UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

PROYECTO FIN DE CARRERA

Ingeniería de Telecomunicación

Diseño de una plataforma para el control de tiempo de acceso a recursos en sistemas distribuidos

Autor: Javier López Andradas

Tutor: David Arroyo Guardeño

Diseño de una plataforma para el control de tiempo de acceso a recursos en sistemas distribuidos

AUTOR: Javier López Andradas

TUTOR: David Arroyo Guardeño

PONENTE: Francisco de Borja Rodriguez

Departamento de Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Julio 2016

Agradecimientos

Primero agradecer a David por haberme brindado la oportunidad de hacer este proyecto, dándome la idea y, por todos los consejos y ayuda que me ha aportado a lo largo de todo el proceso de realización del mismo.

Pero el mayor agradecimiento es para mis padres y hermanos por su cariño, su confianza, su paciencia y por preocuparse en que terminase el proyecto cuanto antes, no dejarlo para el final y hacerlo a toda prisa en el último momento.

Un especial agradecimientos a todos los amigos a los que he ido conociendo a lo largo de la carrera, ya que sin ellos no habría sido posible que pudiese conseguir llegar al final de la carrera.

Por último gracias a mis compañeros del trabajo por ser comprensivos con las jornadas, dándome de vez en cuando tiempo para poder hacer el proyecto, y no solo en mis ratos libres.

A Rubén por haber sido paciente conmigo y enseñarme los conceptos básicos de la programación en Android.

Abstract

The goal of this project is to propose a solution for the protection of critical information in platforms for sharing information assets.

In this project it will be used some tools that can help to protect the information, like encryption. In specific, it will be provided a set of means to endow the owners of documents with mechanisms to control who access those documents within an specific timespan.

In this project we will focus on the concept of documents life cycle to prevent non-authorized access. Therefore, the owner of document could use the proposed infrastructure to distribute it in a secure way and according to an access policy in a given time frame.

List of key words

Lifecycle, asymmetric cryptography, symmetric cryptography, flow, smartphone, security, digital rights management, hypertext transfer protocol secure, cookies, sharing policy.

Resumen

El objetivo de este proyecto es el proponer una solución al problema de la protección de información crítica en plataformas de colaboración de recursos compartidos.

Se utilizarán distintas herramientas de protección de datos, como el cifrado de los mismos, así como se le proporcionará al agente que comparte la información distintas maneras de impedir a otros agentes, que usan la plataforma de colaboración, acceder a la información compartida en la misma.

En concreto este proyecto se centrará en el uso del concepto Tiempo de vida de los documentos para la protección de los mismos, así como de proporcionar al agente la capacidad de gestionar quien puede acceder a dichos documentos de acuerdo a la política de colaboración que el dueño de la información haya proporcionado en una franja de tiempo establecida.

Lista de palabras clave

Tiempo de vida, criptografía asimétrica, criptografía simétrica, flujo, smartphone, seguridad, gestión de derechos digitales, protocolo seguro de transferencia de hipertexto, cookies, política de colaboración

Glosario

AES: Advanced Encryption Standard

CBC: Cipher-Block Chaining

DRM: Digital Rights Management

E2E: End to end

HTTP: Hypertext Transfer Protocol

HTTPS: Hypertext Transfer Protocol Secure

IOs: iPhone Operative System

OSI: Open System Interconnection

PDF: Portable Document Format

PHP: Pre Hypertext –processor

RFC: Request For Comments

SHA: Secure Hash Algorithm

SQL: Structured Query Language

SO: Sistema Operativo

SSL: Secure Sockets Layer

TEE: Trusted Execution Environment

TLS: Transport Layer Security

TV: Tiempo de vida

URL: Uniform Resource Locator

Índice

Índice de figuras

1- INTRODUCCIÓN

1- 1. Motivación

Hoy en día se comparte mucha información vía internet, debido a la gran expansión de los *smartphones*, el almacenamiento en la nube, la compartición de datos en las redes sociales y el *big data*; y sin embargo una vez que hemos compartido dicha información dejamos de tener control sobre ella, lo que puede llegar a dar ciertos problemas, como compartir información clasificada y/o personal y que los archivos acaben en manos no deseadas.

Este problema comienza con la elección de esta información clasificada, por ejemplo, documentos personales como DNI, pasaportes o, en caso de las empresas, documentos en relación a proyectos importantes que puedan comprometer el futuro de las mismas.

Una vez elegida la información que se quiere proteger se ha de tener un sistema de protección que no permita a otros usuarios de la red acceder a la información que no se haya compartido con ellos, para esto el usuario ha de seguir el principio de mínimo privilegio, que se basa en dar privilegios necesarios a un grupo de usuarios para que pueda trabajar sin problemas[1].

Existen varios métodos de protección de información dentro del marco de los DRM, *Digital Rights Management*, que son las tecnologías de control de acceso de los archivos para evitar que vulneren los derechos de autor [12]. Uno de los más claros ejemplos está en la protección de los archivos musicales o DVD en los que los fabricantes de los aparatos reproductores y los productores de los discos llegaban a un acuerdo de cifrado de los discos y los únicos capaces de descifrar tales discos eran los aparatos reproductores, evitando así que se copiase en sistema original.

Sin embargo, ¿Qué ocurre cuando en una organización, por ejemplo una empresa, se comparte un archivo con otra organización, por ejemplo un autónomo u otra empresa, durante la preparación de un proyecto o en una propuesta de negocio? Es, en este punto, donde se presenta el desafío para los DRM a la hora de tener un control a los recursos

[1] http://segurityinformationsmec3c.blogspot.com.es/2014/10/principio-del-privilegio-minimo.html

compartidos, la autorización de acceso a los archivos ya se ha garantizado previamente por lo que ya no podemos proteger esta información de esta manera; y el control de acceso, que se puede vulnerar ya que se han abierto el archivo compartido. Desde este proyecto se propone abarcar una nueva vía, el control del Tiempo de Vida, TV, de un archivo. El control del TV de un archivo es la fecha hasta la que un archivo puede ser leído de manera normal, es decir, no importa cuántas veces se haya compartido el archivo entre distintas personas, saltándose así la parte de control de la autorización de acceso y el control de acceso, este archivo dejará de ser legible para todo el mundo pasado el tiempo acordado por la organización que comparte la información, ganando así el nivel extra de control buscado.

1- 2. Objetivo

El objetivo de este proyecto es la implantación satisfactoria de una plataforma móvil, o aplicación, para la gestión de archivos y en concreto el acceso a los mismos.

Se ha escogido el formato PDF para desarrollar esta aplicación debido a que es el formato estándar que todos los dispositivos móviles pueden abrir, aumentando así el alcance que puede tener la información compartida, y no solo limitar el uso de esta aplicación a unos cuantos dispositivos o que el usuario deba tener una aplicación extra instalada para poder ver estos archivos.

Se escogió una plataforma móvil debido tanto al gran crecimiento de los *smartphones*, en España el 87% de los móviles son de este tipo[2], como a que incluían una pantalla integrada donde poder leer cómodamente estos archivos compartidos.

Se ha optado por Android debido a que aparte de haber un mayor número de dispositivos que usan este Sistema operativo (SO), aproximadamente 82.2% de los dispositivos móviles en 2014, tiene muchas facilidades a la hora de desarrollar una aplicación para este SO, IOs se descartó al no tener acceso a una máquina de desarrollo y Windows phone tiene muy pocos dispositivos en comparación a Android, tiene aproximadamente el 2,5% de los dispositivos móviles en 2014[3].

Otro objetivo de la realización del proyecto es la posibilidad de trabajar en un tema de gran actualidad y con una gran demanda de profesionales como la seguridad en Internet, así como el desarrollo de aplicaciones móviles, debido a que es un campo que no he visto durante toda la carrera y me puede beneficiar en el futuro.

[2] http://www.ditrendia.es/wp-content/uploads/2015/07/Ditrendia-Informe-Mobile-en-Espa%C3%B1a-y-en-el-Mundo-2015.pdf

[3] https://www.wayerless.com/2015/08/android-sigue-dominando-el-mercado-de-smartphones/

1- 3. Estructura de la memoria

El resto de la memoria estará estructurada de la siguiente manera, siguiendo los pasos que se realizaron para el desarrollo de todo el proyecto:

1. Estado del arte: Donde se expondrá la base desde donde se ha partido para la realización del proyecto.
2. Diseño: Donde se explica la definición que se realizó para el desarrollo de la aplicación. Especifica el alcance del proyecto, el funcionamiento, la arquitectura del proyecto, el desarrollo del servidor, junto con sus pruebas; el desarrollo del cliente, con sus pruebas modulares, y las pruebas completas de la aplicación.
3. Conclusiones y trabajo futuro: Donde se expondrá el resultado de la aplicación, si ha sido un éxito o un fracaso, así como las posibles mejoras que se podrán aportar en siguientes versiones.

2- ESTADO DEL ARTE

En la Sección 1 del Capítulo 1 de la memoria se ha hablado sobre la información compartida en una empresa, como por ejemplo en proyectos de colaboración, por lo que es importante conocer qué son los activos de información de una organización y lo que es más el establecimiento de una política de colaboración [11].

Antes de establecer una política de colaboración es necesario identificar los activos de la organización que en el caso que se quiere abarcar en este proyecto son tanto los datos, que se comparten mediante la aplicación, como los que tienen acceso a estos datos, los usuarios que usan la aplicación y los dispositivos donde se almacenan los datos.

Una vez establecidos los activos de la organización es necesario el establecimiento de unos grupos de acceso a la información compartida en grupos bien definidos, por ejemplo, en una empresa serían los departamentos como el de Recursos Humanos, el de Tecnologías de la Información, etc.

Finalmente, después de haber identificado cuales son los activos de la organización y de haber establecido los grupos de acceso a los mismos, queda definir la política de colaboración y acceso a la información. En este punto es importante recalcar que el usuario que decida usar la aplicación de seguridad tenga las herramientas necesarias para poder establecer una política de seguridad lo más flexible y cómoda posible. Existen varios puntos en una política de acceso a la información compartida, como por ejemplo:

- Definir un inventario de la información: Que sería establecer que información es la más crítica a la hora de ser compartida.
- Determinar la estructura de la organización: Que debería ser lo suficientemente abierta, como para que no haya problemas a la hora de compartir información, y suficientemente concreta, como para evitar que se comparta información no necesaria con un grupo de la organización.

- Establecer los permisos sobre la información: Que sería lo que puede hacer cada grupo de la organización con la información compartida con él.
- Establecer un protocolo sobre los permisos: Que manejase si a un grupo se le decide quitar o añadir más permisos de los que tiene, así como con qué frecuencia los cambia.

Un punto importante a tener en cuenta sería la necesidad de seguir el denominado principio de mínimo privilegio, que es otorgarle a cada grupo lo mínimo para poder trabajar día a día.

Otro punto que se ha comentado en la motivación del proyecto es que hoy por hoy se comparten muchos documentos de vital importancia mediante Internet por lo que en los últimos años se han centrado en el desarrollo de protocolos de seguridad, como el protocolo Hypertext Transfer Protocol Secure (HTTPS), basado en el protocolo SSL [13]. Este protocolo surgió de la necesidad de poner una capa de seguridad en la transmisión de los datos, este protocolo esta implementado en la capa de Aplicación del modelo OSI, *Open System Interconnection*, que es el modelo estándar para la comunicación entre redes y está dividido en varias capas siendo la primera la capa de Aplicación que la encargada de definir los protocolos que se usan en las distintas aplicaciones que comunicas las redes[4].

EL protocolo HTTPS no es más que el mismo protocolo Hypertext Transfer Protocol (HTTP), que es el protocolo usado para mandar información de un dispositivo que llamaríamos cliente y otro que se denominaría servidor, pero en versión segura y esto se consigue gracias a que se usa un canal autenticado y cifrado.

Antes de hablar sobre el proceso de establecimiento de una conexión segura hay que definir dos conceptos claves para comprender este tipo de conexiones. El primero de estos conceptos es la **criptografía simétrica** [14], este tipo de criptografía se basa en cifrar los datos siguiendo un algoritmo y una clave para más adelante obtener los datos originales usando la misma clave que se usó para cifrarlos. El segundo concepto es el

[4] https://es.wikipedia.org/wiki/Modelo_OSI

de **criptografía asimétrica** [14], en este tipo se usan dos claves, comúnmente conocidas como clave pública y clave privada, y se basa en el cifrado de los datos mediante un algoritmo con una de las dos claves para más adelante obtener los datos originales usando la otra clave. Una vez explicado los dos tipos de criptografía podemos dar paso a explicar el canal de cifrado usado en el protocolo HTTPS.

El protocolo HTTPS [16] fue creado por Netscape Communications en 1994 para su navegador Netscape Navigator y estuvo construido sobre el protocolo SSL, que más tarde evolucionó en el protocolo Transport Layer Security (TLS).

El canal cifrado se basa en el cifrado de todos los mensajes compartidos entre el servidor y el cliente usando criptografía simétrica, el funcionamiento de este protocolo HTTPS [7] funciona de la siguiente manera:

1. El servidor ha de tener una clave pública y otra privada, en este caso asimétrica.

2. El cliente emite una petición al servidor de establecimiento de una conexión segura.

3. El servidor envía su certificado digital, o certificado SSL en el que incluye la clave pública del mismo.

4. El cliente verifica el que certificado sea de verdad de confianza y si es así manda la confirmación al servidor y una clave simétrica cifrada con la clave pública del servidor.

5. El servidor descifrará la clave simétrica y el servidor y el cliente podrán empezar a comunicarse de forma segura.

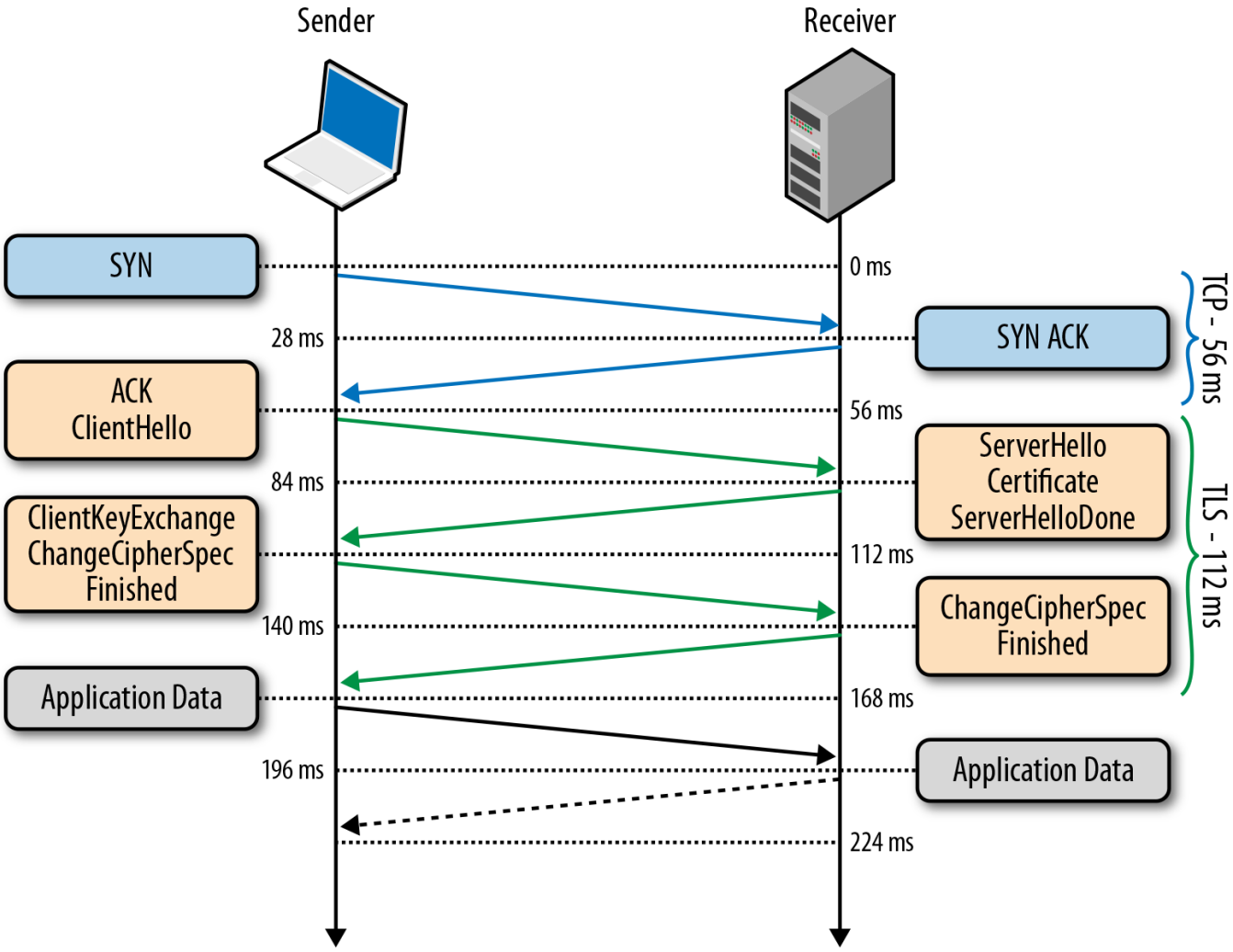


Figura 1: Protocolo HTTPS[5]

Adicionalmente el servidor se ha desarrollado para que trabaje con consultas, o *queries*, preparadas; este tipo de consultas son muy útiles por dos razones, la primera porque el servidor solo necesita compilar las consultas una vez, una vez hecho no tiene que volver a compilarlas y prepararlas para su ejecución, ahorrando recursos y haciendo que la aplicación sea más rápida; la segunda porque de esta manera protegemos la base de datos frente a ataques por inyección SQL, que no es nada más que añadir código SQL a una consulta, haciendo que el servidor pueda ejecutar instrucciones distintas a las esperadas [8].

[5] http://serverfault.com/questions/570387/https-overhead-compared-to-http

Para la seguridad es muy importante que la encriptación de los datos sea robusta y, sobre todo, que nadie sin la llave pueda “abrir” nuestros datos generados. A lo largo del proyecto se hablará mucho sobre esta parte del cifrado de los datos, ya que la seguridad de los sistemas de información depende de ello en gran medida.

El cifrado de los datos no es más ni menos que la generación de datos nuevos a partir de los originales usando un algoritmo de encriptación, de tal manera que usando el algoritmo inverso se pueda recuperar la información previamente cifrada.

El algoritmo usado para la encriptación de los archivos PDF del proyecto es el algoritmo AES. Este algoritmo es el estándar actual de cifrado simétrico, y fue definido en un principio en 1997 y aprobado a nivel mundial en 2002, tras una exhaustiva batería de pruebas y de estudio de diferentes algoritmos de encriptación. Este algoritmo se basa en el cifrado de bloques de datos, en nuestro caso de 128 bits, mediante la permutación de los mismos, el uso de una serie de tablas predefinidas en el algoritmo y la XOR de la clave de cifrado y descifrado que ha de tener el mismo número de bit que el tamaño del bloque.

El funcionamiento del algoritmo AES funciona de la siguiente manera [10]:

1. Se dividen los datos a tratar en matrices de 128 bits, es decir en un *array* de 4x4 bytes.

2. A cada byte del *array* se le realiza la operación XOR con el byte correspondiente de la clave de cifrado, a esta operación se le denomina AddRoundKey.

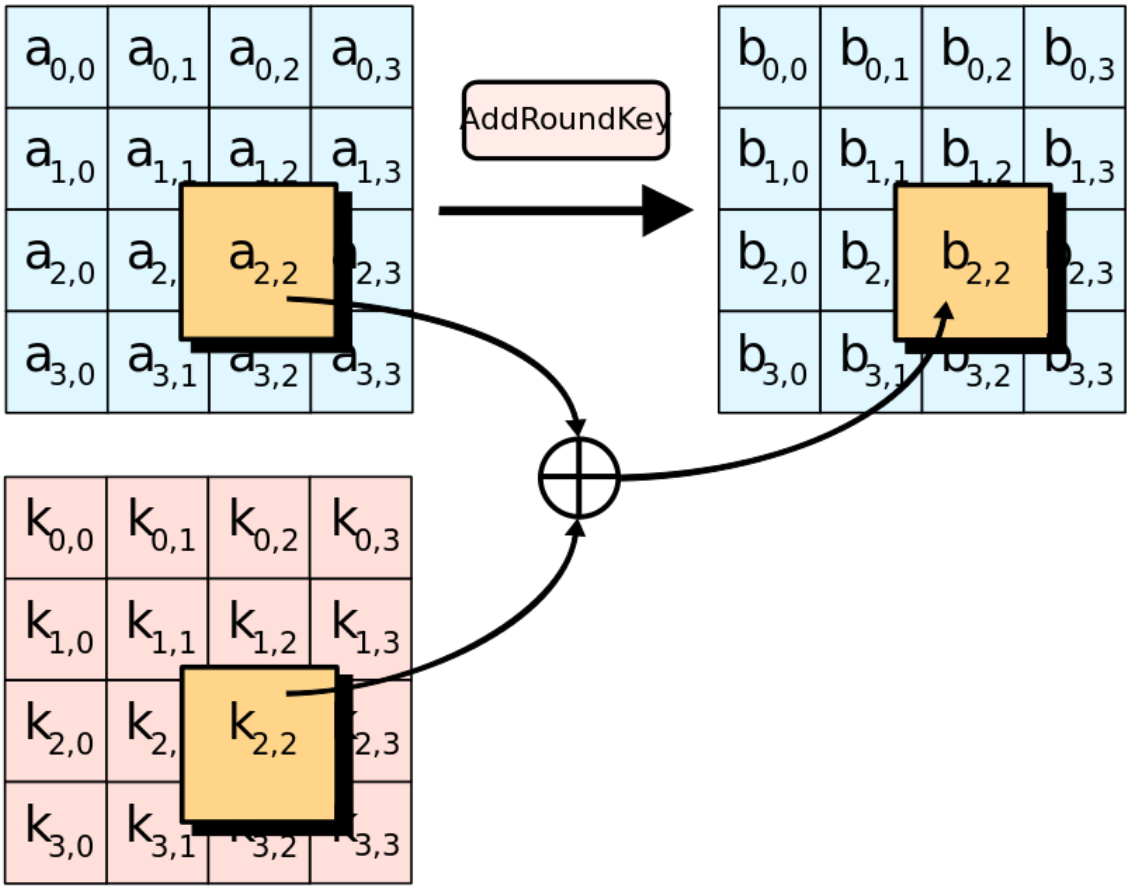


Figura 2: AddRoundKey (Fuente [10])

3. Se sustituyen los bytes conseguidos en la operación por otros con respecto a los valores definidos en una tabla de búsqueda estándar, a esta operación se le denomina SubBytes.

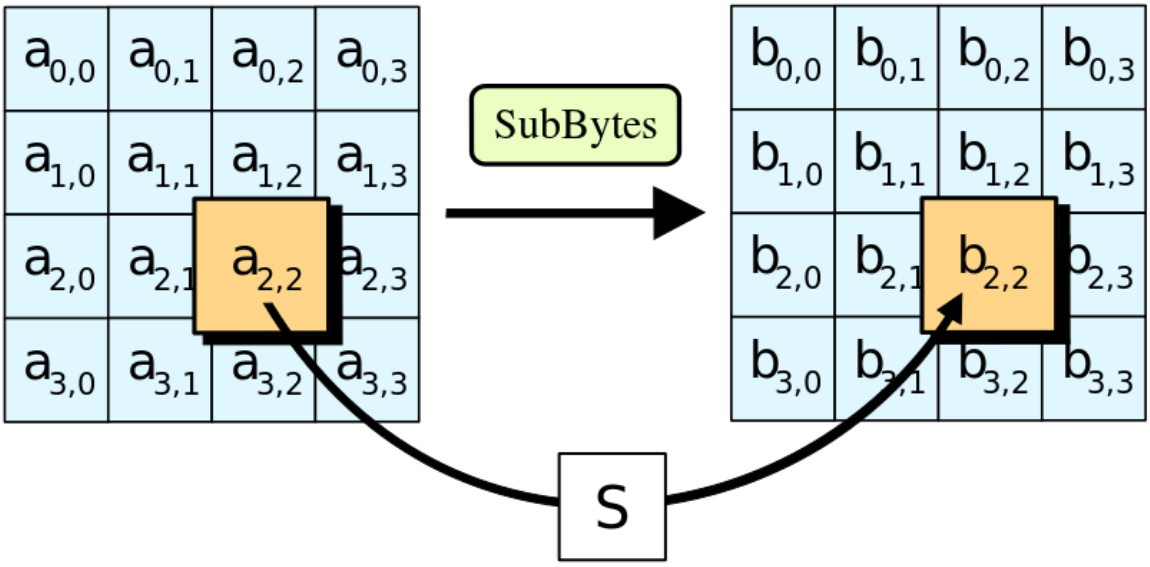


Figura 3: SubBytes (Fuente [10])

4. Se realiza una transposición, es decir, se desplazan los bytes de cada fila un número determinado de veces, siendo el valor de los desplazamientos distintos en cada fila, a esta operación se llama ShiftRows.

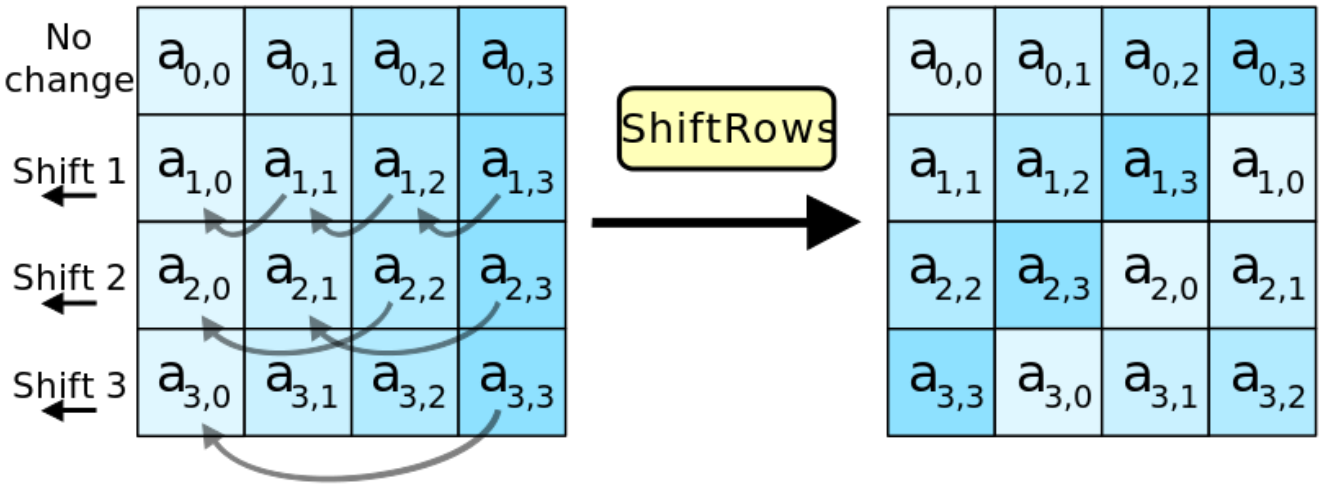


Figura 4: ShifRows (Fuente [10])

5. Se operan los bytes de cada columna usando un polinomio constante, este proceso se denomina MixColumns.

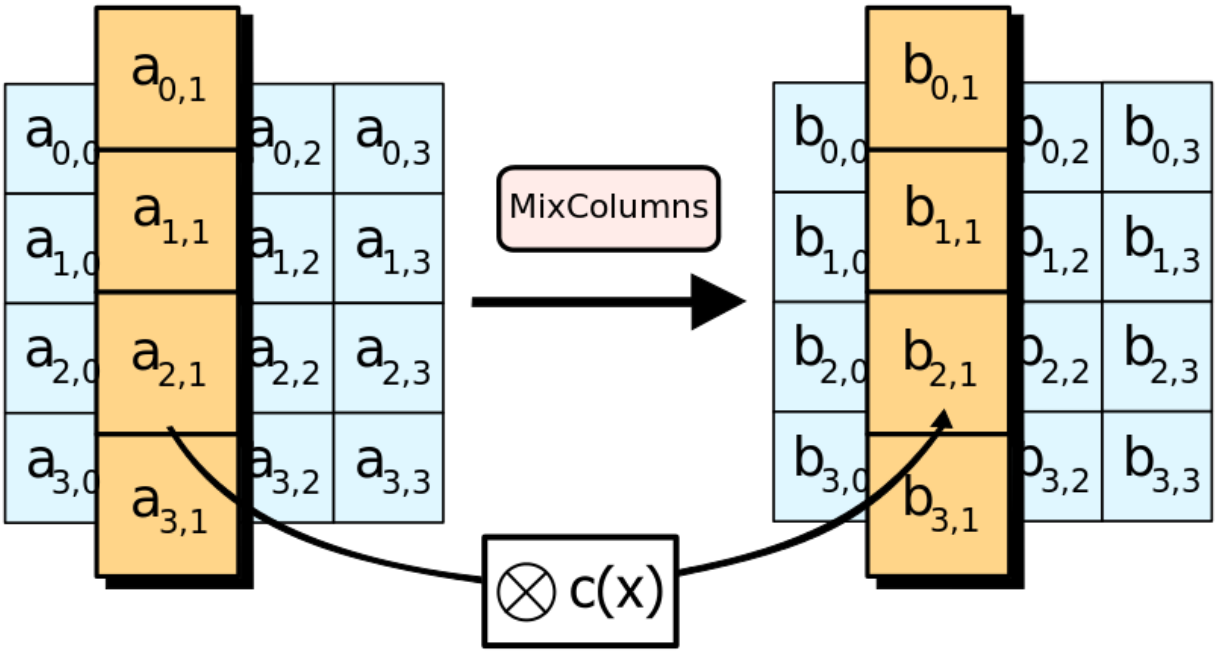


Figura 5: MixColumns (Fuente [10])

6. Se repite la operación AddRoundKey con una clave generada a partir de la clave original.
7. Se repiten las operaciones SubBytes, ShiftRows, MixColumns y AddRoundKey tantas veces como se necesite.

Este algoritmo se ha utilizado usando el modo CBC [9], Cipher-Block Chaining, el cual está basado en realizar una operación XOR con el bloque de datos cifrado anterior, esto sirve en el caso del cifrado de documentos con cabeceras constantes en todo el documento aparezca siempre los mismos datos cifrados en partes concretas del mismo, también es útil para cifrar las imágenes debido a que los bordes de las mismas al ser constantes serian el mismo problema que las cabeceras de los documentos.

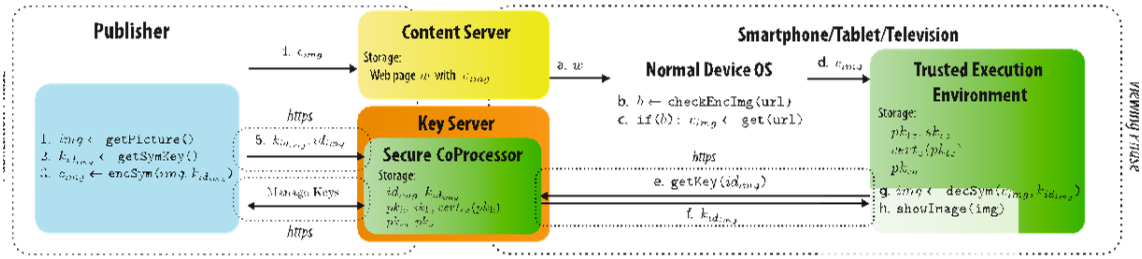
Adicionalmente al algoritmo de cifrado se le ha añadido un *padding*, o relleno, PKCS5, este algoritmo rellena los bloques que se queden incompletos, después de la encriptación, con datos para que se alcance el tamaño de bloque establecido en la encriptación, por ejemplo, si en un determinado bloque faltasen 3 bytes por rellenar este algoritmo los rellenaría los tres con el valor 3, es decir, con la cadena de bits 00000011.

También se ha usado el algoritmo SHA-512[6] para obtener el hash de las contraseñas guardadas en la base de datos del servidor. Este algoritmo, a diferencia de los algoritmos de encriptación previamente mencionados, no trata de recuperar los datos originales, sino que lo que genera un código, a partir de la contraseña proporcionada por el usuario y una clave aleatoria, con una longitud específica de 512 bits. De esta manera si alguien consiguiese entrar en la base de datos y ver las contraseñas no le serviría de nada, ya que lo que está guardado en la base de datos es este hash.

Como se ha indicado en la motivación del proyecto, toda este cifrado y preparación de capas de seguridad no sirve de nada si el usuario ya puede acceder a la información descifrada y no tenemos la capacidad de bloquearle el acceso a ella, de ahí que se haya realizado la aproximación

[6] https://es.wikipedia.org/wiki/SHA-2

mediante el uso del TV de los documentos, sin embargo esta no es la primera aproximación al uso de los TV de los archivos. Este proyecto ha usado como referencia el proyecto de X-pire 2.0 [1] que era usado no para asegurar archivos en PDF sino archivos de tipo de imagen, este proyecto fue llevado a cabo de manera cooperativa por las universidades de Saarland y Ruhr de Bochum, junto al Max Planck Institute for Software Systems y trataba de , como ya se ha comentado, de implementar la



solución del TV en imágenes compartidas mediante el uso de las redes sociales sin tener que usar un sistema externo al de la misma red Social.

Figura 6: Arquitectura X-pire 2.0 (Fuente [1])

Este proyecto se basaba en tener una clave simétrica, explicada anteriormente en este mismo punto de la memoria, almacenada en un servidor, y cuando el cliente quisiese ver la imagen cifrada recibir la clave de la imagen, generada para esta imagen únicamente, comprobar si la fecha de visualización permitía verla y, en caso de poder ser una fecha válida, mandar la clave para que el cliente pudiese descifrar la imagen.

Esta solución, como se ha comentado en la motivación, pertenece a los sistemas DRM de protección de datos, y no es la primera aproximación que usa el concepto de TV. Un ejemplo sería la propuesta de Drumlin[7] también para PDF, el cual sigue una estructura parecida a la del proyecto X-Pire 2.0 [1] desde la que se partió para realizar el proyecto. Este concepto de TV se ha estado usando desde el "boom" de la comunicación vía Internet debido a la gran capacidad de control sobre los documentos compartidos, ya que con la globalización es poco probable que una persona no tenga acceso a Internet para compartir la información, y con este método y sin necesidad de compartir una *password* previa para

[7] http://www.drumlinsecurity.com/publisher.html

descifrar el documento previo puedes garantizar cuando un documento deja de ser visible o no.

El TV no es lo único que se puede controlar gracias a aplicaciones sobre archivos, por ejemplo, la solución implementada por OpenText[8], la cual no solo controla quien puede acceder a los archivos sino que permisos tiene sobre éste el usuario que accede al mismo, por ejemplo, si trata de imprimir el archivo.

Aparte de las aplicaciones existentes para evitar que la información se divulgue también existen soluciones como la de Locklizard[9] , que entre otras soluciones propone el uso de marcas de agua generadas aleatoriamente, lo que hace más difícil su eliminación, para poder hacer un seguimiento de la documentación compartida.

Otro tipo de solución al problema de la divulgación de documentación es la posibilidad de que en el momento de compartir dicha documentación, vía Internet, un tercero pueda acceder a dicha documentación. En este caso la propuesta de eGarante[10] es de hacer un seguimiento total del archivo en el transcurso de la colaboración, desde que se elige el destinatario, mediante un formulario, hasta que el receptor rellena el formulario de respuesta con la confirmación de recepción de la documentación.

Por último, el SO para el que se ha desarrollado la aplicación es Android, que es un sistema basado en el núcleo Linux, especialmente desarrollado para su uso en teléfonos inteligentes, o *smartphones*, cuyo nacimiento data del 2007, aunque el primer Smartphone con este sistema operativo nació en 2008. Originalmente fue desarrollado por la empresa Android Inc. pero en 2007 fue comprada por Google.

[8] http://www.opentext.com/what-we-do/products/enterprise-content-management/content-management/opentext-rights-management
[9] http://www.locklizard.com/solutions/
[10] https://www.egarante.com/eg-doc/

La arquitectura de Android está dividida en las siguientes áreas, tal y como se muestra más adelante en la Figura 3:

- Kernel de Linux: Es la capa más baja del software de Android, es la encargada de gestionar los recursos del smartphone y del sistema operativo. El desarrollador no accede directamente a ella si no que se apoya en las librerías.
- Bibliotecas: Son nativas de Android y están escritas en C o C++ y están compiladas para el hardware específico de cada Smartphone. Están diseñadas para llevar a cabo las tareas que se repiten más comúnmente de manera más eficiente.
- Marco de aplicaciones: Está formada por las clases y servicios que usan las aplicaciones para realizar sus funciones.
- Aplicaciones: Es la última capa en las que se incluyen todas las aplicaciones del dispositivo.
- Entorno de ejecución: Este no está definido como una capa en sí misma, si no que pertenece a la capa de las Bibliotecas, o *Library*, está formado principalmente por una máquina virtual ART, para versiones 4.4, Android KitKat, y posterior; o por una máquina virtual Dalvik, desde la versión 2.2, Android Foyo, hasta la versión 5.0, Android Lollipop, en la que desaparece completamente; que codifica las aplicaciones en Java y la gran ventaja que tiene es que solo hace falta compilarlas una vez para poder distribuirlas libremente.

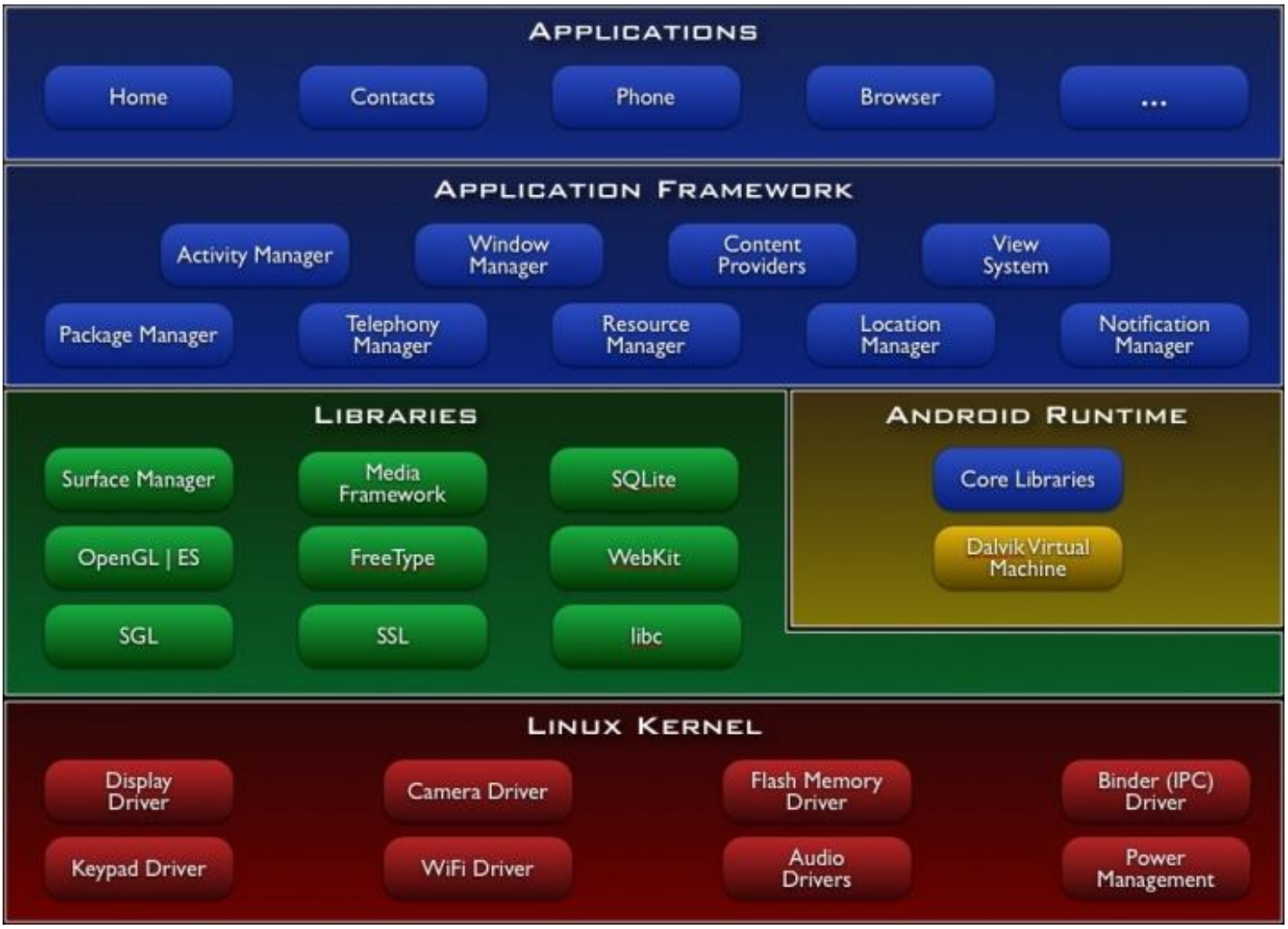


Figura 7: Arquitectura Android[11]

[11] http://picandocodigo.net/2007/android-la-nueva-plataforma-de-desarrollos-moviles/

3- DISEÑO

3- 1. Definición del alcance del proyecto

Para la realización de este proyecto se ha tenido en cuenta únicamente la funcionalidad de la solución, obviando en parte la estética de la misma.

Dentro del alcance se sitúa la definición de la arquitectura de la aplicación, así como los algoritmos usados para el cifrado, tanto para los datos personales de los usuarios como para los archivos PDF que se usan en la misma.

Dentro de la seguridad está incluida la seguridad a nivel de comunicación entre el servidor y el cliente, así como un control de los usuarios *logados* en la aplicación.

Asimismo se le proporcionará al usuario la posibilidad de desarrollar su propia política de seguridad de una manera flexible.

No se ha tenido en cuenta la seguridad a nivel de Hardware de la aplicación, es decir, la seguridad del dispositivo móvil con acceso a la aplicación debido a que para dispositivos Android no existe un procesador que envíe las señales a través de su Hardware de manera cifrada, a este tipo de seguridad se le denomina TEE, Trusted Execution Environment, del cual se hablará más adelante en el Capítulo 4 de la memoria.

3- 2. Definición del funcionamiento

Una vez definido el alcance del proyecto se estudió la funcionalidad que iba a tener, teniendo como base el proyecto del Xpire 2.0 [1] de las imágenes la idea fue una aplicación en la que se abriese el archivo PDF encriptado para su visualización.

Para el usuario final la aplicación solo se usaría en el móvil, es decir nunca se le mandaría a una página web externa a la aplicación. Esto se eligió así debido a que solo queríamos que el usuario usase una plataforma, es decir, que usase una única aplicación y no saliese de ella para ver el PDF.

A fin de establecer ofrecer una política de seguridad flexible al usuario se le ofrecerá la posibilidad de crear organizaciones jerárquicas, tantas como quiera, con las cuales ofrecer a los usuarios destino la posibilidad de poder leer los archivos.

Los pasos para usar la aplicación serían los siguientes:

1. Darse de alta como usuario de la aplicación, aquí se le pediría un correo electrónico, un nombre de usuario y una *password*.
2. Una vez el usuario ya está creado, tendrá que hacer *LogIn*, o entrar a la aplicación.
3. El usuario creará una Organización en la que añadirá los roles que quiera a ella. Una vez que haya creado los Roles y tenga la Organización a su gusto añadirá usuarios que ya estén dados de alta en la aplicación a los Roles que él quiera.
4. Finalmente seleccionará un archivo PDF de su dispositivo móvil, le pondrá una fecha límite para poder ver el archivo y lo compartirá con un Rol de la aplicación. Hay que tener en cuenta que los miembros del Rol y todos los que estén por encima podrán ver el archivo.
5. Una vez tenga el fichero encriptado el usuario lo compartirá con quien desee de manera común, es decir terceros que no controla esta aplicación.

3- 3. Definición de la arquitectura

Una vez definido el comportamiento funcional se procedió a la definición de la parte de arquitectura que daría soporte a la aplicación.

Al principio del proyecto se definieron tres posibles arquitecturas de la aplicación:

1. Solo plataforma móvil, es decir sin gestión de ningún agente externo.
2. Una arquitectura cliente-servidor único.
3. Una arquitectura de cliente-servidor + almacenamiento externo.

La primera requería un menor esfuerzo de planificación y desarrollo debido a que únicamente íbamos a tratar con la aplicación móvil, sin necesidad de enviar datos a ningún agente externo. Se descartó debido a que para configurar el TV del archivo se requería guardar información clave sin encriptar dentro del mismo archivo, haciendo a este vulnerable a cambios de cifrado en sus datos.

En cuanto a la segunda y tercera opción la base era la misma, una arquitectura de una aplicación móvil, o cliente, y un ordenador que hiciese de servidor. La diferencia entre ellas es que mientras la primera seria el servidor mismo quien almacenase los archivos, por lo que no haría falta la encriptación de los mismos dentro del cliente, la segunda el servidor seria únicamente el que almacenase las claves de encriptado de los archivos y las proporcionase si fuese necesario.

Como solución final se eligió la tercera debido a que se quería evitar el exceso de información en el servidor, es decir, que llegado un momento habría que aumentar la capacidad del servidor, mientras que, al solo almacenar las claves de los archivos se requeriría mucho más tiempo y archivos protegidos para que esta situación se produjese; así como para archivos muy grandes el tiempo de envío de los datos era muy grande, el servidor está alojado en una red lenta, otro último punto que se tuvo en cuenta fue la dificultad de la implementación de esta solución y, finalmente, debido a que la seguridad de esta solución era menor ya que

requería, una vez que se abriese el archivo en el móvil, guardarlo en el mismo dispositivo, haciendo posible su distribución y haría inviable la solución propuesta, esta arquitectura es la misma que la del proyecto X-Pire 2.0 [1].

Una vez establecida la arquitectura de la aplicación se pasó a definir a las capas de seguridad de la misma.

- Se requería una comunicación cifrada entre el cliente y servidor por lo que se estableció en la arquitectura que se comunicasen mediante el protocolo HTTPS.
- Evitar enviar mensajes a través de ataques de usuarios fuera de la aplicación, es decir, que un usuario pueda acceder a información clave de la aplicación mediante envío de mensajes correctos, para evitar esto se definirá un parámetro de sesión para los usuarios mediante el uso de *cookies* [17].
- Usar un algoritmo de encriptación lo suficientemente robusto como para evitar el descifrado de los archivos mediante ataques, se decide usar un algoritmo de encriptación *AES/CBC/PKCS5Padding* explicado en el Capítulo 2.
- Se le daría al usuario el poder de decidir con que demás usuarios decide compartir los archivos, es decir, que si el usuario decide compartir un archivo con otro usuario, este último podría compartir el archivo con otro fuera del control del primero y podría ver el archivo siempre y cuando estuviese dentro del TV definido por el primer usuario. Para solucionar esta disyuntiva se eligió que el usuario tendría que compartir los archivos con usuarios dados de alta dentro de la misma aplicación. Para hacer esta solución más amena se utilizará una estructura jerárquica para compartir los archivos.

Terminada la definición de las capas de seguridad se procedió a definir la base de datos del servidor para almacenar la estructura que se requería para la aplicación.

La base de datos está constituida por las siguientes tablas, y de acuerdo con el siguiente diagrama Entidad-Relación:

- Usuario (Email, userName, pass).
- Organización (Nombre, IdOrg, Email).
- Rol (Nombre,IdRol,↑Organización,↑RolPadre).
- UserRol (↑EmailUser,↑RolOrg).
- Archivo (Id, clave, fecha, ↑Rol,↑Email).

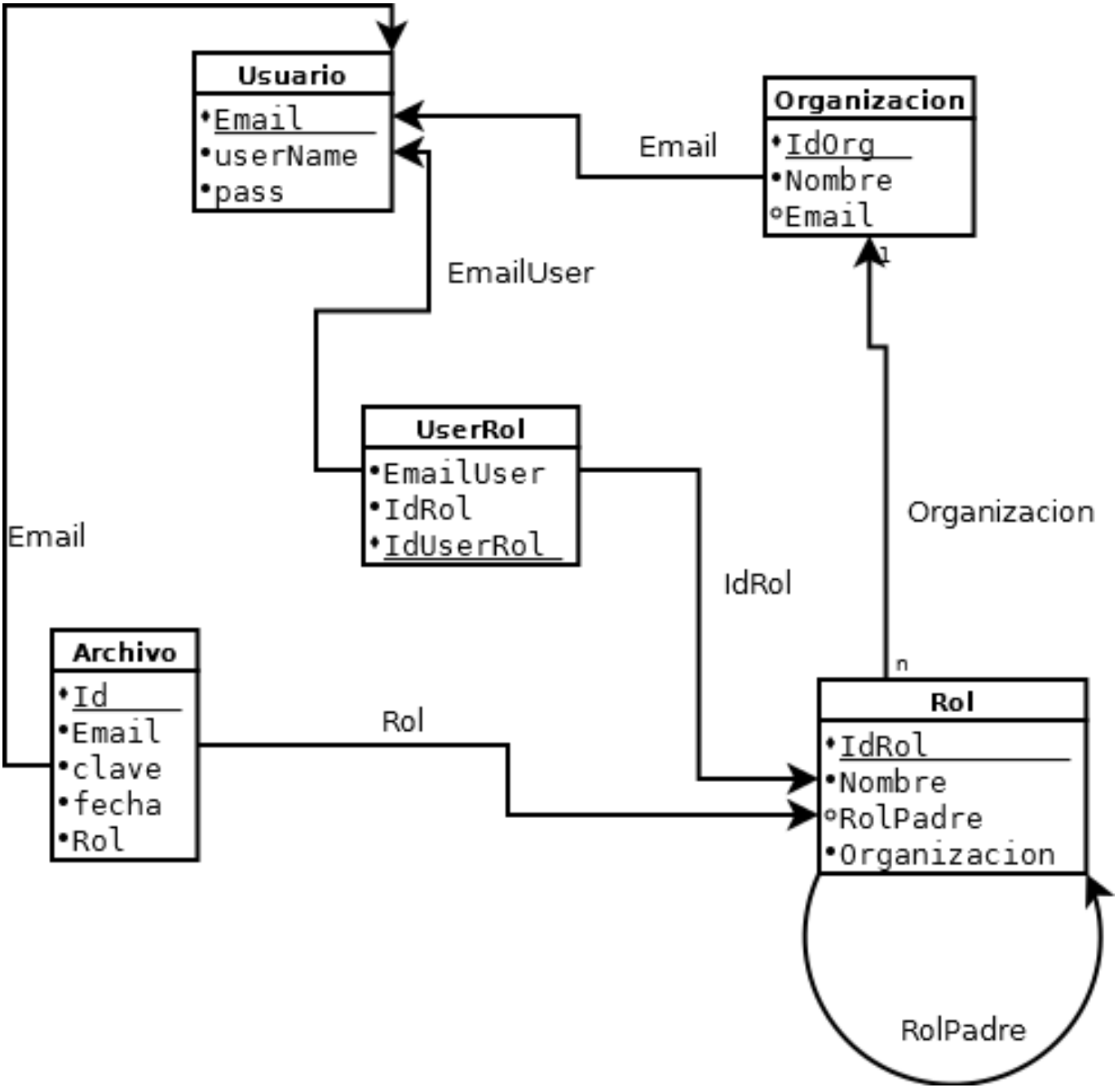


Figura 8: estructura base de datos

Se definió como Id del usuario el correo electrónico del mismo, evitando al usuario la necesidad de buscarse un nombre usuario único teniendo en cuenta que el correo electrónico de cada persona es único.

Como Id de la Organización se usó la concatenación del correo electrónico del Usuario que crea la organización y el nombre de la Organización, estableciendo de esta manera la limitación de que el usuario esté vinculado a de una única organización con el nombre que el elija.

Para el Rol se eligió como identificador el Identificador de la Organización concatenado con el nombre del Rol, evitando así Roles con el mismo Nombre dentro de una misma Organización. Para conseguir una organización jerárquica se estableció un Rol padre dentro de cada Rol, este sería proporcionado por el usuario dueño de la Organización, salvo por el primer Rol que se genera con la creación de la Organización, este Rol tendrá como nombre Admin y no tendrá padre.

La tupla intermedia que une Rol con Usuario es userRol, esta relaciona directamente un usuario a un Rol, en la creación de una Organización se crea por defecto una que une el Rol original Admin con el usuario que creó la organización. El identificador del UserRol es la concatenación del Id del Rol con el correo electrónico del usuario. Se estableció la limitante de que un usuario solo pudiese estar una vez por Organización, evitándose así malfuncionamiento de la aplicación.

Finalmente Archivo define la información necesaria para el archivo, el Id es calculado a partir del Hash del archivo cifrado, se explicará más adelante en la Sección 5.5 del Capítulo 3 y está relacionado con el usuario a través del correo electrónico y del Rol a través de la clave de nombre Rol.

Como punto extra, el servidor se desarrollará con consultas preparadas para evitar la inyección SQL [8].

Después de definir la base de datos quedó en definir el tipo de protocolo HHTP que se usaría para para comunicación de mensajes entre el cliente y el servidor, esta decisión fue sencilla y clara, se decidió usar el

método POST, ya que no manda en la URL la información que se requiere ni esta se guarda en el log.

En adición a la arquitectura decir que se ha llevado un control de versiones mediante GIT, alojando el repositorio en internet en bitbucket[12].

[12] https://bitbucket.org/

3- 4. Desarrollo del servidor

El servidor se programó en el ordenador facilitado por la UAM en el laboratorio, un Intel CORE i5 con 4 GB de RAM y 500 GB de disco duro y el OS Ubuntu, donde se realizó la aplicación; para desarrollar un servidor que proporcionase las características que se impusieron en la definición de la arquitectura se definió el servidor como un servidor Apache.

Una vez instalado Apache explicado en la página web de apache Friends [4] en el ordenador se inició su configuración, lo primero que se hizo fue configuración la de la base de datos, definir las carpetas, así como los usuarios, en este caso usuario; que podría modificar los archivos guardados; donde se guardarían los *scripts* para el funcionamiento de la aplicación.

Los *scripts* usados para el funcionamiento de la aplicación han sido desarrollados usando PHP apoyándonos en el libro Desarrollo web con PHP y MySQL [2]y usando ejemplos extraídos tanto de la página web de php.net [6] como de la página web stackoverflow [5], se usó una carpeta dentro de las carpetas donde estaban almacenados los *scripts* principales de la aplicación en la cual se incluyeron los métodos usados para el acceso a la base de datos, el control de cookies y el control de acceso a archivos por el sistema de la organización.

Después de esta parte se pasó a definir la seguridad para que el servidor pudiese funcionar como un servidor que usase HTTPS, para ello hubo que configurar el puerto en el cual se conectaría el cliente mediante este protocolo, el 443, y seguidamente se pasó a crear el certificado de seguridad, que en nuestro caso al no contar con presupuesto para que lo certificase un organismo oficial fue “autofirmado”, esta opción es viable para el desarrollo y las pruebas de funcionamiento, una vez que la aplicación se requiera explotar de manera comercial se requerirá un certificado firmado por una entidad certificadora.

Para conseguir estos certificados se siguieron los siguientes pasos explicados en la página web de apache friends [4]:

1. Instalar openssl con el siguiente comando:
   sudo apt-get install openssl

2. Generar una clave de 1024 bits:
   sudo opnessl genrsa -des3 -out servidor.key 1024

3. Proporcionar al certificado la información necesaria: Nombre del país, provincia, localidad, nombre de la organización, departamento de la organización, nombre del dominio y correo electrónico.
   sudo openssl -req -new -key servidor.key -out servidor.csr

4. El siguiente paso sería el que una entidad certificadora tuviese que realizar, pero como hemos comentado antes lo autofirmaremos:
   openssl x509 -req -days 365 -in servidor.csr -signkey servidor.key -out servidor.crt

5. Finalmente pondremos las claves del certificado en la carpeta correspondiente y habilitaremos en el servidor el uso del puerto 443 y la capacidad del servidor de usar el protocolo SSL.

3- 4.1. Registro de usuarios

La primera vez que un usuario accede al servidor tiene que darse de alta, de lo contrario no podrá hacer nada. Al servidor le llegará como cabeceras de la llamada, del protocolo https, el correo electrónico del usuario, el nombre de usuario con el que este quiera que se le dirija la aplicación así como una *password* que él elija. Lo primero que hace el servidor es comprobar si existe un usuario con ese correo, de existir devuelve un mensaje de error, si no existe guarda en la tupla de la base de datos el correo electrónico y el nombre del usuario, así como la contraseña cifrada con el algoritmo de hash *SHA-512,* quedando así la *password* cifrada y evitando que en posibles ataques a la base de datos se puede acceder a las contraseñas de los usuarios de manera directa.

3- 4.2. Login

Cuando el usuario quiera hacer *Login* le llegará al servidor el correo electrónico del usuario y la *password*, el servidor primero comprobará si el correo electrónico existe dentro de la base de datos, si existe calculará el *SHA-512* de la *password* recibida y la comprobará con la extraída de la base de datos, si coincidiese con la del correo electrónico enviaría un mensaje de éxito y generaría las *cookies* necesarias para guardar su sesión, si no cumpliese ninguna de las dos condiciones anteriores se le mandaría un mensaje de error al hacer *Login*, en el cual no especifica si está mal el correo electrónico o la contraseña si no un mensaje genérico.

Como adición y para hacer frente a ataques por fuerza bruta se añadió una tupla a la base de datos en la que se guardan los intentos de acceso a la aplicación erróneos por correo electrónico, si en menos de 5 minutos se ha intentado acceder más de 5 veces de manera errónea, se mandaría un mensaje de bloqueo y un usuario con ese correo no podrá conectarse en la próxima media hora.

3-4.3. Organización

Para conseguir un control sobre el acceso de los Archivos compartidos se ha decidido seguir un modelo de una Organización jerárquica.

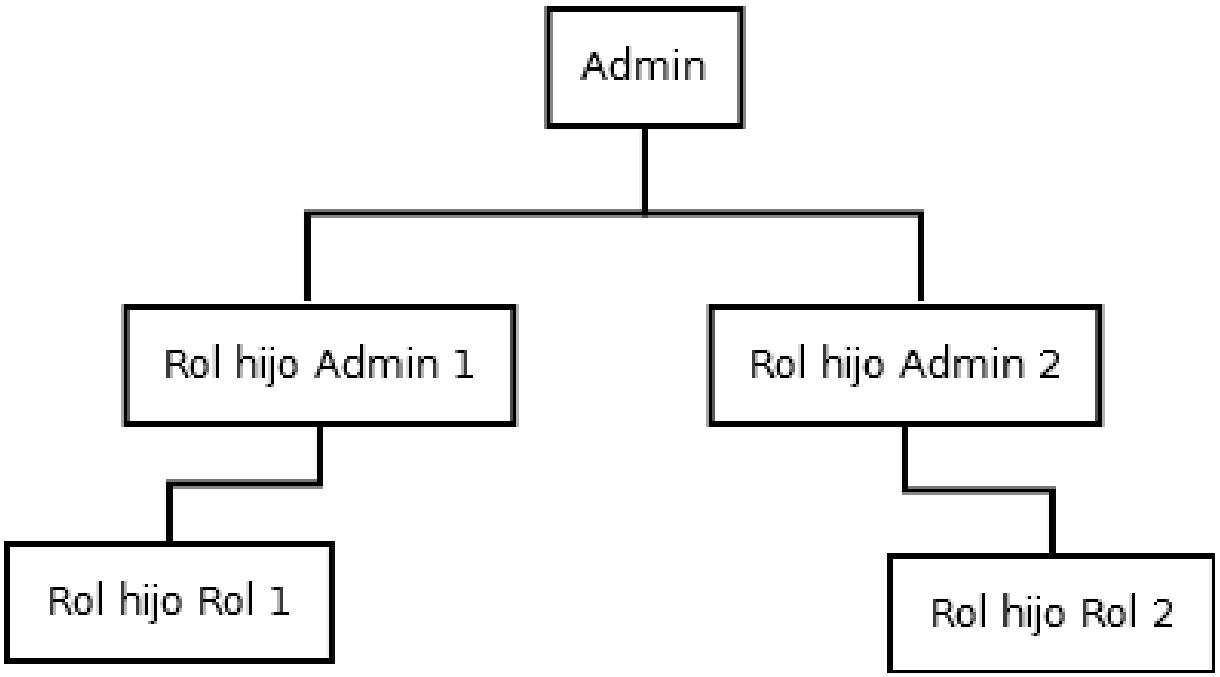


Figura 9: Diagrama Organización

De esta manera los usuarios dentro de un Rol inferior, o hijo, no podrán ver los Archivos del Rol superior, o padre, mientras que los usuarios en un rol padre sí que podrán ver los Archivos de los Roles inferiores a ellos. Los usuarios de los Roles que estén a la misma altura en la Organización, por ejemplo los roles Rol hijo Admin 1 y Rol hijo Admin 2, no podrán ver los Archivos de sus ramas correspondientes, esto es, el Rol hijo Admin no podrá ver los de Rol hijo Rol 1 aun estando en un nivel superior de la jerarquía.

Dentro del control de las Organizaciones del usuario existen varios puntos, en todos los procesos justo después de recibir la llamada el servidor comprueba si el correo electrónico que le llega tiene las *cookies* de control de sesión correctas, si no mandaría un mensaje de error al cliente:

1. Organizaciones usuario: El servidor recibe en la cabecera de la llamada el correo electrónico del usuario y devolverá el nombre de estas al cliente.

2. Creación de una nueva Organización: El servidor recibe el correo electrónico del usuario y el nombre de la nueva Organización, si hubiese alguna organización con ese nombre devolvería un mensaje de error, si no crearía una nueva Organización y le añadiría un Rol de nombre Admin al que se le añadiría el usuario que ha creado la Organización.

3- 4.4. Roles

En esta Sección se explica cómo se controlan los Roles de una Organización en concreto, como en el punto anterior antes de realizar cualquier acción el servidor comprueba si el usuario tiene su sesión correcta, sino devolverá error:

3. Roles de una Organización: El servidor recibe el correo electrónico del usuario y el nombre de la Organización, no puede recibir el nombre de una Organización inexistente, esto es porque el usuario accede a una Organización mediante los nombres devueltos en el Sección 4.3 del Capítulo 3, página anterior, y devuelve el nombre de los Roles de la Organización así como el nombre del Rol padre de los mismos.

4. Roles hijos: El servidor recibe el correo electrónico del usuario, el nombre de la Organización y el nombre del Rol padre, igual que el punto anterior el nombre del Rol existe dentro de la Organización porque se accede al mismo mediante la consulta anterior, el servidor devolverá la lista de todos los Roles de la Organización que tengan como padre al Rol que ha recibido.

5. Creación de un Rol: El servidor recibe el correo electrónico del usuario, el nombre de la Organización, el nombre del Rol y el nombre del Rol Padre, este último obtenido a través de la consulta número 1 de esta Sección; comprobará que en esta organización no existe un Rol con el mismo nombre y si no existe creará un nuevo Rol.

6. Adición de un Usuario a un Rol: El servidor recibe el nombre del Rol, el correo electrónico del usuario que usa la aplicación, el correo electrónico del usuario al que se quiera asignar el Rol y el nombre de la Organización; se comprueban dos cosas, primero que exista el usuario al que se quiere adjuntar el Rol y segundo si ese usuario ya tiene un Rol en la Organización; si alguna de las dos condiciones se cumpliese se devolvería un error al servidor, si no se cumplen se crearía un registro en la tupla de RolUser.

7. Eliminación de un Rol: El servidor recibe el correo electrónico del usuario, el nombre de la Organización y el nombre del Rol a borrar, el servidor borrará el Rol así como todos sus hijos.

3- 4.5 Archivos

Por último, en la parte de desarrollo del servidor, se procedió a dar forma a toda la parte referente al control de los ficheros, tanto de acceso en la Organización como de fecha de control.

1. Crear un archivo: El servidor recibe el correo electrónico del usuario, el nombre del archivo, el Rol al que se quiera adjuntar, la Id del archivo, la clave de descifrado y la fecha límite hasta la que el usuario haya establecido que se puede leer el archivo. El servidor lo único que hará será crear el nuevo registro en la tupla de Archivo.

2. Archivos en un Rol: El servidor recibe el correo electrónico del usuario, el nombre de la Organización y el nombre del Rol, el Rol se extrae de la consulta hecha en el Sección 4.2 del Capítulo 3, y devuelve una lista con los nombres de los archivos, su Id y la fecha hasta la que se pueden leer.

3. Borrar un archivo: El servidor recibe el Id del archivo, obtenido en la consulta 2 de esta Sección y lo borra.

4. Cambio de fecha: El servidor recibe el Id del archivo, obtenido en la consulta 2 de esta Sección, y la nueva fecha hasta la que se podrá leer el archivo y actualiza su fecha.

5. Leer un archivo: El servidor recibe la Id del archivo y el correo electrónico del usuario que quiere leerlo, a través de la Id obtiene el Rol al que está adjuntado y comprueba si el usuario tiene privilegios suficientes como para poder leer el archivo, es decir, que el Rol al que el usuario esta ligado es igual o superior al que el archivo está adjuntado, si no lo tiene el servidor devuelve un mensaje de error por falta de permisos de lectura. Una vez comprobado el permiso comprobará si la fecha de lectura es superior a la de hoy, si no lo es mandará un mensaje de error, si lo es mandará la clave de descifrado.

Con esto queda definido como se desarrolló el servidor y las distintas partes que se llevaron a cabo para que pudiese atender las necesidades de cliente.

3- 5. Pruebas servidor

Para realizar las pruebas en el servidor se fue siguiendo una metodología modular, es decir, se probó cada módulo del servidor por separado.

Para la Sección 4.1 del Capítulo 3, se realizaron los siguientes pasos:

1. Se enviaron las cabeceras para la creación de un usuario con un correo electrónico, *password* y nombre de usuario. Una vez recibida la respuesta de éxito en la creación del usuario se pasó a verificarlo en la base de datos, y comprobar si la contraseña estaba de verdad encriptada.
2. Se volvió a mandar ese mismo correo electrónico de usuario pero con distinta *password* y nombre de usuario, el servidor devolvió mensaje de error y se dio por válida la prueba.

La siguiente Sección fue la 4.2 del Capítulo 3:

1. Usando como referencia el usuario creado en el apartado anterior se procedió a mandar un correo electrónico que no existía en la base de datos pero con la *password* del usuario ya creado, el servidor devolvió un mensaje de error.
2. Se usó el correo electrónico correcto pero con una *password* erronea, el servidor devolvió un mensaje de error, se repitió la operación 5 veces y el servidor respondió con el mensaje de impedimento de acceso debido a los numerosos intentos de acceso de sesión erróneos.
3. Finalmente se pasó a enviar el correo electrónico y la *password* correctos y el servidor devolvió el mensaje de verificación, así como la información de las *cookies*, se procedió a comprobar que la variable de sesión se guardó correctamente en el servidor.

Para las pruebas, de aquí en adelante, se desactivó la verificación de la sesión del usuario que accedía al servidor para evitar molestias y retrasos por problemas de acceso, así como para ahorrarnos en los scripts de pruebas la necesidad de mandar las *cookies* al servidor cada vez que se

hacía una prueba. A continuación se describen las pruebas sobre la Sección 4.3:

1. Fue necesario la creación de más usuarios para realizar unas pruebas más exhaustivas.

2. Se mandó al servidor el correo electrónico del primer usuario creado con el nombre de una organización aleatorio. Una vez recibido el mensaje de éxito del servidor se procedió a comprobar que se había guardado en la base de datos el nombre de la Organización, que se había creado un Rol de nombre "Admin" asociado correctamente a la organización creada y finalmente que se había creado el registro que indicaba que el usuario estaba ligado a este Rol en la tupla de UserRol.

3. Se mandó al servidor exactamente los mismos datos que en el paso anterior, se recibió un mensaje de error.

4. Se procedió a crear más Organizaciones de nombres aleatorios para la continuidad de las pruebas, todas ellas con su correspondiente comprobación de que se habían guardado correctamente en la base de datos.

5. Se mandó al servidor el correo electrónico del usuario, la Organización y un *flag* que indica que se quiere "ver las organizaciones", se recibió un mensaje de éxito así como todas la Organizaciones del usuario.

La siguiente Sección a probar fue 4.4, esta es la Sección más extensa ya que es el que lleva a cabo toda la funcionalidad de la aplicación, para ello seguimos los siguientes pasos:

1. Usando una de la Organizaciones previamente creadas, y que las que el servidor devolvió el nombre en la prueba 5 de las pruebas de la Sección 4.3, se probó a ver los Roles de la Organización. Se mandó al servidor el nombre de la Organización escogida, el correo electrónico del usuario dueño de la Organización y el *flag* indicando que se querían ver los Roles de la Organización. El servidor devolvió los Roles de esa Organización, en este caso un único Rol el de "Admin" creado por

defecto cuando se crea la Organización, reprobando así la prueba número 2 de la Sección 4.3.

2. Para comprobar la creación correcta de un Rol se mandó al servidor el nombre de la Organización donde estaría el Rol, el correo electrónico del usuario, el nombre del nuevo Rol y el nombre del Rol padre, en este caso el Rol “Admin”; ya que al ser una Organización nueva no tiene más Roles. El servidor mandó un mensaje de creación del Rol y volviendo a repetir el apartado 1 de las pruebas sobre la Sección 4.4 el servidor devolvió el Rol nuevo y el Rol de “Admin”.

3. Se volvió a mandar las mismas cabeceras que el punto 2 de las pruebas de la Sección 4.4 al servidor y este devolvió un mensaje de error ya que no se pueden crear Roles duplicados. Repitiendo el punto 1 el servidor devolvió sólo los dos Roles creados, mostrándose así que no se había creado el Rol duplicado.

4. Se mandó al servidor las mismas cabeceras que el punto 2, de las pruebas sobre la Sección 4.4, sólo que sin nombre del Rol padre, el servidor devolvió error, ya que no puede existir un Rol sin Padre salvo el Rol “Admin” creado por defecto.

5. Para verificar el comportamiento de la gestión de los usuarios ligados a los Roles se procedió mandando al servidor el nombre del Rol, el usuario dueño de la Organización, el usuario al que se le quiere añadir al Rol y la Organización del Rol. El servidor devolvió mensaje de éxito, esto fue corroborado directamente en la base de datos.

6. Se volvió a mandar el mismo mensaje que el punto anterior y el servidor devolvió mensaje de error, seguidamente se mandó el mismo mensaje salvo que se cambió el Rol al que se iba a ligar el usuario y el servidor volvió a devolver error, debido a que un usuario no puede tener más de un Rol por Organización, a continuación, se probó a mandar el mensaje del punto anterior solo que con un Rol no perteneciente a la Organización, el servidor contestó con un mensaje de error, ya que no tenía sentido que se crease un usuario ligado a Rol inexistente; y por último se probó a mandar un usuario que no existía dentro del sistema, con lo que el servidor devolvió error, esto fue

porque la aplicación solo permite que se liguen a un Rol usuarios que estén dados de alta en la aplicación.

7. El siguiente apartado que se probó fue el eliminar un usuario de su Rol, para ello se mandó al servidor el usuario ligado al Rol, el nombre del Rol, el nombre de la Organización, el usuario dueño de la Organización y el *flag* de borrado de usuario-Rol, el mensaje que se recibió del servidor fue de éxito y se comprobó que ya no existía ese registro en la tupla de UserRol.

8. Por último, se probó al borrado de un Rol, para ello fue necesario la creación de más Roles que "colgasen" del Rol creado en el punto 2, de las pruebas sobre la Sección 4.4. Una vez creados se ligaron usuarios a esos Roles para tener una Organización completa. A continuación, se probó a borrar el Rol del punto 2 de las pruebas sobre la Sección 4.4 y se comprobó que se borraron todos sus Roles hijos, así como los hijos de estos y todos los registros de la tupla UserRol que relacionaban estos Roles a distintos usuarios. Todo esto se comprobó a nivel de base de datos. De manera adicional se probó a borrar el Rol de "Admin" y el servidor respondió con error ya que ese es el único Rol que no se puede borrar.

9. De manera relacionada con las pruebas la Sección 4.3, explicadas anteriormente, se probó a borrar la Organización que habíamos creado para el punto 8 para las pruebas de la Sección 4.3. Se comprobó en la base de datos que todos los datos referentes a esta organización se habían borrado, incluidos Roles y UserRoles.

Finalmente la última Sección a probar fue la 4.5, este fue más simple de probar que la Sección anterior. Para comprobar su correcto funcionamiento se siguieron los siguientes pasos:

1. Se comprobó la correcta creación de un registro en la tupla de Archivos, para ello se mandó el correo electrónico del usuario, un Rol previamente creado, la Organización de ese Rol, el id del archivo, la fecha hasta la que se podía leer el archivo y finalmente la clave del mismo. El servidor devolvió mensaje de éxito y se comprobó en la base de datos que se había creado. De manera adicional se mandaron varias de las cabeceras

vacías y el servidor devolvió error en cada una de las pruebas, ya que todos los datos son necesarios.

2. Se probó a mandar un Rol no perteneciente a la Organización, el servidor devolvió error debido a que no existía ese Rol en la Organización indicada.

3. Para comprobar la correcta funcionalidad del control de acceso a los archivos fue necesario la creación de Organización completa, con Roles nietos de "Admin" y usuarios ligados a esos Roles. Se creó un nuevo registro de Archivo, se comprobó que con usuario en el Rol al que estaba ligado devolvía la clave, así como un usuario de un nivel con el Rol padre de este. A continuación se probó a tratar de acceder al Archivo con un usuario con un Rol no padre al que estaba compartido el Archivo y el servidor devolvió error: permisos insuficientes.

4. El siguiente paso fue comprobar que se podía cambiar la fecha límite para ver el archivo, se mandó el id del archivo y la nueva fecha. El servidor respondió con éxito y se comprobó en la base de datos que la fecha se había cambiado con éxito. A continuación se probó con los usuarios que antes tenían visibilidad sobre el archivo y el servidor respondió con error, la fecha es anterior a hoy.

5. Finalmente se procedió a comprobar el borrado del archivo, se mandó al servidor el id del archivo y el *flag* que ordena su borrado. El servidor devolvió éxito y se comprobó que se había borrado con éxito.

6. Adicionalmente de procedió a borrar una Organización con Archivos, se comprobó que se borraban los archivos de esa Organización, así como si sólo se borraba un Rol se borraban los Archivos ligados a este.

Con esto se dio finalizada la parte del proyecto que abarcaba todo el servidor, esto no quiere decir que no se volviese a tocar la parte del servidor, ya que con la parte del desarrollo del cliente fue necesario realizar unos ajustes e incorporar nueva funcionalidad al servidor, esto pequeños cambios se probaron con la regresión completa de la aplicación.

3- 6. Desarrollo del cliente

Como ya se ha explicado anteriormente en el punto de esta memoria el cliente será una plataforma móvil, se ha desarrollado para que se pueda en un SO Android 6.6, Lollipop, sin embargo es compatible para demás versiones hasta la versión 4.0, Ice Cream Sandwich, para ello se ha utilizado el ordenador proporcionado por la UAM anteriormente mencionado y la aplicación del escritorio "Android Studio", así como el libro Hello, Android: introducing Google's mobile development platform [3] y ejemplos encontrados en la página web stackoverflow [5].

Como ya se ha avanzado en la descripción de la arquitectura y el servidor, Secciones 3 y 4 del Capítulo 3 de la memoria, anteriormente explicados; la mayoría de la carga operacional de la aplicación la llevará acabo el servidor, sobre todo en el control de las Organizaciones de los usuarios, el dispositivo móvil solo llevará acabo la parte gráfica de la aplicación, el cifrado y el descifrado de los archivos.

Antes de empezar el desarrollo de la aplicación móvil se definió el flujo que debía seguir la misma para dar cabida al desarrollo del servidor previamente realizado.

Una vez definido el flujo fue necesario habilitar en el *Manifest.xml* de la aplicación que el móvil pudiese conectarse a Internet:

<uses-permission android:name="android.permission.INTERNET/>

Y adicionalmente fue necesario darle permisos para poder navegar por la memoria del dispositivo para poder buscar los Archivos a encriptar:

<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

Una vez definido el flujo y la configuración de los permisos de la aplicación se procedió al desarrollo de cada uno de los distintos módulos que se definieron, siguiendo el modelo del servidor, es decir:

1. Resgistro de usuarios.
2. LogIn.
3. Organizaciones.
4. Roles.
5. Archivos.

Lo primero que se definió de la aplicación fue el *layout* de “Menú de Inicio” donde el usuario tendría que elegir entre darse de alta o “loguearse” en la aplicación.

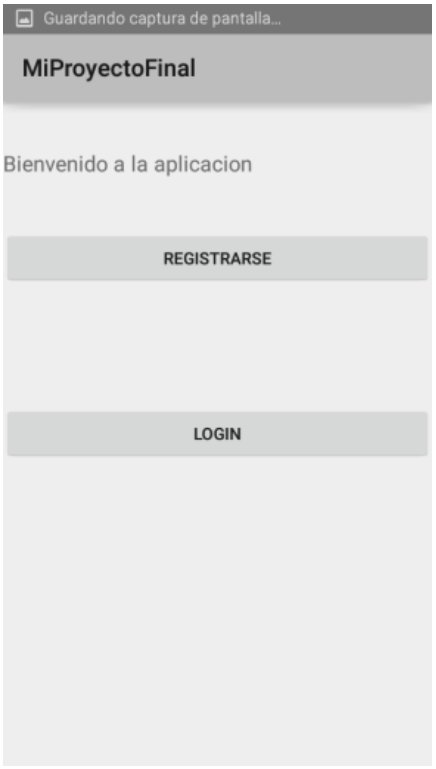


Figura 10: Menú de Inicio

3- 6.1 Registro de usuarios

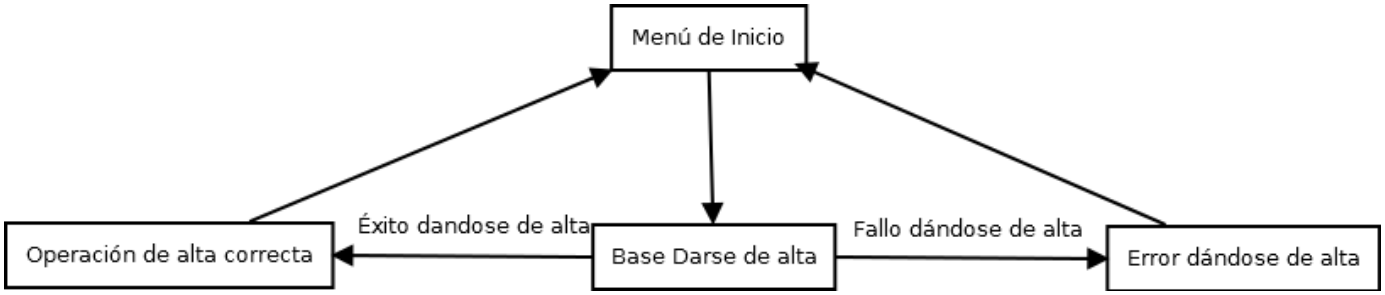


Figura 11: Flujo darse de alta

La primera fase del desarrollo consistió en definir el *layout* de "Darse de Alta", página siguiente, este sería simple, es decir, bastaría con que tuviese una caja de texto donde recoger el correo electrónico del usuario, el nombre de usuario y la *password*. En este *layout* deberían estar contemplados también los posibles errores, tal y como se muestra en el flujo, Figura 11 de esta misma página, que tuviese el usuario que está tratando de darse de alta en la aplicación, esto se aplicará en los *layouts* posteriores. Es importante comentar que en el dispositivo móvil no se comprueba nada en cuanto a la gestión de la contraseña del usuario, que, como se explicó anteriormente se lleva a cabo en el servidor.

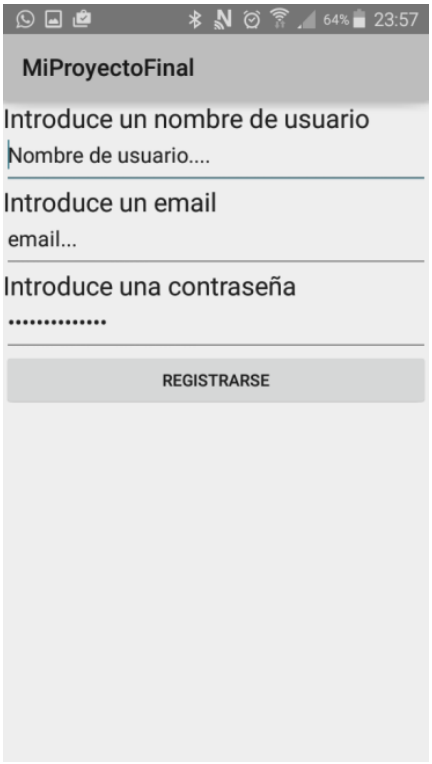


Figura 12: Base Darse de alta

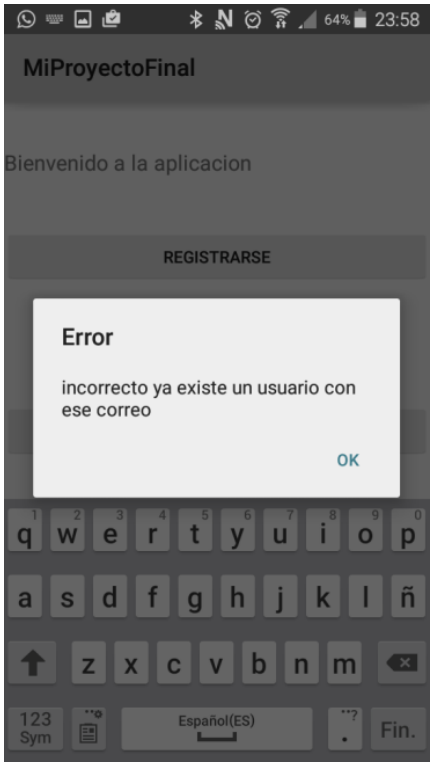


Figura 13: Error dándose de alta

Una vez dado de alta la aplicación le devolverá a la pantalla de inicio y mandará un mensaje de éxito, tal y como se muestra en el diagrama de flujo en la Figura 11.

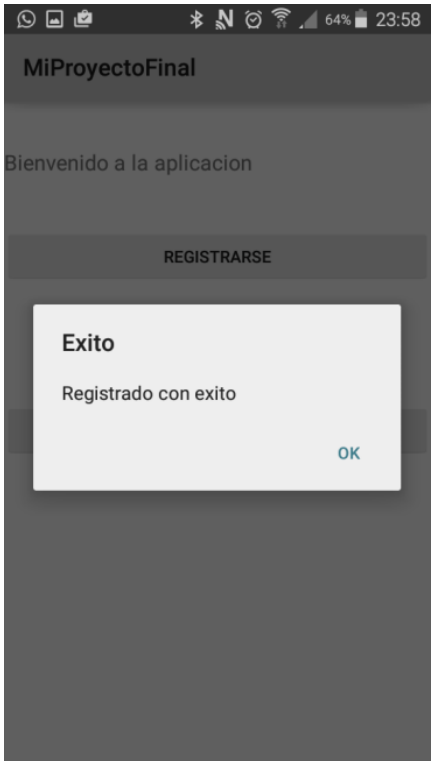


Figura 14: Operación de alta correcta

3- 6.2 Login

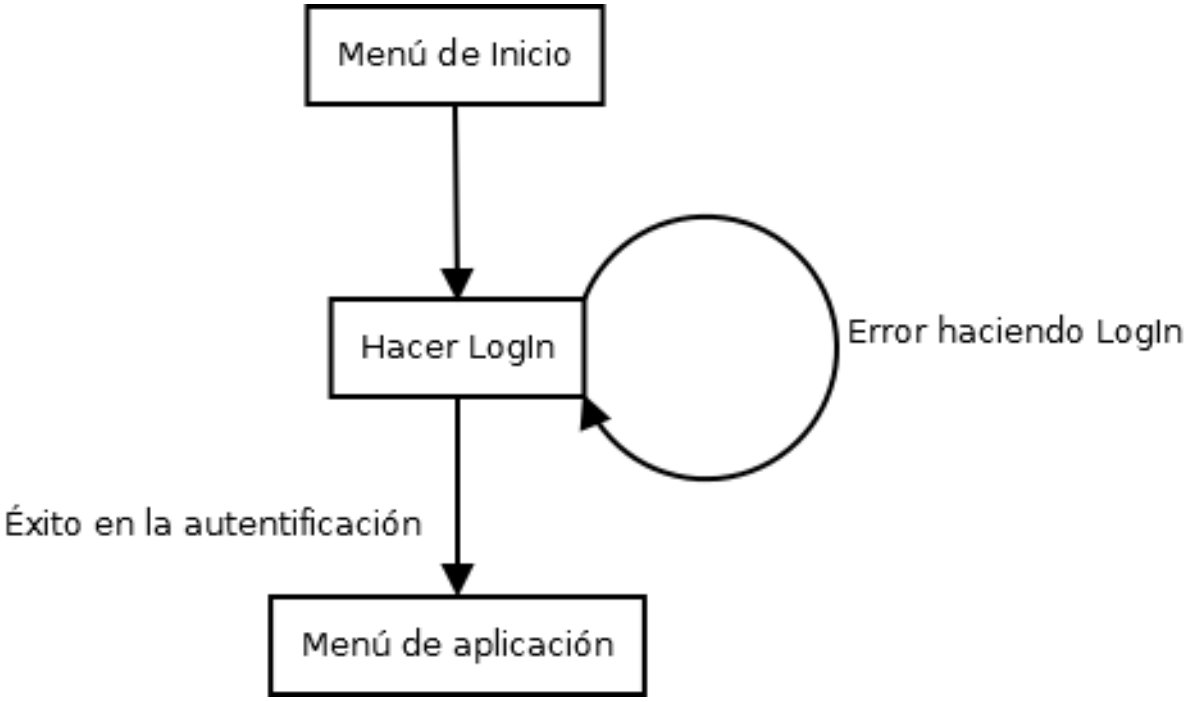


Figura 15: Flujo Login

Una vez que el usuario se ha dado de alta ya puede proceder a acceder a la aplicación poniendo en el *layout* de "Hacer *Login*", Figura 16, su correo electrónico como clave de identificación y la contraseña.

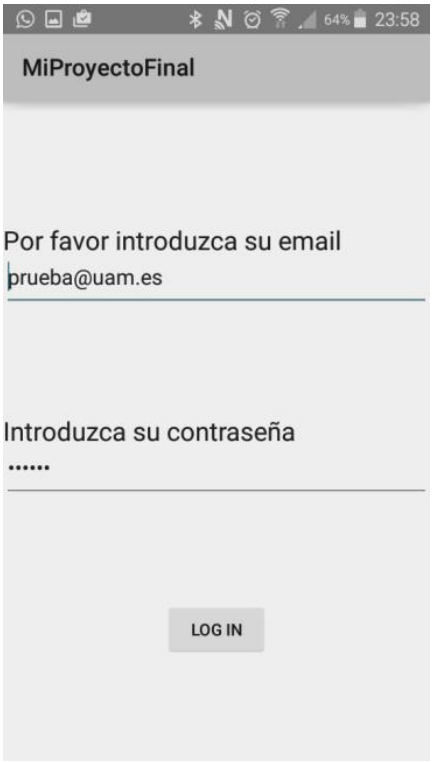


Figura 16: Hacer Login

El móvil enviará estos datos al servidor y si hay éxito de autentificación pasará al *layout* de “Menú de la aplicación”, Figura 18, también recibirá los parámetros de las *cookies* para poder mandar su sesión al servidor y que el usuario pueda seguir usando la aplicación, si no tiene éxito volverá al *layout* de “Error haciendo *LogIn*”, Figura 17, tal y como se muestra en el flujo de la aplicación, Figura 15 de la página anterior.

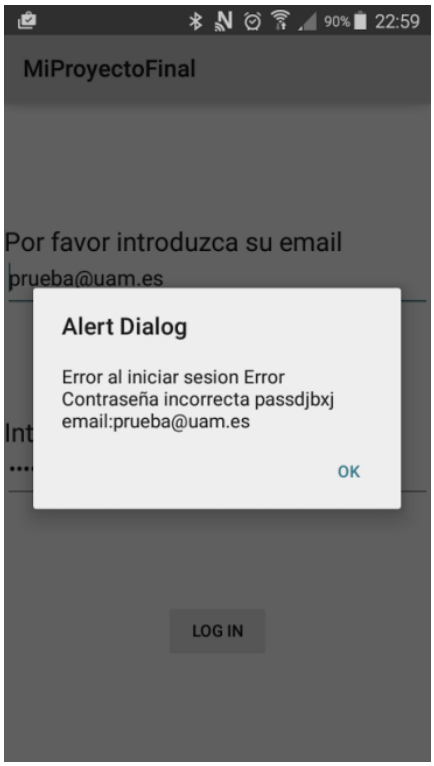


Figura 17: Error haciendo Login

Figura 18: Menú de Aplicación

3- 6.3 Organización

Una vez que el usuario ha hecho *Login* correctamente en la aplicación ya puede empezar a crear Organizaciones y compartir archivos con demás usuarios, todo ello siguiendo el flujo de la aplicación, Figura 20. Lo primero que el usuario debe hacer es crear una Organización donde poder compartir los archivos con los usuarios que desee. Para ello el usuario pulsará en "Ver Organización" y le aparecerá el *layout* de "Organizaciones del usuario", Figura 19.

Figura 19: Organizaciones del usuario

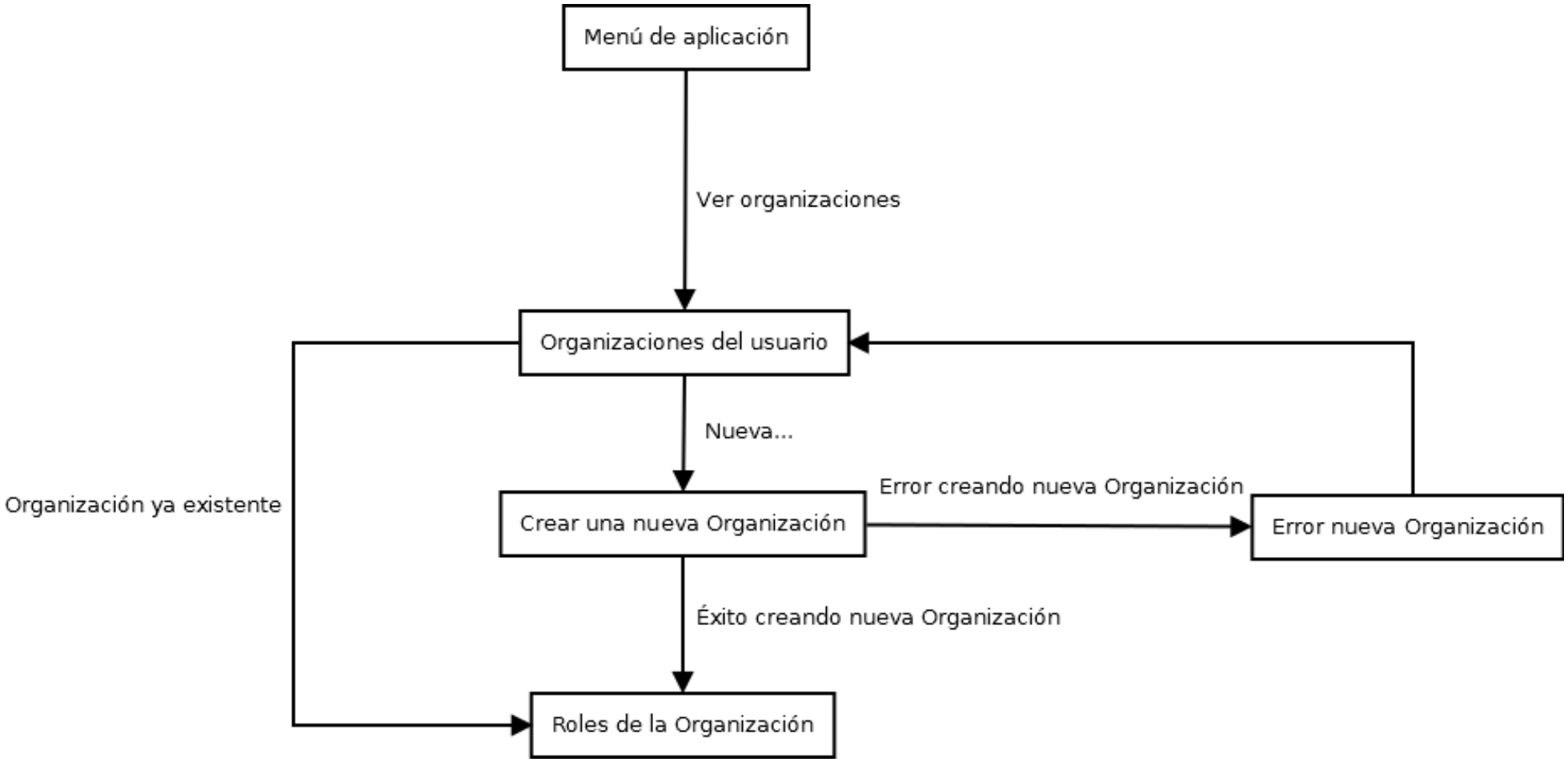


Figura 20: Flujo de gestión de Organizaciones

Aquí el usuario deberá elegir si desea editar una Organización que ya existiese o crear una nueva.

Si desea crear una nueva Organización deberá pulsar en la casilla de "Nueva…", y le mandará al *layout* de "Crear una nueva Organización", Figura 21, donde introducirá el nombre de la nueva Organización. Si el proceso de crear una nueva Organización fallase, normalmente por tener una Organización con ese nombre ya creada, la aplicación le mandaría al *layout* anterior, "Organizaciones del usuario", y mostraría el mensaje de error, Figura 22; si por el contrario tuviese éxito le mandaría al *layout* de "Roles en la Organización", Figura 23, en la siguiente página; donde mostraría el Rol de "Admin" creado por defecto.

Figura 21: Roles de la Organización

Si el usuario desease editar la Organización, en vez de crearla, deberá elegir una de las múltiples organizaciones que se le muestran en el *layout* de "Organizaciones del usuario", Figura 20 de la página anterior, y la aplicación le mandaría al *layout* previamente comentado de "Roles en la Organización" donde se le mostrarían los Roles de la Organización escogida por el usuario, Figura 23.

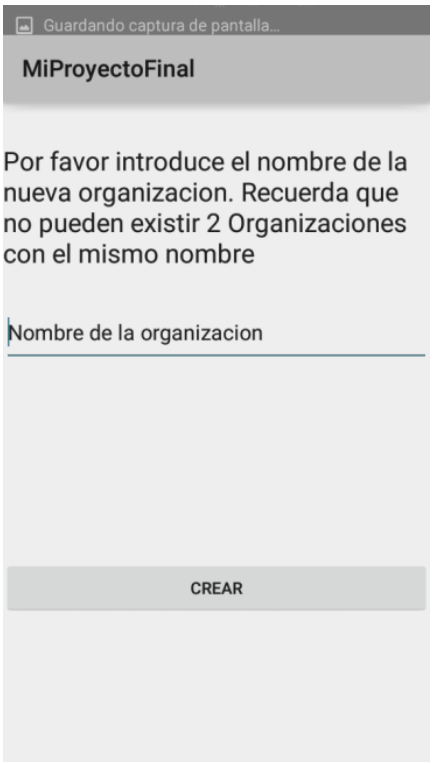


Figura 22: Crear nueva Organización

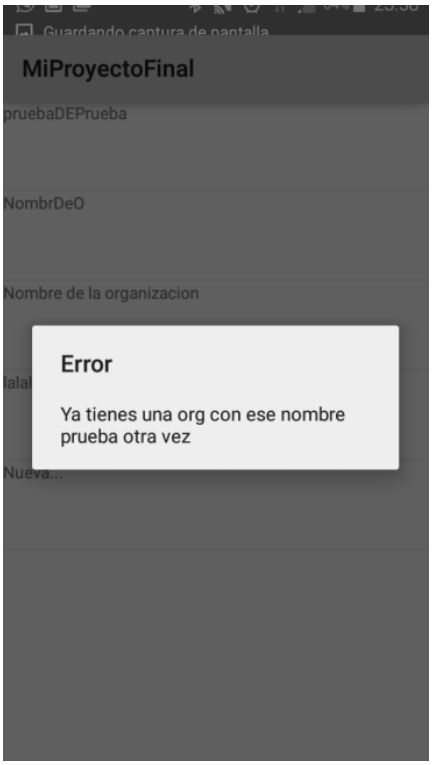


Figura 23: Error nueva Organización

3- 6.4 Roles

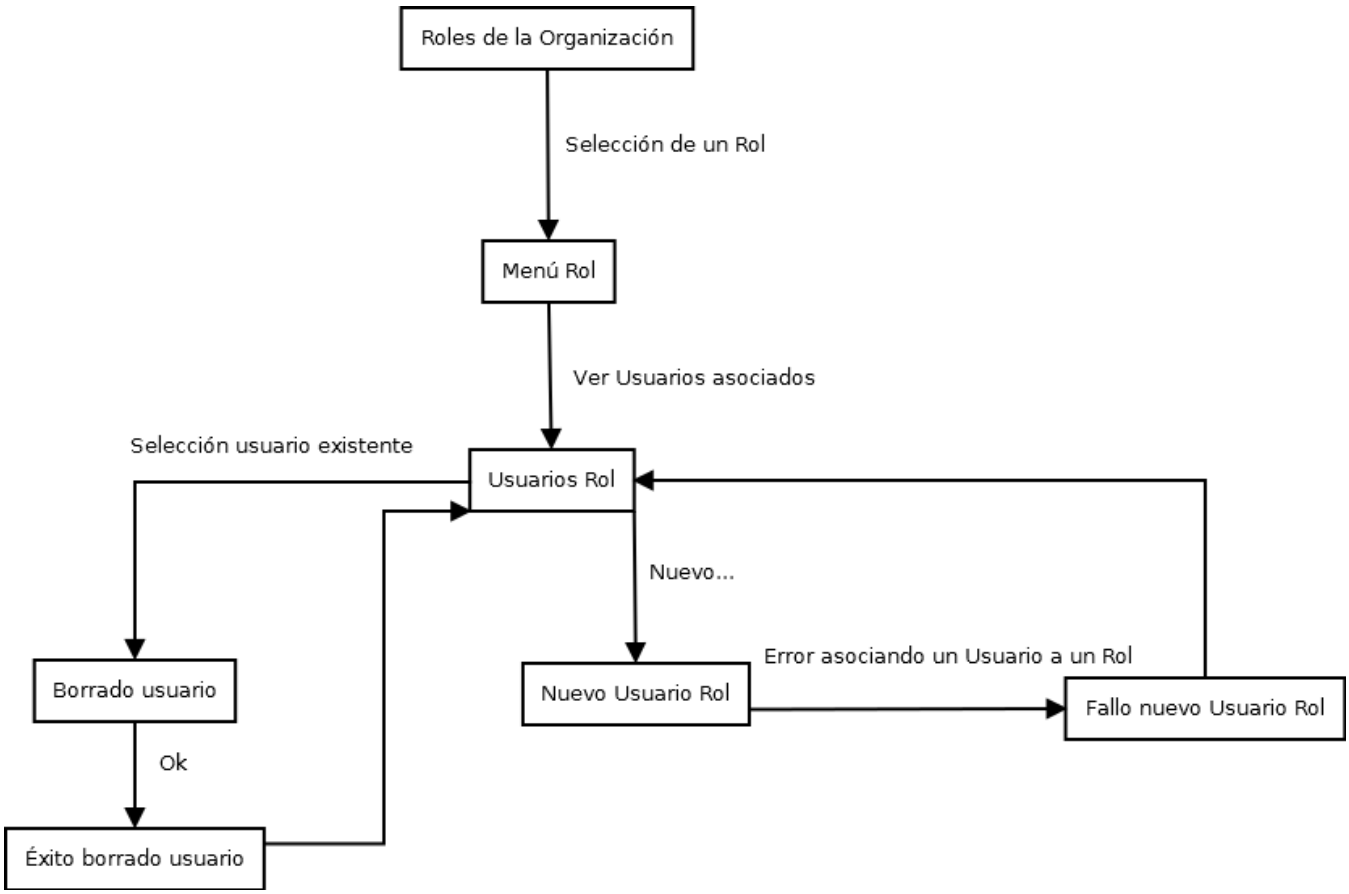


Figura 24: Flujo control Usuario Rol

En el *layout* de “Roles de la Organización”, anteriormente mostrado, aparecen todos los Roles de la Organización elegida por el usuario para editar, o en caso de ser nueva le aparecerá solo el rol de “Admin”.

Para continuar, tal y como se muestra en el Flujo de la Figura 24, el usuario deberá elegir un Rol a editar, en el cual se le mostraran las acciones que puede realizar en ese Rol, tal y como se muestra en el *layout* de “Menú Rol”, Figura 25 de la siguiente página; las tres posibles acciones a realizar son: editar Roles hijos, editar usuarios adjuntos al Rol y editar Archivos adjuntos al Rol.

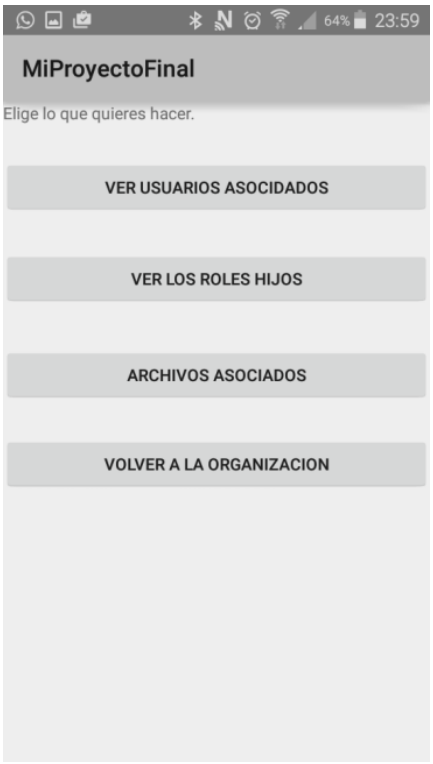


Figura 25: Menú Rol

Si el usuario elige la opción de “Ver Usuarios Asociados”, la aplicación le mandara al *layout* de “Usuarios Rol”, Figura 26, donde se le posibilita la opción de ligar un nuevo usuario, pulsando en “Nuevo…”, a este Rol o eliminar un usuario de este Rol, pulsando en el correo electrónico del usuario en cuestión, Figura 27.

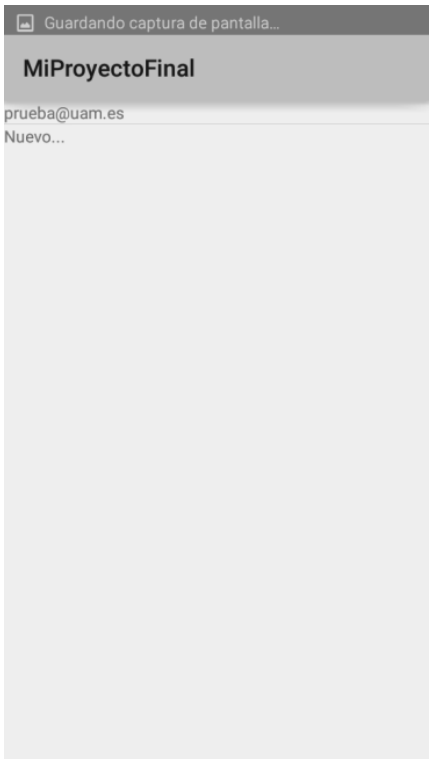


Figura 26: Usuarios Rol

Figura 27: Nuevo Usuario Rol

Si hubiese algún problema a la hora de ligar o eliminar un usuario del Rol la aplicación le devolvería al *layout* de "Usuarios Rol" y le mostraría un mensaje de error, Figuras 28 y 29. En el caso de que el usuario tuviese éxito a la hora de crear o eliminar un Usuario del Rol los cambios aparecerían en el *layout* de "Ver Usuarios Asociados" anteriormente comentado.

Si se desease borrar un usuario de un Rol el usuario pulsaría encima del correo electrónico de los usuarios ligados al Rol, Figura 26, donde le indicaría si realmente quisiese borrar el Rol, Figura 30.

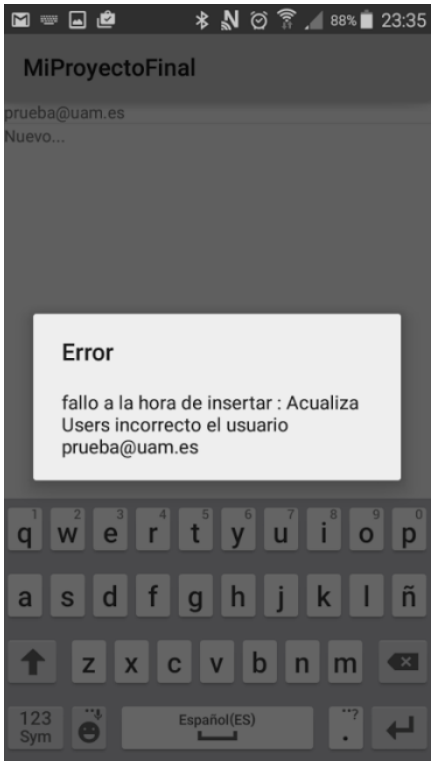


Figura 28: Error Usuario Rol duplicado

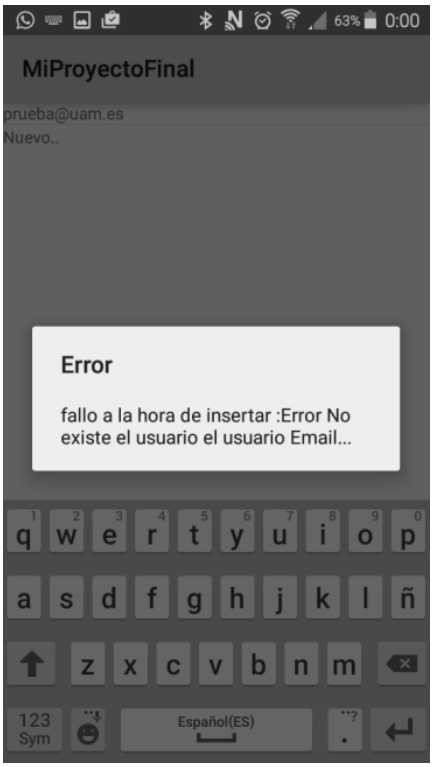


Figura 29: Error Usuario inexistente

Figura 30: Borrado usuario

Ahora bien, si el usuario pulsa en la opción de " Ver Roles Hijos", le mandará al *layout* de "Ver Roles Hijo", Figura 33 de la siguiente página, donde aparecerá una lista con los Roles que esten por debajo en la jerarquia de la Organización, asi como la opcion de crear un nuevo Rol hijo de este Rol, pulsando en "Nuevo.."; donde la aplicación lo enviará al *layout* de "Nuevo Rol Hijo", Figura 32 de la siguiente página; todo esto siguiendo el Flujo de la aplicación de la Figura 31 que se muestra a continuación.

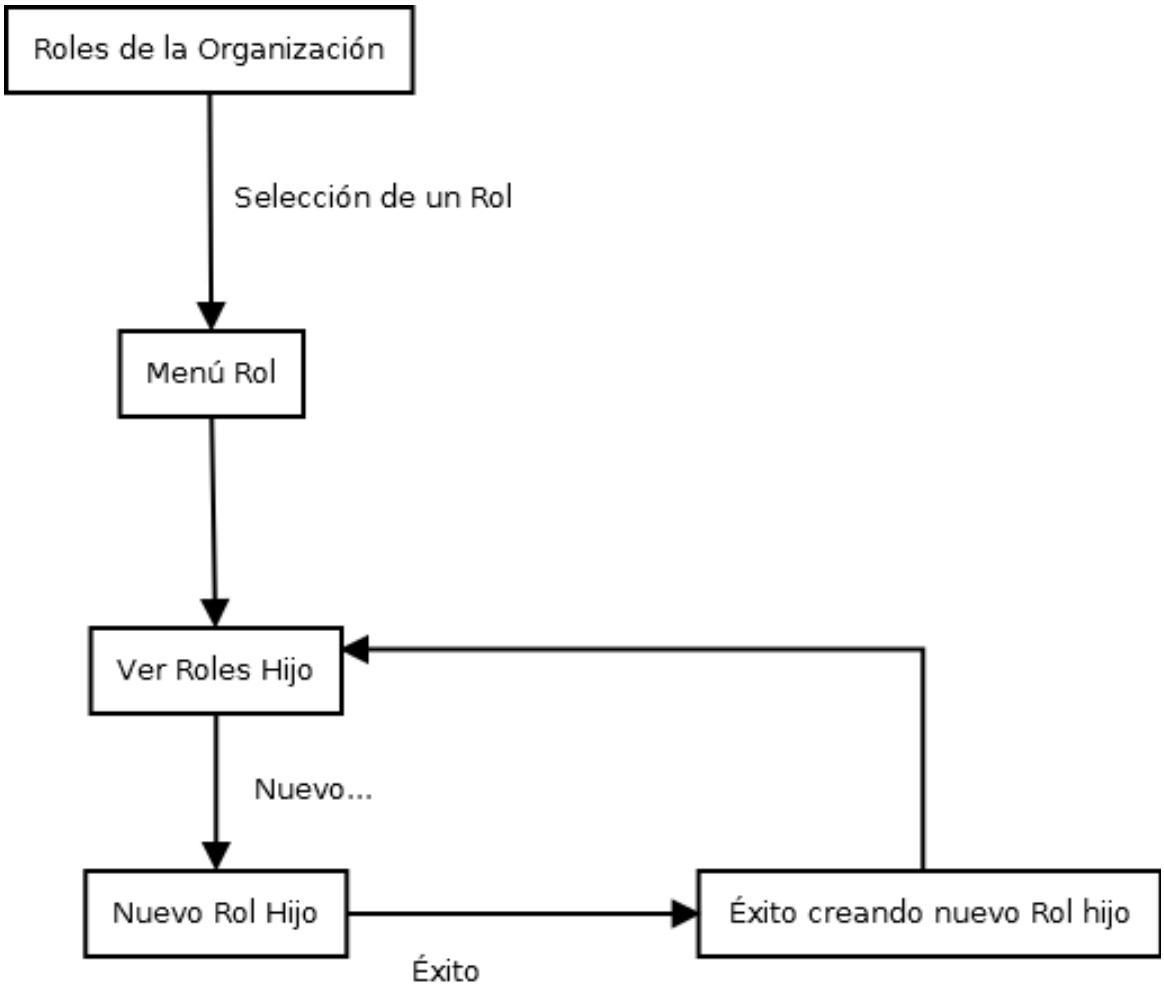


Figura 31: Flujo gestión Rol

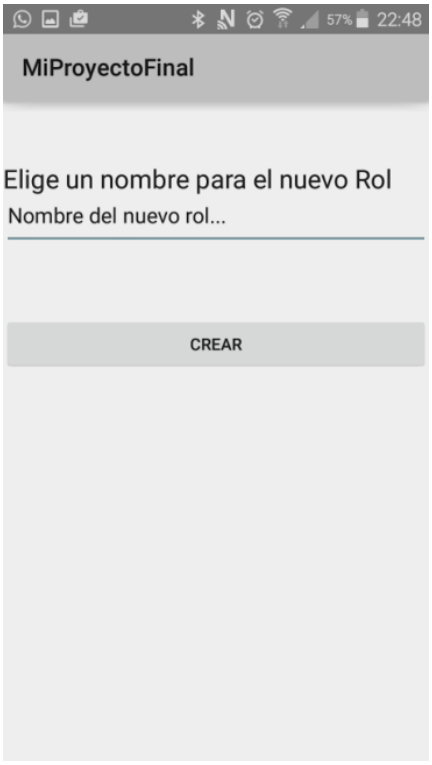


Figura 32: Nuevo Rol Hijo

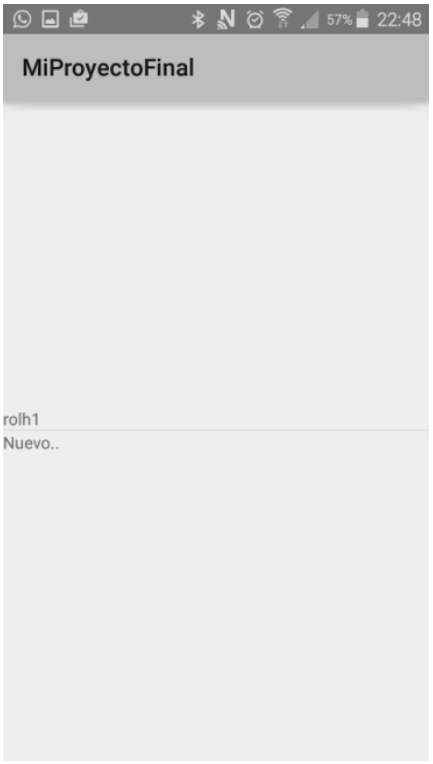


Figura 33: Ver Roles Hijo

Por último si el usuario elige la opción de “Ver Archivos Asociados” la aplicación le mandará al *layout* de “Menú de Archivos”, Figura 34, donde se le mostrará dos posibles opciones, ver los Archivos adjuntados al Rol y la opción de subir un nuevo Rol.

Figura 34: Menú de Archivo

3- 6.5 Archivos

La parte de preparación de los Archivos es la única de toda la aplicación en la que el cálculo operacional la lleva el móvil, esto se decidió así debido a que es mucho más rápido la generación del Archivo cifrado directamente en el móvil que mandar el archivo sin cifrar al servidor, que lo cifrase y que finalmente lo mandase al móvil de vuelta.

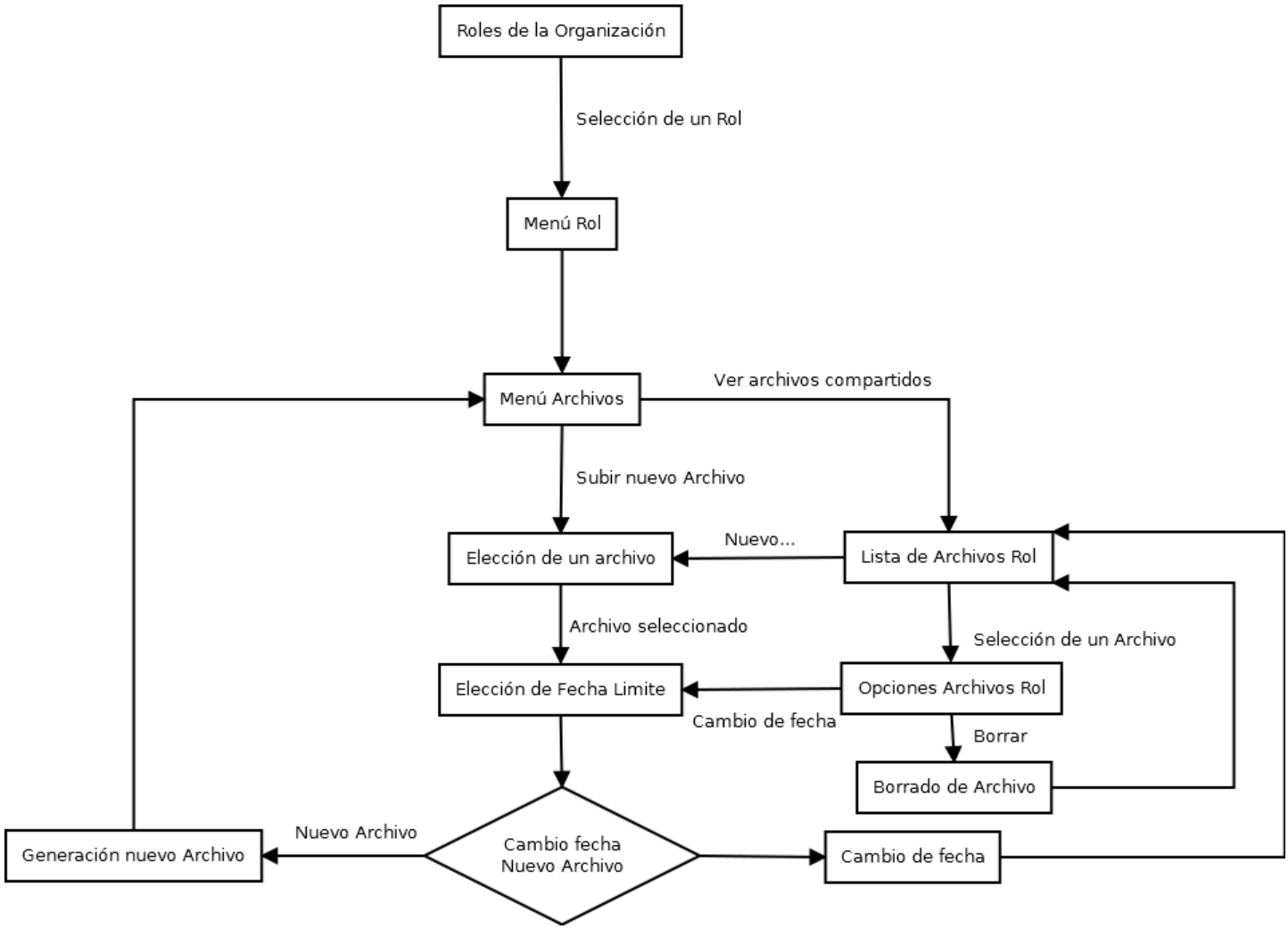
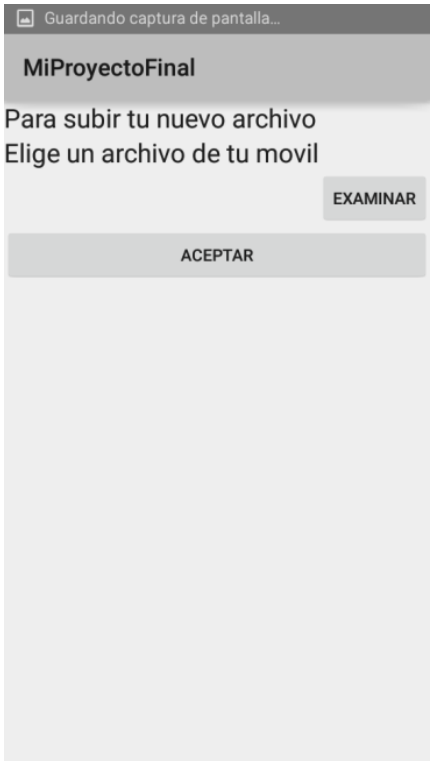


Figura 35: Gestión de Archivos Rol

Para compartir un Archivo el usuario, siguiendo el flujo de la aplicación mostrado en la Figura 35, partirá del *layout* “Menú de Archivos” , Figura 34,en la página anterior; al cual se ha llegado eligiendo un Rol de una Organización; y pulsará en “Subir un nuevo Archivo”, que lo llevará al *layout* de “Elección de un archivo”, Figura 36 de la página siguiente, la cual es una interfaz para acceder a la memoria del móvil, Figura 37, y elegir el Archivo en cuestión.



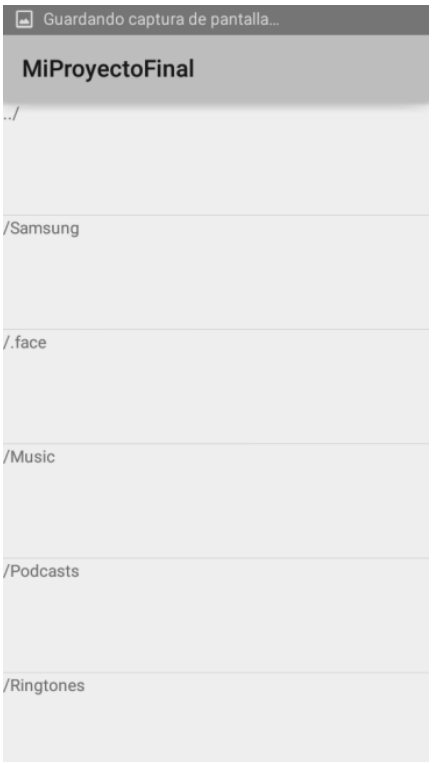


Figura 36: Eleccion de un archivo

Figura 37: Navegación móvil

Una vez seleccionado un archivo con extensión “.pdf”, ya que el móvil sólo muestra las carpetas del dispositivo móvil y los archivos con extensión “.pdf”, la aplicación le mandará al *layout* de “Elección de Fecha Limite”, Figura 39, en el cual solo se podrán poner números y con ello el usuario escribirá la fecha.

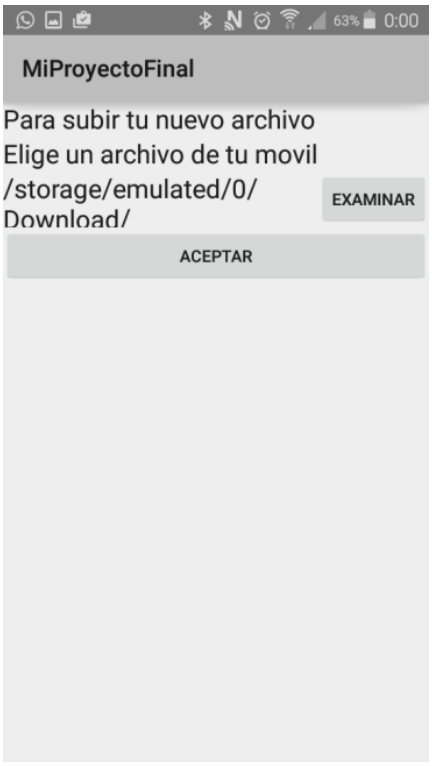


Figura 38: Elección de un archivo (Ya elegido) Figura 39: Elección de fecha Límite

Una vez realizado todos estos pasos el móvil empezará con la operativa para poder cifrar el archivo y poder transformarlo en un Archivo protegido, para ello lo primero que realiza es la generación de una *password* aleatoria mediante el algoritmo de *SecureRandom* para conseguir con ella, usándola como semilla, una clave o "key" mediante la clase *SecretKey*. Esta nueva "key" será la clave usada para cifrar y descifrar el Archivo, para cifrar el Archivo se ha usado el algoritmo "AES/CBC/PKCS5PADDING", explicado en el Capítulo 2.

Una vez generado el Archivo cifrado, con el mismo nombre que el original salvo que añadido un prefijo "Protected", se calculará su Hash mediante el algoritmo SHA-1 para obtener un identificador único usado para guardar la clave de descifrado en la base de datos, este tipo de cálculo es teóricamente atacable, sin embargo, por no añadir carga computacional al dispositivo móvil se ha decido dejar este, ya que para que se produjese colisión de Id de archivos se requerirían 2^{60} Archivos[13].

[13] https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html

Una vez ya enviado el archivo al servidor este se podrá visualizar en el *layout* de “Lista de Archivos Rol”, Figura 40, si el usuario pulsa en la opción de “Ver Archivos compartidos” en el *layout* de “Menú Archivos”, Figura 34 de la página 54; en la cual si pulsa en cualquier archivo compartido podrá ver la fecha límite del Archivo, así como su nombre. Si el usuario pulsase en cualquiera de los Archivos podrá o bien cambiar la fecha límite, la aplicación le mandaría al *layout* de “Elección de fecha Límite” donde el usuario escribiría la nueva fecha límite del archivo, Figura 39 de la pagina anterior, o borrar el archivo de la base de datos, mostrado en la Figura 41. Tambien podrá, si lo desea, crear un nuevo Archivo desde este *layout* sin necesidad de volver atrás y elegir la opción de “Subir un nuevo Archivo”.

Figura 40: Lista de Archivos Rol

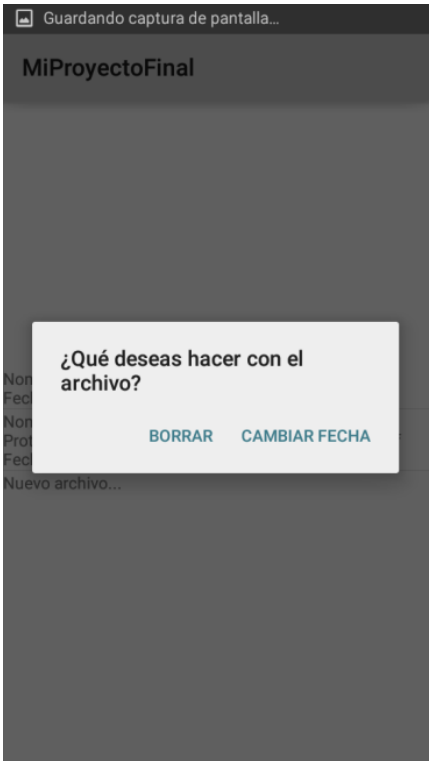


Figura 41: Opciones Archivos Rol

Con esto quedaría concluida la parte de la configuración de acceso a los Archivos de la aplicación, ahora el usuario podrá compartir estos Archivos cifrados con los demás usuarios que elija, siempre y cuando esten dados de alta en la aplicación, ya que sino no podrán acceder al Archivo.

La última parte por explicar sería la de visualización de un Archivo dentro de la aplicación, para ello, se ha llevando como guia el flujo de la aplicación mostrado en la Figura 42.

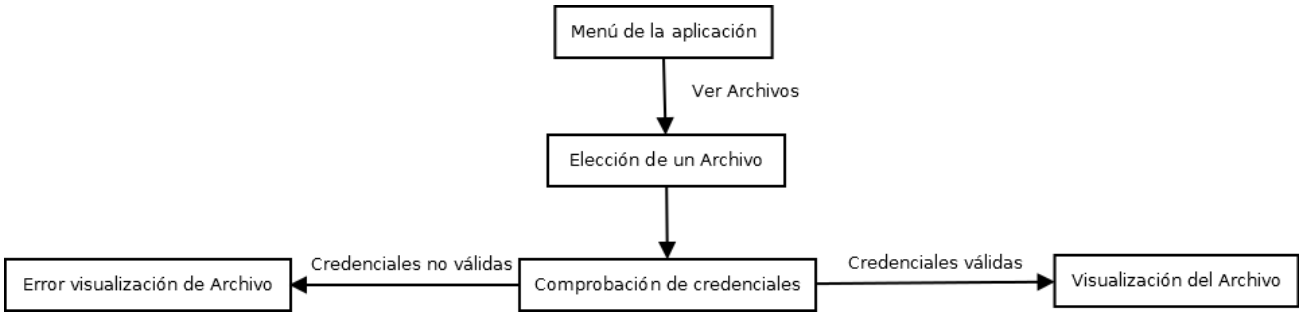


Figura 42: Flujo visualización de Archivo

Para ver un Archivo el usuario ha de volver prácticamente al inicio de la aplicación en el *layout* de “Menú de la aplicación”, Figura 18 mostrada en Sección 6.2 del Capítulo 3, y pulsar en “Ver Archivos”; la aplicación lo mandará a un *layout* identico, salvo por el mesaje que aparece, y con la misma funcionalidad de “Elección de un Archivo”,Figura 33, explicado anteriormente en este punto, Figura 38. Tras lo que, despues de verificar que tanto el usuario que intenta leer el Archivo tiene los permisos disponibles para leerlo, como que la fecha de hoy es anterior a la fecha límte establecida por el usuario dueño del Archivo y finalmente descifrar el Archivo; la aplicación le mandará al *layout* de “Visualización del Archivo”, Figura 46, donde el usuario podrá ver el archivo en cuestión.

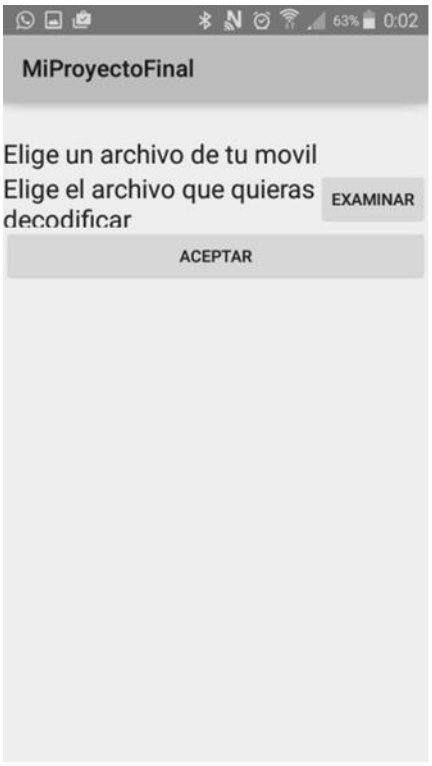


Figura 43: Elección de un Archivo

Figura 44: Visualización archivo sin "zoom" izquierda, con "zoom" derecha

En caso de que el archivo no pudiese ser visualizado debido a que el archivo ya no exsite debido a que el usuario que lo ha compartido lo ha borrado se mostraria un mesaje de error, como se muestra en la Figura 44, o que la fecha para la visualización del Archivo a expirado y no se puede ver, Figura 45, también mandaria mensaje de error debido a que por jerarquia de la Organización el usuario que trata de leer el Archivo no tiene los suficientes privilegios, Figura 47.

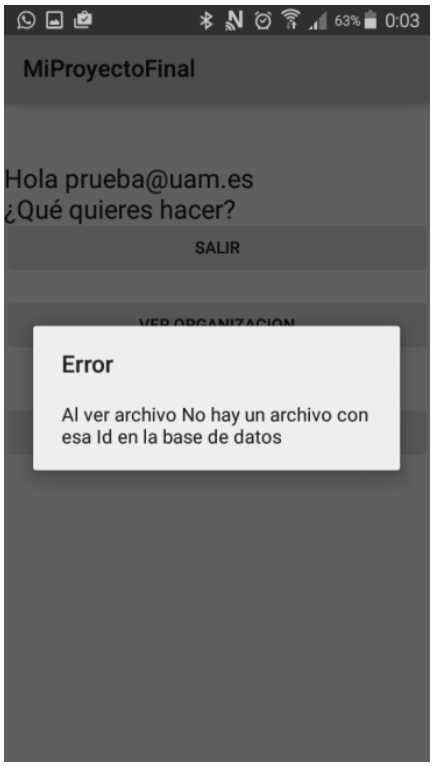


Figura 45: Archivo borrado

Figura 46: Fecha anterior a hoy

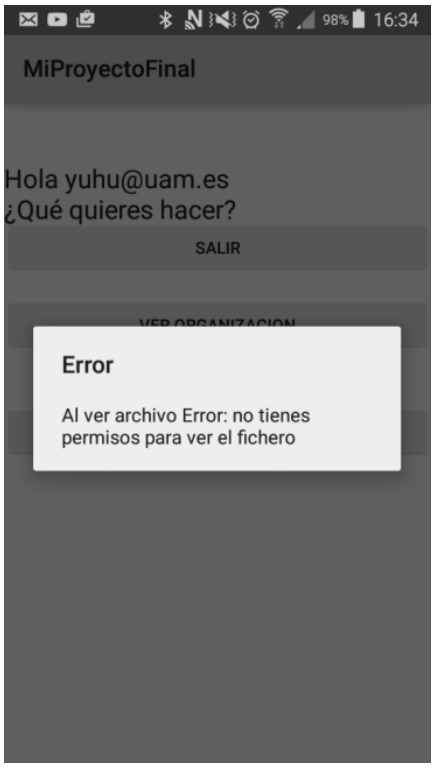


Figura 47: Error privilegios insuficientes

3- 7. Pruebas Cliente

Las pruebas del cliente han sido realizadas punto a punto, según se realizaba el desarrollo, tal y como se ha ido mostrando a través de la explicación del desarrollo del cliente, sin embargo, hubo un último módulo que fue necesario probar con el tipo de pruebas denominadas *end to end* (E2E)[14], este tipo de prueba se basa en realizar el flujo completo de la aplicación, incluyendo el control de errores como el de hacer logIn o el de tratar de visualizar un archivo sin los permisos necesario. El módulo incluido para estas pruebas finales es el establecimiento de la última capa de seguridad, es decir, la implantación del protocolo HTTPS.

En la realización del desarrollo del cliente se estuvo usando el protocolo HTTP para evitar complicaciones a la hora de desarrollar y que generase fallos debido a que el protocolo HTTPS no estuviese bien implementado. Por ello una vez terminado todo el desarrollo de la parte del cliente se pasó a añadir el certificado al cliente y a hacer que lo usase.

Finalizadas las pruebas E2E de manera satisfactoria se dio por finalizado el desarrollo de la aplicación.

[14] https://www.techopedia.com/definition/7035/end-to-end-test

4- CONCLUSIONES Y TRABAJO FUTURO

Tal y como se aclaraba en el alcance del proyecto, este estaba centrado sobre todo en la seguridad de los archivos cifrados así como la viabilidad de que estos tuviesen un tiempo de vida que el usuario dueño de los mismos pudiese cambiar a placer.

En cuanto a la seguridad el proyecto da respuesta a los interrogantes de los DRM, explicado en el Capítulo 1 de la memoria; sobre la capacidad del usuario de controlar el acceso de agentes a un recurso compartido, es decir, de poder establecer una política de seguridad; a través de la estructura jerárquica de la aplicación, solventando el punto de poder hacer grupos de colaboración dentro de la organización y al hacerlos jerárquicos dando privilegios unos sobre otros. A la capacidad de adicional de ser capaz de interrumpir el acceso a dicho recurso a partir de una fecha, dando respuesta a la posibilidad de quitar permisos sobre un activo de información a un grupo o a todo la organización, por lo que el objetivo del proyecto puede darse por cumplido, sin embargo, el proyecto puede mejorarse de diversas maneras.

La primera sería una mejora visual, ya que no estaba dentro de los objetivos del proyecto no se ha puesto ningún tipo de foco en los formatos visuales de la aplicación, más allá de que se pudiese usar de forma simple. En este punto habría que tener en especial cuenta la parte relacionada con la lectura de Archivos, ya que aunque se puedan leer es una forma un tanto incómoda debido a la necesidad de hacer zoom para poder ver los archivos de forma correcta.

La segunda parte giraría en torno a la seguridad de la aplicación, como por ejemplo conseguir un certificado de seguridad de una compañía especializada con lo que conseguiríamos una fiabilidad mayor que un certificado autofirmado, debido a que si en un futuro se decidiese aumentar la capacidad de la aplicación y se pudiese conectar el servidor a otros dispositivos sería mucho más fácil que se integrase con un certificado firmado por una entidad certificadora.

En cuanto a la seguridad hay un apartado crítico que no se ha tratado en este proyecto, que ha sido la seguridad a nivel de hardware, ya que por mucho que los cifrados y encriptaciones sean muy potentes si se puede "leer" a nivel de hardware una persona puede acceder al dato descifrado y obtener la información que se desea, esto se puede conseguir ya que todos los datos para ser descifrados pasan por la memoria RAM del cliente, por lo que sin una buena gestión de ella alguien interesado a acceder a la información podría leer directamente de la memoria el dato descifrado.

Para poder dar solución a este problema se debería usar un Entorno de ejecución seguro (TEE)[15], esto es una combinación entre hardware y software para la protección de datos de ejecución de la aplicación, la protección mediante hardware viene dada por el aislamiento del procesador principal que impide el acceso de las aplicaciones instaladas en el mismo, mientras que la protección brindada por el software cifra los datos dentro del mismo hardware del TEE para separar las distintas aplicaciones instaladas en él. El más claro ejemplo de TEE en un s*martphone* es el programa de introducción del PIN de la SIM Card a la hora de encender el teléfono móvil, ya que la memoria está separada de la memoria general y usa una pequeña pate del procesador que únicamente manda mensajes entre la SIM y la introducción del código.

Otra mejora ya sería a nivel de usuario, es decir, la manera de cómo el usuario establece la conexión con la aplicación. En este punto hay una gran variedad de protocolos que se pueden usar a fin de mejorar la seguridad y la usabilidad de la aplicación.

Un punto a tener en cuenta sería el habilitar que, a través del correo electrónico del usuario, necesario para darse de alta; se pudiese manejar quien ha visto un archivo compartido, así como quién ha intentado verlo y no tenía los permisos necesarios como para verlos, aumentando la capacidad de la aplicación para hacer la trazabilidad a los activos compartidos.

[15] https://en.wikipedia.org/wiki/Trusted_execution_environment

Otro el control de inicio de sesión mediante IP, este protocolo se basa en tener un listado de IPs en las cuales el usuario puede hacer *Login* y en caso de tratar de hacerlo en con IP no incluida en esta lista mandar un correo electrónico para confirmar que se puede entrar a la aplicación desde esa IP e incluirla en la lista de IPs válidas.

El siguiente paso para mejorar la seguridad sería el control de contraseñas en el cual se pueden seguir varias estrategias, como el pedir al usuario cambiar la contraseña cada cierto tiempo, la oportunidad de recuperar la contraseña y la comprobación de uso de una contraseña fuertemente tipada, es decir, con un número mínimo de 8 caracteres debiendo usar al menos una letra mayúscula, un número y un carácter especial.

Otro tipo de control de acceso a la aplicación sería el de implementar un protocolo con doble autentificación, este protocolo se basa en que una vez que el usuario a introducido el correo electrónico y la *password* correctos la aplicación le mandará a su móvil un código de acceso, con duración limitada, que el usuario deberá introducir para poder hacer *Login* definitivamente.

Como se ha comentado en el apartado 6.5 del Capítulo 3, se está generando un identificador del Archivo encriptado con una clave *SHA-1,* la cual teóricamente puede generar colisiones, una mejora de la aplicación seria actualizar a un generador más complejo como el *SHA-2*, que es un derivado del anterior método SHA-1, o el *SHA-3*[16], que surgió de manera oficial en 2015 y no trataba, como el SHA-2, de remplazar al anterior modelo de cálculo de hash, ya que no se ha demostrado que el SHA-2 tenga colisiones, y además tiene un diseño totalmente distinto al de los dos anteriores, que sí que compartían las bases del cálculo del algoritmo; que son versiones más modernas, y por lo tanto más seguras, que el anteriormente mencionado.

[16] https://en.wikipedia.org/wiki/SHA-3

Otro punto que se trata en la Sección 2 del Capítulo 1 de la memoria es la posibilidad de aprender sobre la seguridad en aplicaciones con acceso a Internet, así como la posibilidad de adquirir conocimientos en el desarrollo de aplicaciones en Android, pero en durante la realización del proyecto no se ha conseguido solo estos dos puntos, sino que además se ha conseguido refrescar los conocimientos en el desarrollo de aplicaciones en PHP y adquirir conocimientos sobre la redacción de documentación escrita de manera formal.

Bibliografía

[1] BACKES, Michael, et al. X-pire 2.0: A user-controlled expiration date and copy protection mechanism. En Proceedings of the 29th Annual ACM Symposium on Applied Computing. ACM, 2014. p. 1633-1640.

[2] Welling, L., & Thomson, L. (2005). Desarrollo web con PHP y MySQL. ANAYA MULTIMEDIA, 8441525536

[3] Burnette, E. Hello, Android\nIntroducing Google's Mobile Development Platform. Tech-Books-Pdf.Googlecode.Com 1163, 145-56 (2010).

[4] https://www.apachefriends.org/ Última entrada: 2016-01

[5] http://stackoverflow.com/ Última entrada: 2016-01

[6] https://secure.php.net/ Última entrada: 2015-11

[7]http://www.genbeta.com/web/https-asi-funciona/
Última entrada: 2016-06

[8] https://www.owasp.org/index.php/SQL_Injection/

Última entrada: 2016-06

[9] Maiorano, A. H. (2009). Criptografía: técnicas de desarrollo para profesionales. Alfaomega Grupo Editor Argentino. ISBN 978-987-23113-8-4, Argentina.

[10]https://es.wikipedia.org/wiki/Advanced_Encryption_Standard/
Última entrada: 25/06/2016.

[11]https://www.incibe.es/blogs/post/Empresas/BlogSeguridad/Articulo_y_comentarios/Politica_acceso_informacion_principio_minimo_conocimiento/ Última entrada: 27/06/2016

[12] ZENG, Wenjun; YU, Heather; LIN, Ching-Yung (ed.). Multimedia security technologies for digital rights management. Academic Press, 2011.ISBN 9780123694768

[13] https://tools.ietf.org/html/rfc6101/ Última entrada: 28/06/2016

[14] KAHN, David. The codebreakers. Weidenfeld and Nicolson, 1974. Published by Weidenfeld & Nicolson (1966) ISBN 0722151527

[15] https://tools.ietf.org/html/rfc5246/ Última entrada: 28/06/2016

[16] https://tools.ietf.org/html/rfc2818/ Última entrada: 28/06/2016

[17] https://tools.ietf.org/html/rfc6265/ Última entrada: 28/06/2016

ANEXOS

A. Presupuesto del proyecto

1) Ejecución Material

- Compra de ordenador personal (Software incluido).....1000 €
- Dispositivo móvil Android .....350 €
- Material de oficina.....15 €
- Total de ejecución material..... 1345 €

2) Gastos generales

- 16 % sobre Ejecución Material..... 215.2 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material..... 80.7 €

4) Honorarios Proyecto

- Horas de pre aprendizaje: 50
- Horas de desarrollo del servidor: 140
- Horas desarrollo del cliente: 135
- Horas de escritura de la memoria: 70
- 395 horas a 15 € / hora..... 5925 €

5) Material fungible

- Gastos de impresión..... 50 €
- Encuadernación..... 40 €

6) Subtotal del presupuesto

- Subtotal Presupuesto..... 7565.9 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto..... 1588.84 €

8) Total presupuesto

- Total Presupuesto ..... 9145.74 €

Fdo.: Javier López Andradas. Ingeniero de Telecomunicación

B. PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de **una plataforma para el control de tiempo de acceso a recursos en sistemas distribuidos**. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad “Presupuesto de Ejecución de Contrata” y anteriormente llamado ”Presupuesto de Ejecución Material” que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.