivo termina en la app y pasa al navegador web.
7. Desde el navegador, se lanza una petición de acceso al Sistema de Votación con el *access code* y una URL de *callback* como parámetros.
8. El Sistema de Votación realiza una petición a la URL del Sistema de Autenticación que va en el *callback* (/api/token) con el *access code* como parámetro.
9. El Sistema de Autenticación genera y devuelve un *token* al Sistema de Votación.
10. Con el *token* recibido, el Sistema de Autenticación asegura la identidad del usuario y lo utiliza para poder información al Sistema de Autenticación. Por ejemplo, con una petición al endpoint /web/me del Sistema de Autenticación...
11. Éste le devuelve información acerca del usuario.
12. Una vez que el sistema ha recibido el *token* y la información de usuario del votante, considera probada la identidad digital de este, por lo que se le considera *logado* en el sistema y se le permite la interacción con éste.

### 6.6.3. Definición de la papeleta

El principal cambio que se ha implementado en la tarea de definición de las opciones de la papeleta ha sido el de añadir automáticamente la opción del voto en blanco a la papeleta.

Ahora, cada vez que un administrador cree una consulta en una papeleta, junto con las opciones que defina, automáticamente se adjuntará la opción del voto en blanco.

Esta modificación trata de satisfacer el requisito **RE.1** definido entre los requisitos específicos (5.1.7).

### 6.6.4. Votación

Para la votación, el sistema implementa una *Cabina de Votación* lógica. Es un módulo que se ejecuta en el cliente y no en el servidor.

El sistema se encarga de esta forma de proteger el secreto del voto y la privacidad del votante. El voto se cifra en el navegador del cliente antes de ser enviado al servidor.

El sistema lo recibe encriptado y lo guarda así en la base de datos (urna digital).

En la totalización (6.6.5), el sistema utilizará técnicas criptográficas avanzadas para combinar todos los votos cifrados y descifrar simplemente la suma de ellos. En ningún momento del proceso electoral el sistema maneja un solo voto descifrado de forma individual.

El módulo de votación es un fichero HTML, vote.html, orquestado por código Javascript, escrito dentro del mismo fichero entre etiquetas <script></script>(ver figura 6.25).

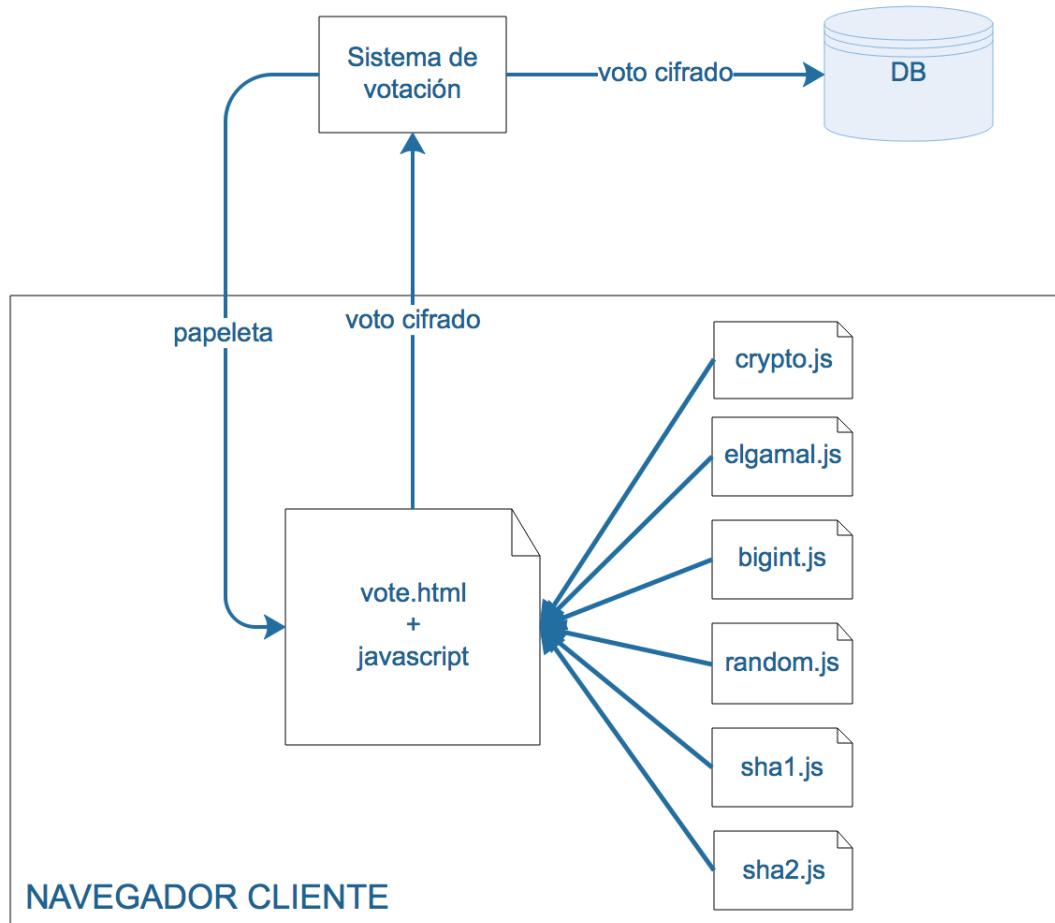


Figura 6.25: Esquema de la Cabina de Votación

Además, incluye varias librerías Javascript externas, como son:

- **jQuery** Librería que contiene código que extiende la funcionalidad de Javascript realizando más intuitiva la gestión del DOM. Incluye código para gestionar plantillas y estilo para la web. Es una librería muy utilizada en el mundo del desarrollo Web.
- **bigint.js** Implementación de un wrapper para la clase java.math.BigInteger. Necesaria para la gestión de los números grandes utilizados para los diferentes cífrados, tanto como generadores, elementos, semillas o retos.
- **random.js** Generador de números aleatorios.
- **elgamal.js** Implementación en Javascript del algoritmo de cifrado del ElGamal.
- **sha1.js, sha2.js** Implementación en Javascript de los algoritmos de cifrado SHA-1 y SHA-2.

Durante todo el proceso de votación, aunque éste se divide en tres pasos, el votante se encuentra en la misma página *vote.html*. La orquestación entre una fase y la siguiente se gestiona a través de código Javascript.

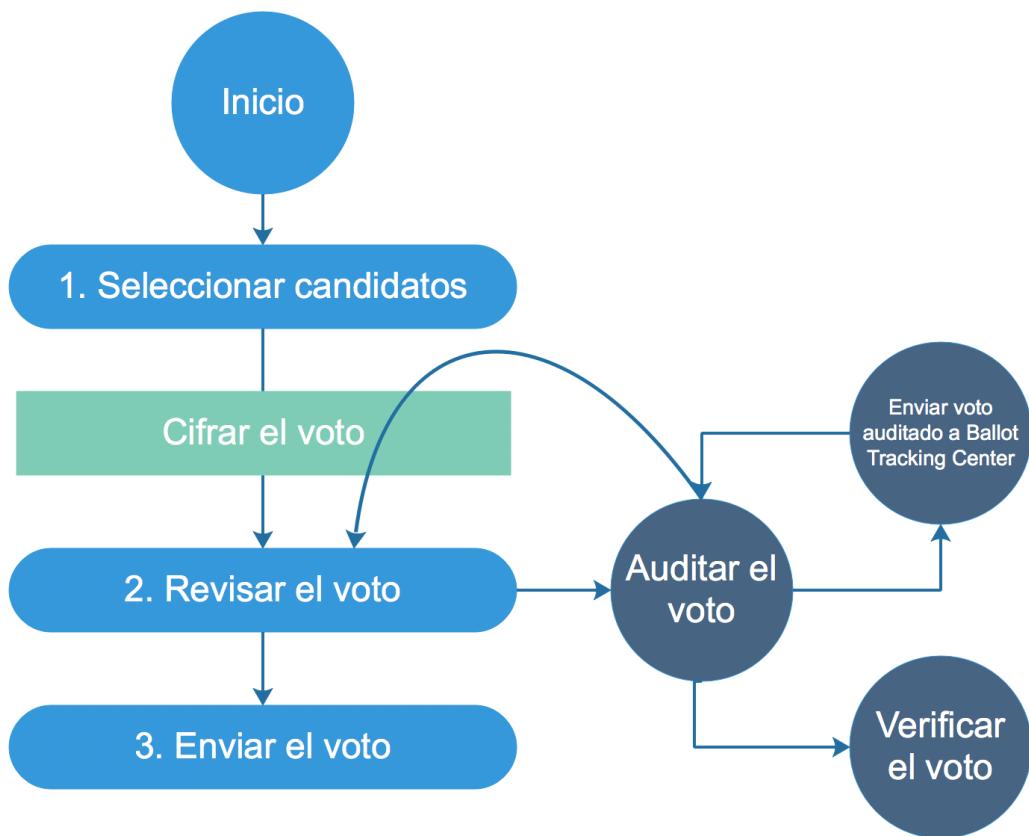


Figura 6.26: Diagrama de estados del proceso de votación

### Inicio

Se muestra al votante un explicación del proceso que va a realizar para votar, con una enumeración de las fases que proceden.

Se incluye un botón para comenzar a votar.

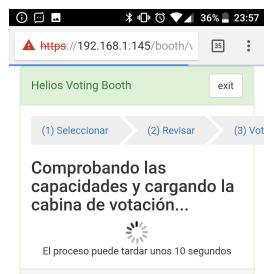


Figura 6.27: Carga de la cabina de votación.

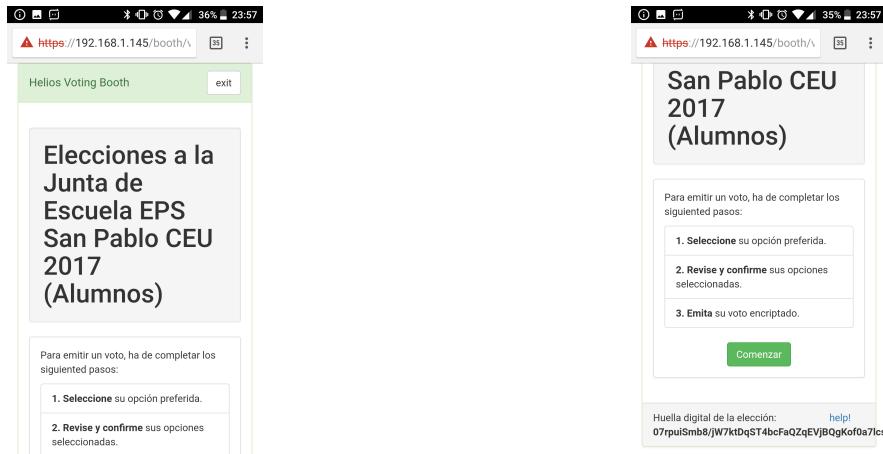


Figura 6.28: Inicio de la cabina de votación.

### Paso 1. Selección de candidatos

Se muestra al votante los candidatos para cada una de las preguntas que la elección le permite votar. Dependiendo de la configuración de la elección y de cada una de las preguntas, el votante podrá no elegir ningún candidato, elegir sólo 1, un rango de candidatos, abstenerse o votar en blanco.

En la zona inferior se le indica al votante cuál es la Huella Digital de la Elección en la que está votando. Esta huella es el hash del JSON que define la propia elección. Por tanto, la huella depende de la lista de votantes censados o registrados. Así, en caso de modificarse el censo una vez que se ha abierto la votación, aquí hay una herramienta para detectarlo, ya que esta huella no se correspondería con la que se genera una vez modificada la elección.

Tras seleccionar los candidatos, el votante puede pulsar el botón de continuar. Ver figura 6.29.

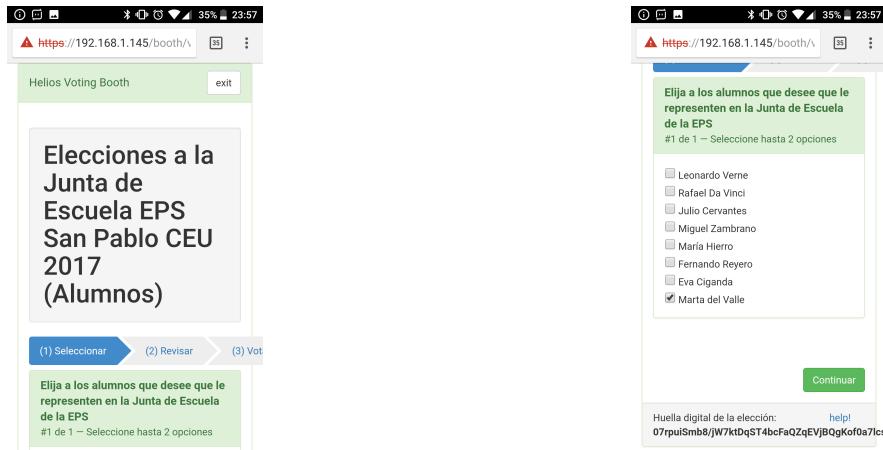


Figura 6.29: Elección de candidatos

### Cifrado del voto

Entre el paso 1, donde se eligen los candidatos, y el paso 2, donde se revisa y audita el voto, el navegador cifra la papeleta.

A continuación se muestra el esquema de un voto cifrado:

```

1 {
2   "answers": [
3     {
4       "choices": [
5         {

```

```

6      "alpha": "11821738026205240772968255127044802517047(...)",
7      "beta": "956239812618638278768915552438226298243021(...)"
8    },
9    {
10      "alpha": "89930541535253719726930300864046225791303(...)",
11      "beta": "280059812585494032480202717650783375169670(...)"
12    },
13    {
14      "alpha": "57975502776035240719836621535262618419174(...)",
15      "beta": "978382211305766369879688192639985576524767(...)"
16    }
17  ],
18  "individual_proofs": [
19    [
20      {
21        "challenge": "43874538621142379916145492966409289430255985276793233839535144670232793929873",
22        "commitment": {
23          "A": "14614624460580166084310239809768957161396(...)",
24          "B": "12129363808208853658297038681359129740187(...)"
25        },
26        "response": "35111610374554771924491600617435125565551351678779095565911261872333649419481"
27      },
28      {
29        "challenge": "17455027627200521376398379804435998722483175501173674963493255507426299982582",
30        "commitment": {
31          "A": "14276144535969498348508852984424238679819(...)",
32          "B": "54937556950085798690537717077801057528298(...)"
33        },
34        "response": "55879767585126448548489172509549767078598169937256353971492493135916950997178"
35      }
36    ],
37    [
38      {
39        "challenge": "27214371254593561525103755246527245415365098004925989240328291157531952501665",
40        "commitment": {
41          "A": "33082796917013189624560150325204466774925(...)",
42          "B": "12435741661345782068706114279439280143590(...)"
43        },
44        "response": "10558534534412804489745997990430567976230006025326885338612204248676172306992"
45      },
46      {
47        "challenge": "3411519499374933976744011752433388461738390285322076679882585342267066946683",
48        "commitment": {
49          "A": "13731698192008366840053574489091548941274(...)",
50          "B": "93495042406845358075899133494203376927794(...)"
51        },
52        "response": "5176763459334639863434268463535510901594167363643527842720657697790809305897"
53      }
54    ],
55    [
56      {
57        "challenge": "2054024664513351024850877212605480433227913183112027519913555960644095246601",
58        "commitment": {
59          "A": "10106397188055812737742462645321636338276(...)",
60          "B": "37029221908705879439687143590563027454957(...)"
61        },
62        "response": "7673773933284719536412259876677204164956289948480475588198037430892323898227"
63      },
64      {
65        "challenge": "40789319603209391044035100643943546996234603809846160735535922393763413197438",
66        "commitment": {
67          "A": "15229611588330633510585960074995900185751(...)",
68          "B": "21334632161958188873895465185349941827553(...)"
69        },
70        "response": "6852566648753632522678906133616629808473941306022210723994794012868737228377"
71      }
72    ]
73  ],
74  "overall_proof": [
75    {
76      "challenge": "1253752696342774217501410594567017563436392503337",
77      "commitment": {
78        "A": "1179324149342428195002072934744877439171442(...)",
79        "B": "8523081220255060312531704762208280291556391(...)"
80      },
81      "response": "13968935777329836215146425324634396744895179592294679261573217697722622897973"
82    }
83  ]
84 }

```

```

85 1,
86 "election_hash": "gb+MGZVlw8UmB1ljlELcfzSeYfYEAXj9h+xqcr+eE2W0",
87 "election_uuid": "4eadbc3a-bbfe-11e6-b149-b827eb1e9722"
88 }

```

El contenido de un voto completo se puede ver en el anexo D.

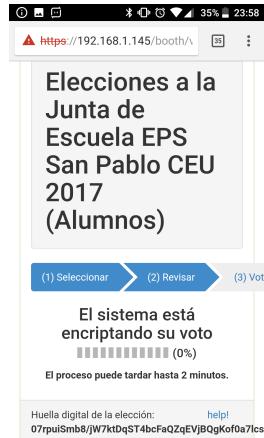


Figura 6.30: Pantalla de espera durante el proceso de cifrado del voto.

## Paso 2. Revisar el voto

Entre el Paso 1 y el 2, se realiza el cifrado del voto.

Una vez seleccionados los candidatos, el sistema nos permite realizar una revisión de nuestra elección. Así, se puede modificar el voto volviendo al Paso 1.

En este punto, se genera un código que sirve para realizar el seguimiento del voto si se decide enviar a auditoría.

En este paso, el votante tiene la opción de continuar para emitir su voto, con lo que continuaría al Paso 3, o tiene la opción de descifrar su voto y enviarlo a auditoría.

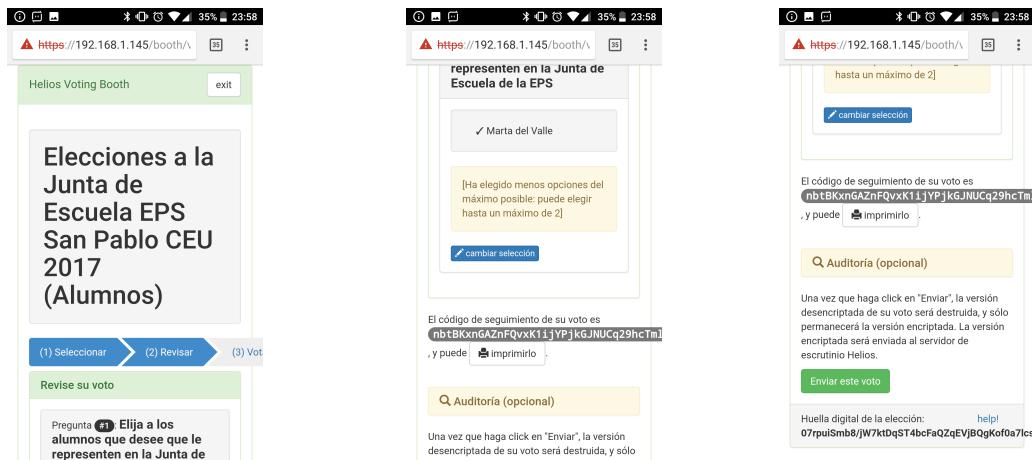


Figura 6.31: Pantalla de revisión del voto.

## Paso 3. Emitir el voto

Con el voto ya preparado y cifrado, la Cabina de Votación envía el voto cifrado al Servidor de Votación. En este momento el votante todavía tiene la opción de votar o de cancelar el voto.

Una vez emita su voto, este se guardará cifrado en la base de datos y nunca será descifrado. Se combinará con el resto de votos cifrados y por homomorfismo aditivo lo que se descifraría será la suma de todos los votos cifrados.

Este sistema de voto permite al votante votar tantas veces desee. Sólo se tendrá en cuenta su último voto, quedando los anteriores descartados y fuera del escrutinio. El motivo de este detalle es proteger al votante de un posible escenario de coacción.



Figura 6.32: Pantalla de emisión del voto.

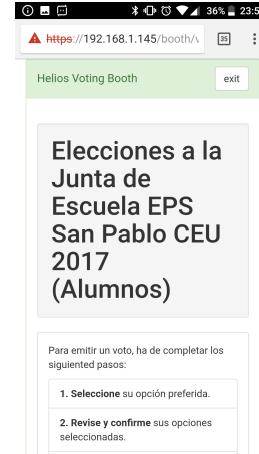


Figura 6.33: Pantalla de confirmación de voto emitido.

### Auditar el voto

El votante tiene la opción de auditar el voto. Una vez su voto ha sido cifrado, puede elegir revelar cómo se ha realizado el cifrado de los candidatos que ha seleccionado. El sistema muestra el JSON en el que se han cifrado las opciones del votante y estas pueden utilizarse en una herramienta externa para verificar este proceso de cifrado.

Este voto, que ha sido expuesto, no se contabilizará en la totalización. El votante debe volver a la Cabina de Votación para que se vuelva a cifrar su voto y pueda votar con un voto no comprometido. El proceso de evitar que un votante pueda emitir el mismo voto que audita se realiza para dar algo de protección al votante contra la coacción.

Igualmente, el votante tiene la posibilidad de enviar su voto auditado a un tablón de seguimiento de votos (Ballot Tracking Center), donde cualquier votante u observador puede realizar la verificación de ese voto.

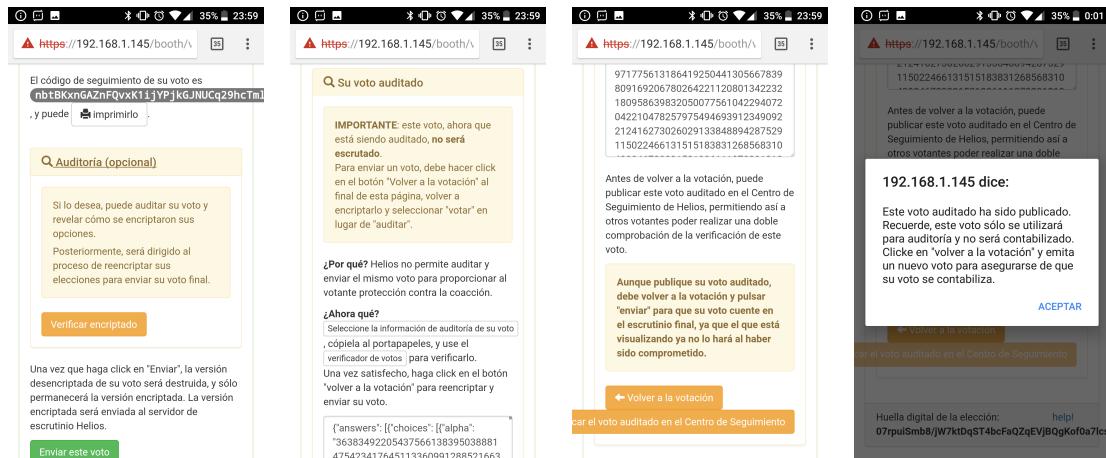


Figura 6.34: Pantalla con información para auditar el voto y publicarlo.

### Verificación del voto

La herramienta de verificación del voto permite a un votante u observador verificar que el voto fue preparado y cifrado correctamente.

Requiere como entrada de datos el JSON del voto y la URL de la elección.

El verificador hace uso de pruebas de conocimiento zero para comprobar la validez del voto y de su cifrado. Comprueba que la huella de la elección se corresponde con la huella del voto. También comprueba la validez de cada una de las opciones elegidas por el votante.

```

1   loading election...
2   election fingerprint is 4txg7mWSzN8tuZY2l7hvi7HtbdB15fZ1RRPBdwZNQMU
3   smart ballot tracker is oBeyzSaS2yi0BN+ubTUMzSPNck73+yQxZU2mSNrH/7s
4   election fingerprint matches ballot
5   Ballot Contents:
6   Question #1 - Preguntando :
7   Encryption Verified
8   Proofs ok.
9
10
11  SUCCESSFUL VERIFICATION, DONE!

```

En el anexo E se puede encontrar información acer de cómo se realiza la verificación de un voto individual obtenida de la web de documentación de Helios Voting.



Figura 6.35: Pantalla del verificador de voto individual.

### 6.6.5. Totalización

La totalización de los votos en el sistema se realiza a través de procedimientos criptográficos.

Los votos emitidos han sido almacenados cifrados en la base de datos. De ellos, han sido descartados aquellos que no fuesen el último emitido en aquellos casos en los que un votante votó más de una vez.

Por medio del homomorfismo (2.5.1.5), no es necesario que se descifren los votos para totalizar el resultado de la elección.

Los votos no se descifran, se totalizan los votos cifrados y lo que se descifra es el resultado de esta suma. Este proceso se realiza basándose en la propiedad de cifrado homomórfico aditivo que ofrece el algoritmo de cifrado de ElGamal Exponencial (2.5.1.5.2).

En el código se muestra que para cada voto final de cada votante, aquél se añade a la totalización. Una vez todos añadidos, se almacena la totalización, encriptada, a la espera de ser descifrada.

```

1 def compute_tally(self):
2     tally = self.init_tally()
3     for voter in self.voter_set.exclude(vote=None):
4         # Para cada voto final de cada votante, se añade a
5         # la totalización
6         tally.add_vote(voter.vote, verify_p=False)
7
8     # La totalización, cifrada, se almacena a la espera
9     # de ser descifrada.
10    self.encrypted_tally = tally
11    self.save()
12
13 # Class Tally
14 def add_vote(self, encrypted_vote, verify_p=True):
15     # do we verify?
16     if verify_p:
17         if not encrypted_vote.verify(self.election):
18             raise Exception('Bad Vote')
19
20     # for each question
21     for question_num in range(len(self.questions)):
22         question = self.questions[question_num]
23         answers = question['answers']
24
25         # for each possible answer to each question
26         for answer_num in range(len(answers)):
27             # do the homomorphic addition into the tally
28             enc_vote_choice = encrypted_vote.encrypted_answers[question_num].choices[
29             ↳ answer_num]
30             enc_vote_choice.pk = self.public_key
31             self.tally[question_num][answer_num] = encrypted_vote.encrypted_answers[
32             ↳ question_num].choices[answer_num] * self.tally[question_num][answer_num]
33
34     self.num_tallied += 1

```

Del código anterior se puede obtener que, efectivamente, durante el proceso de totalización (*def compute\_tally()*), el sistema se dedica a sumar resultados (*tally.add\_vote(..)*) que están cifrados y en ningún momento se dispone a descifrarlos para realizar la suma.

```

1 def tally_helios_decrypt(election_id):
2     election = Election.objects.get(id = election_id)
3     election.helios_trustee_decrypt()
4     election_notify_admin.delay(election_id = election_id,
5                                  subject = 'Helios Decrypt',
6                                  body = """
7                                     Helios has decrypted its portion of the tally
8                                     for election %s.
9                                     --
10                                    Helios
11                                     """ % election.name)

```

Tras la suma de todos los votos cifrados, es cuando se puede ejecutar la tarea de descifrado de esta suma. Como se observa en el código, es necesario que los Trustees participen descifrando su parte de la totalización (secreto compartido).

El sistema es, por defecto, un Trustee. Cuando es el único, tiene toda la totalización, así que cuando descifra su parte de ésta, está descifrando todo el resultado.

Sin embargo, cuando hay más trustees, una vez el sistema descifra su parte, avisa al resto de trustees de que deben descifrar su parte con su clave y queda a la espera de que esto ocurra.

### 6.6.6. Difusión de resultados

El último paso de la elección es el de dar oficialidad a los resultados obtenidos en el conteo. Esto se realiza en la fase de Difusión de los Resultados.

El sistema permite la difusión de resultados en dos fases:

1. Se presentan los resultados a los administradores. Sólo estos tienen permiso para ver los datos.
2. Una vez los administradores lo crean convenientes, son ellos los que publican los resultados para que puedan ser consultados públicamente.

En este sistema, el formato de presentación de los resultados es muy simple. Es un sistema tabular.

Para cada una de las cuestiones a votar se muestra una tabla con las opciones que había y el número de votos que ha obtenido.



A screenshot of a mobile web browser displaying an election results page. The URL in the address bar is <https://192.168.1.145/helios/>. The page title is "Escrutinio". A sub-header says "Pregunta #1: Elija un consejero". Below is a table showing the results for this question:

Pepe Mel	0
Poli Rincón	1
Rafael Gordillo	0
Lorenzo Serra Ferrer	0
Alfonso Pérez Muñoz	0

Figura 6.36: Tabla de resultados del escrutinio de una elección simple.

Con respecto al modelo original de Helios, en la difusión se han realizado varias mejoras:

- Se ha actualizado la interfaz gráfica de la página de resultados, cumpliendo en lo posible con el estándar W3C y tratando de dar visibilidad a los resultados.
- Se han añadido datos electorales como el del número de votantes y la abstención. Esta información era fundamental que se añadiese y en los requisitos del sistema así se estipulaba. Además de informar al observador, dota de transparencia al sistema.
- Se han añadido funcionalidades tales como poder ordenar la tabla de resultados según la columna seleccionada.
- Se ha integrado una suite de exportación por la cual los resultados se pueden exportar a diferentes formatos, como PDF, CSV ó Excel. Así se ofrece la oportunidad a los observadores que puedan obtener realizar los análisis que crean oportunos sobre los datos.

- Incluso es posible compartir los resultados por redes sociales.

## 6.7. Diseño de la interfaz de usuario

Se ha realizado una renovación de la interfaz de usuario del sistema original Helios. La interfaz original, si bien no tenía grandes problemas de visualización, ha resultado no adaptarse correctamente a los dispositivos móviles, con pantallas de menor tamaño y una forma diferente de ser utilizadas, ya que los usuarios suelen interactuar con diferentes hábitos al visitar la misma página web desde un dispositivo de escritorio o uno móvil.

Las modificaciones más importantes se han realizado sobre la estructura de construcción de las páginas y sobre el estilo que en ellas aplica.

Se ha intentado que las páginas cumplan con el estándar W3C y con reglas básicas de accesibilidad.

Para la interfaz de usuario hay varias necesidades que se han debido de satisfacer.

Por un lado, la imagen corporativa. Al tratarse de un proceso electoral diseñado para una entidad, la plataforma en la que se basa debería mostrar inequívocamente la imagen de la entidad que lo organiza.

Otro aspecto a tener en cuenta será la adaptación a dispositivos móviles, pues lo que se busca es el voto seguro desde este tipo de dispositivos, algo que no cubre con suficiencia la versión actual de Helios Voting.

### 6.7.1. Estructura de la página web

El principal reto en cuanto a las modificaciones necesarias en la interfaz del sistema original Helios han correspondido a convertir las páginas en *responsive*. Con ello, lo que se busca es adaptar la visualización de las diferentes páginas en distintos dispositivos, con diferentes recursos y distintos tamaños de pantalla, así como la capacidad de adaptarse, en un mismo dispositivo, a una situación en la que la pantalla se observa en vertical y se gira a horizontal o viceversa.

Para llevar a cabo esta tarea, nos hemos centrado en las hojas de estilo existentes a las cuales se le han realizado modificaciones. Principalmente, se han incluido reglas que aplican dependiendo del ancho de la pantalla que

### 6.7.2. Estructura de la aplicación móvil

La interfaz de usuario de la app Android necesaria para ingresar en el sistema con los certificados digitales del DNIe 3.0 no ha sido modificada ostensiblemente.

Aunque es una parte primordial del sistema y su adaptación ha requerido bastante estudio y desarrollo, realmente al final no se utiliza durante un tiempo suficiente. Por ello, se ha reutilizado la diseñada originariamente por los desarrolladores de la app original de la DGP, con las modificaciones necesarias debidas al cambio de funcionalidad de la aplicación.

### 6.7.3. Accesibilidad

Ya se ha destacado que uno de los objetivos que debe cumplir una web de un proceso electoral es la de poder llegar al mayor público posible. Para ello, una de las características que se deben tener en cuenta es la accesibilidad.

Es importante que la web sea accesible, que usuarios con diversas discapacidades tengan la posibilidad de utilizarla de forma independiente.

Detalles que se han tenido en cuenta para procurar la mayor accesibilidad del sistema han sido:

**WCAG 2.0** El diseño de la web ha sido modificado para cumplir con el máximo de pautas posibles marcadas en el estándar de accesibilidad WCAG 2.0<sup>5</sup> propuesto por el consorcio W3C.

**Colores** Los colores de la web han sido elegidos para que el contraste cumpla con lo estipulado en el estándar.

**Audio** El código HTML de la web ha sido modificado añadiendo las etiquetas e información necesarias para que se puedan utilizar sistemas de lectura de pantalla para personas con una importante capacidad visual.

## 6.8. Prototipo

Para implementar el software desarrollado en este Proyecto, se decidió montar un prototipo que cumpliera con los requisitos funcionales y diera una muestra de las capacidades del mismo a una escala reducida.

Para ello, había que desarrollar los 3 módulos de software:

- Servidor de votación
- Servidor de identificación y autenticación
- App Android de identificación con DNIe

### 6.8.1. Arquitectura física

Para montar los dos servidores, se decidió pasar de las máquina virtuales en las que se desarrollaron al uso de dos Raspberry Pi como base hardware.

Raspberry Pi es un computador de placa reducida, única o simple de bajo coste desarrollado por la Fundación Raspberry Pi, con base en Reino Unido, con el objetivo de estimular la enseñanza de la informática en las escuelas.

El lanzamiento del primero de los modelos fue a finales de febrero de 2012. Hasta el momento, Raspberry Pi ha comercializado 5 modelos de placas, desde la Raspberry Pi 1 Modelo A en 2012 hasta la Raspberry Pi 3 Modelo B, lanzada en 2016.

Los modelos con los que se ha montado el prototipo son:

**Servidor de votación** Raspberry Pi Model 3 B (fig. 6.37)

**Servidor de identificación y autenticación** Raspberry Pi Model B (Rev. 2.0, 512Mb) (fig. 6.38)

---

<sup>5</sup> <http://www.sidar.org/traducciones/wcag20/es/>



Figura 6.37: *Raspberry Pi Model 3 B*, utilizada para albergar el Servidor de votación



Figura 6.38: *Raspberry Pi Model B (Rev. 2.0, 512Mb)*, utilizada para albergar el Servidor de identificación y autenticación

En la tabla 6.1 se detallan y comparan las características de ambos modelos de hardware.

El Sistema Operativo elegido para ambos servidores es Raspbian<sup>6</sup>, una distribución de GNU/Linux basada en su origen en Debian Wheezy. Al ser un sistema basado en GNU/Linux, se infiere que es software libre y de código abierto. No obstante, para el prototipo, se ha instalado la versión estable de Raspbian a fecha de Noviembre de 2016, en la cual, el Sistema Operativo se basa en la versión de GNU/Linux Debian Jessie<sup>7</sup>, posterior a Wheezy.

### 6.8.2. Arquitectura de red

La arquitectura de red propuesta para el prototipo incluye varios elementos:

- Servidor de identificación
- Servidor de votación
- Dispositivos de votación
- Router

Los dos servidores están conectados a la misma red. Particularmente, por motivos de diferencia en los modelos de las dos Raspberry Pis en las que están montados, uno de ellos - el Servidor de Identificación - utilizará un dongle USB de conexión a la red, mientras que el Servidor de Votación hace uso de la placa de red interna de su Raspberry Pi.

En la configuración del sistema se puede optar porque el sistema funcione por Internet o en una Intranet. La idea es que el servicio sea ofrecido por Internet y así se diseñó, pero por motivos de infraestructura a la hora de presentarlo, se decidió desarrollar la posibilidad de funcionar también en una intranet.

Para ello, hay que configurar tanto el router como la app Android y los servidores. En el caso del router, para funcionar por Internet habría que abrir los puertos necesarios y realizar su pertinente redirecciónamiento a los puertos de los servidores necesarios.

Una configuración típica de esta forma de conexión a través de Internet incluye configurar estos redireccionamientos en los puertos del router:

<sup>6</sup> <https://www.raspberrypi.org/downloads/raspbian/>

<sup>7</sup> <https://www.raspberrypi.org/blog/raspbian-jessie-is-here/>

En caso de la configuración del sistema para trabajar en intranet, por motivos de infraestructura, ya sea por una presentación como la de este propio proyecto o por querer realizar una votación cerrada, la propuesta implementada incluye un router simple funcionando sin conexión a Internet.

El router utilizado para el prototipo ligero ha sido un TP-LINK TL-WR702N<sup>8</sup> (fig. 6.39). Se trata de un modelo de router muy simple y ultraportable, con lo que se manifiesta la facilidad de puesta en producción de una elección a pequeña escala que resulte asequible económicamente.



Figura 6.39: Router TP-LINK TL-WR702N utilizado para la construcción del prototipo.

---

<sup>8</sup><http://www.tp-link.com/ar/products/details/TL-WR702N.html>

Característica	Raspberry Pi Model B (Rev. 2.0, 512Mb)	Raspberry Pi Model 3 B
Fecha de lanzamiento	Q4 2012	Q1 2016
Precio <sup>a</sup>	US\$35	US\$35
SoC (System-on-a-chip)	Broadcom BCM2835	Broadcom BCM2837
CPU	700 MHz ARM11 ARM1176JZF-S core	1.2GHz 64-bit quad-core ARMv8 Cortex-A53
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p30 H.264 high-profile encode/decode, 250 MHz	Broadcom VideoCore IV, OpenGL ES 2.0, OpenVG 1080p60 H.264 high-profile encode/decode, 400 MHz
Memoria SDRAM	512 MB	1024 MB
USB	2 x USB 2.0	4 x USB 2.0
Salida de vídeo	Composite video, Composite RCA, HDMI	HDMI
Salida de audio	Conector de 3.5mm, HDMI	Conector de 3.5mm, HDMI
Almacenamiento	Ranura MicroSD	Ranura MicroSD
Redes	10/100 Ethernet (RJ-45), WiFi vía USB dongle	10/100 Ethernet (RJ-45), WiFi 802.11n integrada, Bluetooth 4.1
Periféricos de bajo nivel	26 x GPIO pins, Serial Peripheral Interface Bus (SPI), I <sup>2</sup> C, I <sup>2</sup> S, UART	40 x GPIO pins, Serial Peripheral Interface Bus (SPI), I <sup>2</sup> C, I <sup>2</sup> S, UART, I2C IDC Pins
Consumo energético	700 mA, (3.5 W)	800 m(4.0 W)
Fuente de alimentación	5 V (DC) vía Micro USB o pines GPIO	5 V (DC) vía Micro USB o pines GPIO
Tamaño	85.0 mm x 56.0 mm x 17 mm	85.6 mm x 56.5 mm x 17 mm
Peso	40g	45g

<sup>a</sup> **Precio:** Precio en el momento de su lanzamiento.

Tabla 6.1: *Detalles técnicos de las Raspberry Pi utilizadas en el prototipo.*

Servidor	Puerto externo	IP servidor interno	Puerto interno	Descripción
Autenticación	8662	192.168.1.144	442	Puerto de acceso al servidor Apache por HTTPS
	8888	192.168.1.144	80	Puerto de acceso al servidor Apache por HTTP
	8443	192.168.1.144	443	Puerto de acceso al servidor Apache por HTTPS
	8022	192.168.1.144	22	Acceso SSH
Votación	8889	192.168.1.145	80	Puerto de acceso al servidor Apache por HTTP
	8445	192.168.1.145	443	Puerto de acceso al servidor Apache por HTTPS
	8044	192.168.1.145	22	Acceso SSH

Tabla 6.2: *Port forwarding en la red de desarrollo.*



The internet? Is that thing still  
around?

---

Homer Simpson<sup>1</sup>

## Capítulo 7

# Líneas futuras

Este proyecto representa una prueba de concepto de cómo se puede implementar un sistema de voto por Internet que utiliza el DNIe 3.0 como herramienta para la identificación digital del votante y que busca facilitar que el acceso seguro al sistema de voto se pueda realizar desde dispositivos móviles tales como un smartphone o una tablet Android.

No obstante, como cualquier prueba de concepto, tras la elaboración de la memoria del proyecto y la implementación del mismo, se advierten varios ámbitos donde hay carencias tecnológicas o se observan oportunidades de mejora en el desarrollo.

### 7.1. Auditoría de seguridad / criptografía

El principal punto en el que se debería intervenir es en la seguridad criptográfica del proyecto. Ciento es que el desarrollo se basa en un sistema como Helios Voting, que está avalado tecnológicamente por instituciones como el MIT y por grandes expertos internacionales en criptografía (avanzados en 2.4.3). Pero en el momento en el que se ha debido desarrollar una solución para admitir el DNIe como herramienta de identificación digital, ya se ha impactado en la seguridad del sistema. En caso de querer llevar esta solución a la práctica para un proceso electoral real, es necesario que se realice un estudio de los elementos criptográficos utilizados en el proyecto, así como de la seguridad de cada uno de los elementos implementados. Especialmente crítico es esto teniendo en cuenta que se trata de un sistema accesible por Internet, cuyo core está escrito hace algunos años.

Por ello, en caso de querer avanzar en este sentido, recomiendo desarrollar estas cuestiones incluyendo una auditoría de seguridad rigurosa del sistema. Por un lado permitiría incrementar la seguridad y fiabilidad del mismo y, por otro, dota al sistema de capacidades legales que puedan cumplir criterios de los estamentos u organizaciones que puedan requerir de este servicio electoral.

---

<sup>1</sup>Personaje ficticio protagonista de la serie animada Los Simpson. Capítulo 226 (23 de la temporada 10) *Thirty Minutes over Tokyo* 16/05/1999 <https://youtu.be/Q9A0Vufw3NQ> <http://www.simpsonsarchive.com/episodes/AABF20.txt>

## 7.2. Blockchain

Últimamente se habla bastante del blockchain. Es una muy interesante opción para complementar la seguridad criptográfica del sistema de votación. No deja de ser una tecnología criptográfica, por lo que no resuelve los problemas operacionales o sociales del voto electrónico por Internet, pero es una buena opción para incrementar la seguridad criptográfica. Esto provoca, a su vez, un aumento de confianza en el proceso. Es muy recomendable la propuesta sobre la utilización de blockchain en sistemas de voto remotos que se plantea en el PFC de Marín Bermúdez [37].

## 7.3. Procurar escalabilidad del sistema

El código del core de Helios Voting está escrito en Python sobre un framework Django.

Una posible línea de desarrollo futuro para este proyecto podría ser la migración del mismo hacia sistemas modernos escalables. En estos momentos, la posibilidad de escalar el sistema implementado consiste en migrarlo a máquina con mayor potencia y recursos. Está muy limitado en caso de aumentar el número de votantes.

Pienso que sería una muy buena línea de desarrollo el migrar el sistema a una tecnología que permita escalabilidad de forma simple. Para ellos, pienso en tecnologías como Nodejs y microservicios. Nodejs permite la implementación de sistemas estables, con una comunidad inmensa sobre la que apoyarse, con una ingente cantidad de módulos que reutilizar. Esto proporciona una herramienta importante para favorecer el desarrollo ágil. Junto a esta tecnología, se puede introducir el uso de microservicios. Migrando el sistema al uso de microservicios podemos mejorar el rendimiento del sistema. Por un lado es más fácil probar la funcionalidad del sistema, ya que con pruebas unitarias de cada microservicio, aseguras la funcionalidad de cada uno. Además el uso de microservicios facilita la fácil escalabilidad del sistema, pues si la carga aumenta, en lugar de una migración a una máquina más potente, se pueden lanzar nuevas instancias de cada uno de los microservicios más impactados para que estos trabajen en paralelo.

Considero que este desarrollo es bastante interesante porque permite el crecimiento de las capacidades del proyecto con una inversión muy baja.

## 7.4. Pruebas

Un sistema de voto, y más si es accesible por Internet, necesita un sistema de tests que sea robusto, estable y minucioso. Hay que asegurar en la medida de lo posible que el sistema no falla.

Este proyecto carece de un sistema fiable de test, lo que identifica este punto como la mayor vulnerabilidad del sistema.

Es fundamental que se desarrolle una batería de pruebas para el sistema que permita asegurar su fiabilidad.

Para ello, una posibilidad que recomiendo, es la reescritura del sistema siguiendo una metodología de desarrollo basada en TDD.

Así, se desarrollaría el sistema obligando al mismo a pasar los tests pensados para cada módulo y para el sistema de forma global.

Esta metodología permite desarrollar de forma segura y fiable, permitiendo, igualmente, refactorizar

código con la misma seguridad, con lo que al final el la velocidad de desarrollo se acelera bastante.

Un sistema tan importante como una votación, que debe minimizar los errores al máximo por la criticidad de los mismos, debería apoyarse en un sistema automatizado de tests. Deberían implementarse tests unitarios y punto a punto para todos los subsistemas y funcionalidades del sistema. Con esto, se puede asegurar la integridad del mismo y facilitar los cambios en el software, provocando un software más estable.

Combinando estas líneas de desarrollo futuro pienso en una metodología de desarrollo TDD, con herramientas de Integración Continua, lenguajes como Python o Nodejs y el uso de microservicios. Creo que se puede implementar un sistema estable y con un rendimiento muy elevado, que facilite el desarrollo de funcionalidades de forma segura. Este sistema se basaría en pruebas automatizadas lo que, además de asegurar la estabilidad del sistema frente a cambios, permitiría la refactorización del código de forma segura, incrementando la adaptabilidad del sistema y su rendimiento.

## 7.5. Sistema de identificación del votante

Por limitaciones tecnológicas del componente WebView de Android se debió desarrollar una solución alternativa de identificación del votante contra el sistema basada en OAuth. Este desarrollo entiendo que es el punto más vulnerable del sistema, por lo que considero que es una buena línea de desarrollo para el futuro.

Sería interesante sustituir este mecanismo de identificación por uno realmente seguro.

Por un lado, se podría estudiar el componente WebView de Android para observar cómo es su implementación. Se puede comentar con los desarrolladores de Google el problema que tiene en cuanto al uso de certificados de cliente y navegación segura por una web (en contra de un sistema REST) para ver si hay opciones de implementación de un nuevo módulo que cumpla con las necesidades. También cabe la posibilidad de implementar un componente WebView modificado para cumplir con los requisitos de este sistema.

Otra opción, no obstante es cambiar el sistema para que pueda funcionar con el componente WebView de Android. Para ello sería necesario convertirlo en un sistema completamente REST. Con ello, se puede pensar en la implementación de una app Android que no sólo sirva para la autenticación, sino para la votación completa. esto implica reescribir el sistema pasando de web a Android. El backend del sistema sería el mismo, sólo que serviría HTML a las conexiones realizadas con navegadores clientes de escritorio, pero desde apps Android se basaría en peticiones HTTPS directas contra servicios REST incluyendo el certificado en cada una de ellas.

No obstante, la limitación del sistema ha sido provocada por los requisitos del mismo. En concreto, el uso del DNIe 3.0 con su chip NFC. Otra línea de desarrollo puede ser la búsqueda de nuevos servicios de identificación digitales y la implementación para este sistema. Para ello pienso que es interesante la investigación en identificación digital basada en biometría (huellas digitales, lectura ocular, facial, etc.), pasaporte digital, mobId (usado en Estonia2.2.1, [42]), bitcoins, multifactor, etc. Considero que es un mundo con unas posibilidades enormes que merece la pena ser investigado para encontrar soluciones que adaptar e incorporar al sistema desarrollado en este proyecto.



En democracia, cada cuatro años todos somos iguales, todos valemos lo mismo, y con un lápiz y un papel dibujamos el país que queremos.

---

*Ricardo Lagos<sup>1</sup>*

## Capítulo 8

# Conclusiones

El problema del voto por Internet es más complejo de lo que a priori puede parecer.

El hecho de tener que lidiar con el reto de la dualidad verificabilidad - secreto (2.1.2) complica el diseño de cualquier sistema de voto electrónico por Internet. Es muy complicado llegar al punto medio en el cual la privacidad del voto de un votante se mantiene lo justo para poder demostrar que el voto es correcto y de la misma forma ha sido incluido en el escrutinio.

Esta lucha entre verificabilidad y secreto provoca desconfianza. Tanto en los propios votantes, por la privacidad de su elección. Como en los afectados por el resultado, que pueden dudar de la transparencia del sistema, de la honestidad del mismo a la hora de contar los votos sufragados.

La mayoría de los retos tecnológicos están superados. La seguridad de las comunicaciones, la identificación digital de los votantes, las herramientas criptográficas. Incluso en este proyecto avanzamos una prueba de concepto para utilizar el nuevo DNIe 3.0 usando sensores NFC para poder votar desde un dispositivo móvil. A diario utilizamos sistemas que requieren una seguridad muy importante, como las transacciones bancarias, consultas médicas, VPNs corporativas, etc.

Sin embargo, en la mayoría de estos casos de uso, un fallo en estos sistemas es visible cuando ocurre, incluso demostrable. Si falla el sistema del banco y nos cobra irregularmente más dinero del que debe, lo podemos ver en el extracto de nuestra cuenta y reclamarlo, por ejemplo.

En un sistema de voto por Internet, esto no es posible. Al menos si queremos asegurar la privacidad del votante. Si el sistema no totaliza correctamente nuestro voto, no tenemos forma de saberlo o demostrarlo si no procedemos a desencriptar e identificar, de alguna forma, nuestro voto en la "urna digital".

La confianza es básica, por tanto, para estos sistemas.

Otro problema importante es la coacción. La capacidad de que un sistema proteja al votante de presiones externas a la hora de emitir su voto.

El desarrollo de este proyecto, basado en Helios Voting, es factible para llevar a cabo elecciones con bajo riesgo de coacción, como las de nivel universitario, organizacional o corporativas. En estos casos, aunque la importancia de la elección puede variar, normalmente el riesgo de coacción al votante es bajo. No es lo mismo cuando hablamos de elecciones de cargos públicos, donde sí que se pueden producir presiones

---

<sup>1</sup>Abogado, economista y expresidente de Chile (Día de las elecciones a la presidencia, 2005) Fuente: [http://www.elperiodicoextremadura.com/noticias/extremadura/pinochet-no-llego-a-su-colegio-electoral\\_210090.html](http://www.elperiodicoextremadura.com/noticias/extremadura/pinochet-no-llego-a-su-colegio-electoral_210090.html)

importantes al censo.

Todavía queda un largo recorrido para que el voto por Internet llegue a elecciones legislativas en España. Pese a que existe una lista de ejemplos de países o territorios en los que sí que se ha establecido esta tecnología, siendo Estonia el máximo exponente, personalmente creo que en España no es posible hacerlo de corto a medio plazo.

El primer escollo para implantar el voto por Internet en España se encuentra en la Constitución Española.

Tecnológicamente es posible diseñar e implementar sistemas de votación remota seguros. Existen multitud de protocolos criptográficos para asegurar el voto, conexiones seguras, formas de identificar digitalmente a los votantes. El problema no creo que sea la tecnología. Más bien, desde mi punto de vista, el problema principal es la confianza.

En la Constitución Española, dentro de la Ley Electoral, se dispone un escenario para las elecciones tradicionales basado en la desconfianza de los partidos políticos en el proceso. Por ello, hay multitud de testigos y conteo manual de los votos. Las mesas electorales constan de Presidente y dos vocales, con la posibilidad de estar acompañados de interventores de los diferentes partidos políticos, designados para ser testigos del proceso de voto y conteo, controlando que no haya manipulación del proceso.

El de España es un sistema bastante seguro. Tiene riesgos de integridad y coacción bastante bajos. Junto a ello, los sistemas utilizados actualmente para la difusión de resultados impacta claramente en la transparencia.

Los votos son contados manualmente por la mesa electoral de forma pública. El resultado, que consta en acta, es transmitido al Centro de Recogida de Datos, donde se digitaliza para cada mesa y se realiza la totalización de estos. Esta totalización es la que se difunde por numerosos medios en tiempo casi real. El resultado es que, tal como llega un acta, se difunde el resultado. Esta idea de tiempo real tiene como objetivo minimizar la desconfianza del electorado de que los resultados puedan ser manipulados. No obstante, los resultados ofrecidos durante la noche electoral no dejan de ser unos resultados provisionales. En los días posteriores se realiza un proceso público en el que los jueces de las diferentes Juntas Electorales revisan las actas de escrutinio de cada mesa para resolver impugnaciones o diferencias entre los resultados reales y los digitalizados de forma provisional. Es tras este proceso cuando se declara el resultado oficial, varios días después de la celebración de la elección.

Se observa que en un Proceso Electoral interviene mucha gente. Para poder manipular unas elecciones en España hay que poner de acuerdo a muchas personas para que se comporten de forma deshonesta. Es un sistema con una integridad alta.

No obstante, podemos presentar algunos motivos argumentados desde las plataformas a favor de la implantación del voto remoto.

**Recursos** El voto remoto por Internet es una gran opción en cuanto a ahorro de recursos. Se podría disminuir el papel invertido en papeletas, apertura de colegios electorales, personal, despliegue policial. Transporte de urnas, miembros de mesa.

**Accesibilidad** El voto remoto facilitaría el voto a muchas personas con movilidad reducida o algún problema explícito que le impida acudir a su Colegio Electoral.

**Movilidad** Al poder votar desde casa, el voto remoto reduciría la movilidad de votantes hasta sus colegios electorales.

**Rapidez** Otro argumento a favor del voto remoto es la velocidad de escrutinio y totalización.

**Participación** En círculos favorables a la implantación del voto ubicuo se incluye la argumentación de que éste favorecería la disminución de la abstención, ya que motivaría a muchos votantes a ejercer

su derecho sin necesidad de desplazarse al Colegio Electoral. Además, se espera que incentive el voto joven, ya que se considera que los jóvenes son el estrato social más indicado para adoptar el uso de la tecnología en este tipo de procesos.

Además de los riesgos de seguridad y todos los expuestos en el capítulo 4, considero que algunos de estos argumentos no son suficientes para sustituir el voto tradicional por el remoto.

El problema, una vez más, es la confianza en el Sistema. En los últimos comicios legislativos de 2016, un gran grupo de votantes puso en tela de juicio el proceso de conteo de votos, al considerar que había una diferencia muy grande en los votos que obtuvo un determinado partido en comparación con los resultados de los sondeos a pie de urna de esa misma tarde. Recuerdo que el conteo de votos es manual y **público** en las mesas y que se envía a un sistema central para que realice la totalización **provisional** mientras los recibe. En los días posteriores se realiza el Escrutinio Definitivo. Es decir, se puso en entredicho la honestidad de un sistema informático que realiza el escrutinio provisional. Todo esto dentro de un sistema electoral diseñado para combatir la coacción y asegurar la integridad con la intervención de muchos actores y todo de forma pública. Si esto ocurrió con este Sistema, es fácil discernir los problemas de confianza que puede ofrecer un sistema que parecerá de caja negra, sin intervención de terceros de forma manual, sin recibos de voto, sin posibilidad física de recuperación en caso de catástrofe.

Si pensamos en este sistema en cuanto a velocidad de escrutinio y difusión de resultados, en las últimas elecciones, tres horas después del cierre de los colegios ya se sabe, provisionalmente, el resultado escrutado de alrededor de un 80 % de las mesas. No considero, pues, que este sea un argumento determinante a tener en cuenta. Sobre todo teniendo en cuenta que otros estados, democráticamente longevos, incluso cierran los colegios tras cerrarlos la noche electoral y es al día siguiente cuando comienzan el escrutinio.

Para que el sistema no sea una caja negra al votante, hay que proveer de herramientas para que este pueda asegurar la fiabilidad del proceso, tanto en el escrutinio, como que su voto ha sido contabilizado y no manipulado. Esto se puede conseguir con técnicas criptográficas. Utilizando un escrutinio homomórfico (el voto nunca es descifrado, se totalizan los votos cifrados y es este resultado el que se descifra). Con criptografía se puede asegurar que un voto no ha sido manipulado, con lo que un votante puede usar estas técnicas y realizar por sí mismo la comprobación. Pero esto, en la práctica no es lo mismo que ver cómo tu voto se introduce en una urna transparente. Que el voto se metió en un sobre cerrado, secreto. Y se puede asistir al conteo de todos los votos. En términos de votantes no acostumbrados a sistemas criptográficos la comparación deriva a que el sistema informático es una caja peligrosa en la que sumar un voto a un partido o candidato contrario por parte de los desarrolladores del sistema es claramente factible.

Con software libre se minimiza este problema, y muchas voces abogan porque así debería ser. Los algoritmos accesibles a la auditoría ciudadana. Pero a alguien no interesado en la parte técnica del proceso puede resultar complicado demostrarle que el código libre compartido es el mismo que finalmente se ejecute durante el día electoral.

Un sistema remoto tiene demasiados puntos que pueden provocar desconfianza en el proceso. Algo que hay que valorar bastante antes de sustituir el voto tradicional, diseñado para evitar esta sensación de desconfianza.

Me gustaría que fuésemos capaces de superar estas limitaciones y poder implementar el voto por Internet a escala legislativa, pero hoy en día todavía hay muchas barreras, sobre todo sociales, que hay atravesar. No obstante, algún día tendremos la oportunidad, el derecho y el deber de votar desde casa a través de nuestros dispositivos conectados a Internet.



# Glosario de términos

**API** Application Programming Interface. 59, 102, 155

**CAN** Card Access Number. 161, 163–165

**CAS** Central Authentication Service. 122

**CERES-FNMT** CERtificación ESpañola - Fábrica Nacional de Moneda y Timbre. 50

**CRL** Certificate Revocation List (Lista de Revocación de Certificados). 151

**CSV** Comma-Separated Values. 104

**DDoS** Denegación de Servicio Distribuido - Distributed Denial of Service. 59

**DGP** Dirección General de la Policía. 150, 160, 184

**DNI** Documento Nacional de Identidad. 94, 99

**DNIe** Documento Nacional de Identidad Electrónico. 25, 26, 50, 79–81, 84, 93, 97–99, 103, 104, 116, 120–123, 131, 132, 135, 143–145, 149, 150, 156, 158, 160, 161, 163–166, 168–170, 173, 174, 184, 185, 191, 193, 195, 207, 210

**DRE** Direct-Recording Electronic. 28

**E2E** End-to-End (Punto a punto). 49, 51, 54, 55, 102, 119, 120, 141

**EML** Election Markup Language. 88, 89

**FNMT** Fábrica Nacional de Moneda y Timbre. 81, 150

**GPRS** General Packet Radio Service. 110

**HSDPA** High Speed Downlink Packet Access. 110

**HTML** HyperText Markup Language. 136, 185

**HTTP** Hypertext Transfer Protocol. 136, 155, 189

**HTTPS** Hypertext Transfer Protocol Secure. 122, 131, 135–137, 144, 145, 161, 162, 166, 189, 207

**INE** Instituto Nacional de Estadística. 92, 95

**MIT** Massachussets Institute of Technology. 61, 191

**MRZ** Machine-Readable Zone. 99

**MVC** Modelo-Vista-Controlador. 156

**NFC** Near Field Communication. 80, 81, 83, 84, 99, 123, 160, 165, 173, 193, 195

**OASIS** Organization for the Advancement of Structured Information Standards. 88

**OCSP** Online Certificate Status Protocol. 81, 150, 151, 169

**PACE** Password Authentication Connection Establishment. 80, 163, 164

**PC** Personal Computer. 160

**PFC** Proyecto Final de Carrera. 26, 29, 30, 51, 81, 87, 121, 158, 192

**PIN** Personal Identification Number. 80, 82, 132, 163, 166

**REST** REpresentational State Transfer. 135, 155, 193

**RFID** Radio Frequency Identification. 79, 81, 83

**SSL** Secure Sockets Layer. 54, 131, 173

**TDD** Test Driven Development - Desarrollo guiado por pruebas. 192

**TLS** Transport Layer Security. 59, 60, 173

**TPV** Terminal Punto de Venta. 84

**TUI** Tarjeta Universitaria Inteligente. 93

**UAH** Universidad de Alcalá de Henares. 42

**UNED** Universidad Nacional de Educación a Distancia. 42, 49, 50

**UPV/EHU** Universidad del País Vasco / Euskal Herriko Unibertsitatea. 49

**UPyD** Unión, Progreso y Democracia. 42

**URL** Uniform Resource Locator. 129, 130, 135, 142, 169, 174, 181

**VPN** Virtual Private Network - Red Privada Virtual. 195

**W3C** World Wide Web Consortium. 99, 123, 138, 183, 185

**WCAG** Web Content Accessibility Guidelines - Pautas de Accesibilidad para el Contenido Web. 185

# Bibliografía

- [1] ADIDA, B. Advances in Cryptographic Voting Systems. Master's thesis, MIT - Massachusetts Institute of Technology - Department of Electrical Engineering and Computer Science, Agosto 2006. <http://assets.adida.net/research/phd-thesis.pdf>.
- [2] ADIDA, B. Helios: Web-based open-audit voting. In *Proceedings of the Seventeenth Usenix Security Symposium (USENIX Security 2008)* (July 2008), pp. 335–348.
- [3] BENALOH, J. Verifiable Secret-Ballot Elections. Master's thesis, Yale University, Septiembre 1987. <http://www.cs.yale.edu/publications/techreports/tr561.pdf>.
- [4] BENALOH, J. Simple verifiable elections. In *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop 2006 on Electronic Voting Technology Workshop* (Berkeley, CA, USA, 2006), EVT'06, USENIX Association, pp. 5–5.
- [5] BOKSLAG, W., AND DE VRIES, M. Evaluating e-voting: theory and practice. *CoRR abs/1602.02509* (2016).
- [6] BOOKS, W. D. *Pocket Patriot*. Writer's Digest Books, 4 2005.
- [7] BURNAND, F. E-voting to advance slowly in 2011. <http://www.swissinfo.ch/eng/e-voting-to-advance-slowly-in-2011/29138944>, 2011.
- [8] CABARCAS JARAMILLO, D. El voto electrónico y los retos criptográficos relacionados. *Revista Facultad de Ciencias Universidad Nacional de Colombia, Sede Medellín* (Diciembre 2015).
- [9] CABELLO PARDOS, A.B., HERNÁNDEZ ENCINAS, A., HOYA WHITE, S., MARTÍN DEL REY, A., AND RODRÍGUEZ SÁNCHEZ, G. Un protocolo de votación electrónica basado en firmas digitales ciegas. *Universidad de Sevilla. XX Congreso de Ecuaciones Diferenciales y Aplicaciones. X Congreso de Matemática Aplicada* (Septiembre 2007).
- [10] CARRACEDO VERDE, JOSÉ DAVID, GÓMEZ OLIVA, ANA, MORENO BLÁZQUEZ, JESÚS, PÉREZ BELLEBONI, EMILIA, AND CARRACEDO GALLARDO, JUSTO. Votación electrónica basada en criptografía avanzada (Proyecto VOTESCRIPT). *Universidad Politécnica de Madrid* (2002). [http://vototelematico.diatel.upm.es/articulos/articulo\\_venezuela\\_revisado.pdf](http://vototelematico.diatel.upm.es/articulos/articulo_venezuela_revisado.pdf).
- [11] CARTER CENTER. Internet Voting Pilot: Norway's 2013 Parliamentary Elections, Mar. 2014.
- [12] Certificados Digitales - FNMT. <https://www.cert.fnmt.es/curso-de-criptografia/certificados-digitales>.
- [13] CHAUM, D. L. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Commun. ACM* 24, 2 (Feb. 1981), 84–90.
- [14] CHEN, X., WU, Q., ZHANG, F., TIAN, H., WEI, B., LEE, B., LEE, H., AND KIM, K. New receipt-free voting scheme using double-trapdoor commitment. *Information Sciences* 181, 8 (2011), 1493 – 1502.
- [15] CLARKSON, M. R., CHONG, S., AND MYERS, A. C. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy* (2008), IEEE Computer Society, pp. 354–368.

- [16] CODINA LLIGOÑA, J. Bases teóricas y herramientas para el análisis y comparación de sistemas de votación de código abierto. Master's thesis, Universitat Oberta de Catalunya - Institut Municipal d'Informàtica de Barcelona, Enero 2014. <http://openaccess.uoc.edu/webapps/o2/handle/10609/28282>.
- [17] CORTÉS POLO, DAVID MIGUEL, HORNERO ÍNCERA, ALEXEI, MARTÍNEZ BRAVO, LORENZO, AND GONZÁLEZ-SÁNCHEZ, JOSÉ LUIS. Estudio de infraestructura para sistemas de voto electrónico. *Departamento de Informática, Escuela Politécnica. Universidad de Extremadura (-)*. <http://gitaca.unex.es/agila/voto/voto.pdf>.
- [18] CUERPO NACIONAL DE POLICÍA DE ESPAÑA, FABRICA NACIONAL DE MONEDA Y TIMBRE, MINISTERIO DEL INTERIOR. *Descripción APP "DNIe Autenticación"1.0*, 2015.
- [19] DE POLICÍA DE ESPAÑA, C. N. Guía de Referencia del DNIe con NFC. [https://www.dnielectronico.es/PDFs/Guia\\_de\\_Refencia\\_DNIe\\_con\\_NFC.pdf](https://www.dnielectronico.es/PDFs/Guia_de_Refencia_DNIe_con_NFC.pdf), Febrero 2015.
- [20] DE RESPUESTA A INCIDENTES DE SEGURIDAD (INTECO-CERT), C. Guía para desarrolladores con el DNI electrónico. <https://www.incibe.es/extfrontinteco/img/File/intecocert/dnie/pdf/guiades.pdf>, Octubre 2007.
- [21] DELLA PAOLERA, P. La prueba de Conocimiento Cero o Nulo. <http://paolera.wordpress.com/2014/06/27/la-prueba-de-conocimiento-cero-o-nulo/>, Junio 2014.
- [22] DHILLON, KYLE. Challenges for LargeScale Internet Voting Implementations. Tech. rep., Princeton University Department of Computer Science, Enero 2015. [https://www.cs.princeton.edu/sites/default/files/uploads/kyle\\_dhillon.pdf](https://www.cs.princeton.edu/sites/default/files/uploads/kyle_dhillon.pdf).
- [23] DIRECCIÓN GENERAL DE LA POLICÍA - MINISTERIO DEL INTERIOR - ESPAÑA. Infraestructura de Clave Pública - DNI Electrónico - Proyecto de Declaración de Prácticas y Políticas de Certificación. [http://www.dnielectronico.es/PDFs/politicas\\_de\\_certificacion.pdf](http://www.dnielectronico.es/PDFs/politicas_de_certificacion.pdf), Marzo 2006.
- [24] DIRECCIÓN GENERAL DE LA POLICÍA - MINISTERIO DEL INTERIOR - ESPAÑA. Infraestructura de Clave Pública de la Dirección General de la Policía - Declaración de Prácticas y Políticas de Certificación. <http://www.policia.es/DPC/dpc.pdf>, Enero 2014.
- [25] EL GAMAL, T. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in Cryptology* (New York, NY, USA, 1985), Springer-Verlag New York, Inc., pp. 469–472.
- [26] Internet Voting - Voting methods in Estonia - Estonian National Electoral Committee. <http://www.vvk.ee/voting-methods-in-estonia/engindex/>.
- [27] FUJIOKA, ATSUSHI, OKAMOTO, TATSUAKI, AND OHTA, KAZUO. A Practical Secret Voting Scheme for Large Scale Elections. In *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques: Advances in Cryptology* (London, UK, UK, 1993), ASIACRYPT '92, Springer-Verlag, pp. 244–251.
- [28] GARCIA MONDARAY, S. Especificación de requisitos software con IEEE 830-1998. <http://www.godtic.com/blog/2012/11/18/especificacion-de-requisitos-software-con-ieee-830-1998/>, Noviembre 2012.
- [29] GARCÍA ZAMORA, C. P. Diseño y Desarrollo de un Sistema para Elecciones Electrónicas Seguras (SELES). Master's thesis, Centro de Investigacion y de Estudios Avanzados del Instituto Politecnico Nacional. Departamento de Ingeniería Eléctrica. Sección de Computación, Septiembre 2005. <http://delta.cs.cinvestav.mx/~francisco/Repository/tesisCPGZ.pdf>.
- [30] HERSCHEBERG, M. A. Secure Electronic Voting Over the World Wide Web. Master's thesis, Department of Electrical Engineering and Computer Science, MIT - Massachusetts Institute of Technology, Mayo 1997.
- [31] IEEE. IEEE Recommended Practice for Software Requirements Specifications. *IEEE Std 830-1998* (1998).

- [32] KIAYIAS, A., KORMAN, M., AND WALLUCK, D. An internet voting system supporting user privacy. In *ACSAC* (2006), IEEE Computer Society, pp. 165–174.
- [33] LÓPEZ GARCIA, M. D. L. Sistema Electrónico de Votación. Master's thesis, Benemérita Universidad Autónoma de Puebla. Facultad de Ciencias de la Computación, Febrero 2007. [http://delta.cs.cinvestav.mx/~francisco/TesisMaestriaFinal\\_Lourdes.pdf](http://delta.cs.cinvestav.mx/~francisco/TesisMaestriaFinal_Lourdes.pdf).
- [34] LÓPEZ GARCIA, M. d. L. *Diseño de un protocolo para votaciones electrónicas basado en firmas a ciegas definidas sobre emparejamientos bilineales*. PhD thesis, Centro de Investigacion y de Estudios Avanzados del Instituto Politecnico Nacional. Departamento de Computación, Junio 2011. <http://www.cs.cinvestav.mx/TesisGraduados/2011/TesisLourdesLopez.pdf>.
- [35] MAENE, P. Online verkiezingen in de praktijk: verbetering en toepassing van het Helios verkiezingssysteem. Master's thesis, KU Leuven - Faculteit Ingenieurswetenschappen, 2014. <https://www.esat.kuleuven.be/cosic/publications/thesis-249.pdf>.
- [36] MARTINS TOURAIAS PEREIRA, G. D. Scroll, Match & Vote: An E2E Coercion Resistant Mobile Voting System. Master's thesis, Instituto Superior Técnico Lisboa, Junio 2014. <https://fenix.tecnico.ulisboa.pt/downloadFile/395146459054/thesis.pdf>.
- [37] MARÍN BERMÚDEZ, A. Estudio de la utilización de protocolos blockchain en sistemas de votación electrónica, 2016. [https://upcommons.upc.edu/bitstream/handle/2117/98545/PFC%20Blockchain\\_Evoting\\_v01.pdf](https://upcommons.upc.edu/bitstream/handle/2117/98545/PFC%20Blockchain_Evoting_v01.pdf).
- [38] MEISSEN, R. A Mathematical Approach to Fully Homomorphic Encryption. Master's thesis, Worcester Polytechnic Institute. [https://web.wpi.edu/Pubs/E-project/Available/E-project-042612-132350/unrestricted/Meissen\\_MQP2.pdf](https://web.wpi.edu/Pubs/E-project/Available/E-project-042612-132350/unrestricted/Meissen_MQP2.pdf).
- [39] MINISTERS, C. *Legal, Operational and Technical Standards for E-voting: Recommendation Rec(2004)11 Adopted by the Committee of Ministers of the Council of Europe on 30 September 2004 and Explanatory Memorandum*. Legal Issues. Council of Europe Pub., 2005.
- [40] MORALES ROCHA, V. M. *Seguridad en los procesos de voto electrónico remoto: registro, votación, consolidación de resultados y auditoría*. PhD thesis, Universitat Politècnica de Catalunya. Departament d'Enginyeria Telemàtica, Marzo 2009. <http://www.tdx.cat/bitstream/handle/10803/7043/01VMmr01de01.pdf>.
- [41] MORENO PEÑA, ADRIÁN. Portal web para la gestión de información de un departamento universitario en la USC. Master's thesis, Escuela Técnica Superior de Ingeniería - Universidad de Santiago de Compostela, Diciembre 2007. <http://bloqnum.com/pfc/proyecto/proyecto.html>.
- [42] MORSHED CHOWDHURY, M J. Comparison of e-voting schemes: Estonian and Norwegian solutions. *NordSecMob, University of Tartu* (2010). <http://courses.cs.ut.ee/2010/security-seminar-fall/uploads/Main/chowdhury-final.pdf>.
- [43] MUÑOZ-MANSILLA, ROCÍO, MARCOS DEL CANO, ANA MARÍA, GÓMEZ GARZÁS, JESÚS, AND MARTÍN DEL LLANO, ISABEL. Elecciones a claustro 2010 en la uned. [http://portal.uned.es/pls/portal/docs/PAGE/UNED\\_MAIN/LAUNIVERSIDAD/VICERRECTORADOS/SECRETARIA/ELECCIONES%20RECTOR%202013/DOCUMENTACI%C3%93N%20ELECTORAL/VOTO%20ELECTRÓNICO%20CLAUSTRO%20UNED-COSEG-TIC-CARTAGENA%2011ABR13V3.PDF](http://portal.uned.es/pls/portal/docs/PAGE/UNED_MAIN/LAUNIVERSIDAD/VICERRECTORADOS/SECRETARIA/ELECCIONES%20RECTOR%202013/DOCUMENTACI%C3%93N%20ELECTORAL/VOTO%20ELECTRÓNICO%20CLAUSTRO%20UNED-COSEG-TIC-CARTAGENA%2011ABR13V3.PDF), Junio 2013.
- [44] Normas de organización y funcionamiento de la Universidad San Pablo-CEU. [http://www.uspceu.com/\\_docs/conocenos/normativa-universitaria/1.pdf](http://www.uspceu.com/_docs/conocenos/normativa-universitaria/1.pdf).
- [45] OCHOA JIMÉNEZ, J. E. Función picadillo determinista al grupo G2 y su aplicación en autenticación para dispositivos móviles. Master's thesis, Centro de Investigacion y de Estudios Avanzados del Instituto Politecnico Nacional. Departamento de Computación, México D.F., México, Diciembre 2013. <http://www.cs.cinvestav.mx/TesisGraduados/2013/TesisJoseOchoa.pdf>.
- [46] PANIZO ALONSO, L. Desarrollo de una metodología para el análisis y la clasificación de los sistemas de voto electrónico. Master's thesis, Universidad de León - Departamento de Ingeniería Eléctrica y

- de Sistemas y Automática, Diciembre 2014. [https://buleria.unileon.es/bitstream/handle/10612/4237/tesis\\_b4cf6.PDF](https://buleria.unileon.es/bitstream/handle/10612/4237/tesis_b4cf6.PDF).
- [47] PEDRO GERALDO M. R. ALVES. Aplicação Conceitual de Criptografia Homomórfica. *Instituto de Computação - Universidade Estadual de Campinas* (Diciembre 2014). [http://www.ic.unicamp.br/~ra085994/reports\\_and\\_papers/ouetros/Applicacao\\_Conceitual\\_de\\_Criptografia\\_Homomorfica-MO422-20141223.pdf](http://www.ic.unicamp.br/~ra085994/reports_and_papers/ouetros/Applicacao_Conceitual_de_Criptografia_Homomorfica-MO422-20141223.pdf).
- [48] PUIGGALÍ, JORDI, CHÓLIZ, JESÚS, AND GUASCH, SANDRA. Best Practices in Internet Voting. *Scytl Secure Electronic Voting* (2010). [http://www.scytl.com/wp-content/uploads/2013/05/PUIGGALI\\_BestPracticesInternetVoting.pdf](http://www.scytl.com/wp-content/uploads/2013/05/PUIGGALI_BestPracticesInternetVoting.pdf).
- [49] PÉREZ BELLEBONI, E. Aplicación de documentos de identificación electrónica a un esquema de voto telemático a escala paneuropea, seguro, auditible y verificable. Master's thesis, Universidad Politécnica de Madrid - Escuela Universitaria de Ingeniería Técnica de Telecomunicación - Departamento de Ingeniería y Arquitecturas Telemáticas, Febrero 2013. [http://oa.upm.es/14925/1/EMILIA\\_PEREZ\\_BELLEBONI.pdf](http://oa.upm.es/14925/1/EMILIA_PEREZ_BELLEBONI.pdf).
- [50] PÉREZ BELLEBONI, EMILIA, AND CARRACEDO GALLARDO, JUSTO. Uso del DNIE para reforzar el anonimato en el voto telemático mediante tarjetas inteligentes. *Departamento de Ingeniería y Arquitecturas Telemáticas. Escuela Universitaria de Ingeniería Técnica de Telecomunicación. Universidad Politécnica de Madrid* (2009). [http://vototelematico.diatel.upm.es/articulos/Uso\\_DNIE\\_anonimato\\_voto.pdf](http://vototelematico.diatel.upm.es/articulos/Uso_DNIE_anonimato_voto.pdf).
- [51] SCYTL. Elección a Rector/a UNED 2013. Voto por Internet. [http://portal.uned.es/pls/portal/docs/PAGE/UNED\\_MAIN/LAUNIVERSIDAD/VICERRECTORADOS/SECRETARIA/ELECCIONES%20RECTOR%202013/DOCUMENTACION%20ELECTORAL/6.PRESENTACION%20SCYTL%20VOTO%20ELECTR%C3%93NICO%20V1.PDF](http://portal.uned.es/pls/portal/docs/PAGE/UNED_MAIN/LAUNIVERSIDAD/VICERRECTORADOS/SECRETARIA/ELECCIONES%20RECTOR%202013/DOCUMENTACION%20ELECTORAL/6.PRESENTACION%20SCYTL%20VOTO%20ELECTR%C3%93NICO%20V1.PDF), Mayo 2013.
- [52] SMYTH, B., FRINK, S., AND CLARKSON, M. R. Election Verifiability: Cryptographic Definitions and an Analysis of Helios and JCJ. *Cryptology ePrint Archive, Report 2015/233*, 2015. <https://eprint.iacr.org/2015/233.pdf>.
- [53] U.S. ELECTION ASSISTANT COMMISSION. A Survey of Internet Voting. <http://www.eac.gov/assets/1/Documents/SIV-FINAL.pdf>, Septiembre 2011.
- [54] VENTURA BONELL-TEROL, M. A. Propuesta de implantación de votación electrónica en las elecciones a rector de la Universidad Politécnica de Valencia. Master's thesis, Universitat Politècnica de València. Facultat d'Administració i Direcció d'Empreses, Octubre 2011. <http://riunet.upv.es/bitstream/handle/10251/14584/PROPIUESTA%20DE%20IMPLANTACI%C3%93N%20DE%20VOTACI%C3%93N%20ELECTR%C3%93NICA%20EN%20LAS%20ELECCIONES%20A%20RECTOR%20DE%20LA%20UNIVERSIDAD%20PO.pdf?sequence=1>.
- [55] Voting Machines Pros and Cons. <http://votingmachines.procon.org/view.timeline.php?timelineID=000021>.
- [56] ØBERG, M. W. Improving the Norwegian Internet Voting Protocol. Master's thesis, Department of Mathematical Sciences, NTNU - Norwegian University of Science and Technology, Junio 2011. <https://daim.idi.ntnu.no/masteroppgaver/005/5823/masteroppgave.pdf>.

# Anexos

## A. Instalación de Helios

Estos son los pasos llevados a cabo para instalar el proyecto Helios en un servidor Linux. Son los que yo ejecuté para montar el prototipo del Sistema, por lo que puede ser que para alguien que intente realizar esta instalación, haya pasos que no le funcionen y/o no le permitan finalizar la instalación.

```
1 sudo apt-get update
2
3 # ## Si hace falta conectarse a Internet a través de un proxy:
4 export http_proxy=http://usuario:password@url:puerto
5 export https_proxy=https://usuario:password@url:puerto
6
7 sudo apt-get update
8 sudo -E apt-get update
9 sudo apt-get install postgresql
10 sudo -E apt-get install postgresql
11 cd Descargas/
12 git clone https://github.com/benadida/helios-server.git
13 cd helios-server/
14 sudo -E apt-get install python-virtualenv
15 virtualenv venv
16 source venv/bin/activate
17 sudo -E apt-get install libpq-dev
18 sudo -E apt-get install python-dev
19 pip install -r requirements.txt
20
21 # ## Hay que modificar el fichero --> reset.sh <--. Hay que sustituir la línea
22
23 # ## echo "from helios_auth.models import User; User.objects.create(user_type='google
24     ↪ ',user_id='ben@adida.net', info={'name':'Ben Adida'})" | python manage.py shell
25
26 # ## por la línea
27
28 # ## echo "from helios_auth.models import User; User.objects.create(user_type='
29     ↪ password',user_id='tuemail@servidor.com', info={'name':'Tu nombre'})" | python
30     ↪ manage.py shell
31
32 ./reset.sh
33 sudo su -c "createuser --superuser carlos" postgres
34 sudo service postgresql restart
35 ./reset.sh
```

```
35 pip install amqp
36 ./reset.sh
37 pip install billiard
38 ./reset.sh
39 pip install pytz
40 ./reset.sh
41 psql --dbname helios
42     >>> alter role carlos with password 'unapassword';
43 ##### El password que hay que poner es el que tenemos en settings.py
44 ./reset.sh
```

## B. Configuración del Servidor Web del Sistema de Votación

El framework web Django provee su propio Servidor Web, pero para poder utilizar un canal seguro HTTPS y configurar el acceso con el DNIe es preciso utilizar un Servidor Web como Apache en el que apoyar el framework. Aquí una lista de pasos que se han llevado a cabo para su configuración.

```

1  # Instalamos Apache2 y OpenSSL
2  sudo apt-get install apache2
3  sudo apt-get install openSSL
4
5  cd /etc/apache2/mods-available
6
7  # Activamos SSL en Apache
8  sudo a2enmod ssl
9
10 sudo apt-get install libapache2-mod-wsgi
11 sudo a2enmod wsgi
12
13 cd /etc/apache2/sites-available
14 sudo cp 000-default.conf sitednie.conf
15 cd /etc/apache2/sites-available
16
17 # Preparamos los certificados para el DNIe
18 mkdir ~/Descargas/certificados
19 cd certificados/
20
21 # Creamos una clave
22 openssl genrsa -des3 -out server.key 2048
23
24 # Petición del certificado, asociándolo con la clave que acabamos de crear
25 openssl req -new -key server.key -out server.csr
26
27 # Pese a que no somos una entidad certificadora, firmamos nuestro propio certificado,
28   ↪ así
29 # obtenemos un certificado autofirmado.
30 openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
31
32 #Bajamos el certificado de la AC Raíz del DNIe desde:
33 #http://www.dnilelectronico.es/PortalDNIe/PRF1_Cons02.action?pag=REF_077
34 wget http://www.dnilelectronico.es/ZIP/ACRAIZ-SHA2.CAB
35
36 # Lo extraemos
37 cabextract ACRAIZ-SHA2.CAB
38
39 # Creamos el certificado x509
40 openssl x509 -in ACRAIZ-SHA2.crt -inform DER -out ACRAIZ-SHA2.crt -outform PEM
41 sudo cp ACRAIZ-SHA2.crt acraiz-dnie.cer
42
43 cd /etc/apache2/sites-available
44 sudo cat default-ssl.conf >> sitednie.conf
45
46
47 # Configuramos el servidor web para que escuche por un puerto seguro, con SSL y
48 # requiriendo los certificados del DNIe
49 sudo gedit sitednie.conf &

```

Puede ser con el comando anterior para usar el editor gedit o con cualquier otro editor de nuestra

preferencia, pero se deben modificar los siguientes parámetros del fichero:

En <VirtualHost \*:80>:

```

1 DocumentRoot /home/carlos/Descargas/helios-server
2 <Directory /home/carlos/Descargas/helios-server>
3   <Files wsgi.py>
4     Require all granted
5   </Files>
6 </Directory>
7
8 WSGIDaemonProcess / python-path=/home/carlos/Descargas/helios-server:/home/carlos/
  ↳ Descargas/helios-server/venv/lib/python2.7/site-packages
9   WSGIProcessGroup /
10  WSGIScriptAlias / /home/carlos/Descargas/helios-server/wsgi.py

```

En <VirtualHost \_default\_:443>:

```

1 DocumentRoot /home/carlos/Descargas/helios-server
2 ##### Antes de cerrar </VirtualHost>:
3
4 <Directory /home/carlos/Descargas/helios-server/>
5   Options Indexes FollowSymLinks MultiViews
6   AllowOverride None
7   Order allow,deny
8   allow from all
9 </Directory>
10
11 #### Modificar:
12 SSLCertificateFile /home/carlos/Descargas/certificados/server.crt
13 SSLCertificateKeyFile /home/carlos/Descargas/certificados/server.key
14 SSLCACertificateFile /home/carlos/Descargas/certificados/acraiz-dnie.cer
15 SSLVerifyClient require
16 SSLVerifyDepth 2
17
18 #### Modificar donde se deba:
19 SSLOptions +StdEnvVars +ExportCertData
20
21
22 <Directory /home/carlos/Descargas/helios-server>
23   <Files wsgi.py>
24     Require all granted
25   </Files>
26 </Directory>
27
28 #WSGIDaemonProcess / python-path=/home/carlos/Descargas/helios-server:/home/carlos/
  ↳ Descargas/helios-server/venv/lib/python2.7/site-packages
29   WSGIProcessGroup /
30   WSGIScriptAlias / /home/carlos/Descargas/helios-server/wsgi.py

```

A continuación hay que editar el fichero ports.conf si es necesario:

```

1 sudo gedit /etc/apache2/ports.conf

1 ##### Modificar si es necesario:
2 <IfModule ssl_module>
3   Listen 443
4   Listen 1443
5 </IfModule>

```

Cargamos la configuración modificada en Apache y reseteamos el servicio:

```
1 sudo a2ensite sitednie.conf  
2 sudo service apache2 restart
```

## C. Exportar los certificados a la App Android

Pasos para exportar los certificados necesarios para conectar con servidores que requieren el DNIe desde una app Android.

```
1 keytool -keystore keystore.jks -import -alias root -file acraiz-dnie.cer -trustcacerts  
2  
3 keytool -keystore keystore.jks -import -alias servidor -file server.crt -trustcacerts  
4  
5 keytool -importkeystore -srckeystore keystore.jks -destkeystore keystore.bks  
6     -srcstoretype JKS -deststoretype BKS  
7     -srcstorepass <unapassword> -deststorepass <otrapassword>  
8     -provider org.bouncycastle.jce.provider.BouncyCastleProvider  
9     -providerpath "../bcprov-jdk16-1.45.jar"
```

Igualmente, es necesario obtener el hash del certificado del servidor para que la App Android confíe en él y sea capaz de establecer una conexión segura. El comando para la obtención de este hash es:

```
1 openssl x509 -fingerprint -noout -in server.crt
```

## D. Ejemplo de un voto en JSON

```

1  {
2      "answers": [
3          {
4              "choices": [
5                  {
6                      "alpha":
7                          "1182173802620524077296825512704480251704747008363675872351830783502565958108
8                          "79563933254777683913433460725096986398483643506466191091791794603744422363802234
9                          "36839844981593365057325056970359476019829362893435549114172515859548010845050316
10                         "56189202406400724079340455973917657866270274547132446594916845940087946766439914
11                         "28824720762303537161040526789809174023512198484656023155182213392679838148328690
12                         "44328021711207489979218996497185750403653557190590589203525249802182445788274397
13                         "10208209751620512844960268219569383685241045031651899944191603110538800514467824
14                         "5150184413574174275885253975594156703272641284359822947872735",
15                     "beta":
16                         "95623981261863827876891555243822629824302150121170638569858592305381156289005
17                         "60113793281913987706298959102398704054860503504635722097599744968107405574943928
18                         "57629292660798831188910461746566654460067161762296112236838028136355088189082014
19                         "15978094097330365271434868460612783652577122193510859793816995188460330824397880
20                         "31477791860458367495236868282003509517542879553533801975577055301479983488411988
21                         "13566454256529506990120591398210252551246192646658431868101252367525012654581050
22                         "80513202242966933792150637904289148738394644723861334731498656327705899476063957
23                         "1478101613631877429034747982352150153634463497389690307051"
24                 },
25             {
26                 "alpha":
27                     "8993054153525371972693030086404622579130304992917357091078431397675790112518
28                     "23093170529755739867339208237815259860508526217104674828368807366636947794528871
29                     "66775985578335541879972459012518896673591742376770068492987612722388991735178073
30                     "11688046039658489603138511444754550114043134776963149440982335781459361519918758
31                     "28783142420441397107204636342943360084977813782622463733972769255456218854227752
32                     "15904788748795127387409550820591469875173841951112593673275310804019002116955783
33                     "95456904521738039520375974359976785327130399157070231536535556898602788483521439
34                     "152000185172466981532637355092319446331912628690124603512426",
35                 "beta":
36                     "28005981258549403248020271765078337516967037347524917333891883142178449343455
37                     "97508751725447795931796271041862028816468784963598625633241442898616520057117558
38                     "22553654776378404519332803403176331189787969588979049772383786198656679222398908
39                     "47246264302754070834097277498473670091834294932093130834072510442916595976424867
40                     "67045786136053161614506152285661701024274893873321363725394664075582489167665954
41                     "82636774206667705092254911610024531706750871148873055338160168165533088391207051
42                     "0442454721754769010789834175125259568526049007187495341944889011631304576334522
43                     "67737474207452344180380338346764215759389883505701451869544"
44             },
45         {
46             "alpha":
47                 "5797550277603524071983662153526261841917419250857991929396367724626858728678
48                 "33681119556535266683370684366307991242401859326781775428264207799756886303774867
49                 "30460874796545342026615130658561574113683003278898151028195872882762977179409973
50                 "77655364108526728351567449154317803611972426794899388697036400984287695485551802
51                 "16727564236629159458189071177293675025900364728190672593507420571578981833871456
52                 "9672234825184563048970948026058455170230859924222978849007778075952021144041964
53                 "56587521662580577664759764652227587900337409250736293703379186908729016891971420
54                 "447184450715418279393483973202599390916967084735924380448450",
55             "beta":
56                 "97838221130576636987968819263998557652476740324669908865157863578599087836308

```

```

    ↵ 76992137488611360191841692735386068618127822014051965763223745942950328427351051
    ↵ 74548744990983329998353555651339847917114510284386213271167518413058326819653814
    ↵ 86660352538669139707023902121479516954449595130255017812367402821817029223068444
    ↵ 20797558226130195276122054714657395713446833616056686986988796677607173043046648
    ↵ 45132186222045049503822934508069799439193242023641396312903510039481046387072470
    ↵ 77323930041561956279266348184383636841777025420876361169353678604104501070441474
    ↵ 55846579146808386061642115923540198014231275045194117958650"
16      }
17  ],
18  "individual_proofs": [
19    [
20      {
21        "challenge": "
22          "commitment": {
23            "A": "
24              "1461462446058016608431023980976895716139611662354962433259653557759275267971
25              "104860380892101249234838768495323858269031846174905943014731532733961438503302
26              "950977676648330120635282940105885695974573537911543758330610958781966664849734
27              "461386295018092118406224465999381498873396723056737533924818620365553476988351
28              "56818397684727647170912743759887153806489928796606514686863911478815923475895
29              "324949740747904042392025374546959151775737773087768610059525234182163831024857
30              "601548081131322331016880663034330209045137932836092327400169074460991201142954
31              "2764098376733440463267866410942385165152881708201695343175940178074822401",
32            "B": "
33              "1212936380820885365829703868135912974018703608024672537012933735974970995643
34              "281945055542009504470008404986379882095412234121033516417910092820135184188402
35              "163266775850954104635564741830375045272312735456192202506426501534566141592451
36              "142791377636781419420483183400476876178072689362460684751713542846867598155078
37              "258804730680561049294940720363826229872072143942249000177381601131868366391469
38              "841532510436828391071137343673996686837355912428389937288301279862371539224693
39              "101625305716134373967937549536719728811995917760473081814560824670456995660608
40              "77718041357865349035040750517542693579680477958716866720664369838299325"
41            },
42            "response": "
43              "35111610374554771924491600617435125565551351678779095565911261872333649419481"
44            },
45            {
46              "challenge": "
47                "commitment": {
48                  "A": "
49                    "14276144535969498348508852984424238679819553017514301371490044295281127395468
50                    "759740169021445733340678443738870856692313557778755630812045903152979708663636
51                    "2754301454939582012854533706615733645393677740501015054063051202804792215156681
52                    "061430665458672370427775336691795612315361999972342423550411659500925356844239
53                    "
54                    "9020376240788799169956213071172644606691023916736130745728474521194136660164541
55                    "1313269605618412456325543223823699260607151773591958667296579781853249641422722
56                    "0676511476533760907849661718539193298363112966105771817312590096136723849042526
57                    "956522808903658610059330697057512578199924681071371152326580317305",
58                  "B": "
59                    "54937556950085798690537717077801057528298003599437494905573964520637406628771
60                    "569179938852011544293078541937184291646592619268933489506041262397373880718307
61                    "3077000544425493264757137010662955918646293253950997292177985854255364326894624
62                    "6303737080198674653042584197091660619103384668266093557290393167365856534096874
63                    "
64                    "0351193991102533307670800063408472157038404211638983177729985084696556241958814

```

```

42     "B": [
43         "1243574166134578206870611427943928014359047815647639337215422529298846002855",
44         "217549159712336979710292589184338660867498905184134659598121237810236491959166",
45         "3418044756555791518660523981157447270784595747284585992352234323356893654166704",
46         "1623979327992121539007257612419649760994289919148810039107937344697981991360066",
47         "5016886364286667331232662816285827003554969765432256192144185335740735557012081",
48         "4163576827686351193017135689967371697119230501270779433769648355301152523593938",
49         "7956087462554350776211842743552201441329876475760958332595028534413109600329696",
50         "38950737689578735869658226330986536619780002697602675596912259889"
    ],
    "response": [
        "10558534534412804489745997990430567976230006025326885338612204248676172306992"
    ]
}

```

```

51     ↵ 7336753867566548156561900118132642617357191874057103912757373683499866111598971
52     ↵ 6908347520179897429496898939108276891576369589757426770428921803469779392688330
53     ↵ 7653561547162356533872537806994594958839661879121331683002453995"
54     ↵ },
55     ↵ "response":
56     ↵ }
57     ↵ ],
58     ↵ [
59     ↵ {
60     ↵     "challenge":
61     ↵     "20540246645133510248508772126054804332279131831120275199135559606644095246601",
62     ↵     "commitment": {
63     ↵         "A":
64     ↵         "10106397188055812737742462645321636338276171374830057164294374704044606864876
65     ↵         "5054311566646696977406984940933267674815273772169834596639218620555599866407048
66     ↵         "8874714238997243528532253642972050582709298244626120326276629398778100339569399
67     ↵         "4649196965200379585444754712622209120140759830044702815807600763584031643162622
68     ↵         "1921661727355951599753467294848255718274893857202060538204136352639349622091128
69     ↵         "7604040466571495730763587231041979960970337894001811732336888781006494177264929
70     ↵         "0388513882749673519924922646696188373733535921559090047132224769418573899070908
71     ↵         "775027139806028085449248084452348088751591481368102970270273305056",
72     ↵         "B":
73     ↵         "37029221908705879439687143590563027454957624755842619465000331621721039896737
74     ↵         "5394827026470829609752465395704675823888907401709840423162559350245865771281481
75     ↵         "4181116461669517046714074995265531516967986438875535847255313956638412671864829
76     ↵         "4602480905803126242707657167626514546756126565028957318961391203842411773921268
77     ↵         "3350228654223211840164341716545372801313948692125553572869015717955752028690546
78     ↵         "1407031485881651812931613049827000519467581063644920177874129260699148887495652
79     ↵         "2497328067379825275681594386195476681143967021309091432711894323634991706766520
80     ↵         "15436437623094671839465179297869205969840906142295923716587140311"
81     ↵     },
82     ↵     "response":
83     ↵     "7673773933284719536412259876677204164956289948480475588198037430892323898227"
84     ↵ },
85     ↵ {
86     ↵     "challenge":
87     ↵     "40789319603209391044035100643943546996234603809846160735535922393763413197438",
88     ↵     "commitment": {
89     ↵         "A":
90     ↵         "15229611588330633510585960074995900185751201124687998282791828588731617663311
91     ↵         "1827916145843698916515159015585359044756854175759247294362907849993221809196506
92     ↵         "9844826054635243100265052049685763752747916551018674528553963465589641468487511
93     ↵         "5610176920758654690533035249301941021064691804893253376241913478181553485244591
94     ↵         "434687172258358308807479393872885129750885346854261023412242820490339759043606
95     ↵         "7656815536293595866526435146412469539611089256224277853367298445392870940204815
96     ↵         "6145262838706232245996375014553568235049763444923448652951516991969626840520288
97     ↵         "485483549287065955011159828896203484530430233569545704266428247444",
98     ↵         "B":
99     ↵         "21334632161958188873895465185349941827553313181129142057647121909972172104037
100    ↵         "0949412241646657185408216969463573404632886870944162064218624579304707582596963
101    ↵         "1463102706387848662412043963071493560309031416343436170178427441398470731367961
102    ↵         "9720268143378567522676309236863547630785888969434834731677831101473099368385817
103    ↵         "107766604490297640760562535040551921511550400683704116438144285322656485495629

```

```

69     ↵   8157377351307395542062054793696559633659979486600619606323390321870209617513997
70     ↵   6061273455809444891579626207508951312757273322037512929081709355594529142485152
71     ↵   26324978619128597568159858944881543489114174757701896378373443930"
72     ↵   },
73     ↵   "response":
74     ↵   [
75     ↵   ],
76     ↵   "overall_proof": [
77     ↵   {
78     ↵   "challenge": "1253752696342774217501410594567017563436392503337",
79     ↵   "commitment": {
80     ↵   "A":
81     ↵   "1179324149342428195002072934744877439171442250003770284795336756546235197855426
82     ↵   522368814443356058142917835926443970471984230941718758876370539080496399506042358
83     ↵   ,
84     ↵   "297434965651684165686362175741500299935141655668000840802550911251400036834474612
85     ↵   ,
86     ↵   "020598595215012264868696266412144652834373905616525609096783761876831280426954238
87     ↵   ,
88     ↵   "278218244798977035728872062436541551606728311786942751056715270840866948571160005
89     ↵   ,
90     ↵   "609662267601985919099928548362119811768139778048746613900502098999342388389566561
91     ↵   ,
92     ↵   "960089652150652037867504301390143961882218136190350836519202020032732723138632251
93     ↵   ,
94     ↵   "0242860182563842951236515350826520170445680809765372",
95     ↵   "B":
96     ↵   "8523081220255060312531704762208280291556391097555759256273278156642291877294539
97     ↵   379818412199740194527301592570066713785778715582408572315600649043670738675502506
98     ↵   ,
99     ↵   "322123177740755141491588887438379591015083076253246661376306636433726055276137430
100    ↵   ,
101    ↵   "774676235191595632760920893890562242435654463118609805610877865224833719224505707
102    ↵   ,
103    ↵   "546527870456472221735378530415001161474831179054944733192746637968528976877448745
104    ↵   ,
105    ↵   "102473305106168780031762765143121722822257653961728125897487534947433989719389148
106    ↵   ,
107    ↵   "224322217956613984877112974912423624014979674228133079990895185777359134409315459
108    ↵   ,
109    ↵   "048831449767034075226290672172720351277335813686911"
110    ↵   },
111    ↵   "response":
112    ↵   "13968935777329836215146425324634396744895179592294679261573217697722622897973"
113    ↵   },
114    ↵   ],
115    ↵   },
116    ↵   ],
117    ↵   "election_hash": "gb+MGZVlw8UmB1ljELcfzSeYfYEAXj9h+xqcr+eE2W0",
118    ↵   "election_uuid": "4eadbc3a-bbff-11e6-b149-b827eb1e9722"
119  }

```

## E. Verificar un voto individualmente

Esta documentación se puede consultar en la documentación de Helios Voting<sup>2</sup>.

Recall the Chaum-Pedersen proof that a ciphertext  $(\alpha, \beta)$  under public key  $(y, (g, p, q))$  is proven to encode the value  $m$  by proving knowledge of  $r$ , the randomness used to create the ciphertext, specifically that  $g, y, \alpha, \beta/g^m$  is a DDH tuple, noting that  $\alpha = g^r$  and  $\beta/g^m = y^r$ .

- Prover sends  $A = g^w \bmod p$  and  $B = y^w \bmod p$  for a random  $w$ .
- Verifier sends challenge, a random challenge mod  $q$ .
- Prover sends  $response = w + challenge * r$ .
- Verifier checks that:
  - $g^r response = A * \alpha^c challenge$
  - $y^r response = B * (\beta/g^m)^c challenge$

```

1 def verify_proof(ciphertext, plaintext, proof, public_key):
2     if pow(public_key.g, proof.response, public_key.p) !=
3         ((proof.commitment.A * pow(ciphertext.alpha, proof.challenge, public_key.p)) %
4          public_key.p):
5         return False
6
7     beta_over_m = modinverse(pow(public_key.g, plaintext, public_key.p), public_key.p) *
8     ciphertext.beta
9     beta_over_m_mod_p = beta_over_m % public_key.p
10
11    if pow(public_key.y, proof.response, public_key.p) !=
12        ((proof.commitment.B * pow(beta_over_m_mod_p, proof.challenge, public_key.p)) %
13         public_key.p):
14        return False
15
16    return True

```

In a disjunctive proof that the ciphertext is the encryption of one value between 0 and max, all max+1 proof transcripts are checked, and the sum of the challenges is checked against the expected challenge value. Since we use this proof in non-interactive Fiat-Shamir form, we generate the expected challenge value as SHA1(A0 + "- B0 + "- A1 + "- B1 + ... + ^max - Bmax) with A0, B0, A1, B1, ... ,Amax, Bmax in decimal form. (Ai and Bi are the components of the commitment for the i'th proof.)

Thus, to verify a <ZK\_PROOF\_0..max> on a <ELGAMAL\_CIPHERTEXT>, the following steps are taken.

```

1 def verify_disjunctive_0..max_proof(ciphertext, max, disjunctive_proof, public_key):
2     for i in range(max+1):
3         # the proof for plaintext "i"
4         if not verify_proof(ciphertext, i, disjunctive_proof[i], public_key):
5             return False
6
7     # the overall challenge
8     computed_challenge = sum([proof.challenge for proof in disjunctive_proof]) %
9     public_key.q
10
11    # concatenate the arrays of A,B values

```

<sup>2</sup><http://documentation.heliosvoting.org/verification-specs/helios-v4>

```

11     list_of_values_to_hash = sum([[p.commitment.A, p.commitment.B] for p in
12         ↪ disjunctive_proof], [])
13
14     # concatenate as strings
15     str_to_hash = ",".join(list_of_values_to_hash)
16
17     # hash
18     expected_challenge = int_sha(str_to_hash)
19
20     # last check
21     return computed_challenge == expected_challenge

```

Thus, given <ELECTION>and a <VOTE>, the verification steps are as follows:

```

1 def verify_vote(election, vote):
2     # check hash (remove the last character which is a useless '=')
3     computed_hash = base64.b64encode(hash.new(election.toJSON()).digest())[:-1]
4     if computed_hash != vote.election_hash:
5         return False
6
7     # go through each encrypted answer by index, because we need the index
8     # into the question array, too for figuring out election information
9     for question_num in range(len(vote.answers)):
10        encrypted_answer = vote.answers[question_num]
11        question = election.questions[question_num]
12
13        # initialize homomorphic sum (assume operator overload on __add__ with 0 special
14        ↪ case.)
14        homomorphic_sum = 0
15
16        # go through each choice for the question (loop by integer because two arrays)
17        for choice_num in range(len(encrypted_answer.choices)):
18            ciphertext = encrypted_answer.choices[choice_num]
19            disjunctive_proof = encrypted_answer.individual_proofs[choice_num]
20
21            # check the individual proof (disjunctive max is 1)
22            if not verify_disjunctive_0..max_proof(ciphertext, 1, disjunctive_proof,
23            ↪ election.public_key):
23                return False
24
25            # keep track of homomorphic sum
26            homomorphic_sum = ciphertext + homomorphic_sum
27
28        # check the overall proof
29        if not verify_disjunctive_0..max_proof(homomorphic_sum, question.max,
30                                              encrypted_answer.overall_proof,
31                                              election.public_key):
32            return False
33
34    # done, we succeeded
35    return True

```

## F. Auditoría de un voto

Esta documentación se puede consultar en la documentación de Helios Voting<sup>3</sup>.

Given a  $\langle VOTE\_WITH\_PLAINTEXTS \rangle$  and a claimed vote fingerprint, verification entails checking the fingerprint, checking all of the proofs to make sure the ballot is well-formed, and finally ensuring that the ballot actually encodes the claimed choices.

```

1  def verify_ballot_audit(vote_with_plaintexts, election, vote_fingerprint):
2      # check the proofs
3      if not verify_vote(election, vote_with_plaintexts):
4          return False
5
6      # check the proper encryption of each choice within each question
7      # go through each encrypted answer
8      for encrypted_answer in vote_with_plaintexts.answers:
9          # loop through each choice by integer (multiple arrays)
10         for choice_num in range(len(encrypted_answer.choices)):
11             # the ciphertext and randomness used to encrypt it
12             ciphertext = encrypted_answer.choices[choice_num]
13             randomness = encrypted_answer.randomness[choice_num]
14
15             # the plaintext we expect, g^1 if selected, or g^0 if not selected
16             if choice_num == encrypted_answer.answer:
17                 plaintext = public_key.g
18             else:
19                 plaintext = 1
20
21             # check alpha
22             if pow(public_key.g, randomness, public_key.p) != ciphertext.alpha:
23                 return False
24
25             # check beta
26             expected_beta = (pow(public_key.y, randomness, public_key.p) * plaintext) %
27             ↪ public_key.p
28             if expected_beta != ciphertext.beta:
29                 return False
30
31             # check the fingerprint
32             vote_without_plaintexts = vote_with_plaintexts.remove_plaintexts()
33             computed_fingerprint = base64.b64encode(hash.new(vote_without_plaintexts.toJSON()) .
34             ↪ digest())[:-1]
35
36             return computed_fingerprint == vote_fingerprint

```

---

<sup>3</sup><http://documentation.heliosvoting.org/verification-specs/helios-v4>

## G. Verificar la Totalización de una Elección

Esta documentación se puede consultar en la documentación de Helios Voting<sup>4</sup>.

To verify a complete election tally, one should:

- display the computed election fingerprint.
- ensure that the list of voters matches the election voter-list hash.
- display the fingerprint of each cast ballot.
- check that each cast ballot is correctly formed by verifying the proofs.
- homomorphically compute the encrypted tallies
- verify each trustee's partial decryption
- combine the partial decryptions and verify that those decryptions, the homomorphic encrypted tallies, and the claimed plaintext results are consistent.

In other words, the complete results of a verified election includes: the election fingerprint, the list of ballot fingerprints, the trustee decryption factors and proofs, and the final plaintext counts. Any party who verifies the election should re-publish all of these items, as they are meaningless without one another. This is effectively a re-tally".

Part of this re-tally requires checking a partial decryption proof, which is almost the same, but not quite the same, as checking an encryption proof with given randomness.

Given a ciphertext denoted  $(\alpha, \beta)$ , and a trustee's private key  $x$  corresponding to his public key  $y$ , a partial decryption is:

$$\text{decfactor} = \alpha^x \bmod p \quad (1)$$

The trustee then provides a proof that  $(g, y, \alpha, \text{decfactor})$  is a proper DDH tuple, which yields a Chaum Pedersen proof of discrete log equality. Verification proceeds as follows:

```

1 def verify_partial_decryption_proof(ciphertext, decryption_factor, proof, public_key):
2     # Here, we prove that (g, y, ciphertext.alpha, decryption_factor) is a DDH tuple,
3     # proving knowledge of secret key x.
4     # Before we were working with (g, alpha, y, beta/g^m), proving knowledge of the random
5     # factor r.
6     if pow(public_key.g, proof.response, public_key.p) != ((proof.commitment.A * pow(public_key.y, proof.challenge, public_key.p)) %
7         public_key.p):
8         return False
9
10    if pow(ciphertext.alpha, proof.response, public_key.p) != ((proof.commitment.B * pow(decryption_factor, proof.challenge, public_key.p)) %
11        public_key.p):
12        return False
13
14    # compute the challenge generation, Fiat-Shamir style
15    str_to_hash = str(proof.commitment.A) + "," + str(proof.commitment.B)
16    computed_challenge = int_sha(str_to_hash)
17
18    # check that the challenge matches
19    return computed_challenge == proof.challenge

```

<sup>4</sup><http://documentation.heliosvoting.org/verification-specs/helios-v4>

Then, the decryption factors must be combined, and we check that:

$$\text{decfactor}_1 * \text{decfactor}_2 * \dots * \text{decfactor}_k * m = \beta \pmod{p} \quad (2)$$

Then, the re-tally proceeds as follows.

```

1 def retally_election(election, voters, result, result_proof): #compute the election
2     ↪ fingerprint
3     election_fingerprint = b64_sha(election.toJSON())
4
5     # keep track of voter fingerprints
6     vote_fingerprints = []
7
8     # keep track of running tallies, initialize at 0# again, assuming operator
9     ↪ overloading
10    for homomorphic addition
11        tallies = [
12            [0
13                for a in question.answers
14            ]
15            for question in election.questions
16        ]
17
18    # go through each voter, check it
19    for voter in voters:
20        if not verify_vote(election, voter.vote):
21            return False
22
23    # compute fingerprint
24    vote_fingerprints.append(b64_sha(voter.vote.toJSON()))
25
26    # update tallies, looping through questions and answers within them
27    for question_num in range(len(election.questions)):
28        for choice_num in range(len(election.questions[question_num].answers)):
29            tallies[question_num][choice_num] = voter.vote.answers[question_num].choices[
30                ↪ choice_num] +
31            tallies[question_num][choice_num]
32
33    # now we have tallied everything in ciphertexts, we must verify proofs
34    for question_num in range(len(election.questions)):
35        for choice_num in range(len(election.questions[question_num].answers)):
36            decryption_factor_combination = 1
37            for trustee_num in range(len(election.trustees)):
38                trustee = election.trustees[trustee_num] # verify the tally
39                for that choice within that question# check that it decrypts to the claimed result
40                ↪ with the claimed proof
41                if not verify_partial_decryption_proof(tallies[question_num][choice_num],
42                    trustee.decryption_factors[question_num][choice_num],
43                    trustee.decryption_proof[question_num][choice_num],
44                    trustee.public_key):
45                    return False# combine the decryption factors progressively
46                decryption_factor_combination *= trustee.decryption_factors[question_num][choice_num]
47                ↪ ]
48                if (decryption_factor_combination * election.result[question_num][choice_num]) %
49                ↪ election.public_key.p != tallies[question_num][choice_num].beta % election.
50                ↪ public_key.p:

```

```
45     return False
46
47     # return the complete tally, now that it is confirmed
48     return {
49         'election_fingerprint': election_fingerprint,
50         'vote_fingerprints': vote_fingerprints,
51         'verified_tally': result
52     }
```