10. CONCLUSIONES

Tal y como se ha comentado a lo largo del desarrollo de esta memoria, existen muchas posibilidades a la hora de adoptar un formato adecuado para la firma electrónica. En este caso, se ha elegido el formato PKCS7, por tratarse de un estándar oficial que proporciona mecanismos de cifrado asimétrico y que permite encriptación del contenido de la firma. El cifrado utilizado es Sha1RSA y el tipo de contenido elegido, entre los distintos soportados por PKCS7, es SignedData, ya que se trata de uno de los tipos de contenidos más simples y más completos.

Además, entre las distintas opciones para implementar librerías criptográficas en el cliente móvil, se ha elegido BounCy Castle por ser una librería, principalmente de código abierto y por poseer todas las clases necesarias para la extracción de claves del certificado e implementación de la firma electrónica. No obstante, ha sido necesario reimplementar ciertas clases y eliminar aquéllas que no eran necesarias, siempre con el objetivo de liberar al cliente de la mayor carga posible.

El cliente no firma el contenido del documento que se desea firmar, sino que realiza la firma electrónica del hash de dicho documento. Dicho hash es devuelto por @firma con la invocación del servicio web correspondiente. Esto es debido, principalmente, a la limitación de memoria del móvil. Cuando el documento a firmar superaba un tamaño determinado ,aproximadamente unos 20K, el simulador dejaba de responder. Por este motivo, se optó por la firma del hash del documento, que suponía una carga mucho menos pesada para el móvil.

El contenido del documento que se pasa como argumento al servicio web que permite obtener el hash de dicho documento, debe ser un String codificado en formato Base64. De no ser así, existían importantes problemas derivados de la codificación.

Uno de los principales objetivos planteados al inicio del desarrollo de esta memoria, era la realización de una firma electrónica que pudiera ser verificada por algún organismo o plataforma oficial. En este sentido, podemos afirmar que la firma realizada es verificada correctamente por la plataforma @Firma de la Junta de Andalucía. Concretamente se ha optado por la versión 5.0, que supone importantes cambios con respecto a la versión anterior, la 4.0.

Tras la conclusión de la aplicación, puede observarse que dicha aplicación resulta muy flexible y no requiere grandes especificaciones de memoria ni computacionales. De hecho, los únicos requisitos que debe poseer el terminal para que la aplicación pueda ser ejecutada son los siguientes:

- Interfaz JSR-75, que proporciona soporte al sistema de ficheros del móvil
- Interfaz JSR-172, que permite al cliente móvil la ejecución de los servicios web
- Aplicación PDF, que permita visualizar el documento antes de ser firmado.

Por último, tras las pruebas realizadas, puede observarse que el consumo de memoria del dispositivo móvil oscila entre el 61,1% y el 66,72%, dependiendo del tamaño del archivo que se desea firmar y de la lista de los documentos que se obtienen para ser firmados.

Desde el punto de vista del servidor web, se ha optado por el servidor Apache-Tomcat, por considerarse uno de los servidores más extendidos y estables. Dicho servidor, permite la integración con el módulo apache-axis sin ningún problema adicional y de forma sencilla.

La versión utilizada de axis es la versión 2, ya que implementa importantes cambios con respecto a la versión 1, entre ellos:

- Velocidad de ejecución
- Bajo consumo de memoria
- Mayor flexibilidad y estabilidad

Para poder desarrollar los métodos que permiten obtener la lista de documentos del servidor WebDAV, así como el contenido de dichos documentos y la información relativa a éstos, se ha tomado como base el cliente de código abierto denominado *skunkdav* y que está basado en el protocolo DAV y que puede descargarse de la página web del autor (referenciada en la bibliografía).

En cuanto a la dificultad de los distintas acciones realizadas a lo largo de este proyecto, ha resultado de especial dificultad, la extracción del par de claves del certificado, ya que dicha clase ha tenido que ser reimplementada para J2ME a partir de su original en Bouncy Castle.

También ha sido de especial dificultad, la codificación de datos intercambiados entre el cliente móvil y los servicios web de @firma implementados el servidor web. Por este motivo, el contenido del documento va siempre codificado en Base64, para unificar criterios.

Por último, cabe destacar que para facilitar la navegación en el móvil a través de la aplicación, se han facilitado en casi todas las pantallas existentes la opción de ayuda. Además, se ha simplificado el uso de botones al máximo.