Universidad
de La Laguna

Escuela Superior de
Ingeniería y Tecnología
Sección de Ingeniería Informática

Trabajo de Fin de Grado

Aplicación móvil segura para combatir la violencia de género

Secure Mobile Application to Combat Gender Violence

José Ángel Concepción Sánchez

La Laguna, 1 de junio de 2016

D.ª **Pino Teresa Caballero Gil**, con N.I.F. 45.534.310-Z Catedrática de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutora

D.ª **Jezabel Míriam Molina Gil**, con N.I.F. 78.507.682-B Profesora Contratada Laboral Interina de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutora

C E R T I F I C A N

Que la presente memoria titulada:

"Aplicación móvil segura para combatir la violencia de género."

ha sido realizada bajo su dirección por D. **José Ángel Concepción Sánchez**, con N.I.F. 42.234.897-C.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 1 de junio de 2016.

Agradecimientos

Este presente trabajo fue realizado bajo la supervisión de D.ª Pino Teresa Caballero Gil y D. ª Jezabel Míriam Molina Gil, a quienes me gustaría expresar mi más profundo agradecimiento por la dedicación, ayuda e interés mostrado en este tiempo.

Al grupo CryptULL al completo por su interés, ayudando siempre y aportando su experiencia y conocimientos para que todo siempre saliese y de la mejor forma posible.

A mis padres, por todo. Sin su paciencia y apoyo en todo lo que me he propuesto no hubiera logrado mis metas y sueños. Por ser mis ejemplos a seguir y enseñarme a continuar siempre para adelante, sin importar las circunstancias.

A mis hermanos, abuelos, tíos, primos y familiares por siempre estar apoyándome y ayudándome en lo que necesitara.

A todos mis amigos, tanto los que he conocido en este periodo como los que ya estaban de antes, por ser parte de mi vida en todo tipo de momentos (buenos y malos). Por siempre estar ahí.

Y finalmente, a los profesores que a lo largo de mi vida han compartido sus conocimientos conmigo, animándome a seguir.

A todos, muchas gracias.

Licencia

Resumen

El objetivo de este trabajo ha sido crear una aplicación móvil que permita combatir la violencia de género. Esta aplicación será utilizada como dispositivo electrónico de seguimiento de maltratadores y proveerá de información a la víctima en caso de que el agresor se acerque demasiado y supere un umbral de distancia, avisándola de su presencia para que esté alerta y, por tanto, pueda evitar situaciones no deseadas.

Además, en caso de que el agresor supere dicho umbral, se activará una grabación de vídeo automáticamente con envío streaming junto con los datos de la víctima a una aplicación web encargada del control.

El desarrollo de la aplicación se ha realizado usando tecnologías como son Android, BLE (Bluetooth Low Energy), Wi-Fi y LTE (Long Term Evolution). Además, dado el alto nivel de sensibilidad de la información manejada, la aplicación cuenta con un alto nivel de seguridad mediante la implementación de diversos sistemas y protocolos criptográficos.

Palabras clave: Seguridad, Aplicación Móvil, Violencia de Género, Protección, Vídeo Streaming, Android, Bluetooth Low Energy, Long Term Evolution, Wi-Fi.

Abstract

The purpose of this work has been to create a mobile application that allows combat gender violence. This application will be used as an offender electronic tracking device and it will provide information to the victim if the offender gets too close and exceeds a threshold, warning her presence to be alert and therefore can avoid unwanted situations.

Besides, if the offender exceeds this threshold, a video recording streaming will be activated automatically, along with the data of the victim, to a control point (web application).

The development of the application has been made using technologies such as Android, BLE (Bluetooth Low Energy), Wi-Fi and LTE (Long Term Evolution). In addition, given the high level of sensitivity of information handled, the application has a high level of security by implementing various systems and cryptographic protocols.

Keywords: Security, Mobile Application, Gender Violence, Protection, Video Streaming, Android, Bluetooth Low Energy, Long Term Evolution, Wi-Fi.

Índice General

Capítulo 1. Introducción al trabajo **1**

1.1 Motivación 1

1.2 Objetivos 2

1.3 Fases del desarrollo 2

1.4 Estructura de la memoria 3

Capítulo 2. Introducción a la herramienta **4**

2.1 Aplicación móvil 4

2.2 Aplicación web 6

2.3 Conceptualización 7

Capítulo 3. Tecnologías **9**

3.1 Lenguajes de programación utilizados 9

3.1.1 Android 9

3.1.2 MVC y NodeJS 10

3.1.3 AngularJS 10

3.1.4 HTML5 11

3.1.5 CSS y Materialize 11

3.2 Tecnologías inalámbricas 12

3.2.1 Bluetooth Low Energy (BLE) 12

3.2.2 Long Term Evolution (LTE) 13

3.2.3 Wi-Fi 13

3.3 Seguridad 13

3.3.1 Diffie Hellman de Curva Elíptica 14

3.3.2 256-bit AES modo CBC 15

3.3.3 Firma Digital con Curva Elíptica 16

Capítulo 4. Sistema de localización **17**
4.1 Datos utilizados ... 17
4.2 Funcionamiento ... 18
4.3 Dificultades ... 19
4.4 Mejoras ... 20

Capítulo 5. Sistema de streaming **21**
5.1 Componentes ... 21
5.2 Librería libstreaming ... 21
5.3 RTSP y RTMP ... 23
5.4 Funcionamiento ... 24
5.5 Dificultades ... 26

Capítulo 6. Herramienta **27**
6.1 Definición ... 27
6.2 Configuración de la aplicación móvil ... 28
6.3 Uso de la aplicación móvil ... 32
6.4 Uso de la aplicación web ... 34

Capítulo 7. Implementación **37**
7.1 Planificación ... 37
7.2 Esquema de comunicaciones ... 38
7.3 Base de datos para la aplicación móvil ... 39
7.4 Base de datos para la aplicación web ... 40
7.5 Clase Contacto ... 41
7.6 Servicios en la aplicación Android ... 42
7.6.1 Búsquedas en segundo plano ... 42
7.6.2 Grabación de vídeo ... 42
7.7 Librería Volley ... 42
7.8 Envío de datos desde móvil a web ... 43
7.9 Desafíos ... 44

Capítulo 8. Presupuesto **45**
8.1 Personal 45
8.2 Componentes 46
8.3 Coste total 46

Capítulo 9. Conclusiones y trabajos futuros **47**

Capítulo 10. Conclusions and future works **48**

Bibliografía **49**

Apéndice A. Código destacable **51**
A.1. Inicio de grabación de vídeo 51
A.2. Receiver cuando se detecta al agresor 52
A.3. Tablas de la BDD de la aplicación móvil 54
A.4. Servicio para la búsqueda usando Bluetooth 55
A.5. Controlador web *putonline* 56
A.6. Modelos de la aplicación web 57

Apéndice B. Conference Paper **58**

Índice de figuras

Figura 2.1.1 Notificación de detección del agresor 4
Figura 2.1.2 Notificación de acercamiento del agresor 5
Figura 2.1.3 Notificación de Bluetooth desactivado 6
Figura 2.2.1 Aplicación web recibiendo vídeo en streaming 7
Figura 2.3.1 Estadística de mortalidad en los últimos años 7
Figura 2.3.2 Dispositivo de la víctima 8
Figura 2.3.3 Dispositivos del agresor 8
Figura 3.1.2.1 Modelo Vista Controlador 10
Figura 3.3.1.1 Ejemplo del protocolo Diffie Hellman con Curvas Elípticas 14
Figura 3.3.2.1 Cifrado en modo CBC 15
Figura 3.3.3.1 Firma digital con clave pública 16
Figura 4.1.1. Función para calcular distancia aproximada 17
Figura 4.2.1 Agresor no detectado 18
Figura 4.2.2 Agresor detectado fuera del rango de peligro 18
Figura 4.2.3 Agresor detectado dentro del rango de peligro 19
Figura 5.2.1. Permisos para la librería libstreaming 22
Figura 5.2.2. Sesión de ejemplo para el streaming 22
Figura 5.2.3 Cliente RTSP 22
Figura 5.4.1 Esquema de funcionamiento del vídeo en streaming 24
Figura 5.4.2 Notificación de la grabación en curso 25
Figura 5.4.3 Información personal del usuario 26
Figura 6.2.1 Pantalla de acceso a la aplicación 29
Figura 6.2.2 Cambio de contraseña. 29
Figura 6.2.3 Actualizar datos de usuario 30
Figura 6.2.4 Contactos en la aplicación 30
Figura 6.2.5 Menú para añadir nuevo contacto 31

Figura 6.2.6 Añadir nuevo contacto en la aplicación móvil 31
Figura 6.2.7 Editar contacto 31
Figura 6.2.8 Contactos habilitados y deshabilitados 32
Figura 6.2.9 Menú emergente al pulsar sobre un contacto 32
Figura 6.3.1 Confirmación para cerrar aplicación 33
Figura 6.3.2 Aplicación con el Bluetooth desactivado 33
Figura 6.3.3 Notificación de Bluetooth desactivado 33
Figura 6.3.4 Botón de pánico 34
Figura 6.4.1 Página de inicio de la web 35
Figura 6.4.2 Pestaña live en la aplicación web 35
Figura 6.4.3 Historial almacenado en la web 36
Figura 6.4.4 Dispositivos almacenados en la web 36
Figura 7.2.1 Esquema de comunicaciones 38
Figura 7.3.1 Función *insertarCONTACTO()* 40
Figura 7.5.1 Estructura del objeto Contact 41
Figura 7.8.1 Función para enviar datos del dispositivo a la web 43

Índice de tablas

Tabla 6.1.1 Clases usadas en el proyecto..... 27
Tabla 7.1.1 Hitos en el desarrollo del proyecto..... 37
Tabla 7.3.1.1 Base de datos de la aplicación móvil..... 39
Tabla 7.3.2.1 Base de datos de la aplicación web.. 41
Tabla 8.1.1 Tabla de horas por tarea realizada. 45
Tabla 8.2.1 Tabla de componentes usados en el proyecto. 46
Tabla 8.3.1 Tabla del presupuesto total.. 46

Capítulo 1.
Introducción al trabajo

En este capítulo se incluye una introducción acerca del planteamiento del proyecto, así como de la estructura de este documento.

1.1 Motivación

En la actualidad, uno de los temas de mayor relevancia, desgraciadamente, es el de la violencia de género. El presente proyecto tiene como objeto principal desarrollar una aplicación para su uso como dispositivo electrónico de seguimiento de maltratadores, proveyendo a la víctima de información en caso de que el agresor se acerque a un rango de distancia inferior a un umbral y avisando mediante un envío de grabación de vídeo en streaming a la policía y con SMS a los contactos que tenga añadidos en la aplicación. Junto con el vídeo también se enviará la información personal que tenga configurada la víctima en la aplicación móvil.

Las motivaciones para el desarrollo de este Trabajo de Fin de Grado han sido varias:

Por un lado, el aprendizaje de un lenguaje de programación como es Android y su IDE Android Studio, ya que cada vez se está expandiendo y usando más este sistema operativo.

Por otro lado, la idea de trabajar en un proyecto ya establecido por una gran empresa junto con el grupo CryptULL, los cuales me han aportado bastantes conocimientos y experiencias que me han servido y me servirán en un futuro.

Y finalmente, y la más importante de todas, la motivación de aportar mi granito de arena para intentar acabar con un problema tan grave como está siendo el de la violencia de género en la sociedad.

1.2 Objetivos

El objetivo principal de este proyecto es el de ayudar no solo a evitar situaciones no deseadas entre la víctima y el agresor, sino también transmitirle seguridad a la víctima en la vida cotidiana, de forma que pueda salir a la calle sin estar preocupada por si aparecerá el agresor, ya que el sistema se encargará de comprobar que esto no se produce y, en caso de que sí, avisarla con tiempo.

Este objetivo general se puede concretar a su vez en una serie de objetivos específicos:

I. El más importante: reducir el número de casos de maltratos debido a la violencia de género.

II. Evitar situaciones no deseadas entre ambos.

III. Proporcionar seguridad a la víctima para que pueda realizar una vida lo más normal posible.

IV. Conseguir que la aplicación sea lo más asequible posible.

V. Relacionado con el anterior punto, implantar a un mayor número de casos y no solo a los de riesgo alto o extremo un sistema con el que se sientan protegidas las víctimas.

1.3 Fases del desarrollo

El desarrollo del proyecto se ha realizado en diversas fases:

En primer lugar, como nunca antes habíamos trabajado con el lenguaje de programación Android, empezamos por familiarizarnos con él, realizando diversos cursos online y presenciales, para de esta manera aprender cómo funcionaban todos los elementos de los que consta el lenguaje y el sistema operativo.

Seguidamente, una vez ya familiarizados con el entorno de trabajo, se comenzó con el desarrollo de la aplicación móvil en sí hasta tener un primer prototipo finalizado. Además, dentro de esta fase también se desarrolló otro prototipo, esta vez de aplicación web y basado en NodeJS el cual tendrá la función de centro de control, donde se podrán visualizar los vídeos que se envíen en streaming desde las aplicaciones móviles.

A esto le sigue la tercera fase, que radica en la mejora de estos prototipos, eliminando todos los bugs y problemas que aún quedasen y optimizando el funcionamiento de las aplicaciones.

Y para acabar, la última fase, basada en el desarrollo de la memoria del Trabajo de Fin de Grado.

1.4 Estructura de la memoria

La estructura de la memoria comienza con los dos primeros capítulos, donde se tratan temas como son la motivación, los objetivos, las fases de desarrollo del proyecto junto con una breve descripción de la estructura de la que consta la memoria y la definición y la descripción de la herramienta, así como su conceptualización en la actualidad.

Seguidamente, pasaremos a tratar por separado y de una forma más completa cada uno de los sistemas más importantes de los que consta el proyecto, como son las tecnologías inalámbricas, la seguridad implantada, la localización del agresor y el de envío de vídeo usando streaming.

Después, nos centraremos en la definición y explicación del funcionamiento de las herramientas, tanto la aplicación móvil como la web, así como en los pasos y plazos llevados a cabo durante el desarrollo y los problemas encontrados.

Posteriormente, acabaremos mencionando el presupuesto del desarrollo del proyecto, así como las conclusiones finales, tanto en español como en inglés.

Y finalmente, nos encontraremos como apéndices los códigos destacables y el paper presentado de este proyecto para 10^{th} International Conference on Ubiquitous Computing and Ambient Intelligence UCAmI 2016.

Capítulo 2. Introducción a la herramienta

En este capítulo se definen las herramientas desarrolladas para el Trabajo de Fin de Grado, así como su conceptualización en el mundo actual.

2.1 Aplicación móvil

Se ha desarrollado una aplicación móvil para el sistema operativo Android (con posibilidad de portabilidad a iOS y Windows Phone en el futuro) que, usando BLE (Bluetooth Low Energy), se queda en segundo plano comprobando que el dispositivo que lleva el agresor (una pulsera BLE) no es detectado.

De esa manera, si la aplicación detecta al agresor, existen dos niveles de aviso, usando para ellos las notificaciones de Android además de vibración y sonido. Estos niveles son:

I. Nivel de precaución: El agresor ha sido detectado en las proximidades, pero no supera la intensidad de señal necesaria como para concluir que está demasiado cerca de la víctima. De esta manera, la víctima podrá estar pendiente y evitar situaciones no deseadas.

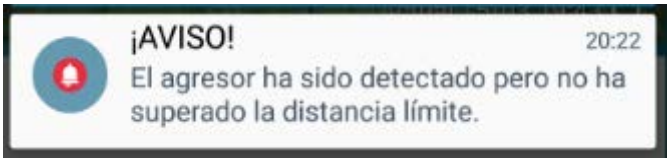


Figura 2.1.1 Notificación de detección del agresor

II. Nivel de peligro: El agresor ha sido detectado, y como además ha superado un umbral de intensidad de señal, se concluye que está demasiado cerca de la víctima.

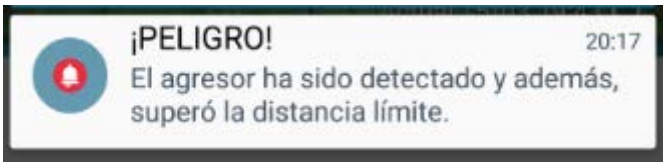


Figura 2.1.2 Notificación de acercamiento del agresor

Además de notificar a la víctima en este segundo tipo de aviso, la aplicación inicia una grabación de vídeo que es enviada en tiempo real usando para ello la tecnología Wi-Fi o LTE, a una estación encargada de recibirlo, que en nuestro caso será la aplicación web explicada más adelante.

Además de la funcionalidad básica de la aplicación comentada anteriormente, ésta posee también otros elementos de gran interés:

- Botón de pánico: En la pantalla principal de la aplicación existe un botón de pánico lo suficientemente grande para que, si la víctima se ve apurada, con solo pulsarlo de manera instintiva se realice una llamada automáticamente al servicio de emergencias. Esta funcionalidad tiene gran importancia ya que, en momentos de gran tensión, muchas veces no damos avío a marcar un número o seleccionar un contacto en nuestro teléfono.
- Lista de contactos: La víctima puede configurar en la aplicación una lista con los números de los contactos que desea que sean notificados mediante SMS cuando el agresor superó el segundo nivel de aviso de manera que, por ejemplo, si la víctima está en su casa, un vecino pueda acudir en su ayuda mientras llega la policía.
- Grabación en streaming por voluntad propia: La víctima también podrá enviar un vídeo al centro de control pulsando un botón habilitado para ello en la aplicación móvil.
- Aviso del Bluetooth desactivado: La aplicación avisará a la víctima de que tiene en Bluetooth desactivado en el dispositivo y que por tanto corre el peligro de que el agresor esté cerca y no pueda ser notificada. Esta notificación no se podrá quitar de la bandeja de notificaciones de Android hasta que lo active por su propia seguridad.

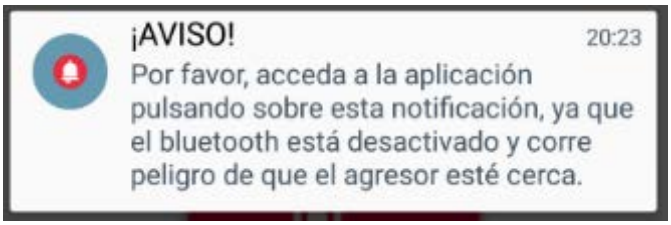


Figura 2.1.3 Notificación de Bluetooth desactivado

- Privacidad de la aplicación: El icono de la aplicación será trasparente para que la víctima tenga privacidad y ni el propio agresor ni otra persona pueda ver que está instalada. Además, para poder abrirla, primero saldrá una pantalla simulando la configuración del teléfono que solo permitirá acceder a la aplicación si introduce correctamente una contraseña que la víctima puede configurar en la propia aplicación.

A lo largo del documento se irán viendo con más detalle las funcionalidades de la aplicación junto con capturas de su funcionamiento.

2.2 Aplicación web

La aplicación web desarrollada es la encargada de llevar el control de todos los eventos que se envíen desde las aplicaciones móviles. Las funcionalidades que presenta son las siguientes:

- Visualización de vídeos: Desde la web se podrán visualizar todos los vídeos que se estén enviando desde las aplicaciones móviles en tiempo real. Esto permitirá que se pueda ver en directo la situación en la que se encuentran las víctimas y actuar lo antes posible.
- Registro de conexiones: Cada vez que se realiza una conexión con el centro de control, se registra en un historial con la fecha, hora, nombre del usuario y número de contacto.
- Registro de dispositivos: Al igual que el registro de conexiones, en la aplicación web también quedan guardados todos los dispositivos que han hecho uso del sistema.

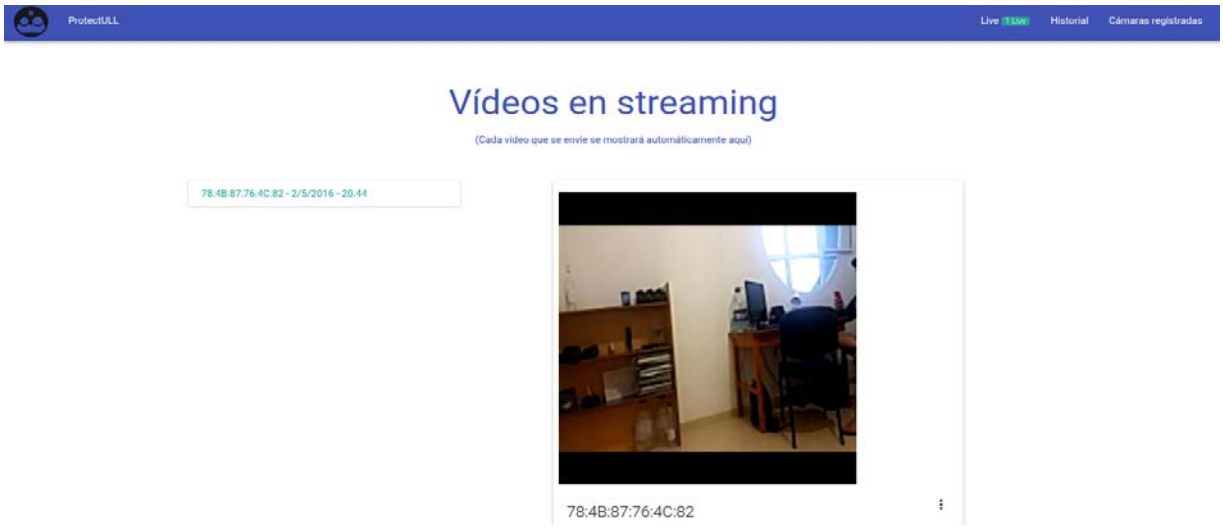


Figura 2.2.1 Aplicación web recibiendo vídeo en streaming

2.3 Conceptualización

En la actualidad, uno de los temas de mayor relevancia en el mundo es, desgraciadamente, la violencia de género. Por ejemplo, en España, en los últimos 10 años ha habido una media de 62 víctimas mortales, siendo una gran mayoría mujeres. Además, a esto hay que añadirle todos los casos que se dan y que no llegan a tal extremo pero que son igualmente de gran importancia.

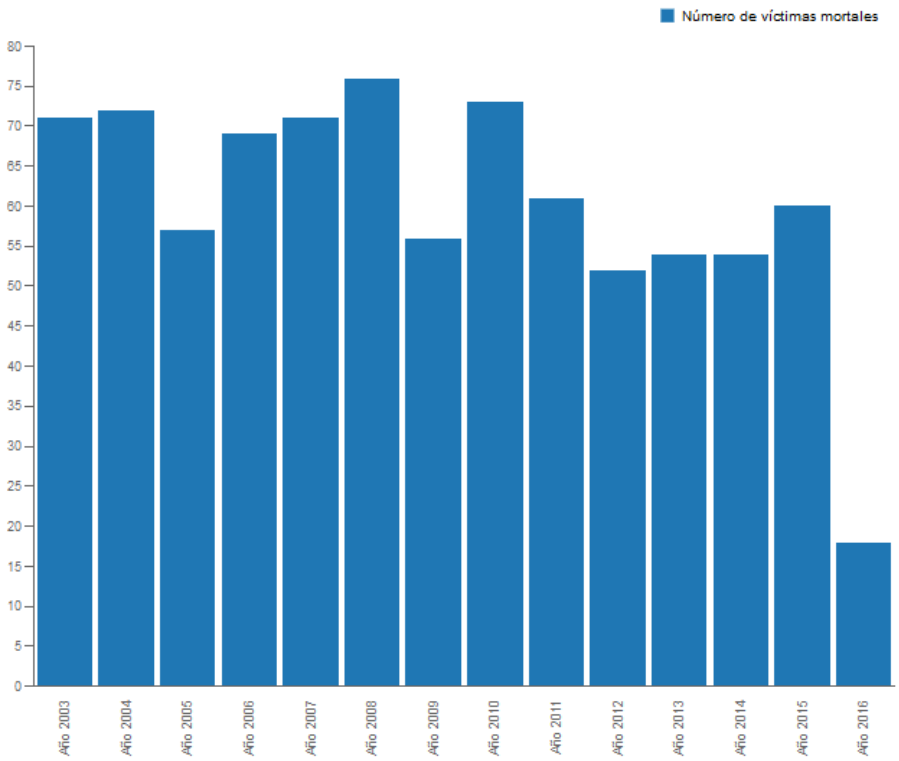


Figura 2.3.1 Estadística de mortalidad en los últimos años

En 2009 se introdujo un sistema de pulseras que portan los maltratadores para notificar a un centro de control si el agresor se salta los límites. Dicho sistema sólo es implantado para casos de riesgo alto o extremo ya que es muy caro, lo que provoca que la gran mayoría de víctimas no estén protegidas. Además, los dispositivos que llevan tanto el agresor como la víctima son un poco aparatosos y pueden no pasar desapercibidos de cara al público.

Figura 2.3.2 Dispositivo de la víctima

Figura 2.3.3 Dispositivos del agresor

Todo esto nos indica que existe un grave problema en nuestra sociedad para el cual las soluciones actuales no son del todo eficaces.

De esta manera, este proyecto pretende ayudar no solo evitar situaciones no deseadas entre la víctima y el agresor, sino también acabar con este problema, pues muchas veces si la víctima tuviese información en tiempo real se evitarían muchas tragedias.

Asimismo, en la práctica es un sistema mucho más viable, ya que el desembolso sería mucho menor porque actualmente la mayoría de las personas ya poseen un dispositivo móvil, que es lo único que se necesita junto con la pulsera BLE del agresor, pasando mucho más desapercibido que los sistemas actuales.

Capítulo 3. Tecnologías

En este capítulo hablaremos de las tecnologías usadas en el proyecto, así como de la seguridad implantada para el cifrado de la información sensible que se envía desde la aplicación móvil.

3.1 Lenguajes de programación utilizados

Tanto para la aplicación móvil como para la web hemos hecho uso de diferentes lenguajes de programación. Seguidamente, hablaremos de cada uno de ellos y su función dentro del proyecto.

3.1.1 Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tablets, o tabléfonos; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, la compró. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Actualmente, los dispositivos Android se venden más que los sistemas iOS y Windows Phone juntos.

Nuestra aplicación está desarrollada para este sistema operativo, aunque cabe la posibilidad de que en un futuro se pueda adaptar a otros (iOS y Windows Phone). Además, para la implementación hemos usado como editor Android Studio, entorno desarrollado para la plataforma Android. Está basado en el software IntelliJ IDEA de JetBrains, es publicado de forma gratuita a través de la Licencia Apache 2.0, y está disponible para las plataformas Microsoft WIndows, Mac OS X y GNU/Linux.

3.1.2 MVC y NodeJS

El patrón de arquitectura MVC (Modelo Vista Controlador) es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del workflow de la aplicación). De esta forma, dividimos el sistema en tres capas donde tenemos la encapsulación de los datos, la interfaz o vista por otro lado y por último la lógica interna o controlador.

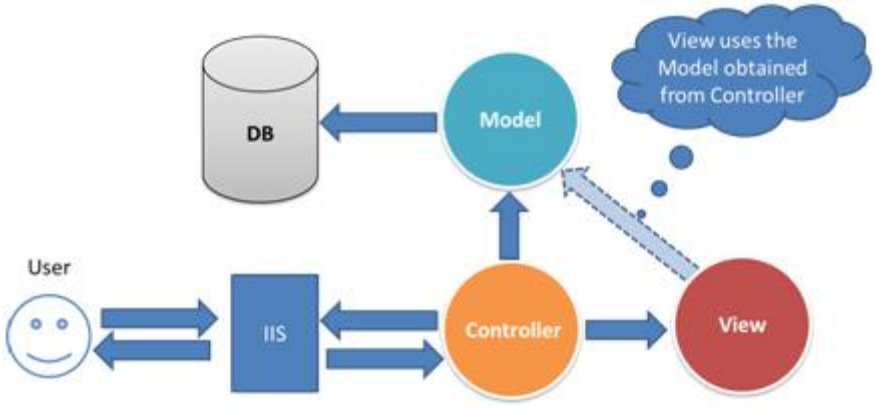


Figura 3.1.2.1 Modelo Vista Controlador

Node.js es un entorno en tiempo de ejecución multiplataforma basado en este modelo, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables como, por ejemplo, servidores web.

La aplicación web encargada del control y visualización de los vídeos enviados en streaming desde las aplicaciones móviles está basada en esta estructura.

3.1.3 AngularJS

AngularJS, o simplemente Angular, es un framework de Javascript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos

personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de Javascript. Además, los valores de las variables de Javascript se pueden configurar manualmente o recuperarlos de los recursos JSON estáticos o dinámicos.

Angular, en nuestra aplicación web, se combina con el entorno en tiempo de ejecución Node.js, el framework para servidor Express.js y la base de datos MongoDB para formar el conjunto MEAN.

3.1.4 HTML5

HTML5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una clásica, HTML, conocida como HTML5, y una variante XHTML, conocida como sintaxis XHTML5 que deberá servirse con sintaxis XML. Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.

En nuestro proyecto, este lenguaje es usado para definir las vistas dentro de NodeJS para la aplicación web.

3.1.5 CSS y Materialize

Hoja de estilo en cascada o CSS es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML. El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario y navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Por su parte, usamos también Materialize, un framework web front-end moderno y responsivo basado en Material Design.

En nuestra aplicación ambas tecnologías son usadas para el estilo de las vistas.

3.2 Tecnologías inalámbricas

Las tecnologías inalámbricas de las que se hace uso en este proyecto son Bluetooth Low Energy (BLE), Long Term Evolution (LTE) y Wi-Fi. A continuación, describiremos un poco más cada una de ellas y explicaremos la función que tienen dentro de la aplicación móvil.

3.2.1 Bluetooth Low Energy (BLE)

Bluetooth Low Energy es una nueva tecnología digital de radio interoperable (inalámbrica) para pequeños dispositivos desarrollada por Bluetooth. Es la primera tecnología abierta de comunicación inalámbrica, que ofrece comunicación entre dispositivos móviles u ordenadores y otros dispositivos más pequeños, la cual está diseñada para que funcione con poca energía.

Asimismo, permite la comunicación entre dispositivos de pila de botón y dispositivos Bluetooth, con una tasa de transferencia de 1Mbps en la capa física y con una distancia de alcance de 100 metros como máximo en un entorno favorable libre de obstáculos para el Bluetooth de Clase A. Está basado en un microchip de bajo costo con opciones más amplias para su empleo en la industria y además tiene el mismo tamaño que cualquier otro dispositivo Bluetooth.

Finalmente, respecto a la seguridad, emplea el sistema de cifrado AES y esquemas de seguridad configurables.

En nuestro proyecto, es usado para detectar la proximidad del agresor. Para ello, la aplicación realiza búsquedas cada un tiempo determinado (actualmente cada 15 segundos) y comprueba que no detecta al agresor. En caso de que sí lo detecte, se notificaría con algún tipo de aviso y las acciones correspondientes de las cuales hemos hablado anteriormente.

Actualmente la aplicación está desarrollada con las librerías del Bluetooth común, debido a la falta de un dispositivo BLE que pueda actuar como baliza para la realización de las pruebas. De todos modos, el funcionamiento sería exactamente el mismo, lo único cambiando las librerías por las LE.

3.2.2 Long Term Evolution (LTE)

Long Term Evolution o LTE en telecomunicaciones es un estándar para comunicaciones inalámbricas de transmisión de datos de alta velocidad para teléfonos móviles y terminales de datos. También es conocido como 4G.

La idea es que la aplicación envíe el vídeo en streaming haciendo uso de esta tecnología independientemente del lugar donde se encuentre la víctima, pero como actualmente no se posee de una infraestructura de datos 4G ilimitados para realizar todas las pruebas oportunas de funcionamiento, hemos decidido realizar el prototipo haciendo uso de la tecnología Wi-Fi.

3.2.3 Wi-Fi

Wi-Fi es uno de los mecanismos de conexión de dispositivos electrónicos de forma inalámbrica más utilizados hoy en día. Los dispositivos habilitados con esta tecnología, pueden conectarse a Internet a través de un punto de acceso a la red inalámbrica.

En nuestro caso, hacemos uso de esta tecnología para el envío del vídeo en streaming como alternativa al LTE, comprobando si el usuario está conectado a una red de este tipo para así evitar un mayor consumo de datos.

3.3 Seguridad

Dada la alta sensibilidad de los datos con los que trata la aplicación, como puede ser el envío de vídeo mediante streaming o los datos de contacto de la víctima, es necesario garantizar la confidencialidad de estos. Para ello hacemos uso de diferentes esquemas criptográficos que nos ayudan a proteger la autenticidad, confidencialidad e integridad de los datos.

Para ello, la aplicación ha sido enriquecida con una implementación de OpenSSL incluyendo Curvas Elípticas Diffie Hellman para el acuerdo de clave secreta, cifrado 256-bit AES en modo CBC para los datos que se envían y el algoritmo de Firma Digital de Curva Elíptica para la verificación.

Seguidamente veremos una descripción más detallada de cada uno de estos esquemas criptográficos.

3.3.1 Diffie Hellman de Curva Elíptica

Diffie-Hellman es un protocolo criptográfico de establecimiento de claves entre partes que no han tenido contacto previo, utilizando un canal inseguro, y de manera anónima (no autentificada).

Se emplea generalmente como medio para acordar claves simétricas que serán empleadas para el cifrado de una sesión (establecer clave de sesión). Siendo no autenticado, sin embargo, provee las bases para varios protocolos autenticados.

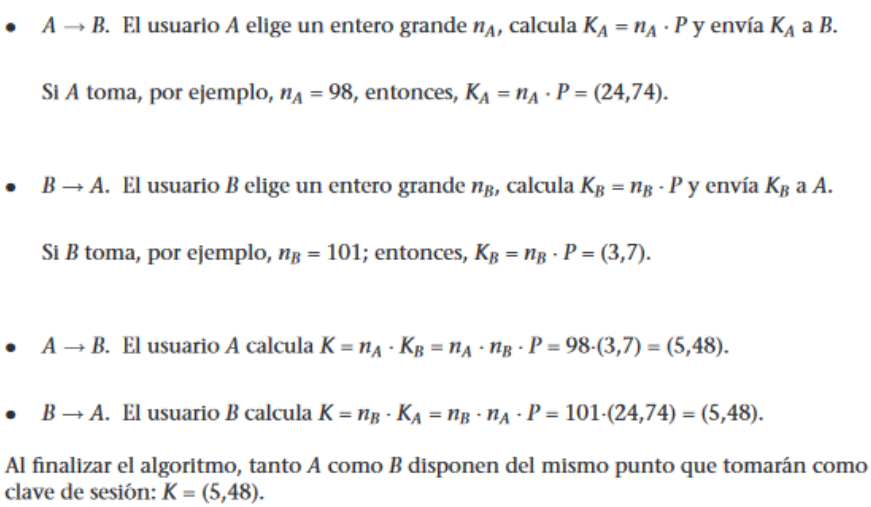
Su seguridad radica en la extrema dificultad (conjeturada, no demostrada) de calcular logaritmos discretos en un cuerpo finito.

La utilización de curvas elípticas en este protocolo nos permite usar claves más cortas al tiempo que proporcionan una mayor velocidad y un nivel de seguridad equivalente. El funcionamiento de este protocolo criptográfico con curvas elípticas es el siguiente:

En primer lugar, los usuarios A y B acuerdan una curva elíptica E, un conjunto finito sobre n y un punto P perteneciente a la curva y al conjunto finito.

Seguidamente, se calculan las claves públicas, donde cada usuario escoge un entero secreto (n_A y n_B), calculan los correspondientes n_AP y n_BP y lo envían al otro como K_A *y* K_B respectivamente.

Y, finalmente, pasan a calcular las claves privadas tomando cada usuario el punto enviado por el otro y calculando los correspondientes K, siendo el resultado final de ambas claves el mismo.



- $A \rightarrow B$. El usuario A elige un entero grande n_A, calcula $K_A = n_A \cdot P$ y envía K_A a B.

  Si A toma, por ejemplo, $n_A = 98$, entonces, $K_A = n_A \cdot P = (24,74)$.

- $B \rightarrow A$. El usuario B elige un entero grande n_B, calcula $K_B = n_B \cdot P$ y envía K_B a A.

  Si B toma, por ejemplo, $n_B = 101$; entonces, $K_B = n_B \cdot P = (3,7)$.

- $A \rightarrow B$. El usuario A calcula $K = n_A \cdot K_B = n_A \cdot n_B \cdot P = 98 \cdot (3,7) = (5,48)$.

- $B \rightarrow A$. El usuario B calcula $K = n_B \cdot K_A = n_B \cdot n_A \cdot P = 101 \cdot (24,74) = (5,48)$.

Al finalizar el algoritmo, tanto A como B disponen del mismo punto que tomarán como clave de sesión: $K = (5,48)$.

Figura 3.3.1.1 Ejemplo del protocolo Diffie Hellman con Curvas Elípticas

3.3.2 256-bit AES modo CBC

Advanced Encryption Standard (AES), también conocido como Rijndael, es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos, transformándose en un estándar efectivo el 26 de mayo de 2002 y siendo desde 2006 uno de los algoritmos más populares usados en criptografía simétrica.

Las ventajas en el diseño de este algoritmo son la seguridad que aporta, la eficacia computacional, es simple, paralelizable y no de tipo Feistel.

Algunos datos interesantes de este esquema criptográfico son:

- Longitud de bloque de 128 bits y de clave de 128, 192 y 256 bits.
- Número de iteraciones flexible (10, 12, 14).
- Operaciones a nivel de byte, con palabras de 4 bytes y cálculos en $GF(2^8)$.

En cuanto al cifrado, en cada una de las iteraciones se realizan cuatro pasos en el siguiente orden, mientras que para el descifrado el orden es a la inversa.

- Sustitución de cada byte por su recíproco (*ByteSub*).
- Desplazamiento de bytes (*ShiftRow*).
- Producto de cada columna por una matriz (*MixColumn*).
- XOR de la subclave y la información del estado intermedio actual (*AddRoundKey*).

Además, haciendo uso del modo CBC (Cipher Block Chaining), a cada bloque de texto plano se le aplica la operación XOR con el bloque cifrado anterior antes de ser cifrado. De esta forma, cada bloque de texto cifrado depende de todo el texto en claro procesado hasta este punto. Para hacer cada mensaje único se utiliza asimismo un vector de inicialización.

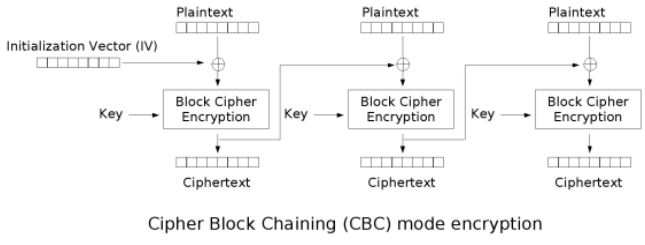


Figura 3.3.2.1 Cifrado en modo CBC

En nuestra aplicación es usado para cifrar los datos que se envían desde la aplicación móvil a la aplicación web, de manera que nadie pueda acceder a estos.

3.3.3 Firma Digital con Curva Elíptica

Una de las principales ventajas de la criptografía de clave pública es que ofrece un método para el desarrollo de firmas digitales. La firma digital permite al receptor de un mensaje verificar la autenticidad del origen de la información, así como verificar que dicha información no ha sido modificada desde su generación.

La firma digital se basa en la propiedad ya comentada sobre que un mensaje cifrado utilizando la clave privada de un usuario sólo puede ser descifrado utilizando la clave pública asociada. La firma digital, por tanto, es un cifrado del mensaje que se está firmando, pero utilizando la clave privada en lugar de la pública.

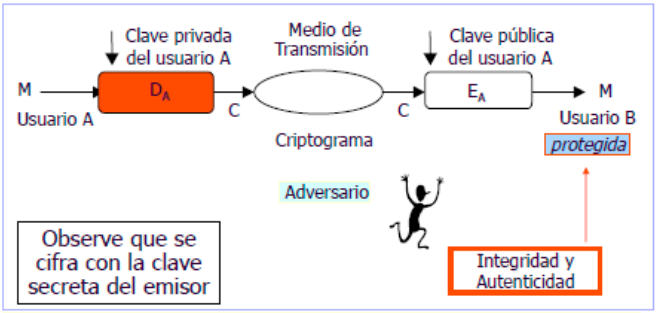


Figura 3.3.3.1 Firma digital con clave pública

DSA (Digital Signature Algorithm) es un estándar desde 1994 de Estados Unidos para firmas digitales y será el que utilicemos para la firma y verificación de los datos en el envío entre aplicaciones móvil y web.

Las fases de las que consta este algoritmo son la generación de las claves, la firma y la verificación, donde las dos primeras las realiza el emisor y la última el receptor.

La desventaja que posee este algoritmo es que requiere mucho más tiempo de cómputo que RSA, pero por ello hacemos uso de las curvas elípticas, ya que así requiere números de tamaños menores para brindar la misma seguridad y una mayor velocidad.

Capítulo 4.
Sistema de localización

En este capítulo se explica el sistema utilizado para determinar una distancia aproximada a partir de la intensidad de señal Bluetooth recibida.

4.1 Datos utilizados

Como ya hemos mencionado en el capítulo anterior, la tecnología inalámbrica Bluetooth es la encargada de detectar la proximidad del agresor respecto a la víctima. Para ello se hace uso de la intensidad de señal que se recibe del dispositivo del agresor, la cual posteriormente se transforma en una distancia en metros aproximada.

Para esta transformación, se hace uso de la siguiente aproximación matemática:

```
// Devuelve la distancia aproximada en metros entre dos dispositivos
double getDistance(double rssi, double txPower) {
    // El 2.7 es el valor de n y si no hay obstáculos de por medio se usa el valor 2
    return Math.pow(10d, ((double) txPower - rssi) / (10 * 2.7));
}
```

Figura 4.1.1. Función para calcular distancia aproximada

En dicha expresión se usa la siguiente notación:

- *txPower*: Valor de intensidad de señal entre los dispositivos a un metro de distancia.
- *rssi*: Intensidad de señal que se recibe del dispositivo Bluetooth que porta el agresor.
- *2.7*: Valor que varía dependiendo del entorno en el que se encuentre el dispositivo. Si el entorno estuviera libre de obstáculos, el valor sería 2, mientras que cuanto más tupido se encuentre, este valor se debe ir aumentando.

4.2 Funcionamiento

El funcionamiento del sistema de localización del agresor comienza cuando la aplicación empieza a realizar las búsquedas cada un tiempo determinado. Estas búsquedas se realizan usando los servicios de Android para que se puedan realizar en segundo plano y no haga falta tener la aplicación abierta.

Figura 4.2.1 Agresor no detectado

Así mismo, el tiempo entre búsquedas es configurable. Actualmente lo tenemos fijado cada 15 segundos ya que las búsquedas mediante Bluetooth en Android duran como máximo 12 segundos y para las pruebas nos viene mejor que sea tan seguido.

Por otra parte, si el agresor es detectado, la aplicación recogerá la intensidad de señal que le llega del dispositivo y posteriormente, realizará los cálculos con la aproximación matemática comentada en el anterior punto.

Figura 4.2.2 Agresor detectado fuera del rango de peligro

Una vez ya obtenida la distancia, se procede a notificar a la víctima de la cercanía del agresor con un nivel de aviso u otro, de los cuales ya hemos hablado y que varían respecto a la distancia resultante.

En caso de que el agresor supere la distancia de peligro asignada, además de avisar a la víctima también se iniciará automáticamente la grabación de vídeo con envío en tiempo real al centro de control y se le notificará mediante SMS a los contactos que tenga añadidos en la aplicación.

Figura 4.2.3 Agresor detectado dentro del rango de peligro

4.3 Dificultades

Durante el desarrollo de este sistema nos hemos ido encontrando con diferentes problemas o dificultades, las cuales se han intentado resolver de la mejor forma posible.

Por un lado, a la hora de detectar el dispositivo del agresor desde la aplicación móvil no se conseguía. El problema resultó ser que las librerías BLE de Android solo detectan dispositivos con la versión 4.1 o superior de Bluetooth, y en nuestro caso, el dispositivo que usábamos tenía la versión 4.0.

Aunque ambas versiones del Bluetooth son LE, la diferencia principal radica en que la 4.0 no tiene implementada la funcionalidad de usar el dispositivo como baliza, cosa que la versión 4.1 sí. Además, esta nueva versión presenta también mejoras en la transferencia de datos por lotes y en la coexistencia con la conexión móvil 4G LTE.

La solución tomada finalmente fue usar las librerías estándar de Bluetooth en Android para así poder probar el funcionamiento de la aplicación, ya que apenas hay diferencias con las LE y en un futuro se podrían adaptar sin ningún problema.

Y, por otro lado, el otro problema que nos surgió fue con la aproximación matemática de la intensidad de señal Bluetooth a una distancia, ya que dependiendo del entorno donde se encontrasen los dispositivos, la distancia se acercaba más o menos a la realidad.

Finalmente, la decisión final fue ajustar los valores para que la aproximación siempre se realice a la baja, ayudando a que en entornos muy tupidos también se le avise a la víctima de la presencia del agresor.

4.4 Mejoras

El sistema de búsquedas para detectar al agresor no es del todo eficiente actualmente, y es que este método puede llegar a consumir mucha batería. Esto es porque al intentar localizar el dispositivo del agresor, la aplicación realiza búsquedas constantemente lo que provoca este consumo excesivo. Además, hay que añadirle que como estamos usando las librerías estándar de Bluetooth, no nos beneficiamos de las mejoras del BLE.

La solución planteada para una mejora en el futuro es que cuando adaptemos las librerías Bluetooth a las LE, aprovechar y cambiar el método de detección, intentando conectar directamente con el dispositivo del agresor y no realizando búsquedas de hasta 12 segundos, logrando una mejora considerable del uso de la batería por parte de la aplicación.

Así, en caso de que se realizara la conexión con el dispositivo del agresor, significaría que éste se encuentra cerca, a lo cual la aplicación procedería luego ya con el resto de los pasos que ya se han mencionado y que no se verían alterados.

Capítulo 5.
Sistema de streaming

En este capítulo se hablará del sistema de envío de vídeo mediante streaming que se realiza cuando el agresor supera una distancia límite con respecto a la víctima.

5.1 Componentes

Este sistema es utilizado para proporcionar una visión clara del estado de la víctima cuando ésta se encuentra en peligro, ya sea porque el agresor se acercó demasiado y la grabación se activó automáticamente, o porque la propia víctima vio necesario su uso para notificar algún incidente relacionado.

Los componentes del sistema son los siguientes:

- Aplicación móvil: Encargada de realizar la grabación y envío del vídeo al servidor Wowza.
- Librería *libstreaming*: Librería para Android que realiza la conexión con el servidor automáticamente.
- Aplicación web: Realiza la conexión con el servidor y visualiza todos los vídeos en streaming que se estén enviando.

Servidor multimedia Wowza: Servidor que recibe el vídeo en streaming enviado desde las aplicaciones móviles y lugar donde se conecta la aplicación web para recibir los datos.

5.2 Librería libstreaming

Libstreaming es una API *open source* que permite, con sólo unas pocas líneas de código, transmitir la cámara y/o el micrófono de un dispositivo Android usando RTP sobre UDP. Como único requerimiento, se necesita que la aplicación use Android 4.0 o superior.

Los pasos que hemos llevado a cabo para realizar un envío de vídeo en streaming con esta librería son:

I. Instalación y configuración de la librería dentro del proyecto Android.

II. Añadir los permisos necesarios en el *manifest.xml*.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
```

Figura 5.2.1. Permisos para la librería libstreaming

III. Creamos una sesión de captura que se utilizará para el *preview*. La sesión será un objeto de tipo Session perteneciente a la librería libstreaming.

```
mSession = SessionBuilder.getInstance()
        .setCallback(this)
        .setSurfaceView(mSurfaceView)
        .setPreviewOrientation(0)
        .setContext(getApplicationContext())
        .setAudioEncoder(SessionBuilder.AUDIO_AAC)
        .setAudioQuality(new AudioQuality(8000, 16000))
        .setVideoEncoder(SessionBuilder.VIDEO_H264)
        .setVideoQuality(new VideoQuality(640,480,30,600000))
                .build();
```

Figura 5.2.2. Sesión de ejemplo para el streaming

IV. Creamos un cliente RTSP que se encargará de enviar el stream de vídeo capturado en la sesión al servidor Wowza. Este cliente es creado con un objeto de la clase *RtspClient*, pasándole como parámetro los datos de acceso al servidor.

```
mClient = new RtspClient();
mClient.setSession(mSession);
mClient.setCallback(this);

mClient.setCredentials("publisher", "mastermoviles");
mClient.setServerAddress("192.168.1.44", 1935);
mClient.setStreamPath("/live/canal1");
```

Figura 5.2.3 Cliente RTSP

V. Finalmente, ya solo tenemos que usar las funciones de *startStream()* o *stopStream()* para iniciar o parar el envío de vídeo en streaming al servidor Wowza.

Cabe mencionar que hemos hecho uso de esta librería debido a que además de ser *open source*, con unos pocos pasos nos permite enviar vídeo en streaming a través de un protocolo seguro como es el RTSP, mientras que para otras soluciones había que crear toda la infraestructura de envío a mano, lo cual era mucho más trabajoso.

5.3 RTSP y RTMP

Los protocolos que son usados para la transmisión en tiempo real de vídeo son RTSP y RTMP. Ambos son protocolos de comunicación que comparten más similitudes que diferencias y ayudan a hacer distribución de multimedia interactiva en tiempo real.

Por un lado, el protocolo RTMP, fue creado por Adobe para ayudar a los servidores web a distribuir contenido de baja latencia y bajo demanda en la Web de manera eficiente. La baja latencia es importante cuando se desea ver vídeos sin problemas en un navegador.

Y, por otro lado, el protocolo RTSP, el cual establece y controla uno o muchos flujos sincronizados de datos, ya sean de audio o de vídeo. El RTSP actúa como un mando a distancia mediante la red para servidores multimedia. Este protocolo soporta las siguientes operaciones:

- Recuperar contenidos multimedia del servidor: El cliente puede solicitar la descripción de una presentación por HTTP o cualquier otro método.
- Invitación de un servidor multimedia a una conferencia: Un servidor puede ser invitado a unirse a una conferencia existente en lugar de reproducir la presentación o grabar todo o una parte del contenido.
- Adición multimedia a una presentación existente: Particularmente para presentaciones en vivo, útil si el servidor puede avisar al cliente sobre los nuevos contenidos disponibles.

Además, en cuanto a la seguridad, ambos protocolos son seguros ya que utilizan mecanismos de seguridad ya sea a los protocolos de transporte (TLS) o dentro del mismo protocolo. Debido a esto, no es necesario el cifrado del vídeo que se envía mediante streaming ya que los propios protocolos se encargan de ello.

5.4 Funcionamiento

El esquema de funcionamiento de este sistema es el siguiente:

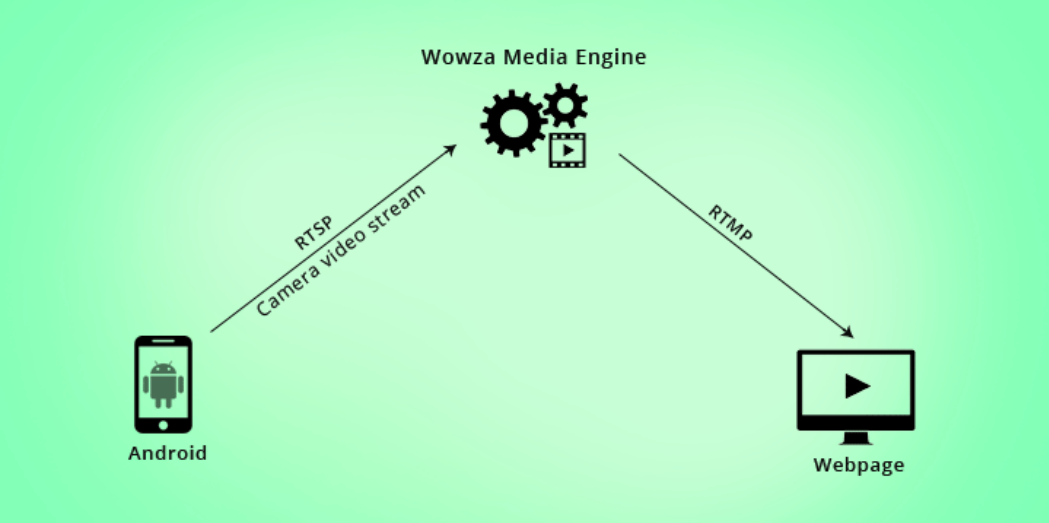


Figura 5.4.1 Esquema de funcionamiento del vídeo en streaming

Como podemos ver en la anterior figura, el dispositivo móvil envía el vídeo mediante el protocolo RTSP gracias a la librería *libstreaming* al servidor Wowza y este transmite el vídeo a la web por medio del protocolo RTMP.

Así mismo, en la aplicación móvil hay creadas tres funciones que se llaman dependiendo de la acción a realizar y que sirven para comunicar a la aplicación web el estado del envío del vídeo. Estas funciones son:

- *newUser*: Se realiza una llamada la primera vez que la aplicación móvil envía vídeo al centro de control. En esta función se envía la MAC del dispositivo y el enlace de conexión, los cuales quedan registrados en el sistema y, además, indica a la propia web de que el enlace está activo y por tanto se está enviando vídeo.
- *streamOnline*: A esta función se le invoca cuando el dispositivo ya está registrado en el centro de control y lo único que hace es indicarle que

el enlace de vídeo está activo. Una vez hecho esto, la propia web se encarga de reproducir el vídeo.

- *streamOffline*: Tiene el papel contrario a la anterior función, ya que esta se encarga de notificar a la web de que acaba de terminar la emisión de streaming. De esta manera, la web detendrá la reproducción de vídeo.

Además, a la hora de enviar el vídeo, la aplicación móvil tiene algunas características más añadidas:

- Notificación: En la pantalla principal de la aplicación se le indicará al usuario de que se está realizando una grabación.

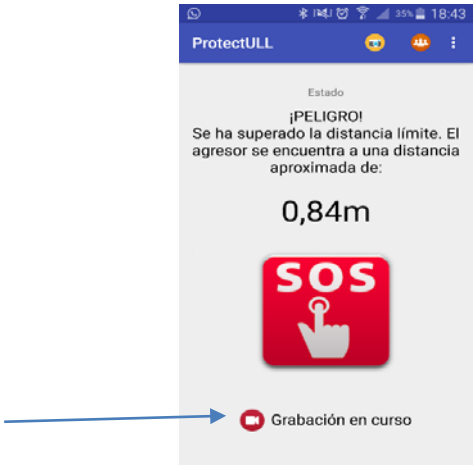


Figura 5.4.2 Notificación de la grabación en curso

- Guardado de vídeo: En caso de que la víctima no tenga datos en ese momento para enviar el vídeo, este se almacenará automáticamente en el móvil para que, si es el caso, pueda ser usado como prueba de algún acercamiento o acoso del agresor.
- Información del usuario: La víctima podrá configurar en la aplicación algunos de sus datos de interés como pueden ser el número de teléfono o su nombre, los cuales serán enviados al centro de control junto con el vídeo, para que estos puedan ponerse en contacto con ella lo antes posible.

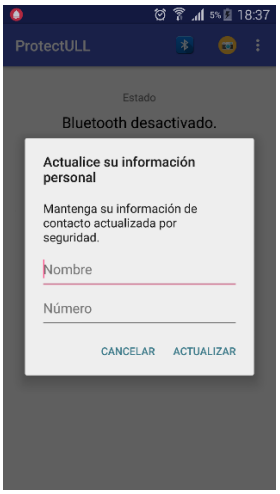


Figura 5.4.3 Información personal del usuario

5.5 Dificultades

A la hora de implementar este sistema en el proyecto, surgieron principalmente dos problemas que finalmente fueron solucionados.

Por un lado, antes de realizar la grabación y envío del vídeo, nos centramos en grabar únicamente. El problema en ese momento fue la imposibilidad de realizar la grabación cuando el móvil estaba bloqueado, ya que el error que generaba la aplicación era que no poseía una SurfaceView donde cargar la vista previa de lo que se está grabando.

La solución finalmente fue asignar la SurfaceView a WindowManager para que cuando el móvil estuviese bloqueado, pudiese cargarla. De esta manera, ya pudimos grabar vídeo, aunque el móvil estuviese bloqueado, en cualquier otra aplicación, etc.

Y, por otro lado, el otro problema fue cuando se adaptó la aplicación con la librería *libstreaming*, ya que el constructor que poseía la librería creaba una SurfaceView propia que en nuestro caso no nos interesaba porque no realizaba la grabación cuando el móvil estaba bloqueado.

En esta ocasión, la solución fue acceder a la propia librería y donde se creaba la SurfaceView automáticamente, añadir un constructor donde pasarle nuestra propia SurfaceView, de manera que así la librería nos permitiese grabar con el móvil en cualquier estado, mientras éste no esté apagado.

Capítulo 6. Herramienta

En este capítulo se realizará una definición, descripción del funcionamiento y ejemplos ilustrativos de la aplicación una vez implantados todos los módulos anteriores.

6.1 Definición

Para poder integrar todos los sistemas mencionados anteriormente, ha hecho falta programar cada uno de ellos de manera modular, para que luego a la otra de combinarlos, fuese mucho más sencillo y nos dieran el menor número de problemas.

En total, han hecho falta 15 clases en la aplicación. La finalidad de cada una de ellas es la siguiente:

Nombre	Función
Autentication.java	Activity de Android encargada de mostrar la pantalla para la autenticación mediante contraseña para poder acceder a la aplicación en sí.
BService.java	Servicio que se ejecuta en segundo plano y que realiza las búsquedas para comprobar que el agresor no está cerca.
BackgroundVideoRecorder.java	Servicio encargado de iniciar la grabación de vídeo y su envío mediante streaming.
BluetoothConnection.java	Clase donde se encuentra la función de respuesta cuando el agresor es detectado. Se encarga de comprobar la distancia aproximada, avisar al usuario, y si es necesario avisar a los contactos que tenga la víctima agregados y de iniciar la grabación de vídeo.
Config.java	Clase donde se configuran los datos de acceso a la aplicación web, como es su enlace, puerto, etc.
Contact.java	Clase con la finalidad de crear un objeto para cada contacto que se agregue. Se guarda el número, el nombre y si está activado en la lista de contactos.
ContactArrayAdapter.java	Clase que crea un ArrayAdapter de tipo Contacto. Se cargará en la actividad donde se muestran los contactos.

ContactList.java	Activity que muestra los contactos que tiene la víctima agregados en la aplicación para ser avisados además de tener las funciones también de poder añadirlos y editarlos.
DBase.java	Clase que posee la base de datos SQLite que posee la aplicación así como las diferentes funciones que nos dan acceso a los datos.
Inicio.java	Actividad principal, donde se muestra el estado de las búsquedas, grabación y avisos.
InicioFragment.java	Fragment de la Activity Inicio.
Notification.java	Clase en la que se definen todas las notificaciones de las que hace uso la aplicación.
PanicButton.java	Clase que define el botón de pánico y su función.
Request.java	Clase donde se encuentran las funciones encargadas de enviar la información desde la aplicación móvil a la aplicación web.
Streaming.java	Clase con los datos de acceso al servidor Wowza para el envío de vídeo en streaming.

Tabla 6.1.1 Clases usadas en el proyecto

Además, también se hace uso de un total de 10 layout encargados de indicar la posición y el diseño de todos los elementos en la aplicación, varias imágenes, la librería *libstreaming* y diferentes ficheros de configuración característicos de un proyecto Android.

A continuación, haremos un ejemplo ilustrativo de una situación donde haría falta el uso de esta aplicación, así como del funcionamiento de la misma en cada fase del ejemplo.

6.2 Configuración de la aplicación móvil

La aplicación móvil se basa por su sencillez a la hora de tratar con ella, para que así cualquier persona con un nivel de conocimientos mínimo pueda utilizarla.

Actualmente, la primera vez que se inicia la aplicación, ésta tiene una contraseña por defecto con valor *123*. En la siguiente figura podremos ver una captura de pantalla camuflada como configuración del dispositivo donde hay que introducir la contraseña para la privacidad de la víctima, como ya dijimos.

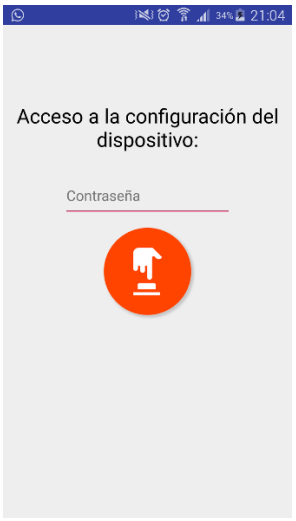


Figura 6.2.1 Pantalla de acceso a la aplicación

Una vez introducida la contraseña, lo primero que deberá hacer el usuario será acceder al menú de la aplicación tras introducir la contraseña y añadir una nueva para garantizar que sólo ella tendrá acceso a la aplicación.

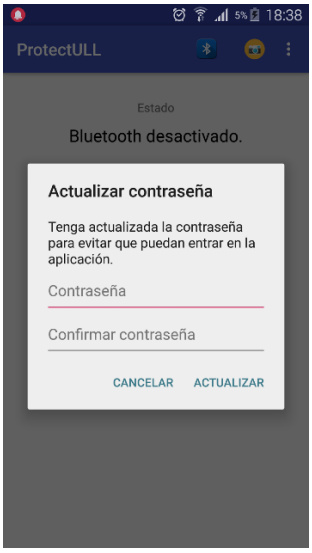


Figura 6.2.2 Cambio de contraseña.

Otro de los pasos que deberá realizar será introducir su nombre y su número de teléfono para que esta información pueda ser enviada a la aplicación web encargada del control, y si es el caso, estos puedan ponerse en contacto con la víctima.

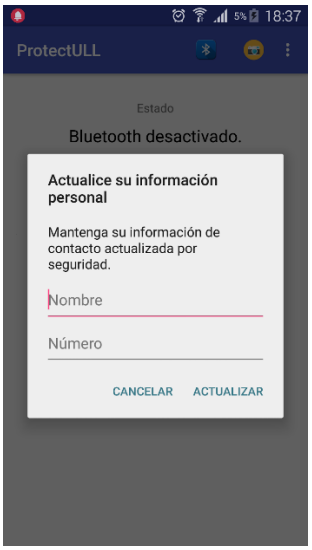


Figura 6.2.3 Actualizar datos de usuario

Finalmente, también podrá añadir los contactos que desee para su notificación en caso de detectar al agresor cerca.

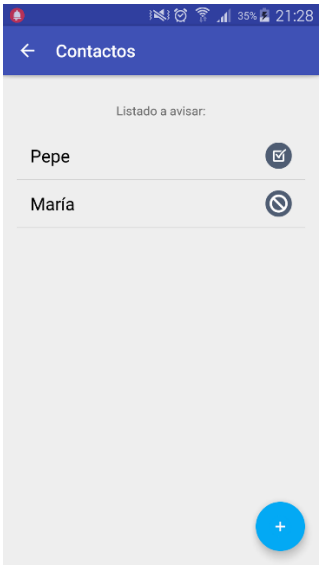


Figura 6.2.4 Contactos en la aplicación

Para ello, podrá crear un nuevo contacto a mano o incluir uno que tenga añadido en su lista de contactos del teléfono.

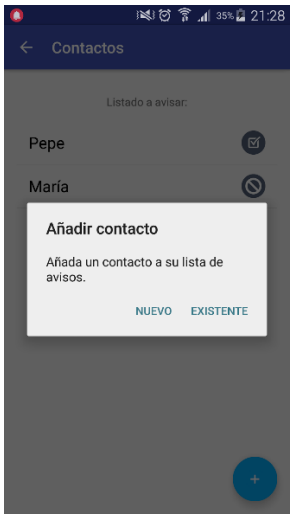


Figura 6.2.5 Menú para añadir nuevo contacto

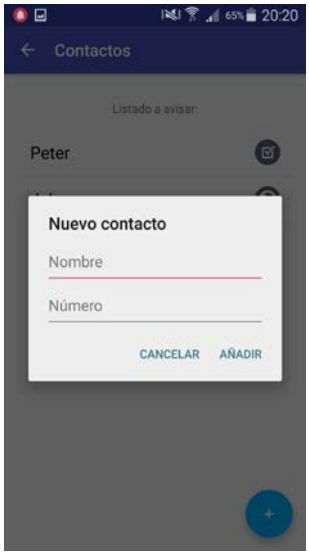


Figura 6.2.6 Añadir nuevo contacto en la aplicación móvil

Dentro de pantalla de Contactos también podrá seleccionar cualquiera de los contactos que tenga añadidos dejándolo pulsado un momento, a lo cual sale un menú con las siguientes opciones:

- Editar contacto: Permitirá a la víctima editar el nombre o el número de teléfono del contacto seleccionado.

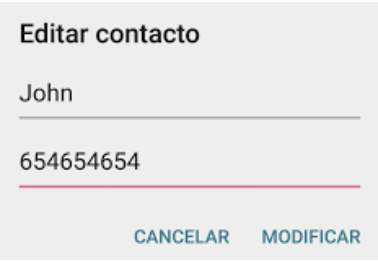


Figura 6.2.7 Editar contacto

- Habilitar/deshabilitar contacto: La víctima podrá tener añadidos contactos que pueden ser avisados o no, dependiendo de si están habilitados. Esto permite que pueda tener contactos que quizá no desea notificar en un momento determinado.

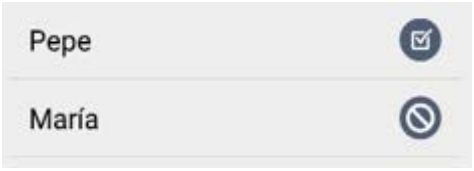


Figura 6.2.8 Contactos habilitados y deshabilitados

- Eliminar contacto: Elimina el contacto de la lista.

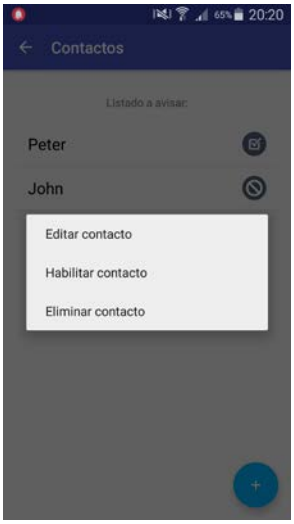


Figura 6.2.9 Menú emergente al pulsar sobre un contacto

Cabe señalar que todas estas funciones están disponibles siempre para su edición y no sólo la primera vez que se ejecuta la aplicación.

6.3 Uso de la aplicación móvil

La aplicación siempre se estará ejecutando en segundo plano una vez iniciada a no ser que la víctima decidiese por su propia voluntad cerrarla, ya que se modificó la función del botón atrás en Android de manera que no cierre la aplicación al pulsarlo, sino que se quede ejecutando en segundo plano. Antes de cerrarla, la víctima deberá confirmar que así lo desea.

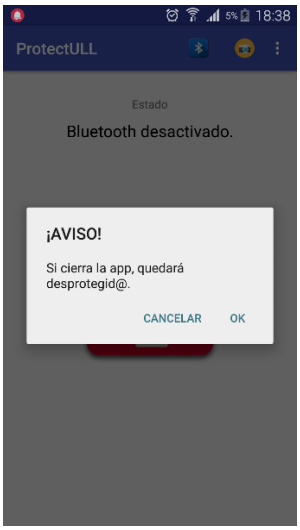


Figura 6.3.1 Confirmación para cerrar aplicación

Además, también comprueba antes de realizar las búsquedas que el Bluetooth esté activado para sino enviarle una notificación, pudiendo activarlo con un botón que aparece en la aplicación automáticamente.

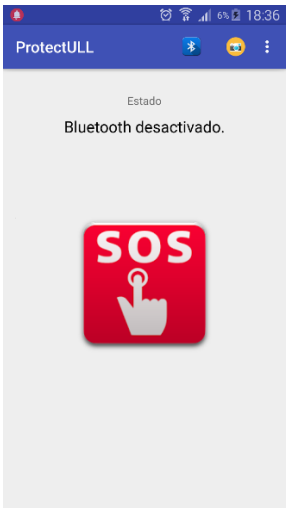


Figura 6.3.2 Aplicación con el Bluetooth desactivado

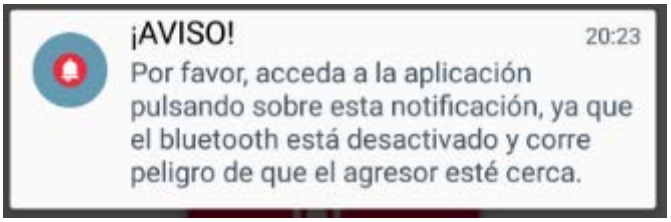


Figura 6.3.3 Notificación de Bluetooth desactivado

Una vez la aplicación está iniciada y el Bluetooth activado, se procede a realizar las búsquedas cada un periodo de tiempo determinado. Aquí pueden pasar tres cosas:

En primer lugar, que no sea detectado el agresor, por lo que la aplicación indicará que no hay peligro y no enviará ningún tipo de notificación especial a la víctima.

En segundo lugar, que el agresor sea detectado, pero no esté dentro de la zona de peligro, notificando en este caso a la víctima para que pueda estar alerta.

Y, en tercer lugar, que el agresor sea detectado y además supere la distancia de peligro. En esta situación, además de notificar a la víctima, se le enviaría un SMS a los contactos que tenga añadidos en la aplicación y se iniciaría automáticamente la grabación de vídeo con envío en streaming si hay cobertura y, en caso contrario, guardándolo en el dispositivo.

Finalmente, la víctima también podrá hacer uso del botón de pánico para llamar automáticamente al servicio de emergencias en caso de que le haga falta o realizar una grabación de vídeo con streaming si lo viese necesario pulsado los botones habilitados para ello.

Figura 6.3.4 Botón de pánico

6.4 Uso de la aplicación web

Como ya hemos dicho, la aplicación web es la encargada de reproducir todos los vídeos que se envían mediante streaming desde las aplicaciones móviles y guardar todas las conexiones y dispositivos que han hecho uso de la misma. La web consta de cuatro secciones:

En primer lugar, una página de inicio cuya única finalidad es presentar el proyecto.

Figura 6.4.1 Página de inicio de la web

En segundo lugar, la pestaña *Live*, en la cual se visualizan todos los vídeos que se están enviando en streaming en el momento. En el lado izquierdo se formará una lista de todos los vídeos con la MAC del dispositivo y la fecha y hora de inicio de la grabación. Si pinchamos en cualquier elemento de la lista nos llevará automáticamente al vídeo asociado, los cuales se cargan en el lado derecho de la web.

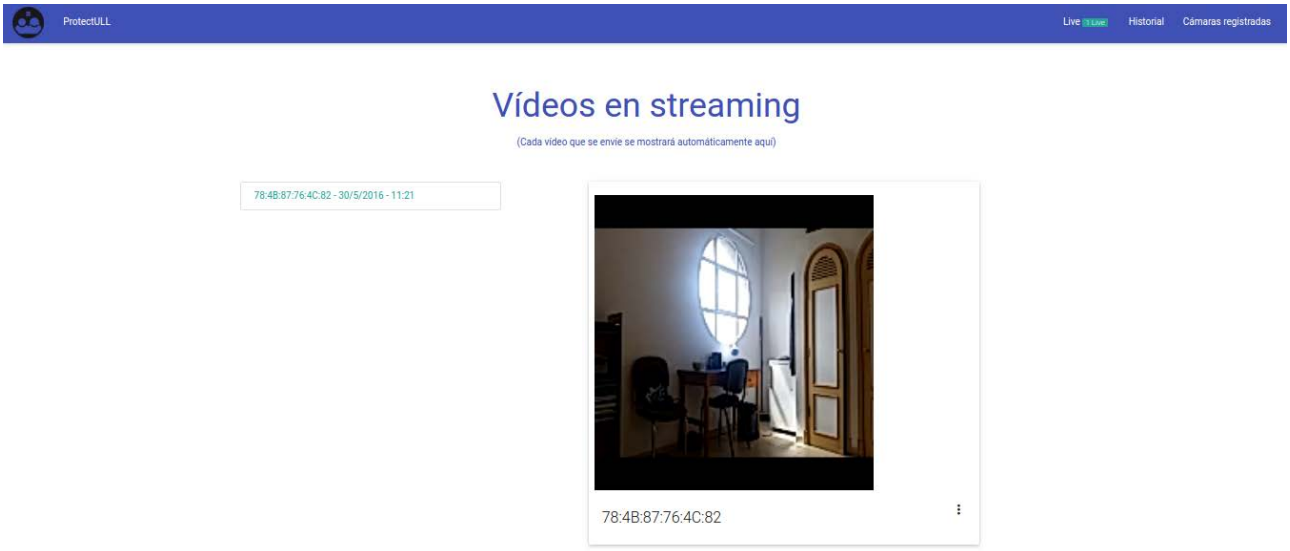


Figura 6.4.2 Pestaña live en la aplicación web

En tercer lugar, la sección *Historial*, donde se van guardando en una tabla todas las conexiones que se han ido realizando con el sistema, con la MAC del dispositivo móvil, el nombre y número de teléfono de contacto de la víctima y la hora del envío del vídeo.

Además, también posee un botón que permite eliminar el historial guardado hasta ese entonces.

Figura 6.4.3 Historial almacenado en la web

Y finalmente, la pestaña de Cámaras registradas, donde se encuentran todos los dispositivos que han hecho uso alguna vez de la web.

Figura 6.4.4 Dispositivos almacenados en la web

Capítulo 7. Implementación

En este capítulo se hará una definición de los pasos llevados a cabo para la implementación de la aplicación, fechas establecidas de entregas y problemas originados durante el desarrollo.

7.1 Planificación

La planificación para el desarrollo del Trabajo de Fin de Grado consta de los hitos que se encuentran en la siguiente tabla:

Fechas	Hito
22/02/2016 – 06/03/2016	Crear una aplicación que, usando BLE, vibre uno de los dos móviles cuando uno de ellos se acerque al otro. Además, realizar corresponden con la realidad.
07/03/2016 – 20/03/2016	Llamar a un número o enviar un aviso en caso de que la intensidad de señal supere un umbral X.
21/03/2016 – 03/04/2016	Añadir un botón de pánico que al presionarlo llame a un número o envíe un aviso.
04/04/2016 – 17/04/2016	Comienzo de grabación automática cuando surja algún evento (botón de pánico o se acerca demasiado).
18/04/2016 – 01/05/2016	Realizar envío de vídeo streaming y en caso de no tener cobertura, almacenarlo para que luego pueda ser enviado. Además, realizar un estudio de si debemos usar algún tipo de compresión de imagen.
02/05/2016 – 15/05/2016	Ver si existe alguna manera de instalar la aplicación sin dejar rastro (que no haya icono, permisos para poder acceder...).
16/05/2016 – 29/05/2016	Realizar estudio de cómo desarrollar la parte encargada de recibir esta información como si se tratara de la policía u otro servicio encargado de controlar estos temas. Además, comenzar con el desarrollo de la memoria.
Mes de Junio	Retoques finales del proyecto y memoria para su entrega.

Tabla 7.1.1 Hitos en el desarrollo del proyecto

De los hitos que hemos visto en la anterior tabla, algunos se han realizado en un tiempo menor al programado, como puede ser el de crear el botón de pánico o el de ocultar la aplicación para mejorar la privacidad de la víctima, pero también hay otros que se han alargado más de lo esperado, siendo en este caso la grabación de vídeo con el móvil bloqueado (problema que se comentó anteriormente) o el envío de vídeo mediante streaming.

A lo largo del periodo de desarrollo del proyecto, se han realizado varias reuniones con las tutoras para ir viendo los avances, resolver problemas y redirigir aquellos que tenían una solución más difícil.

Además, también hemos realizado diversos cursos introductorios para que nos fuese más fácil el inicio, como son "Introducción al lenguaje de programación Android" y "Envío de vídeo mediante streaming", ambos realizados por componentes de CryptULL.

7.2 Esquema de comunicaciones

El esquema de comunicaciones usado en el proyecto es el siguiente:

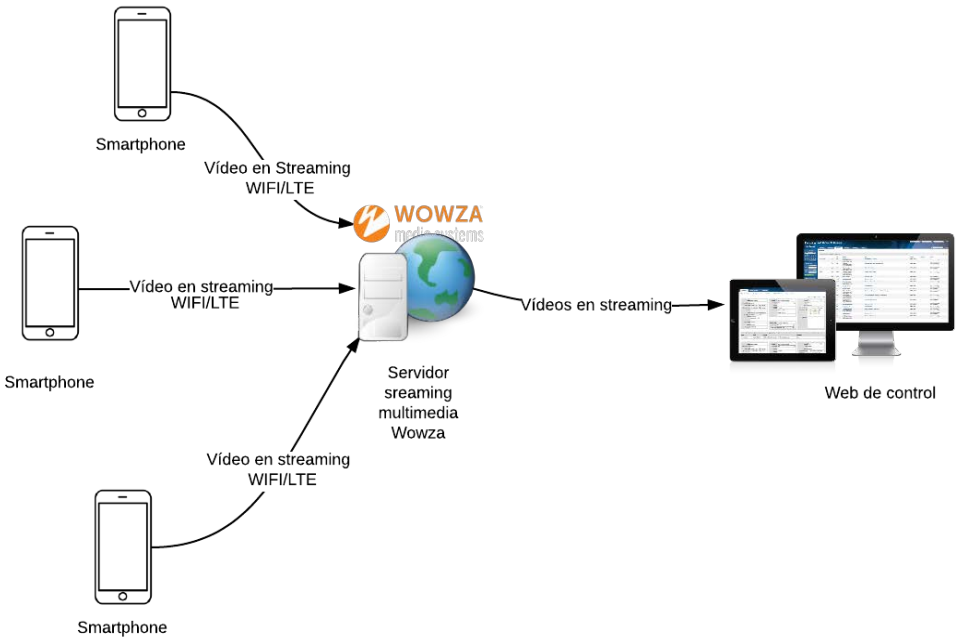


Figura 7.2.1 Esquema de comunicaciones

Como podemos ver, las aplicaciones móviles de las víctimas se conectan al servidor Wowza a través de Wi-Fi o LTE para el envío del vídeo en streaming, el cual luego es reproducido desde la aplicación web.

Este esquema está compuesto por los siguientes elementos:

- Dispositivos móviles: Encargados de realizar el envío del vídeo en streaming con la aplicación instalada.
- Servidor Wowza: Recibe los vídeos que se están enviando y es el punto de acceso para poder visualizarlos.
- Aplicación web: Reproduce todos los vídeos que son enviados desde la aplicación móvil al servidor Wowza.

7.3 Base de datos para la aplicación móvil

En la aplicación móvil tenemos una base de datos SQLite que consta de tres tablas para guardar la información de la que hace uso, cuya estructura es la que vemos a continuación:

Tablas	Funciones	Descripción
CONTACTOS (Contactos a los que se le notifica mediante SMS si están habilitados)	insertarCONTACTO	Inserta un nuevo contacto en la base de datos.
	modificarCONTACTO	Actualiza alguno de los campos del contacto que seleccione la víctima.
	borrarCONTACTO	Borra un contacto de la base de datos.
	recuperarCONTACTO	Devuelve un contacto en específico. Como clave se usa el número de teléfono de cada contacto.
	recuperarCONTACTOS	Devuelve una lista con todos los contactos que se encuentren en la base de datos.
INFO_USUARIO (Información de la víctima que se envía junto con el vídeo en streaming)	insertarINFO_USUARIO	Inserta la información de la víctima (nombre y número de teléfono). Esta función solo es llamada si no se encuentra ningún registro en la tabla.
	modificarINFO_USUARIO	Modifica alguno de los campos de la información de la víctima.
	borrarINFO_USUARIO	Borra la información de la víctima de la base de datos.
	recuperarINFO_USUARIO	Devuelve la información que tiene guardada el usuario.
CONFIG_APP (Contraseña de la aplicación)	insertarCONFIG_APP	Inserta la contraseña de acceso a la aplicación. Esta función solo es llamada si la tabla está vacía, creando una contraseña por defecto.
	modificarCONFIG_APP	Modifica la contraseña de acceso a la aplicación.
	recuperarCONFIG_APP	Devuelve la contraseña usada para el acceso a la aplicación.

Tabla 7.3.1 Base de datos de la aplicación móvil

Como ejemplo de una función de la base de datos podemos usar i*nsertarCONTACTO()*, a la cual le pasamos como parámetros el nombre, número de teléfono y el estado del contacto (habilitado o deshabilitado) al que quiere notificar la víctima en caso de aproximación del agresor.

```
// Inserta un contacto nuevo
public void insertarCONTACTO(String tlf, String nom, int act) {
    tlf = tlf.replace(" ","");
    tlf = tlf.replace("+34","");

    SQLiteDatabase db = getWritableDatabase();
    if(db != null){
        ContentValues valores = new ContentValues();
        valores.put("telefono", tlf);
        valores.put("nombre", nom);
        valores.put("activo", act);
        db.insert("contactos", null, valores);
        db.close();
    }
}
```

Figura 7.3.1 Función *insertarCONTACTO()*

Antes de insertar los datos en la base de datos como un nuevo contacto, le quitamos los espacios en blanco y el prefijo que puede tener el número de teléfono ya que en la base de datos se guarda como un entero y no como una cadena.

Esta función es llamada cuando la víctima pulsa sobre el botón de añadir un nuevo contacto o cuando selecciona uno de su lista del teléfono.

7.4 Base de datos para la aplicación web

Para la aplicación web hacemos uso de MongoDB y sus modelos en NodeJS para guardar la información de los dispositivos y el historial de conexiones que se han realizado en la aplicación. Actualmente, hacemos uso de dos modelos: uno que guarda el historial de conexiones con la web, y otro los dispositivos que han hecho uso del sistema.

A continuación, en la siguiente tabla describimos más detalladamente la función de cada uno de ellos:

Modelos	Descripción
Camara.js	Esquema que guarda el enlace que tiene cada dispositivo cuando envía vídeo en streaming al servidor Wowza, la MAC del dispositivo, el estado (boolean) indicando si está activo el vídeo o no, la hora de inicio y finalización de la última retransmisión y el nombre y número de teléfono de la víctima.
Historial.js	Esquema que guarda todas las conexiones que se han realizado en la aplicación web. Los datos que contiene son la MAC del dispositivo, el nombre y número de teléfono de la víctima y la fecha y hora de la conexión.

Tabla 7.4.1 Base de datos de la aplicación web

7.5 Clase Contacto

Para definir los contactos a los que se les notifica mediante SMS cuando el agresor es detectado muy próximo a la víctima, hemos creado un objeto Contact definido con la siguiente estructura:

```
private String name;
private String telefono;
private int activo;

public Contact(String nm, String num, int act){
    name = nm;
    telefono = num;
    activo = act;
}
```

Figura 7.5.1 Estructura del objeto Contact

Este objeto consta de tres elementos:

- name: Nombre del contacto al que se desea notificar.
- telefono: Número de teléfono que se usará para ponerse en contacto con la víctima en caso de proximidad.
- activo: Indica si el contacto está habilitado o deshabilitado para las notificaciones.

7.6 Servicios en la aplicación Android

En la aplicación móvil hemos hecho uso de los Servicios, los cuales nos permiten ejecutar tareas mientras la aplicación está en segundo plano. Concretamente, los hemos utilizado para las siguientes tareas.

7.6.1 Búsquedas en segundo plano

Como ya se comentó anteriormente, la aplicación realiza búsqueda cada cierto tiempo, pudiendo ser este configurable. Estas búsquedas se realizan usando un Servicio.

El funcionamiento consiste en la ejecución de un Runnable cada 15 segundos (tiempo que puede modificarse) donde se inicia el Servicio. Una vez es iniciado, éste procede a realizar las búsquedas usando Bluetooth, las cuales tardan alrededor de 12 segundos. Finalmente, si el dispositivo del agresor es encontrado se para el Servicio y se ejecuta una Callback de la librería Bluetooth que tiene las acciones correspondientes, mientras que, si el agresor no es detectado, se detiene el Servicio automáticamente hasta que desde el Runnable se vuelva a ejecutar.

7.6.2 Grabación de vídeo

Para la grabación de vídeo hemos hecho uso de un Servicio ya que queríamos que ésta se iniciara independientemente de si la víctima se encontraba en la aplicación o no.

En este caso, todo el sistema de grabación se encuentra dentro del propio Servicio, el cual es iniciado cuando el agresor supera una distancia límite o cuando la propia víctima acciona la grabación, mientras que para pararlo es necesario pulsar el botón habilitado para ello.

7.7 Librería Volley

Volley es una librería desarrollada por Google para optimizar el envío de peticiones HTTP desde las aplicaciones Android hacia servidores externos. Este componente actúa como una interfaz de alto nivel, liberando al programador

de la administración de hilos y procesos tediosos de parsing, para permitir publicar fácilmente resultados en el hilo principal.

Entre las características de Volley destacan:

I. Gestión de prioridades y de la cola de peticiones.

II. Gestión de memoria y caché.

III. Facilidad para ampliar y personalizar la librería a nuestras necesidades.

IV. Cancelación de peticiones.

7.8 Envío de datos desde móvil a web

En la siguiente figura veremos como ejemplo la función que se utiliza para enviar los datos del dispositivo móvil de la víctima a la web:

```
//Función que registra a un usuario en el servicio web la primera vez que usa la app
public static void newUser(String MAC) {

    HashMap<String, String> params = new HashMap<String, String>();
    params.put("name", MAC);
    params.put("server", StreamingConfig.STREAM_SHORT_URL);

    JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(com.android.volley.Request.Method.POST, Config.SERVER_URL + "/camara", new JSONObject(params),
            new com.android.volley.Response.Listener<JSONObject>() {
                @Override
                public void onResponse(JSONObject response) {
                    Log.i("Volley newUser Request ", response.toString());
                }
            },
            new com.android.volley.Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    Log.i("Volley newU Req Error ", error.toString());
                }
            });

    Config.requestQueue.add(jsonObjectRequest);
}
```

Figura 7.8.1 Función para enviar datos del dispositivo a la web

Como se puede ver, el funcionamiento es bastante sencillo. Para ello primero se crea un HashMap con clave/valor de tipo String donde guardamos la MAC del dispositivo móvil de la víctima y la URL de conexión con el servidor Wowza para acceder al vídeo en streaming.

Seguidamente, creamos la variable de tipo JsonObjectRequest, la cual será la que posteriormente se pase para realizar la conexión con la aplicación web. Los parámetros que se le pasan al constructor de esta variable son:

- Petición: Tipo de petición que le realizamos al servidor. En este caso usamos POST, ya que se creará un nuevo recurso en el servidor.
- URL: Dirección a la que se le va a realizar la petición en el servidor.
- Parámetros: Datos que enviamos en la petición. En este caso es la MAC y la dirección del servidor Wowza donde se está enviando el vídeo en streaming.

7.9 Desafíos

Los desafíos que hemos encontrado a la hora de desarrollar el proyecto han sido varios.

En primer lugar, el reto de aprender Android desde cero para desarrollar una aplicación con bastantes funcionalidades. Para ello, primero nos hemos centrado en aprender la estructura general del lenguaje, viendo para que sirve y para que se utiliza cada cosa. Una vez hecho esto, fuimos ya profundizando más en el tema que nos tocase abordar en cada momento (Bluetooth, Wi-Fi, grabación de vídeo, etc.).

Seguidamente, y no muy distinto, aprender otro lenguaje, en este caso NodeJS y Angular para la aplicación web. Los pasos a realizar fueron bastantes parecidos, centrándonos primero en aprender cómo funcionaba la estructura general de un proyecto en NodeJS y ya luego profundizando en cada aspecto a desarrollar.

Y finalmente, el resto de desafíos ya fueron apareciendo en forma de problemas y errores para los que había que buscar soluciones. Los problemas más importantes fueron con la grabación de vídeo mientras el móvil se encontrada en estado de bloqueo y también con el Bluetooth Low Energy, como ya hemos comentado anteriormente en algunos puntos.

Capítulo 8. Presupuesto

En este capítulo hablaremos del presupuesto total necesario que ha hecho falta para el desarrollo de este proyecto.

8.1 Personal

El desarrollo del proyecto ha sido llevado a cabo en su totalidad por José Ángel Concepción Sánchez junto con la participación de la tutora y cotutora para la resolución de dudas y diferentes problemas que han ido apareciendo, y para aportar sugerencias y mejoras.

En total, el proyecto se ha desarrollado en 300h estimadas, cumpliendo los requisitos de la Guía Docente de la asignatura. Para cada una de las tareas marcadas inicialmente se han hecho uso de las siguientes horas:

Tarea	Nº de horas
Crear una aplicación que, usando BLE, vibre uno de los dos móviles cuando uno de ellos se acerque al otro.	40h
Llamar a un número o enviar un aviso en caso de que la intensidad de señal supere un umbral X.	10h
Añadir un botón de pánico que al presionarlo llame a un número o envíe un aviso.	10h
Comienzo de grabación automática cuando surja algún evento (botón de pánico o se acerca demasiado).	60h
Realizar envío de vídeo en streaming y en caso de no tener cobertura, almacenarlo.	20h
Ver si existe alguna manera de instalar la aplicación sin dejar rastro (que no haya icono, permisos para poder acceder…).	30h
Desarrollar la parte encargada de recibir esta información como si se tratara de la policía u otro servicio encargado de controlar estos temas.	100h
Funcionalidades opcionales añadidas	30h
Total	300h

Tabla 8.1.1 Tabla de horas por tarea realizada

En este desglose no se ha tenido en cuenta el tiempo dedicado a la asistencia a seminarios, las reuniones con las tutoras del proyecto, la redacción de documentos y otras tareas no relacionadas directamente con el proyecto.

8.2 Componentes

Los componentes de los que se han hecho uso a lo largo del desarrollo del proyecto son los que se encuentran a continuación en la tabla.

Componente	Coste
Samsung Galaxy S5	400€
Wiko LENNY	90€
Adaptador Bluetooth para pc	10€
Servidor Wowza (licencia gratuita)	0€
Total	500€

Tabla 8.1.1 Tabla de componentes usados en el proyecto

Cabe destacar que todos estos componentes han sido aportados por las tutoras o bien, el alumno ya los poseía, por lo que no ha hecho falta comprar nada nuevo.

8.3 Coste total

El presupuesto total aproximado necesario para el desarrollo del proyecto, sumando los cálculos de los anteriores puntos y tomando como referencia un costo de 10€/hora, es de un total de:

Componente	Coste
Total en componentes	500€
Total horas (300*10)	3000€
Total	3500€

Tabla 8.3.1 Tabla del presupuesto total

Capítulo 9. Conclusiones y trabajos futuros

A modo de conclusión, creemos que es necesaria la implantación de este sistema de aviso para proteger la vida de todas esas personas que están sufriendo la violencia de género en la actualidad y no tienen su caso catalogado como de riesgo alto o extremo, por lo que no se controla su situación. Con un poco de información que posea la víctima, se pueden evitar muchas situaciones.

De esta manera, este proyecto intentará cubrir este vacío informativo, aprovechando la ventaja de que es un sistema mucho más económico que los que hay implantados actualmente para que pueda llegar a un mayor número de personas.

Por otro lado, de cara al futuro, pretendemos que la aplicación reciba nuevas mejoras en forma de funcionalidades que permitan que ésta esté cada vez más completa, segura y la más importante, que sea portable a otros sistemas operativos como son iOS y Windows Phone, para que pueda ser accesible desde cualquier dispositivo móvil.

Finalmente, a nivel personal, el Trabajo de Fin de Grado me ha introducido en el desarrollo de aplicaciones móviles, temática que he descubierto que me motiva y en la que seguiré profundizando y aprendiendo de aquí en adelante. También, me ha dado la oportunidad de conocer al grupo CryptULL y aprender mucho de ellos y de su experiencia (curso de Android, de envío de vídeo en streaming, reuniones, etc.). Y para acabar, me ha enseñado que, aunque las cosas un día, dos, o tres no terminen de salir, con paciencia y ganas, siempre se encuentra la solución.

Capítulo 10.
Conclusions and future works

In conclusion, we believe that the implementation of this protection system is necessary to protect the lives of all those people who are suffering gender violence today. With a little information for the victim, they will can avoid many awkward situations.

Thus, this project will attempt to fill this information gap, taking advantage of that is a lot cheaper than currently systems, so that it can reach a greater number of people.

On the other hand, for the future, we hope that the application receives new enhancements in form of features that allow it to be ever more complete, reliable and most important, portable to other operating systems such as iOS and Windows Phone for it can be accessible from any mobile device.

Finally, on a personal level, the Final Degree Work has introduced me in mobile application development, theme I've found that motivates me. I will continue to deepen and learning from here on. It has also given me the opportunity to meet the group CryptULL and learn a lot from them and their experience (Android course, sending streaming video, meetings, etc.). And finally, I learn that with patience and enthusiasm, always I found the solution to the problems.

Bibliografía

[1] Bluetooth LE. https://en.wikipedia.org/wiki/Bluetooth_low_energy.

[2] Bluetooth. https://es.wikipedia.org/wiki/Bluetooth

[3] LTE. https://es.wikipedia.org/wiki/Long_Term_Evolution.

[4] Wi-Fi. https://en.wikipedia.org/wiki/Wi-Fi.

[5] Android Streaming Live Camera Video to Web Page. http://www.androidhive.info/2014/06/android-streaming-live-camera-video-to-web-page/.

[6] NodeJS. https://nodejs.org/en/

[7] Librería Volley. http://www.hermosaprogramacion.com/2015/02/android-volley-peticiones-http/

[8] SQLite in Android. https://developer.android.com/reference/android/database/sqlite/package-summary.html

[9] MongoDB in Node.js. https://docs.mongodb.com/ecosystem/drivers/node-js/

[10] Streaming Video Protocols. https://www.streamingvideoprovider.com/streaming_video_protocols.html

[11] Tutorial de uso de la librería libstreaming. https://mastermoviles.gitbooks.io/graficos-y- multimedia/content/procesamiento-android.html

[12] AES. https://es.wikipedia.org/wiki/Advanced_Encryption_Standard

[13] Cifrado por bloques. https://es.wikipedia.org/wiki/Cifrado_por_bloques#Cipher-block_chaining_.28CBC.29

[14] Firma digital. http://www.cert.fnmt.es/content/pages_std/html/tutoriales/tuto7.htm

[15] Por una sociedad libre de violencia de género – Gobierno de España. http://www.violenciagenero.msssi.gob.es/

[16] Recopilatorio de conocimiento sobre la violencia de género. http://observatorioviolencia.org/

[17] La violencia de género como problema actual, Directorio de abogados. http://www.dab.com.ar/articles/137/la-violencia-de-g%C3%A9nero-como-problem%C3%A1tica-actual.aspx

[18] Estadística de Violencia Doméstica y Violencia de Género 2015. http://www.ine.es/prensa/np906.pdf.

[19] 3MT Domestic Violence GPS Proximity Notification System. http://apav.pt/25/images/PDF/3MT_Domestic_Violence_GPS_Proximity_Notification_System.pdf

[20] Protocolo de actuación del Sistema de seguimiento por medios telemáticos del cumplimiento de las medidas y penas de alejamiento en materia de violencia de género. https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwiQmePWqfPMAhWDbhQKHRZYD3cQFggjMAE&url=http%3A%2F%2Fwww.poderjudicial.es%2Fstfls%2FCGPJ%2FOBSERVATORIO%2520DE%2520VIOLENCIA%2520DOM%25C3%2589STICA%2FFICHEROS%2F20140210%2520Protocolo%2520VG%2520%25E2%2580%2593%2520Seguimiento%2520telem%25C3%25A1tico%2520cumplimiento%2520de%2520medidas.pdf&usg=AFQjCNFCjab1qIfb3k-zMgLrCckK5ZDRBA&bvm=bv.122676328,d.d24

[21] Ellsberg, M., Arango, D. J., Morton, M., Gennari, F., Kiplesund, S., Contreras, M., Watts, C. 2015. Prevention of violence against women and girls: what does the evidence say? The Lancet, 385(9977), 1555-1566.

[22] Whitfield, D. 2001. The Magic Bracelet: technology and offender supervision. Waterside Press.

Apéndice A.
Código destacable

A.1. Inicio de grabación de vídeo

```
// Método llamado despues de crear la surface (inicializa y empieza la grabación)
@Override
    public void surfaceCreated(SurfaceHolder surfaceHolder) {
        // Si el wifi está desconectado, guardamos el vídeo.
        if (!mWifi.isConnected()) {

            camera = Camera.open();
            Camera.Parameters params = camera.getParameters();
            params.setFocusMode(Camera.Parameters.FOCUS_MODE_CONTINUOUS_PICTURE);
            camera.setParameters(params);

            mediaRecorder = new MediaRecorder();
            camera.unlock();
            mediaRecorder.setPreviewDisplay(surfaceHolder.getSurface());
            mediaRecorder.setCamera(camera);
            mediaRecorder.setOrientationHint(90);
            mediaRecorder.setAudioSource(MediaRecorder.AudioSource.CAMCORDER);
            mediaRecorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);

            mediaRecorder.setProfile(CamcorderProfile.get(CamcorderProfile.QUALITY_HI
            GH));

            mediaRecorder.setOutputFile(
              Environment.getExternalStorageDirectory()+"/"+
                DateFormat.format("yyyy-MM-dd_kk-mm-ss", new Date().getTime())+".mp4"
            );

            try { mediaRecorder.prepare(); }  catch (Exception e) {}

            mediaRecorder.start();
        }  // Sino, iniciamos el vídeo en streaming
        else {  toggleStreaming();  } }
```

A.2. Receiver cuando se detecta al agresor

```
// Se llama al receiver cada vez que encuentra un dispositivo nuevo
private final BroadcastReceiver receiver = new BroadcastReceiver(){
    @Override
    public void onReceive(Context context, Intent intent) {
        String action = intent.getAction();


            if(BluetoothDevice.ACTION_FOUND.equals(action)) {
                String name = intent.getStringExtra(BluetoothDevice.EXTRA_NAME);
                // Si encuentra el dispositivo del agresor/a entra:
                if(name.equals(nombre_dispositivo)){
                // Calculamos la distancia aproximada
                   double  rssi  =  intent.getShortExtra(BluetoothDevice.EXTRA_RSSI,
                   Short.MIN_VALUE);
                    double distance = getDistance(rssi, px);
                    deviceFound = true; // Encontró el dispositivo
                    // Parseamos el resultado para que muestre dos decimales
                    DecimalFormat df = new DecimalFormat("#.##");
                    String rdistance = df.format(distance);


                   TextView            rssi_msg            =            (TextView)
                   mActivity.findViewById(R.id.res_busqueda);
                   TextView            res_dist            =            (TextView)
                   mActivity.findViewById(R.id.res_distancia);


                    // Si está dentro de la distancia límite se le avisa
                    if(distance < getDistancia_limite()){
                       rssi_msg.setText(context.getString(R.string.peligro) + "\n" +
                       context.getString(R.string.mensaje_peligro));
                        res_dist.setText(rdistance + "m");
                        // Notificación del límite superado
                        notifi.notificar_limite();
                        // Envío de aviso a los contactos
                        notifi.enviar_sms();
                        notifi.setSms_enviado(1);


                        // Abre la activity, si esta cerrada, con los resultados
                        if(mActivity.hasWindowFocus() == false) {
                            Intent intento = new Intent(mContext, Inicio.class);
                            intento.setFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
                            mContext.startActivity(intento);
```

```
                    }

                    // Iniciamos el servicio con la grabación de vídeo
                    if(isMyServiceRunning(BackgroundVideoRecorder.class) == false)
                    {
                        mContext.startService(new Intent(mContext,
                        BackgroundVideoRecorder.class));
                    }

                    TextView grabando = (TextView)
                    mActivity.findViewById(R.id.grabando);

                    grabando.setCompoundDrawablesWithIntrinsicBounds(R.drawable.g
                    rabando, 0, 0, 0);
                    grabando.setVisibility(View.VISIBLE);
                }

                // Si lo encuentra pero no la supera, se le dice
                else {
                    rssi_msg.setText(context.getString(R.string.mensaje_aviso));
                    res_dist.setText(rdistance + "m");
                    // Notificación de que se encuentra por los alrededores
                    notifi.notificar_radio();
                }

                // Finalizamos la búsqueda si lo encontramos
                BTAdapter.cancelDiscovery();
                mContext.stopService(new Intent(mContext, BService.class));

                mActivity.invalidateOptionsMenu(); // Refrescamos el menú
            }

        }
    }
};
```

A.3. Tablas de la BDD de la aplicación móvil

```
  private static final String NOMBRE_BASEDATOS = "protectULL.db";

  private static final String TABLA_CONTACTOS = "CREATE TABLE contactos (telefono TEXT
PRIMARY KEY, nombre TEXT, activo INTEGER)";

  private static final String TABLA_INFO_USUARIO = "CREATE TABLE tabla_info_usuario
(telefono TEXT PRIMARY KEY, nombre TEXT)";

  private static final String TABLA_CONFIG_APP = "CREATE TABLE tabla_config_app (id TEXT
PRIMARY KEY, password TEXT)";

  private static final int VERSION_BASEDATOS = 5;


  // Constructor de la clase
  public DBase(Context context) {
      super(context, NOMBRE_BASEDATOS, null, VERSION_BASEDATOS);
  }


  @Override
  public void onCreate(SQLiteDatabase db) {
      db.execSQL(TABLA_CONTACTOS);
      db.execSQL(TABLA_INFO_USUARIO);
      db.execSQL(TABLA_CONFIG_APP);
  }


  @Override
  public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
      db.execSQL("DROP TABLE IF EXISTS " + TABLA_CONTACTOS);
      db.execSQL("DROP TABLE IF EXISTS " + TABLA_INFO_USUARIO);
      db.execSQL("DROP TABLE IF EXISTS " + TABLA_CONFIG_APP);
      onCreate(db);
  }
```

A.4. Servicio para la búsqueda usando Bluetooth

```
public class BService extends Service {

    private Inicio inicio;

    // Constructor por defecto
    public BService() {
        inicio = new Inicio();
    }

    // Se ejecuta nada más ejecutarse el Servicio
    @Override
    public void onCreate() {}

    // Comienza el Servicio
    @Override
    public int onStartCommand(Intent intenc, int flags, int idArranque) {

        if(inicio.getBluetoothConnection().getBTAdapter().isEnabled()){
            // Si esta buscando, para la busqueda
            inicio.getBluetoothConnection().estaBuscando();
            // Inicia la busqueda
            inicio.getBluetoothConnection().buscar();
        }

        return START_STICKY;
    }

    // Destructor del Servicio
    @Override
    public void onDestroy() {}

    // Método encargado de enlazar con la Activity
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```

A.5. Controlador web *putonline*

```
/* Pone al dispositivo como online e incluye la conexión en el historial. */
exports.putonline = function (request, response) {

    if (Utilities.isEmpty(request.params.name)) return response.send(error_400);

    if (Utilities.isEmpty(request.body.name)) return response.send(error_400);
    if (Utilities.isEmpty(request.body.server)) return response.send(error_400);
    if (Utilities.isEmpty(request.body.time_now)) return response.send(error_400_);

    Camara.find({name: request.params.name}).exec(function (err, camara) {

        if (err) response.send(error_400);
        if (Utilities.isEmpty(camara)) return response.send(error_400);

        camara[0].online = true;
        camara[0].time_online = request.body.time_now;
        camara[0].number = request.body.numero;
        camara[0].nombre = request.body.nombre;
        camara[0].save();

        Historial.find({name: request.body.name}).exec(function (err, historiales) {
            if (err) return response.send(error);

            var server = "rtmp://" + request.body.server + request.body.name;
            var historial_nuevo = new Historial({ name: request.body.name, nombre:
             request.body.nombre,       numero:       request.body.numero,       time:
             request.body.time_now });
            historial_nuevo.save();

            response.send(ok);
        });
    });
};
```

A.6. Modelos de la aplicación web

```
// Camara Class
var mongoose = require('mongoose'),
    Schema = mongoose.Schema;

var camaraSchema = new Schema({
    server: String,
    name:  String,
    time_online: String,
    time_offline: String,
    number: String,
    nombre: String,
    online: { type: Boolean, default: false }
});


//Export the schema
module.exports = mongoose.model('Camara', camaraSchema);

// History Class
var mongoose = require('mongoose'),
    Schema = mongoose.Schema;

var historialDataSchema = new Schema({
    name:  String,
    nombre: String,
    numero: String,
    time: String
});


//Export the schema
module.exports = mongoose.model('DatosHistorial', historialDataSchema);
```

Apéndice B.
Conference Paper

A continuación, se encuentra el paper presentado para 10th International Conference on Ubiquitous Computing and Ambient Intelligence UCAmI 2016.

Secure Mobile Application to Combat Gender Violence

J.A. Concepción Sánchez[1], P. Caballero Gil[2], J. Molina Gil[3]

[1] Departament of Computer Engineering, University of La Laguna
La Laguna. Tenerife. Spain
alu0100697414@ull.edu.es

[2] Departament of Computer Engineering, University of La Laguna
La Laguna. Tenerife. Spain
pcaballe@ull.es

[3] Departament of Computer Engineering, University of La Laguna
La Laguna. Tenerife. Spain
jmmolina@ull.es

Abstract. This work describes the idea and operation of a mobile application to combat gender violence. Among the main used technologies are Android, BLE and LTE. If an offender gets close to a victim, even if the threshold distance has not been broken, the victim is warned thanks to the application. Besides, if the threshold distance is broken, an automatic streaming of a recording in real time is sent to the police while a list of stored contacts are warned so that they can try to help to protect the victim till the police arrives.

1 Introduction

Currently, one of the most terrible problems in the world is, unfortunately, gender violence. For example, in Spain, in the last 10 years the annual average of fatalities has been 62, and the annual average rate of victims of gender violence was 1.3 per 1,000 women aged 14 and older. In addition, there are many other cases that did not end in death but they were also seriously injured. In particular, the annual average number of victims of domestic violence was 7,000 [1] [2].

Recently, a system was developed based on GPS monitoring devices consisting of ankle bracelets to be attached to offenders [3], so that they allow notifying a control center if the threshold distance between offender and victim is violated. Such a system is implemented only in cases of high or extreme risk, which causes the vast majority of victims are not protected.

The aforementioned means that there is a serious problem in our society for which current solutions are not being effective.

Thus, this work aims not only to prevent unwanted situations between victim and aggressor, but also to transmit security to the victim in everyday life, so that she can go outside without being worried that the aggressor appears, because the system is

continuously checking the distance between them and if it is too short, a warning is sent both to the victim and to the emergency services with time to react.

The next sections explain in more detail the operation procedure of the proposed system, and mention the technologies that are used. Finally, a brief conclusion and some open problems close this paper.

2 Operation

The mobile application that is being developed is an app for the Android operating system (with possibility of portability to iOS and Windows Phone in the future), which uses BLE (Bluetooth Low Energy) in the background to check whether the device attached to the aggressor (typically an ankle bracelet) is detected nearby.

Thus, if the application detects the aggressor, it can use two different levels of warning using Android notifications together with vibration and sound. These levels are:

- Low level (caution): The aggressor has been detected in the vicinity, but does not exceed the signal strength required to alert the police. In this case, the victim may be aware in order to try to avoid unwanted situation of a meeting. See Fig. 1.
- High level (hazard): The aggressor is too close to the victim because he has exceeded the threshold distance according to the signal strength of the devices. See Fig. 2.

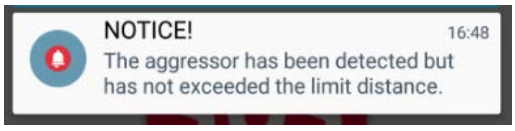


Figure 1. Warning of attacker detection

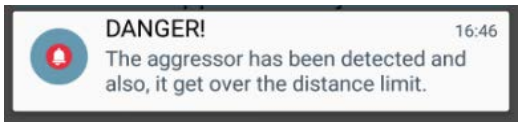


Figure 2. Warning of attacker approach

In the high level of hazard warning, besides notifying the victim, the application starts a video recording that is sent in real time using 4G mobile technologies to receiver station of the police.

In addition to the basic functionality of the application discussed above, the application has other elements of great interest:

- Panic Button: On the main page of the application there is a panic button, which is large enough so that the victim can instinctively press it and a call is automatically made to the emergency service. This is a detail of great interest because under high stress, it is difficult to dial a number or to select a contact.

- Contact List: The victim can configure a list of numbers to be notified by SMS if the attacker exceeds the second warning level, so that, for example, a neighbor can come to help while the police arrive.
- Recording and Streaming: The victim may choose to record and send a video to a control center by pressing a programmed button.
- Privacy of the application: The application icon is transparent for the victim can to have privacy. Neither the aggressor nor anyone can see it is installed. In addition, when application is initialized, it is necessary to put a password for to be able to access to the app.
- Warning of Disabled Bluetooth: The application warns the victim if the Bluetooth device on her device is off because this can mean the risk that the aggressor is close and the application does not warn about that. This warning cannot be removed from the Android notification until Bluetooth is activated. See Fig. 3

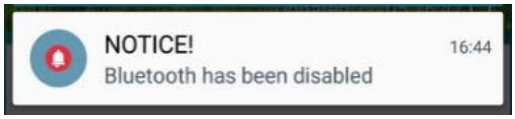


Figure 3. Warning of disabled Bluetooth

Fig. 4 and Fig 5 show two screenshots of the application in different situations. Fig. 6 shows a list of contacts to warn in dangerous situations.

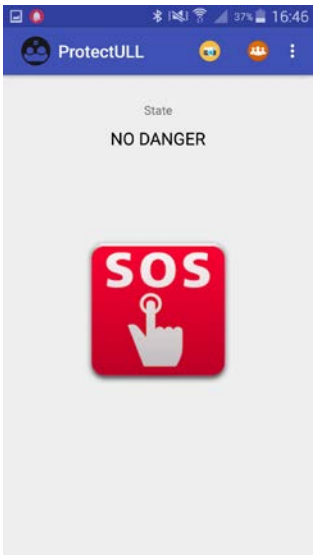


Figure 4. Screenshot in caution level

Figure 5. Screenshot in hazard level

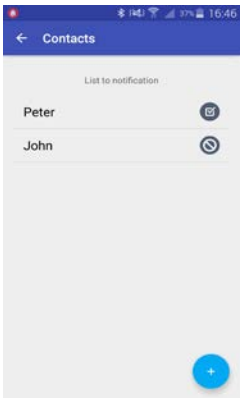


Figure 6: List of contacts for warning

In addition, we have also developed the party responsible for receiving videos sent from the mobile application with the information of the victim. To do this, the web application has made with technologies such as NodeJS and Angular.

The main functionality of the web, as we said, is to receive videos sent via streaming from mobile devices with the application, but also has a database that allows us to store certain information:

- A history of all connections have been made on the platform along with contact information for the victim.
- All devices that have made use of the web application.

The purpose of this website is to allow control of all detections between the victim and the aggressor with mobile applications, letting to see the current situation of the victim and act properly, can contact her if necessary by the information provided. See Fig. 7

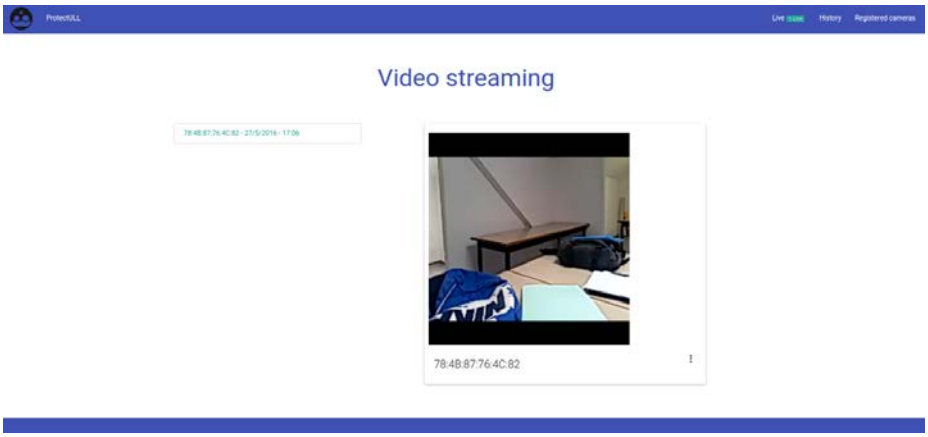


Figure 7: Web application

3 Used technologies

The main technologies used in this work are as follows:

- Android mobile operating system based on Linux, along with middleware applications focused to be used on mobile devices.
- Bluetooth Low Energy: Also known as BLE or Bluetooth LE, it is a new digital technology developed for interoperable Bluetooth radio devices. The main difference from previous Bluetooth versions is that being LE allows working with very low power consumption, solving the problem of their predecessors. The maximum range for a BLE device class 1 is one hundred meters.
- LTE (Long Term Evolution): Also known as 4G, it is a wireless communication standard for transmitting high-speed data for mobile phones and data terminals. It offers much higher speeds than their predecessors and it is also safer than them.
- NodeJS, Angular, HTML5 and CSS3 for deploy the Web application.
- Wowza server: Responsible for receiving streaming video sent from the mobile applications and place where the Web application connect for the videos.

Also, as the application handles sensitive data, especially when sending video recording, different cryptographic schemes have to be used to protect authenticity, confidentiality and integrity. In order to achieve it, the system has been enriched with an OpenSSL implementation including Elliptic Curve Diffie Hellman for secret key agreement, 256-bit AES encryption in CBC mode and Elliptic Curve Digital Signature Algorithm for signing/verifying.

Finally, regarding the communications scheme (see Fig. 8) the mobile applications of the victims are connected to the Wowza server via Wi-Fi or LTE for sending video streaming and then, this is reproduced from the Web application.

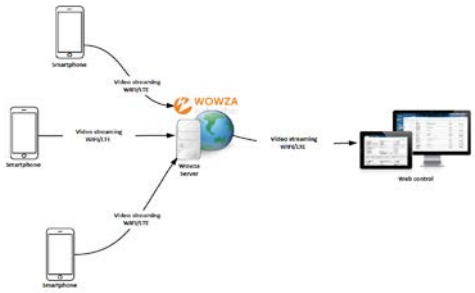


Figure 8: Scheme communications

5 Conclusions

This work describes the implementation of a novel warning system to protect the lives of all those people who are suffering gender violence today. With this proposal, the victim will feel safer because the developed application guarantees that if the offender gets close, even if the threshold distance has not been broken, the victim is warned so that protection measures can be taken. Besides, if the threshold distance is broken, an automatic streaming of a recording in real time is sent to the police while a list of stored contacts are warned so that they can try to help to protect the victim as the police arrives.

Since this is part of a work in progress, further improvements for the application in the form of additional features are being developed. Also, another future work is to make the application available for iOS and Windows Phone operating systems, so that in that way it will be accessible for any mobile device.

5 Acknowledgments

Research supported by projects RTC-2014-1648-8 and TEC2004-54110-R.

References

[1] Estadística de Violencia Doméstica y Violencia de Género. 2015. http://www.ine.es/prensa/np906.pdf

[2] Ellsberg, M., Arango, D. J., Morton, M., Gennari, F., Kiplesund, S., Contreras, M., Watts, C. 2015. *Prevention of violence against women and girls: what does the evidence say?* The Lancet, 385(9977), 1555-1566.

[3] Whitfield, D. 2001. *The Magic Bracelet: technology and offender supervision*. Waterside Press.