



Universidad de Valladolid

Análisis de Privacidad en una App de Rastreo

Juan Velázquez García

Tutores

Julián Arroyo Álvarez
Mercedes Martínez González

Agradecimientos

En primer lugar, me gustaría agradecer a mis tutores Julián Arroyo Álvarez y Mercedes Martínez González por su esfuerzo en aconsejar y corregir este trabajo. También me gustaría darle las gracias al profesor Amador Aparicio de la Fuente por sus consejos e interés en el desarrollo de este proyecto.

En segundo lugar, me gustaría dar las gracias a mis padres Juan Luis y Henar, a mi hermano Enrique, a mis abuelas Eulalia y Petra, a mis tíos Ceci y Azucena y a toda mi familia por creer en mí y darme la oportunidad de estudiar lo que quisiera.

También quería agradecer a mis abuelos Ángel y Eleuterio, allá donde estén, su esfuerzo por darles lo mejor a mis padres y tíos. Quién les iba a decir a un humilde pastor y a un honrado panadero de pueblo que su nieto estudiaría en la universidad.

Por último quería dar las gracias a mis amigos de la universidad Víctor, Christopher, Héctor, Pedro, Susana y a mi compañera de proyecto María por las risas, los llantos, las locuras, los agobios... En definitiva por hacer más feliz mi estancia en la universidad. Jamás lo olvidaré.

Resumen

Este proyecto aborda los riesgos de seguridad y privacidad en aplicaciones móviles. El trabajo aquí plasmado consiste en una propuesta de evaluación de riesgos e impacto en la privacidad y seguridad del usuario usando el estándar propuesto por el CCN-CERT (España). Para ello se ha desarrollado, junto a María Ruiz Molina, una aplicación prototípico de rastreo de contactos [COVID](#), sobre la cual se aplica dicha evaluación.

Abstract

This project aims to study some privacy and security standards, specifically, the proposals from the administrative authority, Centro Criptológico Nacional Computer Emergency Response Team (CCN-CERT).

Likewise, the current state of affairs regarding mobile contact-tracing apps' privacy and security risks will be reviewed. A risk evaluation will be proposed as a conclusion from the aforementioned standards.

Therefore, a prototyped app based on the current Covid contact-tracing applications will be developed, together with María Ruiz Molina, and the previously-established proposal will be applied to it.

Finally, a series of conclusions from this study will be presented in further detail.

Índice

1. Introducción	12
2. Objetivos	13
3. Planificación y Metodología	14
4. Análisis de la aplicación	18
4.1. Requisitos no funcionales	19
4.1.1. Requisitos de seguridad	19
4.1.2. Requisitos de privacidad	20
4.1.3. Requisitos que se aplican únicamente cuando la aplicación envía al servidor una lista de contactos	23
4.1.4. Requisitos que se aplican únicamente cuando la aplicación envía al servidor una lista de sus propios identificadores	24
4.1.5. Requisitos de Usabilidad	25
4.2. Requisitos Funcionales	25
4.2.1. De cara al usuario	25
4.2.2. De cara a otros dispositivos	26
4.2.3. De cara a la propia aplicación	26
4.3. Requisitos de Información	26
4.4. Base de Datos	27
4.5. Modelo de intercambio de datos	30
4.6. Desarrollo de los casos de uso	30
4.6.1. Enviar diagnóstico	32
4.6.2. Cambiar idioma	32
4.6.3. Comunicar semillas asociadas a IDs infectados	33
4.6.4. Comprobar riesgo de contagio	33
4.6.5. Enviar diagnóstico falso	34
4.6.6. Recibir ID	34
4.6.7. Enviar ID	35
5. Diseño de la aplicación	36
5.1. Estado del Arte: Protocolos	36
5.1.1. Decentralized Privacy-Preserving Proximity Tracing (DP-3T)	36
5.1.2. (Google/Apple) Exposure Notification (GAEN) system	37
5.1.3. Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT/PEPP)	38

5.1.4. BlueTrace	39
5.1.5. Otros	39
5.2. Elección del protocolo	40
5.3. Imprevistos respecto al protocolo elegido y consecuencias	41
5.4. Implementación de los requisitos acorde a nuestra versión de DP-3T	45
5.5. Implementación de los Requisitos de Usabilidad	51
5.6. Diseño de la Interfaz e Implementación de los Requisitos	52
5.6.1. Pantalla Principal	52
5.6.2. Pantalla de Información	55
5.6.3. Pantalla de Ajustes de Idioma	55
5.7. Implementación de la aplicación	57
5.7.1. Implementación final de la interfaz	57
5.7.2. Conexiones de red TCP	59
5.7.3. Conexiones de red UDP	59
5.7.4. Decisión sobre Bluetooth	59
5.7.5. Cifrado de los datos	60
5.7.6. Adaptaciones realizadas cara al prototipo	61
6. Casos de prueba	63
6.1. Pruebas de caja negra	63
6.1.1. Intercambiar un ID entre dos dispositivos vía BT en rango	63
6.1.2. Intercambiar un ID entre dos dispositivos vía BT en el límite del rango	65
6.1.3. Intercambiar un ID entre dos dispositivos vía BT fuera de rango	65
6.1.4. Intercambiar un ID entre dos dispositivos vía BT y desconectarse justo en el momento del envío	65
6.1.5. Intercambiar un ID entre dos dispositivos vía BT y desconectarse justo 1 segundo antes del momento del envío	66
6.1.6. Intercambiar un ID entre dos dispositivos vía BT y desconectarse justo 1 segundo después del momento del envío	66
6.1.7. Uso de la aplicación sin conexión a BT	67
6.1.8. Envío de un código correcto al servidor. (Código correcto: el pedido por la aplicación, otorgado por la autoridad sanitaria.)	68
6.1.9. Envío de un código incorrecto al servidor	70
6.1.10. Envío de diagnóstico sin fecha	70
6.1.11. Envío de diagnóstico sin inserción de código	71
6.1.12. Envío de código con menos de 12 cifras	72
6.1.13. Envío de código con más de 12 cifras	73
6.1.14. Envío de código con caracteres no numéricos	73

6.1.15. Envío con fecha anterior a 14 días	73
6.1.16. Envío con fecha posterior a 14 días	74
6.1.17. Multicast de IDs infectados desde el servidor a los clientes	75
6.1.18. Uso de la aplicación sin conexión a Internet (y sin recepción de multicast)	76
6.1.19. Recepción por multicast de un ID infectado que se encuentra en la base de datos local	78
6.1.20. Recepción por multicast de IDs infectados y ninguno se encuentra en la base de datos local	80
6.1.21. Recepción por multicast de IDs infectados y ninguno se encuentra en la base de datos local, pero el ID es un número por encima de uno almacenado	80
6.1.22. Recepción por multicast de IDs infectados y ninguno se encuentra en la base de datos local, pero el ID es un número por debajo de uno almacenado	81
6.1.23. Recepción por multicast de IDs infectados y no se posee ningún ID en la base de datos local con los que comparar	81
6.1.24. Envío del multicast pero sin recepción (clientes inactivos)	82
6.1.25. Escucha, por parte del cliente, de multicast pero sin envío (servidor inactivo)	82
6.1.26. Cambio de idioma	83
6.2. Pruebas de caja blanca	84
6.2.1. Escucha del canal de conexión en el momento de envío de un código con los IDs al servidor	84
6.2.2. Escucha del canal de conexión en el momento de envío de un código incorrecto al servidor	85
6.2.3. Escucha del canal de conexión en el momento del envío del multicast	85
6.2.4. Barrido de puertos de la máquina cliente	86
6.2.5. Barrido de puertos de la máquina servidor	87
6.2.6. Ataque DDoS	89
6.2.7. Inyección de código desde la aplicación cliente	91
7. Análisis de riesgos de seguridad y privacidad	92
7.1. Estado del arte: Seguridad y Privacidad	92
7.1.1. Fase 1: Definición del alcance	92
7.1.2. Fase 2: Identificación de activos	92
7.1.3. Fase 3: Identificación y selección de amenazas	92
7.1.4. Fase 4: Identificación de vulnerabilidades y salvaguardas	92
7.1.5. Fase 5: Evaluación del riesgo	93
7.1.6. Fase 6: Tratamiento del riesgo	93
7.1.7. CCN-CERT	94
7.1.8. PILAR	94
7.1.9. Metodología MAGERIT	95
7.1.10. Esquema Nacional de Seguridad	95

7.1.11. Amenazas en la actualidad	97
7.2. Identificación de Activos	98
7.3. Valoración de los activos	100
7.4. Amenazas	107
7.5. Riesgos de privacidad	123
7.6. Salvaguardas	125
8. Conclusiones	127
Bibliografía	143

Índice de figuras

1.	Planificación inicial	14
2.	Planificación final	16
3.	Modelo conceptual	27
4.	Modelo lógico Cliente	28
5.	Modelo lógico Servidor	29
6.	Modelo de intercambio de datos	30
7.	Diagrama de casos de uso	31
8.	Esquema de generación de identificadores efímeros	36
9.	Solicitud a cumplimentar para tener acceso a la API	41
10.	Documentación requerida	41
11.	Modelo físico de la base de datos de los clientes	49
12.	Modelo físico de la base de datos del servidor	50
13.	Menú de navegación	52
14.	Pantalla Principal	53
15.	Botón Comunica tu contagio	53
16.	Confirmación del envío	54
17.	Simulación de los distintos tipos de daltonismo	54
18.	Pantalla de información	55
19.	Pantalla de idiomas	55
20.	Pantalla de inicio. Colores de la aplicación	57
21.	Pantalla de inicio. Protanopia	58
22.	Pantalla de inicio. Deuteranopia	58
23.	Pantalla de inicio. Tritanopia	58
24.	Resultado de entropías	60
25.	Intercambio de ID entre dispositivos	64
26.	Solicitud para activar Bluetooth	67
27.	Aplicación con contagio	69
28.	Aviso sobre código incompleto	71
29.	Aviso sobre código incompleto	72
30.	Límite anterior de aceptación de fechas	73
31.	Límite posterior de aceptación de fechas	74
32.	Aviso de desconexión	76
33.	Aplicación ejecutándose correctamente	77
34.	Aplicación ejecutándose correctamente	79

35.	Aplicación en inglés	83
36.	Resultado del programa de sniffing	84
37.	Resultado del sniffing	85
38.	Puerto 4446. Recepción de multicast. UDP	86
39.	Puerto 4445. Envío de multicast. UDP.	87
40.	Puerto 3327. Acceso a base de datos. TCP.	88
41.	Puerto 3384. Envío de códigos. TCP.	88
42.	Resultado de ataque de denegación de servicio.	90
43.	Ejemplo de matriz de riesgos	93
44.	Tabla de valoración de activos	102
45.	Listado de amenazas de PILAR Basic (Parte 1)	108
46.	Listado de amenazas de PILAR Basic (Parte 2)	109
47.	Listado de amenazas de PILAR Basic (Parte 3)	110
48.	Listado de amenazas de PILAR Basic (Parte 4)	111
49.	Listado de amenazas de PILAR Basic (Parte 5)	112
50.	Listado de amenazas de PILAR Basic (Parte 6)	113
51.	Build APK	132

Índice de tablas

1.	Caso de uso: Enviar diagnóstico	32
2.	Caso de uso: Cambiar idioma	32
3.	Caso de uso: Comunicar semillas asociadas a IDs infectados	33
4.	Caso de uso: Comprobar riesgo de contagio	33
5.	Caso de uso: Enviar diagnóstico falso	34
6.	Caso de uso: Recibir ID	34
7.	Caso de uso: Enviar ID	35
8.	Tabla de amenazas y riesgos (Parte 1)	114
9.	Tabla de amenazas y riesgos (Parte 2)	115
10.	Tabla de amenazas y riesgos (Parte 3)	116
11.	Tabla de amenazas y riesgos (Parte 4)	117
12.	Tabla de amenazas y riesgos (Parte 5)	118
13.	Tabla de amenazas y riesgos (Parte 6)	118
14.	Tabla de amenazas y riesgos (Parte 7)	119
15.	Tabla de amenazas y riesgos (Parte 8)	120
16.	Tabla de amenazas y riesgos (Parte 9)	121
17.	Tabla de amenazas y riesgos (Parte 10)	122

1. Introducción

Debido a la transformación tecnológica tan acelerada que estamos viviendo, el crecimiento de datos y la cantidad de información generada son abrumadores. En 2020 se generaban diariamente más de un billón de megabytes de datos, y la cantidad de datos almacenados se veía duplicada cada 10 meses. El flujo de todos ellos, debido al gran negocio existente en torno a la compra-venta de datos personales, hace que cada usuario aparezca en torno a entre 800 y 1000 bases de datos. Esto supone un importante riesgo para la privacidad de las personas de las que proceden estos datos, pues llega un momento en el que desconocen quiénes poseen su información. [30]

Muchos de estos datos se generan prácticamente sin darnos cuenta, como con el uso habitual de dispositivos móviles y personales para acceder a servicios. El supuesto presentado es una aplicación que trata con datos de carácter sensible, tales que los relacionados con la salud de las personas, y por ello merecen de un especial cuidado. Es por ello que este tipo de servicios deben respetar y clarificar a sus usuarios qué datos son recogidos y con qué fines van a ser tratados, prestando gran atención a su privacidad tal y como especifican las autoridades reguladoras de la protección de datos.

La aplicación que se ha desarrollado pues, se plantea desde dicho punto de vista, cumpliendo con una serie de estrictos requisitos de privacidad impuestos por el European Data Protection Board, así como otras entidades y aquellos que se han considerado oportunos para el correcto funcionamiento. Posteriormente para verificar el correcto cumplimiento con respecto a la normativa ENS, se realizará un análisis de seguridad y privacidad de la misma empleando la herramienta PILAR, la cual implementa tales normativas.

El ENS establece directrices de seguridad a nivel técnico e incluye una evaluación de riesgos, siendo que algunas de estas reglas entran en estrecha relación con la protección de la privacidad.

De la mano de esto se habrán realizado previamente los correspondientes casos de prueba para detectar posibles brechas de datos que pudieran afectar a la privacidad de los usuarios.

La motivación de este proyecto se debe a la escasa profundización y concienciación sobre este tema, que aun se encuentra en una fase incipiente, pero que afecta a una enorme cantidad de la población. Además, el hecho de tratarse de una aplicación de rastreo de contactos hace del proyecto de especial interés debido a la situación vivida debido a la pandemia del COVID-19, donde el uso de aplicaciones de rastreo ha sido algo especialmente incentivado por las autoridades estatales y sanitarias.

2. Objetivos

El objetivo general de este proyecto es realizar una propuesta de evaluación de privacidad y seguridad en aplicaciones de rastreo de contactos y aplicarla a un caso real.

Con el fin de lograr dicho objetivo y manteniendo siempre como puntos principales la seguridad y la privacidad, se llevarán a cabo los siguientes subobjetivos:

- **Objetivo 1.** Realizar el análisis, diseño y desarrollo de una aplicación prototipo de rastreo de contactos, y sus respectivos casos de prueba. Se empleará la técnica de la programación en pareja alternando roles.
- **Objetivo 2.** Conocer el Estado del Arte de los protocolos de rastreo de contactos, de la evaluación de riesgos de privacidad y seguridad y determinar qué protocolos y directrices tomar como referencia a tener en cuenta durante el desarrollo.
- **Objetivo 3.** Elaborar y aplicar una propuesta de análisis de riesgos de seguridad y privacidad a partir de lo presentado por las soluciones del CCN-CERT, CNIL y de lo visto en el estudio del Estado del Arte.
- **Objetivo 4.** Obtener una serie de conclusiones a partir del estudio realizado para determinar los puntos fuertes y débiles de la aplicación en cuestiones de seguridad y privacidad.

Finalmente, de las conclusiones obtenidas del estudio, se espera que las posibles brechas de privacidad y/o seguridad detectadas, se tengan en cuenta en futuros proyectos de índole similar.

3. Planificación y Metodología

Debido a que el desarrollo de la aplicación es llevado a cabo por dos alumnos, María Ruiz Molina y Juan Velázquez García, se ha optado por una metodología de desarrollo Extreme Programming [33], en concreto a la hora de la implementación de la aplicación, Pair Programming o programación en pareja.

Dicha metodología se basa en los siguientes puntos:

- **Constante comunicación cara a cara y feedback.** Si bien debido a que el desarrollo de la aplicación se ha realizado durante la pandemia del COVID-19, se ha hecho uso de aplicaciones de videoconferencias con el fin de realizar una comunicación lo más cercana posible. A los tutores se les ha ido informando de manera incremental de los cambios y de las nuevas implementaciones según se iban realizando.
- **Simplicidad.** A la hora de tomar la decisión que concierne al diseño de la aplicación, se optará por aquella más sencilla y que implemente los requisitos previamente elicitedados.
- **Responsabilidad.** La calidad del software recae en los propios desarrolladores. En este marco se incluye el desarrollo de casos de prueba así como del posterior análisis de seguridad y privacidad.
- **Coraje.** Capacidad para desechar trabajo ya realizado, si así se requiere, con el fin de investigar otras ideas que puedan ser más apropiadas, se adapten mejor a lo solicitado o solucionen algún posible imprevisto.

En concreto, la programación en pareja se basa en que el código sea escrito por parejas, una en este caso, de desarrolladores.

Mientras uno escribe el código como tal, el otro observa, discute las ideas y hace comentarios o sugerencias mientras investiga sobre ello. Dichos roles se han de ir intercambiando con el fin de que el trabajo dedicado a cada papel sea igual para ambos miembros. [33]

Con esta metodología en mente, se lleva a cabo la siguiente planificación:

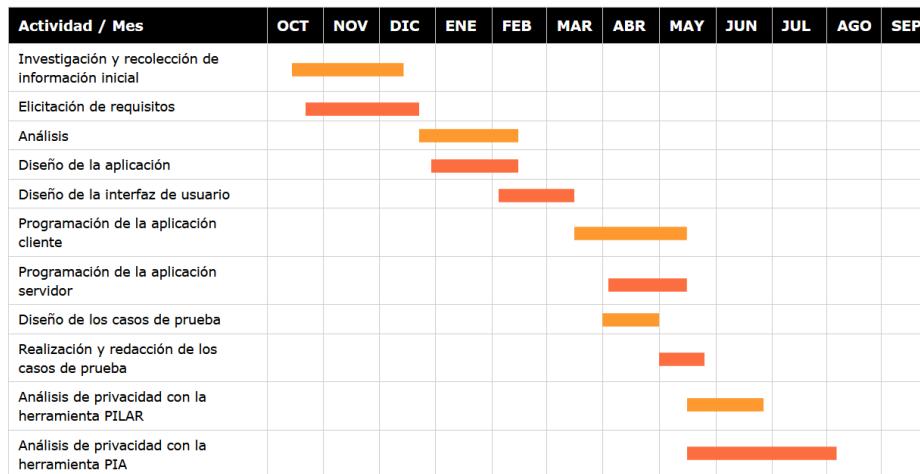


Figura 1: Planificación inicial

Como puede verse, el proyecto se organizó y distribuyó en distintas fases a lo largo de un total de 44 semanas, entre mediados de octubre, momento de inicio, hasta mediados de agosto. Al ser un trabajo con una parte en común y otra individual, los análisis de seguridad y privacidad, se reflejan ambos en el diagrama de Gantt.

Juan Velázquez García termina el proyecto y su parte el mismo día que la actividad de *Análisis con la herramienta PILAR* acaba.

María Ruiz Molina termina el proyecto y su parte el mismo día que la actividad de *Análisis con la herramienta PIA* acaba.

- **Investigación y recolección de Información Inicial.** Desde la semana 1 a la semana 8. Se subdivide en:
 - Investigación de protocolos de rastreo de contactos.
 - Investigación de la aplicación Radar Covid.
 - Investigación sobre estándares de privacidad y seguridad.
 - Investigación sobre el protocolo DP-3T.
 - Investigación sobre la generación de **identificadores efímeros**.
- **Elicitación de requisitos.** Desde la semana 2 a la semana 10.
- **Análisis.** Desde la semana 10 a la semana 15. Se subdivide en:
 - Caso de uso.
 - Actores.
 - Diagrama de casos de uso.
 - Modelo de intercambio datos.
- **Diseño de la aplicación.** Desde la semana 10 a la semana 16. Se subdivide en:
 - Diseño del protocolo.
 - Diseño de la base de datos del cliente.
 - Diseño de la base de datos del servidor.
- **Diseño de la interfaz de usuario.** También se considera su programación. Desde la semana 15 a la semana 20.
- **Programación de la aplicación cliente.** Desde la semana 20 a la semana 28.
- **Programación de la aplicación servidor.** Desde la semana 24 a la semana 28.
- **Diseño de los casos de prueba.** Desde la semana 24 a la semana 28.
- **Realización y redacción de los casos de prueba.** Desde la semana 28 a la semana 31.
- **Análisis con la herramienta PILAR.** Desde la semana 30 a la semana 35, teniendo un margen de 3 semanas más hasta el momento de cierre de actas.
- **Análisis con la herramienta PIA.** Desde la semana 30 a la semana 42, teniendo un margen de 3 semanas más hasta el momento de cierre de actas.

Debido a un imprevisto a la hora de hacer uso del protocolo de rastreo de contactos, hubo que ajustar los tiempos de la planificación.

En concreto, hubo que desarrollar un protocolo desde cero, por lo que los tiempos de programación de la aplicación tuvieron que aumentarse. Esto llevó a un desplazamiento de las actividades posteriores a esta y también a una disminución del tiempo disponible para realizarlas.

El diagrama de Gantt tras esto queda así:

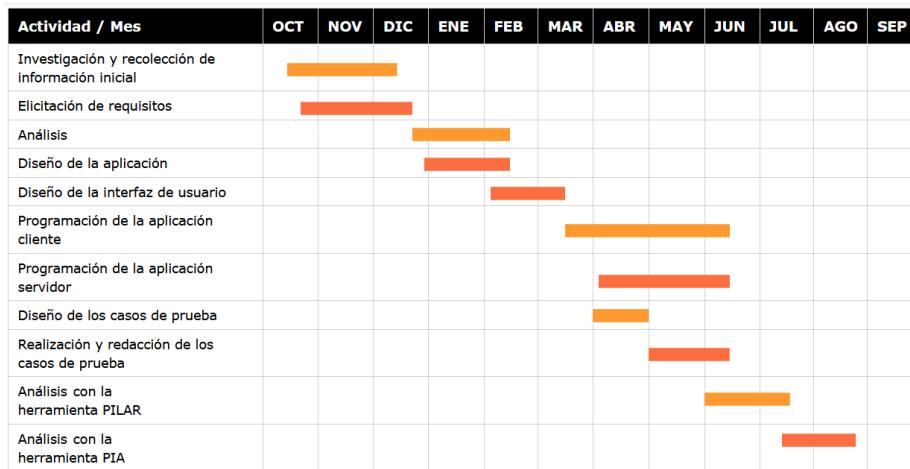


Figura 2: Planificación final

La planificación final queda distribuida de la siguiente manera:

- **Investigación y recolección de Información Inicial.** Desde la semana 1 a la semana 8. Se subdivide en:
 - Investigación de protocolos de rastreo de contactos.
 - Investigación de la aplicación Radar Covid.
 - Investigación sobre estándares de privacidad y seguridad.
 - Investigación sobre el protocolo DP-3T.
 - Investigación sobre la generación de identificadores efímeros.
- **Elicitación de requisitos.** Desde la semana 2 a la semana 10.
- **Análisis.** Desde la semana 10 a la semana 15. Se subdivide en:
 - Caso de uso.
 - Actores.
 - Diagrama de casos de uso.
 - Modelo de intercambio datos.

- **Diseño de la aplicación.** Desde la semana 10 a la semana 16. Se subdivide en:
 - **Diseño del protocolo.**
 - **Diseño de la base de datos del cliente.**
 - **Diseño de la base de datos del servidor.**
- **Diseño de la interfaz de usuario.** También se considera su programación. Desde la semana 15 a la semana 20.
- **Programación de la aplicación cliente.** Desde la semana 20 a la semana 31.
- **Programación de la aplicación servidor.** Desde la semana 24 a la semana 31.
- **Diseño de los casos de prueba.** Desde la semana 24 a la semana 28.
- **Realización y redacción de los casos de prueba.** Desde la semana 31 a la semana 34.
- **Análisis con la herramienta PILAR.** Desde la semana 33 a la semana 38.
- **Análisis con la herramienta PIA.** Desde la semana 37 a la semana 44.

4. Análisis de la aplicación

En este trabajo se ha llevado a cabo el desarrollo de una aplicación de rastreo de contactos, la cual interacciona con:

- **Usuarios.** Los cuales instalan una aplicación cliente en sus dispositivos móviles.
- **Autoridad Sanitaria.** La cual gestiona un servidor. Dado que aquí se ha desarrollado un prototipo, se simulan las funcionalidades básicas de un caso real.

Para comenzar el proyecto, primeramente se realizó una búsqueda de información sobre aplicaciones de rastreo de contactos existentes en el mercado, como son [SwissCovid](#), [COVIDSafe](#), [Smittestopp](#) o [Radar COVID](#). El fin de dicho estudio ha sido comprender las necesidades y funcionamiento primordiales para poder deducir y aplicar los requisitos esenciales. En especial se toma en consideración aquellos relacionados con la privacidad del usuario debido al carácter médico y sensible de los datos.

Estudios como el realizado por Paul-Olivier Dehay y Joel Reardon [21] y artículos como el escrito por Patrick Howell O'Neil [50], nos marcan que hay países, en este caso Suiza y Noruega, que consideran la privacidad como algo indispensable, hasta el punto de retirar del mercado sus aplicaciones de rastreo de contactos.

Debido al especial hincapié en dicha cuestión, estos se han tomado como guías esenciales para concluir en la fuerte necesidad de otorgar a la privacidad la merecida atención. A raíz de ello, se decidió tomar como referentes a diversos organismos legislativos que dictaminan requisitos de privacidad sobre este tipo de aplicaciones.

En este ámbito encontramos dos enfoques principales para realizar el rastreo de contactos.

- **Intercambio de Identificadores.** Este enfoque se basa en el uso de identificadores efímeros. Los identificadores efímeros son códigos alfanuméricos que identifican a un usuario anónimamente durante un cierto periodo de tiempo, tras el cual se desechan por otros.
- **Geolocalización.** Por otro lado, en este [paradigma](#) se emplean técnicas de geolocalización, tal que de haber un usuario contagiado se alerta a aquellos que hayan estado en las mismas zonas que él a la vez. Este enfoque es mucho más invasivo que el anterior para la privacidad de los usuarios, pues se hace uso de un dato sensible.

Debido a que los diferentes organismos legislativos empleados como referentes a lo largo del proyecto refieren al uso de identificadores, todo el desarrollo se orientará hacia este paradigma en lugar de al de geolocalización.

Los datos de geolocalización pueden revelar mucha información personal del usuario a partir de los lugares que visita, como son por ejemplo su dirección personal o la de su trabajo, pero también datos sobre su religión si visita algún edificio de culto, o incluso sobre su orientación sexual determinado por algunos lugares que pudiera visitar.

EL EDPB (European Data Protection Board) establece que los datos de geolocalización no deberían requerirse excepto si son de absoluta necesidad. En concreto dicta que «El seguimiento sistemático y masivo de la localización o los contactos de las personas físicas es una grave injerencia en su privacidad. Esta práctica solo puede legitimarse sobre la base de su adopción voluntaria por parte de los usuarios para cada uno de los fines respectivos, lo que implica, entre otras cosas, que las personas que decidan no utilizar esas aplicaciones, o no sepan hacerlo, no deben sufrir ninguna desventaja.» ([European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Párrafo 24](#))

De esta investigación hemos obtenido como resultado la siguiente lista de requisitos para la aplicación. Estos se clasifican en funcionales, no funcionales, donde se incluyen los aspectos de privacidad.

4.1. Requisitos no funcionales

1. **Aplicación móvil.** Al ser una aplicación en la que se requiere recopilar con quién se tiene contacto, esta ha de ser portátil y por tanto debe instalarse en un teléfono móvil, ya que es un objeto que llevamos siempre encima.
2. **Rotación de identificadores.** Con el fin de incrementar la confusión y difusión, diariamente se generan un número fijo de identificadores que rotan cada cierto tiempo. [19]
3. **El servidor debe enviar únicamente los identificadores activos.** Se considera identificador activo aquel cuya fecha es menor a la fecha de recepción en el servidor. Esto es con el fin de evitar almacenar identificadores asociados a individuos ya recuperados de la enfermedad.

4.1.1. Requisitos de seguridad

1. **Control de entrada de datos.** Se supervisarán las entradas de datos de la aplicación con el fin de evitar [inyecciones de código](#).
2. «**Un mecanismo debe verificar el estado de los usuarios que notifican en la aplicación su condición de positivos en infección.** [...] Por ejemplo, facilitando un código de un solo uso vinculado con un laboratorio de pruebas o a un profesional de atención sanitaria. Si no se puede obtener confirmación de forma segura, no debe procederse al tratamiento de datos». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-1, 2020) [16]
3. «**Los datos enviados al servidor central han de transmitirse a través de un canal seguro.** El uso de servicios de notificación prestados por proveedores de plataformas de sistema operativo debe evaluarse cuidadosamente y no debe dar lugar a la divulgación de ningún dato a terceros». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-2, 2020)
4. «**Las solicitudes no deben ser vulnerables a la manipulación por parte de un usuario malintencionado.** Para evitar falsos positivos. (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-3, 2020)
5. **Uso de técnicas criptográficas.** «Deben aplicarse las técnicas criptográficas más avanzadas para asegurar los intercambios entre la aplicación y el servidor, y entre aplicaciones, y, como regla general, para proteger la información almacenada en las aplicaciones y en el servidor. Entre las técnicas que pueden utilizarse figuran, por ejemplo, las siguientes: cifrado simétrico y asimétrico, funciones hash, prueba privada de pertenencia (private membership test, PMT), intersección privada de conjuntos adoptadas 19 (private set intersection, PSI), filtros Bloom, recuperación de información privada, cifrado homomórfico, etc.» (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-4, 2020)
6. **El servidor central no debe conservar los identificadores de conexión a la red de ningún usuario.** «El servidor central no debe conservar los identificadores de conexión a la red (p. ej., las [direcciones IP](#)) de ningún usuario, incluidos los que han sido diagnosticados positivamente y que han transmitido su historial de contactos o sus propios identificadores». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-5, 2020)

7. **El servidor debe autenticar la aplicación y viceversa.** «Para evitar la suplantación o la creación de falsos usuarios, el servidor debe autenticar la aplicación». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-6, 2020)
8. **«La aplicación debe autenticar el servidor central».** (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-7, 2020)
9. **«Las funcionalidades del servidor deben estar protegidas frente a ataques de repetición».** Para evitar suplantación de identidad y ataques de denegación de servicio. (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-8, 2020)
10. **«La información transmitida por el servidor central debe estar firmada para autenticar su origen e integridad».** Esto se logra mediante la criptografía asimétrica. (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-9, 2020)
11. **Administración de acceso al servidor.** «El acceso a todos los datos almacenados en el servidor central y que no estén a disposición del público debe circunscribirse a las personas autorizadas». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-10, 2020)
12. **Administración de permisos de la aplicación.** «El gestor de permisos del dispositivo en el nivel del sistema operativo solo debe solicitar los permisos necesarios para acceder a los módulos de comunicación y utilizarlos cuando resulte necesario, para almacenar los datos en el equipo terminal y para intercambiar información con el servidor central». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo SEC-11, 2020)

4.1.2. Requisitos de privacidad

1. **Minimalidad de los datos.** «Los intercambios de datos deben respetar la privacidad de los usuarios (y, en particular, el principio de minimización de datos)». Para respetar la privacidad de los datos, la aplicación se limitará a trabajar con datos que no permitan averiguar la identidad del usuario, tales que los identificadores efímeros o el intercambio de **semillas generadoras** con el servidor para que no viajen los identificadores por la red. (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-1, 2020)

- **Datos que se almacenan.**

- Los **identificadores diarios propios y las semillas** con las que se generen.
- Los **identificadores de los usuarios con los que se haya tenido contacto**.
- **Las semillas de los contactos positivos.** Para generar los identificadores asociados a esta y compararlos con los almacenados.

- Datos referentes a **permisos de la aplicación**.
 - android.permission.INTERNET: permite comunicarse con el backend del servidor.
 - android.permission.ACCESS_NETWORK_STATE: permite a la aplicación saber si el dispositivo está conectado a Internet.
 - android.permission.BLUETOOTH: usado para poder comunicarse entre dispositivos móviles. [36]
 - android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS: permite que la aplicación se ejecute en cualquier momento en segundo plano, permitiendo que las sincronizaciones entre la aplicación y el servidor sucedan en los momentos oportunos.
- **Datos que se envían**
 - **De cliente a servidor de la entidad sanitaria.**
 - **El código dado por la autoridad sanitaria.** Solo en caso de dar positivo en una prueba diagnóstica.
 - De manera opcional **la fecha de síntomas o de prueba diagnóstica positiva.**
 - **Las semillas con la que se generan los identificadores efímeros.** De esta forma se respeta la privacidad del usuario, pues no se puede asociar dicho positivo a ningún identificador.
 - **Tráfico de paquetes disuasorios.** Para evitar que agentes externos identifiquen usuarios infectados, pues la comunicación cliente dirección servidor solo se da en caso de positivo.
 - **De servidor a cliente.**
 - **Las semillas para generar los identificadores.** Las emplean para generar los identificadores y compararlos después con los almacenados. En caso de coincidir se avisa de posible contacto de riesgo. Estas se envían mediante **broadcast** a todos los clientes para evitar distinciones entre usuarios que pudieran afectar a su privacidad.
 - **Entre clientes.**
 - Se intercambian los **identificadores efímeros** que están activos en ese momento a través de **Bluetooth**.
- 2. **Evitar que la aplicación identifique o rastree a los usuarios.** «La aplicación no puede permitir identificar directamente a los usuarios al utilizar la aplicación». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-2, 2020)
- 3. **«La aplicación no ha de permitir que se rastreen los movimientos de los usuarios».** (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-3, 2020)
- 4. **«El uso de la aplicación no debe permitir que los usuarios obtengan información de otros usuarios (y, en particular, que sepan si son o no portadores del virus)».** (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-4, 2020)
- 5. **La confianza en el servidor debe ser limitada.** «La gestión del servidor central debe seguir normas de gobernanza claramente definidas e incluir todas las medidas necesarias para garantizar su seguridad. La ubicación del servidor central debe permitir una supervisión eficaz por parte de la autoridad supervisora competente». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-5, 2020)

6. «**Ha de llevarse acabo una evaluación de impacto relativa a la protección de datos, que debería ponerse a disposición del público.**» (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-6, 2020)
7. «**La aplicación solo debe revelar al usuario si ha estado expuesto al virus y, en la medida de lo posible, sin facilitar información sobre otros usuarios, el número de veces y las fechas de la exposición.**» (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-7, 2020)
8. «**La información transmitida por la aplicación no debe permitir a los usuarios identificar a los usuarios portadores del virus ni conocer sus movimientos.**» (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-8, 2020)
9. **Preservar la identidad de los usuarios frente a las autoridades sanitarias.** «La información transmitida por la aplicación no debe permitir a las autoridades sanitarias identificar a los usuarios que pueden estar expuestos sin el consentimiento de estos». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-9, 2020)
10. «**Las solicitudes cursadas por la aplicación al servidor central no deben revelar ninguna información sobre el portador del virus.**» (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-10, 2020)
11. «**Las solicitudes cursadas por la aplicación al servidor central no deben revelar ninguna información innecesaria sobre el usuario, excepto, posiblemente —y solo cuando resulte necesario—, sus identificadores seudónimos y su lista de contactos.**» (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-11, 2020)
12. «**Impedir ataques de enlace.**» Para evitar el robo de datos de los usuarios. (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-12, 2020)
13. «**Los usuarios han de poder ejercer sus derechos a través de la aplicación.**» (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-13, 2020)
14. «**La supresión de la aplicación debe entrañar la eliminación de todos los datos recogidos a nivel local.**» Esto incluiría la lista de identificadores de los contactos, así como los identificadores propios y las semillas recibidas por el servidor al hacer broadcast. (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-14, 2020)
15. «**La aplicación solo puede recoger datos transmitidos por instancias de la aplicación o de aplicaciones interoperables equivalentes.** No pueden recogerse datos sobre otras aplicaciones ni otros dispositivos de comunicación de proximidad». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-15, 2020)

16. **Implementación de servidores proxy.** «Para evitar la reidentificación por parte del servidor central, deben implementarse servidores proxy. La finalidad de estos servidores no colusores es combinar los identificadores de varios usuarios (tanto los de los portadores del virus como los enviados por los solicitantes) antes de compartirlos con el servidor central, para evitar que este conozca los identificadores de los usuarios (como las direcciones IP)». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-16, 2020)
17. «**La aplicación y el servidor deben desarrollarse y configurarse cuidadosamente con el fin de que no recojan datos innecesarios.** (P. ej., no debe incluirse ningún identificador en los registros del servidor, etc.) y de evitar el uso de **SDK** de terceros que recojan datos para otros fines». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo PRIV-17, 2020)
18. **Salvaguardar la privacidad de la lista de contactos de cara al servidor.** El servidor no invade la lista de contactos de los clientes siendo el mismo cliente el que envía periódicamente la lista de contagios que posea en la base de datos.
19. **Evitar la identificación de los usuarios a partir del código de diagnóstico.** Esto se evita haciendo que dicho código sea solamente asociado al identificador y desde el servidor se envíe la notificación a todos los clientes como se especifica en el patrón software observador modelo de suscripción.
20. **Evitar el acceso al IMEI del dispositivo móvil.** Pues mediante este número de 15 dígitos, se identifica a un terminal cuando se conecta a una red móvil.
21. **Evitar el acceso a la dirección MAC de Bluetooth del móvil.** Pues este identificador único se emplea durante conexiones Bluetooth.
22. **Evitar el acceso a la dirección IPv4 y MAC de la tarjeta WiFi.** Pues pueden utilizarse para identificarnos al conectarnos a la red.

El desarrollo de las aplicaciones de rastreo de contactos puede realizarse desde dos enfoques. Dependiendo de la opción elegida en la parte de diseño, tras analizar el estado del arte y los diferentes protocolos existentes para su desarrollo, se aplicarán un grupo de los siguientes requisitos.

En el caso de que un usuario se declarase infectado, si la aplicación envía a un servidor el historial de los contactos de proximidad que se han obtenido mediante escaneo, se aplicarán los siguientes requisitos:

4.1.3. Requisitos que se aplican únicamente cuando la aplicación envía al servidor una lista de contactos

1. «El servidor central debe recoger el historial de contactos de los usuarios declarados positivos [...] como resultado de una acción voluntaria por parte de estos». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo CON-1, 2020)
2. «El servidor central no debe mantener ni difundir una lista de los identificadores seudónimos de usuarios portadores del virus». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo CON-2, 2020)
3. «El historial de contactos almacenado en el servidor central debe eliminarse una vez se haya notificado a los usuarios su proximidad a una persona con diagnóstico positivo». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo CON-3, 2020)

4. «Excepto si un usuario detectado como positivo comparte su historial de contactos con el servidor central o si un usuario solicita al servidor que investigue su posible exposición su posible exposición al virus, ningún dato debe salir del equipo del usuario». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo CON-4, 2020)
5. «Cualquier identificador incluido en el historial local debe eliminarse a los X días de su recogida (corresponde a las autoridades sanitarias definir el valor X)». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo CON-5, 2020)
6. «Los historiales de contactos enviados por distintos usuarios no deben someterse a tratamiento adicional; por ejemplo, no debe examinarse su correlación cruzada para elaborar mapas globales de proximidad». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo CON-6, 2020)
7. «Los datos contenidos en los registros del servidor han de minimizarse y deben cumplir los requisitos de protección de datos». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo CON-7, 2020)

Por el contrario, en el caso de que un usuario se declarase infectado si la aplicación envía a un servidor la lista de sus propios identificadores difundidos, se aplicarán los siguientes requisitos:

4.1.4. Requisitos que se aplican únicamente cuando la aplicación envía al servidor una lista de sus propios identificadores

1. «El servidor central debe recoger los identificadores de los usuarios declarados positivos [...] difundidos por la aplicación, como resultado de una acción voluntaria por parte de estos». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo ID-1, 2020)
2. «El servidor central no debe mantener ni difundir el historial de contactos de usuarios portadores del virus». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo ID-2, 2020)
3. «Los identificadores almacenados en el servidor central deben eliminarse una vez distribuidos a las demás aplicaciones». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo ID-3, 2020)
4. «Excepto si un usuario detectado como positivo comparte sus identificadores con el servidor central o si un usuario solicita al servidor que investigue su posible exposición al virus, ningún dato debe salir del equipo del usuario». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo ID-4, 2020)
5. «Los datos contenidos en los registros del servidor han de minimizarse y deben cumplir los requisitos de protección de datos». (European Data Protection Board, Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19, Anexo ID-5, 2020)

4.1.5. Requisitos de Usabilidad

Uno de los aspectos más importantes de esta aplicación es que sea accesible para la gran mayoría de las personas debido a su estrecha relación con la salud. El hecho de que en uno de los puntos de la aplicación se requiera que los usuarios comuniquen su contagio e intercambien identificadores entre ellos (explicado más adelante en [Requisitos Funcionales](#)), convierte en una necesidad que una gran mayoría de la población haga uso de la aplicación. Esto, por tanto, obliga a que la interfaz de usuario tenga que estar pensada y diseñada a conciencia.

Los requisitos que debe cumplir el diseño de la interfaz son los siguientes:

1. **La interfaz debe ser sencilla.**
2. **La interfaz debe ser suficientemente intuitiva.**
3. **La interfaz debe ser agradable a la vista.**
4. **La interfaz debe ser accesible para personas con daltonismo.** Se ha empleado un simulador para determinar una paleta de colores apropiada. [67]
5. **El texto debe ser claro y conciso.**
6. **El texto debe ser legible para todas las edades y capacidades visuales.**
7. **La interfaz debe invitar a publicitar su uso.** La interfaz incorporará detalles atractivos e identificativos originales de la aplicación.
8. **La interfaz debe concienciar sobre los síntomas del estado de exposición del usuario.**
9. **Internacionalización.** La interfaz será accesible en diversos idiomas.

4.2. Requisitos Funcionales

Debido a la naturaleza del proyecto, donde habrá una interacción entre distintos despliegues de la aplicación, se ha realizado una distinción entre los requisitos que llegan al usuario y los que ven otros dispositivos.

4.2.1. De cara al usuario

1. **Visualización de mi riesgo de exposición.** El usuario debe recibir retroalimentación visual sobre si ha estado expuesto o no a otro usuario contagiado y en qué medida.
2. **Posibilidad de comunicación de contagio.** Tanto a las autoridades sanitarias como a contactos cercanos.
3. **Notificación y muestra del nivel de exposición a un contagio.** El usuario debe recibir un aviso en el caso de haber tenido un contacto cercano, así como información sobre el nivel de riesgo del mismo.
4. **La aplicación dará directrices y recomendaciones al usuario.** Dependiendo de su nivel de riesgo, la aplicación dará unas u otras recomendaciones acorde a la enfermedad.

4.2.2. De cara a otros dispositivos

1. **Intercambio de identificadores.** Trascurrido un intervalo de tiempo determinado, los dispositivos procederán a intercambiar los identificadores para marcarse entre sí como contactos cercanos.
2. **Permitir la identificación de otros dispositivos con la aplicación.** Para llevar un seguimiento de los contactos cercanos.
3. **Medición del tiempo de exposición a un contacto.** A partir de cierto intervalo de exposición se considerará un contacto directo.
4. **Informar de un contagio.** Se deberá notificar en caso de contagio a los dispositivos que hayan estado en contacto al usuario.
5. **Cálculo de la intensidad del contacto.** Dependerá de la distancia entre usuarios. A menor distancia mayor intensidad.
6. **Cálculo de la distancia entre dispositivos.** La aplicación será capaz de determinar la distancia con otro dispositivo con el fin de determinar el nivel de riesgo.
7. **Envío de códigos de contagio al servidor de la autoridad sanitaria.** Una vez allí ya se encarga la propia autoridad de comprobar y verificar que sea un código válido.
8. **Comunicación con el servidor para obtener lista de nuevos contagios.** El servidor comunicará periódicamente a los clientes la lista de nuevos contagios.

4.2.3. De cara a la propia aplicación

1. **Cálculo de los identificadores.** Se deberán generar tal que sea computacionalmente imposible relacionarlos con el usuario al que están asociados.
2. **Cálculo de estado de exposición.** Con el objetivo de proporcionar al usuario una aproximación del posible riesgo de contagio al que se ha visto expuesto.

4.3. Requisitos de Información

Información almacenada localmente en el dispositivo del usuario

1. **Identificadores anónimos propios de cada usuario.** Estos deberán preservar la privacidad de la información asociada al usuario a la vez que permitirán el seguimiento de su estado de salud.
2. **Información anonimizada sobre qué usuarios han estado en contacto.** Dado que no es necesario saber con quién se ha estado sino solamente si se ha estado en contacto con usuarios infectados.

Información almacenada en el servidor

1. **Datos asociados a los identificadores infectados.** Con el fin de evitar el almacenamiento de los identificadores propiamente, en su lugar se almacenarán en el servidor los elementos o semillas necesarios para su generación.
2. **Fecha de recepción.** Se almacenará la fecha de recepción de cada comunicado al servidor.

4.4. Base de Datos

Como se ha especificado antes, en esta aplicación es necesario el almacenamiento de cierta información tanto en los clientes como en el servidor. Por esta razón se debe estructurar una forma de almacenamiento de dichos datos, la cual permita organizarlos y tenerlos a disposición.

La existencia de una aplicación cliente y otra servidor obliga a diferenciar dos sistemas de almacenamiento, cada uno orientado a sus necesidades específicas.

Es por ello que se procede a modelar dos bases de datos, cada una dedicada a una parte del proyecto, es decir, cliente y servidor.

Modelo Conceptual

Los datos que se manejan en esta aplicación son los identificadores efímeros. Estos consisten en un número asociado de manera anónima a un individuo. En concreto, poseen los siguientes atributos.

- **Valor.** Es el número que define como tal al identificador. Se genera mediante [HMAC-SHA256](#) a partir de la clave y fecha generadoras, así como de una variable global.
- **Clave generadora.** Es uno de los parámetros necesarios para generar un identificador. Esta clave es el resultado SHA-256 de la clave anterior.
- **Fecha generadora.** Es otro de los parámetros necesarios para generar un identificador. Se corresponde a la fecha de generación del identificador.
- **Fecha de recepción.** Es la fecha de recepción del identificador en el dispositivo. Se emplea para llevar cuenta de los catorce días que está activo un identificador.

Por lo tanto, un identificador sería tal que así:

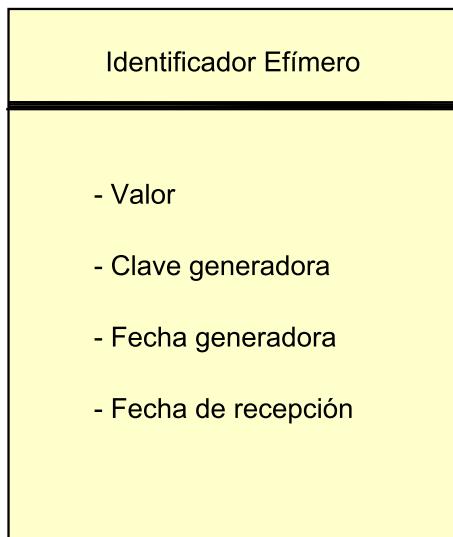


Figura 3: Modelo conceptual

Modelo Lógico

Debido a las características de este proyecto se han considerado dos bases de datos. Una de ellas será la base de datos local, donde se almacenarán los identificadores tanto propios como los ajenos, es decir, los obtenidos vía Bluetooth.

Por otro lado, la base de datos del servidor almacenará aquellas **claves** y **fechas generadoras** asociadas a identificadores contagiados. Tan solo retendrá las que haya recibido en los últimos catorce días, tiempo durante el cual el virus permanece activo.

La base de datos local tendrá diferenciados los identificadores propios de los ajenos. Esto es debido a que poseen distintos atributos, y de este modo, se evitará el guardado de múltiples campos con valor *NULL* o vacío. En concreto cada tabla contiene los siguientes atributos:

- **Identificadores propios.**

- **Valor.** Es el número que define propiamente al identificador.
- **Clave generadora.** Es uno de los parámetros necesarios para la generación del identificador.
- **Fecha generadora.** Es uno de los parámetros necesarios para la generación del identificador.

- **Identificadores ajenos.**

- **Valor.** Es el número que define propiamente al identificador. Se recibe en el intercambio vía Bluetooth.
- **Fecha de recepción.** Es la fecha de recepción en el dispositivo del valor del identificador.

Así pues, queda de la siguiente forma:

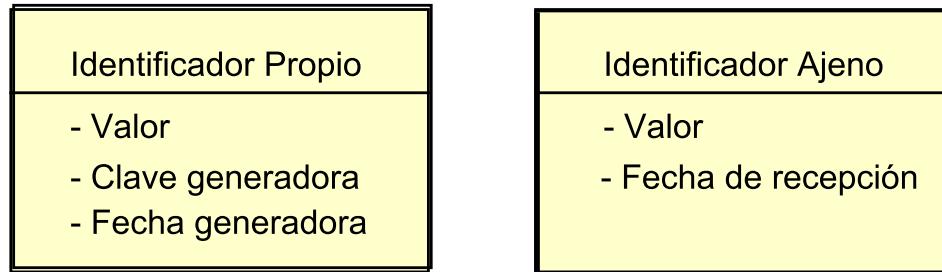


Figura 4: Modelo lógico Cliente

Por otro lado, la base de datos del servidor almacena aquella información asociada a los identificadores contagiados. Contendrá la clave y fecha generadoras de cada uno de ellos, así como el momento de recepción de cada uno. Esto último es con el fin de ir eliminando aquellos que posean una fecha extinta, es decir, hayan transcurrido catorce días.

- **Clave generadora.** Es uno de los parámetros necesarios para la generación del identificador.
- **Fecha generadora.** Es uno de los parámetros necesarios para la generación del identificador.
- **Fecha de recepción.** Es la fecha de recepción en el servidor de los datos del identificador contagiado.

De este modo, quedaría de la siguiente forma:

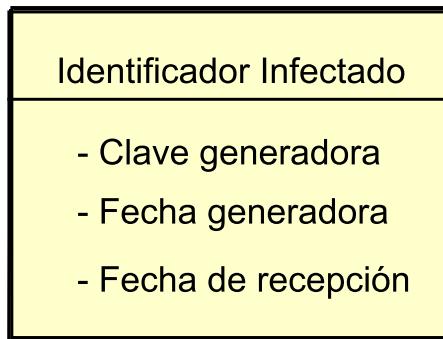


Figura 5: Modelo lógico Servidor

4.5. Modelo de intercambio de datos

Los actores son los siguientes:

- **Aplicaciones cliente.** Realizarán el intercambio de identificadores actuando entre ellas como cliente y servidor alternando roles.
- **Servidor.** El servidor recibe de un cliente contagiado el código y las semillas asociadas a sus identificadores. Además, de manera aleatoria, todos los clientes envían datos análogos pero falsos con el fin de generar ruido. A la hora de interactuar el servidor con los clientes para enviar los datos, se realiza un broadcast con el fin de salvaguardar la privacidad al no hacer diferenciación entre clientes.

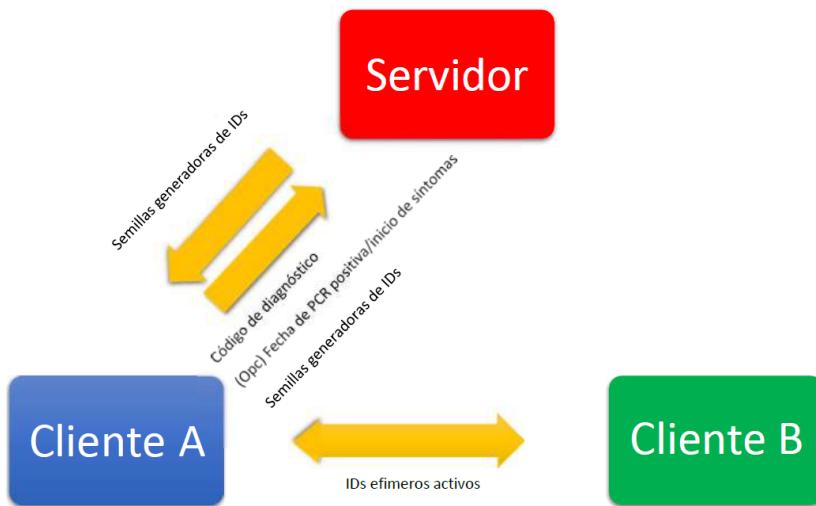


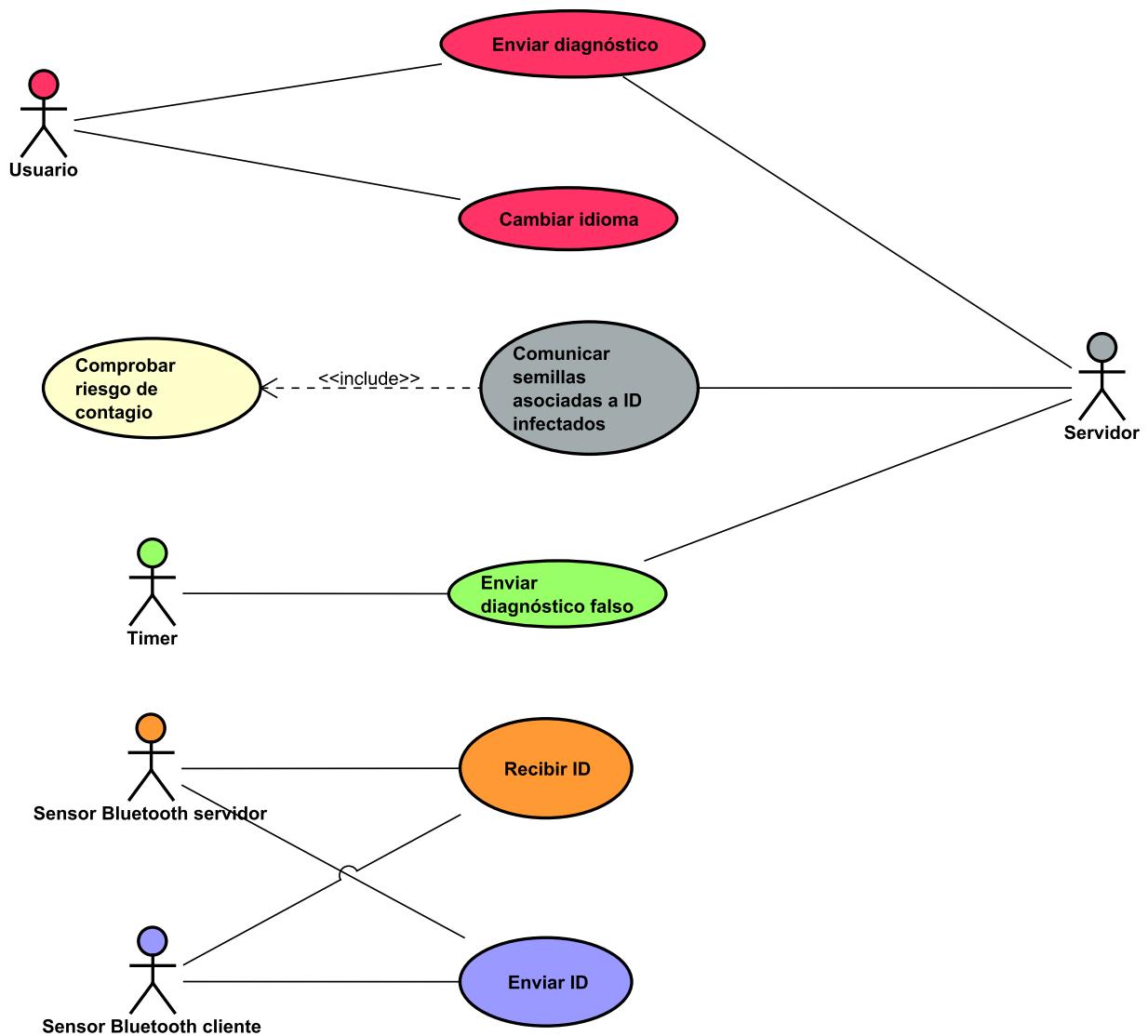
Figura 6: Modelo de intercambio de datos

4.6. Desarrollo de los casos de uso

Tras un análisis de los requisitos elicítados en este documento, hemos llegado a la conclusión de que nuestro sistema implica la actuación de cinco actores:

- **Usuario.** Sería el actor principal, es decir, a quien va pensada la funcionalidad de la aplicación.
- **Servidor.** Sirve como punto de contacto con la autoridad sanitaria pertinente. Su función principal es la gestión de los datos referentes a los identificadores efímeros y los diagnósticos, tanto su envío y recepción como el borrado de registros antiguos o no vigentes.
- **Timer.** Es un representación del paso del tiempo, el cual desencadena aquellos casos de uso que se llevan a cabo según un periodo temporal. En segundo plano, envía al servidor de manera periódica códigos falsos. Esto se hace con la finalidad de generar ruido, pues de otro modo podría detectarse quién está infectado ya que el envío de datos dirección cliente servidor solo se haría al comunicar un contagio.

- **Sensor Bluetooth servidor.** Se trata de un actor que actúa como parte de la comunicación Bluetooth, el cual es el encargado de recibir los identificadores enviados por el Sensor Bluetooth cliente de otros dispositivos implicados en el intercambio.
- **Sensor Bluetooth cliente.** Al igual que el anterior, se trata de un actor que actúa como parte de la comunicación Bluetooth. En cambio, la función de este es transmitir los identificadores a recibir por el Sensor Bluetooth servidor del resto de dispositivos implicados en el intercambio.

**Figura 7:** Diagrama de casos de uso

Como se puede apreciar en el diagrama anterior los casos de uso serían los siguientes:

4.6.1. Enviar diagnóstico

ELEMENTO	VALOR
Caso de Uso	Enviar diagnóstico
Resumen	Se realiza este caso de uso cuando el actor Usuario quiere comunicar su contagio.
Actor	Usuario, Servidor
Precondición	Tener conexión a Internet.
Postcondición	Las semillas generadoras del actor Usuario aparecen como infectadas en el servidor. La aplicación (<i>usuario</i>) cambia su estado a infectado.
Secuencia Base	<p>1- El actor Usuario introduce el código de diagnóstico proporcionado por la autoridad sanitaria.</p> <p>2- El sistema pide confirmación al usuario.</p> <p>3- El actor usuario verifica la acción.</p> <p>4- El sistema envía el código al servidor.</p> <p>5- El sistema le comunica al actor Usuario que el código es correcto y le informa de las medidas que debe tomar.</p> <p>6- El sistema envía las semillas generadoras de los ID del actor Usuario al servidor.</p>
Secuencia Alternativa	
Excepciones	<p>3'- El actor usuario no verifica la acción y se vuelve al paso 1.</p> <p>5'- El sistema comunica un error de conexión y el caso de uso queda sin efecto.</p> <p>5"- El sistema comunica al actor Usuario que el código es incorrecto y el caso de uso queda sin efecto.</p>
Sub Caso de Uso	

Tabla 1: Caso de uso: Enviar diagnóstico

4.6.2. Cambiar idioma

ELEMENTO	VALOR
Caso de Uso	Cambiar idioma
Resumen	Se realiza este caso de uso cuando el actor Usuario quiere cambiar el idioma de la aplicación.
Actor	Usuario
Precondición	
Postcondición	El idioma de la aplicación ha cambiado al seleccionado por el actor Usuario.
Secuencia Base	<p>1- El actor Usuario selecciona un idioma entre los disponibles.</p> <p>2- El sistema le pide confirmación al actor Usuario.</p> <p>3- El actor Usuario confirma su selección.</p> <p>4- El sistema aplica los cambios sobre la aplicación.</p>
Secuencia Alternativa	
Excepciones	3'- El actor Usuario no confirma la acción y el caso de uso queda sin efecto.
Sub Caso de Uso	

Tabla 2: Caso de uso: Cambiar idioma

4.6.3. Comunicar semillas asociadas a IDs infectados

ELEMENTO	VALOR
Caso de Uso	Comunicar semillas asociadas a ID infectados
Resumen	Se realiza un broadcast del contenido de la base de datos del servidor de forma periódica con el fin de notificar posibles contagios a los clientes.
Actor	Servidor
Precondición	Debe haber pares clave - fecha marcados como infectados en la base de datos. Tener conexión a Internet
Postcondición	Los clientes obtienen los pares clave - fecha necesarios para generar los ID's infectados.
Secuencia Base	1- El actor Servidor envía los pares infectados a todos los clientes. 2- Se realiza el caso de uso <i>Comprobar riesgo de contagio</i>
Secuencia Alternativa	
Excepciones	
Sub Caso de Uso	Comprobar riesgo de contagio

Tabla 3: Caso de uso: Comunicar semillas asociadas a IDs infectados

4.6.4. Comprobar riesgo de contagio

ELEMENTO	VALOR
Caso de Uso	Comprobar riesgo de contagio
Resumen	Este caso de uso se realiza para comprobar si ha habido algún contacto con algún usuario infectado.
Actor	
Precondición	Debes haber recibido pares clave - fecha infectados.
Postcondición	El sistema muestra el riesgo de contagio más alto entre los identificadores comprobados.
Secuencia Base	1- El sistema genera los ID's asociados a los pares recibidos. 2- El sistema compara los ID's generados con aquellos obtenidos mediante intercambio Bluetooth. 3- El sistema muestra el riesgo de contagio más alto de entre los ID's coincidentes.
Secuencia Alternativa	
Excepciones	3'- Ningún ID generado coincide el caso de uso queda sin efecto.
Sub Caso de Uso	

Tabla 4: Caso de uso: Comprobar riesgo de contagio

4.6.5. Enviar diagnóstico falso

ELEMENTO	VALOR
Caso de Uso	Enviar diagnóstico falso
Resumen	Este caso de uso se realiza un número aleatorio de veces a lo largo del dia
Actor	Timer, Servidor
Precondición	Tener conexión a Internet.
Postcondición	Se genera tráfico falso que realiza la función de ruido en la red con la finalidad de prevenir ataques pasivos de escucha.
Secuencia Base	<p>1- El actor Timer genera unas claves falsas y una fecha aleatoria fuera de los últimos 14 días.</p> <p>2- El sistema genera un código de diagnóstico falso.</p> <p>3- El sistema envía el código al servidor.</p> <p>4- El sistema envía las claves y fechas generadoras de los ID del actor Usuario al servidor.</p>
Secuencia Alternativa	
Excepciones	3'- El sistema comunica un error de conexión y el caso de uso queda sin efecto.
Sub Caso de Uso	

Tabla 5: Caso de uso: Enviar diagnóstico falso

4.6.6. Recibir ID

ELEMENTO	VALOR
Caso de Uso	Recibir ID
Resumen	Este caso de uso se realiza cuando el actor Sensor Bluetooth servidor detecta una señal Bluetooth de otro dispositivo para recibir su ID.
Actor	Sensor Bluetooth servidor, Sensor Bluetooth cliente
Precondición	El dispositivo debe tener el Bluetooth y la geolocalización activados.
Postcondición	El ID recibido es almacenado en la base de datos del sistema.
Secuencia Base	<p>1- El actor Sensor Bluetooth servidor detecta una señal Bluetooth de la aplicación.</p> <p>2- El sistema inicia un contador de 15 min.</p> <p>3- El actor Sensor Bluetooth servidor recibe del actor Sensor Bluetooth cliente el ID efímero correspondiente a ese periodo de tiempo.</p> <p>4- El sistema almacena el ID recibido en la base de datos junto con la intensidad de señal recibida.</p>
Secuencia Alternativa	
Excepciones	3' - Se deja de detectar señal Bluetooth antes de agotar los 15 min y el caso de uso queda sin efecto.
Sub Caso de Uso	

Tabla 6: Caso de uso: Recibir ID

4.6.7. Enviar ID

ELEMENTO	VALOR
Caso de Uso	Enviar ID
Resumen	Este caso de uso se realiza cuando el actor Sensor Bluetooth cliente detecta una señal Bluetooth de otro dispositivo para enviar su propio ID.
Actor	Sensor Bluetooth cliente, Sensor Bluetooth servidor
Precondición	El dispositivo debe tener el Bluetooth y la geolocalización activados.
Postcondición	El ID efímero del periodo de tiempo correspondiente se envía al Sensor Bluetooth servidor.
Secuencia Base	<p>1- El actor Sensor Bluetooth cliente detecta una señal Bluetooth de la aplicación.</p> <p>2- El sistema inicia un contador de 15 min.</p> <p>3- El actor Sensor Bluetooth cliente envía al actor Sensor Bluetooth servidor el ID efímero correspondiente a ese periodo de tiempo.</p>
Secuencia Alternativa	
Excepciones	3' - Se deja de detectar señal Bluetooth antes de agotar los 15 minutos y el caso de uso queda sin efecto.
Sub Caso de Uso	

Tabla 7: Caso de uso: Enviar ID

5. Diseño de la aplicación

Partiendo de los requisitos anteriormente elicitudes, se deben tomar ciertas decisiones importantes de diseño. Para ello, se acotan las soluciones que vamos a implementar para los problemas que han sido presentados anteriormente en los requisitos.

5.1. Estado del Arte: Protocolos

El protocolo de comunicación de exposición es una de las partes principales de la aplicación. Existen diferentes protocolos para poder llevarlo a cabo. Se recogen a continuación los más relevantes. [38]

5.1.1. Decentralized Privacy-Preserving Proximity Tracing (DP-3T)

Se trata de un protocolo descentralizado de código abierto. Este protocolo se basa en la generación de identificadores efímeros, los cuales se intercambian cuando dos clientes se encuentran a una distancia inferior a 2 metros y durante más de 15 minutos de exposición.

El proceso para generar estos identificadores, ilustrado en la Figura 8 es el siguiente:

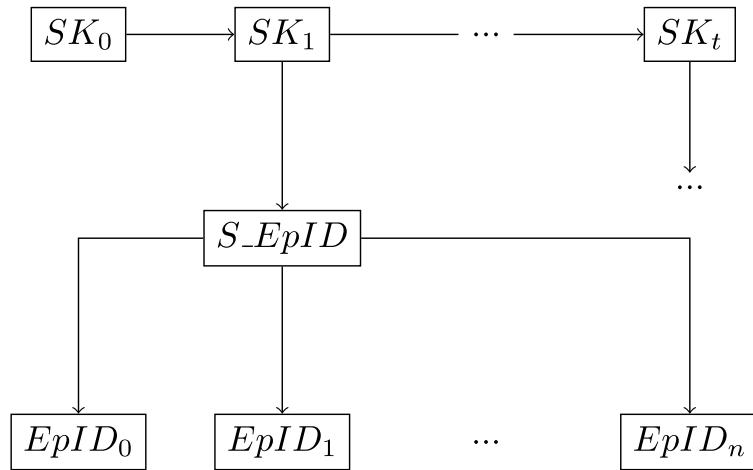


Figura 8: Esquema de generación de identificadores efímeros

1. **Generación de SK_t .** O Secret Key (clave secreta) número t . Inicialmente se genera una clave secreta SK_t correspondiente al día t actual. Esta clave se obtiene a partir del resumen hash SHA-256 de la clave SK_{t-1} . La generación de la primera SK_0 se hace mediante el algoritmo de curvas de Edward Ed25519.
2. **Generación del S_EphID .** O Secret Ephemeral IDentifier (identificador efímero secreto). A partir del SK_t del día se genera el S_EphID empleando una función:

$$S_EphID(BK) = PRG(PRF(SK_t, BK))$$

Donde PRG es un cifrado de flujo que produce $n * 16$ bytes, siendo n el número de identificadores diarios. El número n se determina tal que $n = (2460)/l$, siendo l el tiempo de vida en minutos de un identificador efímero. PRF es una función pseudoaleatoria de la forma HMAC-SHA256 y BK es una variable global.

3. **Generación de EphID_n** Tras ello el S_EphID se subdivide en n fragmentos de tamaño 16 bytes. Cada uno de estos fragmentos es un EphID cuyo orden de uso se determina de forma aleatoria.

En concreto, en DP-3T el tiempo de uso para su intercambio de un identificador efímero es de 15 minutos.

Cuando dos usuarios se encuentran, las aplicaciones móviles comienzan a actuar entre ellas como servidor-cliente, intercambiando los roles, para de esta manera enviarse mutuamente los identificadores.

Cuando se produce un contagio, el usuario envía un código de verificación que previamente la autoridad sanitaria le ha proporcionado. Este código se envía junto a las semillas generadoras. Esta información recibida por el servidor es enviada de forma periódica a los clientes. Así, las aplicaciones cliente pueden calcular los identificadores contagiados a partir de las semillas generadoras recibidas desde el servidor. Si alguno de estos identificadores generados coincide con uno de los almacenados significa que ha habido una exposición a contagio y el protocolo avisa al usuario de ello a través de la aplicación cliente.

Al enviar las semillas generadoras, se preserva la privacidad de los usuarios contagiados, pues los identificadores no son enviados nunca como tal. [19] [28] [40]

Este protocolo se creó para apoyar el protocolo GAEN, funcionando sobre él aunque con algunos cambios a nivel de tratamiento y a la hora de crear las semillas generadoras y los identificadores.

5.1.2. (Google/Apple) Exposure Notification (GAEN) system

Originalmente conocido como *Privacy-Preserving Contact Tracing Project*.

Este protocolo emplea un enfoque descentralizado. Fue creado con el fin de que existiera una comunicación entre dispositivos **Android** e **iOS**. Sin embargo no es compatible con los dispositivos **Huawei** posteriores a mayo de 2019. Está implementado a nivel de sistema operativo para de esta forma ser más eficiente al realizar todos los procesos en segundo plano. Funciona de manera muy similar a DP-3T, empleando identificadores efímeros (EphIDs), los cuales cambian cada 15-20 minutos (al resetearse la MAC Bluetooth del dispositivo). Estos son calculados mediante una [clave AES](#) y una marca de tiempo calculada a partir de [Unix Epoch Time](#).

Cuando se produce un contagio, desde la aplicación cliente se suben al servidor las semillas generadoras. De esta forma, el servidor puede reenviar esa información a los demás usuarios y estos generar los identificadores correspondientes. Si alguno coincidiera con uno almacenado, el protocolo avisa a través de la aplicación cliente de que se ha estado expuesto a un posible contagio. [29]

La diferencia principal con DP-3T se encuentra en la generación de las claves secretas SK. En DP-3T estas claves se obtienen a partir de un resumen hash de la clave SK del día anterior. Sin embargo, en GAEN todas las claves secretas son generadas a partir del mismo inicializador.

Otra diferencia reside en el sello temporal empleado para generar esos identificadores. DP-3T utiliza una marca de tiempo más basta o un resumen hash de la misma. Por ello garantiza la privacidad mejor que GAEN, aunque a costa de ser más vulnerable ante los ataques de repetición, pues son más fáciles de llevar a cabo debido a un período de validez más largo (ya que las estampas temporales son menos precisas y por lo tanto hay más décimas de tiempo entre ellas). [23]

5.1.3. Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT/PEPP)

Este protocolo es descartado debido a la necesidad de registro en el servidor, medida tomada para evitar multicuentas. Esto se hace mediante datos personales pseudónimos que se usan para generar el identificador **PUID**. Dicho identificador es necesario para que el servidor asocie dicho dispositivo y sea capaz de enviarle los datos pertinentes ante el registro de casos positivos. El PUID se emplea junto a una clave global, la cual cambia cada 60 minutos, para generar en el servidor los ID efímeros **EBID**. Estos se envían mediante *broadcast* a los clientes de la aplicación. Dichas claves globales se eliminan a las 4 semanas.

El hecho de que el servidor sea capaz de obtener los PUID originales mediante la clave y los EBID, lo convierte en un sistema con gran capacidad para identificar a usuarios, de divulgación completa y correlación. Esto le convierte en un objetivo con un riesgo muy alto, pues es posible reidentificar a los usuarios mediante los PUID y las claves, ya que estas se quedan almacenadas durante bastante tiempo.

Por otro lado, los EBID permiten el rastreo en tiempo real de los usuarios, infectados o no. Esto es debido a que el servidor *backend* permite su conversión en identificadores permanentes, relacionando cualquier reporte o interacción del portador con dispositivos y sensores Bluetooth al PUID original.

Además, debido a que no existe una metodología de identificación y autenticación de los EBID, es posible generarlos de manera falsa, asociándolos a un usuario de manera externa. Esta amenaza puede concretarse en un ataque de un tercero que, asociando un EBID falso a un usuario, sea capaz de rastrearlo. El servidor nunca identificaría dicha anomalía al carecer de una firma digital o certificado que corrobore la autenticidad del EBID.

Esto desanonimiza a los usuarios sin necesidad de atacar al servidor al asignarles un EBID persistente externo, permitiendo su geolocalización constante mediante sensores Bluetooth.

Los detalles de los contactos de un caso positivo son revisados de manera manual por la entidad sanitaria con el fin de evitar falsos positivos, haciendo el trabajo más lento que estando gestionado por un servidor como en las variantes de DP-3T. [51]

Algunos riesgos principales a raíz de estas características son:

- **Falsificación de un positivo.**

Dado que los usuarios infectados suben al servidor su lista de contactos, es posible realizar una inyección de un EBID en dicha lista para generar un falso positivo. Dado que la verificación de los encuentros no es posible, no hay manera de defenderse de dicho ataque.

Este problema no se encuentra en protocolos descentralizados, pues los identificadores del registro de contactos no se suben al servidor en ningún momento. Además es necesario el consentimiento de la autoridad sanitaria para notificar de un positivo.

- **Riesgo de compromiso de datos en dispositivos desbloqueados.**

Actualmente, el desarrollo de este protocolo es imposibilitado debido a que el día 10 de abril de 2020 Apple y Google, acorde a la minimalización de datos, introdujeron una nueva api de Rastreo de Contactos, donde se imposibilita la transmisión de la lista de contactos vía red como exige el protocolo PEPP-PT/PEPP.

5.1.4. BlueTrace

El principal inconveniente de este protocolo, al igual que en [PEPP-PT/PEPP](#) es el uso del procesamiento de reportes centralizado.

Los protocolos que utilizan este tipo de procesamiento tienen como principal inconveniente el envío de los datos de contacto del usuario a las autoridades sanitarias.

Entre las responsabilidades de dichas autoridades se encuentran asignar los detalles del contacto a cada usuario, determinar si ha habido un contagio y finalmente advertir a los usuarios si este ha ocurrido. Esto implica una correlación directa entre el usuario y los contactos.

En cambio, los protocolos descentralizados delegan todas estas funciones en la red, aumentando así la eficiencia y la privacidad de los usuarios.

A diferencia de PEPP-PT/PEPP, BlueTrace genera los identificadores temporales (TempIDs) utilizando el identificador del usuario, el instante de tiempo en el que se crea el TempID, el tiempo de expiración del ID, un [vector de inicialización \(IV\)](#) y una clave privada proveniente de la autoridad sanitaria.

Los tres primeros parámetros se transmiten encriptados pero el IV lo hace en texto plano. Si a esto le sumamos un ataque a los servidores de las autoridades sanitarias, se podrían llegar a desencriptar los TempID.

Otro inconveniente, propio de este protocolo, es que es necesario proveer el número de teléfono para poder iniciar las aplicaciones que lo utilicen y que las autoridades comuniquen a ese número de teléfono un posible contacto, lo que vulnera claramente la privacidad del usuario. [37]

5.1.5. Otros

A parte de los protocolos mencionados en la sección anterior, existen otras alternativas menos probadas, pero bastantes prometedoras. Estas son consideradas alternativas factibles al protocolo DP-3T, pues presentan un grado similar de privacidad. Estos protocolos son:

- **ConTra Corona.**

Es un protocolo que pretende aprovechar las virtudes de los protocolos centralizados, pero delegando cierta funciones principales en otras organizaciones, para minimizar la confianza requerida en el servidor central.

Para ello, esta solución se basa en el paradigma [Upload What You Observed](#), que consiste en enviar al servidor todos los pseudónimos (los identificadores efímeros de este protocolo) que ha recogido en un periodo determinado. Esto marca una diferencia con respecto a DP-3T, que utiliza el paradigma [Upload What You Sent](#), es decir enviar los pseudónimos que has usado durante el periodo de tiempo. [12]

- **EpiOne.**

Se trata de un protocolo híbrido que emplea criptografía de clave pública. Permite identificar cuántos tokens almacenados por un usuario coinciden con los que posee el servidor pero sin la necesidad de que el usuario revele sus tokens.

Esto se logra con cardinalidad de intersección de conjuntos privados o [PSI-CA](#). La PSI-CA de EpiOne permite que haya dos partes, cada una con un conjunto privado de tokens, conozcan el tamaño de la intersección entre sus conjuntos sin revelar más información. Con ello se puede ver si hay alguna coincidencia, pero no cuál. De esta forma se respeta la privacidad y a su vez se puede alertar en caso de contacto con alguien contagiado. [62]

■ Pronto-C2.

Este protocolo es totalmente descentralizado. Este utiliza el [algoritmo de Diffie-Hellman](#) para establecer la clave privada con la que trabajará. También emplea [firmas digitales ciegas](#) para preservar la anonimidad del emisor a la vez que se autentifica. Este protocolo emplea direcciones e identificadores efímeros para cada usuario, las cuales se envían al servidor en caso de contagio.

Las comunicaciones con el servidor se realizan vía redes privadas como [TOR](#) con el fin de preservar el anonimato de los datos enviados. [10]

Por otro lado los protocolos que menos vulnerabilidades poseen son *Hamagen* y *COVID Safe Paths*. Sin embargo, emplean constante geolocalización y los datos enviados son las rutas de los usuarios contagiados, por lo que a nivel de privacidad dejan mucho que desear.

Existen más protocolos descentralizados como son PACT (East-coast), PACT (West-coast), DP-3T Unlinkable, TCN o DESIRE, que es híbrido. [8]

5.2. Elección del protocolo

Basándonos en los argumentos proporcionados por el estudio realizado por Serge Vaudenay [65] hemos decidido el uso de DP-3T para nuestra aplicación. Esto es debido a los siguientes puntos:

- En el caso de los sistemas centralizados, los ataques suelen tener consecuencias más graves, pues el acceso es a un único lugar, poniendo en peligro la privacidad de los usuarios ante usuarios malintencionados. El uso de un protocolo descentralizado nos permite mitigar el riesgo a que los usuarios sean rastreados por terceros.
- Si ponemos el foco en el grafo de contactos entre los usuarios, los sistemas centralizados pueden llegar a revelar parte del grafo a un servidor malintencionado. En cambio, los descentralizados solo pueden revelar a un determinado usuario si ha habido contacto entre únicamente dos usuarios.
- Aunque los protocolos descentralizados pueden permitir ciertos ataques de robo de las identidades de usuarios contagiados, DP-3T busca cubrir algunos de ellos. Una de sus soluciones es ante [ataques de Paparazzi](#). Esta consiste en enviar el identificador “desmenuzado”, enviando la clave secreta y la fecha necesarias para generarla por separado. Sin embargo, es cierto que desde este enfoque sería mejor un protocolo centralizado debido a que otros ataques como [Nerd attack](#) o el [Militia attack](#) no están cubiertos en DP-3T. [66]

Sin embargo, debido a que, a grandes rasgos, cara a la privacidad el riesgo de manejar un protocolo centralizado es mayor que uno descentralizado, optamos por el uso de DP-3T. Esto es porque consideramos más peligroso el hecho de que se pueda acceder a una parte del grafo de contactos de los usuarios o el rastreo de los mismos, que determinar que un individuo puntual esté contagiado.

5.3. Imprevistos respecto al protocolo elegido y consecuencias

Tras comenzar a investigar sobre la implementación de DP-3T para la aplicación, surge el inconveniente de que, si bien su código es abierto, es necesario cumplimentar una solicitud¹ para tener acceso a la API.

Provide advance notice to the Google Play App Review team

You are a governmental public health authority that would like to use the Android Exposure Notifications APIs and you are providing written documentation attesting to the ownership of a developer account or submitting authorization for a developer you have commissioned to create an app on your behalf. [Learn more](#).

⚠ Please Note: If you submit a request that's not covered by the above scenario(s), you may not receive a response. For other questions, [contact our support team](#).

* Required field

Figura 9: Solicitud a cumplimentar para tener acceso a la API

Attach a written documentation proving that the government is intending to use the Android Exposure Notifications APIs in response to covid-19 and attesting to the ownership of a developer account or submitting authorization for a developer you have commissioned to create an app on your behalf.

Please do not submit any documents or IDs with personally identifiable information. Make sure to block out/remove any sensitive information (financial and payment information, phone numbers, etc) from any documents.

*

No files chosen

[+ Choose files](#)

Figura 10: Documentación requerida

Uno de los requisitos exigidos para tener acceso a la API es ser representante o tener el permiso de una Autoridad Sanitaria Pública Gubernamental, así como proveer de una documentación atestando que se está en posesión de una cuenta Google de desarrollador o dando dichos permisos a otra cuenta de la que no se es propietario. Dado que no es nuestra situación, no es posible implementar la aplicación con el protocolo inicialmente elegido.

A causa de este contratiempo, es necesario reorientar el desarrollo a un nuevo protocolo. El hecho de generar nuevos planteamientos entra dentro de la metodología inicialmente elegida para el desarrollo e implementación de la aplicación, en este caso, Extreme Programming. Así pues, aunque se trate de algo imprevisto, es un cambio factible y que no trastocará la planificación inicial.

Como previamente se realizó un estudio de los posibles protocolos candidatos, se procede a considerar cuáles pueden ser otras buenas opciones.

¹https://support.google.com/googleplay/android-developer/contact/expo_notif_api

Uno de los requisitos más importantes a tener en cuenta para el desarrollo de la aplicación es el uso de un protocolo basado en el intercambio de claves y no en otros métodos, como la geolocalización o uso de datos personales, que son mucho más invasivas en lo que refiere a la privacidad de los usuarios. Cualquier método que no cumpla los requisitos impuestos por la [European Data Protection Board, 2020](#), será descartado.

Con esto en mente, se vuelven a analizar las alternativas con las que contamos, esta vez considerando en mayor profundidad sus aportaciones y adaptación a la idea definida en los requisitos.

(Google/Apple) Exposure Notification (GAEN) system

Esta opción es **descartada** de base, pues de hecho es la causa de no poder emplear DP-3T para el desarrollo. Debido a la alta correlación existente entre DP-3T y GAEN, ambos protocolos emplean la misma API de Google. Esto es debido a que DP-3T está implementado empleando GAEN como base. De este modo, el problema no estaría resuelto y la implementación seguiría siendo imposible con el alcance que se posee actualmente.

BlueTrace

A la hora de analizar este protocolo, se encuentra que es de código abierto y su implementación sería posible debido a que todo lo necesario para ello se puede encontrar en GitHub².

Si bien inicialmente parece un buen candidato, ya que su estructura basada en el intercambio de claves encaja en la idea plasmada en los requisitos, nos encontramos con dos motivos a considerar en su contra.

- **Upload What You Observed.** BlueTrace emplea una metodología centralizada. En concreto, esto se traduce en que el protocolo emplea un paradigma *Upload What You Observed* en lugar de *Upload What You Sent*, que es el usado por DP-3T o GAEN.

Esto implica que el procesamiento de qué identificadores han estado en contacto con un individuo infectado se lleva a cabo de forma centralizada en el servidor. Por el contrario, con un protocolo descentralizado este procesamiento recaería en los clientes.

Así pues, el uso de un paradigma *Upload What You Observed* se ajusta peor que uno *Upload What You Sent* al requisito [La confianza en el servidor debe ser limitada](#).

Dado que se trata de un requisito propiamente dictaminado por el European Data Protection Board, se considera importante su cumplimiento.

Otra razón de peso es que además el uso de protocolos centralizados ha sido fuertemente criticado. Esto es debido a la recopilación de datos de los usuarios que pueden realizar, pues el servidor sabría no solo qué identificadores están contagiados, sino sus contactos y por lo tanto quiénes han quedado expuestos. [61] [32] [18] [22] [31] [15] [34]

- **Identificación de los usuarios.** BlueTrace requiere de un registro por parte del usuario, el cual se lleva a cabo cediendo su número de teléfono. Esta información se emplea únicamente con el fin de contactar a pacientes potencialmente infectados. El hecho de usar esta información personal no es realmente necesario como otros protocolos han demostrado.[14]

Son diversos los **requisitos** que no se verían cumplidos de implementar esta opción, entre ellos algunos de los dictaminados por el EDPB en lo que refiere a aplicaciones de rastreo de contactos.

Como puede deducirse de todo lo previamente mencionado, este protocolo puede resultar un tanto invasivo para la privacidad de los usuarios, incumpliendo diversos requisitos dictaminados por el European Data Protection Board. Es por ello que queda **descartado**.

²<https://github.com/OpenTrace-community>

PEPP-PT/PEPP

Este protocolo, al igual que BlueTrace, posee un planteamiento centralizado, que como ya se ha visto, ha sido ampliamente criticado.

Por otro lado, a diferencia de BlueTrace, PEPP-PT/PEPP no hace uso del número de teléfono del usuario como metodología de registro. En su lugar, el servidor procesa datos personales pseudónimos. Si bien parece una aproximación más respetuosa con la privacidad, la desanonymización de los datos es posible. Dado uno de estos identificadores, es posible etiquetar y clasificar a un usuario de tal modo que terceros puedan reconocerlos sin necesidad de acceder a la base de datos del servidor. Esto es debido a que el *backend* puede reconvertir cualquier identificador en un identificador permanente.

El problema se agranda en el momento en el que el servidor es capaz de relacionar qué pseudónimos han estado en contacto mediante las subidas de identificadores al servidor al reportar un contagio. De este modo, es posible averiguar no solo si un individuo está o no contagiado, sino sus movimientos y con quién ha estado en contacto. [4]

Este planteamiento supone una gran invasión de la privacidad de los usuarios pues, aunque se trata de datos pseudónimos, el hecho de poder realizar una conversión inversa de los mismos expone una gran cantidad de información personal y sensible del usuario, tanto relacionada con su salud como con sus desplazamientos. A razón de estos hechos, este protocolo ha sido ampliamente juzgado, hasta el punto de que el 20 de abril de 2020 una carta pública fue firmada por 300 académicos de seguridad y privacidad de hasta 26 países diferentes, criticando su funcionamiento y alegando que «solutions which allow reconstructing invasive information about the population should be rejected without further discussion» [Aquellas soluciones que permitan reconstruir información invasiva sobre la población deben rechazarse sin mayor discusión.] ([Joint Statement on Contact Tracing, Párrafo 2](#)) [39] [51]

Es por ello que el protocolo queda obviamente **descartado**.

Protocolos similares a DP-3T en desarrollo

Como se mencionó en [el apartado de Estado del Arte: Protocolos](#), existen tres protocolos que pudieran ser alternativas a DP-3T con un grado similar de privacidad.

ConTra Corona

El primero de ellos es **ConTra Corona**. Este protocolo supone una combinación de metodología centralizada junto a descentralizada. Si bien la gestión de notificaciones de contagios es realizada a la inversa, subiendo cada usuario el listado de identificadores obtenidos en lugar de los propios, el funcionamiento es muy similar.

Inicialmente puede parecer un buen candidato, pero nos encontramos con dos principales inconvenientes.

- **Protocolo recientemente concebido.** Debido a que DP-3T junto a GAEN son los protocolos más implementados a nivel global, esta derivación similar de ellos ha sido todavía poco explorada. Existe documentación que ciertamente habla sobre su implementación en código, pero esta no se encuentra disponible en fuentes abiertas, o bien por materias relacionadas con la propiedad intelectual, o bien porque no haya llegado a implementarse en una aplicación real.

A causa de dichas carencias, implementar el algoritmo de cero, sin adecuadas referencias o documentación supervisada, extensa o útil, puede suponer un impacto contundente en la planificación, que debido a la limitación de tiempo no es posible.

- **Upload What You Observed.** El uso de paradigmas centralizados ha sido abiertamente criticado como se pudo ver en las anteriores opciones, [BlueTrace](#) y [PEPP-PT/PEPP](#). Si bien este protocolo no es completamente centralizado, pues delega ciertas funciones a otros sistemas, el hecho de recaer en el paradigma *Upload What You Observed* en vez de en *Upload What You Sent*, puede tener consecuencias negativas en cuanto a privacidad se refiere, como se vio en el apartado de [BlueTrace](#).

De igual modo que BlueTrace, el requisito [La confianza en el servidor debe ser limitada](#) queda vagamente cumplimentado, siendo un gran punto en su contra.

Debido a los motivos previamente mencionados, ConTra Corona queda **descartado**.

EpiOne

El siguiente protocolo a considerar es **EpiOne**. Es un protocolo novedoso que parece solucionar múltiples problemas que los anteriores presentaban. Por un lado presenta un paradigma basado en cardinalidad de intersección de conjuntos privados o *PSI-CA*. Esto permite averiguar si existe una intersección entre el conjunto de identificadores contagiados registrados en el servidor y los identificadores que ha recolectado un usuario concreto. Al informar únicamente sobre la existencia o no de dicha intersección, sin revelar datos sobre qué identificadores son los infectados, da solución al principal problema de DP-3T y GAEN, los cuales buscan coincidencias exactas entre los identificadores contagiados y los almacenados por el usuario.

Por otro lado, los servidores no son centralizados, poniendo solución al problema planteado por BlueTrace, PEPP-PT/PEPP o ConTra Corona. De este modo, EpiOne parece presentarse como un buen candidato.

Si bien la idea principal es buena, el protocolo todavía **careace de una implementación en código**, lo que implicaría desarrollarlo desde cero siguiendo la documentación que puede encontrarse sobre ello. [\[63\]](#) [\[64\]](#)

Pronto-C2

Otro protocolo que se ha desarrollado partiendo de la idea de DP-3T es **Pronto-C2**. Este emplea un protocolo descentralizado y utiliza el algoritmo de Diffie-Hellman para establecer la clave privada con la que trabaja, así como firmas digitales ciegas para autenticar a los usuarios. Además, todas las comunicaciones con el servidor se realizan mediante redes privadas, lo cual dificulta la interceptación de la información a la vez que anonimiza los datos enviados.

Este acercamiento resulta bastante atractivo pues, al igual que DP-3T, trabaja con intercambios de claves entre usuarios, las cuales se suben al servidor según el paradigma *Upload What You Sent*. Sin embargo, al igual que ocurre con EpiOne, **todavía no se ha realizado una implementación de este protocolo**. El agravante es que, además, la documentación referente a Pronto-C2 es mucho más escasa, incluso, que la que se puede encontrar de EpiOne.

De realizar este acercamiento, este debería ser desarrollado de cero y con muy escasa documentación. Dada la limitación de tiempo, tratar de realizar este protocolo con tan poca información disponible en fuentes abiertas, queda **descartado**. [\[11\]](#)

Decisión final

De entre lo anteriormente visto, se opta por la realización e implementación de un protocolo desde cero. Inicialmente se plantea el desarrollo de EpiOne por las facilidades que ofrece en cuanto a la privacidad.

Sin embargo, se produce un contratiempo debido a la escasez de documentación que se puede encontrar sobre este protocolo, así como a la inexistencia de un código que lo implemente y se pueda tomar como referencia.

Por lo tanto, se opta por la implementación de lo dictaminado por el protocolo inicialmente elegido, DP-3T, del cual hay mucha más información.

La diferencia con respecto al plan inicial reside en que, en lugar de implementar el protocolo mediante el código abierto que puede encontrarse en GitHub³, el cual recae en la API inaccesible, **se llevará a cabo una implementación desde cero del protocolo, siguiendo los algoritmos, pasos y paradigmas descritos por DP-3T**.

5.4. Implementación de los requisitos acorde a nuestra versión de DP-3T

Una vez se ha decidido el desarrollo de una versión propia del protocolo DP-3T, es necesario comprobar que este sea capaz de cumplir los requisitos estipulados en la fase de análisis y explicar cómo se llevará a cabo.

Implementación de los Requisitos No Funcionales

- **Rotación de identificadores.** Con el fin de incrementar la confusión y difusión, diariamente se generan un número fijo de identificadores que rotan cada cierto tiempo.

Implementación de los Requisitos Funcionales

De cara al usuario

- **Visualización de mi riesgo de exposición.** Se mostrará un panel que variará de color según el nivel de riesgo.
 - **Nulo.** Verde.
 - **Riesgo.** Amarillo.
 - **Contagiado.** Rojo.
- **Possibilidad de comunicación de contagio.** Cuando un usuario haya dado positivo, la autoridad sanitaria le proporcionará un código que este podrá introducir en la aplicación. Dicho código se enviará al servidor junto a las semillas generadoras de los identificadores del usuario y la fecha de PCR positiva o de inicio de síntomas. El servidor se encargará de corroborar que dicho código es válido y, de serlo, almacenar los datos enviados.
- **Notificación y muestra de la exposición a un contagio.** Se avisará al usuario de un posible contagio mediante una notificación en su *smartphone*. También dentro de la aplicación mediante el cambio de color del panel de riesgo de exposición.

³<https://github.com/DP-3T/>

De cara a otros dispositivos

- **Intercambio de identificadores.** Se llevará a cabo al transcurrir 15 minutos seguidos de contacto con otro dispositivo.
- **Permitir la identificación de otros dispositivos con la aplicación.** La aplicación detectará otros dispositivos cercanos que posean la aplicación activa también. El radio abarcado es de dos metros.
- **Medición del tiempo de exposición a un contacto.** Para medir el tiempo de exposición, DP-3T utiliza la atenuación de los paquetes de datos transmitidos mediante Bluetooth, de forma que si dicha atenuación se encuentra por encima de unos valores, se comenzará a contar el tiempo. La estimación se realiza utilizando un conjunto de beacons recibidos de un dispositivo concreto, los cuales se envían cada cierto tiempo (entre 2 minutos y medio y 5 minutos) a modo de cerciorarse de que los dispositivos permanecen en rango del otro.
- **Informar de un contagio.** De manera periódica el servidor emitirá mediante *broadcast* las semillas generadoras asociadas a identificadores contagiados que tiene almacenados. Cuando la aplicación recibe dichas semillas, genera los identificadores en local y los compara con los almacenados. En caso de producirse una coincidencia, la aplicación informa al usuario que ha estado en las cercanías de un individuo contagiado.
- **Contacto tras exposición a otro dispositivo.** A partir de los 15 minutos de exposición se considera contacto.
- **Cálculo de la distancia entre dispositivos.** Se considera distancia cercana cuando se detecta una atenuación inferior a 50dB. Con ello tenemos una muy alta certeza de que la distancia es inferior a dos metros. Para corregir discrepancias entre modelos de dispositivos móviles, al comienzo del encuentro se realiza una calibración. [5]

Al igual que en DP-3T, existirán los siguientes estados:[6]

- *Sin riesgo.* El usuario no ha estado en contacto con ningún usuario contagiado.
 - *Con riesgo.* La aplicación ha detectado un identificador contagiado entre sus almacenados, lo cual indica que el usuario ha estado en contacto cercano con algún usuario contagiado.
 - *Contagiado.* El usuario ha proporcionado un código de contagio válido al servidor, lo cual indica que ha obtenido positivo en una PCR, pues una autoridad sanitaria le ha cedido un código válido.
- **Envío de códigos de contagio al servidor de la autoridad sanitaria.** Para realizar el envío del código de contagio DP-3T utiliza un objeto de tipo GaenRequest. Este **objeto** es una adaptación de la petición básica del **protocolo HTTP** realizada por el **protocolo GAEN**. [7] Debido a la imposibilidad para acceder a un servidor propio de la autoridad sanitaria, la aplicación se orientará inicialmente a una red local. Para ello lo que se hará es enviar el paquete cifrado por la red a un puerto TCP concreto donde se llevarán a cabo las pruebas.
 - **Comunicación con el servidor para obtener lista de nuevos contagios.** El servidor emitirá de manera periódica el listado de todas las semillas generadoras de identificadores contagiados que posea. La aplicación cliente recibirá dicho listado y comparará lo recibido con lo almacenado para determinar si el usuario ha estado expuesto.

De cara a la propia aplicación

- **Cálculo de los identificadores.** Se calculan empleando el algoritmo de DP-3T explicado en la sección dedicada al [protocolo DP-3T](#).
- **Cálculo de estado de exposición.** El estado de exposición pasa a ser de riesgo en el momento en el que se reciben las semillas generadoras de un identificador recibido previamente, es decir, se ha estado en contacto con un usuario contagiado a menos de 2 metros. El estado de contagiado aparece en el momento en el que se envía un código de contagio al servidor y este es validado como tal. [6]

Implementación de los Requisitos de Información

Información almacenada en local

- **Identificadores anónimos propios de cada usuario.** Se usarán identificadores generados de manera pseudoaleatoria, como se ha explicado en el apartado dedicado a [DP-3T](#), con el fin de no proporcionar ningún dato personal del usuario.
- **Información anonimizada sobre qué usuarios han estado en contacto.** Se usarán identificadores efímeros que se intercambiarán entre usuarios que mantengan contacto.

Información almacenada en el servidor

- **Datos asociados a los identificadores infectados.** Se almacenarán las semillas generadoras de los identificadores, es decir la clave y fecha generadoras, y la fecha de recepción.
- **Fecha de recepción.** Se almacenará la fecha de recepción de los datos anteriores con el fin de eliminarlos a los 14 días.

Base de Datos. Modelo Físico

La arquitectura de este proyecto requiere la utilización de dos bases de datos, una situada en el servidor y encargada de gestionar los identificadores contagiados, y otra en los clientes de la aplicación, en local, encargada de almacenar tanto los identificadores propios como los obtenidos por intercambio.

Para su implementación es necesario llevar a cabo un análisis sobre qué gestores de bases de datos nos ofrecen las características óptimas para cada una de ellas.

Base de Datos Local

La base de datos local es aquella que se creará en las instancias cliente de la aplicación, en este caso los dispositivos Android de cada uno de los usuarios. En este caso, las alternativas que se han encontrado son:[45]

- **Oracle Berkeley DB.** Es un familia de productos que ofrece librerías para gestionar datos con un gran rendimiento y escalabilidad. Proporciona flexibilidad ya que se puede manejar o bien como una base de datos clave-valor o bien como una base de datos relacional cuando sea necesario. Su almacenamiento requiere de en torno a 1 MB como mínimo.

A pesar de aparentar ser una buena opción, dado que la base de datos local es muy sencilla, y no haremos uso de su escalabilidad y necesidad de alto rendimiento, es preferible buscar opciones que requieran de menor almacenamiento.

- **Interbase ToGo.** Se trata de un sistema gestor de bases de datos relacionales que requiere de mínimo 400 KB de almacenamiento. Es una base de datos SQL empotrada disponible para Android e iOS. Posee módulos propios que permiten integrar opciones offline a la aplicación, y también, eliminar la necesidad de implementar drivers de cliente que se conecten a la versión servidor de esta base de datos. Aunque es mucho menos pesada que Oracle Berkeley DB, su licencia es privada y no de dominio público.
- **SQLite.** Se define como un gestor de bases de datos relacional. La principal característica de este gestor es que no consta de una arquitectura cliente-servidor. En su lugar, se enlaza con el programa llegando a formar parte del mismo, ya que toda su funcionalidad está contenida en una biblioteca de código relativamente pequeña.

La principal ventaja de este gestor es su tamaño, ya que su tamaño mínimo es de tan solo 500 KB en memoria, pues se almacena como un fichero que la biblioteca bloquea o desbloquea automáticamente en el momento que sea necesario. Además su licencia es de dominio público y existe una amplia documentación sobre su uso en dispositivos Android, lo que facilita su implementación.

La decisión que se ha considerado más apropiada es **utilizar el gestor SQLite**, debido al poco espacio de almacenamiento que ocupa, las facilidades que ofrece a la hora de implementarse y tener licencia de dominio público.

La primera tabla almacenará aquella información relacionada con los identificadores propios.

Tabla ids_propios

- **id** Es el identificador de cada fila de la base de datos. Es de tipo **INTEGER** y se autogenera cada vez que se añade un registro. Simplifica el acceso ordenado a la base de datos.
- **identificador_ef** Se trata del identificador efímero empleado para referir a cada usuario de manera única y anónima. Es de tipo **TEXT**.
- **clave_gen** Es la clave generadora, miembro del par que conforma la semilla generadora del identificador efímero. Es de tipo **TEXT**.
- **fecha_gen** Es la fecha generadora, miembro del par que conforma la semilla generadora del identificador efímero. Es de tipo **TEXT**.

La segunda, refiere a aquellos identificadores efímeros obtenidos mediante intercambio BlueTooth.

Tabla ids_ajenos

- **id** Es el identificador de cada fila de la base de datos. Es de tipo **INTEGER** y se autogenera cada vez que se añade un registro. Simplifica el acceso ordenado a la base de datos.
- **identificador_ef** Se trata del identificador efímero empleado para referir a cada usuario de manera única y anónima. Es de tipo **TEXT**.
- **fecha_rec** Es la fecha de recepción del identificador efímero. Es de tipo **TEXT**. Se emplea para saber cuándo un identificador recibido por intercambio BlueTooth deja de ser contagioso y se pueda eliminar.

**Figura 11:** Modelo físico de la base de datos de los clientes

Base de Datos del Servidor

La base de datos del servidor es aquella que almacenará los identificadores de los usuarios infectados y a la cual tendrán acceso las autoridades sanitarias pertinentes.

Para su implementación se han analizado los principales gestores de bases de datos. Se han considerado como candidatos aquellos cuyas características mejor se adaptaban a la aplicación y a su desarrollo futuro. [43] [47] [17]

- **MySQL.** Es un sistema gestor de base de datos relacional de código abierto, considerado el más popular por una amplia mayoría de usuarios. Se utiliza principalmente para el desarrollo de páginas web, aunque su uso en software libre también está muy extendido.

En el caso de nuestra aplicación, los factores que nos han llevado a considerarlo un buen candidato son principalmente:

- **Facilidad de uso.** Al ser uno de los gestores más populares, la documentación, los usuarios y los ejemplos de uso son abundantes.
- **Buen rendimiento.** MySQL destaca por tener un buen rendimiento para bases de datos con una cantidad de datos no muy elevada.

Precisamente este último punto ha sido determinante y nos ha llevado a **descartarlo**, pues el objetivo de la aplicación es llegar al mayor público posible y por tanto la cantidad de datos sería elevada. [46]

- **MariaDB.** Este gestor de bases de datos es una derivación de MySQL, por lo que son completamente compatibles. Posee una gran escalabilidad y ofrece buena seguridad y velocidad a la hora de realizar transacciones. Además, es de código abierto, por lo que su licencia es de dominio público.

Frente a MySQL, su optimizador funciona mejor ante cargas complejas, poseyendo un mejor rendimiento. También ofrece una mayor usabilidad, pues aporta estadísticas de tablas, mejoras en comandos y mayor precisión en algunos tipos de datos, así como facilidades a la hora de realizar testeos. [42]

- **PostgreSQL.** Este gestor está optimizado para gestionar grandes volúmenes de datos, por lo que puede funcionar algo peor con cantidades de datos menores.

Posee una buena flexibilidad en cuanto a lenguajes de programación y es multiplataforma, por lo que puede adaptarse a múltiples proyectos. Además, dispone de una herramienta mucho más visual, **pgAdmin**, para gestionar las bases de datos.

Se caracteriza por ser robusta, eficiente y estable.

Sin embargo, optimizar su uso y recursos requiere de un mayor conocimiento del gestor. Además, dado que para los casos de prueba se emplearán volúmenes de datos menores, no funcionará de una manera tan optimizada como haría con grandes cantidades de datos.

Se elige, por tanto, **MariaDB** por las facilidades que ofrece tanto a nivel de testeo, como de documentación al tratarse de un gestor de código abierto.

Tabla ids_infectados

- **id** Es el identificador de cada fila de la base de datos. Es de tipo **SERIAL** y se autogenera cada vez que se añade un registro. Simplifica el acceso ordenado a la base de datos.
- **clave_gen** Es la clave generadora, miembro del par que conforma la semilla generadora del identificador efímero. Es de tipo **VARCHAR**.
- **fecha_gen** Es la fecha generadora, miembro del par que conforma la semilla generadora del identificador efímero. Es de tipo **DATE**.
- **fecha_rec** Es la fecha de recepción del par clave y fecha generadoras. Es de tipo **DATE**. Se emplea para saber cuándo un par clave-fecha generadoras deja de ser contagioso y se pueda eliminar.

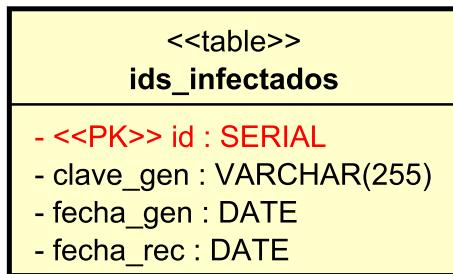


Figura 12: Modelo físico de la base de datos del servidor

Tecnologías utilizadas

Las tecnologías, y sus correspondientes versiones, empleadas en este proyecto, son las siguientes:

- **Java.** JDK8
- **Android.** Versión 10
- **MariaDB.** Versión 10.5.9
- **SQLite.** Versión 3.28

5.5. Implementación de los Requisitos de Usabilidad

Una vez han sido definidos, se debe pensar la manera de implementarlos en la aplicación a desarrollar.

Veamos, entonces, la propuesta que se ha diseñado para cada aspecto de usabilidad y accesibilidad destacado en la sección de *Análisis*. Los requisitos que debe cumplir el diseño de la interfaz son los siguientes:

- **La interfaz debe ser sencilla.** La funcionalidad de la aplicación se condensará en tres pantallas, principal, información y ajustes de idiomas, sin una navegación entre menús excesiva.
- **La interfaz debe ser suficientemente intuitiva.** Se usará simbología fácilmente identifiable y utilizada universalmente en la mayoría de aplicaciones del mercado.
- **La interfaz debe ser agradable a la vista.** Para ello se utilizarán colores suaves. Se buscará que sean gamas de colores afines, evitando contrastes fuertes o visualmente agresivos.
- **Accesibilidad para personas con daltonismo.** Se utilizará un [simulador de daltonismo](#) para ajustar los colores de forma que estos sean lo suficientemente diferenciables para personas con trastornos visuales tales como [protanopia](#), [deuteranopia](#) y [tritanopia](#).
- **El texto debe ser claro y conciso.** Se evitará el uso de tecnicismos así como de palabras redundantes con el fin de hacerlo más fácil de entender a un mayor número de personas.
- **Legibilidad para todas las edades y capacidades visuales.** Se emplearán tipografías claras y sencillas, así como tamaños de letra lo suficientemente grandes. En el caso de que esto no sea posible, ya sea por el tamaño del botón o por el espacio en la pantalla, se emplearán símbolos visuales para complementar el concepto referenciado.
- **La interfaz debe invitar a publicitar su uso.** Se dibujará a Aga y Gava, que actuarán como mascotas de la aplicación, para hacerla más distingüible.
- **La interfaz debe concienciar sobre los síntomas del estado de exposición del usuario.** Dependiendo del estado, sin riesgo, con riesgo o contagiado; Aga y Gava, así como los colores del botón de recomendaciones, aparecerán de un modo u otro.
 - **Sin riesgo.** Aga y Gava aparecerán felices.
 - **Con riesgo.** Aga y Gava aparecerán tomando precauciones y en cuarentena.
 - **Contagiado.** Aga y Gava aparecerán con un termómetro y en una cama siendo cuidados por el enfermero Donehre, otro personaje.
- **Internacionalización.** Se incluirá un apartado de ajustes de idioma para facilitar la accesibilidad a personas con distintas lenguas.

5.6. Diseño de la Interfaz e Implementación de los Requisitos

Esta aplicación está dirigida a un espectro de usuarios muy amplio. Esto impone que la interfaz de usuario sea muy intuitiva y sencilla, con el fin de que un usuario sin experiencia pueda hacer uso de ella con facilidad.

El elemento a destacar en la interfaz de la aplicación es el menú de navegación.



Figura 13: Menú de navegación

Consta de tres botones con amplia superficie para garantizar la máxima precisión a la hora de seleccionar cada uno. Además, se ha cuidado de que únicamente se implementen las funcionalidades necesarias, dejando de lado elementos superfluos. En orden de izquierda a derecha son:

- **Acceso a pantalla principal.** Nos permite ingresar en la pantalla principal de la aplicación cuando nos encontremos en cualquier otra pantalla, salvo en el formulario de comunicación de contagio.
- **Acceso a pantalla de información.** Nos permite ingresar en la pantalla de información cuando nos encontremos en cualquier otra pantalla, salvo en el formulario de comunicación de contagio.
- **Acceso a pantalla de cambio de idioma.** Nos permite ingresar en la pantalla de cambio de idioma cuando nos encontremos en cualquier otra pantalla, salvo en el formulario de comunicación de contagio.

A continuación procederemos a explicar brevemente cada diseño preliminar de las respectivas pantallas. En las imágenes se señala con una flecha en cuál de los botones del menú inferior se encuentra.

5.6.1. Pantalla Principal

La pantalla principal cuenta con un título además de las siguientes áreas:

- **Pantalla Riesgo de Contagio.** Aquí se indicará el nivel de riesgo de contagio del usuario. Dependiendo del nivel de riesgo el color de este recuadro cambiará:
 - **Verde.** El usuario no ha estado en contacto con ningún individuo contagiado, ende el riesgo de contagio **no existe**.
 - **Naranja.** El usuario ha estado cerca de algún individuo contagiado, ende posee **riesgo de contagio**.
 - **Rojo.** El usuario ha enviado un código proporcionado por una entidad sanitaria y el servidor lo ha validado, ende estando **contagiado**.

Como hemos detallado en los [requisitos de usabilidad de la aplicación](#) los colores elegidos son suaves y poco impactantes.

- **Botón Comunica tu contagio.** Si el usuario desea comunicar su contagio, ha de pulsar este botón. Cuando lo haga aparecerá una pantalla como la siguiente:



Figura 14: Pantalla Principal

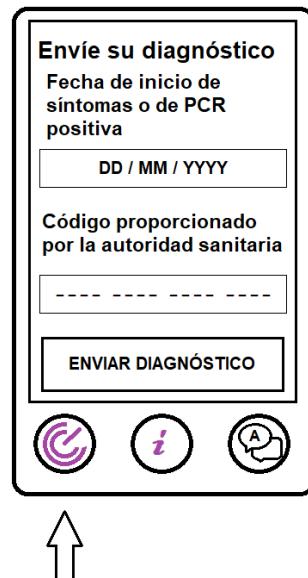


Figura 15: Botón Comunica tu contagio

Esta pantalla incluye dos cuadros, el primero deja introducir la fecha de inicio de síntomas o PCR positiva, el segundo, el código proporcionado por la autoridad sanitaria, el cual seguirá un patrón con el fin de evitar envío de diagnósticos falsos.

Una vez pulsado el botón de *Enviar diagnóstico*, la aplicación procede a mostrar el siguiente cuadro confirmativo.

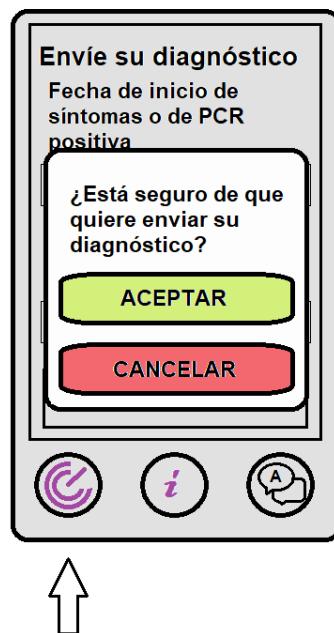


Figura 16: Confirmación del envío

Los colores elegidos para los botones de *Aceptar* y *Cancelar* son verde y rojo suaves para minimizar el impacto visual. Además, son bastante intuitivos y para los tres tipos de daltonismo más habituales, protanopia, deuteranopia y tritanopia, se mantiene una diferenciación suficiente entre los colores.^[67]

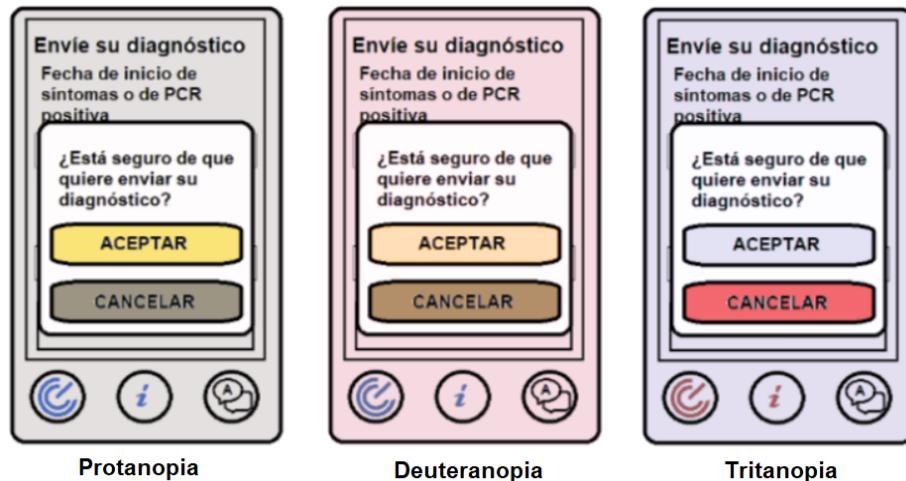


Figura 17: Simulación de los distintos tipos de daltonismo

5.6.2. Pantalla de Información

La pantalla de información contiene una pequeña presentación de la aplicación, así como donde se ubicaría la política de privacidad, para que esta pueda ser consultada en cualquier momento.



Figura 18: Pantalla de información

5.6.3. Pantalla de Ajustes de Idioma

En la pantalla de ajustes de idioma aparecen una serie de botones de selección del idioma en el que se quiere poner la aplicación. Tras seleccionarlo hay que pulsar el botón de aceptar, con el fin de confirmar la acción para evitar que el usuario cometa un error.



Figura 19: Pantalla de idiomas

¿Quiénes son Aga y Gava?

Aga y Gava son dos dragoncitos, pero los nombres provienen de *agaporni* y *gaviota*, respectivamente.

Aga fue creada a inicios de la carrera por María Ruiz Molina y ha sido su marca en diversos trabajos de asignaturas, así como proyectos individuales.

Gava fue creado por Juan Velázquez García como intento de dibujar a Aga. Su nombre proviene de que en su primera versión, la cual fue un intento de dibujar a Aga, su pelo parecía una gaviota. El diseño ha evolucionado perdiendo la forma de pico de gaviota a un pelo más refinado.

Posteriormente Gava se añadió al universo de Aga junto a otros tantos personajes que se crearon, varios de ellos a modo de avatares o *agatares* de amigos del grupo de la facultad.

Si bien la idea final de estos personajes es incorporarlos a futuro como parte de un videojuego o historietas cómicas, debido a su simpleza y lindo diseño se ha decidido incluirles también en este trabajo a modo de mascotas y marca personal.

Como anécdota, el primer trabajo universitario realizado conjuntamente por ambos autores incluyó también a Aga y Gava, en el primer cuatrimestre de segundo de carrera, y desde entonces Aga ha acompañado prácticamente todas las entregas de María Ruiz Molina.

5.7. Implementación de la aplicación

5.7.1. Implementación final de la interfaz

A la hora de implementar la interfaz en la aplicación, se optó por usar colores distintos de los de Radar Covid. Así, se cambió la paleta de colores de morado a azules suaves.

Respetando los requisitos anteriores, la aplicación queda con la siguiente interfaz:

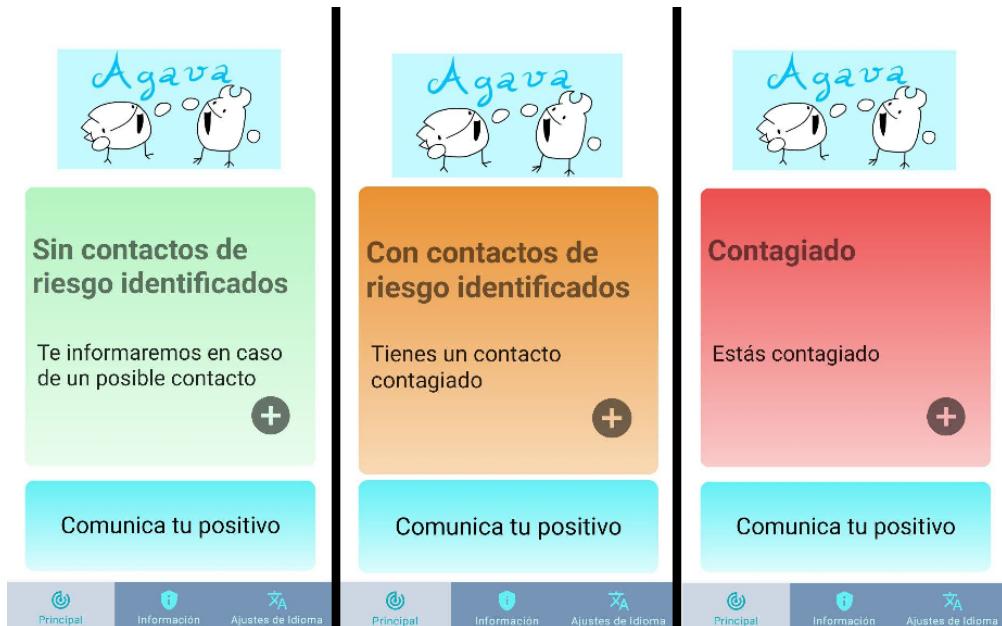


Figura 20: Pantalla de inicio. Colores de la aplicación

Como puede verse, los colores siguen siendo lo suficientemente diferenciados, permitiendo la visualización de los iconos del menú de abajo, así como la legibilidad.

En cuanto a las pruebas de daltonismo, los colores del menú de abajo siguen contrastando lo suficiente. Los colores del nivel de alerta puede que se vean peor en ciertos casos, pero al ir acompañados de un mensaje sobre el estado de contagio, se compensa el menor contraste entre colores asociados a estados.



Figura 21: Pantalla de inicio. Protanopia

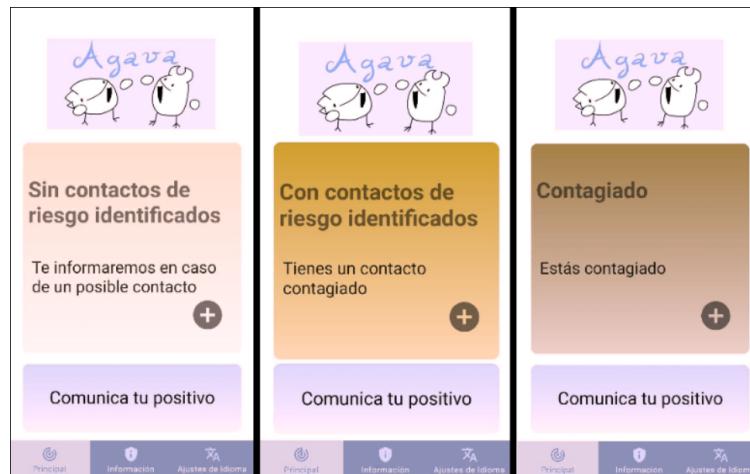


Figura 22: Pantalla de inicio. Deuteranopia

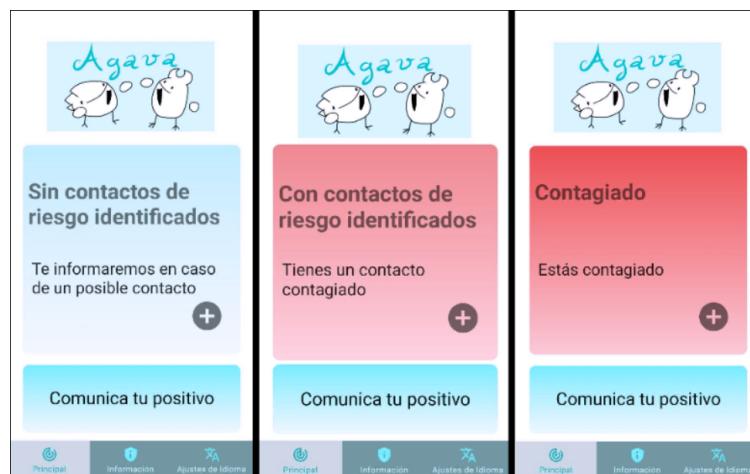


Figura 23: Pantalla de inicio. Tritanopia

5.7.2. Conexiones de red TCP.

En la elaboración del protocolo, así como de las conexiones que realizará el cliente con el servidor, los envíos de códigos contagiados se harán por red empleando el protocolo TCP. De este modo, nos aseguramos de que el orden de envío de los paquetes se mantenga, así como evitar la pérdida de estos. Esto es importante, pues el código solo va a transmitirse una vez por red.

5.7.3. Conexiones de red UDP.

Por otro lado, el servidor enviará identificadores contagiados de manera periódica. Este envío se realiza a todos los clientes, por lo que se trata de un multicast, donde los clientes pertenecen a un grupo concreto, definido con una dirección IP de grupo de multicast.

Debido al uso de multicast el protocolo empleado es UDP, pues la conexión no se realiza solamente entre dos dispositivos, sino que es de un servidor a todos los clientes.

5.7.4. Decisión sobre Bluetooth

El hecho de desarrollar una versión del protocolo DP-3T desde cero, aunque este utilice [Bluetooth Low Energy](#), plantea un dilema sobre qué tipo de protocolo Bluetooth es más aconsejable.

Durante el estudio realizado con anterioridad sobre protocolos de rastreo de contactos, se encontró que BLE era una tecnología que ofrecía como principal ventaja la eficiencia energética, pero a cambio necesitaba permisos de geolocalización para poder funcionar. Debido a la orientación hacia la privacidad de este proyecto, esta ventaja debía ser lo suficientemente considerable como para justificar su uso.

Además, debido a que BLE funciona por broadcast, enviando la MAC Bluetooth, el UUID e información sobre el servicio abierto, los mensajes y esta información pueden ser interceptados con mayor facilidad. Bluetooth, por otro lado, crea canales seguros mediante el pareado de dispositivos. De este modo se realiza un intercambio de clave y el canal se cierra a agentes externos. [58]

Tras muchas búsquedas, se encontraron estudios sobre sus diferencias, aunque los resultados no proporcionaban información sobre la disparidad de gasto energético en cuanto a tiempo de uso de la batería.

Por esta razón, se decidió realizar un estudio propio de forma rápida, para comprobar cuánta batería puede llegar a consumir Bluetooth en un periodo de tiempo amplio.

Para ello, se conectaron unos cascos inalámbricos a un dispositivo móvil y se reprodujeron pistas de audio durante 1 hora. Tras este tiempo, se comprobó el porcentaje aproximado de consumo de batería que ofrece el sistema. El valor obtenido fue un consumo de batería aproximado de un 2% en 1 hora de reproducción de audio. Si extrapolamos a la cantidad de datos que se van a intercambiar con esta aplicación, por supuesto mucho menor, podemos concluir que la diferencia en cuanto a consumo de energía entre Bluetooth y BLE no justifica el solicitar permisos de geolocalización al usuario.

Es por ello que esta aplicación utilizará el protocolo Bluetooth.

5.7.5. Cifrado de los datos

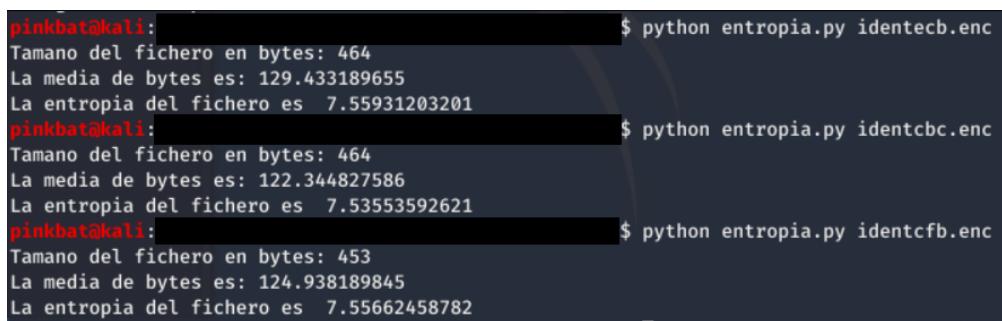
El cifrado de los datos se llevará a cabo durante la transmisión TCP del cliente al servidor, para, de este modo, evitar la interceptación o escucha de los identificadores contagiados y del código de contagio proporcionado por la autoridad sanitaria.

Para ello, el servidor envía al cliente su clave pública. Con ella, el cliente cifraría una clave simétrica AES-256 y se la enviaría al servidor. Una vez que ambos poseen la clave simétrica, el cliente enviaría el paquete con el código e identificadores contagiados. De este modo, solo el servidor podría descifrar el mensaje.

Tras dicho intercambio, se ha optado por un cifrado de clave simétrica de los datos debido a su menor complejidad computacional.

Además, el tipo de cifrado usado se ha determinado calculando la [entropía](#) generada tras cifrar un mensaje formado por código y los identificadores a enviar.

En la siguiente imagen pueden verse los resultados obtenidos con los tipos de cifrado, en orden, ECB (*Electronic Codebook*), CBC (*Cipher Block Chaining*) y CFB (*Cipher Feedback*):



```
pinkbat@kali: ~ $ python entropia.py identecb.enc
Tamano del fichero en bytes: 464
La media de bytes es: 129.433189655
La entropia del fichero es 7.55931203201
pinkbat@kali: ~ $ python entropia.py identcbc.enc
Tamano del fichero en bytes: 464
La media de bytes es: 122.344827586
La entropia del fichero es 7.53553592621
pinkbat@kali: ~ $ python entropia.py identcfb.enc
Tamano del fichero en bytes: 453
La media de bytes es: 124.938189845
La entropia del fichero es 7.55662458782
```

Figura 24: Resultado de entropías

Si bien el tipo de cifrado que **mejor entropía obtiene es ECB, debido a su funcionamiento es descartado**. Al tratarse de un cifrado que a iguales bloques da iguales resultados, es susceptible de ataques de repetición. Esto se debe a que emplea la misma clave para cifrar cada bloque, y de haber dos iguales, el resultado sería el mismo.

Es por ello que se escoge CFB, por ser el siguiente con mejor entropía, además de estar orientado a cifrado de textos.

5.7.6. Adaptaciones realizadas cara al prototipo

Debido a la complejidad de la aplicación, así como del protocolo a implementar, y las restricciones de tiempo, se han realizado simplificaciones en algunos aspectos del desarrollo de la aplicación, llevando a cabo un prototipo que implementa todas las funciones pero con algunos cambios.

Estos o bien facilitan el seguimiento de las pruebas y análisis posteriores, o bien facilitaron la programación de algunos aspectos, permitiendo cuadrar los tiempos dentro de la planificación. Las adaptaciones son las siguientes.

- **Intercambio activo de los identificadores.** Para un mayor control de las pruebas, se ha implementado un botón en la aplicación prototipo. Al pulsarlo se realiza el envío de los identificadores vía Bluetooth a los dispositivos pareados.
- **Envío activo de ruido.** Para evitar postergar en la planificación las fases posteriores a la implementación de la aplicación, se ha prescindido de esta funcionalidad en este prototipo.
- **Generación y rotación manual de los identificadores.** Debido a la planificación prevista, no ha sido posible implementar un contador que generase y gestionase los identificadores en segundo plano. Para la realización de las pruebas, en caso de necesitar esta funcionalidad, se simula la rotación empleando diferentes versiones, cada una con un identificador distinto.
- **Cambio manual de estado de *Contagiado* a *Sin contactos* tras 14 días.** Debido al tiempo del que se disponía, se decidió realizar el cambio de estado tras 14 días de *Contagiado* a *Sin contactos* cada vez que se reiniciase la aplicación. De otro modo, debería mantenerse la aplicación activa en todo momento en segundo plano, para que pasados los 14 días cambiase el estado. Otro modo sería realizarlo tan solo al abrir la aplicación, comprobando la fecha, pero esto falsearía realmente el borrado tras 14 días, pudiendo ser más de no abrirse la aplicación transcurrido exactamente ese tiempo.
- **Eliminado manual de los identificadores con fecha extinta.** Debido al tiempo del que se disponía, se decidió realizar el borrado en el servidor de manera manual desde la consola de MariaDB, mediante el comando:

```
DELETE FROM ids_infectados WHERE CURDATE() - fecha_rec > 14;
```

- **Intensidad de la señal de Bluetooth predeterminada.** Debido al tiempo del que se disponía, se decidió dejar la distancia predeterminada, pues para realizar los casos de prueba y el análisis de riesgos de seguridad y privacidad, esta no iba a influenciar.
- **Pareado manual previo al intercambio de identificadores vía Bluetooth.** El pareado se realiza desde Ajustes del teléfono. Tras varios intentos sin éxito de programar el pareado para que la aplicación lo realizase automáticamente, se decidió no dedicar más tiempo a esto para evitar afectar a la planificación.

Ya que el realizar el pareado dentro o fuera de la aplicación no era algo fundamental para poder realizar las pruebas de intercambio de identificadores, se decidió hacer de manera manual.

- **Simplificación del proceso de cifrado de los datos que viajan por red.** La idea original es que, con cada intercambio de información entre cliente y servidor, se realice el siguiente cifrado. El servidor poseerá un par clave pública-privada. A su vez, el cliente generará una clave simétrica AES256. Cuando el cliente realice una conexión TCP con el servidor, este le enviará al cliente su clave pública, con la que el cliente podrá cifrar la clave simétrica para enviársela al servidor y así comenzar a intercambiar la información cifrada.

En el prototipo de la aplicación desarrollado, se establecerá manualmente una clave simétrica entre servidor y cliente para realizar las pruebas y análisis. Esta clave será constante y no viajará por la red.

Al trabajar con el modo de cifrado CFB, que necesita de un Vector de Inicialización, este también estará fijado, para asegurar mismos resultados en ambos lados de la aplicación, cliente y servidor. De nuevo, esto es una medida tomada con el fin de simplificar el proceso de cifrado en el prototipo.

- **No retroalimentación sobre si el envío del código es o no correcto.** El servidor no enviará al cliente un mensaje de retroalimentación sobre si el código es o no es correcto. Esta funcionalidad, si bien sería clave cara a una aplicación con una buena usabilidad, se ha prescindido cara al posterior análisis de riesgos de seguridad y privacidad.
- **No implementación de la notificación al usuario.** Dado que no es una funcionalidad fundamental para el prototipo, se ha prescindido de que la aplicación avise al usuario mediante una notificación cuando haya un cambio de estado.

6. Casos de prueba

El objetivo de las pruebas software es comprobar que la aplicación cumple con los requisitos que se han dictaminado en la *Fase de Análisis*.

Aunque se pueden clasificar de diversas formas, las principales clases de prueba son pruebas de caja negra y pruebas de caja blanca. Como añadido, se pueden categorizar dependiendo de qué aspecto se está probando.

En este caso, se han clasificado en pruebas de caja negra y caja blanca, y dentro de estas funcional y no funcional. Las pruebas englobarán pruebas de seguridad, de disponibilidad, de interacción y de comunicación Bluetooth, UDP y TCP.

6.1. Pruebas de caja negra

Las pruebas de caja negra son aquellas que se realizan sin saber la especificación de cómo se ha hecho aquello que se prueba, en otras palabras, solo nos interesa ver qué salidas o *outputs* producen las entradas o *inputs* que se introducen en el software.

Las pruebas se consideran correctas cuando se obtiene el resultado esperado, cumpliendo los requisitos previamente definidos. En caso contrario, se detallan los errores obtenidos y cómo se solucionaron hasta obtener el resultado deseado.

6.1.1. Intercambiar un ID entre dos dispositivos vía BT en rango

- **Tipo de prueba: Prueba funcional**
- **Ámbito de la prueba: Comunicaciones Bluetooth**

Se inician dos dispositivos móviles y se activa Bluetooth en ambos. Después, se parean de forma manual a través del menú de Ajustes de cada dispositivo.

En un principio, este proceso previo se iba a realizar de forma automática en la aplicación. La idea inicial era hacerlo mediante código y sin pareado. Se consiguió la detección de los dispositivos, pero la aplicación se cerraba en cuanto ocurría.

Siguiendo la [metodología escogida](#), tras varios intentos sin éxito se tomó la decisión de realizar este proceso pareando los dispositivos y desvinculándolos al terminar la operación, pues investigando, se encontró que la comunicación mediante dispositivos Bluetooth pareados se realiza mediante un canal cerrado.[\[13\]](#)

Finalmente, al obtener los mismos resultados, se decidió optar por realizar el pareo de forma manual.

Se inicia la aplicación y se presiona el botón que se ha habilitado para realizar las pruebas Bluetooth.

Los primeros resultados fueron negativos. Los dispositivos conectaban pero la aplicación se cerraba al instante.

Esto era debido a un mal uso de las funciones Bluetooth, en concreto *cancelDiscovery()*, pues se situó en un lugar erróneo del código.

Posterior a ello, debido a cambios anteriores, había quedado una conexión insegura de Bluetooth, realizada mediante `listenUsingInsecureRfcommWithServiceRecord` y `createInsecureRfcommSocketToServiceRecord`. Debido a esto se producía una incoherencia, pues ese tipo de conexión permite conectar dispositivos sin previo pareado, cuando la aplicación funciona mediante dispositivos pareados.

Tras realizar la conexión en modo seguro de nuevo, uno de los dispositivos recibió el mensaje con los identificadores de manera correcta, manteniendo la aplicación abierta.



Figura 25: Intercambio de ID entre dispositivos

6.1.2. Intercambiar un ID entre dos dispositivos vía BT en el límite del rango

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones Bluetooth

Se inician dos dispositivos móviles con Bluetooth activado. Tras ello se realiza el pareado manual a través del menú de Ajustes del teléfono.

A continuación, se buscó el límite del rango que alcanza la señal de Bluetooth.

Inicialmente se calculó mal y se realizó desde una distancia más cercana. Se fue buscando el punto límite hasta que se perdía la señal.

Una vez encontrado el punto límite se realizó el envío de un identificador.

En el dispositivo que actuaba como servidor apareció el mensaje de *Conectado*, pero no llegó el identificador.

Este resultado es lógico, pues la recepción de la conexión ocupa menos slots que el propio identificador. Esto se debe a que el mensaje se divide en múltiples slots que son enviados y por lo tanto la pérdida de uno es más probable, haciendo que el mensaje ya no llegue completo y correctamente.^[59]

6.1.3. Intercambiar un ID entre dos dispositivos vía BT fuera de rango

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones Bluetooth

Se inician dos dispositivos con Bluetooth activado. Estos se parean manualmente desde la sección de Ajustes del sistema.

Tras ello, se inicia la aplicación, y estando los dispositivos lo suficientemente alejados, se realiza un envío. Como era de esperar, no se recibe ningún tipo de señal.

6.1.4. Intercambiar un ID entre dos dispositivos vía BT y desconectarse justo en el momento del envío

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones Bluetooth

Se inician dos dispositivos con Bluetooth activado. Se parean desde Ajustes del teléfono de manera manual.

Tras ello, se inicializa la aplicación y se pulsa el botón de envío de identificador.

En el momento en el que en el dispositivo que actúa como servidor aparece el mensaje de *Conectado*, se desactiva Bluetooth del dispositivo que actúa como cliente.

Como era de esperar, la conexión se interrumpe, no llegándose a enviar el mensaje con el identificador, por lo que el servidor no recibe la información.

6.1.5. Intercambiar un ID entre dos dispositivos vía BT y desconectarse justo 1 segundo antes del momento del envío

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones Bluetooth

Se inician dos dispositivos con Bluetooth activado. Se parean desde Ajustes del sistema de manera manual.

Tras ello, se inicializa la aplicación y se pulsa el botón de envío de identificador.

Inicialmente aparecen los mensajes de creación de los sockets, mensajes dispuestos a modo de comprobar su correcto despliegue en la aplicación prototipo.

Una vez ambos dispositivos han creado el socket correspondiente para comunicarse entre sí, se desconecta Bluetooth.

Debido a ello, como era de esperar, se obtiene el mensaje de *Conexión fallida*, pues no se llega a establecer la conexión entre dispositivos.

6.1.6. Intercambiar un ID entre dos dispositivos vía BT y desconectarse justo 1 segundo después del momento del envío

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones Bluetooth

Se inician dos dispositivos con Bluetooth activado.

Tras ello se parean de manera manual mediante la opción Ajustes del teléfono.

Inicialmente aparece el mensaje de *Conectado*. Tras ello el de mensaje recibido, con el identificador.

Se desconecta Bluetooth tras ello, y como era de esperar, el identificador se almacena correctamente. Esto es porque el mensaje ya se recibió previamente a la desconexión, en el momento en el que sale un aviso en la pantalla de la aplicación.

6.1.7. Uso de la aplicación sin conexión a BT

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Disponibilidad

Se abre la aplicación y nada más ejecutarse sale el siguiente aviso en pantalla:

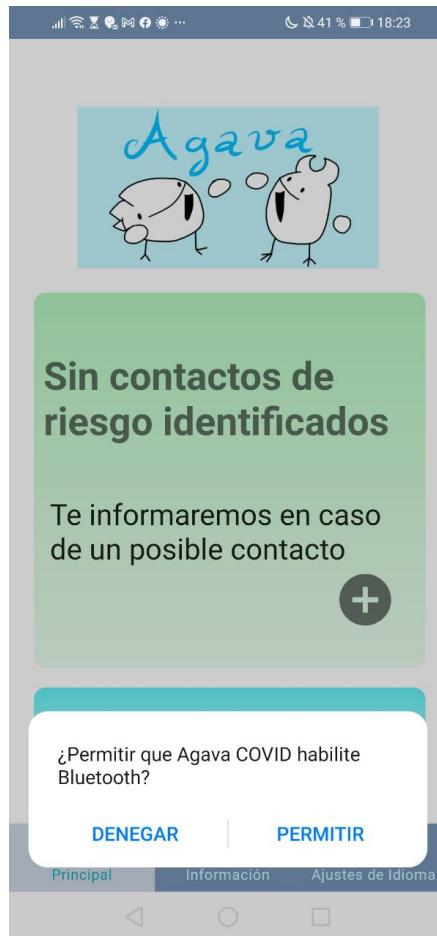


Figura 26: Solicitud para activar Bluetooth

De este modo, la aplicación nos comunica que el dispositivo no tiene activado Bluetooth.

Tras este aviso, pulsamos *Denegar* para que la aplicación siga funcionando sin Bluetooth. Tras ello, la aplicación funciona sin problemas.

Permite las conexiones con el servidor vía Internet, es decir, pueden enviarse códigos de contagio al servidor y recibir el multicast sin problemas.

El funcionamiento es el esperado, pues aquellas funcionalidades relacionadas con Bluetooth dejan de poderse ejecutar al no activarlo.

Por otro lado, aquellas que no necesitan de Bluetooth, funcionan correctamente.

6.1.8. Envío de un código correcto al servidor. (Código correcto: el pedido por la aplicación, otorgado por la autoridad sanitaria.)

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones TCP

Se abre la aplicación estando el dispositivo conectado a Internet.

Tras ello, dentro de la aplicación se selecciona *Comunica tu positivo* y se rellenan los datos. La fecha solo permite seleccionar aquellas fechas entre la actual y 14 días atrás. El código solo permite introducir doce caracteres numéricos, formato de los códigos de contagio.

Inicialmente, el caso de prueba tenía un error, pues permitía enviar claves y fechas generadoras al servidor sin la necesidad de llenar previamente los datos de fecha y código. De este modo se permitía el envío de claves y fechas generadoras sin comprobación, pudiendo enviar claves y fechas generadoras de una persona no contagiada como si lo estuviera.

El problema se solucionó obligando a que el envío recogiera la información introducida en esos campos.

De este modo, la información enviada por la red recoge tanto la fecha, como el código, como la lista de claves y fechas generadoras de identificadores obtenida de la base de datos local.

Inicialmente hubo un problema, pues el servidor no recibía información. Esto fue porque se abrían puertos no habituales, y por lo tanto, el firewall del ordenador empleado para las pruebas impedía la llegada de los paquetes generados por el dispositivo cliente, el móvil.

Se desactivó el firewall para realizar la recepción con el fin de finalizar el caso de prueba, y efectivamente se recibió la información en el servidor sin problemas.

Una vez el servidor recibe esta información, comprueba que el código sea uno de los generados por la autoridad sanitaria, y por tanto correcto y activo. De ser así introduce a la base de datos las claves y fechas recibidas. La comprobación la realiza comparándolo con un listado de códigos que tiene almacenado en un *Array*. En caso de producirse una coincidencia, se permite la inserción.

Una vez recibido, se cambia el valor de la variable que define el estado de contagio. Con ello, la imagen y los textos de la pantalla principal cambian al estado *Contagiado*.

Inicialmente, para que este cambio fuese visible en la interfaz, el usuario debía reiniciar la aplicación. Este error se solucionó haciendo que la aplicación se reiniciara automáticamente cada vez que cambiase el estado.

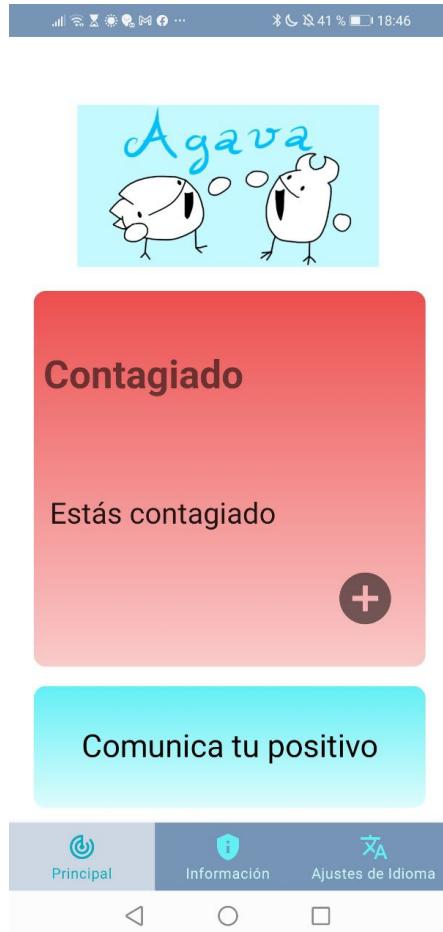


Figura 27: Aplicación con contagio

6.1.9. Envío de un código incorrecto al servidor

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones TCP y Usabilidad

Se abre la aplicación estando el dispositivo conectado a Internet.

Tras ello, dentro de la aplicación se selecciona *Comunica tu positivo* y se rellenan los datos. La fecha solo permite seleccionar aquellas fechas entre la actual y 14 días atrás. El código solo permite introducir doce caracteres numéricos, formato de los códigos de contagio. De este modo se impide la inserción de un código estructurado de manera incorrecta.

Una vez el servidor recibe esta información, comprueba que el código sea uno de los generados por la autoridad sanitaria, y por tanto correcto y activo. De ser así introduce a la base de datos las claves y fechas recibidas. La comprobación la realiza comparándolo con un listado de códigos que tiene almacenado en un *Array*. En caso de producirse una coincidencia, se permite la inserción. En este caso no se produce ninguna coincidencia, por lo que rechaza la información recibida y no la introduce en la base de datos.

Este caso de prueba queda parcialmente incompleto, pues el cliente no recibe una retroalimentación sobre si el código es o no uno de los aceptados por el servidor; pero el servidor actúa como se esperaba, rechazando los identificadores recibidos y sin almacenarlos en la base de datos.

6.1.10. Envío de diagnóstico sin fecha

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones TCP y Usabilidad

Se realiza el mismo proceso que en el anterior caso y se rellena únicamente el campo del código con uno correcto, dejando el de fecha vacío.

A continuación se pulsa en aceptar y se confirma el envío. El mensaje se envía sin problema y nos proporciona el resultado esperado, que es la recepción del mensaje en el servidor.

6.1.11. Envío de diagnóstico sin inserción de código

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Usabilidad

El proceso para llegar a la pantalla de envío es el mismo que en el caso anterior. Una vez se llega al formulario de *Comunica tu positivo* se pulsa el botón de *Aceptar* para enviar el código. Dado que no se han introducido datos, aparece un mensaje que indica que el código está incompleto.



Figura 28: Aviso sobre código incompleto

6.1.12. Envío de código con menos de 12 cifras

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Usabilidad

El proceso para llegar a la pantalla de envío es el mismo que en el caso anterior. Una vez se llega al formulario de *Comunica tu positivo* se escoge una fecha en el rango y se rellena el campo del código con 11 cifras. Tras ello se pulsa sobre aceptar.



Figura 29: Aviso sobre código incompleto

El resultado es el esperado, la aparición de un mensaje que nos indica que el código está incompleto y que no permite avanzar a la siguiente pantalla para aceptar el envío.

6.1.13. Envío de código con más de 12 cifras

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Usabilidad

Se realiza el mismo proceso que en el caso anterior y se intenta insertar un código de 13 cifras. El campo del código nos impide superar las 12 cifras, por tanto el resultado es el esperado.

6.1.14. Envío de código con caracteres no numéricos

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Usabilidad

Se realiza el mismo proceso que en el caso anterior y se intenta insertar un código con caracteres alfabéticos y/o símbolos. El propio campo de la aplicación cliente no acepta ese tipo de caracteres, por lo que la escritura de ellos no se admite.

6.1.15. Envío con fecha anterior a 14 días

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Usabilidad

Se abre la aplicación estando el dispositivo conectado a Internet.

Tras ello, dentro de la aplicación se selecciona *Comunica tu positivo* y se rellenan los datos. La fecha solo permite seleccionar aquellas fechas entre la actual y 14 días atrás. Con esta restricción se impide el envío de identificadores con fecha inválida al servidor, lográndose el caso de prueba correctamente.



Figura 30: Límite anterior de aceptación de fechas

6.1.16. Envío con fecha posterior a 14 días

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Usabilidad

Al igual que en el caso anterior, la aplicación cliente solo permite la selección de fechas 14 días anteriores a la actual. Con ello se impide el envío de identificadores con fechas futuras.



Figura 31: Límite posterior de aceptación de fechas

6.1.17. Multicast de IDs infectados desde el servidor a los clientes

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones UDP

Para realizar el envío de identificadores, se activa el servidor. De manera periódica realiza un envío multicast del contenido de su base de datos, el cual es las claves y fechas generadoras asociadas a identificadores contagiados.

Inicialmente el envío no funcionaba correctamente, pues se empleaba una dirección IP para multicast fuera del rango de las reservadas para ello (es decir, las comprendidas entre la 224.0.0.0 hasta la 239.255.255.255).

Una vez se asignó una dirección de multicast correcta, se pudo comprobar empleando un *sniffer* de paquetes (*Wireshark*), que el envío y la petición de unión a dicha dirección se realizaba correctamente y de manera periódica.

6.1.18. Uso de la aplicación sin conexión a Internet (y sin recepción de multicast)

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Disponibilidad

Se abre la aplicación y nada más ejecutarse sale el siguiente aviso en pantalla:

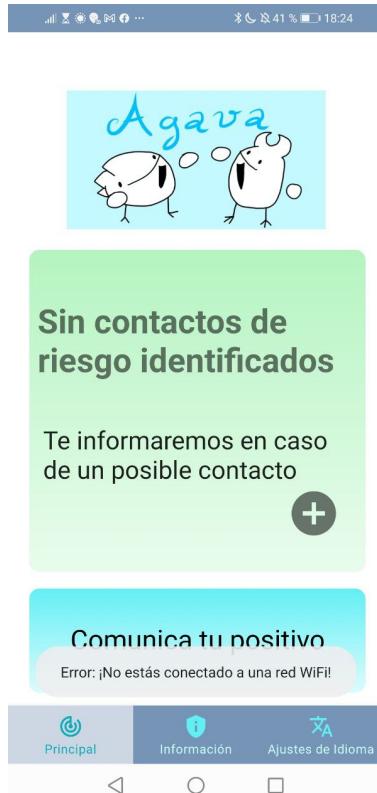


Figura 32: Aviso de desconexión

De este modo, la aplicación nos comunica que no está conectada a una red WiFi. Debido a ello algunas funciones no podrán realizarse, como es el envío de un código al servidor o la recepción de multicast.

Sin embargo, se puede navegar por la aplicación y esta no se queda colgada.



Figura 33: Aplicación ejecutándose correctamente

Del mismo modo, las conexiones Bluetooth se pueden realizar correctamente.

El funcionamiento es el esperado, pues al no haber conexión a Internet, es lógico que las funcionalidades dependientes de la red no puedan llevarse a cabo.

Por otro lado, aquellas que no dependen de una conexión a Internet siguen funcionando correctamente.

6.1.19. Recepción por multicast de un ID infectado que se encuentra en la base de datos local

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones UDP

Teniendo el servidor activado realizando el envío periódico, se abre la aplicación, estando el dispositivo conectado a Internet.

Para la comprobación de la recepción del multicast se genera un mensaje en el cual se avisa con un texto de la recepción de un paquete desde una dirección IP.

Inicialmente la recepción no funcionaba debido a que la dirección empleada dentro del rango de direcciones multicast no era para redes internas. Tras cambiarla una vez más, la recepción funcionaba pero el resto de funcionalidades se bloqueaban.

Para solucionarlo se crearon dos hilos de ejecución, uno para la recepción del multicast y otro para las funcionalidades de la aplicación.

Tras separar el hilo de ejecución de cada proceso, la recepción del multicast dejaba de bloquear a las demás funcionalidades de la aplicación.

De este modo el funcionamiento es el correcto, pudiendo recibir multicast de manera periódica a la vez que se permite el uso de la aplicación correctamente.

Una vez se recibe este multicast, se obtiene la lista de claves y fechas generadoras, se calculan los identificadores y se comparan con los almacenados en la base de datos local, en concreto los de la tabla *ids_ajenos*. Se encuentra una coincidencia y con ello se cambia el valor de la variable que define el estado de contagio. Una vez cambiado, la imagen y los textos de la pantalla principal cambian al estado *Con contactos contagiados*.



Figura 34: Aplicación ejecutándose correctamente

6.1.20. Recepción por multicast de IDs infectados y ninguno se encuentra en la base de datos local

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones UDP

Teniendo el servidor activado realizando el envío periódico, se abre la aplicación, estando el dispositivo conectado a Internet.

Para la comprobación de la recepción del multicast se genera un mensaje, en el cual se avisa con un texto de la recepción de un paquete desde una dirección IP.

Una vez se recibe este multicast, se obtiene la lista de claves y fechas generadoras, se calculan los identificadores y se comparan con los almacenados en la base de datos local, en concreto los de la tabla *ids_ajenos*. Al no encontrarse ninguna coincidencia, no cambia el valor de la variable de contagio, permaneciendo en *sin contactos*.

El caso de prueba acaba con éxito.

6.1.21. Recepción por multicast de IDs infectados y ninguno se encuentra en la base de datos local, pero el ID es un número por encima de uno almacenado

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones UDP

Teniendo el servidor activado realizando el envío periódico, se abre la aplicación, estando el dispositivo conectado a Internet.

Para la comprobación de la recepción del multicast se genera un mensaje, en el cual se avisa con un texto de la recepción de un paquete desde una dirección IP.

Una vez se recibe este multicast, se obtiene la lista de claves y fechas generadoras, se calculan los identificadores y se comparan con los almacenados en la base de datos local, en concreto los de la tabla *ids_ajenos*. Al no encontrar ninguna coincidencia, pues el identificador generado con la clave y la fecha ha de ser exacto a alguno de los almacenados para detectar un contacto, no cambia el valor de la variable de contagio, permaneciendo en *sin contactos*. El caso de prueba acaba con éxito.

6.1.22. Recepción por multicast de IDs infectados y ninguno se encuentra en la base de datos local, pero el ID es un número por debajo de uno almacenado

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones UDP

Teniendo el servidor activado realizando el envío periódico, se abre la aplicación, estando el dispositivo conectado a Internet.

Para la comprobación de la recepción del multicast se genera un mensaje, en el cual se avisa con un texto de la recepción de un paquete desde una dirección IP.

Una vez se recibe este multicast, se obtiene la lista de claves y fechas generadoras, se calculan los identificadores y se comparan con los almacenados en la base de datos local, en concreto los de la tabla *ids_ajenos*. Al no encontrar ninguna coincidencia, pues el identificador generado con la clave y la fecha ha de ser exacto a alguno de los almacenados para detectar un contacto, no cambiaría el valor de la variable de contagio, permaneciendo en *sin contactos*. El caso de prueba acaba con éxito

6.1.23. Recepción por multicast de IDs infectados y no se posee ningún ID en la base de datos local con los que comparar

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones UDP

Teniendo el servidor activado realizando el envío periódico, se abre la aplicación, estando el dispositivo conectado a Internet.

Para la comprobación de la recepción del multicast se genera un mensaje, en el cual se avisa con un texto de la recepción de un paquete desde una dirección IP.

Una vez se recibe este multicast, se obtiene la lista de claves y fechas generadoras, se calculan los identificadores y se procede a comparar con la base de datos local. Esta, al estar vacía, no devuelve nada y por lo tanto no se encuentran coincidencias.

Tras esto, el caso de prueba termina exitoso sin realizar ningún cambio en la interfaz.

6.1.24. Envío del multicast pero sin recepción (clientes inactivos)

- **Tipo de prueba:** Prueba funcional
- **Ámbito de la prueba:** Comunicaciones UDP

Se inicia el servidor, conectado a la red, pero sin abrir la aplicación cliente en ningún momento.

El servidor realiza de manera periódica, un envío a la dirección IPv4 del grupo de multicast con la información de las claves y fechas generadoras de los identificadores contagiados. También, de manera periódica hace un llamamiento a que los dispositivos de dicho grupo se unan a él. Estos envíos se realizan sin necesidad de que haya clientes de dicho grupo de multicast conectados.

El resultado es el esperado, pues es necesario que este envío se realice en todo momento para que siempre puedan unirse nuevos clientes cuando se conecten a la red.

6.1.25. Escucha, por parte del cliente, de multicast pero sin envío (servidor inactivo)

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Disponibilidad

Se inicia la aplicación cliente, con conexión a la red. La espera para recepción de paquetes vía multicast se queda en segundo plano, y el resto de la aplicación funciona correctamente, permitiendo su uso.

La aplicación no envía nada por red, como era de esperar, manteniéndose a la escucha de paquetes multicast de su grupo.

6.1.26. Cambio de idioma

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Usabilidad

Se inicia la aplicación y se va a la pantalla de *Ajustes de idioma*. Una vez ahí, se selecciona el idioma al que se desea cambiar. En este caso, seleccionamos *English*. Una vez pulsado *Aceptar*, la aplicación se reinicia, haciendo un breve pestaneo.

Tras ello, el idioma de los textos aparece cambiado al inglés.

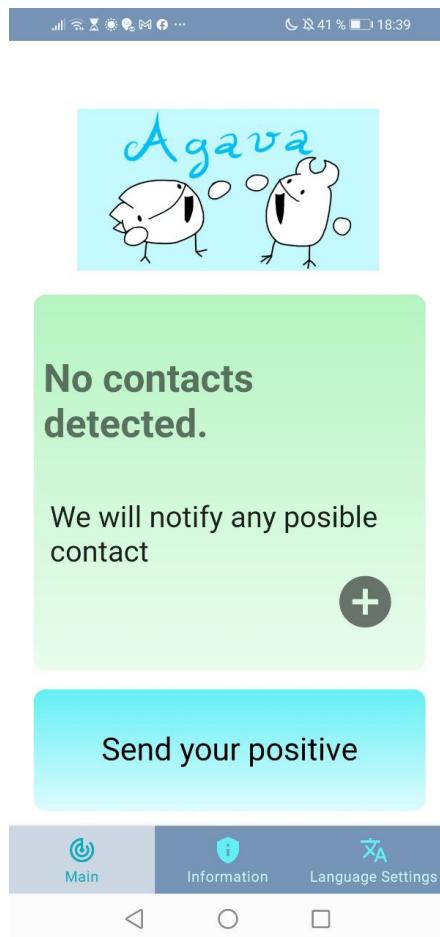


Figura 35: Aplicación en inglés

6.2. Pruebas de caja blanca

Las pruebas de caja blanca son aquellas diseñadas con conocimiento profundo del software. Esto nos permite comprobar funcionalidades concretas del propio software, como por ejemplo comprobar el nivel de seguridad de la aplicación desarrollada.

6.2.1. Escucha del canal de conexión en el momento de envío de un código con los IDs al servidor

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Seguridad

Se abre la aplicación estando el dispositivo conectado a Internet.

Tras ello, dentro de la aplicación se selecciona *Comunica tu positivo* y se rellenan los datos.

Se inicia un programa de sniffing y se envía el código desde la aplicación.

No.	Time	Source	Destination	Info	Protocol	Length
1891	30.654436	192.168.1.1	192.168.1.1	→ 3384 [ACK] Seq=1461 Ack=1 Win=87808 Len=1460	TCP	1
1892	30.654439	192.168.1.1	192.168.1.1	3384 → [ACK] Seq=32 Ack=2921 Win=131328 Len=0	TCP	1
1893	30.654439	192.168.1.1	192.168.1.1	→ 3384 [ACK] Seq=2921 Ack=1 Win=87808 Len=1460	TCP	1
1894	30.654439	192.168.1.1	192.168.1.1	→ 3384 [PSH, ACK] Seq=4381 Ack=1 Win=87808 Len=1126	TCP	1
1895	30.654439	192.168.1.1	192.168.1.1	→ 3384 [FIN, ACK] Seq=5507 Ack=1 Win=87808 Len=0	TCP	1
1896	30.654439	192.168.1.1	192.168.1.1	→ 3384 [RST] Seq=1 Win=0 Len=0	TCP	1
1898	30.654439	192.168.1.1	192.168.1.1	3384 → [ACK] Seq=32 Ack=5508 Win=131328 Len=0	TCP	1
< Calculated window size: 87808 [Window size scaling factor: 256] Checksum: 0x85Aa [unverified] [Checksum Status: Unverified] Urgent Pointer: 0						
0030	01.57 85 4a 00 00 61 63 56 62 35 51 58 75 70 74	192.168.1.1	192.168.1.1	← 3384 [AC] Vb5QXupt	TCP	1
0040	4b 64 33 34 4d 78 5a 33 66 64 66 36 69 65 4d 55	192.168.1.1	192.168.1.1	Kd34MwZ3 jdf6ie1mU	TCP	1
0050	30 6a 5a 70 43 59 7a 6a 78 79 76 4e 33 78 58 54	192.168.1.1	192.168.1.1	0jTpCvXj pyvN3xXT	TCP	1
0060	69 33 6a 6d 72 78 59 52 38 37 44 70 63 7a 33 39	192.168.1.1	192.168.1.1	i3dmrxYR 87Dpcz39	TCP	1
0070	52 73 52 63 2f 51 75 61 71 70 2b 46 67 62 35 73	192.168.1.1	192.168.1.1	R5Rc/Qua qp+Fgb5s	TCP	1
0080	63 67 6a 61 5a 5b 37 78 42 30 39 48 4b 45	192.168.1.1	192.168.1.1	0gPzXARj 727000	TCP	1
0090	6d 67 74 53 4d 42 6d 2f 6d 37 41 73 46 4e	192.168.1.1	192.168.1.1	getSTDRn vRtas3W	TCP	1
0100	53 47 6e 4f 67 36 6e 75 46 4b 63 67 4e 63 36 52	192.168.1.1	192.168.1.1	S6n0gBnu vKcgfH6R	TCP	1
0110	75 6d 37 49 4c 64 4e 78 47 6d 74 41 2f 68 55 2b	192.168.1.1	192.168.1.1	un71ldlxv GetA/Hu	TCP	1
0120	37 2b 33 30 6a 31 36 66 45 6e 74 42 64 31 4c 75	192.168.1.1	192.168.1.1	7+30j16f EntBd1Lu	TCP	1
0130	38 4e 54 53 45 55 66 78 71 39 70 47 31 33 48 4f	192.168.1.1	192.168.1.1	8NTSEUfx q9p013HO	TCP	1
0140	44 4b 30 73 6d 57 72 71 61 4c 74 6b 71 2b 71 37	192.168.1.1	192.168.1.1	DK8SmLrq kLtkq+q7	TCP	1
0150	49 2b 53 47 55 66 4f 78 78 6d 53 75 4c 59 43	192.168.1.1	192.168.1.1	I+SGUnFO vxmSuLYC	TCP	1
0160	69 6a 79 51 41 52 6b 52 4f 50 58 7a 32 51 4c 59 68	192.168.1.1	192.168.1.1	1jyJQRfr APz2QLYh	TCP	1
0170	53 71 71 66 6f 6d 63 37 59 2f 66 44 67 38 52 48	192.168.1.1	192.168.1.1	Sqgfomc7 Y/fdbgRH	TCP	1
0180	69 68 42 72 45 55 65 75 46 68 4b 59 44 73 45 53	192.168.1.1	192.168.1.1	iHBrEUeu FHKM5ES	TCP	1
0190	34 7a 57 68 4e 4e 50 56 38 6a 31 58 4a 72 48 50	192.168.1.1	192.168.1.1	4zWnNPV 831X0HP	TCP	1
01a0	4c 42 49 4e 48 56 69 68 38 53 51 70 65 2f 6b 66	192.168.1.1	192.168.1.1	L81nWVih 55Qpe/kf	TCP	1
01b0	4c 30 60 49 71 39 3d 69 60 55 59 37 39 51 46	192.168.1.1	192.168.1.1	L0p0XkWn hU73XQN	TCP	1
01c0	4c 30 60 49 71 39 3d 69 60 55 59 37 39 51 46	192.168.1.1	192.168.1.1	3KuE3TRn ZpD+26z	TCP	1
01d0	79 63 57 65 4d 26 70 45 63 4f 2b 74 30 45 50 43	192.168.1.1	192.168.1.1	+SM4Mp Y195Z0Pc	TCP	1
01e0	2b 73 4d 4d 34 4d 4a 2f 60 59 6c 39 53 5a 51 39 61	192.168.1.1	192.168.1.1	XGebvryD SyULqmdA	TCP	1
01f0	4d 32 66 45 4e 68 4d 6c 71 43 41 45 2b 4a 4a 4d	192.168.1.1	192.168.1.1	M2fENHm1 qCAE+jJM	TCP	1
01g0	43 45 32 48 58 65 63 6c 44 52 35 70 53 2f 55 58	192.168.1.1	192.168.1.1	C2E9Xec1 DR5pS/JX	TCP	1
01h0	33 58 70 48 55 56 30 4e 45 6c 65 44 54 5a 50 68	192.168.1.1	192.168.1.1	3XpHUWON Ele0T2Ph	TCP	1
01i0	4f 78 61 63 41 49 64 49 30 72 51 32 30 38 41 31	192.168.1.1	192.168.1.1	0xacA1d1 0rQ208A1	TCP	1
01j0	47 50 69 7a 4e 34 64 63 68 46 74 79 46 49 68 32	192.168.1.1	192.168.1.1	GPizA1d Nadc hFtyFIh2	TCP	1

Figura 36: Resultado del programa de sniffing

El resultado nos dice que el mensaje que envía la aplicación viaja cifrado y que por tanto está salvo de observadores externos.

6.2.2. Escucha del canal de conexión en el momento de envío de un código incorrecto al servidor

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Seguridad

El proceso es el mismo que en el caso anterior.

No.	Time	Source	Destination	Info	Protocol	Length
2832	33.543919	192.168.1.1	192.168.1.1	3384 → [PSH, ACK] Seq=1 Ack=1 Win=131328 Len=31	TCP	31
2833	33.550849	192.168.1.1	192.168.1.1	→ 3384 [ACK] Seq=1 Ack=32 Win=87808 Len=0	TCP	0
2837	33.6083292	192.168.1.1	192.168.1.1	→ 3384 [ACK] Seq=1 Ack=32 Win=87808 Len=1460	TCP	1460
2838	33.6083292	192.168.1.1	192.168.1.1	→ 3384 [ACK] Seq=1461 Ack=32 Win=87808 Len=1460	TCP	1460
2839	33.6083292	192.168.1.1	192.168.1.1	→ 3384 [ACK] Seq=2921 Ack=32 Win=87808 Len=1460	TCP	1460
2840	33.6083292	192.168.1.1	192.168.1.1	→ 3384 [ACK] Seq=4381 Ack=32 Win=87808 Len=1126	TCP	1126
2941	33.603450	192.168.1.1	192.168.1.1	→ 3384 [PSH, ACK] Seq=4381 Ack=5507 Win=131328 Len=0	TCP	0
[Calculated window size: 87808] [Window size scaling factor: 256] Checksum: 0xbee9 [unverified] [Checksum Status: Unverified] Urgent Pointer: 0						
0030	01 57 be e9 00 06 76 72 78 74 56 76 57 64 42 41	-W- 0v ptVvldB8A				
0040	66 66 52 74 6b 6e 32 76 6b 75 74 33 4b 73 59 33	fFrTknz2v kut3KsY3				
0050	4a 62 73 76 58 79 62 30 6f 6e 30 33 4f 6b 37 72	JbsxXyb0 on030kY7r				
0060	71 64 35 71 58 52 76 76 2f 79 2b 65 52 58 7a 4f	qd5qXrvv /y+eRxz0				
0070	37 2b 42 4c 6a 59 6c 4b 51 6d 74 45 33 77 56 36	7-BLjY1k Qmte3wV6				
0080	57 6d 5a 66 66 41 2f 49 58 73 46 79 66 6d 54	WnZjffFA IXsYfymT				
0090	79 36 70 54 4a 6e 64 30 6a 36 48 54 34 79 34	y6p7Jmd0 l36Ht4y4				
00a0	64 56 64 43 2f 6b 68 5a 49 6c 6a 38 31 65 51 6d	dVdc/khz 11j8leQm				
00b0	30 58 44 48 79 33 4c 70 42 2b 69 70 41 6b 32 68	0XDHylpJ J+iplk2h				
00c0	59 4f 47 30 6a 4e 54 51 77 68 4a 48 71 2f 43 4e	YONQjhTQ whJKq/CN				
00d0	55 41 47 46 47 70 2b 44 63 79 53 36 42 66 34 56	UAGnGp+D cy56BfAU				
00e0	52 4c 42 62 45 6f 76 55 78 65 54 69 38 47	RslE7E 1xw4d18w				
00f0	72 44 46 62 43 57 47 62 53 62 72 48 40	0a97d14L IKrjXXj4				
0100	6f 61 39 37 64 69 34 4c 49 4b 72 6a 58 6a 34	0a97d14L IKrjXXj4				
0110	37 48 44 63 64 70 65 39 4c 57 4b 59 42 59 4c 49	7HDcp9 LkKYBVlI				
0120	62 34 66 67 77 58 67 45 51 39 72 78 4d 43 7a 61	b4f6pzgE Q9rxMcz4				
0130	78 30 42 37 39 54 69 52 61 39 48 50 69 71 63 64	x0B7971R o9HPiqcd				
0140	4f 64 70 57 4f 79 62 64 61 46 7a 55 49 70 41 75	Ompl0zbd afzUApAU				
0150	61 6e 60 78 4f 59 63 30 72 64 66 76 38 4e 4c 69	anKxDc0 rdfv8NLi				
0160	64 4c 44 37 62 61 74 44 63 79 4c 2f 6f 44 6b 6c	dLD7baP0 cyl/o0kl				
0170	30 77 67 45 54 48 31 62 42 4a 63 73 78 58 48 4c	0wgeZH1b B3csxHL				
0180	6b 70 58 75 7a 71 79 50 73 78 70 51 70 71 59 67	kpZuzyoP sxp0pqyG				
0190	4c 30 53 7a 34 79 39 4a 61 32 74 6a 37 6f 5a	L0Z2Ay93 a2ztJ7o2				
01a0	67 2b 69 61 6d 62 42 6a 55 6f 41 76 53 55 56	g+iamB0j UeoAvsEv				
01b0	33 6c 54 6d 5a 75 4e 77 38 79 43 67 6d 78 62 73	31mz0uW 8yComxks				
01c0	38 64 31 51 64 4e 37 68 78 4a 61 32 69 4d 32 50	8d1q0l7h XjaZ1MP				
01d0	4f 64 41 61 53 62 46 6d 4c 38 35 45 39 62 50 48	0dAStbfm L85E9bPH				
01e0	4c 34 2f 38 37 37 73 4e 74 46 62 33 42 68 5a 4b	L4/877sh tfb3bhZK				

Figura 37: Resultado del sniffing

Como vemos, al igual que en el caso anterior, el resultado es que se puede ver el mensaje cifrado y que por tanto está a salvo de observaciones no deseadas.

6.2.3. Escucha del canal de conexión en el momento del envío del multicast

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Seguridad

Se inicia un programa de sniffing filtrando las direcciones de multicast (224.0.0.0 en adelante hasta 224.0.0.255).
Se abre la aplicación estando el dispositivo conectado a Internet.

El resultado es que el mensaje de multicast se puede observar sin ningún tipo de impedimento. Esto se debe a que en las conexiones UDP no hay un *handshake* donde se puedan intercambiar claves para poder realizar un cifrado.

6.2.4. Barrido de puertos de la máquina cliente

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Seguridad

Se inician tanto la máquina cliente como la máquina servidor. Para la realización de esta prueba se utilizó una máquina virtual Kali y la herramienta *nmap*. La información que devuelve nmap tras usarse, incluye el barrido de puertos realizado junto a el número de estos, protocolo, el nombre del servicio y su estado (abierto, cerrado, filtrado o no filtrado). Si el estado se marcara como *filtrado* esto significa que el puerto está siendo bloqueado por un firewall u otro software similar.

Para llamar a nmap se escribe el comando de la forma:

```
# nmap <Aquí los argumentos> <Aquí el nombre del host>
```

En concreto el comando utilizado fue:

```
sudo nmap -p <PUERTO> 192.168.1.N
```

Donde en <PUERTO> se especifica el puerto a escanear y donde N es el último bloque de la dirección IP en la red local.

Se analizó el puerto 4446 (multicast de recepción).

Inicialmente se obtuvo como respuesta que el puerto estaba cerrado. Esto no tenía sentido, pues el cliente estaba recibiendo paquetes en ese momento. El problema era que el análisis se estaba realizando vía TCP, forma predeterminada de *nmap*.

Tras especificar en el comando el análisis de puertos UDP con el parámetro *-sU*, quedando el comando:

```
sudo nmap -sU -p 4446 192.168.1.N
```

Se obtuvo una respuesta más lógica:

```
pinkbat@kali:~$ sudo nmap -sU -p 4446 192.168.1.N
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-15 06:20 EDT
Nmap scan report for 192.168.1.N
Host is up (0.16s latency).

PORT      STATE      SERVICE
4446/udp  open|filtered  n1-fwp
MAC Address: [REDACTED]

Nmap done: 1 IP address (1 host up) scanned in 2.08 seconds
```

Figura 38: Puerto 4446. Recepción de multicast. UDP

Aquí se puede ver que el puerto está *open|filtered*. Dicho estado significa que puede estar abierto o filtrado. Al tratarse de paquetes UDP, los paquetes para realizar el escaneo se envían sin carga. Debido a esto, el puerto los descarta incluso estando abierto, pues no poseen contenido. Por tanto *nmap* no puede concretar si el estado es abierto o filtrado por un firewall.

Se ha podido comprobar que la aplicación solamente abre el puerto necesario para establecer las comunicaciones con el servidor, y ninguno más.

6.2.5. Barrido de puertos de la máquina servidor

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Seguridad

Se inician tanto la máquina cliente como la máquina servidor. Para la realización de esta prueba se utilizó una máquina virtual Kali y la herramienta *nmap*. El comando utilizado es el siguiente:

Para llamar a nmap se escribe el comando de la forma:

```
# nmap <Aquí los argumentos> <Aquí el nombre del host>
```

En concreto el comando utilizado fue:

```
sudo nmap -p <PUERTO> 192.168.1.N
```

Donde en <PUERTO> se especifica el puerto a escanear y donde N es el último bloque de la dirección IP en la red local.

Se analizaron los puertos 4445 (multicast de envío), 3327 (puerto de conexión de MariaDB) y 3384 (puerto de recepción de datos vía TCP en el servidor).

Para el análisis del puerto 4445, el cual envía los paquetes de multicast, se emplea el parámetro *-sU*, pues el envío de dichos paquetes se realiza vía UDP.

Para los otros dos puertos se realiza un escaneo habitual, vía TCP, que es el protocolo por el que admite las conexiones habituales desde un cliente.

Los resultados son los siguientes:

Para el envío de multicast vía UDP:

```
pinkbat@kali:~$ sudo nmap -sU -p 4445 192.168.1.N
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-15 06:11 EDT
Nmap scan report for 192.168.1.N
Host is up (0.00013s latency).

PORT      STATE      SERVICE
4445/udp  open|filtered  upnotify
MAC Address: [REDACTED]

Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Figura 39: Puerto 4445. Envío de multicast. UDP.

El resultado obtenido es *open|filtered*. Dicho estado significa que puede estar abierto o filtrado.

Para la escucha de paquetes para la base de datos *MariaDB* vía TCP:

```
pinkbat@kali:~$ sudo nmap -p 3327 192.168.1.N
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-15 06:16 EDT
Nmap scan report for 192.168.1.N
Host is up (0.00014s latency).

PORT      STATE SERVICE
3327/tcp  open  bbars
MAC Address: [REDACTED]

Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds
```

Figura 40: Puerto 3327. Acceso a base de datos. TCP.

Para la escucha de conexiones con el servidor de paquetes con códigos de contagio vía TCP:

```
pinkbat@kali:~$ sudo nmap -p 3384 192.168.1.N
Starting Nmap 7.80 ( https://nmap.org ) at 2021-05-15 06:16 EDT
Nmap scan report for 192.168.1.N
Host is up (0.00017s latency).

PORT      STATE SERVICE
3384/tcp  open  hp-clic
MAC Address: [REDACTED]

Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
```

Figura 41: Puerto 3384. Envío de códigos. TCP.

El resultado en ambos casos es puertos abiertos. Por lógica, el puerto UDP también estará abierto.

En el caso de desplegar la aplicación servidor en una máquina servidor, una de las formas de evitar el escaneo de puertos sería la instalación de un cortafuegos o el uso de puertos *honeypot*, los cuales pueden servir para atrapar en bucle bots atacantes.

6.2.6. Ataque DDoS

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Seguridad

Se inicia el dispositivo, conectado a la red. Para poder realizar el ataque se utilizó una máquina virtual Kali y la herramienta *inviteflood*. Este comando envía paquetes vía *UDP* de forma masiva con la intención de que el objetivo se sature y no pueda realizar sus funciones de red correctamente.

En concreto el comando utilizado fue el siguiente:

```
sudo inviteflood eth0 '' 192.168.1.N 192.168.1.N 700000 -D 4446
```

En dicho comando, hay argumentos obligatorios, los cuales son:

- **Interfaz.** En este caso es *eth0*, y es la interfaz o red donde tanto el objetivo como la máquina atacante deben estar conectados.
- **Usuario objetivo.** Aquí se debe especificar el usuario de la máquina objetivo al que se va a atacar. En este caso no hay, luego se deja con comillas vacías.
- **Dominio objetivo.** Este campo admite tanto direcciones URL como direcciones IPv4. Dado que es un servidor montado en una red local, aquí se especifica su IPv4 *192.168.1.N* donde N es el último bloque de la dirección IP en la red local.
- **Objetivo.** En este campo se especifica la dirección IPv4. En caso de haberla puesto en el anterior punto, se repite.
- **Flood stage.** En este campo se introduce el número de paquetes UDP que se mandarán para realizar el ataque. En este caso se realizó un ataque con 700000 paquetes.

Además, en este caso se va a realizar un ataque al puerto 4446.

Tras ejecutar el comando, y pasados unos 2 segundos, se obtuvo la siguiente pantalla:



```
[192.168.1.1] INVITE sip:192.168.1.1 SIP/2.0
Via: SIP/2.0/UDP 192.168.1.1;branch=4
e448e3e-d73b-4609-a0d4-ec0000000002
Max-Forwards: 70
Content-Length: 462
To: <sip:192.168.1.1:4446>
From: <sip:192.168.1.1>;tag=4e449510-d73b-4609-af86-310000000002
Call-ID: 4e449a16-d73b-4609-8864-2b0000000002
CSeq: 0000000002 INVITE
Supported: timer
Allow: NOTIFY
Allow: REFER
Allow: OPTIONS
Allow: INVITE
Allow: ACK
Allow: CANCEL
Allow: BYE
Content-Type: application/sdp
Contact: <sip:192.168.1.1>
Supported: replaces
User-Agent: Elite 1.0 BrCM Callctrl/1.5.1.0
MxSF/v.3.2.6.26

v=0
o=MxSIP 0 639859198 IN IP4
192.168.1.1
s=SIP Call
c=IN IP4 192.168.1.1
t=0
m=audio 16388 RTP/AVP 0 18 101 102 107
104 105 106 4 103
a=rtpmap:0 PCMU/8000
a=rtpmap:18 G729/8000
a=rtpmap:101 BV16/8000
a=rtpmap:102 BV32/16000
a=rtpmap:107 L16/16000
```

Figura 42: Resultado de ataque de denegación de servicio.

Tras la recepción de dicha pantalla un par de veces, el dispositivo se saturó, y dejó de poder conectarse a la red hasta que se reinició su conexión.

Una forma de evitar esto es abriendo el puerto únicamente durante el envío del código. Una vez terminada la comunicación, este se cierra para evitar la llegada de tráfico indeseado al mismo.

6.2.7. Inyección de código desde la aplicación cliente

- **Tipo de prueba:** Prueba no funcional
- **Ámbito de la prueba:** Seguridad

Existen dos vías para la posible inyección de código:

- **Campo de selección de fecha.** Aquí el usuario puede seleccionar una fecha de un calendario que aparece cuando hace click, denegando la posibilidad de escribir caracteres de forma libre. Como esto ocurre siempre que se quiere cambiar este campo no es posible inyectar ningún tipo de código.
- **Campo de introducción de código de contagio.** En este campo se le pide introducir un código al usuario. Los caracteres están restringidos tal que solo se admiten dígitos. De este modo se bloquea la posibilidad de escribir inyecciones de código en la base de datos del servidor, pues no es posible introducir caracteres tales que % para especificar caracteres vía código URL, buscando nombres o direcciones, u otros admitidos en el lenguaje SQL (guiones, comillas...).

7. Análisis de riesgos de seguridad y privacidad

El análisis y la gestión de riesgos es uno de los primeros pasos a realizar en el ámbito de la seguridad de un sistema informático.

Para el desarrollo de este, se ha decidido hacer uso de una de las herramientas que proporciona el Centro Criptológico Nacional o CCN-CERT.

A modo de contexto, a continuación se va a exponer el estado del arte sobre todo lo que involucra realizar un análisis de riesgos en España.

7.1. Estado del arte: Seguridad y Privacidad

En primer lugar, es necesario comprender las distintas fases y el proceso a seguir para la realización de un análisis de riesgo en cada una de ellas.

7.1.1. Fase 1: Definición del alcance

Un análisis de riesgos puede abarcar desde empresas enteras hasta departamentos o proyectos individuales.

Si el análisis ha de ser de una empresa, su alcance lo marcará el Plan Director de Seguridad, una planificación que debe realizar la empresa y donde se marcan las prioridades, los responsables y los recursos a utilizar para mejorar su seguridad. [1] [54]

7.1.2. Fase 2: Identificación de activos

La identificación de activos trata sobre identificar los recursos más importantes que estén relacionados con la empresa, departamento o proyecto sobre el que se va a realizar el análisis.

Para realizarlo existen diferentes alternativas, dependiendo de la complejidad y profundidad del análisis a realizar, como listar en una hoja *Excel* o hacer uso de herramientas orientadas a este tipo de análisis. [1]

7.1.3. Fase 3: Identificación y selección de amenazas

El catálogo de amenazas a las que un activo puede estar expuesto es muy amplio. Por esta razón, se recomienda tomar un enfoque práctico tomando las amenazas más probables, como un incendio, y obviando las más improbables, como la caída de un meteorito. [1]

7.1.4. Fase 4: Identificación de vulnerabilidades y salvaguardas

En esta fase se deben estudiar las vulnerabilidades de nuestro ámbito. Si por ejemplo estamos estudiando el conjunto de ordenadores de una empresa, detectar modelos obsoletos, programas antiguos o antivirus sin actualizar son posibles vulnerabilidades que se podrían encontrar.

Por otro lado, también se deben analizar y documentar las medidas de seguridad que ya están implementadas en nuestro ámbito, las llamadas salvaguardas o contramedidas. Estas, reducen el impacto que puede tener una amenaza sobre los activos y deben ser tenidas en cuenta para calcular el riesgo al que están expuestos los activos del ámbito a analizar. [1]

		IMPACTO		
		Bajo	Medio	Alto
PROBABILIDAD	Baja	Muy bajo	Bajo	Medio
	Media	Bajo	Medio	Alto
	Alta	Medio	Alto	Muy alto

Figura 43: Ejemplo de matriz de riesgos

7.1.5. Fase 5: Evaluación del riesgo

Para cada par activo-amenaza se ha de estimar la probabilidad de que esa amenaza ocurra y el impacto que tendría en caso de que se materialice. Con todo esto el siguiente paso sería calcular el riesgo.

Para calcular el riesgo hay dos alternativas: Análisis cuantitativo y análisis cualitativo. [1]

- **Análisis cualitativo.** Los criterios para la evaluación de los riesgos serían del estilo, por ejemplo, de *Alto*, *Medio* y *Bajo*. Para el cálculo del riesgo se hace uso de una matriz de riesgo similar a la siguiente:
- **Análisis cuantitativo.** En este caso los criterios de evaluación son números. Lo más habitual es utilizar una escala ascendente de forma que el número más alto signifique el máximo riesgo y el número más bajo el mínimo riesgo. En este caso, para el cálculo del riesgo haremos uso de esta simple fórmula:

$$\text{riesgo} = \text{probabilidad} \times \text{impacto}$$

Sea cual sea la alternativa que se utilice siempre se debe tener en cuenta a la hora de estimar la probabilidad y el impacto las amenazas y contramedidas existentes. No tendría el mismo impacto que, por ejemplo, se cayera un servidor del sistema contando con uno de respaldo que sin tener nada.

7.1.6. Fase 6: Tratamiento del riesgo

Finalmente, una vez hemos calculado el riesgo, es necesario tratar aquellos riesgos que superen un límite que hayamos impuesto nosotros mismos, por ejemplo, en el caso de un análisis cualitativo, que superen el *Medio*.

Existen cuatro estrategias primordiales para tratar un riesgo [1] :

- **Transferir el riesgo a un tercero.** Esto se puede llevar a cabo, por ejemplo, contratando un seguro que cubra los posibles daños.
- **Eliminar el riesgo.** Para eliminar el riesgo se tendrá que eliminar el sistema o proceso que este expuesto a él. Por supuesto, esta alternativa se debe usar únicamente si lo que se va a eliminar no es estrictamente necesario.
- **Asumir el riesgo.** Esta situación se puede dar en caso de que las contramedidas que lo mitigan sean demasiado costosas.
- **Implantar medidas para su minimización.**

7.1.7. CCN-CERT

El CCN-CERT tiene como cometido ayudar a mejorar la ciberseguridad española. Creado como CERT Gubernamental Nacional español en el año 2006, coopera y ayuda a proporcionar una respuesta rápida y eficiente a los ciberataques y ciberamenazas y también coordina los distintos Centros de Operaciones de Ciberseguridad existentes a nivel público estatal. [60]

Todo ello tiene como meta conseguir un espacio más seguro y confiable, protegiendo la información clasificada y sensible, defendiendo el patrimonio tecnológico español, otorgando formación a personal experto, empleando políticas y procedimientos de seguridad y utilizando y desarrollando las tecnologías más convenientes para ello. [44]

Actualmente, el CCN-CERT dispone de numerosas soluciones tecnológicas para el ámbito de la seguridad, cada una de ellas orientada a un problema concreto. Algunas de ellas son:

- **AMPARO.** Es una solución cuya finalidad es ayudar a la implantación y gestión de la seguridad en entidades y organismos amoldándose, en concreto, al Esquema Nacional de Seguridad (ENS), del cual se hablará más adelante. Esta herramienta se utiliza en conjunción con otra de las soluciones del CCN-CERT, **INES** [35], cuya función es evaluar el estado de seguridad de los sistemas TIC en las entidades que la utilicen. [9]
- **REYES.** Es un repositorio común y estructurado de amenazas y código dañino. REYES recibe información de diversas fuentes minuciosamente escogidas, permite descargar informes, utiliza grafos de asociación de eventos y procesa y analiza la información para priorizar aquella más relevante para agilizar las funciones del analista. [57]
- **PILAR.** Es una herramienta para el análisis y gestión de riesgos. Permite conocer cuánto vale, cómo es y cómo de protegido está un sistema. [3]

El desarrollo de este análisis se realizará haciendo uso de la herramienta PILAR.

7.1.8. PILAR

PILAR es una herramienta para el análisis y gestión de riesgos que utiliza la metodología MAGERIT. Ha sido desarrollada por el CCN-CERT y existen diferentes variantes en función de las necesidades de sus usuarios. [3]

- **μ PILAR.** Es la versión más básica de PILAR. Se utiliza para realizar análisis de riesgos de forma rápida. Su característica principal es que se distribuye con perfiles específicos, que limitan el alcance del análisis. Aún así, los resultados obtenidos se pueden trasladar a PILAR para realizar un estudio más detallado si fuese necesario. [68]
- **PILAR Basic.** Esta versión de PILAR está orientada a PYMES y administración local. El análisis que permite realizar ya no está limitado por perfiles y además permite proponer salvaguardas, también conocidas como contramedidas, para tratar los riesgos identificados. [53]
- **PILAR.** Es la versión más completa de la herramienta. Consta de dos versiones: PILAR RM, para realizar análisis y gestión de riesgos y PILAR BCM, para realizar análisis de impacto y continuidad de operaciones. [52]

Una vez presentadas las versiones de la herramienta es necesario decidir qué versión utilizar para realizar el análisis.

Valorando las funcionalidades que ofrecen estas herramientas y el volumen del proyecto, se ha decidido hacer uso de *PILAR Basic*.

7.1.9. Metodología MAGERIT

MAGERIT (Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información) es la respuesta proporcionada por el antiguo Consejo Superior de Administración Electrónica (actualmente Comisión de Estrategia TIC) a la creciente dependencia por parte de la sociedad de las tecnologías de la información (TIC) para la realización de cualquier tarea. [41]

Si bien es cierto que el uso de las TIC facilita enormemente la realización de muchas actividades, el hecho de, en muchos casos, depender de ellas, da lugar a la aparición de ciertos riesgos. Estos riesgos se deben tratar de minimizar con medidas cuyo fin es generar confianza en los usuarios sobre estas tecnologías.

MAGERIT es la metodología apropiada para quienes hacen uso de información digital y sistemas informáticos. MAGERIT permite conocer el valor de estos datos y/o servicios proporcionados que están en riesgo y protegerlos. Se han realizado varias versiones, siendo la actual la 3.0.

Sintetizando esta información, se pueden describir los objetivos que persigue MAGERIT:

1. «Concienciar a los responsables de las organizaciones de información de la existencia de riesgos y de la necesidad de gestionarlos.» ([Portal de Administración Electrónica. MAGERIT v.3 : Métodología de Análisis y Gestión de Riesgos de los Sistemas de Información, 2012](#))
2. «Ofrecer un método sistemático para analizar los riesgos derivados del uso de tecnologías de la información y comunicaciones (TIC).» ([Portal de Administración Electrónica. MAGERIT v.3 : Métodología de Análisis y Gestión de Riesgos de los Sistemas de Información, 2012](#))
3. «Ayudar a descubrir y planificar el tratamiento oportuno para mantener los riesgos bajo control Indirectos.» ([Portal de Administración Electrónica. MAGERIT v.3 : Métodología de Análisis y Gestión de Riesgos de los Sistemas de Información, 2012](#))
4. «Preparar a la Organización para procesos de evaluación, auditoría, certificación o acreditación, según corresponda en cada caso.» ([Portal de Administración Electrónica. MAGERIT v.3 : Métodología de Análisis y Gestión de Riesgos de los Sistemas de Información, 2012](#))

7.1.10. Esquema Nacional de Seguridad

El Esquema Nacional de Seguridad o ENS está basado en la Ley 11/2007 [26], donde se trata de regularizar las comunicaciones electrónicas tanto entre las Administraciones públicas y los ciudadanos como entre las propias Administraciones, así como unificar los criterios de agilidad y seguridad.

Lo desarrolla el Centro Criptológico Nacional (CCN-CERT) junto con las administraciones públicas y bajo las recomendaciones de universidades públicas, industrias del sector TIC y directrices internacionales y europeas.

El ENS tiene por objeto implantar una política de seguridad en el uso de medios electrónicos en la Administración pública y garantizar una protección adecuada de la información mediante la definición de unos requisitos básicos y unos requisitos mínimos. [2]

Su regulación final se estableció en el Real Decreto 951/2015 [25], una modificación del Real Decreto 3/2010 [27] donde se comenzó a regular su uso por primera vez.

1. Objetivos del ENS

El ENS persigue tres objetivos fundamentales: [2]

- **Confianza.** Se busca fomentar y lograr generar confianza en los usuarios que hagan uso de los medios electrónicos de la Administración pública.

- **Unificar criterios.** Mediante la puesta en común de las medidas de seguridad empleadas por las Administraciones en materia de seguridad de la información.
- **Integración de sistemas.** Establecer un lenguaje común tanto para la comunicación entre Administraciones, como al la hora de fijar unos requisitos de seguridad a los proveedores de los sistemas de información.

2. Principios Básicos del ENS

Con estos objetivos en mente, el ENS establece unos criterios o principios básicos para realizar la toma de decisiones en el ámbito de la seguridad de la información. Estos principios son: [2]

- Seguridad Integral.
- Gestión de Riesgos.
- Prevención, reacción y recuperación.
- Líneas de Defensa.
- Reevaluación periódica.

3. Requisitos Mínimos del ENS

Como se ha mencionado con anterioridad, con la finalidad de garantizar la adecuada protección de la información, el ENS establece una serie de requisitos mínimos que cualquier tipo de sistema debe cumplir. Estos requisitos son: [2]

- Organización e implantación del sistema de seguridad.
- Análisis y gestión de riesgos.
- Gestión del personal.
- Profesionalidad.
- Autorización y control de accesos.
- Protección de las instalaciones.
- Adquisición de productos.
- Seguridad por defecto.
- Integridad y actualización del sistema.
- Protección de la información almacenada en tránsito.
- Prevención ante otros sistemas de información interconectados.
- Registro de la actividad.
- Incidentes de seguridad.
- Continuidad de la actividad.
- Mejora de la continuidad del proceso.

4. Clasificación de los sistemas de información

Como es lógico, no todos los sistemas de información tratan con información del mismo valor. Siguiendo este razonamiento, información de diferente consideración necesitará de diferente tipo de medidas de seguridad. Por esta razón, el ENS propone unas categorías para la clasificación de estos sistemas: [2]

- **Categoría alta.** Los sistemas de categoría alta son aquellos en los que los riesgos de seguridad pueden causar daños catastróficos.
- **Categoría media.** Los sistemas de categoría media se caracterizan por padecer riesgos de seguridad en los que los daños causados son graves, sin llegar a un nivel superior.

- **Categoría baja.** Un sistema se considera de categoría baja cuando los riesgos en la seguridad de la información no superan la causa de un daño limitado, sin alcanzar niveles graves o superiores.

5. Medidas de seguridad en el ENS.

El cumplimiento de esta normativa tiene como consecuencia la implantación de medidas de seguridad. Estas medidas pueden afectar en diferentes aspectos los sistemas de información, en concreto: [2]

- **Organización.** A nivel organizativo nos encontramos medidas como políticas, normativas y procedimientos de seguridad o procesos de autorización.
- **Operatividad.** A nivel operativo aparecen medidas como la planificación, el control de acceso o la monitorización del sistema.
- **Protección.** A nivel de protección surgen medidas como la protección de activos (equipos, comunicaciones, instalaciones, servicios, etc...) o la gestión del personal.

7.1.11. Amenazas en la actualidad.

El Centro Criptológico Nacional (CCN-CERT) publica todos los años un informe donde recoge las ciberamenazas y tendencias que se han podido observar. Debido a la irrupción de la pandemia de COVID-19 que ha tenido lugar durante el año 2020, se ha hecho hincapié en la ciberseguridad relacionada con este ámbito.

Ya que este proyecto está orientado hacia este dominio, se va a describir a continuación un resumen de las amenazas recogidas en dicho informe.

■ Ataques de ransomware

El ransomware es uno de los tipos más extendidos y peligrosos de ciberataques. Es un malware o programa malicioso cuya función es impedir el acceso a determinadas partes o a ciertos archivos del sistema operativo que ha infectado. Una vez realizado esto el atacante exige a la víctima del programa el pago de un rescate para poder volver a tener acceso a los datos restringidos. [56]

Por lo general, este ataque no se realiza de forma inmediata. Una vez la víctima ha sido infectada, los ciberdelincuentes emplean varios días con el objetivo de localizar los activos de mayor valor ya que, así, el impacto del programa será el máximo posible.

Según el CCN-CERT, la inmensa mayoría de estos ataques se realizaron mediante la cooperación de diferentes programas. A continuación se explican dos ejemplos de ransomware cooperativos y su método de actuación: [48]

- **Emotet, Trickbot y Ryuk.** *Emotet* es un troyano cuyo origen se remonta a 2014. Su principal objetivo era el robo de credenciales bancarias. En la actualidad su ámbito ha evolucionado y se usa en conjunción con el troyano bancario *Trickbot* y el ransomware *Ryuk*.
El método de infección más común es a través de correos electrónicos con documentos ofimáticos manipulados que se encargan de instalar *Emotet*. Este, a su vez, descarga *Trickbot* y se comienzan a expandir por la organización. Una vez se ha conseguido, los ciberdelincuentes evalúan mediante herramientas de control remoto de *Trickbot* si la víctima es adecuada para la instalación de *Ryuk*.
- **Dridex y BitPaymer.** Es un caso similar al anterior. El ataque mediante esta combinación comienza con la infección con Dridex un malware o programa malicioso que reconoce y recolecta información de la red infectada. La finalidad es siempre poder distribuir el ransomware, en este caso BitPaymer, por toda la red para posteriormente pedir el rescate.

Durante el año 2020 y a causa de la pandemia de COVID-19 estos ataques se han realizado principalmente en redes y dispositivos domésticos, industrias, laboratorios de investigación y farmacéuticas.

■ Ataques a sistemas de acceso remoto

Según el CCN-CERT, se ha observado un incremento en el uso de los sistemas de acceso remoto como vías de entrada para atacantes. Esto se debe a que, debido a la pandemia de COVID-19, el teletrabajo se ha extendido de manera exponencial y en casi todos los sectores hacen uso de él. [48]

Los ciberdelincuentes utilizan como baza principal las vulnerabilidades de estos sistemas. Se aprovechan de vulnerabilidades recién descubiertas y que los usuarios no han podido parchear con una actualización, o bien de vulnerabilidades sin solución en ese momento.

También se ha extendido mucho el uso de técnicas de phishing para obtener las credenciales de redes virtuales privadas (VPN), sesiones de escritorio remoto o incluso correos electrónicos.

■ Operaciones de adquisición de información y ciberespionaje

La actual situación geopolítica y el incremento de los países capaces de recopilar inteligencia del ciberespacio ha provocado un aumento de las operaciones de ciberespionaje. Este tipo de acciones las realizan los grupos denominados de *amenaza persistente avanzada* o ATP. Están compuestos por personal especializado con gran cantidad de recursos tanto materiales como económicos y su finalidad es permanecer en la red objetivo el mayor tiempo posible obteniendo información sin ser detectados.[48]

Estos ataques se lanzan sobre industrias, universidades, empresas sanitarias o incluso dispositivos inteligentes.

Cabe destacar en estos últimos que la puerta de entrada suelen ser las comunicaciones inalámbricas como Bluetooth o Bluetooth Low Energy (BLE). Aunque se recomienda desactivar estas comunicaciones siempre que no se usen, muchos usuarios ignoran estos consejos convirtiéndoles en potenciales víctimas. [49]

7.2. Identificación de Activos

Una vez identificadas todas las herramientas y explicado el estado del arte podemos comenzar el análisis.

Para realizarlo, se comenzará por identificar los activos que componen el proyecto. Para ello haremos uso de la herramienta *PILAR Basic*.

Como se ha comentado anteriormente, este programa se basa en la metodología MAGERIT. Según esta metodología, lo primero que se debe realizar es caracterizar los activos utilizando su lista de atributos o, como en este caso la de *PILAR Basic*.

A continuación se expondrán en *cursiva* los atributos, en **[negrita y entre corchetes]** los códigos de identificación de cada activo para *PILAR Basic*, en **negrita** el nombre del activo y a continuación una pequeña descripción en aquellos activos que la necesiten para comprenderlo totalmente.

■ Activos esenciales

- **[essential][info][per]/[sensitive]/[salud]/[estado de salud]** **[INFO-001] Estado de contagio.** Solo puede verlo el usuario de la aplicación móvil.
- **[essential][info][per]/[pseudonymous]** **[INFO-002] Identificadores Contagiados.** Representa los identificadores que envía quien comunica un positivo. Solo tiene acceso a ellos quien administre la base de datos del servidor.
- **[essential][info][per]/[pseudonymous]** **[INFO-003] Identificadores Cliente.** Es el elemento más importante de la aplicación. Nadie puede acceder a ellos ya que se generan en la propia aplicación de forma interna. Es necesario que estén seguros y no se puedan manipular de ninguna forma.

- *[essential]/[info]/[vr]/[classified]/[UC]* **[INFO-004] Documentación Proyecto.** Se trata de todo el trabajo de investigación y diseño de la aplicación y que será presentado al tribunal. En otras palabras, este documento.
- *[essential]/[info]/[vr]/[classified]/[UC] [D] /[backup]* **[INFO-005] Copia de seguridad de documentación proyecto en local.**
- *[essential]/[info]/[vr]/[keys]/[com]/[channel]* **[KEY-001] Clave de cifrado de comunicación con servidor.** Esta clave no puede estar accesible para nadie, ya que es la que garantiza que un ataque de *sniffing* pueda ver los datos que se intercambian entre el servidor y los clientes.
- *[essential]/[info]/[vr]/[D]/[password]* **[PWD-001] Contraseña de acceso a base de datos del servidor.**
- *[essential]/[info]/[vr]/[D]/[source]* **[COD-001] Códigos fuente en local.** Aquí se engloban tanto el código del servidor como el de la aplicación móvil, ya que las medidas de seguridad a las que se deben someter son las mismas.
- *[essential]/[info]/[vr]/[D]/[backup]/[source]* **[COD-002] Códigos fuente en repositorio.** Aquí se engloban, al igual que en el punto anterior, tanto el código del servidor como el de la aplicación móvil, ya que las medidas de seguridad a las que se deben someter son las mismas.
- *[essential]/[info]/[vr]/[D]/[exe]* **[COD-003] Ejecutables.** Como en los puntos anteriores, se consideran tanto el ejecutable del servidor como el de la aplicación móvil, ya que las medidas de seguridad a las que se deben someter son las mismas.
- *[essential]/[service]/[administrative]/[S]/[prov]/[int]* **[SERV-001] Almacenamiento en base de datos.** La información que se almacena son los identificadores contagiados que los clientes envían junto con el código proporcionado por la autoridad sanitaria. De este servicio solo pueden hacer uso los clientes con la aplicación móvil.

■ Aplicaciones Informáticas - Software

- *[essential]/[bp]/[SW]/[prp]* **[SW-001] Aplicación móvil.** Se trata de la aplicación que los usuarios utilizarán en sus dispositivos móviles. Por tanto es accesible a todos los clientes del proyecto.
- *[essential]/[bp]/[SW]/[prp]* **[SW-002] Aplicación servidor.** Es el software que recibe las peticiones, accede a la base de datos y envía los identificadores contagiados a todos los clientes.
- *[essential]/[service]/[administrative]/[SW]/[std]/[dbms]* **[SW-003] MariaDB.** Es el software gestor de base de datos y el que implementa todas las operaciones necesarias para el mantenimiento de la base de datos.

■ Equipamiento - Hardware

- *[HW]/[host]/[data]* **[HW-001] Ordenador Juan.** Es el equipo que actúa como servidor principal y host de la base de datos.
- *[HW]/[backup]/[data]* **[HW-002] Ordenador María.** Es el equipo que actúa como servidor de respaldo. Contiene también una copia de la base de datos.
- *[HW]/[mobile]/[data]* **[HW-003] Dispositivo móvil cliente.** Engloba todos los dispositivos móviles que instalen la aplicación y hagan uso del proyecto.

■ Redes de comunicación

- *[COM]/[LAN]* **[RED-001] Red Local Piso Juan.** Red local utilizada para realizar las simulaciones de comunicación. Es la red principal.
- *[COM]/[LAN]* **[RED-002] Red Local Casa Juan.** Red local utilizada como alternativa y para realización de pruebas.

- **[COM]/[LAN] [RED-003] Red Local Casa María.** Red local utilizada como alternativa y para realización de pruebas.
- **[COM]/[LAN] [RED-004] Red Local Café La Passion.** Red local utilizada para aquellas pruebas en las que se necesitaban dos dispositivos móviles.

■ Soportes de Información

- **[Media]/[electronic]/[disk] [MED-001] Disco duro Juan.**
- **[Media]/[electronic]/[disk] [MED-002] Disco duro María.**
- **[Media]/[electronic]/[san] [MED-003] Canal de Discord.** Almacenamiento en la nube que contiene todas las fuentes y referencias utilizadas para la realización del proyecto. También contiene documentación del proyecto, únicamente es accesible al personal del proyecto.
- **[Media]/[electronic]/[san] [MED-004] Chat de Telegram.** Almacenamiento en la nube que contiene versiones del proyecto. Al igual que lo anterior, únicamente accesible al personal del proyecto.
- **[Media]/[electronic]/[san] [MED-005] Repositorio de la aplicación servidor.** Contiene una copia del proyecto y los pasos realizados hasta la versión final.
- **[Media]/[electronic]/[san] [MED-006] Repositorio de la aplicación móvil.** Contiene una copia del proyecto y los pasos realizados hasta la versión final.
- **[Media]/[electronic]/[san] [MED-007] Overleaf.** Contiene una copia en la nube de este documento. Únicamente accesible al personal del proyecto.

■ Instalaciones

- **[L]/[local] [L-001] Piso Juan.** Local donde se ha desarrollado el proyecto.
- **[L]/[backup] [L-002] Casa Juan.** Local donde se ha desarrollado el proyecto.
- **[L]/[backup] [L-003] Casa María.** Local donde se ha desarrollado el proyecto.
- **[L]/[local] [L-004] Café La Passion.** Local utilizado para la realización de pruebas bluetooth.

■ Personal

- **[P] [P-001] Juan Velázquez García.** Administrador y desarrollador del proyecto.
- **[P] [P-002] María Ruiz Molina.** Administradora y desasorrolladora del proyecto.
- **[P] [P-003] Usuarios.**

7.3. Valoración de los activos

Para determinar el valor de los activos se utilizarán las dimensiones de valoración, características o atributos que hacen valioso un activo.

En primer lugar se definirán las dimensiones tal como especifica la metodología MAGERIT: [\[24\]](#)

- **[D] Disponibilidad.** Propiedad o característica de los activos consistente en que las entidades o procesos autorizados tienen acceso a los mismos cuando lo requieren. [UNE 71504:2008]
- **[I] Integridad de los datos.** Propiedad o característica consistente en que el activo de información no ha sido alterado de manera no autorizada. [ISO/IEC 13335-1:2004]
- **[C] Confidencialidad de la información.** Propiedad o característica consistente en que la información ni se pone a disposición, ni se revela a individuos, entidades o procesos no autorizados. [UNE-ISO/IEC 27001:2007]

- **[A] Autenticidad.** Propiedad o característica consistente en que una entidad es quien dice ser o bien que garantiza la fuente de la que proceden los datos. [UNE 71504:2008]
- **[T] Trazabilidad.** Propiedad o característica consistente en que las actuaciones de una entidad pueden ser imputadas exclusivamente a dicha entidad. [UNE 71504:2008]

Es momento de determinar la escala de valores a utilizar. Según MAGERIT, «Para valorar los activos vale, teóricamente, cualquier escala de valores.»([Portal de Administración Electrónica. MAGERIT v.3 : Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información, 2012](#))

En este caso, al utilizar la herramienta *PILAR Basic*, se tomará la escala que trae el programa. En orden de mayor a menor la escala de valoración es la siguiente:

- **[10] Nivel 10**
- **[9] Nivel 9**
- **[A+] Nivel Alto +**
- **[A] Nivel Alto**
- **[A-] Nivel Alto -**
- **[M+] Nivel Medio +**
- **[M] Nivel Medio**
- **[M-] Nivel Medio -**
- **[B+] Nivel Bajo +**
- **[B] Nivel Bajo**
- **[n.a] No Aplicable**

Finalmente, utilizando estas dimensiones de valoración y esta escala, se obtuvo la siguiente tabla de valoraciones:

▪ Estado de contagio.

- **Disponibilidad:** Se ha valorado con un 10, ya que el usuario no podría saber si está o no contagiado de la enfermedad.
- **Integridad del dato:** Se ha valorado con un 10 porque su modificación no autorizada, podría, por ejemplo, crear un foco de contagio marcando como sanas a personas contagiadas antes de tiempo, llegando a afectar incluso a quienes no utilizan la aplicación.
- **Confidencialidad de la información:** Se ha valorado con un 9, pues el estado de salud es un dato personal cuya difusión debe ser decisión de su propietario, pero su difusión afectaría únicamente a los usuarios de la aplicación.
- **Autenticidad:** Se ha valorado con un 10 debido a que si un atacante pudiera intercambiar este dato entre varios usuarios, se podría crear un foco de contagio al marcar como sanas personas que no lo son. Afectaría incluso a aquellas personas que no utilicen la aplicación.
- **Trazabilidad:** Se ha valorado como no aplicable, ya que este dato está siempre visible en la aplicación y por tanto no es necesario llevar un registro de quién accede a él.

activo / dominio de seguridad	[D]	[I]	[C]	[A]	[T]
[AGA-001] AgavaCovid					
↳ [essential] Activos esenciales					
↳ [INFO-001] Estado de contagio	[10]	[10]	[9]	[10]	n.a.
↳ [INFO-002] Identificadores Contagiados	[10]	[10]	[9]	[10]	[10]
↳ [INFO-003] Identificadores Cliente	[10]	[10]	[9]	[10]	n.a.
↳ [INFO-004] Documentación Proyecto	[B]	[B]	n.a.	[B]	n.a.
↳ [INFO-005] Copia de seguridad de documentación pr	[M-]	[10]	n.a.	[B]	[M]
↳ [KEY-001] Clave de cifrado de comunicación con serv	[10]	[10]	[10]	[10]	[10]
↳ [PWD-001] Contraseña de acceso a base de datos de	[10]	[10]	[10]	[10]	[10]
↳ [COD-001] Códigos fuente en local	[A]	[A]	n.a.	n.a.	n.a.
↳ [COD-002] Códigos fuente en el repositorio	[10]	[10]	n.a.	n.a.	n.a.
↳ [COD-003] Ejecutables	[A]	[10]	n.a.	[10]	n.a.
↳ S [SERV-001] Almacenamiento en Base de Datos	[10]	n.a.	n.a.	[10]	[10]
↳ P [SW-001] Aplicación móvil	[10]	n.a.	n.a.	[10]	n.a.
↳ P [SW-002] Aplicación servidor	[10]	n.a.	n.a.	[10]	n.a.
↳ S [SW-003] MariaDB	[10]	n.a.	n.a.	n.a.	n.a.

Figura 44: Tabla de valoración de activos

■ Identificadores Contagiados.

- **Disponibilidad:** Se ha valorado con un 10, ya que la aplicación no podría funcionar si no están disponibles.
- **Integridad del dato:** Se ha valorado con un 10, ya que cualquier modificación en ellos podría provocar, por ejemplo, que a un usuario contagiado le cambie su estado a no contagiado antes de tiempo, lo que podría generar contagios.
- **Confidencialidad de la información:** Se ha valorado con un 9, ya que, a pesar de ser una información valiosa de la aplicación, estos identificadores son anónimos y conocer, por ejemplo, a quién pertenece es prácticamente imposible.
- **Autenticidad:** Se ha valorado con un 10, ya que si se falsificara el origen de los datos se podría enviar información falsa al servidor.
- **Trazabilidad:** Se ha valorado con un 10, ya que no conocer quién tiene acceso a ellos puede provocar que no se sepa la autoría y/o existencia de una fuga de información.

■ Identificadores Cliente.

- **Disponibilidad:** Se ha valorado con un 10, ya que la aplicación no podría funcionar si no están disponibles.
- **Integridad del dato:** Se ha valorado con un 10, ya que la modificación de este dato podría ocasionar, por ejemplo, que un usuario que haya estado en contacto con un contagiado no reciba el pertinente aviso.
- **Confidencialidad de la información:** Se ha valorado con un 9, pues, a pesar de moverse en un entorno muy amplio, si alguien pudiese coger estos identificadores e introducirlos en otro dispositivo con la aplicación a la otra punta del país, en ese dispositivo se mostraría, en caso de contagio del primero, un contacto de riesgo cuando no ha sido así.
- **Autenticidad:** Se ha valorado con un 10, ya que si alguien consiguiera intercambiar un identificador de otra persona haciéndose pasar como suyo, si esa persona se contagiara, el receptor del identificador no recibiría ninguna alerta. Esto podría generar, si el usuario del dispositivo receptor se hubiera contagiado, un foco de contagio, al no recibir aviso del contacto con un positivo.

- **Trazabilidad:** Se ha valorado como no aplicable puesto que estos identificadores viajan entre los dispositivos y por tanto, sería prácticamente imposible conocer quién ha podido acceder a ellos.

■ Documentación del proyecto.

- **Disponibilidad:** Se ha valorado con un B, ya que el hecho de que no esté disponible no tendría ningún efecto grave en el proyecto. Además existe una copia de seguridad en local, que evitaría los problemas que pudiera generar en términos de disponibilidad.
- **Integridad del dato:** Se ha valorado con un B, ya que la modificación o alteración de este documento no tendría consecuencias graves en el funcionamiento del proyecto. En esta caso la copia de seguridad cubre cualquier problema relacionado con la integridad.
- **Confidencialidad de la información:** Se ha valorado como no aplicable, ya que el documento va a ser de dominio público.
- **Autenticidad:** Se ha valorado con un B puesto que si, por ejemplo, alguien realizara una falsificación de este activo, la información ahí escrita podría ser considerada verdadera por quien la observe y podría afectar a la veracidad e imagen del proyecto aunque no a su funcionamiento.
- **Trazabilidad:** Se ha valorado como no aplicable, ya que el documento va a ser público.

■ Copia de seguridad de documentación proyecto en local.

- **Disponibilidad:** Se ha valorado con un M-, pues a pesar de ser una copia de seguridad, el hecho de no estar disponible no generaría problemas muy graves en el proyecto. Además, hay que tener en cuenta que únicamente se necesitará en caso de que no esté disponible el activo original.
- **Integridad del dato:** Se ha valorado con un 10, pues en el caso de hacer uso de esta copia, el hecho de que se haya modificado haría que este activo no cumpla su función.
- **Confidencialidad de la información:** Se ha valorado como no aplicable, ya que aunque es una copia de seguridad, el documento es el mismo que el original, que es de dominio público.
- **Autenticidad:** Se ha valorado con un B puesto que si, por ejemplo, alguien realizara una falsificación de este activo, la información ahí escrita podría ser considerada verdadera por quien la observe y podría afectar a la veracidad e imagen del proyecto aunque no a su funcionamiento.
- **Trazabilidad:** Se ha valorado con un M, ya que al ser en local, saber quién tiene acceso a estas copias es lo mismo que saber cuántas hay. A pesar de ser importante conocer las copias de seguridad que existen, no conocerlo sería grave pero no vital para el funcionamiento del proyecto.

■ Clave de cifrado de comunicación con servidor.

- **Disponibilidad:** Se ha valorado con un 10 puesto que si no tenemos disponible esta clave la información viajaría expuesta.
- **Integridad del dato:** Se ha valorado con un 10, pues la modificación o borrado de esta, podría dejar a la vista toda la información que intercambia el servidor con los dispositivos.
- **Confidencialidad de la información:** Se ha valorado con 10, ya que si alguien externo a la organización conociera la clave y la difundiera la información intercambiada podría ser descifrada y quedaría expuesta.
- **Autenticidad:** Se ha valorado con un 10 debido a que si, por ejemplo, la clave la hubiese proporcionado una persona ajena al proyecto, podría realizar ataques de escucha y utilizar la clave para descifrar la información.
- **Trazabilidad:** Se ha valorado con un 10, ya que si no se tiene un registro de quién tiene acceso a ella, alguien ajeno al proyecto podría filtrarla al exterior, lo que desembocaría en que las comunicaciones quedarían expuestas.

■ Contraseña de acceso a base de datos del servidor

- **Disponibilidad:** Se ha valorado con un 10, ya que es necesaria para acceder o restaurar la base de datos en caso de que sea necesario.
- **Integridad del dato:** Se ha valorado con un 10, pues si alguien consiguiera alterarla, no se podría acceder a la base de datos. Tampoco podríamos conocer si esta ha sido alterada de ninguna forma.
- **Confidencialidad de la información:** Se ha valorado con un 10, ya que la difusión de esta clave podría dar acceso a la información contenida en la base de datos a personas ajenas al proyecto, generando así una fuga de información.
- **Autenticidad:** Se ha valorado con un 10, ya que, por ejemplo, podría haber sido generada por alguien ajeno al proyecto con fines maliciosos.
- **Trazabilidad:** Se ha valorado con un 10 debido a que si no sabemos quien ha tenido acceso a esta contraseña, podríamos haber obviado el acceso por parte de individuos ajenos al proyecto. Esto podría desembocar en una fuga de información.

■ Códigos fuente en local.

- **Disponibilidad:** Se ha valorado con un A, ya que si, por ejemplo, se detectara un bug o una brecha de seguridad, se debería reparar lo antes posible. Por tanto, la disponibilidad de los códigos es fundamental para este fin. Se debe tener en cuenta que se dispone de un repositorio donde hay copias de los códigos para los casos en los que sea necesario.
- **Integridad del dato:** Se ha valorado con un A, ya que cualquier alteración en el código podría generar que el proyecto no realice las funciones que debe. El hecho de existir las copias en los repositorios con el historial de cambios, nos da un cierto margen para poder detectar la modificación y solucionar el contratiempo relativamente rápido.
- **Confidencialidad de la información:** Se ha valorado como no aplicable puesto que hay copias en el repositorio que es público.
- **Autenticidad:** Se ha valorado como no aplicable puesto que el conocer quién ha originado los códigos fuente no es relevante para el funcionamiento del proyecto.
- **Trazabilidad:** Se ha valorado como no aplicable. Esto es debido a que los repositorios son públicos y el código contenido en ellos debería ser el mismo que en local. Por tanto, no sería posible ni tampoco tendría sentido saber quién ha tenido acceso a ellos.

■ Códigos fuente en repositorio.

- **Disponibilidad:** Se ha valorado con un 10, ya que si, por ejemplo, se detectara un bug o una brecha de seguridad, se debería reparar lo antes posible. Por tanto, la disponibilidad de los códigos es fundamental para este fin.
- **Integridad del dato:** Se ha valorado con un 10. Esto se debe a que al ser la copia de seguridad es necesario que, en caso de cualquier alteración de las copias en local, el código no sea alterado y/o borrado. Esto podría generar un mal funcionamiento del proyecto y obligar a una revisión profunda del código para restaurarlo.
- **Confidencialidad de la información:** Se ha valorado como no aplicable debido a que el repositorio es público y por tanto no hay un ámbito delimitado de donde no debiera salir.
- **Autenticidad:** Se ha valorado como no aplicable puesto que el conocer quién ha originado los códigos fuente no es relevante para el funcionamiento del proyecto.
- **Trazabilidad:** Se ha valorado como no aplicable debido a que el repositorio es público y por tanto no tendría sentido saber quién accede a ellos.

■ Ejecutables.

- **Disponibilidad:** Se ha valorado con un A. Esto se debe a que si bien para poder funcionar el proyecto es necesario tener los ejecutables, en caso de no estar disponibles se podrían generar de nuevo con los códigos fuente.
- **Integridad del dato:** Se ha valorado con un 10, ya que cualquier modificación en este activo provocaría que el proyecto no funcionara correctamente.
- **Confidencialidad de la información:** Se ha valorado como no aplicable porque los ejecutables son los archivos que se necesitan para usar el proyecto y por tanto cualquier usuario actual o potencial tiene acceso a ellos.
- **Autenticidad:** Se ha valorado con un 10 debido a que si el origen de estos es desconocido podrían haber sido alterados para realizar otro tipo de funciones.
- **Trazabilidad:** Se ha valorado como no aplicable, ya que cualquier persona, tanto usuarios como no usuarios, tiene acceso a ellos para hacer uso del proyecto. Por tanto, no tiene sentido tener un registro de acceso a estos activos.

■ Almacenamiento en base de datos.

- **Disponibilidad:** Se ha valorado con un 10, ya que si este servicio no está disponible el proyecto no podría funcionar.
- **Integridad del dato:** Se ha valorado como no aplicable. Esto se debe a que este activo es un servicio, no un dato, y por tanto la integridad del dato no se puede aplicar.
- **Confidencialidad de la información:** Se ha valorado como no aplicable puesto que, al igual que la integridad del dato, esta dimensión de valoración únicamente se aplica a datos y no a servicios como ocurre en este caso.
- **Autenticidad:** Se ha valorado con un 10 puesto que si se permitiera almacenar datos a cualquiera podríamos encontrarnos con datos falsos o intentos de inyecciones de código.
- **Trazabilidad:** Se ha valorado con un 10 porque no saber si se ha hecho uso de este servicio podría inhabilitar la persecución de delitos.

■ Aplicación móvil.

- **Disponibilidad:** Se ha valorado con un 10, ya que si este servicio no está disponible, el proyecto no podría funcionar.
- **Integridad del dato:** Se ha valorado como no aplicable debido a que este activo es un servicio, no un dato y por tanto la integridad del dato no se puede aplicar.
- **Confidencialidad de la información:** Se ha valorado como no aplicable. Esto se debe a que este activo es un servicio y no un dato y en consecuencia la confidencialidad de la información no se puede aplicar.
- **Autenticidad:** Se ha valorado con un 10, ya que si alguien hiciera uso de este servicio por ejemplo, para generar una aplicación móvil manipulada usando un código de contagio robado, ocurrirían falsos avisos a los contactos de esa persona o incluso se llegaría a impedir la comunicación de su contagio a la víctima del robo del código de contagio.
- **Trazabilidad:** Se ha valorado como no aplicable porque la aplicación es de uso público, por tanto no se puede dejar constancia de su uso.

■ Aplicación servidor.

- **Disponibilidad:** Se ha valorado con un 10, ya que si este servicio no está disponible el proyecto no podría funcionar.
- **Integridad del dato:** Se ha valorado como no aplicable puesto que este activo es un servicio, no un dato y por tanto la integridad del dato no se puede aplicar.
- **Confidencialidad de la información:** Se ha valorado como no aplicable ya que el hecho de ser un servicio y no un dato no permite aplicar esta dimensión de valoración.
- **Autenticidad:** Se ha valorado con un 10 debido a que si, por ejemplo, alguien creara una aplicación cliente y se conectase a la aplicación servidor no auténtica, se podría ocasionar una fuga de información.
- **Trazabilidad:** Se ha valorado como no aplicable porque la aplicación es de uso público, por tanto no es posible dejar constancia de su uso.

■ MariaDB.

- **Disponibilidad:** Se ha valorado con un 10, ya que si este servicio no estuviera disponible el proyecto no podría funcionar.
- **Integridad del dato:** Se ha valorado como no aplicable. Esto se debe a que este activo no es un dato, es un servicio y por tanto esta dimensión de valoración no se le puede aplicar.
- **Confidencialidad de la información:** Se ha valorado como no aplicable porque, como se ha mencionado en el punto anterior, este activo es un servicio y esta dimensión de valoración únicamente se puede aplicar a datos.
- **Autenticidad:** Se ha valorado como no aplicable puesto que este activo no ha sido creado en el proyecto y, además, es un software de uso público, no es necesario adquirir, por ejemplo una licencia para hacer uso de él.
- **Trazabilidad:** Se ha valorado como no aplicable debido a que este activo no ha sido creado en el proyecto y, además, es un software de uso público.

7.4. Amenazas

El siguiente paso es identificar las amenazas a las que se pueden ver sometidos los activos del proyecto.

Catálogo de amenazas de PILAR

La herramienta *PILAR Basic* recoge todo tipo de amenazas. La clasificación que utiliza para categorizarlas es la misma que recoge la metodología MAGERIT. Estas categorías son: [24]

- **[N] Desastres naturales.** Son aquellos sucesos que pueden ocurrir sin la intervención directa o indirecta de los seres humanos.
- **[I] De origen industrial.** Son acontecimientos que pueden suceder de manera accidental o deliberada como consecuencia de la actividad industrial por parte de los seres humanos.
- **[E] Errores y fallos no intencionados.** Son aquellos fallos que ocurren accidentalmente y que son causados por las personas.
- **[A] Ataques deliberados.** Al contrario que la categoría anterior, esta se refiere a los fallos que ocurren intencionalmente ocasionados por la acción humana.

Como se ha mencionado en el apartado 7.1.3, es importante tomar un enfoque práctico e identificar únicamente las amenazas más probables, dejando fuera, por ejemplo, meteoritos o tsunamis ya que la probabilidad de que pasen es muy baja o nula por completo.

A continuación se muestra toda la lista de amenazas recogidas por *PILAR Basic*.

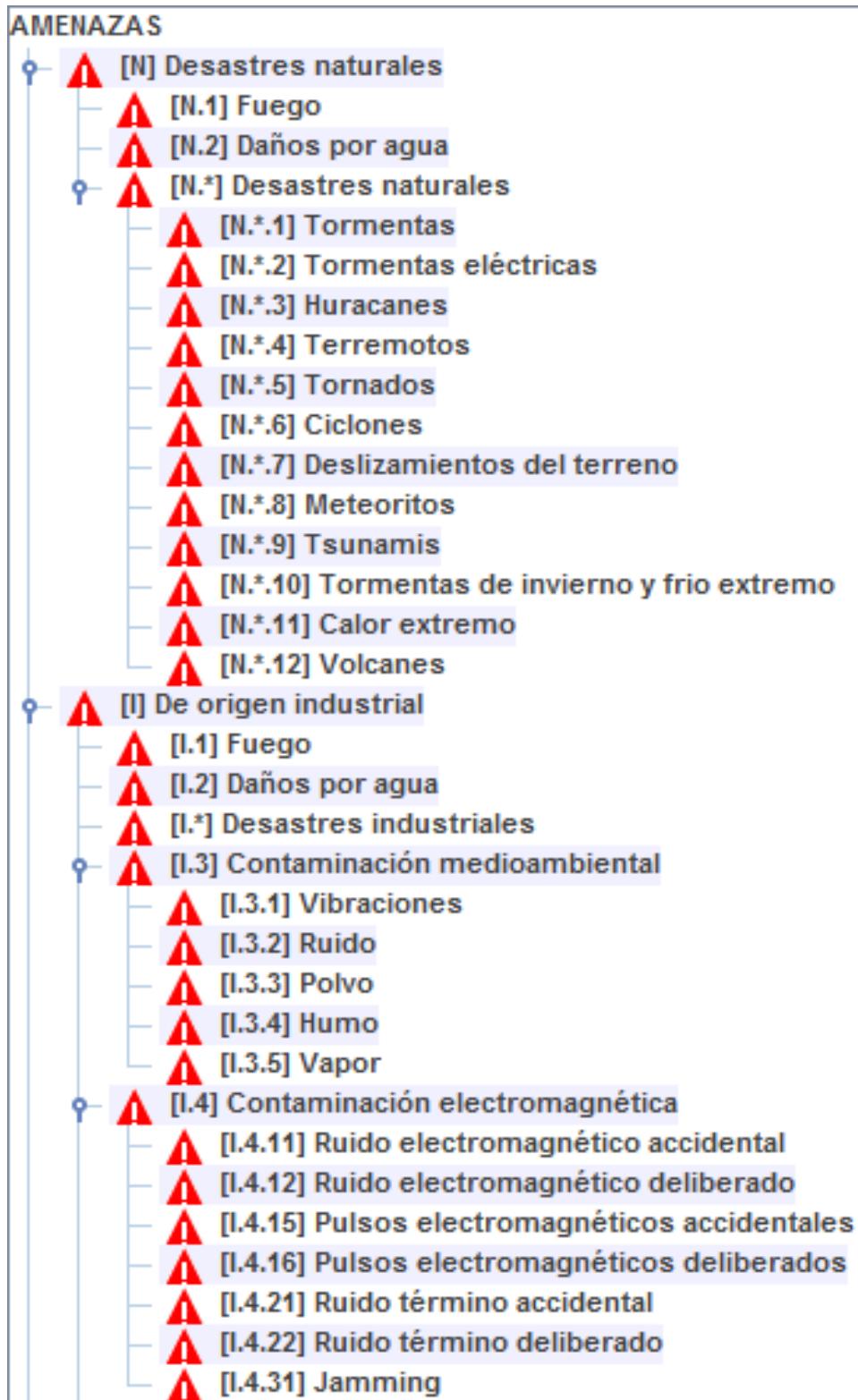


Figura 45: Listado de amenazas de PILAR Basic (Parte 1)

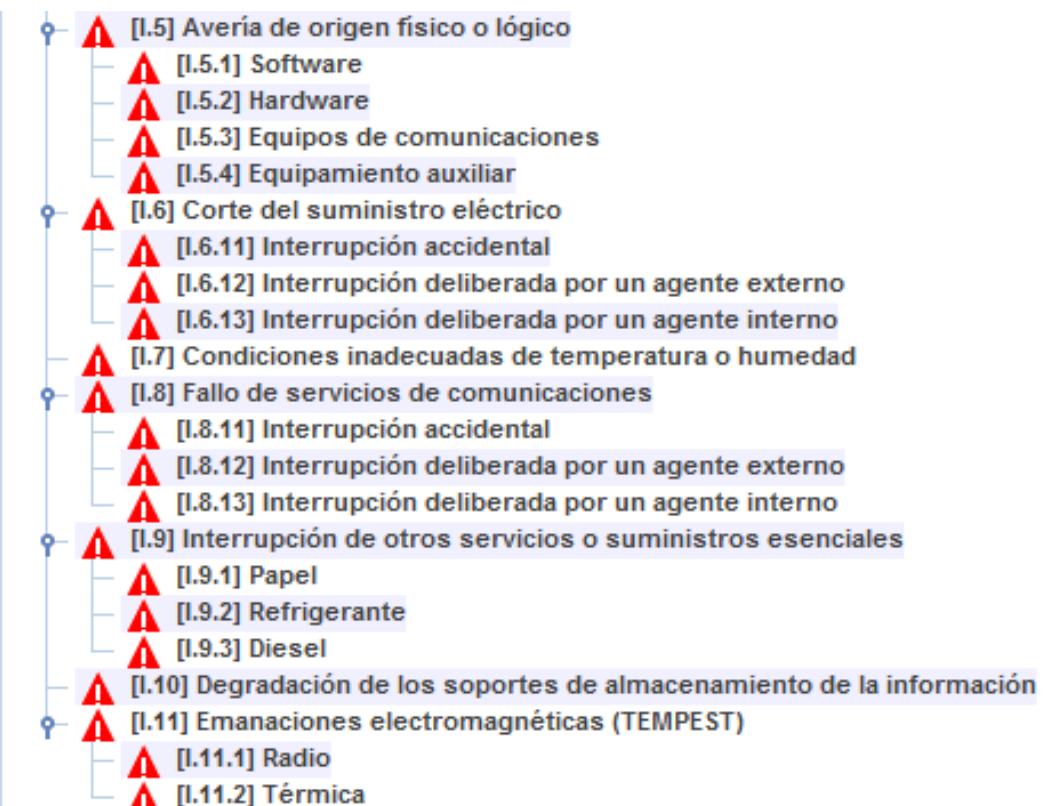


Figura 46: Listado de amenazas de PILAR Basic (Parte 2)

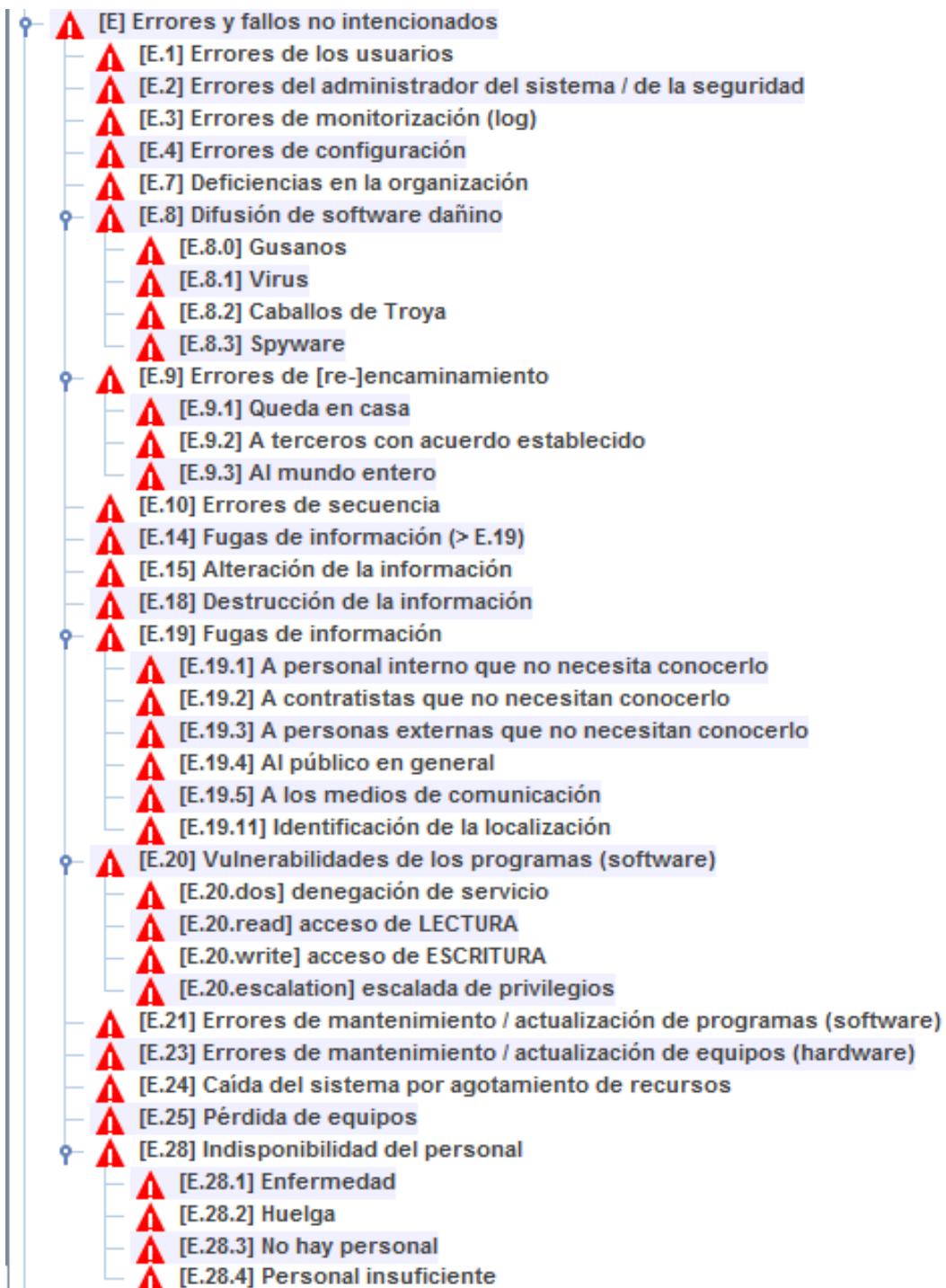


Figura 47: Listado de amenazas de PILAR Basic (Parte 3)

- ⚠ [A] Ataques deliberados
 - ⚠ [A.3] Manipulación de los registros de actividad (log)
 - ⚠ [A.4] Manipulación de los ficheros de configuración
 - ⚠ [A.5] Suplantación de la identidad
 - ⚠ [A.5.1] Por personal interno
 - ⚠ [A.5.2] Por subcontratistas
 - ⚠ [A.5.3] Por personas externas
 - ⚠ [A.6] Abuso de privilegios de acceso
 - ⚠ [A.6.1] Por personal interno
 - ⚠ [A.6.2] Por subcontratistas
 - ⚠ [A.6.3] Por personas externas
 - ⚠ [A.7] Uso no previsto
 - ⚠ [A.7.1] Por personal interno
 - ⚠ [A.7.2] Por subcontratistas
 - ⚠ [A.7.3] Por personas externas
 - ⚠ [A.8] Difusión de software dañino
 - ⚠ [A.8.0] Gusanos
 - ⚠ [A.8.1] Virus
 - ⚠ [A.8.2] Caballos de Troya
 - ⚠ [A.8.3] Spyware
 - ⚠ [A.9] [Re]-encaminamiento de mensajes
 - ⚠ [A.10] Alteración de secuencia
 - ⚠ [A.11] Acceso no autorizado
 - ⚠ [A.11.1] Por personal interno
 - ⚠ [A.11.2] Por subcontratistas
 - ⚠ [A.11.3] Por personas externas
 - ⚠ [A.12] Análisis de tráfico
 - ⚠ [A.12.1] Por personal interno
 - ⚠ [A.12.2] Por subcontratistas
 - ⚠ [A.12.3] Por personas externas
 - ⚠ [A.13] Repudio (negación de actuaciones)
 - ⚠ [A.14] Interceptación de información (escucha)
 - ⚠ [A.14.1] Por personal interno
 - ⚠ [A.14.2] Por subcontratistas
 - ⚠ [A.14.3] Por personas externas
 - ⚠ [A.15] Modificación de la información
 - ⚠ [A.18] Destrucción de la información

Figura 48: Listado de amenazas de PILAR Basic (Parte 4)

- [A.19] Revelación de información
 - [A.19.1] A personal interno que no necesita conocerlo
 - [A.19.2] A contratistas que no necesitan conocerlo
 - [A.19.3] A personas externas que no necesitan conocerlo
 - [A.19.4] Al público en general
 - [A.19.5] A los medios de comunicación
 - [A.19.11] Identificación de la localización
- [A.22] Manipulación de programas
 - [A.22.1] Bombas lógicas
 - [A.22.2] Caballos de Troya
 - [A.22.3] KeyLogger (spyware)
 - [A.22.4] Puertas traseras
 - [A.22.5] Autenticación débil
 - [A.22.6] Se evita la autenticación
- [A.23] Manipulación del hardware
- [A.24] Denegación de servicio
 - [A.24.1] Saturación de los canales de comunicaciones
 - [A.24.2] Saturación de los recursos software
 - [A.24.3] Saturación de los recursos hardware
- [A.25] Robo de equipos
 - [A.25.1] Por personal interno
 - [A.25.2] Por subcontratistas
 - [A.25.3] Por personas externas
- [A.26] Ataque destructivo
 - [A.26.1] Vandalismo
 - [A.26.2] Bomba
 - [A.26.3] Terrorismo
 - [A.26.4] Sabotaje
 - [A.26.5] Guerra
- [A.27] Ocupación enemiga
- [A.28] Indisponibilidad del personal
 - [A.28.1] Enfermedad
 - [A.28.2] Huelga
 - [A.28.3] Absentismo
- [A.29] Extorsión
 - [A.29.1] Ataque desde el exterior
 - [A.29.2] Ataque desde el interior

Figura 49: Listado de amenazas de PILAR Basic (Parte 5)

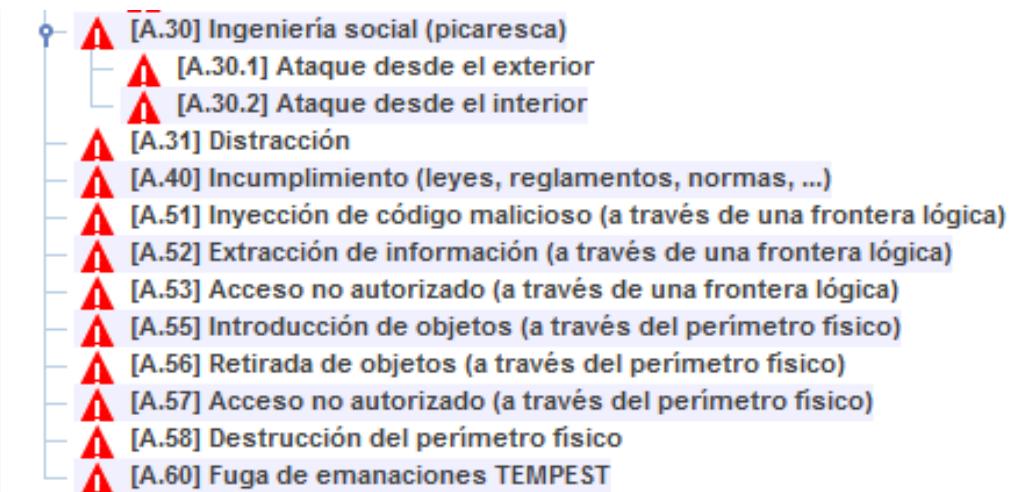


Figura 50: Listado de amenazas de PILAR Basic (Parte 6)

El método a seguir para la identificación de amenazas será comprobar las consideradas por la herramienta, obviar aquellas menos probables y, en caso de ser necesario y haciendo uso del manual de MAGERIT, completarlas. Acto seguido se procederá a establecer a qué dimensiones afectan ([D] Disponibilidad, [I] Integridad, [C] Confidencialidad, [A] Autenticidad y [T] Trazabilidad), escribiéndolas en orden de relevancia de izquierda a derecha. A continuación, se estimará la probabilidad y el impacto siguiendo una escala del 1 al 10 donde 1 sea el valor más bajo y 10 el valor más alto, es decir, se realizará un análisis cuantitativo.

Finalmente, se calculará el riesgo multiplicando la probabilidad por el impacto. Se reflejará el riesgo con ese resultado y con una escala de colores. En ella encontraremos verdes para los riesgos bajos, amarillos para los riesgos medios, naranjas para los riesgos altos y finalmente rojos para los riesgos muy altos.

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Estado de contagio	[A.15] Modificación de la información	[I]	4	10	40
	[A.19] Revelación de información	[C]	9	7	63
Identificadores contagiados	[A.18] Destrucción de información	[D]	2	10	20
	[A.19] Revelación de información	[C]	2	10	20
	[E.19] Fugas de información	[C]	4	10	40
Identificadores cliente	[A.15] Modificación de la información	[I]	2	10	20
	[A.19] Revelación de información	[C]	2	4	8
Documentación del proyecto	[E.15] Alteración de la información	[I]	4	2	8
	[E.18] Destrucción de la información	[D]	2	4	8
	[A.15] Modificación de la información	[I]	4	2	8
	[A.18] Destrucción de la información	[D]	2	4	8
Copia de seguridad de documentación del proyecto en local	[E.15] Modificación de la información	[I]	7	3	28
	[E.18] Destrucción de la información	[D]	2	5	10
	[A.15] Modificación de la información	[I]	4	3	12
	[A.18] Destrucción de la información	[D]	4	4	16
	[A.11] Acceso no autorizado	[C] [I]	8	2	16
	[A.5] Suplantación de la identidad	[C] [A] [I]	1	2	2

Tabla 8: Tabla de amenazas y riesgos (Parte 1)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Clave de cifrado de comunicación con servidor	[E.15] Alteración de la información	[I]	2	10	20
	[E.18] Destrucción de la información	[D]	2	10	20
	[E.19] Fugas de información	[C]	6	10	60
	[A.5] Suplantación de la identidad	[C] [A] [I]	8	10	80
	[A.6] Abuso de privilegios de acceso	[C] [I] [D]	6	10	60
	[A.11] Acceso no autorizado	[C] [I]	8	10	80
	[A.15] Modificación de la información	[I]	8	10	80
	[A.18] Destrucción de la información	[D]	8	10	80
	[A.19] Revelación de la información	[C]	8	10	80
Contraseña de acceso a base de datos del servidor	[E.15] Alteración de la información	[I]	2	10	20
	[E.18] Destrucción de la información	[D]	2	10	20
	[E.19] Fugas de información	[C]	7	10	70
	[A.5] Suplantación de la identidad	[C] [A] [I]	8	10	80
	[A.6] Abuso de privilegios de acceso	[C] [I] [D]	7	10	70
	[A.11] Acceso no autorizado	[C] [I]	8	10	80
	[A.15] Modificación de la información	[I]	7	10	70
	[A.18] Destrucción de la información	[D]	7	10	70
	[A.19] Revelación de la información	[C]	8	10	80
Códigos fuente en local	[E.15] Alteración de la información	[I]	8	7	56
	[E.18] Destrucción de la información	[D]	3	10	30
	[A.11] Acceso no autorizado	[C] [I]	8	5	40
	[A.15] Modificación de la información	[I]	7	7	49
	[A.18] Destrucción de la información	[D]	5	8	40
Códigos fuente en el repositorio	[E.15] Alteración de la información	[I]	6	10	60
	[E.18] Destrucción de la información	[D]	3	10	30
	[A.15] Alteración de la información	[I]	8	10	80
	[A.18] Destrucción de la información	[D]	4	10	40

Tabla 9: Tabla de amenazas y riesgos (Parte 2)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Ejecutables	[E.15] Alteración de la información	[I]	1	7	7
	[E.18] Destrucción de la información	[D]	3	7	21
	[A.15] Alteración de la información	[I]	5	7	35
	[A.18] Destrucción de la información	[D]	4	7	28
Almacenamiento en base de datos	[E.2] Errores del administrador	[D] [I] [C]	7	10	70
	[E.15] Alteración de la información	[I]	7	10	70
	[E.18] Destrucción de la información	[D]	3	10	30
	[E.24] Caída del sistema por agotamiento de recursos	[D]	6	10	60
	[A.6] Abuso de privilegios de acceso	[C] [I] [D]	7	10	70
	[A.7] Uso no previsto	[D] [C] [I]	2	10	20
	[A.11] Acceso no autorizado	[C] [I]	4	10	40
	[A.15] Modificación de la información	[I]	8	10	80
	[A.18] Destrucción de la información	[D]	8	10	80
	[A.24] Denegación de servicio	[D]	8	10	80
Aplicación móvil	[I.5] Avería de origen físico o lógico	[D]	2	10	20
	[E.20] Vulnerabilidades de los programas	[I] [D] [C]	9	10	90
	[E.21] Errores de mantenimiento / actualización de software	[I] [D]	9	10	90
	[A.22] Manipulación de programas	[C] [I] [D]	8	10	80
Aplicación servidor	[I.5] Avería de origen físico o lógico	[D]	2	10	20
	[E.20] Vulnerabilidades de los programas	[I] [D] [C]	9	10	90
	[E.21] Errores de mantenimiento / actualización de software	[I] [D]	8	10	80
	[A.22] Manipulación de programas	[C] [I] [D]	7	10	70

Tabla 10: Tabla de amenazas y riesgos (Parte 3)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
MariaDB	[I.5] Avería de origen físico o lógico	[D]	2	10	20
	[E.8] Difusión de software dañino	[D] [I] [C]	6	10	60
	[E.20] Vulnerabilidades de los programas	[I] [D] [C]	9	10	90
	[E.21] Errores de mantenimiento / actualización de software	[I] [D]	8	10	80
	[A.8] Difusión de software dañino	[D] [I] [C]	8	10	80
	[A.22] Manipulación de programas	[C] [I] [D]	7	10	70
Ordenador Juan	[N.1] Fuego	[D]	1	8	8
	[N.2] Daños por agua	[D]	1	8	8
	[N.*] Desastres naturales	[D]	1	8	8
	[I.1] Fuego	[D]	2	8	16
	[I.2] Daños por agua	[D]	2	8	16
	[I.*] Desastres industriales	[D]	1	8	8
	[I.3] Contaminación mecánica	[D]	1	3	3
	[I.4] Contaminación electromagnética	[D]	1	8	8
	[I.5] Avería de origen físico o lógico	[D]	2	8	16
	[I.6] Corte del suministro eléctrico	[D]	3	8	24
	[E.23] Errores de mantenimiento / actualización de equipos hardware	[D]	8	8	64
	[E.24] Caída del sistema por agotamiento de recursos	[D]	6	8	48
	[E.25] Pérdida de equipos	[D] [C]	6	10	60
	[A.6] Abuso de privilegios de acceso	[C] [I] [D]	8	10	80
	[A.7] Uso no previsto	[D] [C] [I]	9	5	40
	[A.11] Acceso no autorizado	[C] [I]	8	10	80
	[A.23] Manipulación del hardware	[C] [D]	4	10	40
	[A.24] Denegación de servicio	[D]	8	8	64
	[A.25] Robo de equipos	[D] [C]	7	10	70
	[A.26] Ataque destructivo	[D]	2	8	16

Tabla 11: Tabla de amenazas y riesgos (Parte 4)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Ordenador María	[N.1] Fuego	[D]	1	10	10
	[N.2] Daños por agua	[D]	1	10	10
	[N.*] Desastres naturales	[D]	1	10	10
	[I.1] Fuego	[D]	2	10	20
	[I.2] Daños por agua	[D]	2	10	20
	[I.*] Desastres industriales	[D]	1	10	10
	[I.3] Contaminación mecánica	[D]	1	3	3
	[I.4] Contaminación electromagnética	[D]	1	10	10
	[I.5] Avería de origen físico o lógico	[D]	2	10	20
	[I.6] Corte del suministro eléctrico	[D]	3	10	30
	[E.23] Errores de mantenimiento / actualización de equipos hardware	[D]	8	10	80
	[E.24] Caída del sistema por agotamiento de recursos	[D]	6	10	60
	[E.25] Pérdida de equipos	[D] [C]	6	10	60
	[A.6] Abuso de privilegios de acceso	[C] [I] [D]	8	10	80
	[A.7] Uso no previsto	[D] [C] [I]	9	5	40
	[A.11] Acceso no autorizado	[C] [I]	8	10	80
	[A.23] Manipulación del hardware	[C] [D]	4	10	40
	[A.24] Denegación de servicio	[D]	8	10	80
	[A.25] Robo de equipos	[D] [C]	7	10	70
	[A.26] Ataque destructivo	[D]	2	10	20

Tabla 12: Tabla de amenazas y riesgos (Parte 5)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Dispositivo móvil cliente	[N.1] Fuego	[D]	1	10	10
	[N.2] Daños por agua	[D]	2	10	20
	[N.*] Desastres naturales	[D]	1	10	10
	[I.1] Fuego	[D]	1	10	10
	[I.2] Daños por agua	[D]	2	10	20
	[I.*] Desastres industriales	[D]	1	10	10
	[I.3] Contaminación mecánica	[D]	1	3	3
	[I.4] Contaminación electromagnética	[D]	1	10	10
	[I.5] Avería de origen físico o lógico	[D]	2	10	20
	[I.6] Corte del suministro eléctrico	[D]	8	7	56
	[E.23] Errores de mantenimiento / actualización de equipos hardware	[D]	8	10	80
	[E.25] Pérdida de equipos	[D] [C]	9	10	90
	[A.23] Manipulación del hardware	[C] [D]	5	10	50
	[A.25] Robo de equipos	[D] [C]	9	10	90
	[A.26] Ataque destructivo	[D]	6	10	60

Tabla 13: Tabla de amenazas y riesgos (Parte 6)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Red local de piso de Juan	[I.8] Fallo de servicios de comunicaciones	[D]	5	10	50
	[E.2] Errores del administrador	[D] [I] [C]	5	8	40
	[E.24] Caída del sistema por agotamiento de recursos	[D]	5	10	50
	[A.5] Suplantación de la identidad	[C] [A] [I]	5	10	50
	[A.9] Reencaminamiento de mensajes	[C]	8	7	56
	[A.11] Acceso no autorizado	[C] [I]	3	10	30
	[A.12] Análisis de tráfico	[C]	8	1	8
	[A.14] Interceptación de información	[C]	8	7	56
	[A.15] Modificación de la información	[I]	4	10	40
	[A.18] Destrucción de la información	[D]	5	10	50
	[A.24] Denegación de servicio	[D]	8	10	80
Redes locales de pruebas ([RED-002], [RED-003] y [RED-004])	[I.8] Fallo de servicios de comunicaciones	[D]	5	7	35
	[E.2] Errores del administrador	[D] [I] [C]	5	5	25
	[E.24] Caída del sistema por agotamiento de recursos	[D]	5	8	40
	[A.5] Suplantación de la identidad	[C] [A] [I]	5	8	40
	[A.9] Reencaminamiento de mensajes	[C]	8	8	64
	[A.11] Acceso no autorizado	[C] [I]	8	8	64
	[A.12] Análisis de tráfico	[C]	8	1	8
	[A.14] Interceptación de la información	[C]	8	7	56
	[A.15] Modificación de la información	[I]	3	8	24
	[A.18] Destrucción de la información	[D]	3	8	24
	[A.24] Denegación de servicio	[D]	8	8	64

Tabla 14: Tabla de amenazas y riesgos (Parte 7)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Discos duros ([MED-001] y [MED-002])	[N.1] Fuego	[D]	1	10	10
	[N.2] Daños por agua	[D]	1	10	10
	[N.*] Desastres naturales	[D]	1	10	10
	[I.1] Fuego	[D]	1	10	10
	[I.2] Daños por agua	[D]	2	10	20
	[I.*] Desastres industriales	[D]	1	10	10
	[I.4] Contaminación electromagnética	[D]	1	10	10
	[I.5] Avería de origen físico o lógico	[D]	1	10	10
	[I.10] Degradación de los soportes de información	[D]	10	10	100
	[E.15] Alteración de la información	[I]	8	1	8
	[E.18] Destrucción de la información	[D]	5	5	25
	[E.19] Fuga de información	[C]	6	10	60
	[E.23] Errores de mantenimiento / actualización de equipos hardware	[D]	8	10	80
	[E.25] Pérdida de equipos	[D] [C]	6	10	60
	[A.15] Modificación de la información	[I]	6	1	6
	[A.18] Destrucción de la información	[D]	8	5	40
	[A.23] Manipulación del hardware	[C] [D]	6	10	60
	[A.25] Robo de equipos	[D] [C]	7	10	70
	[A.26] Ataque destructivo	[D]	2	10	20

Tabla 15: Tabla de amenazas y riesgos (Parte 8)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Canal de Discord y Chat de Telegram	[E.15] Alteración de la información	[I]	8	8	64
	[E.18] Destrucción de la información	[D]	6	10	60
	[E.19] Fugas de información	[C]	7	10	70
	[A.11] Acceso no autorizado	[C] [I]	5	10	50
	[A.15] Modificación de la información	[I]	5	8	40
	[A.18] Destrucción de la información	[D]	5	10	50
Repositorios de las aplicaciones	[E.15] Alteración de la información	[I]	5	10	50
	[E.18] Destrucción de la información	[D]	3	10	30
	[A.15] Modificación de la información	[I]	7	10	70
	[A.18] Destrucción de la información	[D]	6	10	60
Overleaf	[E.15] Alteración de la información	[I]	7	3	21
	[E.18] Destrucción de la información	[D]	3	5	15
	[A.11] Acceso no autorizado	[C] [I]	5	2	10
	[A.15] Modificación de la información	[I]	4	3	12
	[A.18] Destrucción de la información	[D]	5	5	25
Piso de Juan	[N.1] Fuego	[D]	1	8	8
	[N.2] Daños por agua	[D]	1	8	8
	[N.*] Desastres naturales	[D]	1	8	8
	[I.1] Fuego	[D]	1	8	8
	[I.2] Daños por agua	[D]	2	8	16
	[I.*] Desastres naturales	[D]	1	8	8
	[A.26] Ataque destructivo	[D]	1	8	8
	[A.27] Ocupación enemiga	[D] [C]	1	10	10

Tabla 16: Tabla de amenazas y riesgos (Parte 9)

Activos	Amenazas	Afecta a	Probabilidad	Impacto	Riesgo
Locales de desarrollo secundarios ([L-002] y [L-003])	[N.1] Fuego	[D]	1	10	10
	[N.2] Daños por agua	[D]	1	10	10
	[N.*] Desastres naturales	[D]	1	10	10
	[I.1] Fuego	[D]	1	10	10
	[I.2] Daños por agua	[D]	2	10	20
	[I.*] Desastres industriales	[D]	1	10	10
	[A.26] Ataque destructivo	[D]	1	10	10
Café La Passion	[A.27] Ocupación enemiga	[D] [C]	1	10	10
	[N.1] Fuego	[D]	1	7	7
	[N.2] Daños por agua	[D]	1	7	7
	[N.*] Desastres naturales	[D]	1	7	7
	[I.1] Fuego	[D]	1	7	7
	[I.2] Daños por agua	[D]	2	7	14
	[I.*] Desastres industriales	[D]	1	7	7
Administradores / desarrolladores ([P-001] y [P-002])	[A.26] Ataque destructivo	[D]	1	7	7
	[A.27] Ocupación enemiga	[D] [C]	1	7	7
	[E.19] Fugas de información	[C]	4	10	40
	[E.28] Indisponibilidad del personal	[D]	3	10	30
	[A.28] Indisponibilidad del personal	[D]	3	10	30
Usuarios	[A.29] Extorsión	[C] [I] [D]	1	10	10
	[A.30] Ingeniería social	[C] [I] [D]	1	10	10
Usuarios	[E.19] Fugas de información	[C]	7	8	56

Tabla 17: Tabla de amenazas y riesgos (Parte 10)

7.5. Riesgos de privacidad

A parte de la clasificación de las amenazas que recoge la metodología MAGERIT, *PILAR Basic* dispone de una categoría más, los **riesgos de privacidad [PR]**.

En el año 2016 entró en vigor el nuevo reglamento europeo para la protección y el tratamiento de los datos de las personas físicas, el denominado **RGPD**.

La aparición de este nuevo reglamento ha convertido la protección de los datos en algo primordial para los sistemas de información, razón por la cual *PILAR Basic* incluye esta categoría.

Los riesgos mostrados por la herramienta son aquellos que, basándose en los activos y las características de estos, el propio programa ha considerado para al proyecto.

A continuación se expondrán los riesgos que ha proporcionado la herramienta para este caso y su posible solución:

- **[PR.g1] No facilitar la información en materia de protección de datos o no redactarla de forma accesible o fácil de entender.** La solución a este problema es sencilla. Simplemente habría que redactar una política de privacidad procurando sintetizar lo más posible de manera que no sea muy extensa y utilizando lenguaje común que todo el mundo pueda entender, prescindiendo de tecnicismos.
- **[PR.g2] Tratar datos inadecuados y excesivos para la finalidad del tratamiento.** En este caso se debería realizar un análisis sobre que datos se recogen y ver cuales de ellos son imprescindibles. De esta forma se evitaría la recolección de datos superfluos sin una finalidad concreta.
- **[PR.g3] Carecer de una base jurídica sobre la que se sustenten los tratamientos realizados sobre los datos.** Habría que realizar una revisión sobre la legislación vigente y asegurarse que los tratamientos realizados se ajusten a ella.
- **[PR.g4] Tratar datos personales con una finalidad distinta para la cual fueron recabados.** Para evitar este riesgo se debería realizar una análisis sobre los datos recogidos y su finalidad. Una vez realizado, se redactaría en la política de privacidad, especificando por cada dato recogido su finalidad.
- **[PR.g5] No disponer de una estructura organizativa, procesos y recursos para una adecuada gestión de la privacidad en la organización.** La solución sería crear en el proyecto un equipo destinado únicamente a este fin. Para ello se tendría que contratar expertos en el campo y/o formar al personal en materia de privacidad.
- **[PR.g6] Almacenar los datos por períodos superiores a los necesarios para la finalidad del tratamiento y a la legislación vigente.** Para poner solución a este riesgo se debe establecer el periodo máximo de tiempo durante el cual se van a necesitar cada dato respectivamente. Una vez alcanzado el límite se eliminarían.
- **[PR.g7] Realizar transferencias internacionales a países que no ofrezcan un nivel de protección adecuado.** Es imperativo realizar un estudio sobre el nivel de privacidad de los países en los que se va expandir el proyecto.
- **[PR.g8] No tramitar o dificultar el ejercicio de los derechos de los interesados.** Para solucionarlo se debería indicar con claridad en la política de privacidad el proceso por el cual los usuarios pueden ejercer dichos derechos.
- **[PR.g9] Resolución indebida del ejercicio de derechos de los interesados en tiempo, formato y forma.** Se solucionaría aportando un servicio de atención al usuario con su respectivo contacto redactado en la política de privacidad.

- [PR.g10] **Seleccionar o mantener una relación con un encargado de tratamiento sin disponer de las garantías adecuadas.** Es necesario asegurarse de que la persona encargada del tratamiento de datos sea una persona de confianza y con experiencia suficiente.
- [PR.g11] **Carecer de mecanismos de supervisión y control sobre las medidas que regulan la relación con un encargado el tratamiento.** Para evitar este riesgo se deben estipular claramente la periodicidad de los controles y el modo de supervisión de las medidas establecidas tanto en la política de privacidad así como en el contrato del encargado.
- [PR.g12] **No registrar la creación, modificación o cancelación de las actividades de tratamiento efectuadas bajo su responsabilidad.** La solución a este riesgo pasa por actualizar la política de privacidad siempre que se cambien de alguna manera las actividades de tratamiento.
- [PR.g13] **No llevar a cabo por parte del responsable del tratamiento una evaluación de impacto adecuada en los supuestos detallados por la normativa aplicable.** Para evitar este riesgo se deben realizar análisis de impacto siguiendo rigurosamente una metodología de análisis con el objetivo de eludir cualquier tipo de error en su desarrollo.
- [PR.g23] **Disociación deficiente o reversible que permite la re-identificación de datos.** La solución a este problema pasa por minimalizar los datos necesarios y garantizar una correcta anonimización de los datos proporcionados.
- [PR.g24] **Información no actualizada o incorrecta (pe. registros duplicados con informaciones contradictorias o con campos de datos incorrectos).** Una vez redactada o actualizada la política de privacidad es necesario registrar todos los lugares donde aparezca y asegurarse de modificarla correctamente.
- [PR.g32] **Deficiencias en los protocolos de almacenamiento de los datos personales en formato físico.** Para poner solución a este problema habría que realizar evaluaciones periódicas sobre la conformidad de los protocolos establecidos. Todos los fallos y/o disfunciones encontrados deberían analizarse y corregirse, manteniendo siempre un espíritu crítico.
- [PR.2g] **Obtener un consentimiento dudososo, viciado o inválido para el tratamiento o cesión de datos personales.** Habría que exponer de forma completamente transparente al usuario sobre qué consiente, recogiendo adecuadamente en la política de privacidad.
- [PR.2m] **Accesos no autorizados a datos personales (modificación).** Para evitarlo sería necesario establecer protocolos y medidas de seguridad para controlar de manera firme el acceso a estos datos.
- [PR.2n] **Accesos no autorizados a datos personales (lectura).** Al igual en el caso anterior, para evitarlo sería necesario establecer protocolos y medidas de seguridad para controlar de manera firme el acceso a estos datos.

7.6. Salvaguardas

Una vez identificadas las amenazas y calculado su riesgo se deben establecer las contramedidas para mitigarlos. Los amenazas a mitigar serán aquellas cuyo riesgo sobrepase el valor 50.

1. **Actualización de equipos y hardware.** Un equipo actualizado nos evita amenazas como la **degradación de soportes de información**. Además, puede ofrecer mayor rendimiento que un equipo obsoleto y evitarnos posibles averías por desgaste.
2. **Actualización de herramientas software.** Con esta contramedida hacemos frente a las **vulnerabilidades de los programas**, pues la mayoría tienen solución. También mitiga amenazas como la **manipulación de programas** o la **difusión de software dañino** pues muchos aprovechan las vulnerabilidades de los demás programas para atacar.
3. **Gestionar accesos.** La gestión de accesos es una parte vital de un proyecto. Es necesario recoger quién tiene acceso a qué recurso y cuáles son los privilegios que tiene. Establecer usuarios y contraseñas y limitar los recursos accesibles nos pueden ayudar a mitigar **fugas de información, abusos de privilegios, accesos no autorizados o modificaciones** no deseadas.
4. **Encriptar información de base de datos.** Si la información está encriptada, en caso de un acceso no autorizado, podemos evitar **revelaciones de información confidencial**.
5. **Educar al personal en seguridad.** La educación en materia de seguridad es algo fundamental, y más aún cuando se trata del personal. Contratar cursos para enseñar qué se debe o no hacer nos puede evitar muchos **errores** como los de **administración** de recursos, **escapes de información** o incluso **accesos no autorizados**. También puede llegar a prevenir despistes que puedan desembocar en **pérdidas de equipos, robos, extorsiones**, etc...
6. **Aumento de recursos de respaldo.** La existencia de mayores cantidades de copias de seguridad y equipos de respaldo nos puede ayudar con **caídas del sistema** o con la **modificación y destrucción de información**. Además, esta medida puede ayudar a mitigar los efectos de las **pérdidas o robos de equipos** o las **denegaciones de servicio**.
7. **Educar a los usuarios con respecto la privacidad de sus datos.** En este proyecto las amenazas más difíciles de paliar son aquellas en las que la responsabilidad es en gran parte del usuario. Una pequeña guía o unos consejos sobre los datos de los que dispone en su dispositivo móvil puede ayudar evitar, por ejemplo, despistes que acaben en **pérdidas o robo de su dispositivo**. También se verían afectadas considerablemente **las fugas de información** o incluso **la manipulación de programas y hardware**.
8. **Contratar antivirus para los equipos.** Los antivirus nos pueden ayudar a sortear **software dañino** que pudiera ocasionar **manipulaciones de programas y hardware** o **denegaciones de servicio**. Si bien no siempre se puede evitar, utilizar antivirus y actualizarlos debidamente nos puede ayudar a detectarlos lo antes posible.
9. **Cambiar claves y contraseñas periódicamente.** No tendría sentido establecer claves o contraseñas permanentes, pues son muy sensibles a despistes que podrían revelar mucha información confidencial. Además, en caso de conocer, por ejemplo, una clave de cifrado de comunicaciones, cambiarla haría necesario obtenerla de nuevo para obtener resultados de, por ejemplo, un análisis de tráfico para la **interceptación de información**.
10. **Establecer medidas de seguridad en las redes.** Establecer cifrados, controles de acceso, cortafuegos, capado de puertos vulnerables o tener a disposición redes de respaldo son algunas medidas que nos pueden evitar muchos problemas como el **reencaminamiento de mensajes, la intercepción de información, accesos no autorizados o suplantaciones de identidad**.

11. **Contratar expertos para la gestión de las actualizaciones.** Si contamos con personal experto para mantener los sistemas actualizados podemos mitigar **errores de mantenimiento** o detectarlos y establecer el procedimiento para hacerlos desaparecer.
12. **Establecer cifrados robustos.** Por mucho que utilicemos cifrados para mantener oculta la información de nada sirve si carece de la robustez necesaria. Muchos tipos de cifrado son fácilmente descifrables y por eso debemos utilizar uno de calidad. De esta forma evitaremos problemas como la **interceptación de información**.

8. Conclusiones

Para finalizar este trabajo se van a exponer las conclusiones obtenidas sobre los diferentes aspectos de este trabajo.

Objetivos cumplidos

Al comienzo de este trabajo se han expuesto en la sección *Objetivos* los principales objetivos a cumplir con su desarrollo. Veamos entonces uno a uno si se han logrado:

- **Objetivo 1.** Respecto a la realización de una aplicación prototipo de rastreo de contactos se puede considerar cumplido. Del desarrollo de este trabajo ha nacido AgavaCovid, una aplicación prototipo con mucho potencial de desarrollo.
- **Objetivo 2.** Se ha llevado a cabo una profunda investigación tanto en el campo de protocolos de rastreo de contactos como en los campos de seguridad y privacidad. En ella, se ha podido observar que el ámbito de los protocolos de rastreo de contactos está todavía en proceso de desarrollo. Existen múltiples protocolos, con diferentes perspectivas y cada uno con sus fortalezas y debilidades. Algunos de ellos se han demostrado eficaces y seguros, otros se ha podido observar que no son alternativas viables ya sea por su seguridad, por su privacidad o, incluso ambas a la vez. También se han encontrado protocolos en desarrollo, la mayoría con potencial para ser opciones factibles en un futuro.
En cuanto a la privacidad y seguridad, se ha investigado sobre los organismos, herramientas y leyes que están involucrados en este ámbito en España. También se ha hecho hincapié en los ataques más comunes que puede sufrir un proyecto software en la actualidad.
- **Objetivo 3.** Se ha elaborado un análisis de riesgos y privacidad mediante el uso de la herramienta *PILAR Basic* y la metodología *MAGERIT*. En él se han podido sacar a la luz las amenazas más importantes que puede sufrir este proyecto así como las medidas a tomar para evitar aquellos más peligrosos.
- **Objetivo 4.** Como conclusión de este trabajo se puede apreciar que un proyecto de estas características entraña múltiples riesgos tanto en cuestión de privacidad como en seguridad. Con el fin de ofrecer un buen servicio al usuario, es necesario cubrirlos y establecer fuertes contramedidas.

Trabajo futuro

En esta sección se exponen los puntos a tratar en el futuro del proyecto, pues bien no se han podido llevar a cabo por falta de tiempo o porque se escapan del alcance de este trabajo.

- **Mejoras en el uso de la tecnología Bluetooth.** Hasta ahora el prototipo necesita de interacción activa por parte del usuario para el envío de identificadores por Bluetooth. Esta funcionalidad debería ser implementada tal que funcionase en segundo plano y de manera automática. Además, el alcance de la señal de Bluetooth debería restringirse a 2 metros en lugar de a la distancia predeterminada.
- **Borrado automático de identificadores caducados en la base de datos del servidor.** Este proceso debería ser automatizado para mejorar el rendimiento de la aplicación servidor y evitar el almacenamiento de datos superfluos.

- **Mejor tratamiento de los identificadores.** Hasta ahora, con el fin de controlar las pruebas, los identificadores se rotaban de manera manual. Estos deberían ir rotando de forma automática cada 15 minutos. También, un punto a tener muy en cuenta, es la necesidad de envío de ruido al servidor, pues de otro modo mediante la escucha del canal puede averiguar quién envía códigos de contagio, pues es la única comunicación que hay dirección al servidor. Para ello se implementaría el envío de identificadores falsos, los cuales llegarían igualmente a la base de datos, solo que serían descartados al momento, pues tendrían una fecha superior a 14 días. Con ello, la base de datos los eliminaría al comprobar la existencia de identificadores caducados.
- **Pulir el funcionamiento de la aplicación cliente.** De la aplicación cliente hay varias mejoras que podrían implementarse con el fin de perfeccionar su interacción con el usuario. Estos aspectos a pulir son automatizar el cambio de estado de contagiado a sin contactos, un aviso como notificación móvil en el momento en el que el estado de contagio cambie, y la retroalimentación por parte del servidor en el momento de validar si un código de contagio es o no aceptado.
- **Despliegue de la aplicación servidor en Internet.** Para conseguir que la aplicación funcione a nivel del gran público habría que desplegar la aplicación servidor en la red. Por supuesto, se debería revisar su correcto funcionamiento y, si fuese necesario adaptar el código para ello.
- **Cifrado usando pares de clave pública y privada.** Para la aplicación prototipo se ha utilizado el cifrado simétrico AES-256, pero para una mayor protección de los datos enviados por red, se habría de implementar un cifrado de clave asimétrica, como podría ser RSA. De este modo, al inicio de la conexión TCP, el servidor envía al cliente su clave pública, con la que el cliente cifra la clave simétrica para posteriores comunicaciones. El servidor podría obtener esa clave simétrica descifrando el mensaje con su clave privada. Tras ello usaría la clave simétrica para comunicarse con el cliente.
- **Redacción de política de privacidad.** Como se ha podido observar en el análisis de riesgos de seguridad y privacidad, la mayoría de riesgos en el ámbito de la privacidad tienen como causa la ausencia o la mala redacción de una política de privacidad. Es por ello que para cumplir la normativa se debería redactar una, prestando especial atención a los riesgos ahí expuestos.
- **Implementar las contramedidas extraídas del análisis.** Para que este proyecto sea viable se deberían minimizar los riesgos que surgidos en el análisis. Para lograr paliar aquellos más perjudiciales habría que aplicar las salvaguardas descritas en el apartado anterior.

Como podemos ver, todavía hay muchas líneas de trabajo que pueden ser desarrolladas y mejoradas. Muchas de ellas refieren a la implementación, así como a las salvaguardas deducidas del análisis llevado a cabo con la herramienta PILAR.

Como se ha podido concluir, en este tipo de aplicaciones de control sobre el estado COVID, todavía hay muchos riesgos desde el punto de vista de diseño del protocolo.

Es por ello que una buena idea sería llevar a cabo un enfoque distinto empleando otras tecnologías, como puede ser el uso de tecnologías descentralizadas que permitan al usuario almacenar su propia información, las cuales proporcionan una mejor privacidad de los datos. En este campo encontramos el concepto de la Identidad Autosoberana o SSI (Self-Sovereign Identity). Lo que este concepto viene a decir en líneas generales es que sea el propio usuario el propietario de sus datos, almacenándolos en un monedero virtual o *wallet*, sin recurrir a otras entidades centralizadas. Así cada usuario posee su propia identidad digital.

De esta forma, en el caso, por ejemplo, de tener que presentar tu estado COVID, en lugar de mostrar un pasaporte con toda tu identificación nacional, acreditarías únicamente el estado de vacunación o PCR asociado a tu identidad digital.

Valoración personal

Con este trabajo he podido conocer en profundidad en el campo de los protocolos de rastreo de contactos. Creo firmemente que el uso de ellos puede ayudar en la contención de futuras epidemias y pandemias. Aunque no es mi deseo en absoluto, en un mundo tan sumamente globalizado la probabilidad de que aparezcan es bastante alta. Por esta razón debemos estar preparados y utilizar la tecnología como una aliada para evitarlas.

Otros aspectos en los que me he podido adentrar son la seguridad y la privacidad. Pienso que son aspectos del software a los cuales no se les dedica el suficiente trabajo y dedicación. Si bien es un campo complejo, me ha resultado interesante conocer los estándares que rigen los desarrollos de software, así como los códigos que se utilizan en ellos y las amenazas a las que están expuestos.

Para finalizar me gustaría también hablar sobre el desarrollo en equipo. Siempre he considerado que el trabajo en equipo, sobre todo en el desarrollo y mantenimiento de software, es algo imprescindible ya que es muy difícil que una única persona disponga de todos los conocimientos necesarios para realizar un desarrollo completo. Con este trabajo he podido vivir en primera persona todo lo que implica: Adecuación de horarios, confrontación de ideas, uso de metodologías de trabajo, puntos muertos, toma de decisiones... En este caso, mi compañera María Ruiz Molina ha realizado una labor fantástica tanto en el aprendizaje como en la transmisión de conocimientos. Siempre ha demostrado compromiso, esfuerzo, profesionalidad y compañerismo lo que ha hecho más ameno el desarrollo de este proyecto.

Anexo I: Contenido adjunto

El contenido adjunto a este documento incluye:

- **DatosTFGJuanVelazquezGarcia.zip.** Que contiene los siguientes ficheros.
 - **agavaserver.zip.** Código aplicación servidor.
 - **agavaclient.zip.** Código aplicación cliente.
 - **agavaanalisis.mgr.** Proyecto para programa PILAR Basic. Contraseña: a

Anexo II: Manual de instalación

Requisitos de instalación

Se necesitan los siguientes elementos previamente a realizar la instalación de todo el proyecto.

- Dispositivo móvil con Android versión 10 y Bluetooth.
- Dispositivo diferente al anterior para despliegue del servidor.
- Acceso a una red para realizar comunicaciones por Internet.

Será necesaria la instalación de los siguientes componentes en el dispositivo donde se desplegará el servidor.

- Java JDK-8.
- NetBeans IDE 8.2 para la ejecución del servidor.
- Configuración del código en UTF-8.
- Gestor de base de datos MariaDB versión 10.5.9.
- MySQL Connector/J para realizar la conexión de la base de datos con NetBeans.
- Android Studio. Solo en caso de querer editar algún componente del código de la aplicación cliente.

Será necesaria la instalación de los siguientes componentes en el dispositivo donde se desplegará el cliente.

- Ejecutable de la aplicación.

Instalación del servidor

Se descargará e instalará NetBeans 8.2 desde <https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>.

Esta versión incluye JDK-8, pero de querer instalar este por separado puede hacerse desde <https://www.oracle.com/es/java/technologies/javase/javase-jdk8-downloads.html>.

Tras ello, se descargará el gestor de base de datos MariaDB desde <https://mariadb.org/download/>, en concreto la versión 10.5.9.

Después se ejecuta el gestor y se crea la base de datos con el siguiente comando:

```
CREATE DATABASE agavacovid;
```

Ejecutamos el comando para usar la base de datos:

```
USE agavacovid;
```

A continuación creamos la tabla:

```
CREATE TABLE ids_infectados(id SERIAL NOT NULL, clave_gen VARCHAR(255) NOT NULL, fecha_gen DATE NOT NULL, fecha_rec DATE NOT NULL, PRIMARY KEY (id));
```

Una vez hecho esto, es necesario instalar el conector MySQL Connector/J (descargable desde <https://dev.mysql.com/downloads/connector/j/8.0.html>).

Se descomprime el contenido de lo descargado. El archivo *mysql-connector-java-8.0.12.jar* se debe situar en el mismo directorio que mantiene los archivos comunes de las librerías de JAVA.

Se hace click derecho encima del nombre del proyecto y se selecciona *Properties*.

Tras ello se pulsa en *Libraries - Compile - Add Jar/Folder*. Aquí se selecciona el archivo *mysql-connector-java-8.0.12.jar* y se pulsa *Open* y tras ello *OK*.

Es importante cambiar en el código del archivo *AgavaCovidServer.java* los datos referentes a usuario y contraseña por los propios.

```
AgavaPreferences.setCredentials("TU_USUARIO", "TU_CONTRASEÑA");
```

Instalación del cliente

Se descargará e instalará Android Studio desde https://developer.android.com/studio?hl=es&gclid=EAIAIQobChMIqr2blNjq8QIVxQwGAB27WQ96EAAYASAAgKFv_D_BwE&gclsrc=aw.ds. Después se abrirá el proyecto *agavaclient.zip*.

A continuación habrá que editar en el código de la aplicación la dirección IP del ordenador con el servidor, así como, de ser necesario, los puertos con los que se comunicará en el archivo *AgavaSocket.java* que se encuentra en el paquete *sockets*. Tras realizar las pertinentes modificaciones, se selecciona la siguiente opción en el menú superior con el fin de generar un ejecutable de la aplicación. Este ejecutable lo descargaremos en nuestro dispositivo y finalmente lo instalaremos.

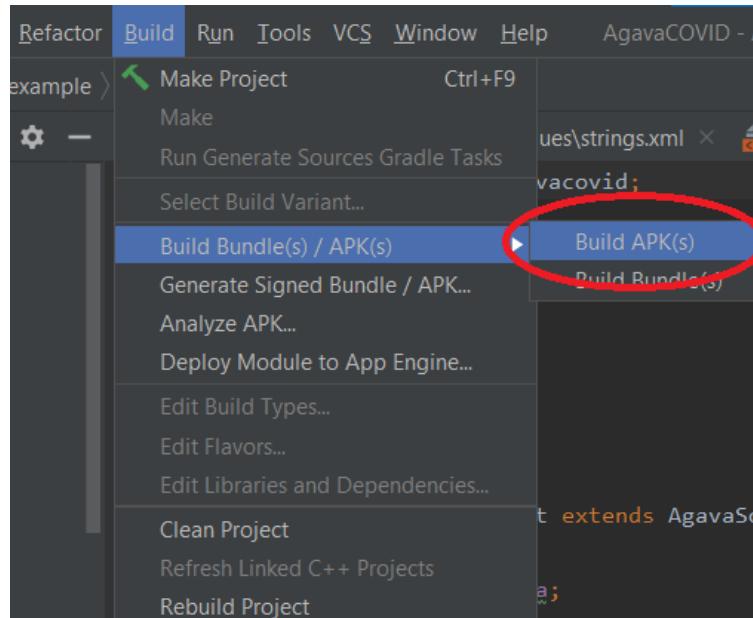


Figura 51: Build APK

Anexo III: Manual de usuario

Aplicación servidor

Para el uso de esta aplicación solo debemos tener en cuenta ciertos detalles:

- **Comprobar el estado del firewall.** En el equipo donde vayamos a ejecutar la aplicación debemos comprobar que el firewall no filtre los puertos que utiliza el programa. Esto se puede comprobar fácilmente al ejecutar el proyecto, pues a la aplicación móvil no le llegarán mensajes si efectivamente se realiza este filtrado. La solución pasa por desactivarlo mientras se lleva a cabo la ejecución. Algunos antivirus, gestionan este filtrado y permitir o no el paso según su configuración. En este caso debemos explorar nuestro antivirus y configurarlo para que permita las comunicaciones.
- **Comprobar el uso de los puertos.** También puede ocurrir que algún otro software tenga en uso los puertos utilizados por la aplicación. Recomendamos revisar si los puertos 3327, 3384 y 4445 están uso.

Para ejecutar la aplicación únicamente debemos abrir Netbeans y ejecutar el archivo *AgavaCovidServer.java* localizado en el paquete *agavacovidserver*.

Aplicación Cliente.

Bienvenid@ a AgavaCovid, tu nueva aplicación de rastreo de contactos. Antes de empezar, nos gustaría darte las gracias por confiar en nosotros para la protección de tu salud y la de tus conocidos.

¿Qué es AgavaCovid?

Como ya sabrás, AgavaCovid es una aplicación de rastreo de contactos pero, ¿eso qué quiere decir?

AgavaCovid permite conocer si has tenido contacto con otro usuario de la aplicación contagiado gracias a la tecnología Bluetooth. Al cruzarte con esa persona vuestras aplicaciones registran el uno al otro anónimamente. En caso de que uno de los dos se contagie, mediante el uso de un código proporcionado por la autoridad sanitaria el infectado podrá comunicar su contagio y la aplicación automáticamente te avisará en caso de haber estado en contacto.

Primeros pasos.

Para abrir la aplicación solo debemos pulsar en el icono con el título AgavaCovid.

Una vez abierta nos aparecerá un pequeño diálogo donde nos preguntará si queremos dar nuestro permiso para utilizar Bluetooth. Para que la aplicación funcione correctamente debemos dar a permitir. Acto seguido, si nos fijamos en la parte superior de nuestro teléfono veremos que nos aparecerá el ícono de Bluetooth encendido.

Pantalla principal.

En la pantalla principal nos encontramos con cuatro elementos. En la parte superior encontramos una imagen con las mascotas de la aplicación Aga y Gava. En esta versión prototipo, al pulsar este botón realizamos el intercambio de información entre dispositivos (en la versión final esto se haría de forma automática).

Justo debajo encontramos un recuadro con nuestro estado de contagio. En caso de no estar contagiado y sin contactos contagiados, aparecerá en verde; en caso de un contacto contagiado, en naranja; y en caso de estar contagiado, en rojo. Si pulsamos en él nos aparecerá una pantalla con unos consejos que cambiarán dependiendo de nuestro estado de contagio.

Si volvemos a la pantalla principal vemos un botón azul que dice *Comunica tu positivo*. Este botón nos da a acceso a un formulario para comunicar nuestro positivo en caso de estar contagiados.

Finalmente, en la parte de abajo encontramos tres botones con los títulos *Principal*, *Información* y *Ajustes de Idioma*. Si pulsamos sobre ellos cambiaremos de pantalla.

Pantalla de información

En la pantalla de información encontramos la política de privacidad y el compromiso con el usuario.

Pantalla de Ajustes de Idioma

En primer lugar encontramos una lista con los diferentes idiomas para los que la aplicación está disponible. Si pulsamos sobre uno de ellos veremos que se oscurecerá, significando que está seleccionado.

En la parte de abajo encontramos un botón de confirmación que al pulsar cambiará el idioma al seleccionado en la lista anterior.

¿Cómo comunico que estoy contagiado?

1. Nos colocamos en la **Pantalla Principal** y pulsamos el botón *Comunica tu positivo*.
2. En la pantalla del formulario encontramos dos campos la fecha y el código de contagio.
3. En caso de conocer alguna, introduciremos o bien la fecha de inicio de síntomas o bien la fecha de toma de muestra para diagnóstico. Para ello simplemente tendremos que pulsar sobre el campo y nos aparecerá un calendario. Ahí nos saldrán en resaltado las fechas de los últimos 14 días para poder seleccionar una de ellas. Pinchamos en una de ellas y pulsamos en *Aceptar*. Una vez seleccionemos la fecha, nos aparecerá escrita en formato *año-mes-día*.
4. Tras esto pulsamos sobre el campo *Introduzca el código*. Introducimos el número proporcionado por la autoridad sanitaria. En este caso al ser un prototipo a modo de simulación hay disponibles únicamente estos códigos válidos 123456789012, 273384273384 y 133713371337.
5. Si el código es correcto al pulsar en *Aceptar* nos saldrán dos nuevos botones para confirmar el envío. En caso de querer rectificar pulsaremos *Cancelar* y modificaremos los datos que sean necesarios. Si está todo bien pulsamos en *Aceptar*.
6. Podremos apreciar en la pantalla principal que nuestro estado de contagio habrá cambiado a *Contagiado*.

Anexo IV: Diccionario

RGPD

«El Reglamento General de Protección de Datos (RGPD) es el reglamento europeo relativo a la protección de las personas físicas en lo que respecta al tratamiento de sus datos personales y a la libre circulación de estos datos. Entró en vigor el 24 de mayo de 2016 y fue de aplicación el 25 de mayo de 2018, dos años durante los cuales las empresas, las organizaciones, los organismos y las instituciones se fueron adaptando para su cumplimiento. Es una normativa a nivel de la Unión Europea, por lo que cualquier empresa de la unión, o aquellas empresas que tengan negocios en la Unión Europea, que manejen información personal de cualquier tipo, deberán acogerse a ella. Las multas por el no cumplimiento del RGPD pueden llegar a los 20 millones de euros.» ([Wikipedia. Reglamento General de Protección de Datos, 2021](#))

ENS

«En el ámbito de la Administración Electrónica española, el Esquema Nacional de Seguridad (ENS) tiene por objeto establecer la política de seguridad en la utilización de medios electrónicos y está constituido por principios básicos y requisitos mínimos que permitan una protección adecuada de la información. Dicho esquema se regula en Real Decreto 3/2010, de 8 de enero, y fue establecido anteriormente en el artículo 42 de la Ley 11/2007, de 22 de junio, de acceso electrónico de los ciudadanos a los Servicio Públicos, que fue modificado por el Real Decreto 951/2015 para actualizarlo a la luz de la experiencia obtenida en su implantación, de la evolución de la tecnología y las ciberamenazas y del contexto regulatorio internacional y europeo.» ([Wikipedia. Esquema Nacional de Seguridad, 2021](#))

COVID-19

«La enfermedad por coronavirus de 2019, más conocida como COVID-19 es una enfermedad infecciosa causada por el virus SARS-CoV-2. Produce síntomas similares a los de la gripe o catarro, entre los que se incluyen fiebre, tos, disnea, mialgia y fatiga. En casos graves se caracteriza por producir neumonía, síndrome de dificultad respiratoria aguda, sepsis y choque séptico que conduce a cerca de 3,75 % de los infectados a la muerte según la OMS.18» ([Wikipedia. COVID-19, 2021](#))

Identificador efímero

Elemento que se emplea para identificar al usuario únicamente. Son únicos con el fin de evitar colisiones. Son generados de manera aleatoria a partir de unas semillas.

Su duración está delimitada por un determinado periodo de tiempo y es sucedido por otro. Esto es así porque en el contexto de esta aplicación, es necesario determinar el periodo temporal en el cual se ha mantenido el contacto entre dos usuarios. De esta manera se dificulta el seguimiento de un usuario ya que el identificador cambia transcurrido dicho tiempo.

En el tipo de aplicaciones como la desarrollada en este trabajo, es fundamental que los identificadores no revelen información personal y/o privada de los usuarios.

SDK

«Un kit de desarrollo de software (en inglés, software development kit o SDK) es generalmente un conjunto de herramientas de desarrollo de software que permite a un desarrollador de software crear una aplicación informática para un sistema concreto, por ejemplo ciertos paquetes de software, entornos de trabajo, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etcétera.» ([Wikipedia. Kit de desarrollo de software, 2020](#))

IMEI

IMEI significa International Mobile Equipment Identity, y es un identificador único que tiene cada teléfono móvil. El código consta de cuatro partes: TAC o Type Allocation Code (los primeros dos indican el RBI o Reporting Body Identifier, es decir, la organización encargada de regular el teléfono), FAC o Final Assembly Code (indica el fabricante), Número de serie, Código verificador (verifica que el código sea correcto y no haya habido errores).

Dirección MAC

«En las redes de computadoras, la dirección MAC (siglas en inglés de Media Access Control) es un identificador de 48 bits (6 bloques de dos caracteres hexadecimales [8 bits]) que corresponde de forma única a una tarjeta o dispositivo de red. Se la conoce también como dirección física, y es única para cada dispositivo. Está determinada y configurada por el IEEE (los últimos 24 bits) y el fabricante (primeros 24 bits)». ([Wikipedia. Dirección MAC, 2021](#))

Dirección MAC de BlueTooth

Se trata de la dirección identificadora de cada dispositivo para establecer conexiones BlueTooth. Estas están conformadas por 12 caracteres hexadecimales.

BlueTooth

«Bluetooth es una especificación industrial para redes inalámbricas de área personal (WPAN) creada por Bluetooth Special Interest Group, Inc. que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles.
- Eliminar los cables y conectores entre estos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan esta tecnología pertenecen a sectores de las telecomunicaciones y la informática personal, como teléfonos móviles, computadoras portátiles [...] o cámaras digitales». ([Wikipedia. Bluetooth, 2021](#))

WiFi

«El wifi (escrito también wi fi) es una tecnología que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con wifi (tales como ordenadores personales, teléfonos, televisores, videoconsolas, reproductores de música, etcétera) pueden conectarse entre sí o a Internet a través de un punto de acceso de red inalámbrica». ([Wikipedia. WiFi, 2021](#))

IP

«La dirección IP es un conjunto de números que identifica, de manera lógica y jerárquica, a una interfaz en la red (elemento de comunicación/conexión) de un dispositivo (computadora, laptop, teléfono inteligente) que utilice el protocolo (Internet Protocol) o, que corresponde al nivel de red del modelo TCP/IP. La dirección IP no debe confundirse con la [dirección MAC](#)». ([Wikipedia. Dirección IP, 2021](#))

BLE/BlueTooth Low Energy

«Bluetooth Low Energy (Bluetooth LE, coloquialmente BLE) es una tecnología de red de área personal [...] destinada a aplicaciones novedosas en el cuidado de la salud, fitness y beacons, seguridad y las industrias de entretenimiento en el hogar. Comparado con el Bluetooth clásico, Bluetooth Low Energy está diseñado para proporcionar un bajo consumo de energía, manteniendo un rango de alcance de comunicación similar». ([Wikipedia. Bluetooth de baja energía, 2020](#))

Semillas generadoras

Estas consisten en dos elementos, la clave generadora, y la fecha generadora.

Claves generadoras

Consiste en una clave generada a partir de una secuencia binaria, la cual es subdividida en n claves generadoras que se emplean a lo largo del día. Dicha secuencia es obtenida de manera pseudoaleatoria a partir de metodologías criptográficas. Estas claves se emplean para generar cada uno de los identificadores efímeros.

Fechas generadoras

Es la fecha que indica el fragmento a seleccionar de la clave generadora. Estas fechas siempre van de 15 en 15 minutos desde las 00:00:00. Cada 15 minutos indica se selecciona el fragmento siguiente a utilizar, el cual determina el identificador efímero de ese periodo de tiempo.

Paradigma

«Para la Ingeniería de Software el paradigma es una agrupación de métodos, herramientas y procedimientos con el fin de describir un modelo.» ([Heli Sulbaran Sistemas. Paradigmas en el desarrollo de software, 2014](#))

Android

«Android es un sistema operativo móvil basado en núcleo Linux y otros software de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes (Wear OS), automóviles (Android Auto) y televisores (Android TV)». ([Wikipedia. Android, 2021](#))

iOS

«iOS es un sistema operativo móvil de la multinacional Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), después se ha usado en dispositivos como el iPod touch y el iPad. Apple no permite la instalación de iOS en hardware de terceros». ([Wikipedia. iOS, 2021](#))

Unix Epoch Time

Unix Epoch Time es el nombre que recibe las 00:00:00 UTC del 1 de enero de 1970, que es la fecha que la mayoría de dispositivos informáticos toman como referencia para empezar a contar el tiempo. Para ello se utiliza el número de milisegundos que han transcurrido desde esta fecha.

AES

«Advanced Encryption Standard (AES), también conocido como Rijndael (pronunciado Rain Doll.^{en} inglés), es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, creado en Bélgica. [...] El cifrado fue desarrollado por dos criptólogos belgas, Joan Daemen y Vincent Rijmen. [...] AES tiene un tamaño de bloque fijo de 128 bits y tamaños de llave de 128, 192 o 256 bits. [...] La mayoría de los cálculos del algoritmo AES se hacen en un campo finito determinado. AES opera en una matriz de 4×4 bytes, llamada state». ([Wikipedia. Advanced Encryption Standard, 2021](#))

Clave AES

Se tratan de claves simétricas empleadas en el algoritmo criptográfico por bloques [AES](#).

Dependiendo del algoritmo AES utilizado, AES-128, AES-192 o AES-256, la longitud de esta será de 128, 192 o 256 bits respectivamente (como indica el nombre de la variante AES).

Durante el cifrado, cuando el texto original es más largo que la clave y cifrados los bloques que abarcaba la longitud de la clave, esta se expande empleando una serie de operaciones con el fin de cifrar los bloques restantes del texto original. Esta expansión se realiza mediante operaciones de rotación, sustitución y XOR.

PUID

Identificador de 128 bits generado de manera pseudoaleatoria en el protocolo PEPP-PT/PEPP. El servidor proporciona uno a cada nuevo usuario registrado con el fin de identificarlos de manera única.

EBID

Un EBID o Ephemeral Bluetooth ID es la implementación de identificador efímero que utiliza el protocolo PEPP-PT/PEPP.

Se trata de identificadores que se intercambian entre usuarios vía BlueTooth con el fin de registrar con quién se ha mantenido contacto.

A diferencia de los [PUIDs](#), como puede verse, los EBIDs sirven para identificar a los usuarios entre sí, mientras que los PUIDs lo hacen frente al servidor.

HMAC-SHA256

Se trata de un código de autentificación de mensajes en clave hash que utiliza para calcular dichos resúmenes hash SHA-256. Puede servir para comprobar la integridad de los datos (estos no han sido modificados), la autenticación del mensaje (el emisor es quien dice ser) o la generación pseudoaleatoria de cadenas de bits (al generar difusión y confusión en la transformación de la entrada).

En el caso de DP-3T, este algoritmo se emplea para generar de manera pseudoaleatoria parte del identificador secreto S_EphID(BK).

Broadcast

«En Informática, la difusión amplia, difusión ancha o broadcast, es una forma de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo». ([Wikipedia. Difusión amplia, 2020](#))

IV

«En criptografía, un vector de inicialización (conocido por sus siglas en inglés IV) es un bloque de bits que es requerido para permitir un cifrado en flujo o un cifrado por bloques, en uno de los modos de cifrado, con un resultado independiente de otros cífrados producidos por la misma clave. El tamaño del IV depende del algoritmo de cifrado y del protocolo criptográfico y a menudo es tan largo como el tamaño de bloque o como el tamaño de la clave.» ([Wikipedia. Vector de inicialización, 2019](#))

Upload What You Observed

Paradigma utilizado en protocolos de rastreo de contactos donde los datos que se envían al servidor son los identificadores recibidos mediante intercambios. De este modo el servidor envía a los clientes una lista de identificadores que han estado en contacto con un contagiado. Así, cada cliente compara la lista recibida con sus propios identificadores.

Upload What You Sent

Paradigma empleado en protocolos de rastreo de contactos donde los datos enviados al servidor son los identificadores del propio usuario. De este modo, el servidor envía a los clientes una lista de identificadores contagiados, la cual cada cliente compara con aquellos que ha recibido.

PSI-CA

La PSI-CA o Private Set Intersection Cardinality es una técnica criptográfica de cálculo multipartida segura (o protocolo de preservación de privacidad) que permite a un emisor y a un receptor computar la cardinalidad de la intersección entre sus conjuntos sin revelar más información al otro. De esta forma se preserva la privacidad del resto de información relacionada, pues solo se revelan los elementos contenidos en la intersección. [55]

Firmas Digitales Ciegas

«La firma digital ciega es un protocolo de firma digital que permite a una persona obtener un mensaje con una firma o sello otorgados por otra entidad para que pueda ser presentada ante terceros, sin necesidad de revelarle a esta información del contenido específico del mensaje.

La principal motivación que tuvo su creador David Chaum fue que cada vez que se llama por teléfono, se compra un producto usando una tarjeta de crédito, se suscribe a una revista o paga algún impuesto, esa información va a parar a una base de datos en algún lugar, lo que trasgreden nuestro derecho a privacidad». ([Wikipedia. Firma digital ciega, 2020](#))

Diffie-Hellman Algorithm

Se trata del primer algoritmo de clave pública, creado en 1976 por W. Diffie y M. Hellman. Se emplea para la distribución de claves y no de mensajes largos debido a que su coste computacional aumenta con el tamaño del mensaje a cifrar. Se aprovecha de la dificultad para calcular logaritmos discretos en un campo finito y emplea funciones matemáticas de la forma

$$g^a \text{mod}(p)$$

donde p es un número primo grande y a un entero.

Handshake

«El establecimiento de comunicación (del inglés handshake, literalmente apretón de manos) es utilizado en tecnologías informáticas, telecomunicaciones, y otras conexiones para establecer automáticamente una negociación entre pares que establece de forma dinámica los parámetros de un canal de comunicación entre ellos antes de que comience la comunicación normal por el canal. De ello se desprende la creación física del canal y precede a la transferencia de información normal.» ([Wikipedia. Establecimiento de comunicación, 2021](#))

Entropía de Shannon

La Entropía de Shannon en el contexto de la teoría de la información es una medida del ratio al que la información es producida por una fuente de datos. Los símbolos con menor probabilidad aportan más información y los que aparecen con mayor frecuencia, menos. Puede ser utilizada para detectar cuándo esos datos son más o menos estructurados. El máximo es 8, representando datos no estructurados “aleatorios”. Datos encriptados o que corresponden a un resumen hash deben poseer una entropía superior a 7.5.

TOR

«Tor (sigla de The Onion Router [...]) Es un proyecto cuyo objetivo principal es el desarrollo de una red de comunicaciones distribuida de baja latencia y superpuesta sobre internet, en la que el encaminamiento de los mensajes intercambiados entre los usuarios no revela su identidad, es decir, su dirección IP (anonimato a nivel de red) y que, además, mantiene la integridad y el secreto de la información que viaja por ella». ([Wikipedia. Tor \(red de anonimato\), 2021](#))

Nerd-attack

Aunque las aplicaciones DP-3T están hechas para recolectar la mínima cantidad de los datos, el código es abierto. Un *Nerd-attack* consiste en que un atacante puede elaborar sus propios clientes DP-3T donde se recolecten más datos como la geolocalización o información del mensaje BlueTooth.

Militia-attack

Se trata de un *Nerd-attack* con más fases, consistentes en vender los datos recolectados, sobre todo los que identifiquen a los infectados, a milicias organizadas.

Ataque de Paparazzi

Ataque consistente en la recolección de información que viaja de la aplicación al servidor y viceversa con el fin de encontrar identificadores infectados de personajes públicos. En definitiva es un ataque de escucha algo más sofisticado, donde el objetivo está predefinido.

Ataque de inyección de código

Consiste en la inserción de código malicioso que modifica el software para otros fines no intencionados. Dicha inserción se aprovecha de vulnerabilidades en entradas de datos, o accesos a puertos mal protegidos del servidor, para provocar un desbordamiento de pila, lo que permite acceder a zonas indebidas del almacenamiento o código del sistema.

Ataque de enlace/Phishing

«Consiste en la emisión masiva de correos electrónicos a usuarios. Estos correos suplantan a entidades de confianza (ejemplo bancos) y persiguen el engaño del usuario y la consecución de información. Por ejemplo en el mensaje se incluyen enlaces a dominios maliciosos. Para camuflar estos enlaces es habitual que el texto del enlace sea la URL correcta, pero el enlace en sí apunte al sitio malicioso». ([Wikipedia. Phishing, 2020](#))

En el tipo de aplicaciones como la desarrollada en este trabajo, el objetivo es evitar que dichos enlaces maliciosos sean insertados en la aplicación o dispuestos al usuario mediante otras vías haciendo pasar por entidades sanitarias.

Objeto

«En el paradigma de programación orientada a objetos (POO, o bien OOP en inglés), un objeto es un ente orientado a objetos (programa de computadoras) que consta de un estado y de un comportamiento, que a su vez constan respectivamente de datos almacenados y de tareas realizables durante el tiempo de ejecución. Un objeto puede ser creado instanciando una clase, como ocurre en la programación orientada a objetos, o mediante escritura directa de código y la replicación de otros objetos, como ocurre en la programación basada en prototipos». ([Wikipedia. Objeto \(programación\), 2021](#))

HTTP

«HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web.

Clientes y servidores se comunican intercambiando mensajes individuales. [...] Los mensajes que envía el cliente, normalmente un navegador Web, se llaman peticiones, y los mensajes enviados por el servidor se llaman respuestas». ([Mozilla. Generalidades del protocolo HTTP, 2020](#))

Protanopia

«La protanopia es la carencia de sensibilidad al color rojo, una disfunción visual relacionada con la percepción del color. Se denomina también dicromacia roja. [...] Por tanto, los individuos que sufren protanopia padecen una pérdida clara de sensibilidad a la luminosidad del extremo rojo del espectro cromático». ([Wikipedia. Protanopia, 2020](#))

Deuteranopia

«La deuteranopia o deuteranopsia es una disfunción visual consistente en alteración para la percepción del color.

Los conos de la retina responsables de la recepción de luz con longitud de onda correspondiente al color verde están ausentes o no son funcionales. Por tanto existe una deficiencia a la hora de discriminar entre verde y rojo». ([Wikipedia. Deuteranopia, 2019](#))

Tritanopia

«La tritanopia es una disfunción visual relacionada con la percepción del color.

Consiste en la carencia de sensibilidad al color azul, denominada también dicromacia azul.

Se trata de una de las alteraciones de la visión cromática menos frecuentes». ([Wikipedia. Tritanopia, 2020](#))

Bibliografia

- [1] ¡Fácil y sencillo! Análisis de riesgos en 6 pasos. Nov. de 2020. URL: <https://www.incibe.es/protegetu-empresa/blog/analisis-riesgos-pasos-sencillo> (visitado 14-06-2021).
- [2] ¿Qué es el Esquema Nacional de Seguridad – ENS? Jun. de 2020. URL: <https://www.normas-iso.com/esquema-nacional-de-seguridad/> (visitado 05-07-2021).
- [3] ¿Qué es PILAR? URL: <https://pilar.ccn-cert.cni.es/index.php/pilar/que-es-pilar> (visitado 06-07-2021).
- [4] DP-3T. DP-3T/dp3t-sdk-android. URL: https://github.com/DP-3T/documents/blob/master/Security%5C%20analysis/PEPP-PT_%5C%20Data%5C%20Protection%5C%20Architechture%5C%20-%5C%20Security%5C%20and%5C%20privacy%5C%20analysis.pdf (visitado 16-04-2021).
- [5] DP-3T. DP-3T/dp3t-sdk-android. URL: <https://github.com/DP-3T/dp3t-sdk-android/blob/master/dp3t-sdk/src/main/java/org/dpppt/android/sdk/InfectionStatus.java> (visitado 16-04-2021).
- [6] DP-3T. DP-3T/dp3t-sdk-android. URL: <https://github.com/DP-3T/documents/blob/master/DP3T%5C%20-%5C%20Exposure%5C%20Score%5C%20Calculation.pdf> (visitado 26-04-2021).
- [7] DP-3T. DP-3T/dp3t-sdk-android. URL: <https://github.com/DP-3T/dp3t-sdk-android/tree/feaf563eb39d1d8c9416f718a81a574b25fa8384> (visitado 16-04-2021).
- [8] Nadeem Ahmed y col. “A Survey of COVID-19 Contact Tracing Apps”. En: *IEEE Access* 8 (2020), págs. 134577-134601. DOI: [10.1109/ACCESS.2020.3010226](https://doi.org/10.1109/ACCESS.2020.3010226). URL: <https://doi.org/10.1109/ACCESS.2020.3010226> (visitado 29-03-2021).
- [9] AMPARO. URL: <https://www.ccn-cert.cni.es/soluciones-seguridad/amparo.html> (visitado 06-07-2021).
- [10] Gennaro Avitabile y col. “Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System”. En: *IACR Cryptol. ePrint Arch.* 2020 (2020), pág. 493. URL: <https://eprint.iacr.org/2020/493> (visitado 23-04-2021).
- [11] Gennaro Avitabile y col. “Towards Defeating Mass Surveillance and SARS-CoV-2: The Pronto-C2 Fully Decentralized Automatic Contact Tracing System”. En: *IACR Cryptol. ePrint Arch.* 2020 (2020), pág. 493. (Visitado 29-04-2021).
- [12] Wasilij Beskorovajnov y col. “ConTra Corona: Contact Tracing against the Coronavirus by Bridging the Centralized - Decentralized Divide for Stronger Privacy”. En: *IACR Cryptol. ePrint Arch.* 2020 (2020), pág. 505. URL: <https://eprint.iacr.org/2020/505> (visitado 28-01-2021).
- [13] Bluetooth overview nbsp;; nbsp; Android Developers. URL: <https://developer.android.com/guide/topics/connectivity/bluetooth.html> (visitado 17-05-2021).
- [14] BlueTrace. [Online]. Feb. de 2021. URL: <https://en.wikipedia.org/wiki/BlueTrace> (visitado 16-04-2021).
- [15] BlueTrace Controversy. [Online]. Feb. de 2021. URL: <https://en.wikipedia.org/wiki/BlueTrace#Controversy> (visitado 16-04-2021).
- [16] European Data Protection Board. “Directrices 04/2020 sobre el uso de datos de localización y herramientas de rastreo de contactos en el contexto de la pandemia de COVID-19”. En: (dic. de 2020). URL: https://edpb.europa.eu/sites/edpb/files/files/file1/edpb_guidelines_20200420_contact_tracing_covid_with_annex_es.pdf (visitado 02-01-2021).
- [17] Cuándo, para qué y por qué utilizar MariaDB. [Online]. Ene. de 2018. URL: <https://www.arsys.es/blog/programacion/mariadb/> (visitado 07-04-2021).

- [18] Paul Dalg y col. “Das gefährliche Chaos um die Corona-App”. En: Tagesspiegel (abr. de 2020). URL: <https://www.tagesspiegel.de/wissen/welche-technologie-soll-es-sein-das-gefaehrliche-chaos-um-die-corona-app/25755338.html> (visitado 23-03-2021).
- [19] Decentralized Privacy-Preserving Proximity Tracing. [Online]. Dic. de 2020. URL: https://en.wikipedia.org/wiki/Decentralized_Privacy-Preserving_Proximity_Tracing (visitado 16-04-2021).
- [20] Definition of IP multicast. URL: <https://www.pcmag.com/encyclopedia/term/ip-multicast> (visitado 19-04-2021).
- [21] Paul-Olivier Dehaye y Joel Reardon. “SwissCovid: a critical analysis of risk assessment by Swiss authorities”. En: CoRR abs/2006.10719 (2020). arXiv: 2006.10719. URL: <https://arxiv.org/abs/2006.10719> (visitado 09-11-2020).
- [22] Desconocido. “Den Tracing-App-Entwicklern laufen die Partner weg”. En: Spiegel Netzwelt (abr. de 2020). URL: <https://www.spiegel.de/netzwelt/apps/pepp-pt-in-corona-krise-den-tracing-app-entwicklern-laufen-die-partner-weg-a-017f50eb-c1e2-4097-8182-53708ca6db59> (visitado 23-03-2021).
- [23] Difference with Apple/Google solution. [Online]. Nov. de 2020. URL: <https://github.com/DP-3T/documents/issues/128> (visitado 19-04-2021).
- [24] Procedimientos e Impulso de la Administración Electrónica Dirección General de Modernización Administrativa. “MAGERIT – versión 3.0. Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Libro II - Catálogo de Elementos”. En: (oct. de 2012). URL: <https://pilar.ccn-cert.cn.es/index.php/docman/documentos/2-magerit-v3-libro-ii-catalogo-de-elementos/file> (visitado 30-06-2021).
- [25] Documento BOE-A-2015-11881. URL: <https://www.boe.es/eli/es/rd/2015/10/23/951> (visitado 05-07-2021).
- [26] Documento consolidado BOE-A-2007-12352. URL: <https://www.boe.es/eli/es/l/2007/06/22/11/con> (visitado 05-07-2021).
- [27] Documento consolidado BOE-A-2010-1330. URL: <https://www.boe.es/eli/es/rd/2010/01/08/3/con> (visitado 05-07-2021).
- [28] DP3T - Decentralized Privacy-Preserving Proximity Tracing. [Online]. Dic. de 2020. URL: <https://github.com/DP-3T/documents> (visitado 16-04-2021).
- [29] Exposure Notification. [Online]. Nov. de 2020. URL: https://en.wikipedia.org/wiki/Exposure_Notification (visitado 15-04-2021).
- [30] Carlos Alonso González. Introducción a la Minería de Datos. https://aulas.inf.uva.es/pluginfile.php/41079/mod_resource/content/6/01IntroduccionMD.pdf. Accessed: 2021-03-03. 2020.
- [31] Lisa Hegemann. “Wissenschaftler warnen vor ”beispieloser Überwachung“”. En: Zeit (abr. de 2020). URL: <https://www.zeit.de/digital/datenschutz/2020-04/corona-app-initiative-pepp-pt-datenschutz-warnung-forscher> (visitado 23-03-2021).
- [32] Alex Hern. “Digital contact tracing will fail unless privacy is respected, experts warn”. En: The Guardian (abr. de 2020). URL: <https://www.theguardian.com/world/2020/apr/20/coronavirus-digital-contact-tracing-will-fail-unless-privacy-is-respected-experts-warn> (visitado 17-11-2020).
- [33] Mike Cotterell . Bob Hughes. ““Software Project Management”, Fifth Edition, Tata McGraw Hill, 2004.” En: 2015.
- [34] Simon Hurtz. “Der Anti-Corona-App droht ein Glaubenskrieg unter Forschern”. En: Süddeutsche Zeitung (abr. de 2020). URL: <https://www.sueddeutsche.de/digital/coronavirus-pepp-pt-dp-3t-smartphone-app-streit-1.4882612> (visitado 23-03-2021).

- [35] INES. URL: <https://www.ccn-cert.cni.es/soluciones-seguridad/ines.html> (visitado 06-07-2021).
- [36] Introducción general a Bluetooth. [Online]. Nov. de 2020. URL: <https://developer.android.com/guide/topics/connectivity/bluetooth?hl=es-419> (visitado 30-05-2021).
- [37] A. Tan; C. Sheng Hau; L. Yongquan; J. Tan J. Bay; J. Kek y T. Anh Quy. “BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders”. En: (dic. de 2020). URL: https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf (visitado 25-01-2021).
- [38] Bobbie Johnson. “Some prominent exposure apps are slowly rolling back freedoms”. En: (nov. de 2020). URL: <https://2020.internethealthreport.org/slideshow-internet-health/#16> (visitado 13-11-2020).
- [39] Sebastian Klöckner. Joint Statement on Contact Tracing. Abr. de 2020. URL: <https://cispa.de/en/news-and-events/news-archive/articles/2020/joint-statement-on-contact-tracing> (visitado 24-03-2021).
- [40] N. Lomas. “Norway pulls its coronavirus contacts tracing app after privacy watchdogs warning”. En: (dic. de 2020). URL: https://techcrunch.com/2020/06/15/norway-pulls-its-coronavirus-contacts-tracing-app-after-privacy-watchdogs-warning/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2x1LmNvbS8&guce_referrer_sig=AQAAANqd1ugQ71yUiD220jcVdJtVmSUPfuVEqULuAwgeXtLYz3ij5XYl0X8ZseqYHiE1Ct4Af9h2mm061Tar2JKKokRTfuejJAwNkdt8-1LMvTajRzId8N4ptyTw4X1Aa-07nN7KrQLiC3v6 (visitado 29-12-2020).
- [41] MAGERIT v.3 : Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. URL: https://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html (visitado 05-07-2021).
- [42] MariaDB. [Online]. Mar. de 2021. URL: <https://es.wikipedia.org/wiki/MariaDB> (visitado 07-04-2021).
- [43] Rafael Marín. Los gestores de bases de datos más usados en la actualidad. URL: <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/> (visitado 07-04-2021).
- [44] Misión y Objetivos. URL: <https://www.ccn-cert.cni.es/sobre-nosotros/mision-y-objetivos.html> (visitado 03-07-2021).
- [45] Mobile databases: SQLite and SQLite alternatives for Android and iOS. [Online]. Dic. de 2020. URL: <https://greenrobot.org/news/mobile-databases-sqlite-alternatives-and-nosql-for-android-and-ios/> (visitado 04-04-2021).
- [46] MySQL. [Online]. Ene. de 2021. URL: <https://es.wikipedia.org/wiki/MySQL> (visitado 04-04-2021).
- [47] MySQL, MariaDB y PostgreSQL: ¿Cuál elegimos? [Online]. Jun. de 2020. URL: <https://www.arsys.es/blog/mysql-mariadb-postgresql/> (visitado 04-04-2021).
- [48] Centro Criptológico Nacional. “Ciberamenazas y tendencias”. En: (sep. de 2020). URL: <https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/5377-ccn-cert-ia-13-20-ciberamenazas-y-tendencias-edicion-2020/file.html> (visitado 02-07-2021).
- [49] Centro Criptológico Nacional. “Dispositivos móviles. Informe de buenas prácticas”. En: (mayo de 2021). URL: <https://www.ccn-cert.cni.es/informes/informes-ccn-cert-publicos/1807-ccn-cert-bp-03-dispositivos-moviles-1/file.html> (visitado 02-07-2021).
- [50] Patrick Howell O'Neill. Norway halts coronavirus app over privacy concerns. Nov. de 2020. URL: <https://www.technologyreview.com/2020/06/15/1003562/norway-halts-coronavirus-app-over-privacy-concerns/> (visitado 15-12-2020).

- [51] Pan-European Privacy-Preserving Proximity Tracing. [Online]. Dic. de 2020. URL: https://en.wikipedia.org/wiki/Pan-European_Privacy-Preserving_Proximity_Tracing (visitado 20-04-2021).
- [52] Pilar. URL: <https://pilar.ccn-cert.cni.es/index.php/pilar/pilar> (visitado 06-07-2021).
- [53] Pilar Basic. URL: <https://pilar.ccn-cert.cni.es/index.php/pilar/pilar-basic> (visitado 06-07-2021).
- [54] Plan Director de Seguridad. Mar. de 2021. URL: <https://www.incibe.es/protege-tu-empresa/que-te-interesa/plan-director-seguridad> (visitado 14-06-2021).
- [55] Protocolo seguro. [Online]. Mar. de 2021. URL: https://es.wikipedia.org/wiki/Protocolo_seguro (visitado 20-03-2021).
- [56] Ransomware. [Online]. Jun. de 2021. URL: <https://es.wikipedia.org/wiki/Ransomware> (visitado 04-07-2021).
- [57] REYES. URL: <https://www.ccn-cert.cni.es/soluciones-seguridad/amparo.html> (visitado 06-07-2021).
- [58] Security Considerations For Bluetooth Smart Devices. URL: <https://www.design-reuse.com/articles/39779/security-considerations-for-bluetooth-smart-devices.html> (visitado 10-05-2021).
- [59] José Luis Sevillano y col. “Soft real-time communications over Bluetooth under interferences from ISM devices”. En: *Int. J. Commun. Syst.* 19.10 (2006), págs. 1103-1116. DOI: [10.1002/dac.796](https://doi.org/10.1002/dac.796). URL: <https://doi.org/10.1002/dac.796> (visitado 22-05-2021).
- [60] Soluciones de Ciberseguridad. URL: <https://www.ccn-cert.cni.es/soluciones-seguridad.html> (visitado 02-07-2021).
- [61] Mark Surman. Privacy Norms and the Pandemic. Abr. de 2020. URL: <https://blog.mozilla.org/blog/2020/04/22/privacy-norms-and-the-pandemic/> (visitado 27-04-2021).
- [62] Ni Trieu y col. “Epione: Lightweight Contact Tracing with Strong Privacy”. En: *IEEE Data Eng. Bull.* 43.2 (2020), págs. 95-107. URL: <http://sites.computer.org/debull/A20june/p95.pdf> (visitado 21-04-2021).
- [63] Ni Trieu y col. “Epione: Lightweight Contact Tracing with Strong Privacy”. En: *ArXiv* abs/2004.13293 (2020). (Visitado 25-04-2021).
- [64] Ni Trieu y col. Epione: Lightweight Contact Tracing with Strong Privacy. URL: <https://sunblaze-ucb.github.io/privacy/projects/epione.html> (visitado 25-04-2021).
- [65] Serge Vaudenay. “Analysis of DP3T”. En: *IACR Cryptol. ePrint Arch.* 2020 (2020), pág. 399. URL: <https://eprint.iacr.org/2020/399> (visitado 27-04-2021).
- [66] Serge Vaudenay. “Centralized or Decentralized? The Contact Tracing Dilemma”. En: *IACR Cryptol. ePrint Arch.* 2020 (2020), pág. 531. URL: <https://eprint.iacr.org/2020/531> (visitado 27-04-2021).
- [67] Matthew Wickline y the Human-Computer Interaction Resource Network. “Color Blind Simulation”. En: (- de 2001). URL: <https://www.color-blindness.com/coblis-color-blindness-simulator/> (visitado 06-07-2021).
- [68] PILAR. URL: <https://pilar.ccn-cert.cni.es/index.php/pilar/pilar-micro> (visitado 06-07-2021).