Escuela Politécnica Superior

Sistema de votación electrónica basado en criptografía

Master Universitario en Ingeniería Informática

Trabajo Fin de Máster

Autor:
Antonio José Fajardo Juan

Tutor/es:
Rafael Ignacio Álvarez Sánchez

Junio 2019

Universitat d'Alacant
Universidad de Alicante

Índice de contenidos

1. Agradecimientos

Tras casi un año desarrollando este proyecto he contado con la colaboración desinteresada de un gran número de personas y entidades que han contribuido de una forma totalmente altruista a la elaboración de Criptocracia. Si bien es cierto que el proyecto es un trabajo realizado por mí, la colaboración de estas personas ha hecho que el resultado obtenido al final pueda ser mucho más notable y como epílogo al trabajo de fin de máster realizado quiero hacer mención especial de agradecimiento a:

- **Arcadia abogados:** de la parte de D. Federico Alarcón Martínez, por su asesoramiento en algunos aspectos sobre materia legal de un proceso de votaciones.
- **Partido Popular de Torrevieja:** porque varias personas se han implicado en orientarme y ayudarme a la hora de conocer los procedimientos de una jornada electoral.
- **A mis compañeros de la asignatura DCU del Máster Universitario en Ingeniería Informática:** porque gracias a ellos he podido obtener un gran feedback en relación a las interfaces gráficas desarrolladas para el proyecto Criptocracia, así como grandes consejos para ir realizando posteriores mejoras.
- **A las personas que se prestaron a definir la arquitectura de la información:** ya que, sin su colaboración, no hubiese podido obtener información que ayudase a la definición de la arquitectura de la información y por lo tanto a cómo debían distribuirse los distintos niveles de profundidad en la navegación a través de las páginas de Criptocracia.

2. Introducción

2.1 Proyecto propuesto

El avance de las capacidades computacionales en los sistemas digitales hoy en día, sobre todo gracias al paralelismo de datos en la computación, nos ha permitido enfrentar y superar cuestiones que hace unos años eran todo un reto dentro de la informática. Debido al avance de las tecnologías y las redes de comunicaciones tenemos por delante un escenario que abarca un mundo totalmente globalizado e interconectado. Las estadísticas [1] nos demuestran que desde hace años un gran porcentaje de usuarios utiliza más de un dispositivo con conexión a Internet y por lo tanto la seguridad en las comunicaciones ha sido una cuestión cada vez más importante.

Por otro lado, este auge de las capacidades computacionales también ha servido para que la seguridad siga evolucionando. Muchos algoritmos de seguridad que en el pasado se consideraban seguros, en la actualidad ya no lo son, lo que ha generado otro reto computacional para seguir ofreciendo algoritmos criptográficos que mantengan un carácter seguro sin comprometer el rendimiento de los sistemas actuales.

Toda esta evolución tecnológica nos ha permitido que en la actualidad podamos disponer de características digitales que nos ayudan a hacer nuestro día a día más fácil, ya sea desde nuestros hogares a nuestros negocios. Existen numerosos retos en la actualidad, desde crear una Internet descentralizada y democrática desarrollando numerosos servicios a través de blockchain a aprovechar las capacidades computacionales de entornos digitales para desarrollar sistemas de aprendizaje autónomo con patrones similares al aprendizaje animal o humano, comúnmente llamados sistemas inteligentes. Parece muy obvio que hemos decidido que la tecnología debe estar al servicio de las personas para facilitarnos la vida cada día, sin embargo, existen numerosos procedimientos que pueden ser cubiertos o facilitados gracias a sistemas digitales y que en la actualidad no se están cubriendo por su carácter personal o legal y parece que seguimos confiando en los procedimientos humanos tradicionales. Es en este punto donde se hace foco sobre el proyecto propuesto.

Como su nombre indica, nos encontramos ante una propuesta de solución software que nos permitirá desarrollar un proceso de votaciones con propósito general a través de sistemas digitales. El proyecto se va a construir sobre una serie de premisas y es

importante destacarlas ya que todo lo que se va a desarrollar dentro de él va a tener que atender a estas cuestiones. A saber:

- **Sencillez:** este es un concepto clave. Definir algo como “sencillo” puede ser subjetivo por lo que tenemos que declararlo en términos cuantificables. Para ello, este concepto responde a las siguientes características. Aunque detallaremos más a fondo los requisitos del sistema en el apartado de diseño, podremos considerar que nuestro sistema será sencillo si cumple a grandes rasgos:
    - Un usuario no necesitará navegar por más de 3 páginas para realizar una determinada tarea.
    - Un usuario no necesitará tomar más de 3 minutos de tiempo para poder realizar cualquier proceso en el sistema.
- **Legalidad:** un proceso de votaciones es un proceso desde el punto de vista legal bastante delicado por lo que hay que ceñirse a las normativas y a las leyes vigentes en territorio europeo.
- **Seguridad:** estamos hablando de un sistema criptográfico seguro y naturalmente hay que considerar la seguridad como un elemento clave. Al igual que ocurre con el concepto de sencillez, definir algo como seguro es una interpretación subjetiva por lo que podremos considerar que nuestro sistema es seguro si contempla:
    - Los algoritmos de cifrado están considerados como seguros por la comunidad científica.
    - Existen mecanismos para verificar la identidad de cualquier usuario que quiera interactuar con el sistema, ofreciendo la información personalizada y correspondiente a cada uno.
    - La información que permanece residente sobre la propia base de datos relacional del servidor permanece cifrada bajo algoritmos seguros de manera que ante una intrusión o una filtración de sus datos no autorizada nadie sea capaz de revelar información confidencial que hay residente en ella.
    - La información almacenada en el servidor de forma persistente debe estar vinculada a mecanismos de protección frente a robos de las unidades de almacenamiento físico.
    - El sistema debe ser capaz de atender a cuestiones de replicación de datos sobre entornos distribuidos ofreciendo una apariencia opaca en

ese sentido a los usuarios, de manera que no se pueda apreciar a simple vista desde la perspectiva de un usuario si la información nos llega desde un servidor u otro, ya que esta mantendrá un carácter homogéneo independientemente de su origen.

- La comunicación entre los usuarios y el servidor deberá ir a través de canales de comunicación cifrados con el fin de evitar ataques MITM o extracción de datos y credenciales empleando monitores de red.
- Relativo a las votaciones, el sistema debe ser capaz de computar fácilmente el escrutinio de unas elecciones, pero no podrá conocer qué usuario votó a qué candidatura. Esto responde al principio de *secreto de sufragio* reconocido en el artículo 68 de la Constitución Española [2].
- El sistema deberá implementar mecanismos para la detección de fraude electoral. Consideraremos fraude aquel escenario donde:
  - El número de votos emitidos no se corresponda con el número de votos computados
  - Que permita a un mismo usuario votar más de una vez en unas mismas elecciones.
  - Que permita a un usuario que no se encuentra dentro de un censo electoral emitir un voto.
  - Que compute el voto de un usuario a un candidato que no es el escogido originalmente por el usuario.
  - Que considere candidaturas que no se encuentran dentro del proceso electoral.
  - Que permita verificar la integridad de los votos una vez emitidos, de manera que se garantice que no ha existido ninguna manipulación a nivel de datos con la información analizada a la hora de emitir los resultados de un escrutinio.

Analizadas estas consideraciones iniciales y antes de proceder al detalle del estado del arte en el terreno de los sistemas digitales de votación es importante hacer una aclaración sobre terminología:

Resumen criptográfico: se denomina así a un conjunto de caracteres originados a partir de una cadena de entrada cualquiera de manera que para cada entrada

obtendremos una salida única. Para considerar a esa salida un resumen debe cumplir varias características:

- Todos los resúmenes deben tener el mismo tamaño.
- Los resúmenes se deben generar muy fácilmente.
- No se debe permitir obtener la entrada a partir de la salida.

Un resumen va directamente ligado con el concepto de verificación. En el caso del proyecto, un resumen nos permitirá verificar una contraseña o validar la integridad de datos de un voto emitido. Además, debido a que no podemos obtener la cadena original a partir de su salida, cumplimos las características de seguridad indicadas previamente.

Cifrado: la técnica de cifrado es básicamente convertir una información de entrada en una información que no atienda a ningún sentido semántico o estructural. A diferencia del resumen, el cifrado sí debe permitirnos reconstruir esa información original a partir de la salida. Este proceso se conseguirá gracias a una o varias claves. Del mismo modo que ocurre con el resumen, la operación de cifrado y descifrado debe ser computacionalmente muy rápida. Dependiendo de qué tipo de cifrado queramos hacer tenemos tres tipos diferentes:

- **Cifrado asimétrico:** en el cifrado asimétrico se genera un par de claves pública y privada. La clave pública está generada a partir de la clave privada y puede ser, como su nombre indica, de dominio público. La clave privada, por el contrario, debe permanecer custodiada y no puede ser de dominio público. Este tipo de cifrados es usado frecuentemente en tareas donde queremos mantener la información con carácter confidencial a través de canales de comunicación inseguros. La criptografía asimétrica también nos permite firmar y verificar las firmas (resúmenes) de un documento con el fin de garantizar la integridad de datos.
- **Cifrado simétrico:** a diferencia del asimétrico, en este caso se utiliza la misma clave para el cifrado y para el descifrado, lo que implica que la clave no debe ser de dominio público. Este cifrado se utiliza cuando queremos dejar nuestra información residente en algún espacio público, pero no queremos que la información sea comprendida por cualquiera que pertenezca a ese espacio público. En el caso del proyecto podremos utilizarlo para cifrar nuestro propio

voto y así verificarlo a posteriori para comprobar que no se ha manipulado por un tercero.

- **Cifrado homomórfico [3]:** un cifrado homomórfico es aquel que nos permite, sin descifrar, ejecutar operaciones algebraicas. Es importante mencionarlo en esta memoria ya que, aunque no vamos a hacer uso de él, hubiese sido un cifrado perfecto para el sistema a la hora de calcular el escrutinio, sin embargo, por cuestiones de capacidades computacionales, a día de hoy con la tecnología que disponemos no es factible y por lo tanto estamos hablando de un concepto más teórico que práctico.
- **Clave pública:** se entiende por clave pública aquella clave empleada en el cifrado asimétrico y que se usa para cifrar la información. La clave pública como su nombre indica puede ser enviada a cualquiera y su conocimiento no sólo no implica un riesgo de seguridad, sino que es fundamental para poder emplear el cifrado asimétrico. La clave pública está generada a partir de la clave privada, sin embargo, conocer la clave pública no permitirá conocer la clave privada. Se puede transmitir sin riesgo a través de canales inseguros. Además, si estamos utilizando criptografía asimétrica para verificar una firma (resumen) utilizaremos la clave pública para verificar dicha firma.
- **Clave privada:** la clave privada se emplea en cifrado asimétrico para descifrar información previamente cifrada con su respectiva clave pública. La operación de descifrado debe ser rápida en términos computacionales y no permite cifrar información. Esta clave ha de permanecer en secreto y por lo tanto su distribución puede conllevar un riesgo de seguridad. Si estamos hablando de criptografía asimétrica con el objetivo de trabajar con firmas (resúmenes) utilizaremos la clave privada para generar dicha firma.

Seguridad física: es importante también considerar la seguridad no sólo desde el punto de vista computacional sino desde el punto de vista físico. Cuestiones de seguridad relativas al soporte físico donde se almacenan los datos, políticas de seguridad sobre la protección de los datos almacenados en ellos, políticas de copia de seguridad, políticas de prevención y actuación frente a accidentes o desastres como incendios, terremotos o ataques terroristas, sobrecargas eléctricas, etc.

Seguridad en los medios de transmisión: mecanismos que nos permiten preservar la confidencialidad y privacidad en las comunicaciones a través de un medio compartido, de manera que toda la información que enviamos esté protegida frente a una posible interceptación de datos por un tercero. Es importante considerar seguridad en este sentido ya que la mayoría de dispositivos con acceso a Internet lo hace a través de canales inalámbricos, a menudo de uso público o compartido, por lo que el riesgo de ataque de estas características es mayor.

En definitiva, estamos hablando de ofrecer características que nos permitan un acceso a la información al mismo tiempo que garantizamos que dicha información debe y va a ser sólo accesible a los destinatarios que se considere en el proyecto. Para ello consideramos diferentes mecanismos de **control de acceso**:

- **Autenticación:** debemos comprobar que el usuario es quien realmente ser. De esta manera la información que le ofrezcamos será la que él realmente debe recibir, discriminando accesos que no sean legítimos
- **Autorización:** no sólo es importante saber que el usuario es quien realmente dice ser, sino que además debemos tener muy claros qué contenidos le vamos a permitir ver y a qué contenidos no va a tener acceso.

Relativo a la transferencia de datos a través de redes de comunicación debemos considerar aspectos clave en seguridad:

- **Autenticación:** el usuario es quien realmente dice ser.
- **Confidencialidad:** la información correspondiente al usuario no podrá ser accesible por nadie que no sea el propio usuario.
- **Integridad:** el mensaje original emitido por el emisor no será manipulado hasta que llegue a su destino.
- **No repudio:** verificar que realmente el autor ha enviado el mensaje.

Los sistemas de autenticación van a ser dos diferentes ya que tenemos dos posibles escenarios. De modo que emplearemos un método de autenticación diferente para cada escenario.

- **Acceso por contraseña:** está basado en la idea de “algo conocido”. El usuario proporciona una palabra que sólo conoce él al sistema y el sistema se encarga de verificar que esa es realmente su credencial. De este modo verifica su identidad. La contraseña siempre debe estar protegida y además debe atender a dos conceptos: difícil de adivinar, fácil de recordar. El principal problema que presenta este modelo es que un usuario olvide su contraseña o bien que ésta sea demasiado fácil de adivinar. Para ello existen políticas para exigir una longitud de palabra mínima con el fin de evitar sistemas de fuerza bruta o bien pedir al usuario que cada cierto tiempo cambie su contraseña. En el caso del proyecto no emplearemos más que el primero. Este sistema será el utilizado para autenticar a los usuarios encargados de interactuar con el sistema. Estos usuarios son los gestores de la aplicación y hablaremos de ellos más a fondo en el apartado de diseño.
- **Acceso por módulo criptográfico:** la idea es proporcionar nuestras credenciales de identidad a través de “algo que tenemos” en este caso estamos hablando del propio DNI electrónico ya que es un soporte que todos tenemos, aunque el principal hándicap es no saber utilizarlo. Su uso está protegido a través de un PIN que se puede cambiar en cualquier comisaría de Policía Nacional en cuestión de segundos a través de un sensor biométrico, en este caso de huellas (algo que “somos”). El principal problema que representa este sistema es la dificultad que tienen muchas personas para utilizarlo ya que requiere de la instalación de un software adicional en los sistemas para que el módulo pueda ser entendido y utilizado por el equipo donde se introduce. En relación a este problema también se permitirá hacer uso de certificados digitales de persona física que van instalados en el navegador web. Al fin y al cabo, el proyecto se va a desarrollar sobre un entorno AMP (Apache + MySQL + PHP) y una vez tenemos el certificado instalado el acceso es inmediato y sin más intervenciones. El principal problema que representan estos certificados es que por un lado hay que estar renovándolos constantemente y por otro que no todo el mundo sabe solicitarlos e instalarlos, aunque es un proceso relativamente sencillo y breve (y más aún si lo haces empleando el propio DNI electrónico).

3. Estado del arte

3.1 Contexto histórico de las democracias

La razón de la existencia de un sistema de votaciones es la democracia. Los sistemas democráticos tienen su origen en Atenas para definir un sistema de gobierno donde las decisiones eran tomadas por las asambleas de ciudadanos en lugar de una única figura autoritaria como eran los reyes o emperadores. Sin embargo, existen evidencias [4] de que en la antigua India existieron repúblicas democráticas incluso antes del inicio de la democracia ateniense. Se tiene constancia de que hacia el año 400 d.C estas repúblicas desaparecieron ya que las fuerzas militares de las monarquías militaristas tomaron el control.

Las democracias modernas comenzaron a aparecer hacia la segunda mitad del siglo IV – V. Se diferencian de las democracias tradicionales porque el sistema de gobierno estaba basado en la mayoría de la población. Para nosotros, el evento histórico que más relevancia ha tenido para constituir las actuales democracias fue la Revolución francesa. En ella se recogen, entre otros derechos fundamentales de las democracias actuales como es el sufragio universal, que (en mi opinión) felizmente hemos evolucionado de su carácter exclusivamente masculino a mixto actual. Especialmente fueron relevantes los acontecimientos que se desarrollaron durante el siglo XX ya que tras la I WW la mayoría de monarquías desaparecieron o se fueron debilitando, reconociendo el derecho a sufragio a las clases pobres, a las mujeres, la descolonización de África y Asia, dando derecho a la autodeterminación de sus pueblos o el movimiento por los Derechos Civiles de los EEUU dando derecho a voto a las minorías raciales en el año 1964. No obstante, conviene destacar que también durante el siglo XX aparecieron sistemas de gobiernos rivales y contrarios a la democracia como son el fascismo o el comunismo.

3.2 Motivación del proyecto

En la actualidad la gran mayoría de sistemas de gobierno funcionan bajo un sistema de democracia rotatoria. Cada cierto periodo de tiempo, los ciudadanos son llamados a unas elecciones y en ellas eligen a sus gobernantes. Sin embargo, existen numerosas deficiencias a la hora de legitimar un gobierno presuntamente elegido por el pueblo. Los sistemas tradicionales de votación se basan en la idea de que cada ciudadano acude a un lugar determinado que le ha sido asignado para emitir su voto. A través de la

identificación de su documento legal de identidad el ciudadano emite de una forma anónima su derecho. Actualmente existen distintos modos de emitir este voto anónimo.

- **Voto en urna:** el ciudadano introduce una papeleta con la candidatura que desea votar dentro de un sobre que impide ver su contenido y deposita el voto en la urna que le ha sido adjudicada en las listas. El voto puede ser interpretado como no válido pasando a ser computado de forma especial según cada sistema si ha introducido una papeleta que no corresponde a ninguna candidatura presentada, ha introducido varias papeletas para un mismo candidato, para distintos candidatos o bien ha introducido una papeleta para un candidato, pero ésta ha sido manipulada de alguna manera.
- **Voto por correo:** en el caso de que el votante se encuentre fuera de su país o por algún motivo no pueda ejercer su derecho a voto el día de las elecciones, se complementa la posibilidad de emitir el voto a través de correo postal. El proceso es similar al anterior con la diferencia de que la papeleta con el voto debe viajar a través de correo mediante un servicio especial. Aunque a priori puede considerarse un sistema fácilmente manipulable, es un sistema totalmente aceptado y válido para países como España.
- **Voto en máquina:** en algunos países como EEUU se emplean máquinas para votar. Las tecnologías que emplean estos sistemas pueden ser de lo más variado. Algunos de ellos pueden ser:
    - **Lectura Óptica del Voto (LOV):** en este caso el sistema cuenta con un lector óptico o escáner que se encarga de reconocer la candidatura introducida en la máquina para registrar los sufragios emitidos por los electores y procesarlos. En este caso los votantes indican su selección en las papeletas mediante el llenado de una casilla. El equipo almacena el escrutinio en su memoria.
    - **Registro Electrónico Directo (RED):** este sistema es el más usado en el mundo. Consiste en marcar votos directamente sobre una máquina mediante un panel táctil. Emitirá los sufragios al cierre del proceso.
    - **Dispositivos con sistemas de marcado:** son máquinas de RED provistas de una interfaz que facilita el sufragio a personas con discapacidades.
    - **Tarjetas perforadas:** en algunos lugares aún funciona este sistema considerado totalmente obsoleto. Los votantes seleccionan sus opciones

haciendo agujeros en la papeleta y seguidamente la tarjeta es colocada en una urna para su conteo manual o sobre una máquina de tabulación.

Dentro de nuestro contexto, consideramos el **voto electrónico** una evolución del sistema RED que nos permite desarrollar un proceso electoral a través de dispositivos con acceso a Internet, aunque podamos considerar también al propio sistema RED como un sistema de voto electrónico propiamente dicho. Este sistema nos va a ofrecer una ventaja fundamental frente al resto de sistemas. La disponibilidad geográfica. Internet, como red de comunicaciones global, nos permite recibir y enviar información a prácticamente cualquier lugar del mundo prácticamente de forma instantánea. Aprovechando las redes de comunicaciones podemos desarrollar plataformas para emitir votos de forma distribuida con todas las ventajas que ello conlleva. No existirá la necesidad de desplazarse a ningún lugar concreto para emitir el voto, eliminaremos la necesidad de utilizar el voto por correo y además podemos apoyarnos sobre las posibilidades computacionales para **impedir los fraudes electorales**, una tarea que, a pesar de realizarse a través de estrictos protocolos y minuciosas auditorías, en la actualidad sigue presentando fallas y falta de garantías para demostrar que los procesos electorales han sido 100% legítimos y transparentes.

3.3 Sistemas analizados

Dentro de los sistemas de votación electrónica se han analizado varias opciones y se han obtenido diversas conclusiones.

- **Sistema de consulta de Podemos [5]:** recientemente ha surgido una polémica en torno al secretario general del partido político español Podemos. A través de su sistema propio de participación ciudadana, han realizado unas votaciones para que los militantes pudiesen expresar su postura respecto a la adquisición de una propiedad. Una de las deficiencias a nivel democrático que tiene este sistema es que los inscritos pueden votar varias veces, ya que al parecer el sistema genera un código de sesión temporal y cuando éste caduca, pueden repetir de nuevo el sufragio. De acuerdo a la información proporcionada por el propio partido, este nuevo voto reemplazaría al anterior. En términos democráticos desde un punto de vista legal, esto es una deficiencia ya que una vez emitido un voto el ciudadano no debería tener opción a cambiarlo.

- **El caso de Argentina:** la implementación del sistema Vot.Ar/BUE, como se conoce en Argentina, dejó muchas dudas dentro de la comunidad informática. De acuerdo a la investigación realizada por Iván Barrera y Javier Smaldone, se encontraron una gran cantidad de vulnerabilidades que podían ser aprovechadas para alterar el funcionamiento de unas elecciones. En particular, encontraron diversas formas de superar la autenticación frente al sistema, modificando y destruyendo votos aprovechando las características de identificación por radiofrecuencia (RFID) en las papeletas, interfaces de comunicación e incluso la posibilidad de sumar varios votos con una sola papeleta. Todas estas incidencias están recogidas en un informe público [6] y apuntan a que muchas de ellas están directamente relacionadas con el diseño del sistema.
- **El caso Scandal:** como caso de estudio complementario de análisis de vulnerabilidades en sistemas de votaciones electrónicas se propone el caso que recoge en la serie televisiva Scandal. En esta serie recogen un caso de uso donde se modifica el software de varias máquinas de votación electrónica para que arrojen un resultado distinto a la voluntad de los electores. Pese a que es un caso recogido en una serie no deja de ser un posible escenario más de fraude electoral.
- **Smartmatic**: se trata de una propuesta que ofrece una solución integral para la implementación de plataformas de votaciones online. La empresa facilita desde el soporte software a todo el hardware necesario para interactuar con los sistemas. Pantallas táctiles, etc. En definitiva, se ofrece como una solución que abarca toda la casuística de una votación electrónica. Sobre papel todo parece bastante atractivo, sin embargo, el sistema fue objeto de polémica en el año 2017, durante unos comicios en Venezuela se detectaron discrepancias entre el número de votos computados y el número de electores que votaron realmente, afirmando que existían diferencias de hasta un millón de votos. El problema no era realmente la tecnología que empleaba la plataforma en sí, sino que el sistema debía auditarse durante las elecciones. En ese caso no hubo ningún tipo de audición, lo que evidencia que pese a venderse como una plataforma que trata de automatizar todo el proceso de votación, realmente requiere de la intervención humana durante los comicios por lo que no cumpliría el objetivo de evitar que la gente tenga que estar soportando una larga y pesada jornada electoral.

4. Justificación y objetivos

La propuesta de este proyecto, denominada de ahora en adelante como Criptocracia, es una solución software que pretende cubrir procesos de votaciones de propósito general atendiendo a cuestiones legales para ser adaptable a cualquier escenario cumpliendo la legislación vigente en territorio español y atendiendo a cuestiones fundamentales de democracia.

En un principio se consideró la posibilidad de implementar esta plataforma mediante un entorno descentralizado gracias a la tecnología blockchain. Debido a que la vulnerabilidad implícita en blockchain radica en poseer al menos la mitad de los nodos que forman parte de la red, se ha decidido implementar Criptocracia a través de un sistema centralizado que a su vez puede ser replicado y escalado de forma transparente para el usuario sin comprometer la integridad del código. Los entornos de producción donde se despliegan estos nodos pueden estar custodiados y auditados por organismos oficiales del Estado en el caso de unas elecciones nacionales o en un entorno local si hablamos de un escenario más concreto, como unas elecciones internas para una empresa. De esta forma el problema mencionado de blockchain se limita a la legitimidad que se quiera otorgar a la entidad encargada de custodiar el entorno de producción, siendo ésta una cuestión subjetiva a cada persona y por lo tanto considerándolo en este caso como seguro.

El sistema debe garantizar en este sentido que hay protección contra fraudes como los anteriormente descritos.

- El votante puede votar o no en un proceso electoral, pero si vota debe hacerlo sólo una vez.
- El votante sólo podrá votar en un proceso electoral si se encuentra dentro de las listas censales.
- El votante sólo podrá votar a una candidatura presentada en un proceso electoral y no podrá votar a otra fuera de éste.
- El votante no podrá cambiar su voto una vez emitido.
- El votante podrá consultar su propio voto.
- Los votos tendrán un carácter anónimo. Se podrá saber quién ha votado en un proceso electoral, pero no a quién ha votado.

- El número de votos emitidos en un proceso electoral debe coincidir con el número de votos computados en total.
- El sistema software debe basarse en código abierto para poder realizar auditorías antes y después de las elecciones con el fin de garantizar que el código no ha sido manipulado y que además puede ser auditado por cualquier interesado sin comprometer la seguridad del proceso electoral.

5. Fase de diseño

5.1 Tecnología

El desarrollo de la plataforma se realizará a través de lenguaje PHP [7]. Se trata de un lenguaje de desarrollo basado en intérprete. Aunque el uso de PHP se puede extender a muchas otras áreas, fundamentalmente se emplea sobre entornos web ya que puede ser incrustado sobre código HTML. En la actualidad, PHP es soportado por numeroso software de servicio web, el más popular de ellos es Apache Web Server. En particular y para el desarrollo y despliegue de Criptocracia se ha realizado sobre un entorno LAMP (Linux + Apache + MySQL + PHP). El servidor consta de un sistema operativo Linux CentOS 7 x64, Apache Web Server 2.4.6, MariaDB 5.5.60 y PHP 5.6.40.

Una de las características de PHP es que se ejecuta del lado del servidor. Esto proporciona una serie de ventajas e inconvenientes. Por un lado, el código evita ser expuesto a los clientes, lo que nos permite establecer las contraseñas y las credenciales correspondientes para poder interactuar con el motor de bases de datos relacionales, así como otros parámetros que no deban ser visibles a los clientes. Por el contrario, también proporciona un problema y es que todo el coste computacional recae sobre el propio servidor, lo que nos obliga a ser bastante eficientes elaborando algoritmos para, en la medida de lo posible, evitar que el servidor quede saturado por no ser capaz de computar en un tiempo razonable la información solicitada a una cantidad de usuarios concurrentes, incumpliendo así uno de los requisitos no funcionales establecidos en la fase de planificación.

Los motivos que se han considerado para establecer este lenguaje como el lenguaje clave son, por un lado, la gran experiencia que poseo trabajando con esta tecnología y por otro lado que el desarrollo de la plataforma debía evitar la necesidad de la instalación de ningún cliente ni software específico del lado del cliente (a excepción del propio certificado digital/DNI electrónico) por lo que el escenario ideal es que el cliente trabaje sobre un navegador web. Prácticamente la totalidad de dispositivos con acceso a Internet contienen un navegador web instalado de serie y la práctica totalidad de los usuarios conocen los fundamentos para navegar por páginas web. El único punto a considerar en este sentido es la compatibilidad del navegador con la plataforma, para ello se ha hecho uso de tecnología Javascript y CSS del lado del cliente. A la hora de trabajar con el frontend de la plataforma se ha hecho uso de frameworks como JQuery

o Bootstrap o la iconografía a través de Font Awesome. Todos ellos funcionan bajo las directrices que dictamina la W3C [8] lo que nos garantiza una compatibilidad con los navegadores a largo plazo. En particular, Bootstrap nos permite desarrollar de una forma sencilla una interfaz adaptable a dispositivos móviles, con el objetivo de cumplir con otro de los requisitos establecidos en la fase de planificación. Se pretende que Criptocracia sea un software con el que se pueda interactuar desde la práctica totalidad de los dispositivos, indistintamente si es un ordenador de escritorio, portátil, tablet o móvil.

Sobre la decisión de utilizar MariaDB como motor de bases de datos relacionales es, básicamente, que es una versión de MySQL totalmente gratuita y sobre la que también poseo una enorme experiencia trabajando y que cubre al 100% las necesidades especificadas en el proyecto Criptocracia.

Durante la fase de diseño también se han tenido en cuenta numerosos aspectos relacionados con la accesibilidad y la usabilidad web. Los conocimientos adquiridos en la asignatura “Diseño Centrado en el Usuario” me han permitido trabajar mucho en ese sentido. La propia asignatura consistía en un proyecto de una aplicación aplicando estándares y patrones de diseño para permitir la interactuación de usuarios con incapacidades físicas y/o cognitivas. Ejemplos de esos patrones son el uso de no más de 3 colores predominantes, emplear colores con un sentido psicológico relacionado con su funcionalidad, emplear iconografía como apoyo o sistemas guiados como el proceso por pasos para efectuar una votación, autocompletar determinados campos o incluso un asistente de configuración inicial para los votantes la primera vez que inician sesión.

De cara a interactuar con MariaDB se ha utilizado el software phpMyAdmin con el fin de tener en todo momento una interfaz gráfica que nos permita controlar el estado y la estructura de nuestra base de datos. Además, nos permite agilizar el proceso de carga de un script SQL y de crear copias de seguridad de la base de datos.

A nivel de desarrollo comentar que se ha hecho uso de un entorno de desarrollo a través del software Netbeans para todo el código de la plataforma y de MySQL Workbench como herramienta para diseño de la base de datos. El despliegue de ficheros se ha realizado automáticamente configurando un entorno de ejecución en Netbeans y subiendo a través de FTP los ficheros al servidor en eventos “*on save*”. A continuación,

se adjunta una captura de pantalla del esquema relacional de la base de datos final generada con la herramienta phpMyAdmin.

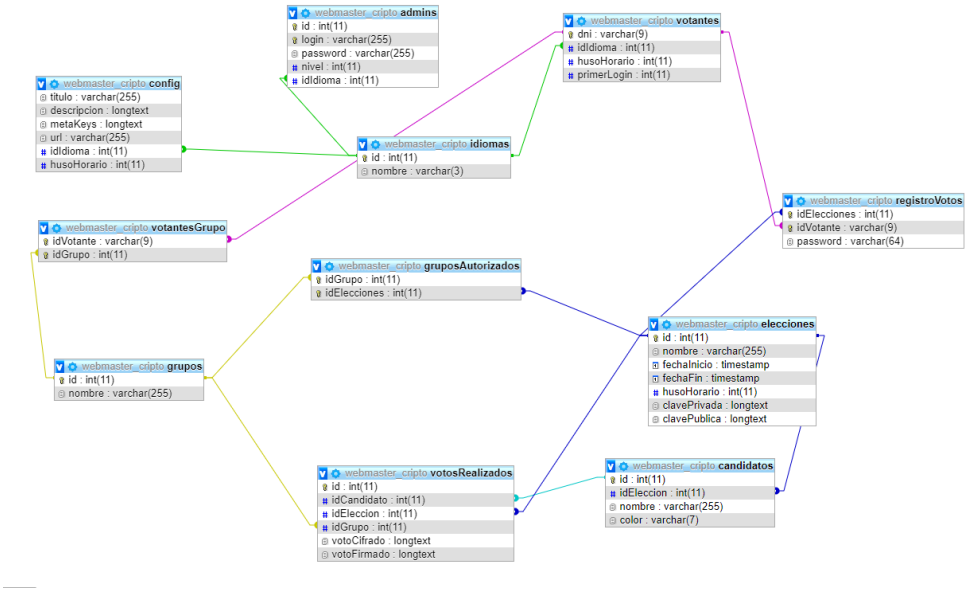

Figura 1. Esquema relacional de la base de datos generada con phpMyAdmin.

5.2 Arquetipos de usuario

Antes de centrarnos en cómo vamos a diseñar el sistema es importante conocer qué usuarios son los que van a utilizar nuestro sistema, ya que se pretende dar un enfoque orientado al usuario. Para ello y con una relación implícita de nuevo con la asignatura de Diseño Centrado en el Usuario cursada en el máster, se van a definir los arquetipos de usuario que van a utilizar nuestra plataforma:

5.2.1 Administrador

Responde a un perfil de usuario con una edad comprendida entre los 18 y los 45 años aproximadamente. Su perfil es proactivo, con conocimientos medios-avanzados en informática y además responde al perfil de responsable en el marco de la constitución de votaciones en el ámbito que proceda.

Objetivo: Instalación del software en un servidor. Interactuar con el sistema para manipular la estructura de datos del sistema y su comportamiento.

5.2.2 Joven

Se trata de un votante con una edad comprendida entre los 18 y los 50 años. Es una persona generalmente ducha en el manejo de dispositivos con acceso a internet. Comprende la mayoría de la terminología relativa a un proceso electoral y sabe usar o puede comprender cómo funciona un DNI electrónico.

Objetivo: Poder realizar votaciones en los procesos que aparezca inscrito y consultar sus propios votos, así como consultar los escrutinios y su legitimidad.

5.2.3 Jubilado

Es un votante con una edad comprendida a partir de los 50 años. Normalmente tiene problemas para interactuar con dispositivos con acceso a Internet. Suele tener constancia de cómo funciona un proceso electoral. Muy raramente ha utilizado el DNI electrónico.

Objetivo: Poder realizar votaciones en los procesos que aparezca inscrito y consultar sus propios votos, así como consultar los escrutinios y su legitimidad.

5.2.4 Snoop (fisgón)

Un usuario de edad indeterminada que anda buscando una solución software para implementar su plataforma de votaciones electrónicas. Generalmente es una persona con conocimientos técnicos informáticos, de edad indeterminada, proactiva y que comprende perfectamente el funcionamiento de un proceso electoral. Tiene conocimientos sobre el uso de DNI electrónico y otros módulos criptográficos.

Objetivo: Comprender el proyecto software, descargarlo e instalarlo en su propia infraestructura informática para personalizarlo a su gusto.

5.3 Sistemas de autenticación

Como parte de la interactuación del sistema con el usuario necesitamos validar la identidad del usuario. El sistema debe tener la certeza de que el usuario es quien realmente dice ser y en consecuencia otorgarle acceso a realizar determinadas acciones sobre el mismo atendiendo a dos criterios:

- **Nivel de acceso:** El sistema proporcionará la posibilidad de que el usuario realice determinadas funciones sobre el sistema para cambiar el

comportamiento del mismo en función del nivel de acceso que tenga. Para ello se han determinado diferentes roles de usuario:

  - **Votantes:** El usuario votante es aquel usuario que no puede cambiar el comportamiento del sistema pero que sin embargo puede realizar un voto sobre un proceso electoral siempre y cuando haya sido habilitado a tal efecto por parte de un administrador.
  - **Administradores limitados:** Son aquellos administradores que tienen la capacidad de crear procesos electorales, definiendo las candidaturas, los plazos de tiempo para votar, así como gestionar la lista de votantes y personalizar determinados parámetros del sistema tales como el título o las palabras meta o descripción del sitio para tener una versión de Criptocracia personalizada desplegada sobre el sistema.
  - **Administradores completos:** Son administradores que heredan todos los permisos de los administradores limitados y que además tienen la facultad de manipular la lista de administradores limitados, pudiendo crearlos, borrarlos o editarlos.

- **Perfil de usuario:** Este criterio es aplicable a efectos de personalizar la interfaz del usuario. Por ejemplo, mediante su perfil podemos obtener la interfaz en un idioma u otro automáticamente o podemos asumir el huso horario que tiene un votante para poder acceder al proceso electoral y ejercer su voto si hay una diferencia horaria frente al huso horario por defecto del proceso electoral.

Anteriormente se ha hablado de la necesidad de desarrollar un sistema “sencillo” y parte de esa sencillez pasa por implementar mecanismos de autenticación lo más prácticos y usables posible. El sistema de autenticación más conocido es el sistema del login, basado en proporcionar algo conocido (un usuario y una contraseña). Por ello, ese sistema es el que se ha implementado para que los administradores puedan acceder a la parte de gestión del sistema, desde el que podrán acceder a toda la lógica de negocio descrita en los roles de administrador.

Por otro lado, tenemos la necesidad de que los usuarios puedan acceder al sistema mediante un sistema de autenticación seguro, pero al mismo tiempo ágil ya que no queremos que tengan que realizar un proceso de registro previo. Para ello se hace uso de módulos criptográficos como puede ser el DNI electrónico español [9] o bien un certificado digital de identidad [10] emitido por una agencia certificadora como la FNMT

(Fábrica Nacional de Moneda y Timbre). Eso no quiere decir que el proyecto se haya limitado a esos dos sistemas, su naturaleza de código abierto nos permitirá incorporar otros módulos criptográficos o bien implementar sistemas alternativos en un futuro. Sin embargo, para este proyecto se ha considerado que ambos proporcionan un mecanismo de seguridad suficiente ya que una vez disponemos del módulo criptográfico instalado en el sistema desde el cual se pretende acceder, el acceso es trivial. Si el sistema no detectase que previamente nos hemos autenticado con esa identidad lo primero que nos pedirá es que personalicemos nuestro perfil, haciendo alusión directa al criterio "perfil de usuario" anteriormente indicado.

5.4 Cifrado

Estamos ante un proyecto cuyo principal valor es ofrecer un sistema seguro. Para ello se va a hacer uso de diferentes técnicas de criptografía desarrolladas en la asignatura del máster "Seguridad y Privacidad". No existe un mejor algoritmo de cifrado para todas las situaciones y por lo tanto, dependiendo del contexto donde se plantee aplicar cifrado es conveniente utilizar un tipo de cifrado u otro y en consecuencia un algoritmo u otro. A continuación, se detalla todo algoritmo de cifrado empleado en el proyecto y su justificación.

5.4.1 Cifrado simétrico: AES 256 bits

El cifrado simétrico como ya se ha indicado, nos permite cifrar una información a partir de una clave, dicha clave debe permanecer en custodia de la parte interesada ya que esa clave nos permitirá tanto cifrar como descifrar la información. Su uso en este caso nos interesa para cifrar el voto del usuario con el fin de poder verificar en un futuro la integridad del voto emitido.

El voto que el usuario emite lleva cifrado con AES 256 un contenido:

<DNI del votante> + <id de la candidatura>

De esa cadena cifrada al mismo tiempo generaremos una firma que se explicará más adelante. Poder descifrar esa información nos garantizará que el voto se ha computado realmente al candidato que dice haber votado.

5.4.2 Cifrado asimétrico: firma con RSA

En este caso se emplea RSA de 2048 bits junto con SHA3 de 512 bits para acelerar la firma sobre el voto cifrado. El objetivo es verificar la integridad del voto para garantizar

que no ha sido manipulado. El votante cuando emite un voto podrá descargar una copia de dicho voto cifrado y además del voto firmado, de esa forma se comparará con el voto que hay registrado en la base de datos y podremos valorar si fue manipulado en algún momento o no.

Aquí hay que tener en cuenta que la firma se efectúa con la clave privada que se almacena en la base de datos del servidor. Cada proceso electoral genera su propia clave privada de manera que no tendremos dos elecciones que puedan tener una firma idéntica, ya que a partir de dicha clave privada generaremos una clave pública que se utilizará para verificar la integridad de la firma del voto.

5.4.3 Hash: BCRYPT

El hash que obtenemos de aplicar el algoritmo BCRYPT, el cual es una función de derivación de clave y nos sirve para almacenar contraseñas en la base de datos de manera que no podamos saber cuál es la contraseña original. Emplear BCRYPT y no SCRYPT como algoritmo para obtener el resumen de una contraseña obedece tan sólo a que PHP ya implementa de serie BCRYPT y en la actualidad se considera un algoritmo seguro. Tenemos dos escenarios donde vamos a utilizar el resumen de una contraseña.

Por un lado, la propia contraseña de los administradores, contrastaremos el resumen hash de la contraseña introducida para autenticarse con el sistema y lo validaremos frente al resumen hash almacenado en la base de datos. Y por otro lado emplearemos el resumen hash para almacenar la contraseña que el votante empleó a la hora de cifrar su voto. Esto tiene doble aplicación. La primera aplicación es que necesitamos tener un cómputo de los votantes que participaron en unas elecciones, por lo que de esta manera podemos guardar en la base de datos un registro del votante, el proceso electoral en el que participó y un resumen hash de la contraseña que empleó para cifrar su voto, garantizando así que se compruebe que un votante tan sólo ha podido votar como mucho una vez en unas elecciones o si éste no ha participado, con el fin de obtener el valor de los votantes que se abstuvieron de votar. Sin embargo, esto no va a permitirnos saber a quién votó, eso es algo que tan sólo el votante podrá saber. Y para verificar en futuras validaciones de su voto que no ha sido manipulado, el sistema podrá saber si el votante realmente está intentando validar su voto a partir de la contraseña correcta, evitando así falsas alarmas de votos manipulados.

5.4.4 ¿Qué es lo que se almacena?

Para conocer los mecanismos de seguridad que Criptocracia implementa es importante conocer a nivel técnico qué información almacena en la base de datos a la hora de registrar un voto y cómo procede a realizar los escrutinios al final de un proceso electoral.

El registro del voto: cuando un usuario emite un voto el sistema guarda un registro con la siguiente estructura para verificar que dicho usuario ha participado en un proceso electoral:

Id de las elecciones (Clave primaria)	Id del votante (Clave primaria)	Contraseña
Es un entero que referencia a la clave primaria de la lista de elecciones creadas.	Es un string que corresponde al DNI del votante y que referencia a la clave primaria de la tabla de votantes.	Almacena el resumen hash de la contraseña usada para cifrar el voto.

Que tenga una clave primaria compuesta nos permite garantizar por integridad referencial que tan sólo vamos a poder tener un único registro <elecciones, votante> de manera que garantizamos que sólo va a poder votar una única vez en un proceso electoral determinado.

El voto: Este registro es el que se utilizar para los escrutinios y que tiene la siguiente estructura:

Id (clave primaria)	id del candidato	Id elecciones	Id grupo	Voto cifrado	Voto firmado
Es un valor entero autogenerado.	Es un valor entero y referencia a la clave primaria de la lista de candidatos para un proceso electoral. Se sabe a quién va dirigido ese voto, pero no quién lo ha emitido. Esta es la	Es un valor entero y referencia a la clave primaria del proceso electoral al que se	Es un valor entero y referencia a la clave primaria del grupo de votantes al que pertenece el votante.	Es el voto con la estructura anteriormente descrita cifrado con una contraseña arbitraria por el usuario mediante el	El voto cifrado está firmado mediante SHA3 de 512 bits para validar la integridad

	solución propuesta al problema del cifrado homomórfico explicado anteriormente.	quiere computar.	Nos permitirá poder discriminar los escrutinios por grupos.	algoritmo AES de 256 bits.	de los datos del voto cifrado

Con esta estructura de datos obtenemos un voto que puede ser computado fácilmente, sin costes computacionales elevados, pudiendo fragmentar los escrutinios a través de grupos y manteniendo en todo momento el anonimato de cada votante y la integridad de los votos para verificar posteriormente que no hayan sido manipulados. Además, tenemos un cómputo de cuántas personas votaron en unas elecciones, por lo que podremos comprobar que el cómputo de votos se corresponde con los votos realizados realmente, así evitamos fraudes que consistan en meter más votos de los que realmente se emitieron, creando así votantes "fantasma" que nunca acudieron a elecciones pese a estar convocados y que sin embargo se computó su voto.

El justificante de voto: finalmente el votante al emitir un voto puede descargarse en un fichero que contiene en formato JSON una estructura de datos que le permitirá validar su voto en el futuro. Dicho fichero contiene la siguiente estructura:

Id del candidato al que votó	Voto cifrado	Voto firmado

Si en algún momento alguien manipuló los datos guardados en la base de datos el usuario podrá verificar su voto con este fichero y comprobar si coincide o no coincide. En caso de que no coincida podrá demostrarse un fraude ya que previamente se demostró que ese usuario ha participado en el proceso electoral y que además cifró su voto con la contraseña que ha proporcionado.

5.4 Diseño de las interfaces

El desarrollo de las interfaces se ha realizado con conceptos estudiados directamente de la asignatura Diseño Centrado en el Usuario. Para su diseño se ha separado el sistema en dos partes: la parte pública, correspondiente a todo lo que es accesible desde la portada y la parte privada, que se refiere a todo lo que es accesible ya sea desde una cuenta de administrador o bien autenticándose con el sistema como usuario

votante. Hay que tener en cuenta que la parte pública que se especifica en la memoria corresponde a la versión comercial y que es accesible a través del sitio web oficial de Criptocracia http://www.criptocracia.org.

Antes de realizar una fase de diseño definitiva en primer lugar se ha hecho una fase de diseños de bajo nivel (mockups) empleando la herramienta MyBalsamiq con la que se han desarrollado prototipos de bajo nivel que han ido siendo validados tanto por usuarios expertos en sistemas informáticos, como son los propios compañeros del máster como por usuarios sin ningún tipo de experiencia, que han sido personas de diversas edades y que no se dedican profesionalmente a la informática o incluso están jubiladas. En base a sus criterios y opiniones se han ido construyendo las interfaces, siempre a partir de una propuesta inicial desarrollada por mí.

Figura 2. Mockup de bajo nivel desarrollado con la herramienta MyBalsamiq correspondiente a la portada del administrador de Criptocracia

Figura 3. Interfaz de la parte pública del sitio web de Criptocracia

5.4.1 Parte pública

Estructura de la página: los menús estarán ubicados en un bloque responsive sobre la parte superior de la página a través de un bloque sticky (adherido a la parte superior).

Los contenidos están distribuidos a lo largo de una página que sigue el patrón *single page* para mejorar la navegación. Serán distribuidos en formato vertical de manera que podremos hacer scroll vertical para poder ir visualizando los contenidos. Además, las distintas opciones del menú nos permitirán ir navegando a los distintos bloques distribuidos sobre la página.

Diseño de la página: la interfaz se va a basar en un título que hace referencia al enlace desde el cual se accede por el menú, con una distribución homogénea entre todos los bloques, con una única imagen para el bloque principal que se ve al inicio. A continuación del título habrá una descripción del contenido. Finalmente habrá un bloque descriptivo de características con imágenes ilustrativas.

Tipografía: se utilizarán tres fuentes predominantes: Merriweather, Montserrat y Open Sans. La clasificación por tamaños será de hasta 4 tamaños. Todas estas fuentes son de uso libre y están disponibles a través de Google Fonts [11].

Color: Se realizará una implementación alternando colores oscuros sobre fondos claros y textos claros sobre fondos oscuros. Los principales colores son el púrpura, adoptado como color corporativo del proyecto, el blanco y el negro.

Iconografía: se empleará la librería Font Awesome con el fin de aligerar el tiempo de las cargas y no sobrecargar la interfaz, ya que estos iconos se representan sobre un único color, ajustable desde las propiedades CSS.

Lenguaje visual: la iconografía hará referencia al contenido que trata de expresar el elemento del DOM, de forma adicional añadiremos un texto aclaratorio referente a cada imagen que se proyecta en el apartado donde explica las características del proyecto, siempre que el usuario sitúe el cursor o haga foco sobre una imagen, considerando que el usuario es cuando está realmente interesado en consultar dicha información y así no saturar de datos la interfaz.

5.4.2 Parte privada

Estructura de la página: el menú estará situado a la izquierda a través de bloques desplegables. La interfaz nos va a resaltar en qué lugar del menú nos encontramos y utilizaremos breadcums para que también dispongamos de otra información alternativa para conocer en qué nivel de navegación nos encontramos.

El pie de página estará situado a la derecha, en al parte inferior, en ella encontraremos información sobre el autor del proyecto y la versión de la aplicación que se está utilizando.

La parte superior de la página tendrá el logotipo de Criptocracia ubicado en la parte izquierda y en la parte derecha encontraremos información sobre la sesión abierta, pudiendo interactuar con ella para cerrar sesión.

El contenido dinámico de la página estará contenido en el bloque que queda situado a la derecha del menú, en la parte entre la barra superior y el pie de página.

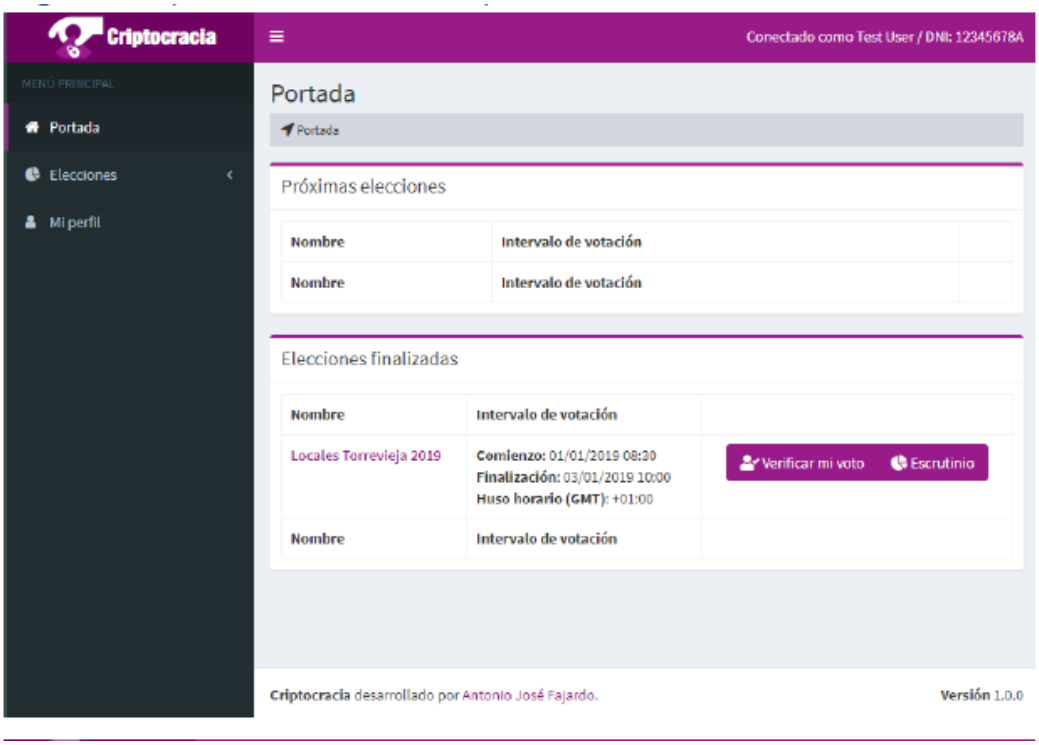

Figura 3. Interfaz visual de la parte privada de Criptocracia

Color: destacaremos un estilo elegante, sencillo y moderno. Para ello el color destacado será el púrpura. La parte dinámica estará contenida en el bloque con colores claros de fondo y oscuros de texto para generar una legibilidad mayor y los menús tendrán los colores invertidos, es decir, fondos oscuros y textos claros.

Tipografía: con el fin de mantener la coherencia con el diseño de la parte pública seguiremos empleando las mismas fuentes y la misma clasificación de tamaños.

Diseño de la página: se pretende dotar de un diseño homogéneo entre todas las páginas de la parte privada para no dificultar la navegación, por lo que los contenidos se distribuirán de forma similar y a menudo encontraremos patrones de repetición a la hora de mostrar contenidos, como por ejemplo las listas de elecciones creadas y también emplearemos elementos “modales” para resaltar acciones críticas, para que el usuario haga especial foco sobre aquellas acciones muy relevantes como podría ser eliminar un elemento de la base de datos.

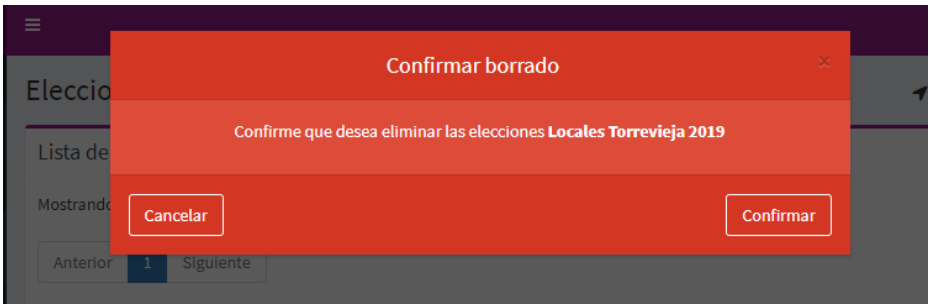

Figura 4. Ventana modal solicitando la intervención del usuario para eliminar unas elecciones creadas

Iconografía: por el mismo argumento que la parte de la tipografía, haremos uso de la librería Font Awesome.

Lenguaje visual: se busca que todos los elementos interactuables del DOM puedan combinar la iconografía con la semántica de su funcionalidad. Por ello, se combina los iconos con texto y colores representativos para facilitar la ubicación y entendimiento. Cabe mencionar que aquí se aplica una psicología de colores propia de las personas occidentales. Por ejemplo, para un occidental, el color rojo significa peligro, atención o riesgo, mientras que para una persona oriental puede tener un significado diferente. También hay que valorar a las personas con deficiencias visuales como pueda ser el daltonismo y que puedan confundir colores, por ese motivo nos apoyaremos sobre la iconografía ya que una persona que no interpreta colores de igual manera que nosotros verá el color rojo de otra forma en Criptocracia y en cualquier interfaz gráfica.

5.4.3 Patrones de uso

Una cuestión sobre la que merece hacer especial hincapié es la de los patrones de uso [12]. Estos patrones de usabilidad web para diseño de interfaces nos proporcionan una información muy relevante. Como cualquier patrón, no significa que haya que aplicarlos a rajatabla, pero nos van a permitir poder diseñar interfaces de usuario adaptadas a las conductas más típicas y repetidas de los usuarios, lo que nos va a permitir aumentar su usabilidad y que la curva de aprendizaje de Criptocracia sea lo más corta posible.

Patrón menús: en la parte pública empleamos un menú horizontal para acceder a las distintas secciones disponibles en el sitio web y en la parte privada empleamos otra variante del menú, el acordeón, que nos permite interpretar rápidamente la profundidad de las distintas categorías de la aplicación y situarnos en ellas de forma rápida y sencilla.

Breadcums: nos van a permitir orientar al usuario sobre su ubicación actual. Es muy útil cuando se hace una navegación sobre distintos niveles de profundidad.

Modales: como se ha indicado anteriormente, el empleo de ventanas modales nos va a permitir informar al usuario cuando está a punto de realizar una acción crítica para que haga especial foco en la información que queremos proporcionarle.

Rellenar blancos en formularios: rellenar formularios puede ser una tarea tediosa para el usuario. Gracias a este patrón agilizaremos el uso de los formularios.

Settings: un usuario administrador puede ser capaz de definir el comportamiento del sitio web cambiando algunos parámetros del mismo.

Expandable input: cuando el usuario tiene que introducir datos muy largos se aplica un elemento textarea en lugar del input corriente para que pueda manejarse mejor a la hora de ver la información introducida en el campo.

Dashboard: tanto la parte privada de los administradores como la parte privada de los votantes contiene un dashboard ubicado como página índice donde se hace un resumen de las votaciones, estadísticas y en definitiva proporciona un punto de partida para interactuar con el sistema.

Paso a paso: en algunos procesos como las votaciones es importante guiar al usuario de forma clara, breve y concisa. Dichos procesos pueden ser muy complejos si mostramos toda la información de golpe, sin embargo, si dividimos un proceso complejo en diferentes subprocesos más sencillos de manera que puedan ser resueltos poco a poco por el usuario facilitaremos la tarea.

Registro perezoso: un votante o un potencial usuario que quiera descargar y utilizar la plataforma Criptocracia puede interactuar con el sistema con ciertas limitaciones a partir de la versión “demo” disponible en la página web donde está alojado el proyecto. Además, en un escenario donde el proyecto esté desplegado con aplicación real el usuario podrá acceder sin necesidad de registro gracias al uso del módulo criptográfico para autenticarse con el sistema.

5.4.4 Estilo editorial

¿Qué es lo que ofrecemos? Esta primera pregunta atiende a qué es lo que se pretende ofrecer con Criptocracia. Se trata de una plataforma web desde la cual, una persona

puede realizar votaciones. Dichas votaciones pueden ser de propósito formal o informal, desde una sencilla encuesta a unas elecciones generales, autonómicas, etc.

¿Cómo somos? Como sitio web hay que representar el rol de una asociación, organización u organismo y por lo tanto habrá que dar un tratamiento formal al usuario que va a visitar el sitio.

Target objetivo: está bastante claro. El proyecto está destinado a organizaciones, asociaciones y entidades públicas, mayoritariamente. Hay que destacar que nos vamos a dirigir al usuario sin tutear, utilizando expresiones “de usted”.

Para no confundir al usuario se empleará ese tono tanto en la parte pública como en la parte privada, tanto de votaciones como de administración. Aplicando este criterio dotaremos de una mayor coherencia al sistema.

5.4.5 Arquitectura de la información

Podemos definir así al arte y la ciencia de dar forma a los productos y experiencias relacionadas con la información con el fin de fomentar la facilidad de uso y la localización de los elementos en una interfaz. En base a esta definición queda patente que la información no puede ir clasificada de cualquiera manera. Para ello, se ha llevado a cabo un estudio basado en card sorting cerrado al mismo grupo de usuarios que no necesariamente tiene conocimientos informáticos y que colaboró en el desarrollo de los prototipos de bajo nivel, explicado anteriormente.

Los resultados obtenidos fueron que los usuarios consideraban que la arquitectura de la información adecuada para Criptocracia era la siguiente:

- Administración
    - Alta votaciones
    - Alta candidatos
- Votaciones
    - Votar
    - Consultar voto propio
    - Consultar votos de elecciones
- Personalizar
    - Escoger idioma
    - Ajustar opciones de la aplicación.

6. Desarrollo

6.1 Metodología de implementación

El desarrollo del proyecto Criptocracia parte de la idea de emplear una metodología ágil de desarrollo de software. Hay muchas razones que nos invitan a emplearlas y a lo largo del grado y del máster se ha visto que son enormemente útiles para conseguir un producto final que ofrezca una gran fidelidad a los requerimientos funcionales y no funcionales que se establecieron durante la fase de diseño. En particular, se ha optado por el uso de la metodología de programación extrema [13] (también conocida como Extreme Programming o XP).

A nivel histórico para contextualizar cabe mencionar que esta metodología fue aplicada por primera vez en un proyecto real en el año 1996 y desde entonces ha sido aplicada en numerosos proyectos de software de todo tipo de envergaduras con un fantástico resultado. Los principales valores a destacar en esta metodología son:

- **Simplicidad:** se parte de la idea de realizar un desarrollo sencillo para permitir futuras extensiones de funcionalidad del software.
- **Comunicación:** cuando XP es aplicada sobre equipos de trabajo es muy importante que exista una comunicación fluida y transparente entre todos los miembros involucrados en el proyecto. Uno de los objetivos es tener claro en todo momento los objetivos del proyecto y durante la fase de desarrollo que exista una filosofía de trabajo en la misma dirección, haciendo que todas las cabezas piensen en el mismo sentido y no haya lugar a distintas interpretaciones. La comunicación también influye con cualquier stakeholder del proyecto y eso implica que los usuarios que han contribuido en el rol de “cliente final” para elaborar los arquetipos de usuario, así como para definir la arquitectura de la información (no olvidemos que es un proyecto desarrollado desde una perspectiva orientada al usuario) también se ven involucrados en este concepto de comunicación.
- **Retroalimentación:** el desarrollo no se considera desde que se escribe la primera línea de código hasta que el producto está totalmente terminado. Como cualquier metodología ágil de desarrollo lo que buscamos es conseguir productos tangibles y usables en pequeñas iteraciones que nos permitan obtener un feedback de aquella persona que interpreta el rol de product owner que en

nuestro caso viene a ser los clientes finales. En otras metodologías ágiles como Scrum no necesariamente tiene que ser el cliente final ya que este rol puede ser interpretado por un miembro concreto del equipo de desarrollo. En XP estamos hablando de aquella persona que se va a encargar de definir el product backlog y nos va a estar indicando qué es lo que espera que el producto haga o cómo debe comportarse y cómo nos debe permitir interactuar con él. Del mismo modo también nos dirá cómo de preciso es el software que desarrollamos y lo expresará siempre en términos cuantificables mediante tests de aceptación. En definitiva, desarrollando el proyecto a través de pequeños (y numerosos) hitos conseguiremos obtener en periodos breves de tiempo productos para ser evaluados y así poder orientar el desarrollo correctamente corrigiendo fallos o refactorizando módulos del proyecto para que cumplan con los criterios de aceptación y evaluación al que es sometido por parte del product owner y al final del proyecto tengamos una garantía de que el proyecto cumple todos los requerimientos establecidos en fase de desarrollo.

Durante el desarrollo del proyecto se justifica la elección de XP como metodología ágil por:

- **Desarrollo iterativo e incremental:** el proyecto se ha ido desarrollando en pequeños módulos con un enfoque top-bottom ya que como cualquier proyecto orientado al usuario hay que hacer un especial hincapié en el desarrollo de las interfaces en las etapas más tempranas. Partiendo de esa base no hay un nivel más alto en la escala de desarrollo de software que la capa de presentación, por lo tanto, es justificable que durante la primera fase del proyecto los hitos estuviesen directamente relacionados con la fase de diseño y hablando estrictamente de desarrollo, con la creación de las interfaces y la dotación de funcionalidad, avanzando a niveles más bajos hasta llegar al backend.
- **Pruebas continuas:** pruebas y testing continuos. Para elaborar las pruebas de aceptación a los usuarios que han ido probando las interfaces gráficas se han establecido métricas cuantificables como son el número de clicks que el usuario ha hecho para realizar una determinada tarea, el tiempo que ha tardado en completar una tarea y finalmente su valoración y satisfacción sobre la funcionalidad en cuestión enmarcada en un rango discreto de valores. El proyecto ha contenido pruebas que abarcan las propias unidades, probando

métodos, módulos y unidades del proyecto en primera instancia como funciones y métodos por separado, avanzando por pruebas de integración y finalmente pruebas de sistema. No llegaría a considerar que se ha empleado un desarrollo dirigido por pruebas (TDD) ya que los requisitos demandados en la fase de diseño no han sido desarrollados en función de ningún test ni tampoco se han desarrollado las unidades considerando funciones, procedimientos o métodos los valores que debían devolver en función de una determinada entrada sino que desde un primer momento se ha desarrollado su comportamiento y luego se han ido probando diferentes entradas y contemplando sus salidas.

- **Integración entre desarrolladores y cliente:** en consonancia con la importancia de la comunicación entre cualquier stakeholder del proyecto, existe una integración permanente del equipo de desarrollo (en el caso del proyecto, por descontado soy el único desarrollador) y el cliente, ya que han sido los propios usuarios quienes han ido dirigiendo toda la fase de desarrollo y han contribuido a definir qué es lo que esperaban del sistema y cómo esperaban manejarse con él.
- **Corrección de errores:** existe una filosofía de comprobación antes de añadir un nuevo módulo al sistema se verifica que éste funciona correctamente de forma aislada y a continuación se comprueba que su integración no ha generado problemas ni en el sistema ni en el resto de componentes ya integrados. Esto es intrínseco al testing continuo, ya que como se ha indicado se aplican pruebas de forma continua.
- **Refactorizar código:** con el fin de mejorar la mantenibilidad y la escalabilidad a nivel de código del proyecto se ha hecho uso de patrones de diseño software. En concreto se ha utilizado MVC [14]. No se ha empleado un framework MVC concreto, sino que se ha hecho una implementación propia. La filosofía es que el modelo solicita al controlador la información necesaria, el controlador solicita al modelo los datos necesarios y luego se los ofrece a la vista para que ésta devuelva al usuario la información solicitada en un formato comprensible para él. Dicho de otro modo, el usuario solicita a través de un javascript mediante un evento AJAX información, por ejemplo un login, para ello el usuario proporcional a controlador un usuario y una contraseña, el controlador solicita al modelo que compruebe si es correcto o no, en función de lo que el modelo diga el controlador solicitará a la vista que devuelva al usuario una información u otra, al controlador

le da igual qué sea esa información, ya se encargará la vista de generar el código HTML necesario para que el usuario pueda visualizar su dashboard si ha iniciado correctamente la sesión o bien el mensaje de error correspondiente si no ha sido exitoso. Por ejemplo, bien porque las credenciales sean incorrectas o porque falte por rellenar alguno de los dos campos solicitados. MVC nos ofrece un menor acoplamiento y una mayor cohesión en nuestro código por lo que cualquier modificación que se quiera hacer en el futuro implicará una menor inversión de recursos. Ese es uno de los principios de cualquier proyecto software desde el punto de vista ingenieril.

- **Propiedad de código compartida:** el código está visible para todos los desarrolladores. Aunque se puede hacer uso de herramientas de control de versiones y cada uno puede trabajar sobre su respectiva rama para no colisionar con el código de otro desarrollador, es importante que el código sea visible a todos, ya que de esta manera cualquier desarrollador puede consultar y/o documentar fácilmente cualquier cuestión relacionada, por ejemplo, con la implementación de un método de clase, no limitándose a ver su interfaz. Esto ayudará a corregir también pruebas de regresión en caso de que a la hora de realizar pruebas de software algo falle. En el caso de Criptocracia esto se ha llevado al extremo al distribuirse como código abierto.
- **Simplicidad del código:** este es el reto más recurrente en los proyectos de software. Un código sencillo de mantener y sobre todo de entender aportará mayor valor al negocio a largo plazo ya que los tiempos empleados en su mantenibilidad y extensión serán menores. El uso de patrones de software y de frameworks ayuda a dicha simplicidad. Por ejemplo, frameworks como Bootstrap o JQuery están muy bien documentados y son de dominio público por lo que ante cuestiones de implementación tendremos a nuestro alcance un amplio repertorio de guías y feedback para solucionar cualquier duda.

6.2 Gestión y planificación del proyecto

Cualquier proyecto software debe cumplir con los tiempos establecidos para su desarrollo, debe ser documentado debidamente y sobre todo debe garantizar que las funcionalidades requeridas para el mismo queden desarrolladas y probadas antes de ser presentado al usuario final. Durante el diseño y desarrollo pueden surgir muchos contratiempos derivados de cambios de requisitos o imprevistos durante ese periodo de

tiempo. Una correcta planificación y gestión del proyecto nos servirá para definir estrategias para su avance y control de calidad, así como cualquier tipo de medida de prevención y contingencia ante los posibles riesgos que puedan surgir.

Como ingeniero informático, he trabajado diferentes herramientas para gestión de proyectos y para mantenimiento de código. Sin entrar a realizar una comparativa o una valoración individual he considerado herramientas para control de un proyecto y herramientas de control de versiones para la gestión del código.

Las herramientas de control de versiones son muy importantes para el desarrollo de un proyecto. En numerosas ocasiones podemos encontrarnos con una implementación que produce fallos y debemos retornar a un estado donde ese fallo no existía o queremos conocer qué cambios se han realizado sobre un fragmento de código. Incluso también hay que considerar momentos en los que tenemos que fusionar código realizado por otro desarrollador con el nuestro. En este caso, para desarrollar Criptocracia sólo he participado yo a nivel de implementación por lo que esta última cuestión no ha sido considerada. Por ello y atendiendo exclusivamente al concepto de control de versiones decidí utilizar la herramienta **Dropbox** para gestión del código. Dropbox nos permite restablecer un fichero a un estado anterior y va recordando cada nueva versión del fichero que hemos introducido. No es muy relevante conocer qué fragmento de código hemos cambiado ya que como se ha indicado anteriormente, realizamos un proceso de testeo continuo y se controla con gran precisión el impacto de cada cambio. Por lo tanto, Dropbox se ajusta como una herramienta excelente para las necesidades consideradas en este proyecto. Además, estamos empleando un mecanismo de almacenamiento de datos que almacena nuestro código de forma cifrada en la nube y distribuye sobre distintos soportes de almacenamiento de datos físicos mediante técnicas de redundancia de datos ofreciendo una sincronización entre distintos equipos que trabajen con su cliente conectado a una misma cuenta o a diferentes cuentas que compartan el código. Además, hay un motivo aún más importante que ha permitido a Dropbox decantarse como el medio elegido. Otros sistemas de almacenamiento en la nube similares como Google Drive exigen autoría de la información almacenada, en cambio Dropbox no. No te pide que cedas la autoría ni la compartas, sigues siendo el legítimo propietario de los datos publicados en su servicio.

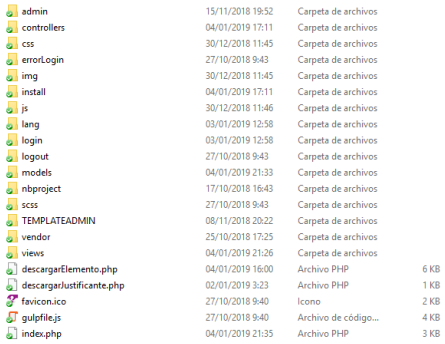

Figura 5. Ficheros de Criptocracia sincronizados con Dropbox sobre un sistema operativo Windows.

En cuanto a la gestión del avance del proyecto, una de las técnicas que más me han gustado a la hora de dirigir proyectos de software sobre metodologías ágiles es el método Kanban [15] consistente en un sistema de información que controla el avance de un proyecto mediante su descomposición en subprocesos, pudiendo controlar en todo momento su estado actual así como los recursos asignados al mismo, lo que nos permite una distribución de carga más eficiente, una metodología de desarrollo en consonancia con las metodologías ágiles y una optimización de los recursos humanos disponibles para un proyecto.

Para trabajar con esta metodología se hace uso de un tablero Kanban. Como todo, soy partidario de emplear la tecnología a nuestro servicio. En Internet hay una gran cantidad de recursos que imitan la funcionalidad de un tablero Kanban. Mi preferida es **Trello**. A modo de resumen podemos definir a esta herramienta como un tablero Kanban que nos permite documentar las funcionalidades de nuestro proyecto, gestionar los recursos a emplear, notificar fechas de vencimiento y fomentar el feedback y la comunicación entre los stakeholders implicados en cada tarea, lo que aumenta el desempeño y la transparencia.

En el caso del tablero Kanban que se ha empleado para el desarrollo de Criptocracia se ha descompuesto en diferentes columnas, siendo cada columna del backlog un estado definido:

- **Sin asignar:** funcionalidades que están pendientes y que aún no tienen un recurso asignado.
- **En desarrollo:** funcionalidades que se encuentran en desarrollo y tienen, al menos, un recurso asignado.

- **Testing:** funcionalidades que han superado la fase de desarrollo y se encuentran en fase de pruebas para verificar que cumplen los criterios de aceptación establecidos por el cliente. Hasta que no tengamos unas métricas cuantificables que permitan situar dicha funcionalidad como aceptable no consideraremos una funcionalidad terminada.
- **Realizado:** en esta fase englobamos todas las funcionalidades desarrolladas y probadas. Es importante que dichas funcionalidades queden bien documentadas ya que en el futuro es posible que deban regresar al comienzo del backlog.

Cada funcionalidad definida en el backlog se representa con tarjetas. Dentro de las tarjetas podemos encontrar campos más complejos como elementos “ToDo” que no son más que una checklist que divide una tarea en varias subtareas más sencillas para poder ir avanzando sobre iteraciones más simples manteniendo siempre una consonancia con la mentalidad ágil.

Figura 6. Ejemplo de una tarjeta Kanban en Trello con una tarea y elementos ToDo

Además, a nivel de notificación también tenemos la posibilidad de definir fechas de vencimiento sobre una tarjeta que nos permita conocer en qué momento nos estamos retrasando con el proyecto y notificar automáticamente a todos los stakeholders involucrados en dicha tarea. Como se ha dicho anteriormente, Trello también nos permite fomentar el feedback entre stakeholders y la transparencia. Para ello cuenta con un sistema de comentarios libres en las tarjetas que nos permitirá grabar sobre ellas cualquier dato que consideremos y que será visible para todos los miembros del proyecto.

7. Mercado

7.1 Despliegue en el mercado

La aplicación, así como la totalidad de su código fuente y recursos serán publicados con carácter copyleft [16]. Para ello, se ha llevado a cabo un estudio de despliegue incluyendo las características de un equipo de desarrollo para mantenimiento del software, costes derivados del d espliegue, riesgos y otros factores que se detallan a continuación:

La estructura del equipo: debe ser una estructura con jerarquía. Este proyecto ha sido desarrollado íntegramente por mi como proyecto para el trabajo de fin de máster universitario y por lo tanto planteo la necesidad de una jerarquía a la hora de implementar nuevas funcionalidades o cambios en el software. Un ejemplo de otro proyecto importante que sigue unas pautas similares a lo que pretendo proponer con Criptocracia es el kernel del sistema operativo Linux [17], donde podemos encontrar a una enorme cantidad de desarrolladores aportando líneas de código al kernel, dicho código modificado se envía al respectivo mantenedor del subsistema, los mantenedores de cada subsistema envían el código a su rama del repositorio git y finalmente será Linus Torvalds quien decida qué cambios se fusionan con el código del kernel y qué cambios son descartados. Evidentemente en el caso del proyecto Criptocracia ni existen ni se pretende que exista una comunidad de desarrolladores de tal magnitud aportando y manteniendo el código por lo que existiría libertad para enviar código al proyecto, pero en última instancia yo o la persona delegada por mi decidiría qué código se incorpora al source del proyecto y qué código es descartado. De manera independiente, cada desarrollador puede incorporar código o cambiar el existente en el proyecto Criptocracia original con el fin de desarrollar su propia versión personalizada acorde a sus necesidades, pero como obliga la licencia Copyleft, ésta debe mantener la misma licencia.

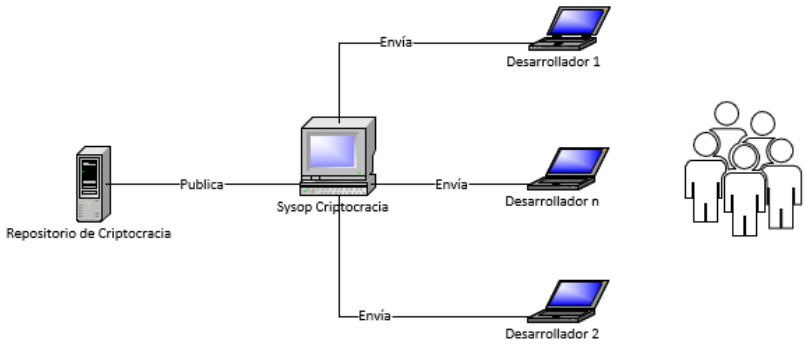

Figura 7. Estructura de desarrollo planteada para Criptocracia

Roles:

- **Sysop:** representa la figura de aquella persona encargada de aprobar todas las aportaciones de código que se pretendan realizar sobre el source original. Generalmente esta figura será representada por mi o bien por alguien a quien haya delegado la responsabilidad.
- **Desarrollador:** debido al carácter open source del proyecto cualquier persona interesada puede descargar y tener acceso al código fuente de Criptocracia. Del mismo modo que puede manipularlo para uso propio (respetando siempre la licencia Copyleft) puede contribuir proponiendo modificaciones sobre el código fuente original a fin de corregir errores de implementación de cualquier tipo (parches) o añadiendo nuevas funcionalidades.

Especificación de la solución implementada

- **Lenguaje de programación:** se trabajará con lenguaje para el backend de PHP. Para el frontend se trabajará XHTML. El estilo estará basado en hojas CSS y Javascript para funcionalidades de cara al cliente. Se hará uso de los frameworks de Bootstrap y JQuery respectivamente con el fin de agilizar los plazos de desarrollo y trabajar sobre tecnologías testadas. Actualmente, ambos framework se encuentran en evolución y trabajan sobre los estándares dictados por la W3C, lo que mantendrá una consonancia con la usabilidad deseada y mencionada en fase de diseño y navegadores web en el futuro.
- **Entorno de desarrollo:** se optará por una IDE de desarrollo gratuita. El caso real durante el desarrollo de Criptocracia se ha trabajado con Netbeans IDE, que

mantiene una excelente compatibilidad con lenguaje PHP, Javascript y CSS así como una actualización vía FTP en tiempo real sobre cada modificación que se ejecute sobre ficheros. Al ser un solo desarrollador y contar con Dropbox como mecanismo de almacenamiento y control de versiones no tenemos que trabajar con repositorios locales y luego hacer ningún tipo de fusión con el source original como sí ocurriría con otros sistemas de control de versiones como Git. En cuanto al diseño y generación de scripts para la base de datos relacional se trabajará con MySQL Workbench y para automatización de pruebas se trabajará con PHPUnit.

- **Protocolo de red:** HTTP 1.0.
- **Persistencia de datos:** se trabajará MariaDB como sistema de base de datos relacional.
- **Servidor web:** se empleará Apache Web Server.
- **Protocolo de comunicaciones:** utilizaremos HTTP y HTTPS en el caso del backend una vez se ha iniciado sesión, bien como administrador de Criptocracia, bien como votante. En el caso del votante la comunicación mediante HTTPS es de carácter obligatorio para poder emplear correctamente los módulos criptográficos como mecanismos de autenticación.

Requisitos del entorno de producción (funcionales):

- Los usuarios deben poder autenticarse con un módulo criptográfico.
- Los administradores de Criptocracia deben poder autenticarse con un usuario y una contraseña arbitraria.
- Los administradores de Criptocracia deben poder desplegar una instancia de Criptocracia en sus propios sistemas si lo desean.
- Los usuarios deben poder emitir votaciones atendiendo a los criterios legales establecidos en territorio español.
- Los usuarios deben poder consultar su voto, así como su integridad.
- Se permite el acceso concurrente desde diferentes equipos a una misma cuenta de administrador o de votante.
- Los usuarios votantes deben poder consultar todos los sufragios emitidos que hayan quedado almacenados en la base de datos.
- No se permitirá la consulta de un sufragio no emitido por un usuario determinado por parte de otro usuario diferente.

- El sistema deberá implementar mecanismos que permitan detectar los diferentes tipos de fraude que pudiesen tener lugar, ya sea por una manipulación directa de la información almacenada en la base de datos, o por una manipulación del código fuente de Criptocracia.
- Nadie que no esté debidamente autenticado debe tener acceso ni a la administración de Criptocracia ni a la consulta de sufragios.
- Sólo los usuarios que hayan sido convocados a unas elecciones podrán consultar los escrutinios de la misma.
- Los usuarios podrán personalizar el idioma de la interfaz a fin de obtener un contenido comprensible.

Requisitos no funcionales:

- El servidor debe ser capaz de ofrecer respuesta a los usuarios ante peticiones concurrentes en tiempos no superiores a 5 segundos (no se contabiliza el tiempo que tarde el navegador del cliente en procesar el código obtenido para representarlo en el DOM).
- El servidor de producción debe atender a cuestiones de escalabilidad permitiendo la incorporación de nuevos nodos sin interrumpir el servicio y de una forma transparente al usuario.
- El servidor de producción ofrecerá la posibilidad de atender diferentes conexiones concurrentes manteniendo el carácter individual de cada una de ellas.
- El servidor de producción ofrecerá un comportamiento funcional similar independientemente de su ubicación geográfica.
- El servidor de producción debe implementar mecanismos de protección e integridad de datos con el fin de mantener la confidencialidad de los datos que almacena, incluso a aquellas personas que pudiesen tener acceso físico a los soportes físicos donde se esté alojando el proyecto.
- El servidor de producción debe además disponer de mecanismos de copia de seguridad para preservar la información ante posibles fallos de lectura o escritura de los soportes físicos de almacenamiento.
- La comunicación entre el servidor y el usuario deberá ir cifrada en aquellos escenarios donde exista una autenticación con el fin de evitar ataques de suplantación de identidad y MITM.

- Los algoritmos de cifrado que se empleen para preservar la confidencialidad y la privacidad de los datos almacenados deben ser considerados “seguros” a fecha de julio del año 2019.
- El sistema debe ofrecer una interfaz gráfica coherente, acorde a las directrices de usabilidad y accesibilidad marcadas por la W3C y que cumpla, al menos, un nivel AA según la WCAG 2.1.
- El sistema debe cumplir con la ley vigente española en materia de protección de datos y en el reglamento de medidas de seguridad.

Costes estimados: se estiman una serie de costes derivados del desarrollo de Criptocracia así como su despliegue. Los costes estimados son considerados mensuales y cubren tanto los costes de su despliegue y mantenimiento en producción. Se ha tomado como proveedora de servicios la empresa Cubenode con quien actualmente se tiene contratado un servidor de idénticas características donde el proyecto Criptocracia está desplegado.

Concepto	Unidades	Cantidad estimada (€)
Servidor dedicado Intel Xeon E5-4655 V4 @ 3,2 GHz + 16 GB RAM DDR4 1600MHz ECC + 240 GB SSD (RAID 1 + 0) + Linux centOS 7.6	1	36,20
Sueldos empleados mensual (en bruto)	1 Único desarrollador del proyecto. Sueldo asignado de 2100 €.	2100
Equipo portátil Dell XPS 9570 i7 8750 HQ + 16 GB RAM DDR4 @ 2666 MHz	1	1899

+ 512 GB SSD M.2 @ 2280 MHz + Linux Ubuntu		
Campaña de publicidad Google Adwords (inversión mensual)	1	250
Campaña anuncios Facebook (inversión mensual)	1	250
Riesgos	1	1000
Registro de denominación comercial (electrónica)	1	125
Registro dominio (anual)	1	11
Gastos derivados del desarrollo (luz, línea de internet, etc)	1	60
Total € constitución + 1º mes		5731,20
Total € coste sucesivos meses		2696,20
Total € coste durante el primer año		35389,40
Total € coste anual sucesivos años		32365,40

Hay que considerar que sobre toda esta inversión 1000 € están apartados para riesgos, se contabilizan durante el primer año, pero no en sucesivos. Además, el volumen de dinero ingresado en campañas de publicidad es limitado, hay que considerar que no todos los meses se van a hacer campañas igual de agresivas ni tampoco con carácter permanente por lo que estos costes se verán reducidos a la baja.

Riesgos del proyecto

El proyecto se puede ver amenazado por una serie de riesgos. Se catalogarán como “alto”, “medio” y “bajo” atendiendo al impacto que producen, la probabilidad con la que

se producen, repercusión, exposición y amenaza. Para comprender esto es importante aclarar en primer lugar estos conceptos:

- **Amenaza:** se trata de un evento que puede desencadenar un incidente durante el desarrollo o despliegue del proyecto produciendo daños materiales o pérdidas de algún tipo. También se consideran amenazas incumplimientos a la normativa legal vigente en territorio español sobre el tratamiento de la información que exista sobre Criptocracia.
- **Riesgo:** es el factor a tener en cuenta a la hora de priorizar actuaciones. Si algo posee un grado mayor de riesgo, la prioridad para resolver dicha incidencia será mayor.
- **Probabilidad:** va determinada por el grado de infracción para los aspectos que se evalúan sobre un riesgo.
- **Impacto:** cuantifica la gravedad de las consecuencias producidas cuando una amenaza se materializa.
- **Exposición:** se trata del factor que mide el grado de visibilidad ante un riesgo.
- **Repercusión:** es el factor que determina el grado en el que el funcionamiento del proyecto en términos generales se puede ver alterado en relación a interrupción o seguridad.

La relación del impacto se obtiene a partir de la exposición y la repercusión sobre los valores descritos en la siguiente tabla:

EXPOSICIÓN	REPERCUSIÓN	IMPACTO
NO	NO	**BAJO**
NO	SÍ	**MEDIO**
SÍ	NO	**MEDIO**
SÍ	SÍ	**ALTO**

Probabilidad:

- **Baja:** aplicable cuando ocurre como mucho para un 15% de los casos de uso.
- **Media:** aplicable cuando ocurre entre un 15% y un 60% de los casos de uso.
- **Alta:** aplicable cuando ocurre para más de un 60% de los casos de uso.

La relación del riesgo se obtiene a partir de la probabilidad y el impacto:

PROBABILIDAD	IMPACTO	RIESGO
BAJA	BAJO	**BAJO**
BAJA	MEDIO	**BAJO**
BAJA	ALTO	**MEDIO**
MEDIA	BAJO	**BAJO**
MEDIA	MEDIO	**MEDIO**
MEDIA	ALTO	**ALTO**
ALTA	BAJO	**MEDIO**
ALTA	MEDIO	**ALTO**
ALTA	ALTO	**ALTO**

De esta relación se priorizará la resolución de incidencias atendiendo a su riesgo, como se ha dicho anteriormente se dará prioridad a aquellas cuyo riesgo sea más alto ya que poseen unas consecuencias más significativas para el avance y desarrollo normal del proyecto y por lo tanto para su rentabilidad y viabilidad.

Registro de riesgos

Identificador	Acción que lo determina	Consecuencia	Probabilidad	Impacto	Riesgo
Lista de riesgos internos: Dependientes de Criptocracia					
R1	Problemas escalando el sistema	El sistema se cae	Bajo	Alto	Medio
R2	El sistema de autenticación falla	Accesos no autorizados	Bajo	Alto	Medio

R3	El servidor se queda sin espacio en disco	El sistema se bloquea	Medio	Alto	Alto
R4	La documentación del proyecto no está actualizada	Problemas para que un desarrollador pueda editar el proyecto	Bajo	Bajo	Bajo
R5	Las copias de seguridad están corruptas o no son accesibles	Las copias de seguridad desaparecen y no contamos con backup	Bajo	Medio	Medio
R6	Equipo informático falla	Se interrumpe el trabajo del desarrollador	Bajo	Alto	Medio
R7	Error de certificado SSL	Los clientes visualizan un error al acceder a la plataforma	Bajo	Bajo	Bajo
R8	Incorporación de un nuevo desarrollador	Las tareas deben reasignarse	Medio	Medio	Medio
Lista de riesgos internos: No dependen de Criptocracia					
R9	Algoritmos de cifrados rotos	La confidencialidad de los datos alojados está comprometida	Bajo	Alto	Medio
R10	Cambios en la LOPD	El proyecto deja de cumplir con la ley vigente.	Medio	Alto	Alto
R11	Ataques DoS al sistema	Interrupción del servicio	Medio	Alto	Alto

R12	Navegadores actualizados incompatibles	Problemas de usabilidad y accesibilidad sobre el frontend	Bajo	Medio	Medio
R13	El proveedor de servicios está en mantenimiento	Interrupción del servicio	Bajo	Alto	Medio
R14	Intrusión en el equipo cliente	Suplantación de identidad / Robo de credenciales	Alto	Bajo	Medio
R15	Administrador deja de tener acceso a su cuenta email y no recuerda su contraseña	El administrador no puede acceder al sistema e interactuar con él para gestionar la plataforma	Bajo	Bajo	Bajo
R16	El cliente deja de tener un certificado digital válido para acceder al sistema	El cliente no puede acceder al sistema para votar	Alto	Bajo	Medio
R17	Acceso físico no autorizado al sistema	Pérdida de datos e información. Riesgo de revelación de datos	Bajo	Alto	Medio
R18	Cliente manipula su justificante de voto	El cliente no puede verificar su voto	Bajo	Bajo	Bajo
R19	Se manipula el código original de Criptocracia para unas elecciones	Diferentes fuentes podrían señalar al proyecto Criptocracia como no seguro y por lo tanto no válido	Alto	Alto	Alto

		para sostener unas elecciones electrónicas			

Contingencias frente a riesgos

Identificador	Riesgo asociado	Definición
S1	**R1**	Determinar la causa que impide el escalado del sistema y subsanarla para incorporar los nodos que sean necesarios para garantizar el normal funcionamiento del servicio.
S2	**R2**	Determinar por qué el sistema de autenticación falla e implementar una solución software que corrija la vulnerabilidad.
S3	**R3**	Desarrollar y ejecutar mediante un cron job diario un script que verifique el porcentaje de espacio libre en disco y cuando éste sea inferior a un 15% enviar un correo electrónico al administrador del sistema para solicitar más espacio en disco a la proveedora de servicios.
S4	**R4**	Determinar qué componente no está debidamente documentado y proceder a su documentación siguiendo el formato que se ha empleado para el resto del proyecto.
S5	**R5**	Es necesario contar con una copia de seguridad de las copias de seguridad y que ésta esté almacenada en un soporte físico diferente a las propias copias de seguridad.
S6	**R6**	Detectar el fallo y corregir. En caso de que sea un problema de hardware contactar con el suministrador de hardware para reemplazar el componente dañado.
S7	**R7**	Contactar con el proveedor de servicios para renovar el certificado digital.
S8	**R8**	Se asignan tareas al nuevo desarrollador en función de su perfil y su grado de responsabilidad.
S9	**R9**	Hay que implementar nuevas variantes del algoritmo que se sigan considerando seguras o en su defecto hay que buscar un algoritmo seguro de la misma clase y volver a cifrar la información con él o en el caso de que no se pueda reconstruir la información original

		generar nuevo cifrado a partir de la información original proporcionada por los usuarios.
S10	**R10**	Estudiar y adaptar el proyecto de acuerdo a las directrices indicadas en los nuevos reglamentos publicados.
S11	**R11**	Implementar bloqueos a los hosts atacantes. Escalar el sistema hasta un nivel donde los ataques no sean efectivos. Emplear medidas de contingencia como un CDN.
S12	**R12**	Detectar los fallos de visualización y refactorizar código para adaptarlo a los nuevos navegadores sin perder la retrocompatibilidad.
S13	**R13**	Los proveedores de servicio suelen avisar con antelación de las interrupciones. Implementar despliegues sobre soportes alternativos o notificar de la interrupción con antelación a los clientes en caso de que sea imposible llevar a cabo dichos despliegues.
S14	**R14**	Es responsabilidad de los clientes mantener su equipo a salvo de malware e intrusos. Si detectamos una actividad sospechosa en un cliente podemos bloquear su acceso al sistema.
S15	**R15**	El administrador del sistema puede acceder a la base de datos y cambiar las credenciales de acceso del administrador.
S16	**R16**	No podemos hacer otra cosa salvo informar al cliente de que su módulo criptográfico está obsoleto y debe renovarlo.
S17	**R17**	Se implementa cifrado en disco sobre el propio servidor a fin de evitar robo de información si se produce una sustracción del soporte de almacenamiento físico. La clave de cifrado queda residente en memoria, si el intruso consigue congelar la memoria RAM podría acceder a la información descifrada en disco. Para ello contamos con el cifrado de la información almacenada en la base de datos.
S18	**R18**	Es responsabilidad del cliente no manipular su justificante de voto con el fin de contrastar su integridad frente al voto almacenado enla base de datos.
S19	**R19**	Se puede verificar la integridad de los ficheros del proyecto mediante su resumen hash durante una auditoría de código y tras las elecciones se puede repetir dicho proceso para verificar que no se ha manipulado su código en ningún momento.

7.2 Explotación económica

Criptocracia, al ser un software distribuido mediante código abierto bajo licencia copyleft, no permite una explotación económica mediante el uso de licencias de usuario ya que el acceso al código fuente del proyecto permitiría realizar modificaciones para anular cualquier limitación que se haya definido en el mismo mediante técnicas de ingeniería inversa. Por ello, se descarta completamente la obligación de un pago en ese sentido. Sin embargo, como la gran mayoría de software libre, existe la posibilidad de explotar económicamente el proyecto por otras vías. En particular, en el caso de Criptocracia, se proponen varios mecanismos para obtener un rendimiento económico:

- **Instalación y despliegue:** en muchos casos nos vamos a encontrar con entidades que demandan el uso de Criptocracia en sus entornos digitales privados y que no cuentan con los recursos humanos apropiados para realizar una instalación y despliegue con la diligencia adecuada. En estos escenarios se presenta la oportunidad de facturar este trabajo. Además, esta posibilidad no se limita exclusivamente al equipo desarrollador sino a cualquier persona interesada en certificarse en Criptocracia, lo que abre una nueva posibilidad de explotación económica. Esta vía permite también obtener rendimiento económico a cualquier persona interesada y capacitada para llevar a cabo los requerimientos del cliente. No es un beneficio necesariamente dirigido de forma exclusiva al equipo desarrollador.
- **Formación y certificación:** al igual que otras empresas, podemos realizar una formación específica en personas interesadas sobre el proyecto Criptocracia, no sólo para su instalación y despliegue, sino para su formación frente a cambios y extensiones que se quieran aplicar sobre el proyecto original con el objetivo de implementar una versión adaptada y funcional a requerimientos específicos de un cliente. Esta modalidad permite obtener rendimiento económico tanto al equipo desarrollador como a cualquier entidad que se quiera autorizar para certificar a terceros, por lo que se puede obtener un beneficio mutuo.
- **Desarrollo a medida:** en consonancia con el apartado anterior, también podemos valorar casos de clientes específicos como organismos públicos que precisen de una versión de Criptocracia a medida y que el propio equipo desarrollador de Criptocracia pueda prestarse a implementar. Hay que tener en cuenta que

Criptocracia sólo se distribuye de manera gratuita a entidades públicas por lo que, si una entidad privada está interesada en una versión del proyecto personalizada, deberá facturarse como un software a medida y en ningún caso hacerlo bajo la nomenclatura Criptocracia, variando la licencia de uso en la medida que el desarrollador considere. En esta modalidad de explotación comercial se necesita el permiso expreso del autor de Criptocracia para poder cambiar los términos de uso, por lo que sólo podrá ser explotada por el propio autor o un tercero sobre el que haya delegado tal autoridad.

8. Conclusiones

8.1 Conclusiones generales

Criptocracia fue un proyecto desarrollado inicialmente para ofrecer una solución software a un entorno de votaciones distribuido. Como hemos visto, este escenario plantea un problema que a día de hoy sigue sin estar 100% resuelto. Estudiado el paradigma actual de la criptografía encontramos que a nivel teórico hay soluciones que solucionarían el problema perfectamente, tal y como se ha referenciado al cifrado homomórfico. Sin embargo, el objetivo de este trabajo de fin de máster no era realizar una aportación teórica sino proponer una solución viable a nivel práctico y tangible, empleando la tecnología actual y por lo tanto solucionando el problema.

La principal ventaja que plantea Criptocracia frente a otros sistemas de votación electrónica son sus robustas consideraciones desde una perspectiva legal, así como los mecanismos de autenticación mediante módulos criptográficos. Otras propuestas admiten una autenticación de usuarios basada en “algo conocido” lo que resta posibilidades para ser considerado en un entorno que abarque unas elecciones legales. Además, la posibilidad de adquirir el proyecto de forma totalmente gratuita para su estudio y valoración aumenta enormemente las posibilidades de difusión que ofrece ya que emplea mecanismos de instalación asistida y una sencilla guía para facilitar su despliegue en entornos privados, por lo que no hará falta grandes conocimientos para contar con un entorno de Criptocracia funcional.

El sistema ha sido implementado a través de tecnologías gratuitas. Para trabajar con PHP no es necesario pagar ninguna licencia privada, lo mismo que un servidor web apache o un servidor de bases de datos relacionales basado en MariaDB. El objetivo es mantener su carácter gratuito tanto para su despliegue como para su modificación con el fin de desarrollar nuevas mejoras, o simplemente una versión personalizada. En ese sentido el proyecto ofrece un abanico de posibilidades que queda abierto a la originalidad de cualquier desarrollador o a las necesidades particulares de cada entorno donde se quiera desplegar Criptocracia.

Otra de las premisas sobre las que me he basado a la hora de desarrollar este proyecto era la seguridad desde el punto de vista de las nuevas tecnologías. El principal hándicap a la hora de vender y presentar un proyecto de software de esta naturaleza es la

capacidad que tiene el sistema para poder detectar fraude. Para ello, se ha llevado a cabo un análisis desde una perspectiva legal contando con el asesoramiento profesional de un despacho de abogados para ayudarme a desarrollar un escenario de votaciones lo más aproximado a los requerimientos legales en España. Mediante los conocimientos adquiridos en diversas asignaturas del máster se ha desarrollado una propuesta empleando algoritmos criptográficos de distinta naturaleza y justificando su aplicación. Lo que viene a demostrar que con la tecnología actual desarrollar un sistema de esta naturaleza no sólo es perfectamente viable, sino que dicha propuesta ha quedado justificada tanto desde una perspectiva teórica como práctica a un nivel de detalle tan refinado que ha permitido que sea publicada de forma gratuita incluyendo un plan de explotación con distintas vías para obtener rendimiento económico y un plan de estudio para conocer el coste de su desarrollo (suponiendo que hubiese sido un trabajo remunerado) que bien nos puede orientar de cara a realizar una extensión o una adaptación del proyecto a unos requisitos específicos para un escenario electoral concreto. Explicados los procedimientos de seguridad implementados queda patente cómo se solucionan todos los escenarios donde pueda ser comprometida. En última instancia, la integridad de Criptocracia, como cualquier software, puede ser comprometida mediante técnicas de ingeniería inversa aplicadas sobre el propio código fuente del programa. Su naturaleza open source nos permite auditar el código antes y después de un proceso electoral para validar la inmutabilidad de todo el código fuente de la instancia del proyecto sobre la que se ha ejecutado el proceso y que dicha auditoría pueda ser realizada libremente por los interesados ya que cualquier modificación debe respetar la licencia copyleft.

En definitiva, si bien es cierto que la aplicación ha tenido un tiempo de desarrollo de varios meses y que ha tenido que ser debidamente documentada a través de esta memoria, se puede considerar que es un proyecto aún al comienzo de su ciclo de vida ya que tras la publicación y evaluación de este trabajo su código fuente será liberado y puesto a disposición de cualquier persona interesada en consultarlo y utilizarlo. Criptocracia es un proyecto que aún puede dar mucho de sí y que ha sido probado en la medida que han alcanzado las pruebas de software durante el desarrollo del mismo, sin embargo, ha tenido un relevante éxito en aquellos escenarios donde se ha implantado a modo de testeo, siendo uno de ellos valorado por políticos con representación parlamentaria y con una notable aceptación en los test de validación realizados para valorar la usabilidad de la aplicación, lo que confirma que un diseño

orientado al usuario es especialmente útil en aquellos proyectos donde la facilidad y la intuición para trabajar con una interfaz gráfica es crítica.

8.2 Posibles ampliaciones futuras

Como se ha dicho anteriormente, Criptocracia es un proyecto que apenas ha comenzado y existe un enorme recorrido por delante. A corto plazo existen numerosas posibilidades para seguir ampliando y convertirlo en un proyecto aún más atractivo.

- **Compatibilidad con más módulos criptográficos:** con el fin de aumentar la compatibilidad con otros sistemas de autenticación para votantes, es posible implementar mecanismos para que Criptocracia reconozca otros certificados digitales, bien sean instalados en el dispositivo desde el que se accede o a través de soportes extraíbles. En este caso tan sólo hay que definir 2 componentes: La clave pública del certificado (.cer/.crt) para validar la clave privada de los certificados digitales proporcionados y las cabeceras HTTP que envía para poder extraer la información del certificado. Con estos dos componentes implementados sobre la instancia de Criptocracia tan sólo hay que configurar adecuadamente el servidor web para que exija a la entidad certificadora validar el certificado proporcionado por el cliente a fin de asegurar que es un certificado válido y así suprimir cualquier intento de autenticación fraudulento con el módulo criptográfico en cuestión.
- **Nuevos idiomas:** parte de la usabilidad y la accesibilidad es la internacionalización de las aplicaciones. Nuevamente, aplicando los principios de alta cohesión y bajo acoplamiento las definiciones de cada idioma van recogidas en ficheros .php aparte y para incorporar nuevos idiomas tan sólo hay que coger cualquiera de los ficheros ya existentes y realizar una traducción manual o con algún software que automatice el proceso al idioma deseado. Eso adjunto a una imagen con el tamaño correspondiente identificativa del idioma deseado es todo lo que se necesita para implementar nuevos idiomas para Criptocracia.
- **Escrutinios por grupos:** con el fin de poder obtener un escrutinio clasificado por grupos para, por ejemplo, elecciones generales agrupadas por las diferentes autonomías (y así poder obtener el respectivo reparto de escaños en función del censo de cada comunidad) Criptocracia incorpora de serie los grupos de votantes, que nos permiten indicar qué grupos de DNIs (votantes) podrán participar en una determinada elección. Una sencilla refactorización en la estructura del voto cifrado

añadiendo un campo para reconocer el grupo al que pertenece el voto nos permitirá tener implementada esta característica.

- **Aplicación nativa para dispositivos iOS/Android:** si bien es cierto que Criptocracia ya incorpora una interfaz responsive, en algunos casos los usuarios pueden encontrar más cómodo el uso de Criptocracia si cuentan con una aplicación nativa. Dicha aplicación nativa puede ser un sencillo contenedor web o bien puede ser una aplicación desarrollada a partir de los propios SDK y componentes de cada sistema operativo. Esta característica debe someterse previamente a un estudio de viabilidad tanto económica como temporal ya que puede representar un proyecto totalmente nuevo al tener que incorporar una API REST en el propio núcleo de Criptocracia. No obstante, la posibilidad está ahí.
- **API REST:** en consonancia con esto último, tal vez pueda ser interesante implementar una API REST que pueda ser consumida bien para tareas de administración o bien para votantes proporcionando las credenciales de autenticación correspondientes en cada caso. Esto puede abrir un abanico de posibilidades ya que facilita a los desarrolladores externos la implementación de una aplicación nativa o incluso automatizar los procesos de votación a través de otras plataformas externas o dispositivos.
- **Compatibilidad para votaciones en el Senado:** la actual implementación de Criptocracia permite la votación de una sola candidatura por separado, pero existen escenarios concretos donde el votante puede emitir sufragio sobre varias candidaturas con un máximo de votos definido. En el caso particular del Senado en España un voto puede dirigirse hasta a 3 candidatos, pero no más de 3, computándose una papeleta con 0 votos como un voto en blanco. Otra posible ampliación para el proyecto podría ser el desarrollo de votaciones que permitan votar a más de una candidatura, pero tan sólo una vez sobre cada una de ellas (o no) lo que daría una mayor flexibilidad a la hora de realizar unas votaciones basadas en “puntos” para otro tipo de elecciones diferentes.

9 Referencias

1- Statista. Estadísticas de dispositivos por usuario, 2017, https://es.statista.com/estadisticas/481066/numero-de-dispositivos-conectados-por-persona-en-espana/

2- Congreso.es, Sinopsis artículo 68 Constitución, http://www.congreso.es/consti/constitucion/indice/sinopsis/sinopsis.jsp?art=68&tipo=2

3- Wikipedia, Cifrado homomórfico, https://es.wikipedia.org/wiki/Cifrado_homom%C3%B3rfico

4- Wikipedia, Historia de la democracia, https://es.wikipedia.org/wiki/Historia_de_la_democracia

5- Antena 3 Noticias, Sistema de consulta de Podemos, https://www.antena3.com/noticias/espana/polemica-sistema-votacion-consulta-podemos-sus-bases-compra-chalet-iglesias-montero_201805235b0567be0cf25f5b27b19a36.html

6- Javier Barrera, Vot.Ar: Una mala elección, Julio 2015, http://ivan.barreraoro.com.ar/vot-ar-una-mala-eleccion/

7- PHP: Hypertext Preprocessor, 1994, http://www.php.net/

8- World Wide Web Consortium (W3C), 1994, https://www.w3.org/

9- Portal del DNI Electrónico, CNP, https://www.dnielectronico.es/PortalDNIe/

10- Portal de certificados digitales, Fábrica Nacional de Moneda y Timbre, https://www.sede.fnmt.gob.es/certificados/persona-fisica

11- Google Fonts, Google, https://fonts.google.com/

12- UI Design patterns, Anders Toxboe, 2019, http://ui-patterns.com/patterns

13- Extreme Programing: A Gentle introduction, Don Wells, 2013, http://www.extremeprogramming.org/

14- Modelo vista controlador, Universidad de Alicante, 2019, https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html

15- Modelo Kanban, Wikipedia, 2019, https://es.wikipedia.org/wiki/Kanban_(desarrollo)

16- Copyleft, GNU, 2018, https://www.gnu.org/licenses/copyleft.es.html

17- Linux Kernel, Wikipedia, 2019, https://en.wikipedia.org/wiki/Linux_kernel