Universidad Autónoma de Madrid
Escuela Politécnica Superior

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Desarrollo de un sistema MDM para monitorización de dispositivos móviles Android

José María Pérez Rodríguez-Cadarso

Julio 2015

Desarrollo de un sistema MDM para monitorización de dispositivos móviles Android

Autor: **José María Pérez Rodríguez-Cadarso**
Tutor: **David Arroyo Guardeño**

Universidad Autónoma de Madrid
Escuela Politécnica Superior
Julio 2015

Agradecimientos

Me gustaría expresar mi agradecimiento a todas aquellas personas que con su ayuda han colaborado y ayudado en la realización del presente trabajo.

Especial agradecimiento a mi tutor, David. En primer lugar, por sugerirme un tema interesante y de carácter actual, y en segundo lugar, por su apoyo, ayuda e implicación para que llegase a buen puerto.

A todos mis compañeros de la carrera, en especial a Cristina, Julián y Guido. Porque han sido cuatro años juntos y seguro que sin alguno de ellos no habría sido lo mismo.

Gracias también a mi familia, por apoyarme en este camino que empezó hace cuatro años y que termina ahora.

A todas las demás personas, que de una manera u otra han formado parte de mi vida a lo largo de esta carrera, gracias.

Resumen

Vivimos en una época de gran avance tecnológico y debido a esto las empresas, en un intento de modernizar sus protocolos e infraestructuras, han incorporado todo tipo de nuevas tecnologías y políticas para poder mantener un nivel elevado de competitividad respecto al resto de las empresas. Estas incorporaciones se realizan muchas veces de manera precipitada y sin haber realizado un estudio y análisis sobre cómo incorporarlas de manera adecuada, lo que provoca numerosos errores tanto de integridad como de seguridad, que pueden ser fácilmente aprovechables por un atacante experimentado.

La ciberseguridad es sin duda una de las principales preocupaciones actualmente a nivel mundial, especialmente a nivel empresarial. Las motivaciones de los atacantes ya no buscan simplemente el hecho delictivo o la mera diversión, sino que cada vez son más los que son usados con el fin de obtener información confidencial o vulnerar la seguridad de las empresas para provocar daños irreparables en su infraestructura. Así mismo, también ha crecido el tráfico de datos personales de los ciudadanos, por lo que los ataques a personas de a pie son muy comunes y, generalmente, más sencillos de realizar.

Debido precisamente a que los ataques no se centran solo en empresas y grandes monopolios, sino que los ataques a ciudadanos son los más comunes, es necesario mantener un control dentro del entorno de trabajo de los dispositivos utilizados por ellos. Más aun, teniendo en cuenta que con la emergencia de los dispositivos móviles, todo el mundo dispone de algún dispositivo atacable, lo cual no solo afectaría al usuario, sino que también podría afectar a la empresa en la que se encuentre ese dispositivo, ya que podría tener acceso a información relevante de la empresa.

La utilización de dispositivo móviles en entornos empresariales por parte de los empleados se debe en gran medida a que han surgido nuevos dilemas como Bring Your Own Device (BYOD), que precisamente plantea la problemática de la tendencia actual de las empresas a permitir a los empleados utilizar sus dispositivos móviles de carácter personal en el ámbito labo-

ral, y debido a esto se incrementa la peligrosidad del dispositivo respecto a la empresa y por ende, entraña peligrosidad para los niveles de seguridad y protección de documentos confidenciales.

Todos estos factores han provocado una gran preocupación a nivel empresarial, debido a la cantidad de información confidencial que se gestiona con dispositivos móviles. Por esta razón, han surgido sistemas de gestión de dispositivos conocidos como MDM (Mobile Device Management), que pueden ser utilizados para mantener un control de la actividad de los dispositivos dentro del sistema de la empresa.

Con el fin de poder solucionar todos los problemas relaciones con este tipo de dilemas modernos, en el presente documento se pretende mostrar algunas de las herramientas existentes de las que dispone un administrador de cara a proteger los activos de una empresa en un contexto BYOD, así como un pequeño prototipo de sistema MDM que podría servir para monitorizar los dispositivos Android dentro de una empresa y para realizar un análisis forense de los *smartphones*.

Palabras clave: MDM, Android, dispositivos móviles, *smartphone*, BYOD, ciberseguridad.

Abstract

We live in a time of great technological progress that forces enterprises to adopt all sorts of new technologies and procedures to improve their underlying protocols and infrastructures.. These changes are usually carried out without a previous and thorough analysis of the advantages and risks of the selected technologies and solutions. As a consequence, the resulting systems contains serious integrity and security problems that can be eventually exploited by experienced attackers.

Cyber security is undoubtedly one of the main current concerns, globally and especially at enterprise level. The motivations of the attackers are no longer simply on the crime or the mere fun. Certainly, nowadays cyber attacks are more intended to obtain confidential information or to compromise the security of firms to further cause irreparable damage to its infrastructure. Likewise, it has also increased the traffic of personal data of citizens, so the attacks on people are very common and easier the day to perform.

Taking into account that the attacks are not focused only on large companies, along with the fact that the plain citizens are a relevant target , it is necessary to control and verify the devices used by the users in a given working environment. Furthermore, considering that with the emergence of mobile devices, everyone has an attackable device that, in case of being affected by any form of cyber attack, implies a risk, does affect not only to the user, but that could also affect the company in which the device is located.

The use of mobile devices in enterprise environments by employees is a result of the emergence of as the Bring Your Own Device (BYOD), which precisely raises the issue of the current trend in companies to allowing employees to use their personal mobile devices in the workplace. This encloses a risk for the company, since the device can be used to access and manage critical resources and sensitive information.

All these factors have caused a great concern in the enterprise environment, because of the amount of sensitive information that is handled by mobile devices. As a solution to this problem, different systems have been proposed to monitor and control mobile devices in corporate infrastructures. These solutions are coined as MDM (Mobile Device Management), which can be used to track the activity of the devices within the enterprise network.

In order to solve all the problems related to this type of modern dilemmas, this work show some of the existing tools available to protect information assets on a BYOD context. After performing an exhaustive study of such tools, the rest of the work is oriented to design and implement a MDM solution to verify the fulfillment of security privacy in a BYOD environment containing Android smart phones .

Keywords: MDM, Android, devices, smart phone, BYOD, cyber security.

Índice General

Agradecimientos I

Resumen III

Abstract V

Índice General VII

Índice de Figuras XI

Índice de Tablas XIII

1. Introducción 1

2. Estado del arte 7

2.1 BYOD 7

2.2 CYOD 9

2.3 Otras tendencias 10

2.4 MDM 10

2.4.1. Modo de funcionamiento 12

2.4.2. Posibles amenazas 12

2.4.3. MDM comerciales 13

2.4.4. MDM open source 15

2.5 Motivación 16

3. Análisis 17

3.1 Definición del proyecto 17

3.1.1. Objetivos y funcionalidad 17

3.1.2. Alcance 18

3.2 Catálogo de Requisitos 19

3.2.1. Requisitos funcionales 19

3.2.2. Requisitos no funcionales 22

4. Diseño 25
4.1 Diseño de la política de seguridad 25
4.1.1. Dispositivos sin rootear 26
4.1.2. VPN 26
4.1.3. Firewall 27
4.1.4. Contraseñas 28
4.1.5. Antivirus 29
4.2 Diseño del sistema MDM 29
4.2.1. Arquitectura del sistema 29
4.2.2. Entorno tecnológico 30
4.2.3. Descripción de los flujos de operación 30

5. Implementación del sistema MDM 33
5.1 Equipo de desarrollo 33
5.1.1. Hardware 33
5.1.2. Software 33
5.2 Plataformas 34
Xampp 34
Eclipse 34
OpenVPN 35
AVG 35
ConnectBot 35
Mobiwol 35
Adb 36
GCM 36
Java 37
PHP 37

OpenSSL....37
Keytool....37
5.3 Estructura y documentación del código....38
5.3.1. Cliente MDM....38
5.3.2. Servidor MDM....41

6. Plan de pruebas....45
6.1 Pruebas unitarias....45
6.1.1. Agente MDM....46
6.1.2. Sistema MDM....47
6.2 Pruebas de integración....48
6.3 Comprobación de rendimiento....49

7. Conclusiones y líneas futuras....51

Anexo A: Diagrama de Gantt....55
Anexo B: Base de datos....57
Anexo C: Vista de la aplicación....61

Bibliografía....65

Glosario....69

Índice de Figuras

Figura 1. Malware más común en Android. 1
Figura 2. Comparativa BYOD-CYOD 9
Figura 3. Arquitectura MDM. 11
Figura 4. Esquema VPN 26
Figura 5. Flujos de operación cliente MDM *30*
Figura 6. Flujo entre el cliente y el servidor 31
Figura 7. Diagrama de Gantt 55
Figura 8. Escala de tiempos 56
Figura 9. Diagrama Entidad-Relación de la BD 57
Figura 10. Pantalla de login inicial 61
Figura 11. Registro completo 61
Figura 12. Pantalla de registro 61
Figura 13. Pantalla cambio contraseña código 62
Figura 14. Pantalla cambio contraseña 62
Figura 15. Pantalla de login completa 62
Figura 16. Pantalla confirmación código 63
Figura 17. Pantalla confirmación 63
Figura 18. Notificaciones tras confirmación 63

Índice de Tablas

Tabla 1. Amenazas a los sistemas MDM 13
Tabla 2.Pruebas unitarias del agente MDM 47
Tabla 3. Pruebas unitarias del sistema MDM 48
Tabla 4. Pruebas de integración 49
Tabla 5. Tabla apks de la BD 57
Tabla 6. Tabla browser de la BD 58
Tabla 7. Tabla gps de la BD 58
Tabla 8. Tabla logcat de la BD 58
Tabla 9. Tabla pdf de la BD 58
Tabla 10. Tabla registered de la BD 59
Tabla 11. Tabla users de la BD 59
Tabla 12. Tabla vpnState de la BD 59

Capítulo 1

Introducción

El desarrollo de los dispositivos y comunicaciones móviles, así como de las tecnologías inalámbricas ha experimentado un gran crecimiento en los últimos años y con ello ha revolucionado la forma de trabajar y comunicarse, no sólo en el ámbito social sino que también en el laboral. El uso creciente de estas tecnologías ha situado a los dispositivos móviles como uno de los principales objetivos de los ciberataques por la valiosa información que puede contener el dispositivo en función de los documentos descargados o de las redes a las que ha estado conectado.

El campo del desarrollo de aplicaciones para los dispositivos móviles es uno de los de mayor crecimiento en los últimos años y más concretamente el destinado a dispositivos con sistema operativo Android, debido sobre todo a su mayor cobertura de mercado actualmente [25]. Dicho desarrollo ha ido acompañado, al igual que en otros casos anteriores de tecnologías en auge como el desarrollo de aplicaciones web, de unas prácticas de programación y unos diseños poco adecuados, que junto con una tecnología por pulir, es el blanco perfecto para los atacantes (ver Figura 1).

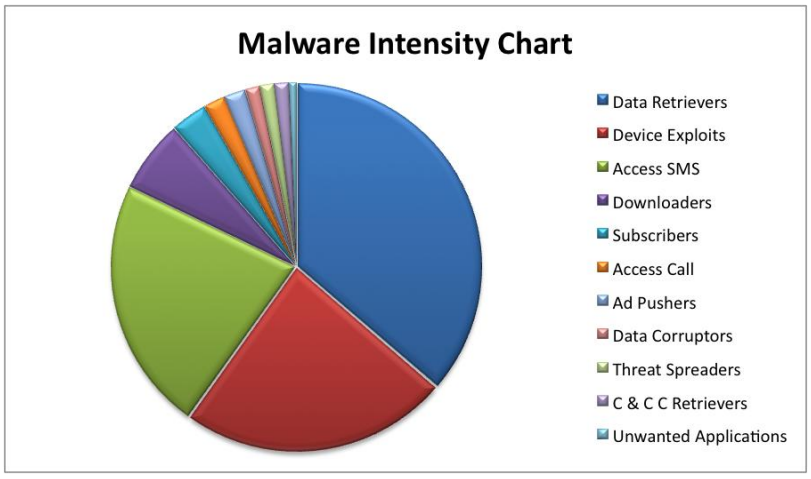

Figura 1. Malware más común en Android.

Recuperado de http://www.droidslk.com/2014/05/do-android-need-antivirus/

En un sistema operativo clásico, las aplicaciones son ejecutadas con distintos niveles de seguridad, de manera que los procesos más importantes y vitales del sistema corren siempre en un entorno seguro y muy poco accesible, mientras que la mayoría de los procesos de usuarios, es decir, el software usado a diario por el usuario, es ejecutado en un entorno en donde están mucho más expuestos a vulnerabilidades. Este tipo de arquitectura es necesaria ya que de otra manera, almacenar un simple documento de texto en el disco sería una tarea muy tediosa en cuanto a los permisos se refiere.

Es precisamente en las capas de software más expuestas en las que los desarrolladores de virus y malware suelen enfocarse, debido a que son mucho más vulnerables. De esta forma, los virus pueden comenzar desde las capas más externas y extenderse para posteriormente atacar las capas del sistema más seguras, donde residen los procesos vitales del sistema. Este tipo de ataque es conocido como **escalada de privilegios** [29].

Mientras que un sistema operativo corriente maneja permisos a nivel de usuario, en Android se manejan permisos a nivel de aplicación. Por tanto, un ataque de escala de privilegios en un dispositivo Android consistiría en obtener permisos de root o conseguir acceder a mas permisos de aplicación mediante el acceso a los permisos de otras aplicaciones [27]. Todos estos permisos son los que se manejan en el manifiesto de las aplicaciones.

El sistema operativo de Android, aunque es básicamente un núcleo Linux, es sustancialmente distinto en cuanto a la forma en la que funciona debido a sus capas. Android ha sido diseñado con **dos capas de seguridad** [19]: la primera de ellas totalmente restringida al usuario donde se ejecutan y almacenan todos los procesos y archivos vitales para el sistema; y la segunda, donde reside la información personal del usuario y las aplicaciones y configuraciones que ha descargado e instalado. De este modo, cuando un *smartphone* es infectado con virus o *malware*, las acciones de estos sólo tienen efecto sobre los datos y configuraciones del usuario. Sin embargo, si una aplicación es un malware puede que no tenga determinados privilegios necesarios para acceder a información que desea obtener, pero puesto que en Android existen los permisos en las aplicaciones, puede que esta tenga

acceso mediante permisos a otras aplicaciones y desde estas otras aplicaciones si tenga acceso a la información que busca.

Por este motivo, la utilización de técnicas como el *rooteo* en dispositivos Android siempre es una tarea que entraña un gran riesgo [30], ya que deja una puerta abierta a la entrada de virus a los niveles más seguros del sistema debido a que el usuario pasa a ser *root* en todo el sistema. Teniendo en cuenta esta consideración, las herramientas y el sistema que se han implementado en este proyecto utilizarán técnicas que no requieran *rootear* el dispositivo.

La proliferación de dispositivos móviles en los últimos años, junto al aumento de las capacidades, prestaciones y posibilidades de utilización de los mismos, hace necesario evaluar en profundidad la seguridad ofrecida por este tipo de dispositivos, así como de los mecanismos de protección de la información que gestionan, dentro de los entornos de Tecnologías de la Información y las Comunicaciones (TIC).

La utilización y amplia adopción, sin precedentes, de los dispositivos móviles como herramientas básicas de productividad en el ámbito profesional, junto a su utilización simultánea en el ámbito personal, hacen necesario que las organizaciones realicen una gestión minuciosa, exhaustiva y continua de los mismos, acorde con las políticas de seguridad de la organización.

Esta estrategia, conocida como **movilidad empresarial**, permite a los usuarios y empleados de las organizaciones llevar a cabo sus actividades diarias de negocio a través de dispositivos móviles que aprovechan las tecnologías que facilitan el acceso remoto a los datos corporativos, incrementando su eficiencia y productividad con independencia de la ubicación física en la que se encuentran y con mayor flexibilidad en los desplazamientos.

Las soluciones tecnológicas que permiten la gestión de los dispositivos móviles a nivel empresarial se conocen como MDM. Estas soluciones permite gestionar de forma eficiente la diversidad y el despliegue masivo, dinámico y a gran escala de dispositivos móviles en una organización, con un enfoque principalmente orientado a incrementar su seguridad mediante

la monitorización personalizada de cada dispositivo, y mejorando colateralmente la productividad del usuario final.

Los **objetivos** de este documento son analizar las principales estrategias o dilemas actuales que implican la utilización de dispositivos en el ámbito empresarial y los inconvenientes que plantean para la seguridad de la empresa, así como las soluciones que se plantean para resolver estos problemas. En este orden, se ha diseñado una aplicación cliente/servidor para la monitorización de los dispositivos móviles. Dicha aplicación es una herramienta open source que sirve para constatar que se cumple una determinada política de seguridad y que además, permitiría realizar análisis forense de los dispositivos registrados en el sistema, todo ello sin necesidad de tener permisos de root en los dispositivos.

La seguridad se define como el cumplimiento unos ciertos requisitos de confidencialidad, integridad y disponibilidad, pero también se necesita cumplir unos ciertos criterios de autenticación, autorización (procesos/usuarios) y auditabilidad [26]. Por tanto, el sistema MDM que se ha diseñado contribuye a la seguridad corporativa en base a este criterio.

El cumplimiento de estos objetivos ha dado origen al presente documento, cuya estructura se expone a continuación.

En primer lugar, en el Capitulo 2, se realizará un análisis de la situación actual de las políticas de inclusión de dispositivos móviles en entornos empresariales y se realizará un análisis de los sistemas MDM actuales.

En el Capitulo 3, se expondrán los objetivos del sistema y su alcance, así como los requisitos funcionales y no funcionales, tanto del cliente MDM como del servidor.

En el Capitulo 4, se expondrá el diseño de la política de seguridad y posteriormente el del sistema MDM.

En el Capitulo 5, se explicará la implementación realizada, mostrando las herramientas estudiadas y utilizadas, y documentando la funcionalidad de cada elemento del sistema MDM.

En el Capitulo 6, se mostrarán los resultados de las pruebas realizadas, ya sea en el cliente como en el servidor, tanto de las pruebas unitarias como las de integración.

Por último, en el Capitulo 7, se expondrán las conclusiones y los puntos a abordar en un trabajo futuro a la hora de mejorar el sistema MDM de cara a poder ser utilizada para monitorizar una red real y servir como una herramienta de análisis forense más avanzada.

Capítulo 2

Estado del arte

Como se ha explicado anteriormente, el campo de los dispositivos móviles ha experimentado un amplio crecimiento en los últimos años y ello ha supuesto una mayor utilización y dependencia por parte de los usuarios.

En el ámbito empresarial, han supuesto también un avance en cuanto al sistema de trabajo y al rendimiento que se puede obtener de los empleados. Debido a que las redes de comunicaciones y las tecnologías de los dispositivos avanzan a una velocidad vertiginosa, hoy en día los empleados pueden trabajar prácticamente desde un dispositivo móvil o al menos realizar muchas de las labores que antes, si no se encontraban en su puesto de trabajo, no podían realizar como descargarse documentos importantes de la empresa, acceder a correos de carácter urgente o consultar horas de reuniones de trabajo [31].

Teniendo en cuenta la tendencia a la utilización de este tipo de tecnología, han surgido varios tipos de políticas que intentan incorporar estos dispositivos atendiendo a distintas características y necesidades corporativas. Dichas políticas se exponen a continuación.

2.1 BYOD

La política Bring Your Own Device, también llamada Bring Your Own Technology, permite a los empleados de una empresa a llevar y utilizar sus dispositivos móviles personales en su entorno de trabajo y a acceder a información y documentación privada de la empresa desde los mismos.

BYOD ha emergido con gran fuerza en el entorno empresarial debido principalmente al hecho de que las empresas son incapaces de evitar que los empleados lleven sus dispositivos y los conecten a la red. Debido a esto, las empresas han adoptado políticas BYOD [31], en primer lugar para poder controlar la utilización de estos dispositivos dentro de la empresa y así

poder gestionar los problemas de seguridad que conllevan y, en segundo lugar, para fomentar la productividad entre los empleados y reducir costes, debido a que los dispositivos móviles actuales son sistemas complejos y avanzados, su precio es elevado aunque variable dependiendo de las prestaciones y la plataforma, hecho que tiene influencia especialmente en las decisiones de la organización cuando es necesario adquirir un número elevado de estos [1].

Según estudios de diversas compañías, permitir la utilización de dispositivos personales en el entorno de trabajo incentiva la productividad, aumenta la moral y hace a la compañía parecer más flexible y atractiva de cara al empleado [3].

Sin embargo, no todo son puntos a favor, BYOD supone ciertos inconvenientes también [32]. Las empresas tienen que asegurarse que los dispositivos que acceden a información confidencial interna están protegidos de malware y para ello, necesitan monitorizar de cierta forma los dispositivos que están conectados a su red, lo cual se puede realizar a través de sistemas MDM, y si el volumen de dispositivos es muy elevado puede que la infraestructura de la empresa no esté preparada para soportar elevadas cantidad de conexiones. Además, tienen que desarrollar una política de seguridad que defina que información es confidencial, cuál debe ser protegida y qué empleados pueden tener acceso a ella.

La tendencia BYOD debe asumir que los dispositivos móviles personales no son de confianza desde el punto de vista de seguridad, salvo que hayan sido configurados con unos requisitos mínimos de seguridad y pasen a ser gestionados y monitorizados por la organización.

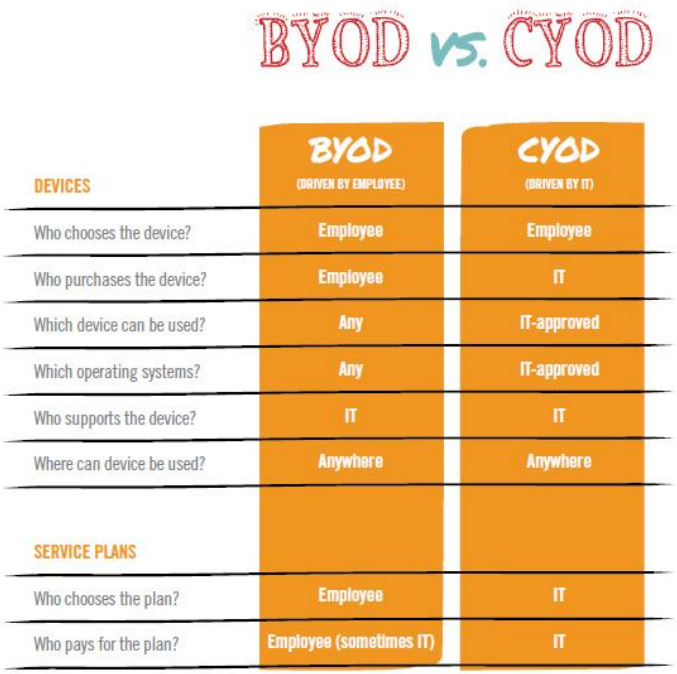

Figura 2. Comparativa BYOD-CYOD

Recuperada de
https://www.uk.insight.com/content/dam/insight/EMEA/uk/services/byod-v-cyod-table.jpg

2.2 CYOD

Una aproximación intermedia adoptada por algunas organizaciones y conocida como CYOD (Choose Your Own Device), permite al usuario elegir el dispositivo móvil que utilizará, pero en lugar de hacerlo de todos los disponibles en el mercado, deberá elegir entre una lista preseleccionada de dispositivos por parte de la empresa, que cumplen una serie de requisitos de seguridad, fiabilidad y durabilidad.

Mientras que la política BYOD propone una mayor flexibilidad y productividad, esta política ofrece una mayor seguridad a cambio de menor flexibilidad y mayor coste, ya que la empresa se hace cargo del coste de los dispositivos [18] (ver Figura 2).

2.3 Otras tendencias

Existen otras tendencias como la conocida como COPE (Corporataly-Owned, Personally-Enabled), donde el dispositivo es propiedad de la organización pero se permite su utilización para determinadas tareas de carácter personal.

Existen otros enfoques, como un modelo de uso reciente en la industria basado en la posibilidad de compartir un dispositivo móvil entre diferentes empleados, debiendo el sistema MDM proporcionar capacidades de gestión de multiusuario para segmentar y compartimentar el acceso a datos y servicios por parte de cada uno de los usuarios.

Debido al dinamismo de la industria móvil, recientemente ha surgido otra tendencia que está siendo adoptada por numerosas empresas, denominada BYOA (Bring Your Own App). Esto hace referencia a que los usuarios instalan aplicaciones de los mercados de aplicaciones públicos en sus dispositivos móviles personales o corporativos con el fin de aumentar su productividad, y manejando datos corporativos [7].

2.4 MDM

Para acomodar todas estas políticas, una tendencia actual es clasificar los dispositivos en dos grandes grupos, los gestionados y no gestionados. Los dispositivos no gestionados son aquellos sobre los que la organización no ha llevado a cabo ninguna tarea previa de configuración y por tanto no dispone de control sobre su configuración o el nivel de seguridad. Por otro lado, los dispositivos gestionados son aquellos que han seguido el proceso de registro en el sistema MDM de la organización y, por tanto, sobre los que se han aplicado la política de seguridad pertinente.

Se conoce por Mobile Device Management a aquel software que permite asegurar, monitorizar y administrar dispositivos móviles sin importar el operador de telefonía o proveedor de servicios. La mayoría de los MDM permiten hacer instalación de aplicaciones, localización y rastreo de equi-

pos, sincronización de archivos, reportes de datos y acceso a dispositivos, todo de manera remota.

Este tipo de software ha tenido una gran aceptación por parte de las empresas y su crecimiento ha sido realmente vertiginoso, debido en gran parte a la popularidad que han alcanzado los *smartphones* en las corporaciones.

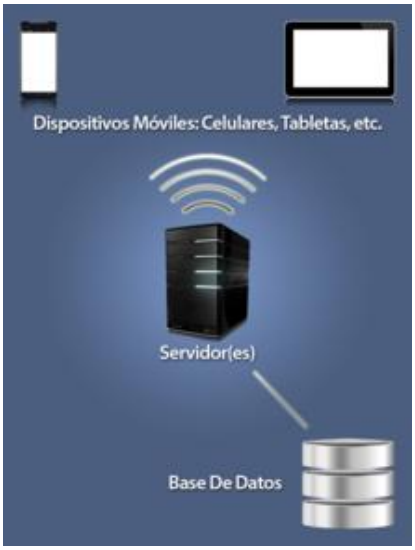

Figura 3. Arquitectura MDM.

Recuperado de https://es.wikipedia.org/wiki/Mobile_device_management

La arquitectura básica de un MDM consiste en un agente, el cual es una aplicación que se instala en cada uno de los dispositivos que se desean administrar, un servidor de implementación, desde donde se ejecuta el MDM, y una base de datos, donde se guardan todos los datos recabados. Dicho de otro modo, un MDM no es sino una solución cliente/servidor orientada a recabar información de dispositivos móviles finales. Dicha información es recogida mediante agentes instalados en cada dispositivo móvil, para lo cual mantienen una conexión con el servidor a través de USB, Wi-Fi, GPRS, 3G o cualquier otro medio de transmisión de datos lo cual le permite al MDM tomar control del dispositivo [3] (ver Figura 3).

2.4.1. Modo de funcionamiento

El modo de funcionamiento de los sistemas MDM suele ser similar para todos ellos y consta de los siguientes pasos [9]:

- Configuración: el dispositivo móvil del usuario es registrado en el sistema MDM y se le aplica la política de seguridad.
- Distribución: el cliente MDM es distribuido e instalado en los dispositivos móviles de los empleados. Esto se puede hacer mediante un repositorio interno de la empresa.
- Autenticación: cuando un usuario va a activar el agente MDM, se comprueban los datos enviándolos al servidor para autenticar la identidad del empleado.
- Instrucción: el servidor MDM envía al agente MDM la política de control del dispositivo y distintos comandos dependiendo del estado del dispositivo y los datos que haya enviado al servidor.
- Control/Notificación: el agente MDM controla las funciones del dispositivo atendiendo a la política del dispositivo y envía los resultados al servidor MDM.

2.4.2. Posibles amenazas

Para poder deducir los requerimientos de seguridad de un sistema es necesario conocer cuáles son los posibles problemas o debilidades de un sistema. Desde ese punto de vista, a continuación se expondrán posibles ataques a un sistema MDM [9]:

Tipo de Ataque	**Descripción**
Software	Un agente atacante puede modificar el sistema operativo o la aplicación del sistema MDM.
Desviación	Un agente atacante puede pasar las funciones de seguridad de un sistema MDM para incapacitar sus características de seguridad.
Datos	Un agente atacante puede cambiar los datos almacenados en el sistema MDM sin autorización.
Datos	Un agente atacante puede modificar los datos que se transmiten al sistema MDM.
Tráfico	Un agente atacante puede capturar y analizar el tráfico transmi-

	tido al sistema MDM y obtener información confidencial.
Spoof	Un agente atacante puede acceder al sistema MDM mediante fuerza bruta para conseguir autenticarse.
Malware	Un agente atacante puede infectar el sistema MDM para conseguir información o dañar el sistema.
DoS	Un agente atacante puede realizar un ataque de denegación de servicio para impedir el funcionamiento normal del sistema MDM.
Almacenamiento	Un agente atacante puede reducir la capacidad tanto del sistema MDM como de su entorno operativo de manera que eventos relacionados con la seguridad y el almacenamiento de datos no se almacene durante un periodo de tiempo.
Vulnerabilidad	Un agente atacante puede atacar un sistema MDM usando una vulnerabilidad desconocida en el sistema.

Tabla 1. Amenazas a los sistemas MDM

2.4.3. MDM comerciales

AirWatch

La administración de dispositivos móviles de AirWatch permite a las empresas enfrentar los retos asociados a la movilidad al proveer una manera simple y eficiente de ver y administrar todos los dispositivos desde una sola consola de administración central. La solución permite registrar rápidamente dispositivos en su entorno empresarial, configurarlos, actualizar los ajustes de los dispositivos y fortificarlos [23].

Sus principales características son:

- Consola de administración única.
- Sencillo proceso de registro.
- Perfiles que permiten definir configuraciones, políticas y restricciones empresariales para los dispositivos sin requerir la interacción de los usuarios.

SAP Afaria

SAP Afaria es un programa de gestión de movilidad empresarial que permite [6]:

- Proteger los datos corporativos importantes con una arquitectura segura y ampliable.
- Hace cumplir las políticas de seguridad en los dispositivos sea cual sea el sistema operativo.
- Bloquear o borrar dispositivos robados o perdidos.
- Elegir métodos de implementación flexibles y ampliables.
- Respaldar escenarios de dispositivos personales como corporativos.
- Escalar con las necesidad de tu empresa, ya sea por tamaño o por la utilización de tecnologías como la nube.

MobileIron

MobileIron es una empresa dedicada a la seguridad y gestión de los dispositivos móviles en entornos corporativos. Su sistema MDM proporciona las siguientes funcionalidades [13]:

- Soporte a diversos sistemas operativos, incluyendo iOS, Android y Windows Mobile.
- Establece una política de seguridad.
- Integración en la infraestructura de la empresa.
- Borrado selectivo.
- Protección de email.
- Configuración de Wi-Fi, email y VPN.
- Sistema avanzado de notificación.

2.4.4. MDM open source

ownMDM

OwnMDM es un gestor de dispositivos, que permite controlar remotamente los dispositivos registrados desde un servidor [10].

Es un software libre, y consta de dos módulos:

- Web: es un php que usa una base de datos MySQL para enviar y recibir los datos a los dispositivos registrados.
- App. La app es un administrador móvil que funciona haciendo las tareas de forma transparente.

Entre las funcionalidades que ofrece se encuentran:

- Mensajería: Envía una notificación con un mensaje al dispositivo.
- Bloqueo: Bloquea la pantalla del dispositivo.
- Activación: el dispositivo recibe un popup para activar la aplicación en el caso de que no esté corriendo.
- Ping: comprueba mediante un ping si el dispositivo responde.
- Localización: devuelve la localización del dispositivo.
- Borrado remoto: el dispositivo quedará borrado en caso de que se haya perdido.
- Grabación de audio: graba un audio de 20 segundos y lo envía al correo.
- Captura de imágenes: realiza una foto y la envía al correo.
- Seguimiento del dispositivo: el dispositivo enviará localización, sonido y fotos cada 3 minutos en un periodo de 15 minutos.
- Recepción/Envío: el dispositivo envía un archivo.

2.5 Motivación

Partiendo del conocimiento de los sistemas MDM anteriores y teniendo en cuenta que los de carácter comercial están muy desarrollados, se ha decidido desarrollar un **sistema MDM *open source***.

Los MDM de carácter *open source* existentes permiten acciones como enviar mensajes, localización del dispositivo, borrado remoto, etc. La intención de este proyecto es crear un sistema que no solo sirva para realizar las acciones mencionadas anteriormente, sino que también funcione como un sistema de análisis forense de los dispositivos.

Para ello, capturará información que se podrá usar posteriormente para detectar posibles aplicaciones que sean *malware* o incluso para poder realizar una imagen virtual del dispositivo como una opción de *backup* en caso de robo del dispositivo o de borrado del mismo.

A continuación, en el Capitulo 3, se procederá a realizar el análisis del prototipo de sistema MDM.

Capítulo 3

Análisis

En esta sección del documento se realizará un análisis del sistema implementado, en el que se detallan los objetivos y el alcance del mismo, así como los requisitos funcionales y no funcionales.

3.1 Definición del proyecto

En este primer apartado se abordan los objetivos principales del sistema, la funcionalidad y el alcance del mismo.

3.1.1. Objetivos y funcionalidad

El objetivo del presente documento es analizar las múltiples utilidades que tienen diversas herramientas existentes en el mercado y que permiten crear una configuración segura para un entorno de trabajo corporativo. A su vez se pretende mostrar una posible política de seguridad a aplicar, junto con un prototipo de sistema MDM para monitorizar los dispositivos y realizar una análisis forense.

El objetivo del sistema MDM no es en sí controlar todas las variables posibles ni monitorizar y obtener todos los datos, sino establecer un producto base con unas funcionalidades básicas, con el fin de mostrar el funcionamiento y al que posteriormente se le puedan incorporar nuevos módulos en un futuro desarrollo con el fin de mejorarlo.

La **funcionalidad** del prototipo consistirá en un sistema MDM que gestionará los dispositivos registrados por los usuarios, y capturará cierta información relevante de los dispositivos, así como comprobar ciertos parámetros de los dispositivos para comprobar que la política de seguridad se cumple.

Así pues, el prototipo se dividirá en **dos aplicaciones** independientes que interactúan la una con la otra y cuyo trabajo conjunto da sentido al proyecto. Por un lado está la aplicación para los dispositivos Android, con la que los usuarios se registran en el sistema MDM y, por otro lado, el siste-

ma MDM, donde se registran los datos capturados de los dispositivos de los empleados.

La aplicación Android permite a los usuarios registrarse, cambiar su contraseña y validarse en el sistema, para ello deberá utilizar su usuario y contraseña utilizados a nivel corporativo. Una vez validado, para evitar posibles suplantaciones de identidad, el dispositivo registrado recibirá una notificación con un código de acceso vía GCM (Google Cloud Messaging). Al introducir ese código de acceso el usuario estará aceptando la monitorización de su dispositivo e inmediatamente información será capturada y enviada al sistema MDM.

El sistema MDM por su parte, consistirá en un servidor que recibe los datos enviados por los distintos dispositivos y los almacenará en una base de datos para que el administrador de red tome las medidas oportunas. Así mismo, en caso de que algún dispositivo no esté usando conexión VPN (Virtual Private Network) o no tenga activado el GPS (Global Positioning System), como viene establecido en la política de seguridad, se encargará de enviarle una notificación para que el usuario los active.

3.1.2. Alcance

El prototipo creado pretende servir como base para la creación de un sistema MDM comerciable y que permita gestionar de forma adecuada las conexiones que se produzcan por parte de dispositivos móviles en un entorno empresarial. No pretende cubrir todas las características que cubren los MDM comerciales, sino establecer una base para la monitorización capturando unos datos iníciales y a partir de ellos enviar notificaciones a los usuarios en caso de algún riesgo para la seguridad de la corporación.

3.2 Catálogo de Requisitos

A continuación se describe el catálogo de requisitos del proyecto, tanto los funcionales como los no funcionales.

3.2.1. Requisitos funcionales

En esta sección, se muestran los requisitos funcionales de la aplicación cliente y, en segundo lugar, del sistema MDM del servidor.

Agente MDM

RF 1 El usuario podrá registrar su dispositivo introduciendo el usuario y la contraseña corporativa en uso. Si los datos son correctos, el dispositivo se registrará en el servidor y, en caso contrario, el cliente recibe un aviso de datos en el formulario incorrectos.

RF 2 Para validarse en la aplicación todos los usuarios deberán utilizar su usuario y contraseña ya registradas en el sistema corporativo. Si los datos introducidos son correctos el usuario accederá a la pantalla de doble autenticación, mientras que si son incorrectos recibirá un mensaje de error en el proceso de login.

RF 3 El usuario podrá modificar su contraseña introduciendo su usuario y contraseña, así como la nueva contraseña. Finalmente deberá introducir un código de confirmación.

RF 4 Una vez en la pantalla de doble autenticación, el dispositivo recibirá una notificación con un código de acceso. Al introducir el código se procederá a monitorizar el dispositivo. En caso de que se introduzca mal el código, se recibirá un mensaje de error y se permitirá introducir de nuevo el código.

RF 5 El dispositivo recibirá una notificación cuando el usuario esté siendo monitorizado y no tenga la conexión mediante VPN activada.

RF 6 El dispositivo recibirá una notificación cuando el usuario esté siendo monitorizado y no tenga activado la localización por GPS.

RF 7 El dispositivo recibirá una notificación cuando esté siendo monitorizado y no tenga instaladas las aplicaciones indicadas en la política de seguridad.

RF 8 El dispositivo recopilará información para la monitorización y la enviará al servidor.

- **RF 8.1** El dispositivo capturará la lista de apks instaladas en el dispositivo y lo enviará al servidor.
- **RF 8.2** El dispositivo capturará el historial de navegación del dispositivo y lo enviará al servidor.
- **RF 8.3** El dispositivo capturará el logcat y lo enviará al servidor.
- **RF 8.4** El dispositivo capturará la lista de pdf descargadas por el usuario y la enviará al servidor.
- **RF 8.5** El dispositivo capturará el estado de la conexión VPN y de la localización GPS y los enviará al servidor.

RF 9 El usuario podrá salir de la aplicación en cualquier momento y dejar de ser monitorizado.

Servidor MDM

RF 1 El servidor comprueba que los datos del registro concuerdan con el usuario y contraseña almacenado en la base de datos de los empleados de la empresa. En caso de validarse, se registra el identificador único del dispositivo y, en caso contrario, se envía al cliente un mensaje de error en el registro.

RF 2 El servidor comprueba que los datos del login concuerdan con los del usuario almacenado en la base de datos y comprueba que el dispositivo desde el que se pretende validar es el que aparece almacenado como perteneciente al usuario. Si la validación es correcta, se envía un mensaje de validación correcta y, en caso contrario, se envía al cliente un mensaje de error en el proceso de login.

RF 3 El servidor recibirá los datos del cambio de contraseña y generará un código de confirmación que enviará al dispositivo mediante una notificación de GCM en caso de que los datos sean correctos y, en caso contrario, se enviará un mensaje de error en el cambio de contraseña.

RF 4 El servidor al recibir solicitud de iniciación de confirmación genera un código aleatorio y lo envía mediante el sistema de mensajería GCM al cliente, al mismo tiempo que lo guarda en la base de datos mediante una función hash.

RF 5 El servidor, al recibir la petición de confirmación, comprueba que el hash del código recibido concuerda con el almacenado en la base de datos y envía un mensaje de confirmación completada. En caso de error en la validación, envía al cliente un mensaje de código incorrecto.

RF 6 Recibirá la transmisión de los distintos datos enviados por la aplicación en el cliente.

- **RF 6.1** Recibirá y guardará los datos con la lista de apks de los dispositivos en la base de datos.
- **RF 6.2** Recibirá y guardará los datos con el historial de navegación de los dispositivos en la base de datos.
- **RF 6.3** Recibirá y guardará los datos del logcat de los dispositivos en la base de datos.

RF 6.4 Recibirá y guardará el estado de la conexión VPN del dispositivo en la base de datos.

RF 6.5 Recibirá y guardará los datos de la localización mediante GPS en la base de datos.

RF 6.6 Recibirá y guardará los datos de la lista de pdfs descargados por el dispositivo en la base de datos.

3.2.2. Requisitos no funcionales

A continuación, se exponen ciertos requisitos que debe tener el sistema en cuanto a seguridad, eficiencia y usabilidad.

Seguridad

RNF 1 Las conexiones con el servidor se realizan mediante conexión segura utilizando SSL.

RNF 2 El identificador único del dispositivo se guarda en la base de datos del servidor aplicándole una función hash.

RNF 3 Las contraseñas de los usuarios se almacenan en la base de datos del servidor aplicándole una función hash previamente.

RNF 4 El código de acceso se genera de manera aleatoria y formando un mensaje alfanumérico de longitud 8, incluyendo minúsculas, mayúsculas y números. Se almacena en el servidor aplicándole una función hash.

RNF 5 El sistema comprobará que la conexión VPN está activada de manera cíclica cada un determinado tiempo especificado por el administrador de red.

RNF 6 El sistema capturará los datos de los dispositivos conectados cada un determinado tiempo especificado por el administrador de red.

RNF 7 El servidor MDM y sus componentes deben estar localizados en un entorno físico seguro y protegido por un sistema de seguridad.

RNF 8 Cualquier vulnerabilidad detectada debe ser corregida con el fin de asegurar la seguridad del sistema MDM.

RNF 9 El sistema deberá mantener un backup de los eventos que ocurren para poder identificar responsabilidades en caso de un fallo en la seguridad del sistema.

RNF 10 El sistema MDM debe mantener la información almacenada de manera segura para evitar el acceso de personas no autorizadas.

RNF 11 El sistema MDM deberá proveer un mecanismo para distribuir el agente MDM mediante el método y la ruta que elija la organización.

RNF 12 El sistema MDM deberá tener un mecanismo para eliminar software potencialmente peligroso de los dispositivos.

RNF 13 El sistema MDM deberá tener alguna forma de controlar que el agente MDM no sea borrado de algún dispositivo por algún software malicioso.

Eficiencia y rendimiento

RNF 14 El tiempo de registro del dispositivo no será mayor a 1 segundo.

RNF 15 El tiempo de acceso al sistema no será mayor a 1 segundo.

RNF 16 El tiempo de creación y envió del código de acceso no será mayor de 2 segundos.

RNF 17 El tiempo de recepción del código de acceso no será mayor de 4 segundos.

RNF 18 El tiempo de transmisión de los datos capturados no será mayor a 5 segundos por dispositivo.

RNF 19 El tiempo de transmisión del estado de la conexión VPN no será mayor a 1 segundo.

Usabilidad

RNF 20 Las interfaces de la aplicación para Android siguen el mismo patrón de estilo y diseño, de forma que se garantiza una mayor consistencia y comodidad al usuario.

RNF 21 La aplicación muestra diversos mensajes de error como de validación correcta a lo largo del proceso de registro, login y confirmación.

RNF 22 La aplicación móvil está diseñada para que se adapte a distintos tamaños y densidad de pantalla.

Mantenimiento

RNF 23 La aplicación móvil sigue una estructura modular, por lo que el código se encuentra dividido en distintos módulos según su funcionalidad, comenzando por *es.uam.eps*.

RNF 24 Con el fin de facilitar el mantenimiento y posibles ampliaciones, los distintos módulos están comentados detalladamente.

Capítulo 4

Diseño

En este capítulo se describe el diseño del sistema implementado, describiendo los componentes del sistema que dan respuesta a las funcionalidades descritas en el capítulo anterior, así como la política de seguridad a aplicar en la corporación.

4.1 Diseño de la política de seguridad

Todas las empresas disponen de activos fundamentales sin los cuales no podrían seguir operando, como la información. Debido a que existen estos activos, las organizaciones deben seguir una estrategia para protegerlos con el fin de buscar la continuidad del negocio, conocida como política de seguridad.

En este apartado del documento se pretende definir una política de seguridad para dispositivos Android. Para ello, se analizarán algunos aspectos relaciones con el modo de utilización de la red y del dispositivo.

Se tendrá en cuenta que la política de seguridad debe servir para un entorno empresarial con una política BYOD y por tanto, deben tenerse en cuenta características especiales de este tipo de políticas, como controlar el numero de dispositivos móviles que puede poseer o conectar cada usuario de la organización.

A su vez, abogando por el ahorro y la economía de la empresa, se utilizarán herramientas de código abierto, que serán explicadas en profundidad en el Capitulo 5, por lo que se evitaran múltiples gastos como las licencias. Además, esto permitirá una mayor capacidad de personalización del sistema dependiendo de las necesidades en cuanto a seguridad que surjan en cada momento.

4.1.1. Dispositivos sin rootear

Como se comentó en secciones anteriores de este documento, no se procederá a realizar el *rooteo* de ninguno de los dispositivos. Esto se debe, en primer lugar, a que para *rootear* todos los dispositivos que se conectan a la red de la organización sería necesario disponer de ellos físicamente, lo cual no es viable, y en segundo lugar y la razón más importante, es que el hecho de *rootear* todos los dispositivos conlleva un riesgo de seguridad muy elevado, ya que si una aplicación es maliciosa y está instalada en el dispositivo, al ser el usuario administrador, todas las aplicaciones tienen acceso a toda la información y por tanto, la aplicación tendría acceso a toda la información del dispositivo como documentos confidenciales de la empresa que el empleado se haya podido descargar.

4.1.2. VPN

Para poder habilitar redes privadas distribuidas y comunicar de forma segura cada uno de los nodos de una red pública hay que evitar que los datos sean interceptados. Con una VPN o Red Privada Virtual, los usuarios remotos que pertenecen a una red privada pueden comunicarse de forma libre y segura entre redes remotas a través de redes públicas.

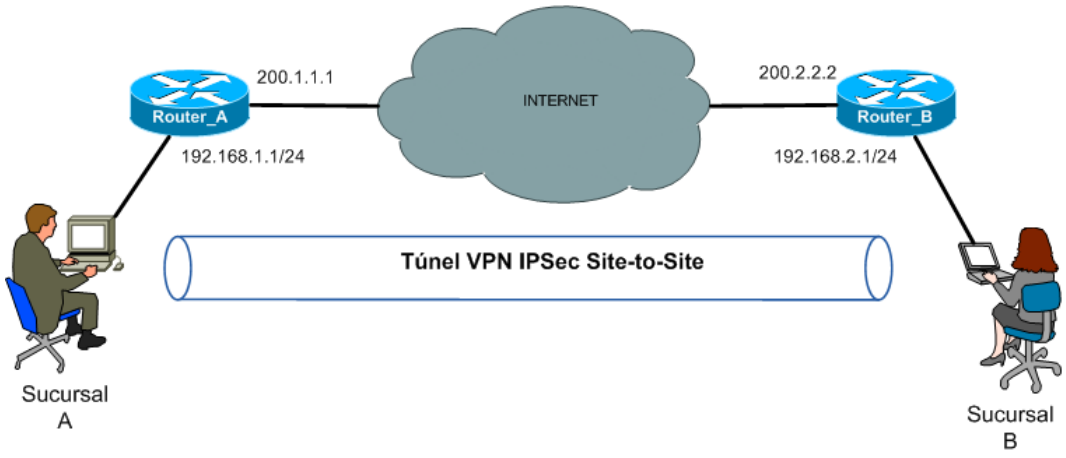

Figura 4. Esquema VPN

Recuperado de http://www.redescisco.net/v2/art/interconectando-sucursales-mediante-una-vpn-ipsec-site-site/

Una VPN transporta de manera segura por Internet a través de un túnel establecido entre dos puntos que negocian un esquema de cifrado y autenticación para el transporte. Los datos trasmitidos entre ambos ordenadores son cifrados mediante el Point-to-Point Protocol (PPP) y posteriormente enrutados por un dispositivo PPTP.

La VPN es de gran utilidad a la hora de conseguir un entorno corporativo seguro, ya que a la vez que permite que los empleados puedan realizar todo tipo de transacciones y comunicaciones con el exterior o interior de la compañía de manera segura, permite al mismo tiempo que empleados que no se encuentren en las instalaciones de la corporación puedan acceder desde cualquier punto de acceso a internet y puedan comunicarse de manera segura, ya que todo el tráfico de datos irá a través del túnel VPN.

4.1.3. Firewall

Un firewall o cortafuegos es una combinación de hardware y software que aísla a la red interna de la organización de Internet, permitiendo pasar algunos paquetes y bloqueando el paso a otros.

El firewall es imprescindible en la infraestructura de una organización, ya que es el punto por el que pasa todo el tráfico, tanto entrante como saliente y por tanto, si se encuentra bien configurado puede controlar las comunicaciones que se están realizando a través del router y bloquear aquellas externas que puedan ser potencialmente peligrosas, así como bloquear el tráfico interno hacia el exterior que pueda suponer un riesgo para la empresa.

En nuestro caso, el firewall se configurará de manera que obligue a utilizar una conexión VPN para poder comunicarse tanto hacia dentro como hacia fuera de la empresa. De este modo el administrador de red se asegura que sólo personas que tengan configurados el canal VPN y que por tanto son empleados de la organización, tengan acceso a la red de la empresa y además, al hacerlo mediante VPN, la comunicación será de manera segura.

4.1.4. Contraseñas

La autenticación de usuarios, tal y como se ha explicado en la Introducción, es un componente esencial en la seguridad de los sistemas de la información. Dicha autenticación se puede llevar a cabo mediante tres procedimientos [26]:

- Autenticación basada en algo que conoce el usuario. Un ejemplo claro serían las contraseñas.
- Autenticación basada en algo que posee el usuario. Este es el caso de cualquier token identificativo como una smartcard, dni, etc.
- Autenticación basada en algo que el usuario es. Este es el caso de la biometría, como las huellas dactilares, escáner de retina, etc.

Las contraseñas forman parte de todos los sistemas de seguridad que utilizamos de forma habitual y aunque no sea el mejor de los sistemas de seguridad que se pueden tener, sirve como un primer muro de defensa que un atacante deberá superar.

Debido a que muchos de los ataques que se producen se deben a que las contraseñas están mal seleccionadas o simplemente se han dejado las contraseñas por defecto, es necesaria que la empresa tenga una política respecto a la fortaleza que debe tener una contraseña para evitar precisamente que un atacante pueda entrar en la red simplemente utilizando contraseñas como "user" o "1234".

A continuación se muestran algunas de las recomendaciones a la hora de hacer una política de seguridad sobre contraseñas en el entorno empresarial [20]:

- Se deben utilizar al menos 8 caracteres para crear la clave.
- Se deben utilizar dígitos, letras y caracteres especiales.
- Se recomienda alternar mayúsculas y minúsculas.
- Elegir una contraseña que sea fácilmente recordable, ya que el hecho de apuntar la contraseña supone un enorme peligro de seguridad.
- Se debe cambiar la contraseña con una cierta regularidad.

- Evitar utilizar la misma contraseña en varios servicios, ya que si alguien obtiene acceso a una de ellas inmediatamente tendrá acceso a todos los demás servicios.
- No se debe utilizar información personal en la contraseña como el nombre o fecha de nacimiento, ya que son fácilmente rompibles.
- Se debe intentar formar una palabra fácil de recordar pero que no exista en el diccionario de ningún idioma, ya que de esta manera se evitan los ataques por diccionario.

4.1.5. Antivirus

Los antivirus son una capa muy importante a la hora de defender nuestros dispositivos. Existen múltiples versiones gratis en los *market* para Android y son de gran ayuda ya que proporcionan una primera capa de seguridad mediante el análisis de nuestro dispositivo, mejora del rendimiento e incluso tienen opciones para navegar de forma segura por internet.

Todas estas razones más que razonables hacen necesario tener un antivirus instalado en el dispositivo y además, los antivirus suelen disponer de logs en el propio dispositivo, por lo que el administrador de red podría tener acceso a ellos para una monitorización más personalizada de cada dispositivo.

4.2 Diseño del sistema MDM

En este apartado se describe el diseño del sistema MDM creado en profundidad. Entre los aspectos que se explican se encuentran la arquitectura del sistema, el entorno tecnológico y la descripción de los flujos de operación.

4.2.1. Arquitectura del sistema

El sistema dispone de una arquitectura del tipo cliente/servidor formada por la aplicación MDM para Android y el sistema MDM del servidor. De esta forma, los usuarios se validan en el sistema MDM y permiten la moni-

torización del dispositivo, enviando información interesante al servidor que podrá ser analizada posteriormente.

4.2.2. Entorno tecnológico

Los equipos y dispositivos sobre los que se ejecutarán tanto el cliente como el sistema MDM deberán cumplir una serie de requisitos para el correcto funcionamiento.

Los dispositivos móviles deberán tener sistema operativo Android y tener acceso a internet, ya sea mediante datos móviles o Wi-Fi, ya que es necesaria para poder comunicarse con el servidor.

El servidor, por su parte, deberá disponer de conexión a internet, tener instalado Apache y la base de datos MySQL.

4.2.3. Descripción de los flujos de operación

El cliente GCM está dividido en 4 actividades: el login, el registro, el cambio de contraseña y la confirmación. La Figura 5 muestra los flujos que se producen entre ellas, comenzando por la actividad de login, que permite al usuario autenticarse, acceder a la actividad de registro o a la de cambio de contraseña. Desde la actividad de registro el usuario puede registrarse o volver a la actividad de login. Desde la actividad de cambio de contraseña se puede volver a la de login. Una vez validado, el usuario accede a la actividad de confirmación donde deberá realizar la doble autenticación. Desde esta actividad se puede volver al login o bien continuar la confirmación para proceder a la monitorización del dispositivo.

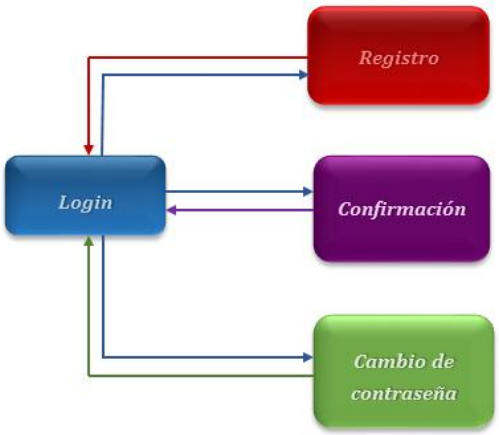

Figura 5. Flujos de operación cliente MDM

Debido a que se trata de una aplicación cliente/servidor, las comunicaciones que se producen entre el cliente y el servidor tienen todas el siguiente modelo (ver Figura 6):

- El agente ejecuta una determinada acción y se la comunica al servidor.
- El servidor recibe los datos y comprueba su validez.
- Si los datos son correctos, procederá a guardarlos en la base de datos.
- Una vez operado el servidor, éste le envía un mensaje de respuesta al cliente.

Figura 6. Flujo entre el cliente y el servidor

En general, todas las comunicaciones cumplen este esquema, debido a que el cliente se encarga de obtener datos del dispositivo y enviarlos al servidor para su almacenamiento.

En el caso del cambio de contraseña y de la confirmación, la respuesta del servidor sería el código de autenticación que deberá utilizar el cliente para confirmar la acción. En los demás casos, la respuesta del servidor será simplemente un mensaje de error, en caso de fallo, o un mensaje de acción completada, en caso de acción realizada con éxito.

Capítulo 5

Implementación del sistema MDM

En esta sección se describen los detalles relacionados con la implementación del sistema MDM desarrollado. Esto incluye las características del equipo de desarrollo, las plataformas utilizadas y la estructura y documentación del código.

5.1 Equipo de desarrollo

A continuación se detallan las características del hardware y software del equipo en el que se ha llevado a cabo el desarrollo del proyecto.

5.1.1. Hardware

Las características hardware del equipo de desarrollo son las siguientes:
Equipo de sobremesa HP Pro con procesador Intel Core i3 de 3.10GHz, 8GB de memoria RAM, disco duro de 500GB y sistema operativo Windows 7 Professional.

5.1.2. Software

Para el desarrollo del sistema MDM se ha hecho uso de una serie de aplicaciones y herramientas para la codificación y el diseño.

Software de programación

Las herramientas utilizadas para la codificación han sido:

- Eclipse Luna, para la codificación del agente MDM.
- Gedit, para la codificación de la parte del servidor.

Software de edición

Para redactar el presente documento se ha utilizado Microsoft Office Word.

Para la gestión de referencias consultadas se ha utilizado el gestor bibliográfico Mendeley [2].

5.2 Plataformas

En este apartado se analizarán las distintas tecnologías, plataformas y lenguajes utilizados durante la realización del presente proyecto. Este apartado incluye tanto herramientas de desarrollo como herramientas de configuración del sistema y cabe destacar, que todas las herramientas utilizadas son de software libre.

Xampp

Es un servidor independiente de plataforma y software libre, que consiste principalmente de una base de datos MySQL, un servidor web Apache y intérpretes para lenguajes de script PHP y Perl. Está liberado bajo licencia GNU(GNU´s Not Unix) y está disponible para multitud de sistemas operativos (Apache Friends). La elección de esta plataforma se debe principalmente a su fácil instalación y manejo a la hora de desarrollar, así como su facilidad para modificar configuraciones en el servidor Apache [22].

Eclipse

Es un IDE que contiene un workspace base y un administrador de plug-in que permite personalizar el entorno. Está escrito en Java y permite desarrollar aplicaciones en multitud de lenguajes de programación: Java, Android, Ada,C, C++, COBOL, etc. La elección de este entorno de trabajo se debe principalmente a la experiencia en su manejo y al soporte y ayuda a la hora de programar en Android [17].

OpenVPN

Es una aplicación de software libre que implementa técnicas de VPN para crear conexiones seguras punto a punto y conexiones remotas. Utiliza un protocolo de seguridad que utiliza SSL/TLS para el intercambio de claves, es capaz de atravesar NATs y firewalls, funciona bajo licencia de GNU y permite a los clientes autenticarse mediante un certificado creado por el servidor. Existen otras aplicaciones que implementan VPN, pero la elección de OpenVPN se debe a la facilidad para configurar y crear los certificados en el servidor, así como la posibilidad de configurar y regular los accesos al mismo. Además, dispone de un cliente diseñado para Android, que permite al cliente conectarse con el servidor al descargarse el certificado del servidor [12].

AVG

Anti-Virus Guard, es una familia de antivirus y software de seguridad que protege múltiples dispositivos en diversos sistemas operativos. En el caso de la implementación de este proyecto, se utiliza como una herramienta auxiliar que proporciona protección y un primer filtro frente a amenazas al dispositivo móvil Android [5].

ConnectBot

Es un cliente de software libre para dispositivos Android que se compone de una shell con SSH. Permite a los usuarios validarse de manera segura en un servidor, permitiéndoles realizar comandos desde el dispositivo como si de una terminal de Linux se tratase. Utiliza el estándar de encriptación utilizado por SSH2 para que la transmisión de datos se realice de manera segura y no pueda ser interceptada [8]. Su utilidad para este proyecto, es que se puede configurar de manera que funciona como un proxy, por lo que las conexiones desde el navegador realizadas desde el dispositivo correrían a través del servidor.

Mobiwol

Es una firewall de software libre que no requiere permisos de root. La ventaja de esto, es que no necesitas rootear el dispositivo para poder con-

trolar las conexiones que se realizan desde tu dispositivo a internet. Esta aplicación te permite observar que dispositivos tienen acceso a internet, ya sea Wifi o datos móviles, y lo que es más importante, te permite decidir que aplicaciones pueden tener acceso o no a internet. Esto es importante de cara a poder tener nuestro dispositivo infectado, ya que podríamos observar una aplicación maliciosa que en principio no debería necesitar acceso a internet que sin embargo si se está conectando a ella y podríamos solucionarlo.

Adb

Es una herramienta que incluye el SKD de Android que consta de un cliente y servidor que se comunican entre ellos. El adb (Android Debug Bridge) es accedido típicamente desde la línea de comandos, aunque existen interfaces gráficas [19].

Adb permite realizar las siguientes acciones:

- Copiar ficheros en el dispositivo.
- Debuguear aplicaciones.
- Ejecutar comandos en la shell.
- Obtener los logs del sistema y de las aplicaciones.
- Instalar y borrar aplicaciones.

GCM

Es un servicio, que proporciona Google, que permite enviar datos desde un servidor hasta aplicaciones Android. Este servicio se suele utilizar para enviar notificaciones desde el servidor al dispositivo móvil para avisarle de capturas de datos, actualizaciones, envíos de números secretos para doble autenticación, etc [24].

Java

Java es un lenguaje de programación orientado a objetos e independiente de la plataforma. Ofrece una gran facilidad a la hora de desarrollar aplicaciones cliente/servidor.

Se ha utilizado para programar la parte del agente MDM, ya que permite programar aplicaciones para dispositivos móviles utilizando las APIs para Android. Se ha utilizado además, la API de Google Volley, que permite realizar las comunicaciones y conexiones con un servidor de manera transparente y sencilla, ya que se encarga de todos los problemas de conexión e hilos.

PHP

PHP es un lenguaje de scripting especialmente diseñado para el desarrollo de aplicaciones web. En el desarrollo del sistema MDM se ha utilizado para codificar toda la parte del servidor, esto es, conexiones con los dispositivos móviles Android y la comunicación con la base de datos MySQL [16]. Además, se han utilizado filtros existentes en php para sanitizar las entradas de datos recibidas del agente MDM [33].

OpenSSL

OpenSSL es un proyecto criptográfico de software libre que implementa los protocolos de SSL(Secure Socket Layer) y TLS (Transport Layer Security), así como otras funcionalidades de carácter criptográfico general [14].

En el desarrollo del proyecto se ha utilizado para crear la clave pública y privada del servidor apache.

Keytool

Keytool es una herramienta incluida en el paquete Java SE, que permite crear credenciales X.509. Permite gestionar y administrar claves y certificados a los usuarios, además de permitir firmar certificados propios, conocidos como certificados autofirmados.

El uso de certificados autofirmados es una mala práctica de seguridad, pues se requiere una autoridad certificadora para que un certificado sea

confiable. Sin embargo, en el caso de este trabajo, y desde un punto de vista puramente práctico, se usará este tipo de certificado.

En particular, se ha utilizado Keytool para crear el certificado autofirmado del servidor y permitir así utilizar conexiones mediante SSL entre el servidor y los dispositivos Android. Para ello, se ha tenido que modificar el formato del certificado para pasarlo a formato BKS [21].

5.3 Estructura y documentación del código

Tanto para el cliente MDM como para la parte del servidor, se ha intentado modular el código atendiendo a su funcionalidad. A continuación se describirán cada una de las clases del cliente MDM y los ficheros php del servidor.

5.3.1. Cliente MDM

En este apartado se describirán cada uno de los paquetes creados para la aplicación Android, así como las clases que se encuentran dentro de ellos.

- ***es.uam.eps.activity***: este paquete contiene todas las clases que son actividades, es decir, que son pantallas que verá el usuario y con las que podrá interactuar. Estas clases son:
  - **register.java**: contiene los métodos necesarios para completar el proceso de registro de un usuario. El usuario introduce su usuario y contraseña corporativos, y esta información es enviada junto al identificador único del dispositivo al servidor. Posteriormente, recibe la respuesta del servidor y muestra un mensaje dependiendo de si registro ha sido satisfactorio o no.
  - **login.java**: contiene los métodos necesarios para permitir al usuario llevar a cabo el proceso de login. El usuario introduce su usuario y contraseña corporativos , y lo envía al servidor. Posteriormente, recibe la respuesta del servidor y muestra un mensaje dependiendo de ella. Así mismo, en

caso de login satisfactorio, se procede a obtener de GCM un número identificador para el dispositivo, el cual es enviado al servidor para guardarlo junto al usuario.

- **confirmation:java**: contiene los métodos necesarios para permitir que el proceso de confirmación se realice de forma correcta. Esta clase se encarga de realizar un proceso de doble autenticación para evitar posibles suplantaciones. En primer lugar, le comunica al servidor GCM el número identificativo obtenido anteriormente. Posteriormente, espera recibir el código de autenticación vía GCM y tras introducirlo, se envía el código introducido al servidor. Cuando el cliente recibe la respuesta, si el código era incorrecto aparecerá un mensaje solicitando un código correcto, sin embargo, si recibe una respuesta satisfactoria, procederá a lanzar los servicios de monitorización del dispositivo.
- **ChangePassword.java**: contiene los métodos necesarios para permitir el proceso de cambio de contraseña del usuario. El usuario deberá introducir su usuario y contraseña antigua, seguido de la nueva contraseña. Se comprobará si el usuario se encuentra registrado, y en caso de ser así, recibirá una notificación con un código de autenticación para confirmar el cambio de contraseña.

- ***es.uam.eps.connection***: este paquete contiene una única clase que es la encargada de gestionar todas las comunicaciones con el servidor.
  - **InterfazServer.java**: contiene los métodos necesarios para establecer la comunicación con el servidor MDM. En esta clase se utiliza la API Volley para realizar todas las comunicaciones con el servidor. Todos los métodos de envío de información al servidor se realizan mediante POST y utilizando un TrustManager para permitir la comunicación vía SSL.

- ***es.uam.eps.GCMConfig***: este paquete contiene dos clases con la configuración y la comunicación con el sistema GCM.
    - **Config.java**: contiene unos valores estáticos relacionados con la configuración del GCM. Entre ellos se encuentran, la url del servidor MDM con el que se comunicará, el identificador del proyecto de Google necesario para utilizar GCM y un parámetro que se utilizará para enviar los mensajes hacia el servidor GCM.
    - **ShareExternalServer.java**: contiene un único método que se encarga de comunicar al servidor GCM el número identificativo obtenido junto al usuario al que irá conectado.

- ***es.uam.eps.receiver***: paquete que contiene tanto los servicios como los recibidores del proyecto.
    - **ApkReceiver.java**: servicio que se encarga de obtener la lista de apks instaladas en el dispositivo monitorizado. Una vez obtenida la lista de apks, obtiene también la lista de firmas de cada una de las apks y calcula un hash de ella. De esta forma, este servicio le envía al servidor las apks encontradas, el hash calculado para cada apk y las firmas de cada una de ellas. Estos datos se calculan ya que en el servidor se puede tener una base de datos de apks que pueden ser malware y con esta información se puede comprobar si una apk es maliciosa comprobando el hash y las firmas de ellas.
    - **GcmBroadcastReceiver.java**: broadcast receiver que se encarga de esperar la recepción de un mensaje del servidor GCM y ,una vez recibido, inicia un servicio para producir una notificación del mensaje recibido.
    - **GCMNotificationIntentService.java**: servicio que se encarga de recibir el mensaje enviado por el servidor GCM y crear una notificación para el dispositivo con él.

- **HistorialReceiver.java**: servicio que se encarga de obtener el historial de navegación del dispositivo, encapsularlo y enviarlo al servidor MDM para su almacenamiento y posterior análisis en caso de malware encontrado en el dispositivo.
- **LogCatReceiver.java**: servicio que se encarga de obtener los procesos activos del dispositivo, filtrando los generales de los dispositivos Android, y enviándolos al servidor para su almacenamiento y posible análisis para detectar procesos maliciosos.
- **VPNReceiver.java**: servicio que se encarga de comprobar si el cliente de OpenVPN está ejecutándose en el dispositivo y enviar dicha información al servidor para que, en caso de no estar activado, el dispositivo reciba una notificación para que active la conexión vpn.
- **GpsReceiver.java**: servicio que se encarga de obtener los datos de la última localización del GPS del dispositivo y de enviarlos al servidor para su almacenamiento.
- **PdfReceiver.java**: servicio que se encarga de obtener los documentos descargados por el usuario y enviarlos al servidor para su almacenamiento y, en caso de que sean documentos confidenciales, para que el dispositivo reciba una notificación.

5.3.2. Servidor MDM

En este apartado se procederá a describir la implementación del servidor mediante php para la gestión del backend y la comunicación con los agentes MDM, estructura que se puede ver en la Figura 3.

- **config.php**: contiene la parte de configuración referente a la conexión con la base de datos, de manera que no se tiene que abrir la conexión con la base de datos desde cada php del servidor, simplemente se necesita incluir este fichero.

- **funcs.php**: contiene dos funciones, una relaciona con el envío de mensajes hacia el servidor GCM y, la otra, con la creación de un código aleatorio para la doble autenticación, que son utilizadas desde los demás archivos del servidor.
- **gcm.php**: se encarga tanto de guardar en la base de datos el número identificativo del dispositivo para la comunicación con el GCM, como de crear el código de doble autenticación, guardarlo en la base de datos, tras haberle calculado la función hash, y enviarlo al dispositivo vía GCM.
- **register.php**: se encarga de comprobar los datos de registro enviados por el dispositivo para registrar a los nuevos usuarios que tengan cuenta en la empresa e informar a los que ya se encuentren registrados en el sistema MDM. Para comprobar si las contraseñas son iguales, se le aplica una función *hash* a la enviada por el usuario para comprobar si es equivalente a la almacenada en la base de datos. Del mismo modo, el ID del dispositivo también es guardado en la base de datos tras habérsele calculado la función hash.
- **login.php**: se encarga de comprobar los datos de login enviados por el usuario del dispositivo y comprobar si el usuario y contraseña son correctos, así como si el ID del dispositivo es el mismo al guardado en la base de datos. Para ello, se calcula el hash de la contraseña y del ID del dispositivo para ver si concuerdan con los almacenados en la base de datos del servidor.
- **changePassword.php**: se encarga de comprobar si el usuario y contraseña enviados son correctos y de enviar un código de confirmación al dispositivo para modificar la contraseña.
- **confirmacion.php**: se encarga de comprobar si el código introducido por el usuario corresponde con el almacenado en el servidor. Para ello calcula el hash del código enviado por el usuario y comprueba si son iguales. En caso de ser así, envía un mensaje de confirmación satisfactoria al dispositivo y, en caso contrario, le envía un mensaje de código incorrecto.

- **apks.php**: se encarga de recibir la lista de apks, hashes y firmas enviadas por el servicio del agente MDM. Una vez recibido, parsea el contenido y lo almacena en la base de datos del servidor.
- **historial.php**: se encarga de recibir el historial del navegador del dispositivo. Una vez lo ha recibido, parsea el contenido y lo almacena en la base de datos del servidor.
- **logcat.php**: se encarga de recibir el listado de procesos en ejecución del dispositivo. Una vez lo ha recibido, parsea el contenido y lo almacena en la base de datos del servidor.
- **vpnState.php**: se encarga de recibir el estado de la conexión VPN del dispositivo y almacenarlo en la base de datos del servidor. Comprueba si está conectada y, en caso de no estarlo, envía una notificación al dispositivo, mediante GCM, para que el usuario la active.
- **gps.php**: se encarga de recibir los datos del GPS como el estado, las coordenadas y la fecha de la localización y de almacenarlas en la base de datos. Esta información puede ser utilizada después como un mecanismo de localización del dispositivo en caso de pérdida o robo.
- **pdf.php**: se encarga de recibir los documentos descargados por el usuario en su dispositivo y de almacenarlos en la base de datos. En caso de ser documentos confidenciales, se podría enviar una notificación al dispositivo alertando de ello.

Capítulo 6

Plan de pruebas

En esta sección se describen las pruebas llevadas a cabo durante el desarrollo y después de la codificación del proyecto. Las pruebas pueden realizarse en cualquier momento durante el desarrollo del software y tienen como objetivo encontrar el mayor número posible de errores con el fin de poder resolverlos antes de la finalización del proyecto. Se han realizado dos tipos de pruebas: pruebas unitarias y pruebas de integración.

Las pruebas unitarias son aquellas que comprueban el correcto funcionamiento de todos los módulos de manera aislada y verifican que el sistema responde de forma correcta frente a todos los casos, especialmente frente a los casos extremos.

Las pruebas de integración se realizan tras las pruebas unitarias y , generalmente, al finalizar los módulos, ya que su objetivo es comprobar si los módulos que por separado funcionan correctamente funcionan también al interactuar con otros módulos.

La planificación y ejecución de pruebas son una parte fundamental en el desarrollo del software, ya que permiten comprobar la funcionalidad del producto y descartar problemas que se puedan producir ante flujos de datos anormales.

6.1 Pruebas unitarias

A continuación se muestran las pruebas unitarias realizadas tanto para el agente MDM como para la parte del servidor junto con el resultado obtenido. Cabe destacar que las pruebas unitarias del agente MDM se realizaron, como es lógico, sin realizar la conexión con el servidor, ya que las pruebas contra este se realizaran en las pruebas de integración.

6.1.1. Agente MDM

Módulo prueba	Descripción	Resultado
Registro	Se deja alguno de los campos del formulario en blanco.	Correcto, el sistema solicita todos los datos en el formulario.
Registro	Se introducen usuario y contraseña y se selecciona la opción registrar.	Correcto, el sistema dispone de todos los datos necesarios y se dispone a registrar al usuario.
Login	Se deja alguno de los campos del formulario en blanco.	Correcto, el sistema solicita la introducción de todos los datos.
Login	Se introduce usuario y contraseña de un usuario registrado.	Correcto, el sistema permite el paso y accede a la pantalla de confirmación.
Login	Se introduce usuario incorrecto.	Correcto, el sistema informa de que los datos introducidos no son correctos.
Login	Se selecciona la opción Registrar.	Correcto, el sistema reacciona al botón y muestra la pantalla de registro.
ChangePassword	Se introduce el usuario, contraseña antigua y la nueva contraseña y se elige la opción Cambiar.	Correcto, el sistema comprueba los datos y en caso correcto envía un código, mientras que si es incorrecto se muestra un mensaje de error.
ChangePassword	Se introduce el código de confirmación de cambio de contraseña.	Correcto, en caso de ser correcto se produce el cambio de contraseña y en caso contrario, se genera un mensaje de código incorrecto.
Confirmación	Se deja el formulario en blanco y se selecciona la opción Confirmar.	Correcto, el sistema solicita la introducción del código.
Confirmación	Se introduce un código incorrecto.	Correcto, el sistema muestra un mensaje de Código incorrecto.
Confirmación	Se introduce un código valido.	Correcto, el sistema valida la confirmación.
ApkReceiver	El servicio se lanza desde la pantalla de Confirmación.	Correcto, el servicio se ejecuta de manera correcta.
HistorialReceiver	El servicio se lanza desde la pantalla de Confirmación.	Correcto, el servicio se ejecuta de manera correcta.
LogCatReceiver	El servicio se lanza desde la pantalla de Confirmación.	Correcto, el servicio se ejecuta de manera correcta.
VPNReceiver	El servicio se ejecuta de forma periódica.	Correcto, el servicio se ejecuta de manera correcta y obtiene información de la conexión VPN.

gpsReceiver	El servicio se ejecuta desde la pantalla de Confirmación.	Correcto, el servicio se ejecuta de forma correcta y obtiene la información del GPS.
pdfReceiver	El servicio se lanza desde la pantalla de Confirmación.	Correcto, el servicio se ejecuta de forma correcta y obtiene los documentos descargados.
GCM	Se envía un mensaje al dispositivo vía GCM.	Correcto, el agente MDM recibe el mensaje y genera la notificación.

Tabla 2.Pruebas unitarias del agente MDM

6.1.2. Sistema MDM

Módulo prueba	Descripción	Resultado
Registro	Se envía los parámetros del formulario en blanco.	Correcto, el sistema genera un mensaje de fallo en los argumentos.
Registro	Se envía los datos de un usuario que no está registrado en la empresa.	Correcto, el sistema devuelve un mensaje de error por no pertenecer a la compañía.
Registro	Se envía los datos de un usuario registrado en la empresa y no registrado en el sistema MDM.	Correcto, el sistema registra al usuario en el servidor.
Registro	Se envía los datos de un usuario ya registrado en el sistema MDM.	Correcto, el sistema genera un mensaje de error por ya estar registrado el usuario.
Login	Se envía los datos de un usuario registrado.	Correcto, el sistema devuelve un mensaje de validación completada.
Login	Se envía los datos de un usuario no registrado.	Correcto, el sistema genera un mensaje de credenciales incorrectas.
ChangePassword	Se envían los datos de cambio de contraseña de un usuario.	Correcto, el sistema comprueba los datos y si son correctos envía un código de confirmación al dispositivo.
Confirmación	Se introduce un código invalido.	Correcto, el sistema genera un mensaje en el que solicita un código valido.
Confirmación	Se introduce un código valido.	Correcto, el sistema genera un mensaje de confirmación completada.
ApkReceiver	Se envía una lista de apks al servidor.	Correcto, el sistema obtiene las apks y las guarda junto al usuario al que corresponden.
HistorialReceiver	Se envía un historial al servidor.	Correcto, el sistema obtiene las distintas urls del historial y las almacena en la base de datos junto al usuario.
LogCatReceiver	Se envía información de proce-	Correcto, el sistema obtiene los procesos que

	sos en ejecución al servidor.	se están ejecutando y los almacena en la base de datos junto al usuario.
VPNReceiver	Se envía información sobre la conexión de la VPN.	Correcto, el sistema analiza el estado de la conexión y lo guarda en la base de datos. Adicionalmente, si se encuentra desconectada genera un mensaje a enviar vía GCM.
gpsReceiver	Se envía información del GPS.	Correcto, el sistema obtiene los datos del GPS y lo almacena de la base de datos.
pdfReceiver	Se envía información sobre documentos descargados.	Correcto, el sistema obtiene los documentos descargados y los almacena en la base de datos.
GCM	Se genera un mensaje y se envía al servidor GCM.	Correcto, el servidor GCM recibe el mensaje y lo redirecciona al dispositivo correspondiente.

Tabla 3. Pruebas unitarias del sistema MDM

6.2 Pruebas de integración

En las pruebas de integración se va a comprobar la correcta interacción entre el agente y el sistema MDM.

Módulo prueba	Descripción	Resultado
Registro	Se introducen los datos de un usuario valido.	Correcto, los datos se reciben en el sistema MDM, el usuario queda registrado y el agente recibe un mensaje de registro correcto.
Login	Se valida un usuario registrado y se le da acceso al sistema.	Correcto, los datos del login llegan al servidor y comprueba que las credenciales concuerdan. Finalmente envía un mensaje de login correcto al agente.
Confirmación	Se introduce un código válido.	Correcto, el servidor comprueba que el código introducido corresponde con el almacenado en la base de datos y envía un mensaje de confirmación valida al agente.
ChangePassword	Se introducen los datos para cambiar de contraseña.	Correcto, el sistema comprueba los datos recibidos y en caso de ser correctos envía al dispositivo un código de confirmación, en caso de ser erróneos se envía un mensaje de error.
GCM	Se genera un código de confirmación y se envía al servidor GCM.	Correcto, el código de acceso se recibe en el servidor GCM y éste lo envía al dispositivo, donde se recibe en modo de notificación.
ApkReceiver	El servicio obtiene la lista	Correcto, el servidor recibe las apks y las alma-

	de apks y la envía al servidor.	cena en la base de datos junto al usuario.
HistorialReceiver	El servicio obtiene el historial de navegación y lo envía al servidor.	Correcto, el servidor recibe el historial y lo almacena en la base de datos junto al usuario.
LogCatReceiver	El servicio obtiene la lista de procesos y la envía al servidor.	Correcto, el servidor recibe la lista de procesos y los almacena en la base de datos junto al usuario correspondiente.
GpsReceiver	El servicio obtiene los datos del GPS y los envía al servidor.	Correcto, el servidor recibe los datos del GPS y los almacena en la base de datos.
PdfReceiver	El servicio obtiene los documentos descargados y los envía al servidor.	Correcto, el servidor recibe los documentos descargados y los almacena en la base de datos.
VPNReceiver	El servicio captura el estado de la conexión VPN y la envía al servidor.	Correcto, el servidor recibe el estado de la conexión, la almacena en la base de datos y en caso de que esté desconectada envía una notificación al dispositivo vía GCM para que active la conexión VPN.

Tabla 4. Pruebas de integración

6.3 Comprobación de rendimiento

De acuerdo a los requisitos no funcionales, existían unas limitaciones temporales para que se ejecutase cada acción dentro del sistema. Para comprobar que esos tiempos establecidos se cumplen, se han realizado múltiples ejecuciones variando el nivel de actividad tanto en el sistema MDM como en el dispositivo móvil de manera que el nivel de estrés del dispositivo fuese más elevado. Tras realizar estas pruebas, se ha podido comprobar que efectivamente los tiempos establecidos para la ejecución de cada una de las actividades, anteriormente descritas en el documento, se cumplen de manera correcta.

Capítulo 7

Conclusiones y líneas futuras

El objetivo principal del trabajo realizado ha sido efectuar un análisis de las políticas actuales respecto a la utilización de los dispositivos móviles en el sector empresarial. Tras ello se han analizado las diversas herramientas MDM existentes en el mercado y cuyo objetivo es la verificación de políticas de seguridad en el dominio BYOD. Finalmente, se ha implementado un prototipo básico de sistema MDM en un entorno BYOD de acuerdo con las necesidades puestas de relieve en los análisis previos. .

En lo referente a las políticas genéricas de seguridad que afectan a dispositivos móviles, se ha realizado un análisis de cada una de ellas, describiendo sus características y destacando los puntos a favor y en contra de cada uno de ellos.

En cuanto a las herramientas investigadas, se han examinado sus características principales y se ha explicado la posible utilización y uso que podrían tener a la hora de desarrollar un entorno MDM eficaz.

Dentro del contexto del desarrollo del sistema MDM, se ha realizado un análisis de los requisitos necesarios para poder desarrollar un sistema de monitorización eficiente y eficaz a la hora de detectar fallos de configuración y vulnerabilidades en los dispositivos móviles de una cierta red de trabajo.

Mediante el diseño de la política de seguridad se ha profundizado en aspectos relacionados con la seguridad en los sistemas de una corporación, y se ha explicado qué herramientas utilizar para conseguir cada uno de los puntos descritos en la política.

Finalmente, en la implementación se ha detallado todo el proceso de desarrollo del sistema MDM, el cual ha sido un proceso largo, no solo por el desarrollo en sí, sino porque se han unificado todos los conocimientos ad-

quiridos a la hora de investigar este tema y se ha tenido que investigar la utilización de varias herramientas para configurar el sistema.

A continuación se exponen algunos de los aspectos a los que ha sido necesario enfrentarse:

- Servidor: se ha configurado un servidor Apache desde cero, y se ha forzado que las comunicaciones cliente-servidor se realicen de modo seguro mediante SSL.
- Base de datos: se ha creado una base de datos MySql para gestionar los dispositivos del sistema MDM.
- VPN: se ha implementado un servidor VPN por el que pasa todo el tráfico de procedente de los dispositivos con el fin de garantizar una navegación segura.
- SSH: se ha configurado el servidor para permitir conexiones SSH desde el dispositivo móvil al servidor, con la consecuente configuración de la herramienta correspondiente en Android, así como la configuración de un proxy para que las búsquedas en el navegador de los dispositivos se haga a través del servidor.
- MDM: se ha estudiado algunas de las herramientas disponibles, así como las funcionalidades que proporcionan, y se ha implementado un sistema básico de monitorización.
- SSL: se ha tenido que implementar en Android el agente de manera que permitiese realizar conexiones vía SSL, lidiando con los pertinentes errores de certificados digitales y los problemas de certificados auto firmados.

Para concluir, aunque el sistema desarrollado no dispone de todas las funcionalidades de las que puede disponer un sistema MDM comercial actual, ha servido para profundizar en el tema de la seguridad referente a los dispositivos móviles en un entorno empresarial, conocer tecnologías que son útiles a la hora de componer un entorno seguro, aprender mecanismos de comunicación entre un servidor y un cliente, así como practicar en la creación de ambos desde cero, y para ampliar conocimientos en bases de datos y Android.

La línea de trabajo futuro más clara sería ampliar el sistema MDM con nuevas funcionalidades como las siguientes:

- Se podría implementar el borrado remoto del dispositivo, de manera que si el sistema MDM detecta algún malware en un dispositivo éste podría borrarlo inmediatamente para evitar daños graves tanto en el dispositivo como en la corporación.
- Utilización de bases de datos con firmas de malware para agilizar la validación de apks. Existen bases de datos de malware, como Androguard [28] o VirusTotal [34], que permiten comprobar si las aplicaciones descargadas son malware, mediante la comparación de las firmas de las aplicaciones y de sus *hashes*, datos que ya son almacenados por el sistema MDM creado, y poder así notificar al usuario para que las elimine o incluso eliminarlas de manera remota.
- Además del análisis estático de aplicaciones, nuestro sistema MDM debería mejorarse mediante la incorporación de herramienta para el análisis dinámico de *malware*, como TaintDroid [34] o Comdroid [35].

Otra posible mejora respecto a la parte del servidor consistiría en realizar una interfaz que permitiese al administrador no solo visualizar los datos de manera más clara, sino que también permitiese elegir un dispositivo y enviarle una orden específica. De esta forma el manejo del sistema sería más fácil y eficiente respecto a localizar problemas con mayor rapidez. Asimismo, junto con la visualización de datos recogidos de los móviles conviene desarrollar mecanismos inteligentes de detección de eventos. A este respecto cabría la posibilidad de incluir nuestro MDM en sistemas SIEM (Security Information Event Management) [38]. Por último, habría que estudiar mecanismos conciliadores de protección de la seguridad y respeto de la privacidad de usuarios, lo que constituye unas de las necesidades más acucianteen el actual contexto tecnológico [36].

Con el objetivo de aprovechar todas las características que poseen los dispositivos móviles para fortalecer la seguridad del dispositivo y favorecer la monitorización, se podría hacer uso de lo que se conoce como autentica-

ción implícita. La autenticación implícita consiste básicamente en modelar el flujo normal de información en un dispositivo debido al uso habitual que hace del mismo un usuario. Aquel modelo permitiría establecer un perfil de las actividades que suele realizar el usuario, lo que, a su vez, permitiría detectar anomalías cuando algo ha ocurrido con el dispositivo o cuando el usuario que lo está utilizando no es el propietario o usuario habitual del mismo [37].

También sería muy interesante dar el salto a las demás plataformas móviles existentes , como iOS, BlackBerry, etc. De esta forma la verificación de políticas de seguridad en esquemasBYOD se podría realizar en su totalidad, ya que los empleados tendrían total libertad para elegir cualquier dispositivo que quisiesen, sin necesidad de que sea exclusivamente Android.

Cabe destacar que, una vez finalizado este proyecto, se continuará investigando en el tema de la seguridad de los dispositivos móviles, las maneras de gestionarlos y monitorizarlos, y se intentará mejorar el sistema MDM creado en este trabajo.

Anexo A: Diagrama de Gantt

En la Figura 7 se muestra el diagrama de Gantt del proyecto, en el que se muestran las distintas fases del mismo, así como su duración. La Figura 8 muestra la escala de tiempos para cada una de las fases del proyecto.

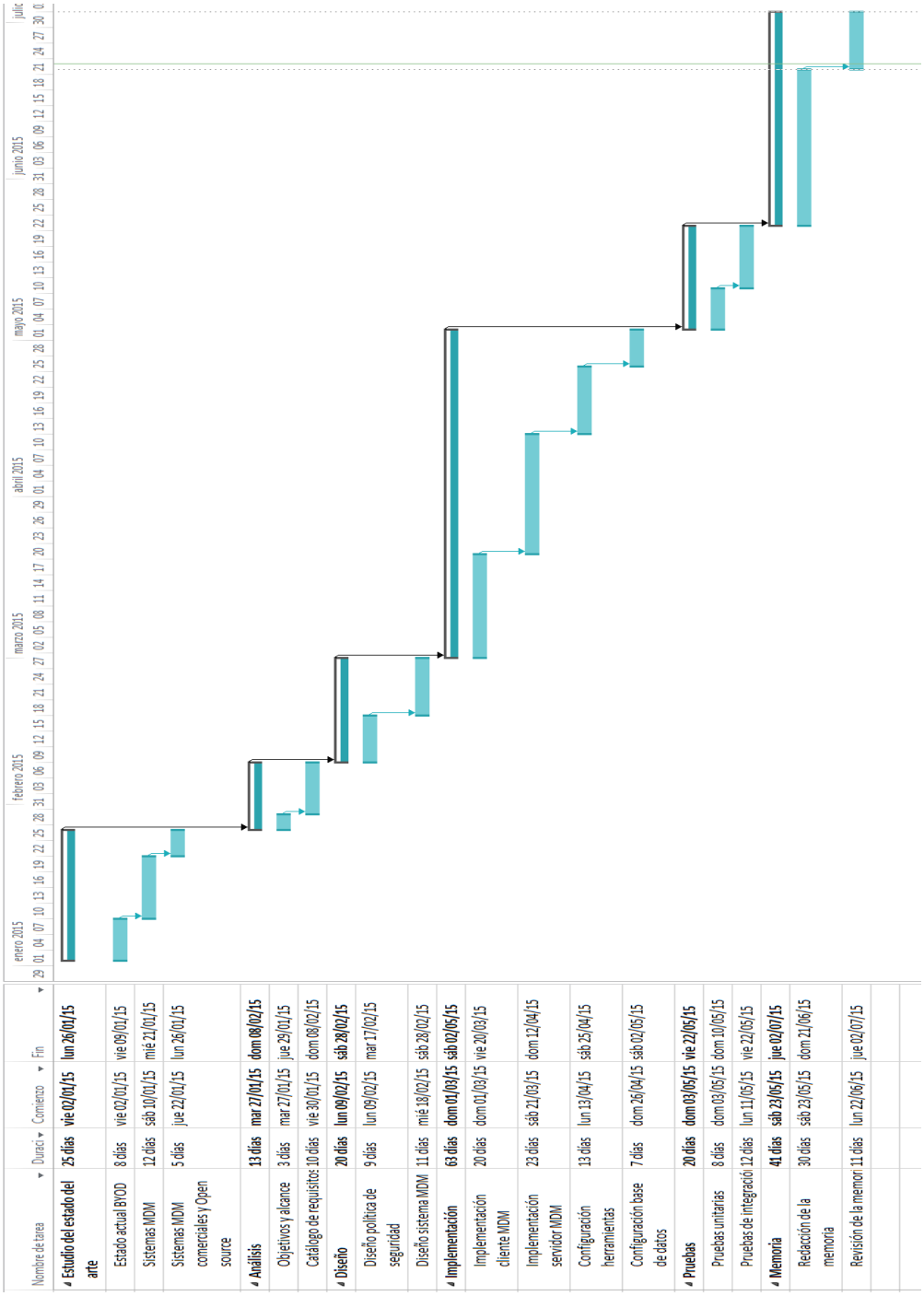

Figura 7. Diagrama de Gantt

Figura 8. Escala de tiempos

Anexo B: Base de datos

A continuación se muestran las tablas que conforman la base de datos de proyecto, explicando el contenido de cada una de ellas, así como el diagrama Entidad-Relación (ver Figura 9).

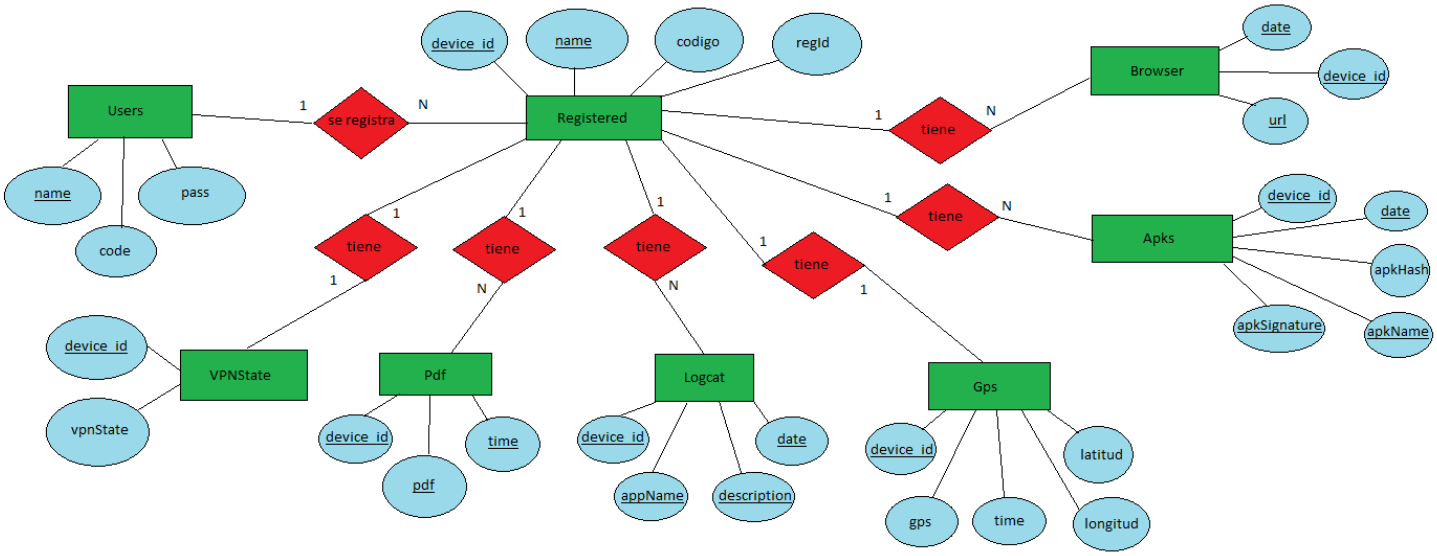

Figura 9. Diagrama Entidad-Relación de la BD

La Tabla 5 es la encargada de almacenar la lista de aplicaciones obtenidas del dispositivo, almacenando el identificador del dispositivo, el nombre de la aplicación, el hash de la aplicación, la firma de la aplicación y la fecha de captura.

Apks				
device_id	apkName	apkHash	apkSignature	date

Tabla 5. Tabla apks de la BD

La Tabla 6 se encarga de almacenar los datos del historial de navegación de los dispositivos. Los datos que almacena son el identificador del dispositivo, la fecha de captura y la url capturada.

Browser		
device_id	date	url

Tabla 6. Tabla browser de la BD

La Tabla 7 se encarga del almacenamiento de los datos de la localización GPS del dispositivo. Los datos que almacena son el identificador del dispositivo, el estado del GPS (activado/desactivado), fecha de los datos del GPS y las coordenadas del dispositivo.

Gps				
device_id	gps	latitud	longitud	time

Tabla 7. Tabla gps de la BD

La Tabla 8 almacena los datos de los procesos que están en ejecución. Entre los datos capturados se encuentran el identificador del dispositivo, el nombre de la aplicación a la que pertenece el proceso, la descripción del proceso y la fecha del proceso.

Logcat			
device_id	appName	description	date

Tabla 8. Tabla logcat de la BD

La Tabla 9 almacena los ficheros descargados por el usuario y almacenados en la sdcard del dispositivo. Los datos almacenados son el identificador del dispositivo, el nombre del archivo descargado y el tiempo en el que fue descargado del archivo.

Pdf		
device_id	pdf	time

Tabla 9. Tabla pdf de la BD

La Tabla 10 se encarga de almacenar los usuarios registrados en el sistema MDM. Los datos que almacena son el identificador del dispositivo y el nombre del usuario. El campo código almacena el código necesario para la doble autenticación. Por último, el campo regId almacena un valor necesario para identificar el dispositivo frente al servidor GCM.

Registered			
device_id	name	codigo	regId

Tabla 10. Tabla registered de la BD

La Tabla 11 funciona como un repositorio donde se guardan todos los usuarios y contraseñas de los empleados de la corporación.

Users		
name	pass	code

Tabla 11. Tabla users de la BD

La Tabla 12 almacena el estado de la conexión VPN del dispositivo para comprobar si se cumple la política de seguridad.

VPNState	
device_id	vpnState

Tabla 12. Tabla vpnState de la BD

Anexo C: Vista de la aplicación

En este Anexo, se mostrarán capturas de la aplicación Android desarrollada y se explicarán las comunicaciones que suceden en cada una de ellas.

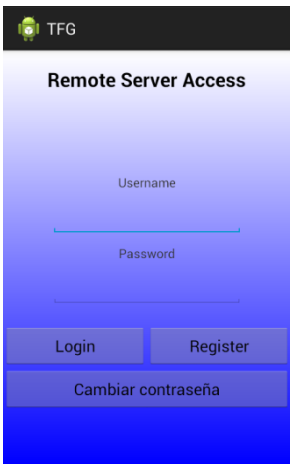

Figura 10. Pantalla de login inicial

En la Figura 10, se muestra la pantalla de login inicial, desde donde el usuario puede entrar al sistema, registrarse o cambiar su contraseña.

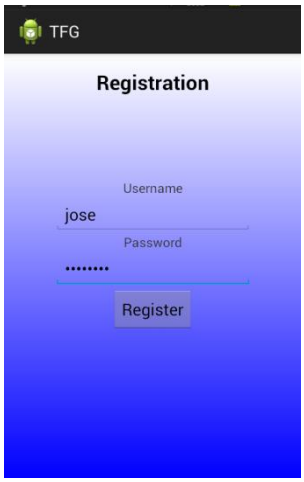

Figura 12. Pantalla de registro

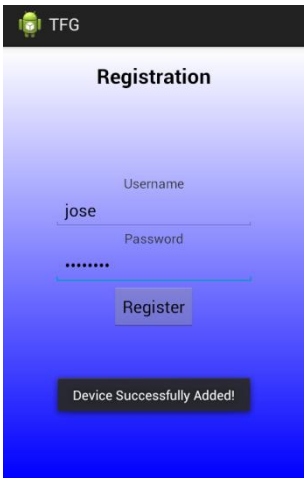

Figura 11. Registro completo

Desde la pantalla de login, pulsando la opción de Register, se accede a la pantalla que muestra la Figura 12. Una vez rellenados los datos del registro de manera correcta y perteneciendo a la red corporativa, si se pulsa el botón Register, se obtiene el mensaje de aceptación de la Figura 11.

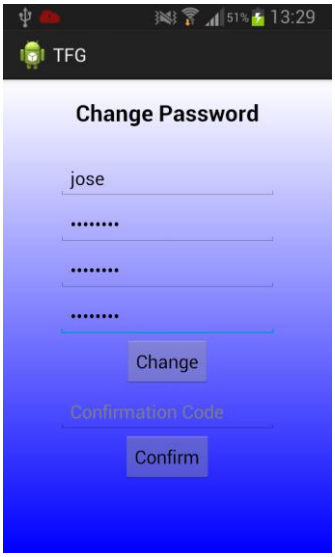

Figura 14. Pantalla cambio contraseña

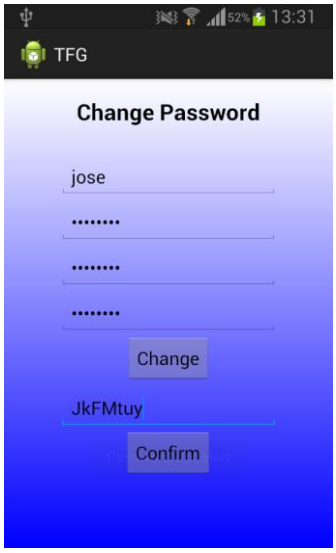

Figura 13. Pantalla cambio contraseña código

Desde el login, si se selecciona la opción de Cambiar contraseña, se accede a la Figura 14. En esta pantalla, el usuario deberá rellenar los datos con su usuario y contraseña antigua y la nueva contraseña. Una vez pulsado el botón Change, llegará una notificación como se observa en la imagen con el código de autenticación, el cual deberá ser introducido para confirmar el cambio de contraseña, como se observa en la Figura 13.

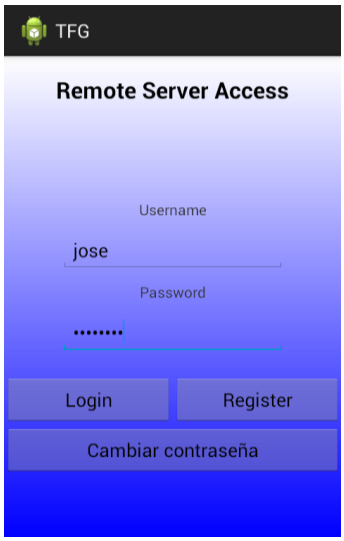

Figura 15. Pantalla de login completa

Una vez rellenados los datos del login, se accederá a la pantalla de confirmación, como se puede ver en la Figura 17.

Figura 17. Pantalla confirmación

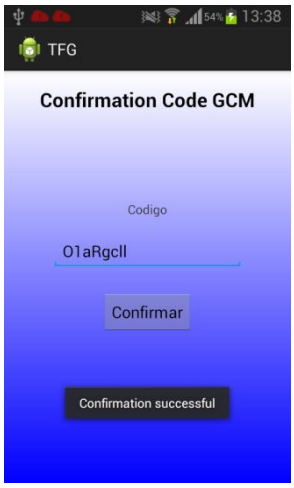

Figura 16. Pantalla confirmación código

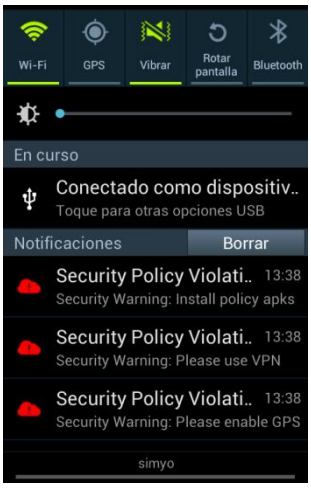

Figura 18. Notificaciones tras confirmación

Una vez en la pantalla de confirmación, el dispositivo recibe una notificación, como se puede ver en la imagen, que contiene el código de autenticación introducido en la Figura 16.

Tras realizar la confirmación, se reciben una serie de notificaciones relacionadas con la política de seguridad descrita en el Capitulo 4. En este caso, se reciben notificaciones debido a que la conexión VPN no está activada, el GPS está desactivado, como se puede ver en la Figura 18, y el antivirus no se encuentra instalado, de ahí la notificación solicitando que se instalen todas las aplicaciones de la política de seguridad.

Bibliografía

[1] "BYOD: Bring your own device." [Online]. Available: http://www.ibm.com/mobilefirst/us/en/bring-your-own-device/byod.html.

[2] "Mendeley." [Online]. Available: https://www.mendeley.com/.

[3] Wikipedia, "Bring your own device." [Online]. Available: https://en.wikipedia.org/wiki/Bring_your_own_device.

[4] TechTarget, "Mobile endpoint security: What enterprise infosec pros must know now." [Online]. Available: http://searchsecurity.techtarget.com/essentialguide/Mobile-endpoint-security-What-enterprise-infosec-pros-must-know-now.

[5] A. Technologies, "AVG." [Online]. Available: http://www.avg.com/.

[6] SAP, "SAP Afairia." [Online]. Available: http://go.sap.com/solution/platform-technology/enterprise-mobility-management.htm.

[7] M. Rouse, "bring your own apps (BYOA)," *TechTarget*, 2014. [Online]. Available: http://searchsecurity.techtarget.com/definition/bring-your-own-apps-BYOA.

[8] K. Root, "ConnectBot." [Online]. Available: https://connectbot.org/.

[9] K. Rhee, W. Jeon, and D. Won, "Security requirements of a mobile device management system," *Int. J. Secur. its Appl.*, vol. 6, no. 2, pp. 353–358, 2012.

[10] Pacosal.com, "ownMDM." [Online]. Available: http://ownmdm.pacosal.com/.

[11] Oracle, "Key and Certificate Management tool." [Online]. Available: https://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html.

[12] I. OpenVPN Technologies, "OpenVPN." [Online]. Available: https://openvpn.net/.

[13] MobileIron, “Mobile Device Management - MDM.” [Online]. Available: https://www.mobileiron.com/en/solutions/mobile-device-management-mdm.

[14] E. Y. and T. Hudson, “OpenSSL.” [Online]. Available: https://www.openssl.org/.

[15] Grover, J. (2013). Android forensics: Automated data collection and reporting from a mobile device. *Digital Investigation*, *10*, S12-S20.

[16] T. P. Group, “PHP.” [Online]. Available: php.net.

[17] E. Foundation, “Eclipse.” [Online]. Available: https://eclipse.org/.

[18] F.Martinez, “¿BYOD o CYOD? ¿Qué elegir?,” *CIO América Latina*, 2013.

[19] Elenkov, N. (2014). *Android Security Internals: An In-depth Guide to Android's Security Architecture*. No Starch Press.

[20] I. N. de T. de la Comunicación, “Política de contraseñas y seguridad de la información,” pp. 1–7, 2006.

[21] L. of the B. Castle, “Keytool.” [Online]. Available: http://bouncycastle.org/.

[22] Apache Friends, “Xampp,” 2015. [Online]. Available: https://www.apachefriends.org/.

[23] L. AirWatch, “AirWatch.” [Online]. Available: http://www.air-watch.com/.

[24] G. Developers, “Could Messaging.” [Online]. Available: https://developers.google.com/cloud-messaging/.

[25] P. Alvarado, “iOS y Android lideran la Cuota de Mercado,” 2015. [Online]. Available: http://www.ipadizate.es/2015/02/25/ios-android-lideran-cuota-mercado-smartphones-143391/.

[26] J. Vacca, *Computer and Information Security Handbook*. Morgan Kaufmann, 2009.

[27] R. Welton, “Remote Code Execution as System User on Samsung Phones,” *NowSecure*, 2015. [Online]. Available:

https://www.nowsecure.com/blog/2015/06/16/remote-code-execution-as-system-user-on-samsung-phones/.

[28] Anthony, "androguard," 2012. [Online]. Available: https://code.google.com/p/androguard/wiki/DatabaseAndroidMalwares.

[29] M. Davi, Lucas and Dmitrienko, Alexandra and Sadeghi, Ahmad-Reza and Winandy, *Privilege escalation attacks on android*. Springer, 2011.

[30] G. Drake, Joshua J and Lanier, Zach and Mulliner, Collin and Fora, Pau Oliva and Ridley, Stephen A and Wicherski, *Android Hacker's Handbook*. John Wiley \& Sons, 2014.

[31] Harvard, "MAKING BYOD WORK: BALANCING PRODUCTIVITY AND SECURITY," 2014.

[32] G. Thomson, "BYOD: enabling the chaos," *ScienceDirect*, vol. 2012, no. 2, pp. 5–8, 2012.

[33] T. P. Group, "Sanitize filters." [Online]. Available: http://php.net/manual/en/filter.filters.sanitize.php.

[34] VirusTotal, "VirusTotal." [Online]. Available: https://www.virustotal.com/en/documentation/mobile-applications/.

[34] and A. N. S. William Enck, Peter Gilbert, Byung-gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones." .

[35] D. Chin, Erika and Felt, Adrienne Porter and Greenwood, Kate and Wagner, "Analyzing inter-application communication in Android," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, ACM, 2011, pp. 239–256.

[36] V. Arroyo, David and Diaz, Jesus and Gayoso, "On the Difficult Tradeoff Between Security and Privacy: Challenges for the Management of Digital Identities," in *International Joint Conference*, Springer, 2015, pp. 455–462.

[37] R. Jakobsson, Markus and Shi, Elaine and Golle, Philippe and Chow, "Implicit authentication for mobile devices," in *Proceedings of the 4th USENIX conference on Hot topics in security*, USENIX Association, 2009, pp. 9–9. [Online]. Available: http://appanalysis.org/.

[38] RSA, "The evolution of siem," 2014.

Glosario

ADB Android Debug Bridge

BD Base de datos

BYOD Bring Your Own Device

CYOD Choose Your Own Device

Escalada de privilegios La escalada de privilegios se produce cuando un privilegio permite a un proceso realizar más tareas de las que debe hacer

Firewall Cortafuegos

GCM Google Cloud Messaging

GPS Global Positioning System

IDE Integrated Development Enviroment

MDM Mobile Device Management

NAT Network Address Translation

Rooteo Cambiar el estado de un dispositivo de usuario regular a usuario con privilegios y ventajas

SSH Secure Shell

SSL Secure Socket Layer

TLS Transport Layer Security

VPN Virtual Private Network