Servicio de firma de documentos almacenados en el cloud

Luis Fernando Santocildes Romero

"Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones". (MISTIC)

Área del trabajo final

Juan Carlos Fernández Jara

Víctor García Font

Fecha Entrega

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Servicio de firma de documentos almacenados en el cloud*
Nombre del autor:	*Luis Fernando Santocildes Romero*
Nombre del consultor/a:	*Juan Carlos Fernández Jara*
Nombre del PRA:	*Víctor García Font*
Fecha de entrega (mm/aaaa):	06/2021
Titulación:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*Sistemas de autenticación y autorización*
Idioma del trabajo:	*Español*
Palabras clave	*Firma electrónica, eIDAS, almacenamiento en la nube*

Resumen del Trabajo:

El objetivo de este trabajo es estudiar las ventajas que ofrece el Reglamento (UE) nº 910/2014 del Parlamento Europeo y el Consejo del 23 de julio de 2014 relativo a la identificación electrónica y los servicios de confianza para las transacciones electrónicas en el mercado interior (eIDAS). El reglamento amplía el uso aceptado de los certificados y de la firma electrónica, permitiendo que los certificados y las firmas electrónicas se puedan crear, almacenar y utilizar en servicios remotos.

Este trabajo estudia el uso práctico de los servicios de confianza cualificados, específicamente los de firma electrónica, desarrollando un demostrador para la firma remota de documentos alojados en la nube, sin que sea necesario la instalación de software extra o el uso de dispositivos criptográficos por parte del usuario. Para ello se utilizarán los servicios de un proveedor de servicios de confianza, y se integrarán varios servicios de almacenamiento remoto.

Abstract:

The purpose of this essay is to study the advantages offered by the REGULATION (EU) No 910/2014 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market (eIDAS). The regulation extends the accepted use of certificates and electronic signatures, allowing that the certificates and electronic signatures can be created, stored and used in remote services

This work studies the practical use of qualified trust services, specifically electronic signature services. For this, a demonstrator will be developed to do a remote signature of documents hosted in the cloud, without requiring the installation of extra software or the use of cryptographic devices by the user. The services of a trusted service provider will be used for the electronic signature, and various remote storage services will be integrated.

Dedicado a Melania

Por su paciencia, apoyo y fuerza todo este tiempo

Índice

1. INTRODUCCIÓN 1

1.1. CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO 1
1.2. OBJETIVOS DEL TRABAJO 4
1.3. ENFOQUE Y MÉTODO SEGUIDO 5
Fase 1: Análisis y diseño 6
Fase 2: Planificación 6
Fase 3: Estudio de tecnologías y protocolos. 6
Fase 4: Desarrollo del producto y puesta en producción 6
Fase 5: Documentación 7
1.4. PLANIFICACIÓN DEL TRABAJO 7
1.5. BREVE SUMARIO DE PRODUCTOS OBTENIDOS 8

2. ARQUITECTURA Y DISEÑO 9

2.1. REQUISITOS DEL SISTEMA 9
2.2. REQUISITOS DEL REGLAMENTO EIDAS 10
Servicios de confianza 10
Prestadores de servicios de confianza 11
Dispositivo de creación de firma electrónica (QSCD) 12
2.3. ARQUITECTURA DEL SISTEMA 13
1. Modelo - Arquitectura de los datos 14
2. Controlador – Arquitectura del flujo de información 15
3. Vistas – Interfaz con la aplicación 15
2.4. PROTOCOLOS 16
1. HTTPS 16
2. REST 17
3. OAuth2 17
4. OpenID 19
2.5. ESTÁNDAR PADES 20
2.6. ESTÁNDAR PKCS 22
1. PKCS#1 22
2. PKCS#12 22
2.7. FLUJOS DE INFORMACIÓN 23
2.8. CASOS DE USO 26
2.9. INTERFAZ DE USUARIO 31

3. DESARROLLO E IMPLEMENTACIÓN 34

3.1. OBJETIVOS DEL SISTEMA 34
Consideraciones 34
3.2. IMPLEMENTACIÓN DEL SISTEMA 35
3.3. IMPLEMENTACIÓN DE FLUJOS DE INFORMACIÓN 38
3.4. INTEGRACIÓN DE SERVICIOS DE DIFERENTES PROVEEDORES DE ALMACENAMIENTO 39
3.5. IMPLEMENTACIÓN DEL PROCESO DE FIRMA 40
3.6. INTERFAZ DE USUARIO 42
3.7. PUESTA EN PRODUCCIÓN 50

4. CONCLUSIONES 51

5. GLOSARIO 52

6. BIBLIOGRAFÍA 54

Lista de figuras

1 - Uso de certificados de firma electrónica en las relaciones de particulares con las AA.PP. a través de Internet. Año 2010. (rojo: DNIe, amarillo: otros certificados). Fuente: Instituto Nacional de Estadística 2

2 - Planificación temporal del trabajo 7

3 - Arquitectura del sistema - Componentes 14

4 - Flujo del protocolo Oauth2 19

5 - Esquema del formato PDF incluyendo la firma electrónica 21

6 - Esquema de la identidad de firma y de la Firma PDF dentro del documento 21

7 - Flujo del proceso de firma 25

8 - Ficheros de FrontEnd 38

9 - Módulos y clases de CloudDocs 38

10 - Ejemplo de código para llamar a APIs RESTfull 39

1. Introducción

1.1. Contexto y justificación del Trabajo

En las últimas dos décadas, se ha generalizado el uso de los medios de comunicación informáticos, especialmente internet, para la realización de tareas que anteriormente exigían de la presencia física de las personas, o al menos de la comunicación directa entre estas en algún momento. En la nota de prensa del 20/11/2020[1], el INE publica que el uso de ordenadores conectados a internet con fines empresariales aumentó del 53,5% al 57,1% entre el 1er trimestre de 2019 y el de 2020. Entre las tareas más habituales podemos encontrar, por ejemplo, la firma de contratos, escrituras y otros documentos, la compra venta de bienes y servicios, la prestación de parte de estos últimos, etc. En la mayoría de los casos, la presencia física ha sido necesaria para validar o certificar la identidad de las partes, o para dar el visto bueno a la operación realizada, bien sea mediante la firma de los documentos necesarios o directamente con el pago de los servicios. En algunos casos, solamente se requería la firma de documentos en papel, siendo necesario enviar físicamente estos documentos, ya firmados, a la empresa u organismo que lo requiriese[2]. Solamente algunos tipos de transacciones, mayoritariamente compras online o contrataciones de servicios electrónicos, no han requerido la comprobación de la identidad de las partes, lo cual ha favorecido el anonimato, pero también ha facilitado la aparición de fraudes por suplantación de identidad.

Con la Directiva 1999/93/CE del Parlamento Europeo y del Consejo de 13 de diciembre de 1999 se intenta establecer un marco europeo que facilite el uso de la firma electrónica y su reconocimiento jurídico, sin embargo, la transposición de esta directiva a las diferentes legislaciones nacionales de los países miembros de la UE dificulta, en la práctica, el reconocimiento de las firmas realizadas con certificados de otros países. Esta directiva es reemplazada por el **Reglamento (UE) Nº 910/2014 del Parlamento Europeo y del Consejo de 23 de julio de 2014 (eIDAS)[3]**, que regula la identificación electrónica y los servicios de confianza para las transacciones electrónicas, siempre en el ámbito del mercado interior de la Unión Europea. Este

[1] https://www.ine.es/prensa/tic_e_2019_2020.pdf

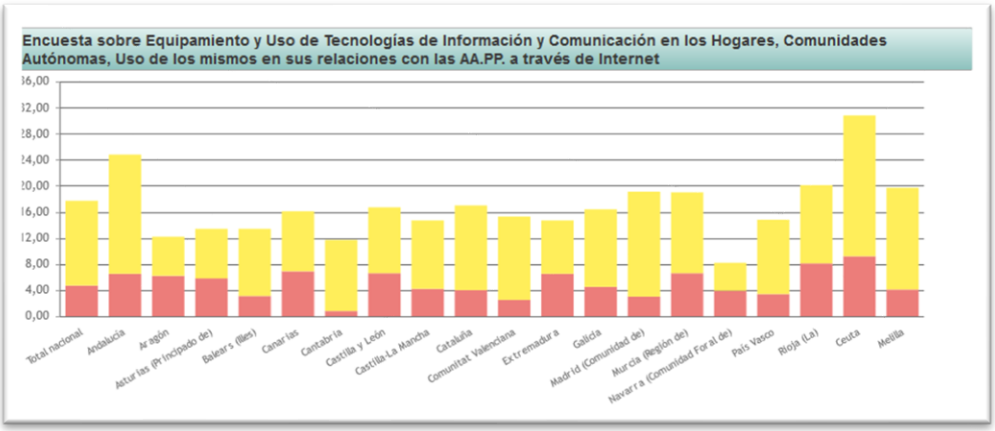
[2] En algunos casos solamente ha sido necesario enviar una copia escaneada de los documentos firmados.

[3] https://www.boe.es/doue/2014/257/L00073-00114.pdf

reglamento pretende mejorar y reforzar la confianza en las transacciones económicas electrónicas. Al ser un reglamento es de obligado cumplimiento en todos los países de la U.E. desde su entrada en vigor[4].

Para conseguir este objetivo, el reglamento amplía y potencia el uso de la firma electrónica, cambiando -entre otras cosas- algunos de los requisitos para la creación, uso y almacenamiento de los certificados y/o identidades de firma utilizados para realizar la firma electrónica. Esta nueva directiva permite el uso de los servicios de un "*prestador cualificado de servicios de confianza*"[5], los cuales pueden emitir y preservar certificados electrónicos, cualificados o no, y crear firmas electrónicas utilizando los certificados custodiados. Estas firmas serán cualificadas siempre y cuando se creen utilizando un dispositivo cualificado de creación de firmas electrónicas y un certificado cualificado. Este es un cambio importante que abre la puerta a la prestación de servicios remotos de firma electrónica, debido a que, en la directiva anterior, los certificados utilizados para la creación de la firma debían estar custodiados por el propio firmante o por una persona autorizada, lo que en la práctica implicaba que la firma electrónica debía realizarse de forma presencial.

Tal y como se observa en la siguiente imagen, en el año 2010 aproximadamente un cuarto de la población española contaba con algún medio de certificado de firma electrónica[6], de los cuales el 4% utilizaba el DNIe y el 13% algún otro tipo de certificado.



1 - Uso de certificados de firma electrónica en las relaciones de particulares con las AA.PP. a través de Internet. Año 2010. (rojo: DNIe, amarillo: otros certificados). Fuente: Instituto Nacional de Estadística[7]

[4] https://ec.europa.eu/info/law/law-making-process/types-eu-law_es#tipos-de-actos-jurdicos-de-la-ue

[5] Los requisitos de los prestadores cualificados se describen en el artículo 24 del Reglamento 910/2014 del Parlamento Europeo y del Consejo de 23 de julio de 2014 (eIDAS).

[6] https://www.ine.es/jaxi/Datos.htm?path=/t25/p450/a2010/l0/&file=08028.px

[7] https://www.ine.es/jaxi/Tabla.htm?path=/t25/p450/a2010/l0/&file=08028.px&L=0

Normalmente, a nivel de particulares y empresas, la firma de documentos electrónicos se ha realizado a través de aplicaciones instaladas en el ordenador utilizado por el firmante, como por ejemplo la aplicación Autofirma, de la Junta de Andalucía[8]. Estas aplicaciones, son lanzadas directamente por el usuario para firmar los documentos necesarios, o son lanzadas desde un sitio web que requiera un documento firmado, como por ejemplo la entrada de una solicitud en el registro electrónico de cualquier AA.PP. Habitualmente, tanto la firma a utilizar como los documentos a firmar están almacenados en el ordenador o sistema del firmante.

El escenario descrito en el párrafo anterior, que ha sido el escenario de uso más habitual en el uso de la forma electrónica, ha sido la causa de varios problemas, por ejemplo:

- **Incompatibilidades entre el navegador y la aplicación. Muchas de las aplicaciones de firma digital están realizadas en Java, ya que es** una tecnología bastante estandarizada y popularizada. Sin embargo, con el tiempo las tecnologías y técnicas utilizadas por los navegadores y por Java han ido cambiando, por lo que muchas aplicaciones de firma han dejado de funcionar siendo imposible realizar la firma. En varios casos ha sido necesario recurrir a instalar versiones antiguas de un navegador, de java (o incluso de un sistema operativo) únicamente para poder firmar digitalmente un documento para presentarlo ante un organismo. Esto implica el tener problemas de seguridad por el uso de versiones desfasadas de las aplicaciones.
- **Uso de un sistema que no es propio**. La popularización del acceso a internet a través de teléfonos móviles, tablets y otros medios (Smart TV, consolas de videojuegos u otros dispositivos), ha llevado a que a día de hoy no sea necesario disponer de un ordenador para realizar la mayoría de las acciones que habitualmente se pueden realizar en Internet. Esto puede conducir a que haya usuarios que no dispongan de un ordenador desde el que puedan firmar digitalmente, al no poder instalar las aplicaciones habituales. Algunas empresas y administraciones lo han resuelto publicando versiones de las aplicaciones para los sistemas operativos móviles más habituales. Los usuarios de sistemas para los que no existe una versión deben de utilizar un sistema prestado, compatible, en el que instalar el certificado con el que se realiza la firma, con el riesgo de suplantación de identidad que ello conlleva.
- **Multiplicación de versiones de las aplicaciones**. Al existir múltiples sistemas operativos es necesario mantener diferentes versiones de la misma aplicación, con el consecuente aumento de los recursos

[8] https://ws024.juntadeandalucia.es/ae/adminelec/areatecnica/autofirma

necesarios para hacerlo. Esto puede llevar a un mayor gasto de los recursos humanos o económicos, o a la desatención de las versiones menos utilizadas de las aplicaciones, publicando versiones más esporádicamente o con menos funcionalidades que las versiones más populares.

- **Necesidad de hardware extra.** Algunas soluciones de firma digital, como el DNI electrónico, requieren del uso de lectores de tarjetas y de la instalación de los drivers correspondientes. Puede ocurrir que un dispositivo que funciona correctamente en una versión de un sistema operativo ya no funciona en la siguiente porque el fabricante haya dejado de dar soporte al dispositivo y no haya desarrollado los drivers correspondientes para la nueva versión. Estos lectores normalmente utilizan una conexión USB, por lo que son difíciles de conectar a los dispositivos móviles[9].

1.2. Objetivos del Trabajo

El objetivo principal del TFM es el desarrollo de un mecanismo de firma digital que resuelva los problemas mencionados en el apartado anterior, además de simplificar el proceso de firma, cumpliendo en todo momento con el reglamento eIDAS. El nuevo reglamento, y su transposición a la legislación nacional en la Ley 6/2020, de 11 de noviembre, regula determinados aspectos de los servicios electrónicos de confianza[10], creando la figura del "prestador cualificado de servicios de confianza" que puede gestionar los datos de la firma electrónica en nombre del firmante, por lo que se pueden utilizar servicios externos y confiables para crear, almacenar y utilizar la firma electrónica.

Los objetivos secundarios, que llevarán al cumplimiento principal son:

- **Eliminar o reducir al mínimo la necesidad de software y hardware para el usuario de la firma digital**. La solución desarrollada utilizará al máximo posible servicios y software existentes en la actualidad, eliminando la dependencia de hardware externo y reduciendo al mínimo la necesidad de instalar software de parte del usuario para poder firmar

[9] Si bien se pueden conectar y utilizar dispositivos USB en bastantes de los dispositivos móviles presentes en el mercado, normalmente es necesario utilizar adaptadores, aunque dependerá del soporte de las aplicaciones y del sistema operativo del dispositivo el que se puedan utilizar los dispositivos conectados.

[10] https://www.boe.es/eli/es/l/2020/11/11/6/con

digitalmente los documentos. Para ello se harán uso de soluciones basadas en la nube, tanto para el almacenamiento de los documentos como para la creación y uso de la firma digital. Se pretende que cualquier usuario pueda utilizar el sistema teniendo solamente un dispositivo con conexión a Internet y un navegador actual como software.

- **Usar arquitecturas, tecnologías y protocolos de comunicación estándar.** La solución de firma digital hará uso de protocolos de comunicación estándar para comunicarse con los diferentes servicios utilizados (almacenamiento, proveedor de servicios de firma digital, servicios de autenticación, etc.), usando protocolos REST para comunicarse con los servicios, el protocolo Oauth2 para la autenticación, etc.
- **Existencia de una única versión de la solución.** Se pretende eliminar la multiplicidad de versiones, reduciendo el código del sistema de firma a una única versión que tendrá adaptaciones menores para que sea accesible desde los diferentes dispositivos con los que se puede acceder a la aplicación.
- **Control del usuario sobre los datos.** El sistema de firma digital, aunque esté implementado usando servicios en la nube, debe darle al usuario la libertad de escoger en que servicio de almacenamiento quiere almacenar sus datos y que servicios de firma utilizar, para que de esta forma siempre tenga el control sobre los datos.

Se podrían plantear algunos objetivos terciarios que ampliaran el alcance del proyecto, como por ejemplo, que el diseño del sistema sea completamente modular, para así poder ampliarlo fácilmente, añadiendo más servicios de almacenamiento, autenticación o de firma digital, incluso usar servicios locales si las leyes o normativas así lo exigen, por ejemplo, en un entorno bancario no se puede utilizar el almacenamiento en clouds públicas, pero si en clouds privadas, además de tener su propio sistema de autenticación, por lo que sería necesario ampliar el sistema con nuevos módulos. Estos objetivos terciarios, como el del ejemplo, se realizarán únicamente si el tiempo disponible para la finalización del TFM lo permite.

1.3. Enfoque y método seguido

Para la consecución de los objetivos del trabajo se seguirá una metodología de dividir el trabajo en diferentes fases y etapas, cada una de ellas con sus objetivos específicos. De esta forma se simplificará el trabajo,

ya que cada fase estará centrada en una parte específica del TFM. Estas fases son incrementales, siendo necesario finalizar cada una de ellas el material antes de poder avanzar a la siguiente.

Fase 1: Análisis y diseño

Se analizarán tanto los objetivos que se quieren cumplir como los problemas que se pretenden resolver con el desarrollo del sistema. Esto permite definir los requisitos que debe satisfacer el sistema.

También se diseñará la arquitectura del sistema, tanto del software a desarrollar como del sistema en el que funcionará (servidores de aplicaciones, web, bases de datos, sistemas operativos, etc.). Será necesario analizar las entradas y salidas que tendrá el sistema, casos de uso, interfaz de usuario, etc.

Fase 2: Planificación

En base al resultado del análisis y diseño se realizará una estimación del tiempo y recursos necesarios para el desarrollo del sistema, así como una estimación general de los recursos necesarios para un uso normal del sistema.

Fase 3: Estudio de tecnologías y protocolos.

En esta fase se estudian las diferentes tecnologías y protocolos que se pretenden usar en el desarrollo del sistema para así decidir si las tecnologías escogidas permiten desarrollar la solución, o si es necesario escoger otras tecnologías o protocolos. Para ellos se desarrollarán pruebas de concepto (por ejemplo, conexión a un proveedor de almacenamiento cloud, obtención de documentos almacenados y creación de nuevos documentos, etc.) que permitan comprobar si estas tecnologías cumplen los requisitos necesarios para resolver los problemas observados.

Fase 4: Desarrollo del producto y puesta en producción

Esta fase consiste en el desarrollo del código necesario para la puesta en marcha del sistema. En ella se integrarán los servicios de los diferentes proveedores escogidos, de forma que el usuario observe un funcionamiento similar de la aplicación, independientemente de los proveedores utilizados.

En esta fase se realizarán tanto pruebas unitarias como del sistema en conjunto para comprobar que el funcionamiento del sistema es correcto.

Si el tiempo disponible lo permite, en esta fase se pretende realizar una puesta en producción de la solución, es decir, dejarla preparada para funcionar en un entorno diferente al de desarrollo. Este entorno puede ser privado o público.

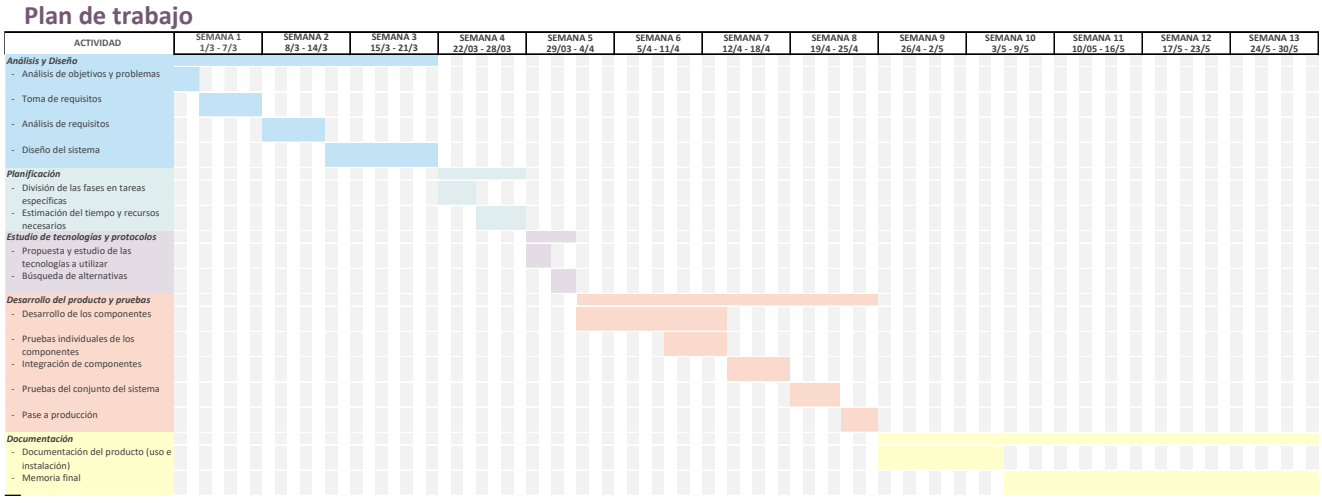
Fase 5: Documentación

Fase en la que se creará la documentación necesaria para el proyecto, tanto para el uso como para la puesta en marcha de la solución, así como la memoria del propio TFM.

1.4. Planificación del Trabajo

Se establece una primera planificación temporal tomando como base las fases en las que se ha dividido el proyecto. En una primera etapa, estas fases se han dividido en subtareas que se dividirán en tareas más específicas si se considera necesario, o se eliminarán si no son necesarias, por ejemplo, la tarea de búsqueda de tecnologías o protocolos alternativos se puede eliminar si se determina que las tecnologías utilizadas son adecuadas para la realización del proyecto.

El siguiente diagrama de Gantt presenta la primera propuesta de la planificación temporal del proyecto. La división de tareas propuesta se ajustará al desarrollo real del proyecto, por lo que, si una tarea o fase se completa en menos tiempo del estimado, ese tiempo se aprovechará en alguna de las tareas siguientes, bien adelantándola o bien ampliando su duración.



2 - Planificación temporal del trabajo

1.5. Breve sumario de productos obtenidos

El producto obtenido consiste en la aplicación web cloudDocs, que permitirá la firma electrónica de documentos PDF alojados en servicios de almacenamiento cloud, como Google Drive o Dropbox. Para la firma de los documentos se utilizarán el servicio de firma electrónica cloud Entrust TrustedX, que ofrece servicios de QTSP, lo que le permite realizar la firma cualificada de documentos.

2. Arquitectura y Diseño

2.1. Requisitos del Sistema

Los requisitos del sistema “expresan las necesidades y las restricciones que afectan a un producto de software que contribuye a la solución de un problema del mundo real y nos sirven para delimitar qué posibles soluciones son adecuadas para el problema (las que cumplen los requisitos) y cuáles no.”[11]

En base a los objetivos que ha de cumplir y las necesidades que ha de cubrir la aplicación, se definen los requisitos del sistema. En los requisitos se definirán aquellas características que debe de cumplir la aplicación, y son los que servirán como base para la arquitectura del sistema.

1. El objetivo de la aplicación es realizar la firma electrónica de documentos PDF. Se seguirá el estándar de firma PaDES.
2. La firma electrónica debe seguir las directrices impuestas por el reglamento eIDAS.
3. En la medida de lo posible no tendrá que ser necesario instalar ningún software en el dispositivo que el usuario utilizará para acceder a la aplicación. El sistema se basará en una aplicación web, por lo que el usuario únicamente deberá disponer de un dispositivo con un navegador web moderno.
4. La firma electrónica de los documentos se realizará utilizando un servicio externo de firma electrónica (TrustedX). El usuario deberá disponer de una cuenta en el servicio de firma.
5. Los documentos a firmar se alojarán en servicios de alojamiento externos (Google Drive, Dropbox, etc.). El usuario deberá disponer de cuenta en los servicios de alojamiento que quiera utilizar.
6. La comunicación, tanto entre el usuario y la aplicación como entre la aplicación y los proveedores de servicios, debe de ser privada y segura.

[11] Módulo 3 – Requisitos – Página 7 – Apuntes de Ingeniería del software - Jordi Pradel Miquel y Jose Raya Martos

7. No se enviará el documento completo al servicio de firma. El servicio firmará un hash resumen del documento, que es el que se integrará con el documento PDF correspondiente.
8. El documento firmado se guardará en el servicio de almacenamiento original, con un nuevo nombre para no sobrescribir el fichero original.
9. El usuario podrá validar un documento firmado que tenga guardado en el servicio de almacenamiento.
10. Se requerirá la autenticación del usuario para poder utilizar la aplicación.

2.2. Requisitos del Reglamento eIDAS

El reglamento eIDAS, una de las bases del presente trabajo, establece una serie de requisitos que deben cumplir los prestadores de servicios de confianza, los servicios de confianza prestados por estos, los productos generados por estos servicios, y los dispositivos utilizados para prestar los servicios y generar los productos. Si bien el reglamento es detallado en los servicios de confianza existentes, el presente apartado se centrará en los requisitos de aquellos servicios, productor y dispositivos necesarios para el presente trabajo, es decir, todo lo necesario para la creación de las firmas electrónicas y de los certificados usados para la firma electrónica, así como la preservación de los certificados electrónicos.

Servicios de confianza

Según el propio reglamento, en el Artículo 3, ap. 16 y 17, un "servicio de confianza" es un servicio electrónico que permite:

"a) la creación, verificación y validación de firmas electrónicas, sellos electrónicos o sellos de tiempo electrónicos, servicios de entrega electrónica certificada y certificados relativos a estos servicios, o

b) la creación, verificación y validación de certificados para la autenticación de sitios web, o

c) la preservación de firmas, sellos o certificados electrónicos relativos a estos servicios;"

Estos servicios de confianza serán cualificados cuando además cumplan con los requisitos del reglamento. En el caso de los servicios de creación de firmas, para que la firma sea cualificada debe ser una firma avanzada que además se haya creado con un dispositivo cualificado de creación de firma. Según el reglamento, en el Art. 26, la firma electrónica avanzada debe:

a) Ser única para la persona que firma,
b) Permitir identificar a la persona que firma,
c) Haberse creado con datos que solo el firmante puede utilizar, bajo control del firmante, y
d) Estar relacionada de tal manera con los datos firmados, de forma que se puedan detectar modificaciones posteriores a la firma.

A su vez, los certificados de firma electrónica, para ser cualificados, deben haber sido emitidos por un prestador cualificado de servicios de confianza, y cumplir con los requisitos mencionados en el Art. 28 y en el anexo I del reglamento. De forma resumida, estos requisitos establecen los datos mínimos que debe contener un certificado cualificados, que son: aquellos datos que identifican al emisor del certificado, los datos del firmante, los datos necesarios para determinar la validez del certificado.

Prestadores de servicios de confianza

Tal y como su nombre expresa, estos son aquellas personas físicas o jurídicas que prestan uno o más servicios de confianza. El reglamento establece los requisitos que debe cumplir un prestador de servicios para ser considerado como cualificado. De forma resumida, los requisitos son[12]:

a) Ser auditados, de forma periódica o puntual, por un organismo de evaluación de la conformidad, para determinar si cumplen o no con los requisitos del reglamento.
b) Haber notificado al organismo de supervisión la intención de prestar servicios cualificados, haber sido verificados por el organismo para determinar si cumple o no con los requisitos del reglamento, y que el organismo de verificación le haya concedido la cualificación para prestar los servicios.
c) Para expedir los certificados cualificados, el prestador debe comprobar la identidad de la persona física o jurídica para la que emite el certificado.
d) Una serie de requisitos que determinan tanto las responsabilidades civiles, legales y económicas de los prestadores, la información que deben transmitir a los usuarios de los servicios acerca de las condiciones de uso de los servicios, la información a transmitir al organismo de supervisión acerca de los cambios en los servicios prestados. También se especifica la necesidad de tomar las medidas necesarias para proteger los sistemas contra toda alteración y garantizar la seguridad, almacenar los datos de forma fiable y

[12] Estos requisitos vienen detallados en los artículos 20, 21 y 24 del Reglamento eIDAS.

verificable, así como tomar las medidas necesarias para evitar falsificaciones y robo de datos. También se ha de hacer previsión para que la información relacionada con los certificados y firmas emitidos sea preservada, aunque cese la actividad de la empresa.

e) Mantener actualizada la base de datos de certificados revocados, añadiendo aquellos certificados de los que se solicita la revocación, y publicar el estado de la revocación del certificado.
f) Proporcionar, de forma fiable, gratuita y eficiente, la información necesaria para determinar la validez o revocación de los certificados cualificados emitidos. Esta información se debe proporcionar a cualquier usuario de los certificados.

Dispositivo de creación de firma electrónica (QSCD[13])

Un dispositivo de creación de firma electrónica es aquel sistema o programa informático usado para crear una firma electrónica. Este dispositivo se considerará como cualificado cuando cumpla con los requisitos que se especifican en el anexo II del reglamento. Estos dispositivos deben:

a) Garantizar la confidencialidad y unicidad de los datos usados para la creación de la firma electrónica.
b) Los datos usados para crear la firma no se pueden descubrir por deducción y la firma ha de estar protegida contra la falsificación.
c) Los datos de creación de la firma pueden protegerse por el firmante contra su uso por otras personas.
d) Los dispositivos utilizados para crear la firma no alterarán los datos a firmar y deben dar la opción de mostrar esos datos antes de firmarlos.
e) La generación o gestión de los datos de la firma electrónica, en nombre del firmante, se realizará solo por un prestador de servicios de confianza.
f) Los prestadores cualificados de servicios de confianza pueden duplicar los datos de creación de firma electrónica solo para realizar copias de seguridad, siempre y cuando estas copias tengan el mismo nivel de seguridad que los datos originales, y que el número de copias sea el mínimo necesario para garantizar la continuidad del servicio.

[13] Por sus siglas en inglés: Qualified (electronic) Signature Creation Device.

2.3. Arquitectura del Sistema

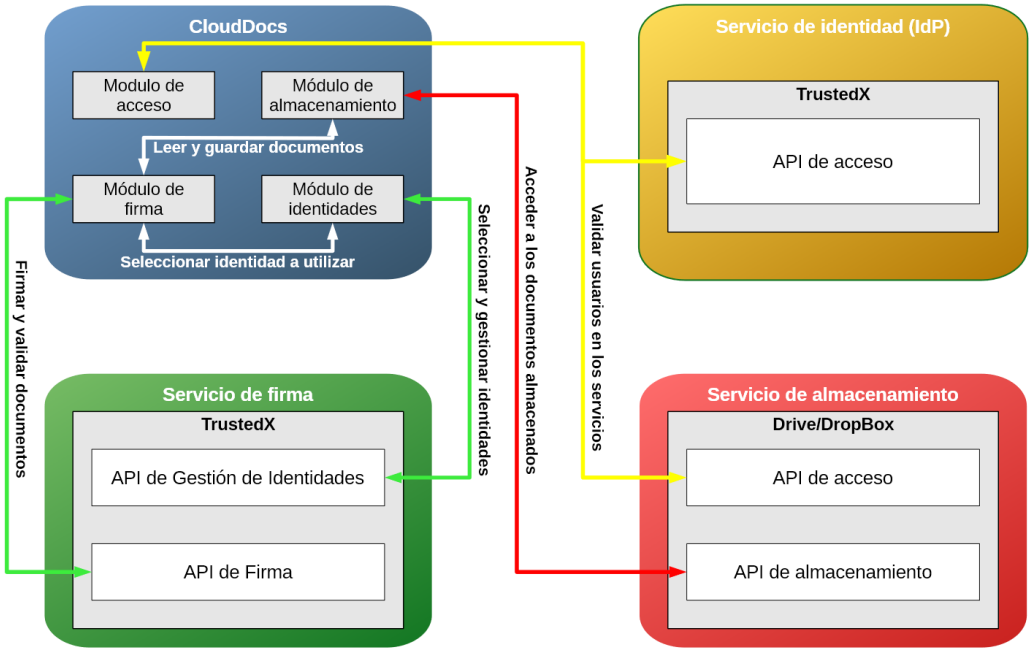
La aplicación para firma electrónica de documentos almacenados en la nube, CloudDocs, está conformada por diferentes módulos que integran y comunican los diferentes servicios -internos y externos- que se utilizan. La aplicación está compuesta por cuatro componentes. Cada uno de estos componentes se usa para una función específica, lo que simplifica el desarrollo al limitar las funcionalidades de cada componente.

Los componentes de la aplicación son:

- **Componente CloudDocs**: Componente principal de la aplicación, encargado de gestionar el flujo del funcionamiento de la aplicación, comunicando con los otros módulos en el orden adecuado para cumplir con la funcionalidad requerida, la firma y validación de documentos almacenados en la nube.
- **Componente del proveedor de identidad (IpD):** Componente que gestiona la comunicación con el IdP seleccionado, para así realizar la autenticación de los usuarios y permitir el acceso a los recursos y servicios necesarios para la firma y validación de documentos.
- **Componente del proveedor de firma (TrustedX):** Componente que proporciona acceso a los servicios prestados de firma y a los recursos necesarios para realizar la firma y validación de los documentos. El componente permitirá realizar la firma y validación de documentos, y la gestión de las identidades de firma. El servicio de firma electrónica utilizado en este trabajo es TrustedX.
- **Componente del proveedor de almacenamiento:** Este componente gestiona el acceso a los documentos almacenados en el proveedor escogido de almacenamiento externo.

La separación de funcionalidades en diferentes módulos permite que se pueda ampliar la funcionalidad de un módulo sin afectar al funcionamiento de los demás módulos. Por ejemplo, se podrían añadir nuevos proveedores de almacenamiento (por ejemplo, OneDrive de Microsoft o S3 de Amazon, entre otros) sin necesidad de modificar el funcionamiento del módulo del proveedor de firma y de la aplicación en general.

CloudDocs se desarrolla siguiendo el patrón de arquitectura Modelo-Vista-Controlador (MVC). Esta arquitectura permite separar los datos utilizados -el modelo- del flujo o tratamiento de los datos -el controlador- y de la representación de los mismos -la vista-.



3 - Arquitectura del sistema - Componentes

1. Modelo - Arquitectura de los datos

La aplicación utiliza diferentes tipos de datos a lo largo del tiempo. Estos datos se organizan en diferentes categorías según su tipo y utilidad:

- **Datos de autenticación y autorización**: Datos necesarios para el acceso a la plataforma de firma y a los servicios de almacenamiento, para el uso de los servicios proporcionados por las plataformas y de los documentos almacenados en las mismas. Entre estos datos se encuentran las credenciales necesarias para acceder a las plataformas, así como los tokens (valores) devueltos por los diferentes servicios y que sirven para autentificar los siguientes accesos a los servicios.
- **Datos de firma:** Son los datos que componen las identidades de firma (claves públicas y privadas) que se utilizarán para firmar y validar los documentos. También se incluyen en este grupo los *hashes* resumen de los documentos que se desea firmar.
- **Documentos:** Documentos a firmar o validar, almacenados en los servicios de almacenamiento externos.

La mayor parte de los datos utilizados son administrados por los servicios externos (servicio de firma y de almacenamiento), en algunos casos la aplicación no tiene acceso a los mismos, como ocurre con las credenciales

de acceso a los servicios externos, que están gestionadas directamente por los proveedores de los servicios. En estos casos CloudDocs solo tendrá acceso a los tokens devueltos por los servicios tras autenticarse el usuario en estos. Por este motivo, la aplicación solo tiene que realizar un almacenamiento temporal de aquellos datos que serán necesarios para realizar su función, eliminándose estos tras la finalización de la tarea. No es necesario el uso de una base de datos para este almacenamiento temporal, pudiéndose guardar la información en memoria o en ficheros temporales que se eliminen tras su uso.

2. Controlador – Arquitectura del flujo de información

Los controladores corresponden con aquellas partes de la aplicación que responden a las peticiones de los usuarios, de los módulos del sistema y de los servicios externos. Gestionan el flujo de información entre los diferentes módulos y realizan las acciones necesarias para acceder a los documentos y realizar la firma o validación de los mismos, utilizando para ello las APIs de los servicios externos.

Los controladores más importantes son:

- **Principal**: Encargado de gestionar el flujo normal de la aplicación (navegación por el contenido del servicio de almacenamiento y selección de los documentos a firmar o validar)
- **AutenticacionUsuario**: Controlador que gestiona el correcto acceso a la aplicación y la autenticación del usuario, manteniendo los registros necesarios para mantener la sesión del usuario en la aplicación
- **Almacenamiento**: Controlador que maneja el acceso a los servicios de almacenamiento externo, obteniendo los contenidos de los directorios, el contenido de los documentos, y la creación o borrado de nuevos documentos. Este controlador está compuesto de tantos módulos como servicios de almacenamiento se utilicen.
- **Firma**: Controlador que gestiona el proceso de firma y validación de los documentos, comunicándose con el servidor de TrustedX.
- **IdentidadesFirma**: Controlador que administra el acceso, la creación y el borrado de las identidades de las identidades de firma.

3. Vistas – Interfaz con la aplicación

Las vistas corresponden con la parte del sistema a través del cual el usuario puede interactuar con la aplicación, seleccionando los documentos a

firmar o validar, gestionando las identidades de firma, o utilizando las mismas para validad o firmar los documentos, etc.

En este proyecto en particular, las vistas de la aplicación coinciden con las diferentes partes del interfaz de usuario, por los que el desarrollo de este apartado se realizará en el apartado *Interfaz de usuario* del presente documento.

2.4. Protocolos

Tal y como se ha detallado en los requisitos y en la descripción de la arquitectura, CloudDocs está formada por módulos que integran diferentes servicios para así poder cumplir con el objetivo de poder firmar electrónicamente documentos pdf sin necesidad de instalar ningún software en los dispositivos del usuario, más allá de un navegador web relativamente actual. Los servicios utilizados en la implementación actual de la aplicación, son externos a la propia aplicación, provistos cada uno por diferentes proveedores. El acceso a estos servicios se realiza a través de Internet, por lo que es necesario que las comunicaciones entre la aplicación y los servicios sean fiables y seguras, de forma que se pueda asegurar la privacidad de los datos transmitidos y recibidos, así como garantizar que los servicios no sean utilizados por personas y/o aplicaciones no autorizadas. Para conseguir estos objetivos existen diferentes protocolos que han llegado a ser un estándar en la industria.

1. HTTPS

La seguridad y privacidad en las comunicaciones se garantiza a través del Protocolo Seguro de Transferencia de Hipertexto, HTTPS[14] (por las siglas en inglés de Hypertext Transfer Protocol Secure).

El protocolo HTTPS añade una capa de encriptación al protocolo HTTP, usando para ello SSL/TLS. Esto crea un canal seguro en una red insegura. Es un protocolo bastante seguro, aunque puede llegar a ser vulnerable a ataques de análisis de texto y a algunos ataques man-in-the-middle[15]. En cualquier caso, los algoritmos utilizados para codificar la información se han ido mejorando a lo largo del tiempo, desarrollándose versiones más seguras y corrigiendo las diferentes vulnerabilidades que se han ido descubriendo. Este

[14] https://en.wikipedia.org/wiki/HTTPS

[15] https://en.wikipedia.org/wiki/HTTPS#Limitations

protocolo ha llegado a ser un estándar en la industria, estando implementado en la práctica totalidad de los navegadores web utilizados actualmente (Firefox, Safari, Chrome y navegadores basados en el mismo motor).

Este protocolo se utiliza como base para la comunicación entre el usuario y la aplicación, así como entre la aplicación y los servicios externos. Por lo tanto, teniendo en cuenta las características del protocolo, se puede considerar que las transferencias de información entre los componentes de la aplicación (propios e integrados) serán seguras.

2. REST

"*REST no es un protocolo o un estándar, sino que se trata de un conjunto de principios de arquitectura.*"[16]. Estos principios ayudan a crear protocolos y estándares de comunicación para acceder a los recursos proporcionados por un servicio a través de la red. El aplicar estos principios al protocolo HTTP ha generado un estándar en el diseño y publicación de las APIs que se usan para acceder a los servicios. Las APIs que siguen los principios REST y están basadas en HTTP se les llama API RESTful.

Aunque existen otros protocolos para el uso de APIs y la comunicación con los servicios, como SOAP, estos suelen ser más pesados y complejos que un protocolo basado en REST, ya que normalmente en una API RESTful solo se implementan aquellas pautas y principios REST que son necesarios, por lo que suelen más ligeras y rápidas que otras APIs basadas en otros protocolos. Esta cualidad es la que ha permitido que las APIs REST sean ideales para el IoT y las aplicaciones de dispositivos móviles.

3. OAuth2

OAuth 2.0 es "un protocolo de autorización, enfocado en la simplicidad de desarrollo para las aplicaciones que utilicen el protocolo mientras que permite un control muy específico de las autorizaciones para aplicaciones web, de escritorio, móviles y de otros dispositivos"[17].

Este protocolo resuelve los problemas asociados al acceso a recursos protegidos en el modelo clásico cliente-servidor usando las credenciales del usuario. En este modelo, normalmente las aplicaciones que acceden a los recursos deben solicitar y almacenar las credenciales del usuario para accesos futuros, los propietarios de los recursos no pueden limitar ni revocar

[16] https://www.redhat.com/es/topics/api/what-is-a-rest-api (Apartado REST)

[17] Traducción libre de https://oauth.net/2/.

el acceso a los recursos, además de que no hay forma de evitar que las credenciales almacenadas se puedan compartir con terceras personas[18]. Para evitar estos problemas, el protocolo introduce una capa de autorización, separando los roles de cliente y de propietario del recurso.

El protocolo se basa en el uso de un *token*, o código, que el cliente debe obtener y con el que tendrá acceso a los recursos protegidos. Este token tiene unos privilegios asociados (lectura, modificación, creación, etc.), que son los privilegios con los que accederá el cliente a los recursos. El token tiene una validez que -normalmente- está limitada en el tiempo. Una vez que el token haya caducado, es necesario que el cliente solicite un nuevo token. Se evita así el uso constante de las credenciales del usuario para acceder a los recursos, así como la necesidad de almacenar las credenciales del usuario para acceder a los recursos.

El protocolo define cuatro roles, que interaccionan entre sí para permitir o denegar el acceso a los recursos, y con qué permisos se puede acceder a los recursos. Los roles definidos son:

- **Propietario del recurso**: Entidad que garantiza el acceso a los recursos protegidos.
- **Servidor de autorización**: Servidor que proporciona *tokens* (códigos) al cliente, después de que este se haya autenticado correctamente frente al propietario del recurso.
- **Servidor de recursos**: Servidor que hospeda el recurso protegido, responde a aquellas peticiones de acceso a los recursos protegidos que usan los tokens para autenticarse.
- **Cliente**: Aquellas aplicaciones que realizan peticiones a los recursos protegidos, en nombre y con la autorización del propietario del recurso.

Estos cuatro roles interaccionan entre sí en el siguiente orden:

1) El cliente solicita la autorización al propietario del recurso. Esta autorización la puede realizar directamente el propietario del recurso, o puede realizarse a través del servidor de autorización.
2) El cliente recibe una concesión de autorización, que representa la autorización del propietario del recurso.
3) El cliente solicita un token de acceso al servidor de autorización. Para ello presenta la concesión de autorización.

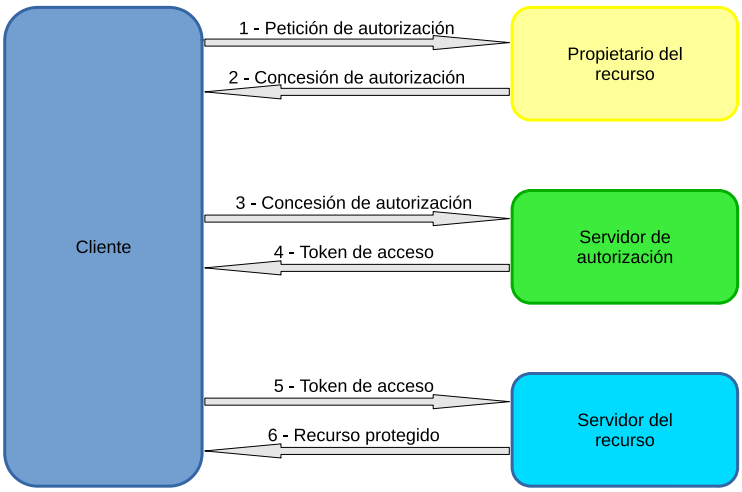
[18] Sea la compartición de las credenciales hecha a propósito o resultado de una fuga de datos.

4) El servidor de autorización autentica al cliente y valida la concesión de autorización. Si esta es válida, entonces le da al cliente un token de acceso.

5) El cliente solicita el recurso protegido al servidor de recursos, autenticándose con el token de acceso.

6) El servidor de recursos valida el token de acceso y, si el token es válido, devuelve el recurso.

El siguiente diagrama representa la interacción entre los cuatro roles del protocolo y el orden en que lo hacen.



4 - Flujo del protocolo Oauth2

4. OpenID

OpenID "es un estándar abierto y descentralizado de identificación digital, con el que un usuario puede identificarse en uno o más servicios con las mismas credenciales, utilizando un tercer servicio independiente"[19]. Este estándar elimina la necesidad de que los servicios mantengan una lista de usuarios y el sistema de autenticación asociado.

El estándar OpenID está basado en Oauth y lo complementa, ya que Oauth2 solo proporciona los medios para dar la autorización para acceder a los recursos, pero no proporciona los medios para autenticar al usuario[20]. Esto último se realiza a través de OpenId[21].

---

[19] Traducción libre de https://en.wikipedia.org/wiki/OpenID

[20] Si bien se puede realizar la autenticación solo usando el protocolo Oauth2, el proceso es más complejo e inseguro que con OpenID.

[21] https://en.wikipedia.org/wiki/OpenID#OpenID_vs._pseudo-authentication_using_OAuth

OpenID es un estándar ampliamente aceptado dentro de la industria, existiendo múltiples proveedores de identidad que lo utilizan[22].

Aunque CloudDocs no utiliza ningún proveedor OpenID para autentificar a los usuarios, DropBox y Google Drive sí que utilizan el protocolo para autenticar los accesos al servicio de almacenamiento, dándose el caso que se puede acceder a DropBox (y otros servicios) utilizando cuentas de usuario de Google y de otros proveedores de autenticación.

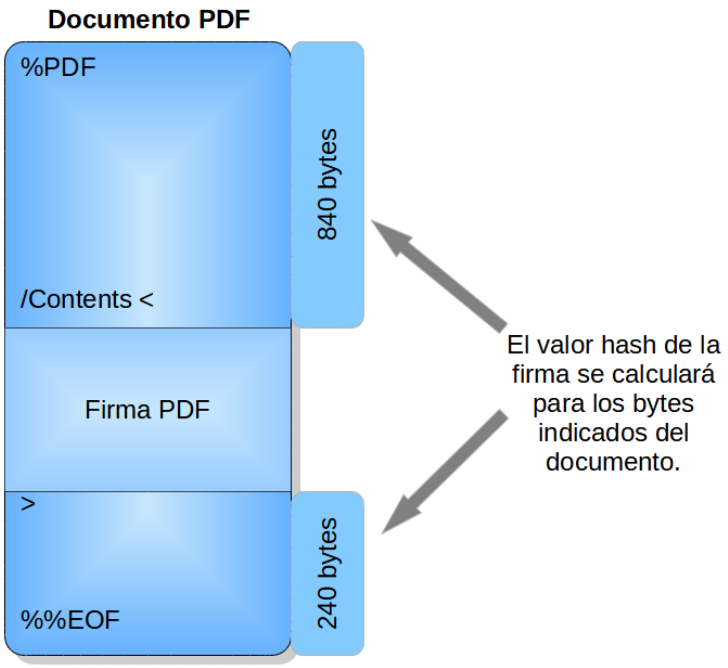
2.5. Estándar PAdES

El formato de la firma electrónica de los documentos PDF, especificado en el estándar **PAdES** (PDF Advanced Electronic Signature[23]), se encuentra dentro del marco indicado por la ISO 32000-1 para la firma electrónica de documentos PDF. Publicado por ETSI en 2009, el estándar PAdES ha sido ampliado en 2016, siendo publicado con el código EN 319 142 (1 y 2). Este estándar especifica el formato de la firma electrónica de forma que sea compatible con los estándares de firma electrónica de ETSI, siguiendo las normativas del reglamento **eIDAS**. Tanto el reglamento eIDAS como la implementación de ETSI permiten que la firma electrónica pueda ser tanto avanzada (**AES**) como cualificada (**QES**).

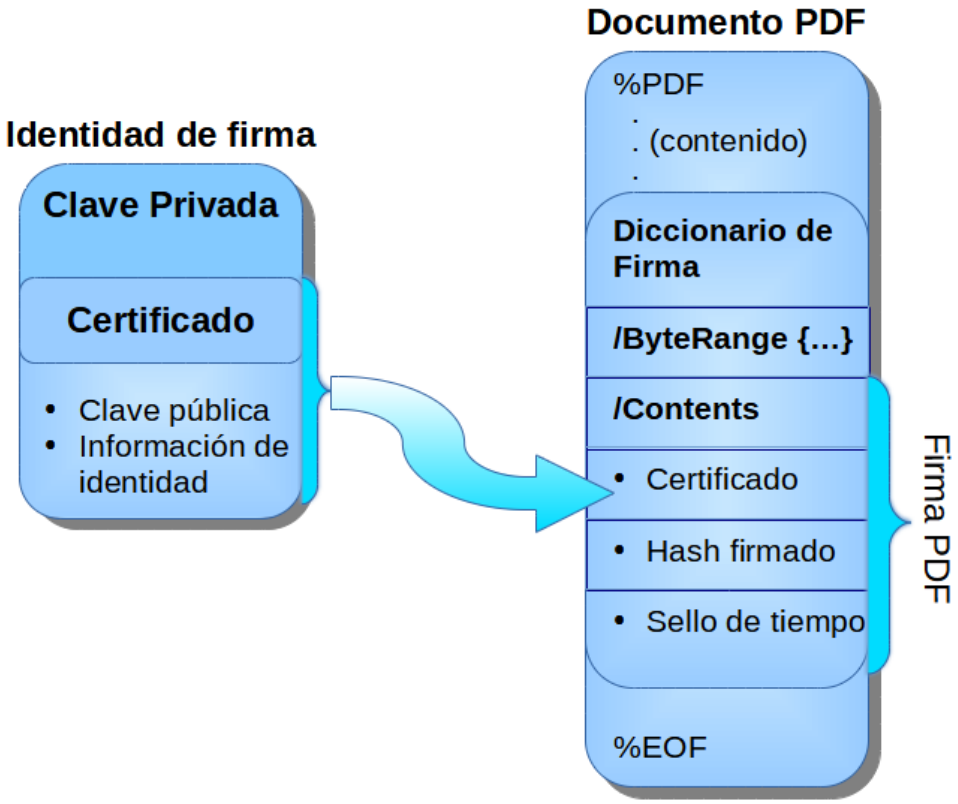
El formato de los documentos PDF, tal y como está definido por el estándar ISO 32000:1, está definido por una serie de secciones en las que se definen los contenidos del documento. El estándar define una sección llamada “Diccionario de firmas” que contendrá la firma del documento PDF, tal y como se muestra en la figura nº 6. La firma estará ubicada en la entrada “Contents” del diccionario.

[22] https://en.wikipedia.org/wiki/OpenID#Adoption

[23] O también Firma Electrónica Avanzada en PDF, por su traducción al castellano.



5 - Esquema del formato PDF incluyendo la firma electrónica



6 - Esquema de la identidad de firma y de la Firma PDF dentro del documento

En el caso particular de **CloudDocs**, la aplicación añadirá una nueva entrada *Contents*, que contendrá la firma del valor hash calculado para el resto del documento. En la siguiente imagen se representa, de forma esquemática

el contenido de la identidad de firma, con la clave privada y pública, así como la información de la identidad del firmante. Esta identidad de firma, almacenada en los servidores de **TrustedX**, utiliza para firmar el hash resumen del documento. Tanto el hash firmado como la parte pública del certificado se insertan en la entrada "*Contents*" del diccionario de firma.

2.6. Estándar PKCS

En criptografía, se entiende PKCS (Public-Key Cryptography Standars) como el conjunto de estándares de criptografía de clave pública, desarrollados y publicados por RSA Security LLC[24]. Dentro de este conjunto, hay dos estándares que son de uso habitual en el campo de la firma electrónica, y que se utilizarán en el desarrollo de CloudDocs. Estos estándares son el **PKCS#1** y el **PKCS#12**

1. PKCS#1

Este estándar está definido en el RFC 8017[25], siendo la versión 2.2 de las especificaciones del estándar criptográfico de clave pública basado en el algoritmo RSA. El estándar proporciona una serie de recomendaciones para la implementación del estándar, cubriendo diversos aspectos, como las primitivas criptográficas, los esquemas de codificación, el formato de las firmas, y la sintaxis ASN.1 que se utiliza para representar las claves públicas y privadas.

En la aplicación, la firma creada seguirá el formato definido por el estándar PKCS#1, por lo que, a efectos prácticos, podemos considerar que este es un estándar que define un contenedor de firmas.

2. PKCS#12

El estándar PKCS#12, definido en el RFC 7292[26], es la versión 1.1 de las especificaciones del formato de archivo para el almacenamiento de objetos criptográficos en un solo fichero. Habitualmente se utiliza para almacenar una clave privada junto su clave pública (en formato X.509), así como los certificados que forman la cadena de confianza.

---

[24] https://en.wikipedia.org/wiki/PKCS

[25] https://datatracker.ietf.org/doc/html/rfc8017

[26] https://datatracker.ietf.org/doc/html/rfc7292

En la práctica el formato se utiliza como contenedor de claves privadas y certificados. En este sentido es en el que se utiliza en CloudDocs, usándolo como base para la importación del par de claves público/privada que se utiliza para dar de alta la identidad de firma.

2.7. Flujos de Información

En este apartado se describen los flujos que sigue la información entre los diferentes servicios al realizar la firma de un documento. Estos flujos están organizados en base a la interacción entre los diferentes módulos de CloudDocs y los servicios utilizados para almacenar los documentos y realizar la firma y/o validación de los mismos. En el diagrama nº 1 se muestran los principales flujos que sigue la información entre los módulos y los servicios, aunque no se muestra la secuencia temporal de llamadas a los servicios y de intercambio de información entre estos y los módulos.

Un caso de uso paradigmático del flujo de información, en CloudDocs, es la firma de un documento, ya que implica la integración y utilización de todos los módulos y servicios disponibles. Debido a ello se utilizará para ejemplificar el flujo de información en el presente apartado[27]. En este ejemplo no se diferencia entre Google Drive y Dropbox ya que para ambos servicios se usa el protocolo Oauth2 para obtener acceso a los documentos almacenados y, además, las diferencias de funcionamiento entre los diferentes servicios se almacenamiento se han ocultado debajo de una capa que estandariza el acceso y uso de estos servicios, así como la información que se mueve entre la aplicación y los servicios de almacenamiento.

El primer paso es que el usuario inicie la sesión en la aplicación. En el diseño de CloudDocs se ha delegado la autenticación de los usuarios en el IdP de TrustedX. Al hacerlo así se garantiza que el usuario de la aplicación también es usuario de TrustedX por lo que no es necesario realizar una gestión de usuarios en la propia aplicación. CloudDocs realiza una llamada al servidor de autenticación de TrustedX para obtener el token Oauth2 que le permitirá acceder a los servicios de firma y a las identidades de firma.

Una vez que el usuario se haya autenticado en TrustedX, el usuario deberá escoger el servicio de almacenamiento donde se encuentran los

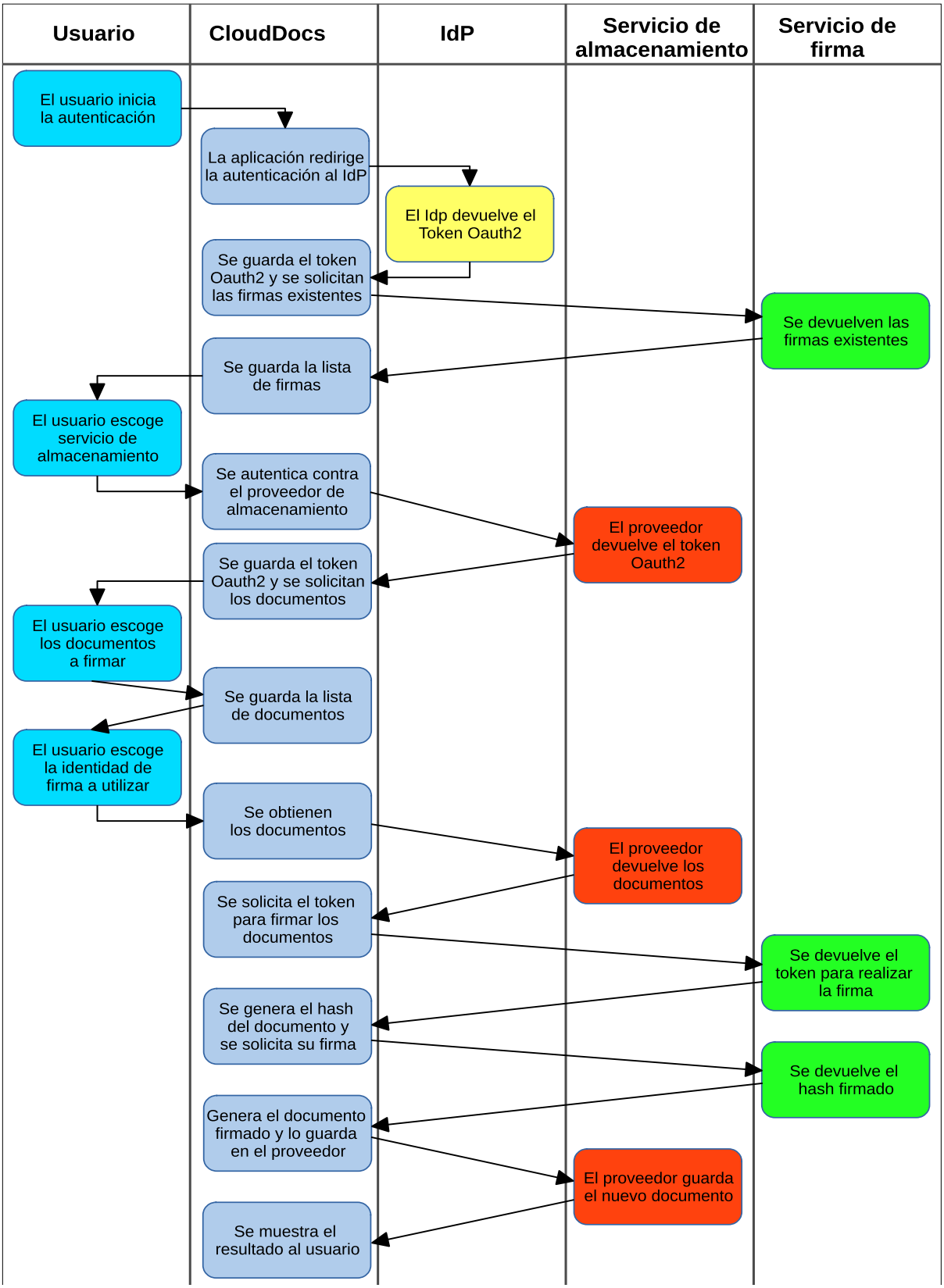
[27] Lo habitual, en la documentación resultante del desarrollo de una aplicación, es que se representen los flujos de información de todos los casos de uso. Teniendo en cuenta las limitaciones de tiempo y espacio del presente trabajo, se ha decidido representar solo los casos de uso paradigmáticos, aquellos en los que se vea el uso y la interacción de casi todos los módulos y servicios.

documentos que desea firmar. A través de la API REST del servicio seleccionado (Google Drive, Dropbox, o cualquier otro servicio integrado en la aplicación) se obtendrá un token Oauth2 que permitirá acceder a los documentos almacenados en el servicio y guardar nuevos documentos. Con el token obtenido se le mostrará al usuario la lista de documentos almacenados y podrá seleccionar los documentos a firmar.

Una vez seleccionados los documentos, el usuario escoge la identidad de firma con la que se firmarán los documentos. Usando el token obtenido al acceder a la aplicación se obtienen las identidades disponibles y se le muestran al usuario, para que escoja la firma que desea utilizar.

Con la identidad de firma escogida y los documentos seleccionados, se procede a realizar la firma de los mismos. Para ello se solicita un nuevo token Oauth2 a TrustedX que se utiliza para acceder a los servicios de firma. Una vez obtenido el token, se genera el hash de cada documento y se realiza la firma del hash a través del servicio de firma, este servicio nos devolverá la firma PKCS#1 del hash. Una vez obtenida la firma, se genera una copia del documento PDF en el que se incluirá el hash firmado. Este documento se almacena en el mismo servicio de almacenamiento donde se encuentra el fichero original, pero con nombre diferente.

El diagrama de la siguiente página refleja el flujo de información descrito en los párrafos anteriores.



7 - Flujo del proceso de firma

2.8. Casos de Uso

En este apartado se describen y analizan los posibles escenarios que indican como interactúa el sistema con el usuario y con otros sistemas para conseguir un objetivo específico. Cada caso de uso se centra en describir una funcionalidad concreta de CloudDocs y la secuencia de pasos necesarios para ejecutar esa funcionalidad, en un lenguaje sencillo.

Tras el estudio de los requisitos y de las funcionalidades de los servicios de firma y almacenamiento, se detectan ocho posibles casos de uso, que son los que se describen a continuación:

<table>
<tr><td colspan="3">CU 1 - Acceder al sistema</td></tr>
<tr><td>Precondición</td><td colspan="2">No hay</td></tr>
<tr><td>Descripción</td><td colspan="2">Pasos para acceder al sistema para utilizar sus servicios</td></tr>
<tr><td rowspan="5">Secuencia</td><td>Paso</td><td>Acción</td></tr>
<tr><td>1</td><td>El usuario accede al sitio web de la aplicación.</td></tr>
<tr><td>2</td><td>El navegador muestra la página de login.</td></tr>
<tr><td>3</td><td>El usuario introduce los datos necesarios para autenticarse en el sistema (nombre de usuario y contraseña, certificado digital, nombre de usuario y código de un solo uso, etc.)</td></tr>
<tr><td>4</td><td>Si los datos corresponden con un usuario autorizado, entonces se inicia una nueva sesión para el usuario, presentándole una página con las opciones disponibles.</td></tr>
<tr><td>Postcondición</td><td colspan="2">El usuario tiene acceso al sistema y puede utilizar las diferentes opciones disponibles.</td></tr>
<tr><td rowspan="2">Excepciones</td><td>Paso</td><td>Acción</td></tr>
<tr><td>4</td><td>Si los datos de autenticación no corresponden con un usuario autorizado, entonces se muestra una página de error.</td></tr>
<tr><td>Comentarios</td><td colspan="2">Este caso de uso representa la situación por defecto si un usuario accede al sistema y no tiene iniciada una sesión válida.</td></tr>
</table>

CU 2 - Listar documentos en almacenamiento externo		
Precondición	El usuario debe estar registrado en el sistema y debe haber iniciado una sesión. El usuario debe tener una cuenta válida en un servicio de almacenamiento externo.	
Descripción	Obtiene la lista de documentos almacenados en un servicio de almacenamiento externo.	
Secuencia	**Paso**	**Acción**
	1	El usuario selecciona, dentro de la aplicación, el servicio de almacenamiento en el que tiene alojados los documentos que quiere firmar.
	2	El sistema redirige al usuario a la página de login del servicio de almacenamiento.
	3	El usuario introduce los datos necesarios para autenticarse en el servicio de almacenamiento.
	4	El servicio de almacenamiento devuelve un token que el sistema guardará para futuros accesos al servicio.
	5	El sistema lee la lista de documentos almacenados en el servicio de almacenamiento y se la muestra al usuario.
Postcondición	Se crea una sesión en el servicio de almacenamiento que permite que la aplicación acceda a los documentos almacenados.	
Excepciones	**Paso**	**Acción**
	4	Si el servicio de almacenamiento devuelve un error, se le comunica al usuario y se indica una posible solución (revisar los datos de autenticación, escoger otro servicio de almacenamiento, etc.).
Comentarios	El usuario podrá recorrer las carpetas que tenga en el servicio de almacenamiento para que pueda localizar el documento que quiere firmar.	

CU 3 - Acceder a documentos en almacenamiento externo		
Precondición	El usuario debe estar registrado en el sistema y debe haber iniciado una sesión. El usuario debe haber iniciado sesión en un servicio de almacenamiento externo.	
Descripción	Obtiene el contenido de un documento alojado en el servicio de almacenamiento.	
Secuencia	**Paso**	**Acción**
	1	El usuario selecciona, de la lista de documentos del servicio de almacenamiento, el documento que desea firmar.
	2	La aplicación descarga el documento seleccionado del servicio de almacenamiento y lo guarda en memoria o en un fichero temporal.
Postcondición	Se guarda una copia del documento en el sistema.	
Excepciones	**Paso**	**Acción**
	2	Si el servicio de almacenamiento devuelve un error, se le comunica al usuario y se indica una posible solución (volver a autenticarse en el servicio de almacenamiento si ha caducado la sesión, revisar los permisos del documento o de la aplicación si se deniega el acceso, reintentarlo más tarde si el servicio de almacenamiento no responde).
Comentarios	Se podrán seleccionar varios documentos para ser firmados en una sola operación.	

CU 4 - Guardar documentos en almacenamiento externo		
Precondición	El usuario debe de haber iniciado sesión tanto en la aplicación como en el sistema de almacenamiento externo. También debe de haber seleccionado y firmado uno o más documentos.	
Descripción	Se guarda un documento firmado en el servicio de almacenamiento externo.	
Secuencia	**Paso**	**Acción**

	1	La aplicación crea un nuevo documento en el servicio de almacenamiento, con el contenido del documento firmado. Este nuevo documento debe de tener un nombre diferente a cualquier otro documento existente.
	2	Se lee la lista de los documentos alojados en el servicio de almacenamiento.
	3	Se muestra la nueva lista de documentos al usuario.
Postcondición	En el servicio de almacenamiento se guardará el documento firmado.	
Excepciones	**Paso**	**Acción**
	1	Si no se puede guardar el documento en el servicio de almacenamiento se mostrará un mensaje al usuario y se propondrá una posible solución.

CU 5 - Firmar documento		
Precondición	El usuario debe de haber iniciado sesión tanto en la aplicación como en el sistema de almacenamiento externo. También debe de haber seleccionado uno o más documentos en el servicio de almacenamiento. El usuario tiene que disponer de una identidad de firma, como mínimo, en el servicio de firma.	
Descripción	Se firma un documento utilizando el servicio de firma.	
Secuencia	**Paso**	**Acción**
	1	El usuario selecciona la opción de firmar el documento seleccionado.
	2	La aplicación descarga el documento en el servidor y comprueba que el documento corresponda con un tipo de válido, que sea posible firmar.
	3	La aplicación, en base al documento, genera la información que necesita el servicio de firma para realizar la firma.
	4	La aplicación envía la información al servicio de firma, que devuelve la firma correspondiente.

	5	La aplicación genera un nuevo documento que integra la información del documento original con la firma obtenida. Este documento se almacena temporalmente en el sistema.
Postcondición	Se obtiene una versión firmada del documento.	
Excepciones	**Paso**	**Acción**
	2	Si el documento no es de un tipo válido, entonces se avisa al usuario y se cancela la operación de firma para el documento no válido.
	4	Si el sistema de firma no puede realizar la firma, se avisará al usuario de la causa y se propondrá una posible solución, si existe. Si el problema no se puede solucionar, entonces se cancelará el proceso de firma.

CU 6 - Alta de identidades de firma		
Precondición	El usuario debe de haber iniciado sesión en la aplicación.	
Descripción	Se podrán dar de alta las identidades de firma del usuario.	
Secuencia	**Paso**	**Acción**
	1	El usuario selecciona la opción de gestionar las identidades de firma.
	2	El usuario selecciona la operación de alta de identidad de firma.
	3	El usuario carga en la aplicación los ficheros correspondientes a la firma electrónica que desea utilizar.
	4	La aplicación envía al servicio de firma los datos de la firma electrónica y crea la nueva identidad de firma.
	5	Se lee la lista de identidades de firma disponibles y se muestra al usuario.
Postcondición	El servicio de firma dispondrá de una nueva identidad de firma.	
Excepciones	**Paso**	**Acción**

	4	Si el servicio de firma no puede crear la nueva identidad de firma, se muestra un error al usuario y la posible solución, si existe.

CU 7 - Baja de identidades de firma		
Precondición	El usuario debe de haber iniciado sesión en la aplicación.	
Descripción	Se podrán dar de baja las identidades de firma del usuario.	
Secuencia	**Paso**	**Acción**
	1	El usuario selecciona la opción de gestionar las identidades de firma.
	2	El sistema lee la lista de identidades de firma disponibles y se muestra al usuario.
	3	El usuario selecciona la identidad de firma que desea dar de baja.
	4	Se envía al sistema de firma los datos de la identidad de firma que se eliminará.
	5	Se lee la lista de identidades de firma disponibles y se muestra al usuario.
Postcondición	El servicio de firma eliminará una identidad de firma.	
Excepciones	**Paso**	**Acción**
	4	Si el servicio de firma no puede eliminar la identidad de firma, se muestra un error al usuario y la posible solución, si existe.

2.9. Interfaz de usuario

Al ser CloudDocs una aplicación web, el interfaz de usuario se desarrolla usando las tecnologías habituales del este tipo de aplicaciones: **HTML**, **CSS** y **Javascript**. Con el lenguaje HTML se representa el interfaz de usuario y el contenido a visualizar, CSS describe el aspecto visual de los elementos del interfaz, es decir la ubicación de los componentes y el aspecto estético de los mismos, y el lenguaje Javascript ejecuta todas aquellas operaciones que implican una interacción con el usuario, así como el control de todas las operaciones que se pueden realizar en el navegador del cliente (comprobación de que se han seleccionado ficheros antes de firmar o validar,

navegación por las carpetas de los proveedores de almacenamiento, etc.), para dar así una respuesta más rápida al usuario sin necesidad de hacer una llamada al servidor donde se encuentra alojada la aplicación, reduciendo el tráfico de datos necesario.

El interfaz de usuario se diseñará de tal manera que sea lo más simple y ligero posible, de forma que se pueda visualizar en dispositivos de diferentes características (potencia de procesamiento, memoria disponible, resolución de pantalla, etc.). El interfaz debe tener un funcionamiento y apariencia similar en todos los dispositivos, limitado siempre por la resolución de la pantalla del dispositivo.

El sistema tendrá las siguientes pantallas o interfaces:

Acceso al sistema

Se requerirá que el usuario inicie sesión para poder utilizar el sistema. Si el usuario accede al sistema y no ha iniciado sesión, se le redirigirá a la página desde donde podrá autenticarse.

Interfaz principal del CloudDocs

La aplicación contará con una página o pantalla principal desde donde el usuario trabajará con la aplicación. Esta pantalla constará de las siguientes partes:

- Un menú con las opciones:
  - Seleccionar un servicio de almacenamiento.
  - Firmar documentos seleccionados.
  - Validar documentos seleccionados.
  - Gestionar identidades de firma.
  - Cerrar la sesión.
- Un elemento de navegación en el que se listará el contenido del servicio de almacenamiento seleccionado y en el que se podrán seleccionar los documentos a firmar o validar.
- Una lista con los documentos seleccionados por el usuario. Esta lista incluirá los controles necesarios para eliminar uno o varios elementos de la lista por si el usuario ha seleccionado algún documento con el que no desea trabajar.

Selección de servicio de almacenamiento

Muestra al usuario una lista con los servicios de almacenamiento soportados por la aplicación, de los que el usuario debe de escoger uno. En cuanto se escoge un servicio se redirige al usuario a la página de inicio de sesión del servicio seleccionado. Si la validación es correcta se vuelve al interfaz principal de CloudDocs y se muestra el contenido de la carpeta raíz del servicio.

Firma de documentos seleccionados

Esta interfaz se usa para firmar los documentos que se han seleccionado en la pantalla principal de CloudDocs. Este interfaz está separado en varias etapas. En la primera etapa el usuario selecciona la identidad de firma que desea utilizar para firmar los documentos, si no se ha seleccionado una firma con anterioridad; en la segunda autoriza el uso de la firma; y en la tercera se muestra el resultado de la firma de los documentos.

La normativa eIDAS obliga a que el usuario autorice el uso de la identidad de firma. Esta autorización se realiza a través del interfaz de la plataforma TrustedX, por lo que es necesario redirigir el navegador del usuario hacia este interfaz. Una vez autorizado o denegado el uso de la firma, se vuelve al interfaz propio de CloudDocs y se muestra al usuario el resultado de la firma.

Gestión de identidades de firma

Interfaz en el que se muestran y gestionan las identidades de firma existentes. La pantalla consta de una lista en la se muestran las identidades existentes, y de un menú de opciones a través del cual se podrán añadir nuevas identidades de firma o eliminar las identidades de firma seleccionadas.

3. Desarrollo e Implementación

3.1. Objetivos del Sistema

Tal y como se menciona en los capítulos anteriores, el principal objetivo de la aplicación CloudDocs es la firma electrónica de documentos PDF almacenados en servicios ubicados en la nube -como Google Drive, DropBox u otros-, utilizando para ello los servicios de firma electrónica TrustedX, de Entrust Corporation.

El segundo objetivo de la aplicación es que se pueda hacer uso de ella sin necesidad de instalar ningún software extra en los dispositivos del usuario. Este objetivo se ha alcanzado desarrollándola como una aplicación web. La aplicación se ha desarrollado en el lenguaje JAVA versión 11, haciéndose el desarrollo y puesta en producción sobre un servidor Apache Tomcat 10.0.5. Esto permite que se pueda hacer uso de CloudDocs solamente disponiendo de acceso a internet y utilizando un navegador web, condiciones que cumplen la mayoría de los dispositivos informáticos a día de hoy, por ejemplo, ordenadores, teléfonos móviles, tablets, etc.

Tal y como se ha propuesto, la comunicación entre la aplicación y los servicios se realiza utilizando canales seguros, así como el acceso del usuario a la aplicación, utilizando el protocolo HTTPS. Para el acceso a la aplicación se ha utilizado un certificado autofirmado, lo cual, aunque es aceptable para un entorno de desarrollo y pruebas, no lo es para prestar un servicio al público en general. En este caso sería necesario disponer de un certificado generado por una autoridad certificadora reconocida. Solamente en el caso de que el ámbito de uso de la aplicación esté limitado a un entorno controlado -como por ejemplo para el uso privado de una empresa, utilizando servidores propios- sería aceptable el uso de certificados generados con una autoridad certificadora propia.

Consideraciones

Debido a que el aplicativo es una prueba de concepto se ha utilizado una versión de prueba y desarrollo del servicio de gestión de identidades y firmas, TrustedX. Debido a esto, algunos de los requisitos del reglamento eIDAS no se podrán cumplir, por lo que la solución no se podrá considerar un servicio cualificado de firma electrónica (QSCD). Entre los requisitos no cumplidos encontramos:

- La firma electrónica cualificada, tal y como se ha visto en el apartado 2.2, debe de ser generada por un dispositivo cualificado. Del anexo II

del reglamento se entiende que los certificados utilizados para la firma cualificada deben ser generados por un prestador cualificado de servicio,
- El sistema de autenticación debe cumplir con los requisitos del nivel de seguridad exigido por el Reglamento de Ejecución 2015/1502 de la Comisión de 8 de septiembre de 2015.

En el demostrador se utilizan certificados almacenados en ficheros PKCS#12, generados externamente al servicio de firma, para crear las identidades de firma. Debido a esto, la solución **no se puede considerar como un servicio cualificado de creación de firmas**. Además, se utiliza un sistema de autenticación basado en usuario y contraseña, lo cual no cumple con el nivel de seguridad sustancial o alto, especificado en el apartado 2.2.1 del anexo del Reglamento de Ejecución 2015/1502, que especifica que se deben utilizar, por lo menos, dos factores de autenticación diferentes y que el sistema de identificación esté diseñado de forma que solo lo pueda utilizar la persona a la que pertenece la identidad.

Aún pese a estos incumplimientos, se ha podido cumplir con el objetivo de que la generación del documento firmado se realice en el propio servidor de CloudDocs, con lo que solamente se envía al servidor de firmas el hash resumen del documento, y no el documento completo.

3.2. Implementación del Sistema

El diseño original de CloudDocs contemplaba la división de la aplicación en cuatro componentes diferenciados por la funcionalidad y los servicios que proporcionaban. Los componentes propuestos son:

- **Componente CloudDocs**: Componente principal de la aplicación, encargado de gestionar el flujo del funcionamiento de la aplicación, comunicando con los otros módulos en el orden adecuado para cumplir con la funcionalidad requerida, la firma y validación de documentos almacenados en la nube.
- **Componente del proveedor de identidad (IpD):** Componente que gestiona la comunicación con el IdP seleccionado, para así realizar la autenticación de los usuarios y permitir el acceso a los recursos y servicios necesarios para la firma y validación de documentos.
- **Componente del proveedor de firma (TrustedX):** Componente que proporciona acceso a los servicios prestados de firma y a los recursos necesarios para realizar la firma y validación de los documentos. El componente permitirá realizar la firma y validación de

documentos, y la gestión de las identidades de firma. El servicio de firma electrónica utilizado en este trabajo es TrustedX.

- **Componente del proveedor de almacenamiento:** Este componente gestiona el acceso a los documentos almacenados en el proveedor escogido de almacenamiento externo.

Si bien la implementación de la aplicación ha respetado esta división y los componentes básicos, al desarrollar CloudDocs como una aplicación web y seguir el patrón MVC ha sido necesario el desarrollo de módulos y librerías de apoyo para implementar la modularidad y la comunicación entre los módulos.

Los módulos más importantes en los que se ha dividido la aplicación son:

- **Modulo Web:** Este módulo se encarga de gestionar la interacción entre el usuario y la aplicación, así como también responde a las peticiones realizadas hacia la aplicación desde otros servicios externos, como por ejemplo en las llamadas necesarias para gestionar el protocolo OAuth2.

  Este módulo está compuesto de la clase java *com.CloudDocs.web.ModuloWeb*, que es la que atiende todas las peticiones realizadas y devuelve el contenido al navegador o a los servicios externos. También forman parte de este módulo los ficheros jsp que implementan las vistas, así como los ficheros javascript y las hojas de estilo css.

  Este módulo corresponde con las vistas del patrón MVC.

- **Módulo de autenticación:** Encargado de gestionar que el usuario pueda acceder o no a los servicios de la aplicación. Se apoya en el servicio IdP de TrustedX, por lo que solo podrán acceder a la aplicación aquellos usuarios que dispongan de cuenta en el servido de firma de TrustedX.

  Está compuesto por la clase java *com.CloudDocs.autenticacion.ModuloAutenticacion*.

- **Módulo de acceso a almacenamiento:** Gestiona el acceso a los proveedores de almacenamiento. Está desarrollado en dos capas, de forma que la primera capa es a la que acceden los demás módulos de la aplicación para acceder a los documentos, y la segunda es la que acceder a los servicios de cada proveedor de almacenamiento

específico. Para esta segunda capa se desarrollan tantas clases como proveedores de almacenamiento se quieran utilizar.

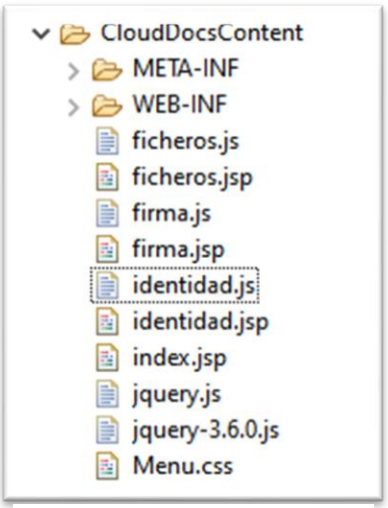
Compuesto por la clase principal *com.CloudDocs.ficheros.ModulosFichero* para la primera capa, las clases *ProveedorDropBox* y *ProveedorGDrive*, y otras clases auxiliares.

- **Módulo de firma:** Se utiliza para la firma electrónica de documentos PDF y la validación de los mismos. Gestiona las llamadas al proveedor de firmas. Está formado por la clase *com.CloudDocs.firma.ModuloFirma*.

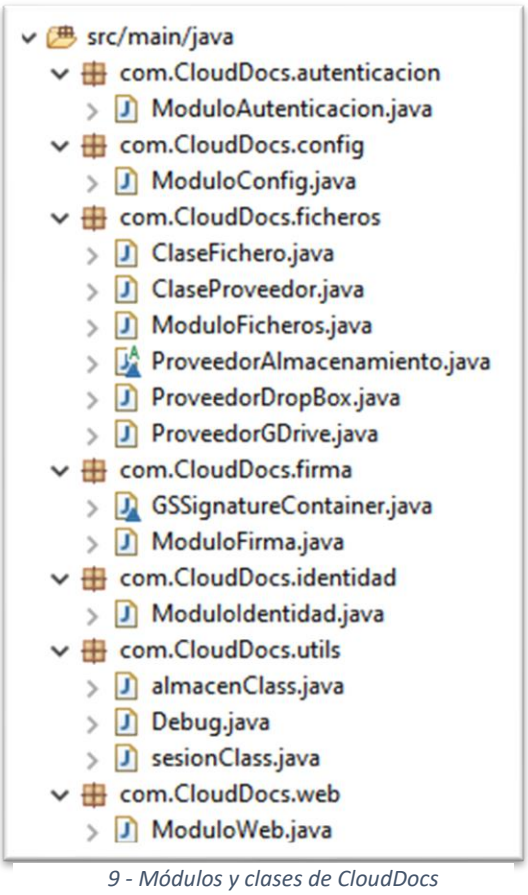
- **Módulo de identidad:** Se utiliza para la comunicación con el proveedor de identidad, permitiendo el acceso a las identidades de firma y gestión de las mismas.
  Está compuesto por la clase *com.CloudDocs.identidad.ModuloIdentidad*.

- **Módulos auxiliares:** Tal y como se ha mencionado, ha sido necesario desarrollar algunos módulos auxiliares para la comunicación entre los módulos principales. Estos módulos secundarios son:
  - **Módulo de configuración:** Permite el acceso a los valores almacenados en los ficheros de configuración.
  - **Módulo de utilidades:** Módulo con utilidades varias. Permite mantener información común a los diferentes módulos, acceder a los datos de sesión, así como también tiene utilidades de depuración.

En la implementación de los módulos se ha intentado respetar el patrón MVC al máximo posible, sin embargo, se han tomado algunas libertades en algunos apartados puntuales, permitiendo que algunos módulos devuelvan directamente la información al interfaz de usuario en lugar de devolvérsela al módulo web.



8 - Ficheros de FrontEnd



9 - Módulos y clases de CloudDocs

3.3. Implementación de flujos de información

CloudDocs es una aplicación monolítica, de forma que todos los módulos forman una unidad. De esta forma la mayor parte del flujo de información entre los módulos se realiza a través de los propios métodos y funciones de las clases, y de los valores devueltos por estos. Sin embargo, existen casos en que es necesario almacenar información para un uso posterior. En estos casos se utiliza un almacén de información común a todos los módulos. Por ejemplo, el módulo de identidad y el módulo de firma no tienen una comunicación directa, sin embargo, es necesario utilizar una identidad de firma para firmar los documentos. Para que el módulo de firma pueda firmar los documentos tiene que tener acceso a la identidad de firma seleccionada por el usuario. Para ello el módulo de identidad guarda el identificador de la entidad de firma seleccionada en el almacén, de donde lo obtendrá el módulo de firma para realizar la firma de los documentos.

Todos los servicios externos utilizados tienen una API RESTfull, por lo que la comunicación entre la aplicación y los servicios se realizarán a través del protocolo HTTPS, utilizando para ello la librería Unirest en su versión 3.11. Esta librería permite realizar llamadas a APIs de una forma sencilla. Por ejemplo, el siguiente código realiza la llamada a la API de DropBox para obtener el token OAuth2 de autorización para poder trabajar con los ficheros de DropBox.

```
HttpResponse<String> respuesta = Unirest.post(ModuloConfig.properties.getProperty("DropBoxURLToken"))
          .header("Content-Type", "application/x-www-form-urlencoded")
          .field("code", code)
          .field("grant_type", "authorization_code")
          .field("redirect_uri", "https://clouddocs.org:8443/CloudDocs/ficheros/callback/")
          .field("client_id", clientSecrets.get("key").toString())
          .field("client_secret", clientSecrets.get("secret").toString())
          .asString();
```

10 - Ejemplo de código para llamar a APIs RESTfull

En el ejemplo podemos observar que en una sola llamada se definen los datos que se enviarán en la llamada, así como las cabeceras HTTP necesarias. En este ejemplo en particular, la respuesta del servicio utilizado será una cadena contenida en el cuerpo de la respuesta. La librería utilizada permite acceder de forma sencilla tanto al cuerpo de la respuesta como a los códigos de estado devueltos a través del protocolo HTTP. De esta forma la aplicación puede saber si ha obtenido una respuesta correcta (con un código HTTP 200) o no (cualquier otro código de error, por ejemplo, un código HTTP 403 si el usuario no tiene permisos para acceder al servicio solicitado).

3.4. Integración de servicios de diferentes proveedores de almacenamiento

CloudDocs permite la firma de documentos almacenados en diferentes proveedores de almacenamiento. Si bien solo se puede utilizar un proveedor de almacenamiento a la vez[28], la aplicación utiliza el mismo módulo de acceso a almacenamiento en todos los casos, aun cuando cada proveedor de

[28] Esto quiere decir que a lo largo de una sesión el usuario solo podrá trabajar con un solo proveedor de almacenamiento. Si el usuario selecciona acceder a Google Drive, solo podrá trabajar con los documentos almacenados en Drive. Si quisiera acceder a documentos almacenados en DropBox, será necesario que cierre la sesión en CloudDocs y vuelva a iniciarla para así poder escoger nuevamente que proveedor de almacenamiento utilizar.

almacenamiento tiene su propia API de autenticación y de acceso a los documentos almacenados.

Para que el interfaz web y los demás módulos de la aplicación puedan acceder a los documentos almacenados es necesario ocultar estas diferencias entre los proveedores, presentando un interfaz homogéneo. Esto se consigue desarrollando el módulo de almacenamiento con una arquitectura en dos capas, tal y como se ha comentado en el apartado 3.2.

La primera capa del módulo es a la que acceden los demás módulos de la aplicación para hacer uso de los servicios de almacenamiento. Esta capa se encarga tanto de gestionar la carga, descarga y llamada a las funciones de los módulos de los servicios de almacenamiento seleccionados por el usuario. La segunda capa se encarga de realizar las llamadas a las APIs de los proveedores de almacenamiento y de convertir las respuestas de estos a un formato común.

Esta estructura en dos capas tiene dos grandes ventajas. La primera es que se pueden añadir más proveedores de almacenamiento realizando una mínima modificación en la aplicación. Solamente sería necesario desarrollar el módulo específico para el nuevo proveedor y añadir este módulo a la lista de proveedores. La segunda ventaja es que permite que el almacenamiento funcione como una especie de caja negra, ocultando los detalles de las implementaciones particulares. No es necesario adaptar el resto de la aplicación a las particularidades de cada proveedor de almacenamiento.

Esta estructura, sin embargo, tiene el inconveniente de que no permite aprovechar las características particulares que puedan tener algunos proveedores, características que podrían permitir un funcionamiento más rápido o sencillo. Por ejemplo, se podría desarrollar un módulo que permita el acceso a documentos almacenados en el propio servidor de la aplicación o en un almacenamiento NAS (en el caso de que la aplicación se utilizara en el entorno privado de una empresa), en este caso en particular el acceso a los documentos no sería tan eficaz comparado con el caso en el que la aplicación accediera directamente a los documentos utilizando las funciones proporcionadas por el lenguaje de programación.

3.5. Implementación del proceso de firma

La firma electrónica de documentos PDF es el objetivo principal del presente trabajo. El proceso de firma se ha descrito a grandes rasgos y de forma teórica en el apartado 2.5 y en el caso de uso 5, en el apartado 2.6. Sin embargo, la implementación en la práctica siempre conlleva algunas

diferencias con lo propuesto en el diseño del proceso, bien sea por las características del lenguaje de programación utilizado, por los requerimientos del entorno o por las funcionalidades y/o deficiencias que tengan las librerías y servicios utilizados. En este apartado se describirá la implementación real del proceso de firma.

El proceso de firma comienza cuando el usuario selecciona una identidad de firma y uno o más documentos que desea firmar. Una vez hecho esto comenzará el proceso de firma pulsando sobre el botón de firmar documentos, y se le mostrará al usuario una pantalla en la que se listarán los documentos a firmar y un botón para iniciar el proceso de firma. Al pulsar el botón, el sistema redirige el navegador del usuario a la plataforma TrustedX para que el usuario autorice el uso de la identidad de firma. Esta autorización se traduce en un token OAuth2 que la aplicación utilizará en cada vez que tenga que firmar un documento con la identidad de firma seleccionada. Una vez obtenido el token, la aplicación comenzará el proceso de firma de los documentos.

En este punto, la aplicación realizará el mismo proceso para cada documento seleccionado. A través del módulo de almacenamiento, la aplicación descargará el documento seleccionado a una carpeta temporal y comprobará que el documento es de tipo PDF. Para ello se intenta abrir el documento y leer la primera página del mismo, utilizando para ello la librería *iText 7*. Esta es una librería que permite trabajar y crear documentos PDF, permitiendo también la firma y validación de los mismos.

Una vez que se ha comprobado que el documento es de tipo PDF se procede a la firma del mismo. Para ello es necesario obtener el certificado almacenado en la identidad de firma, ya que este es necesario en el proceso de generación del hash. El certificado se encuentra junto con los datos de la identidad de firma, por lo que se solicitan los datos al servicio de TrustedX.

Cuando se disponen de todos los datos se inicia el proceso de firmado. Para ello se crea, utilizando la librería iText, un nuevo documento PDF, copia del documento descargado. En este nuevo documento se crea la firma, añadiendo los datos visibles de la misma en la esquina superior izquierda de la primera página del documento.

Estando ya definidos los datos y el nivel de firma, se crea el contenedor de la misma, se extrae el certificado de los datos de la identidad de firma leídos anteriormente, y junto con el certificado raíz se almacenan en una cadena de certificados.

Debido a que se utiliza un servicio externo para generar la firma, es necesario indicar a la librería como hacer uso de este servicio. Para ello se crea una clase que implementa el interfaz IExternalSignature. Los objetos de

esta clase implementan los métodos que realizan la firma llamando a servicios externos, como podría ser un dispositivo hardware de firma, o como en el caso de CloudDocs, un servicio remoto de firma. Una vez inicializado el objeto, se llama al método de firma de la librería iText, pasándole toda la información recopilada y los objeto creados.

La librería iText se encarga de crear el nuevo documento PDF, añadiéndole la firma en el sitio adecuado. Para ello utilizará el objeto de tipo IExternalSignature que tiene un método llamado sign. Este método genera el hash de los datos que se han de firmar y llama al servicio de firma de TrustedX enviándole el hash a firmar. TrustedX generará la firma PKCS#1 del hash. Esta firma PKCS#1 se devuelve a iText, que finaliza el proceso de firma del documento PDF.

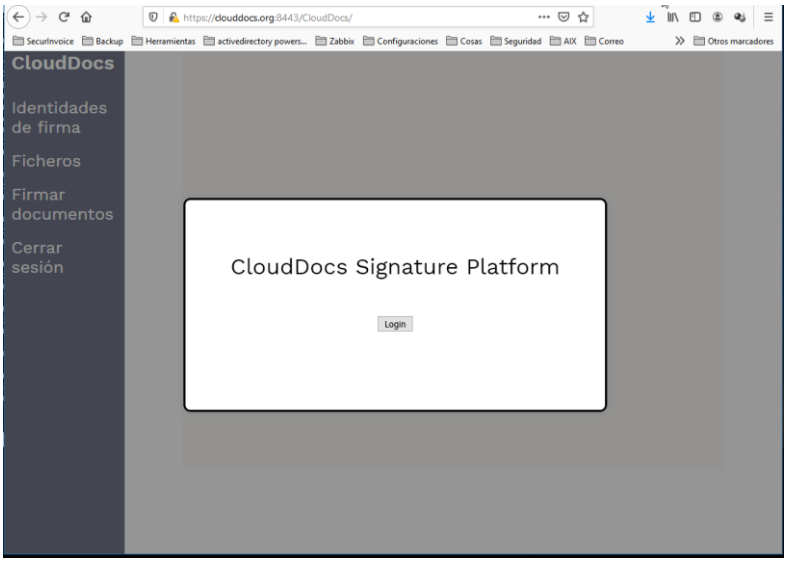
Si todo el proceso de firma ha sido correcto y no ha ocurrido ninguna excepción, se habrá generado un nuevo documento PDF con la firma electrónica. Este documento se guardará en el mismo proveedor de almacenamiento y en la misma ubicación que el fichero original, aunque se le añadirá el sufijo “-signed” al nombre del documento para así distinguirlo del original.

3.6. Interfaz de usuario

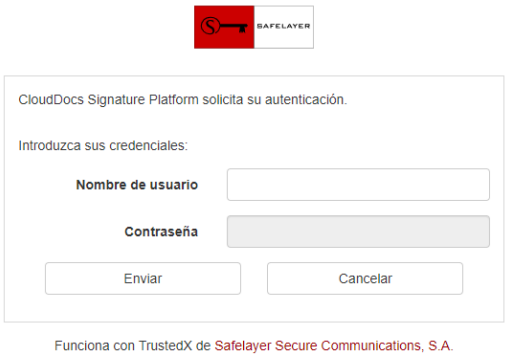
En el siguiente apartado se presenta, en imágenes, el flujo de uso de la aplicación, de forma que se pueden observar al completo el interfaz de usuario.

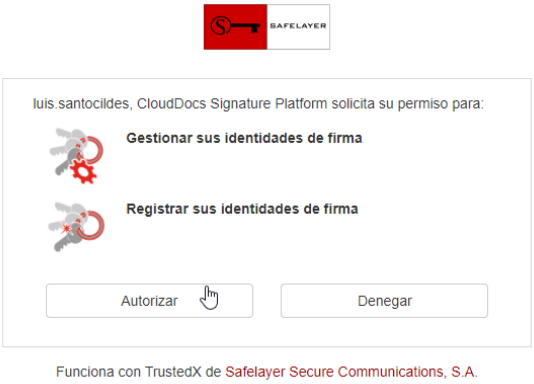
Acceso al sistema y autenticación del usuario:

Pantalla inicial y Acceso al sistema



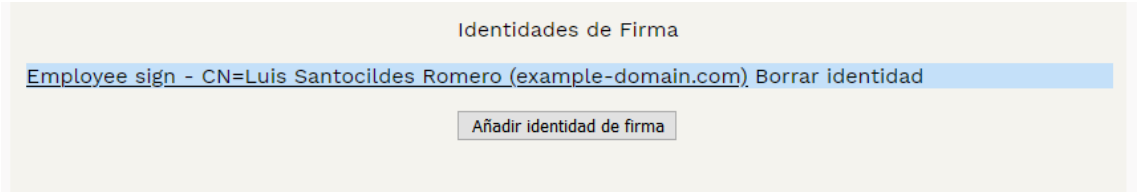
Autenticación y autorización utilizando el usuario de TrustedX

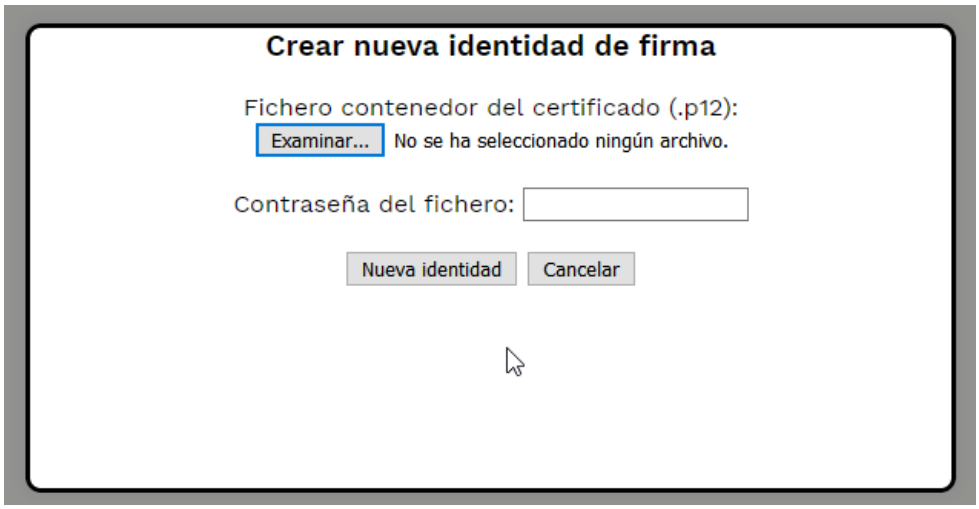


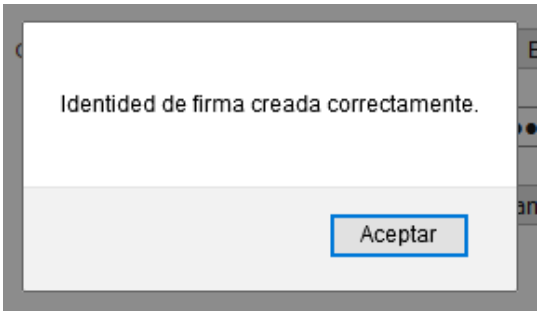


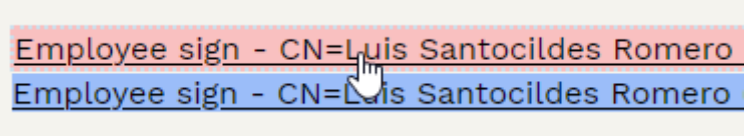
Gestión de identidades de firma

Listado de identidades de firma, creación de nuevas identidades y selección de la identidad a utilizar



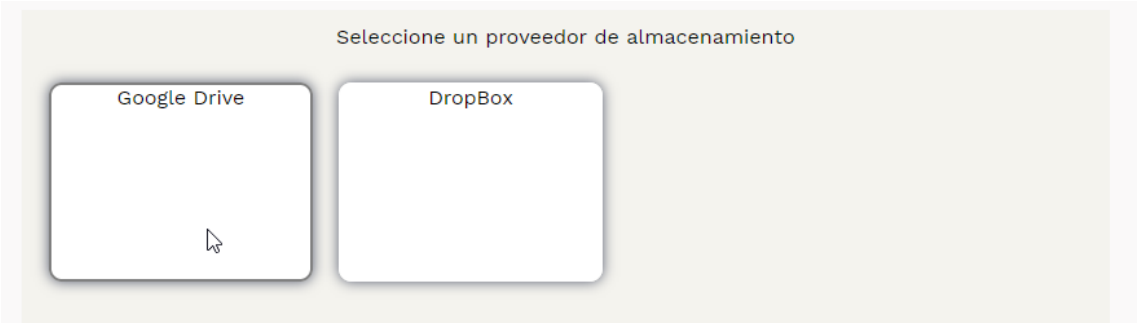






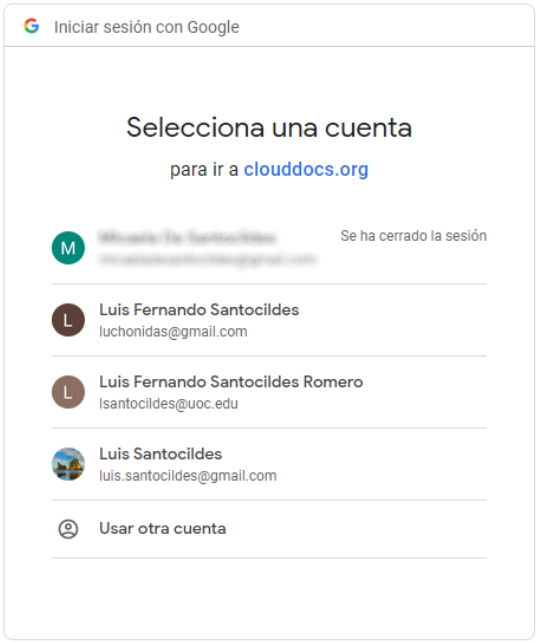
Acceso a proveedores de almacenamiento

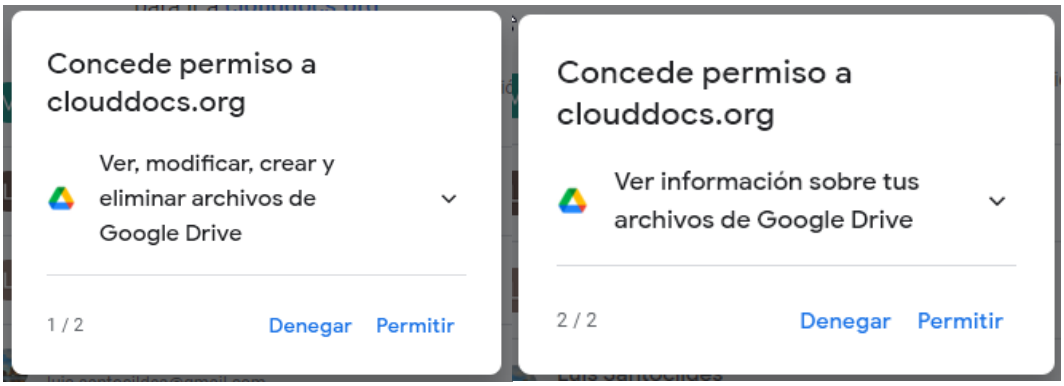
Navegación en los servicios de almacenamiento y selección de documentos a firmar.

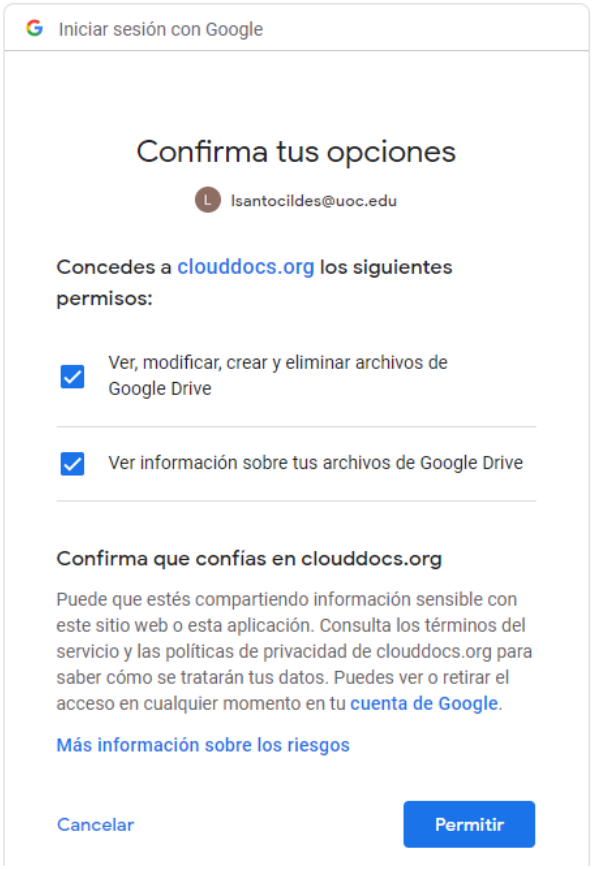


Autenticación y autorización del proveedor de almacenamiento

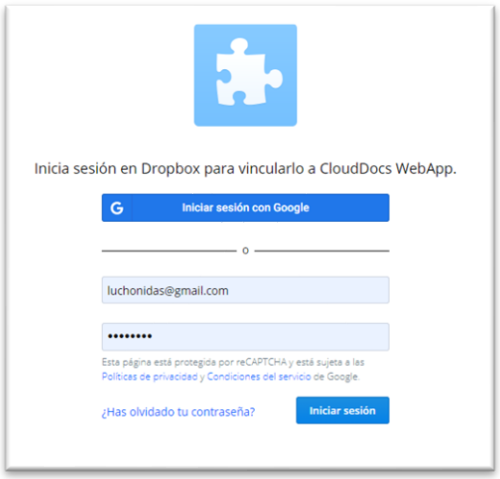
Google Drive

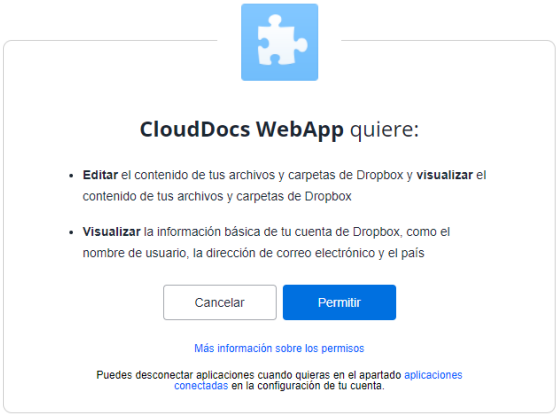



Concede permiso a clouddocs.org
Ver, modificar, crear y eliminar archivos de Google Drive
1 / 2
Denegar
Permitir
Concede permiso a clouddocs.org
Ver información sobre tus archivos de Google Drive
2 / 2
Denegar
Permitir

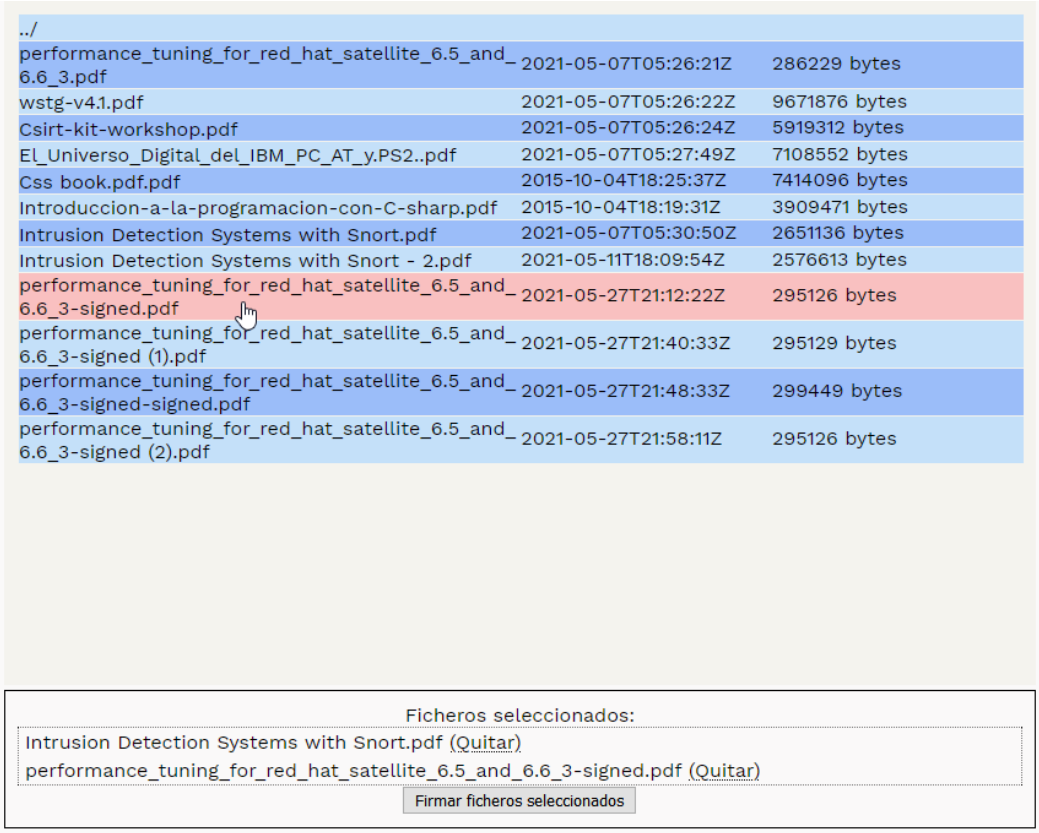

Iniciar sesión con Google
Confirma tus opciones
lsantocildes@uoc.edu
Concedes a clouddocs.org los siguientes permisos:
Ver, modificar, crear y eliminar archivos de Google Drive
Ver información sobre tus archivos de Google Drive
Confirma que confías en clouddocs.org
Puede que estés compartiendo información sensible con este sitio web o esta aplicación. Consulta los términos del servicio y las políticas de privacidad de clouddocs.org para saber cómo se tratarán tus datos. Puedes ver o retirar el acceso en cualquier momento en tu cuenta de Google.
Más información sobre los riesgos
Cancelar
Permitir

DropBox





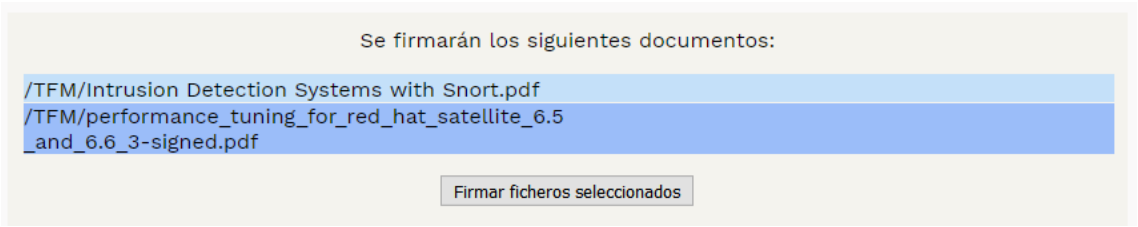
Navegación por los contenidos y selección de documentos

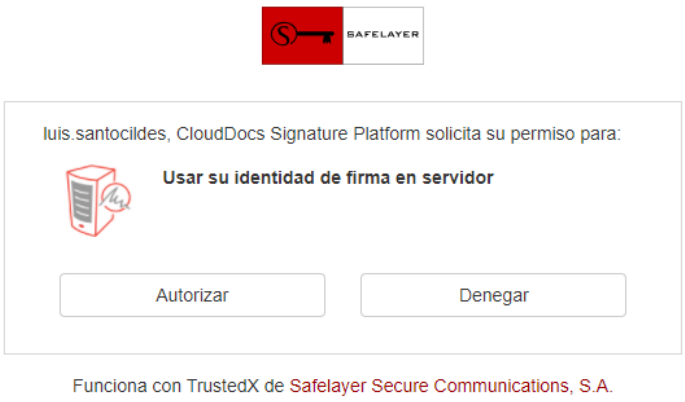


Una vez seleccionados los documentos, se pueden firmar pulsando en el botón correspondiente.

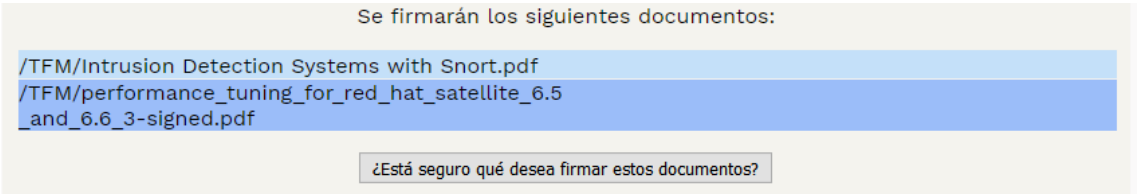
Proceso de firma de documentos

Proceso de firma completo con primera autorización para el uso de la firma electrónica seleccionada

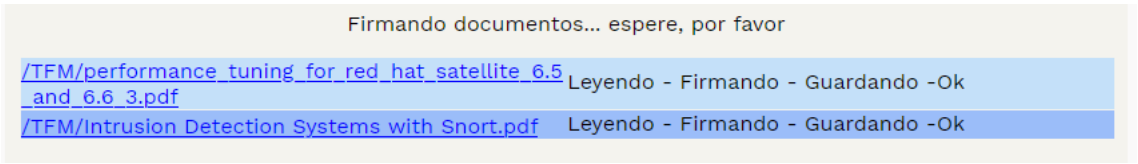




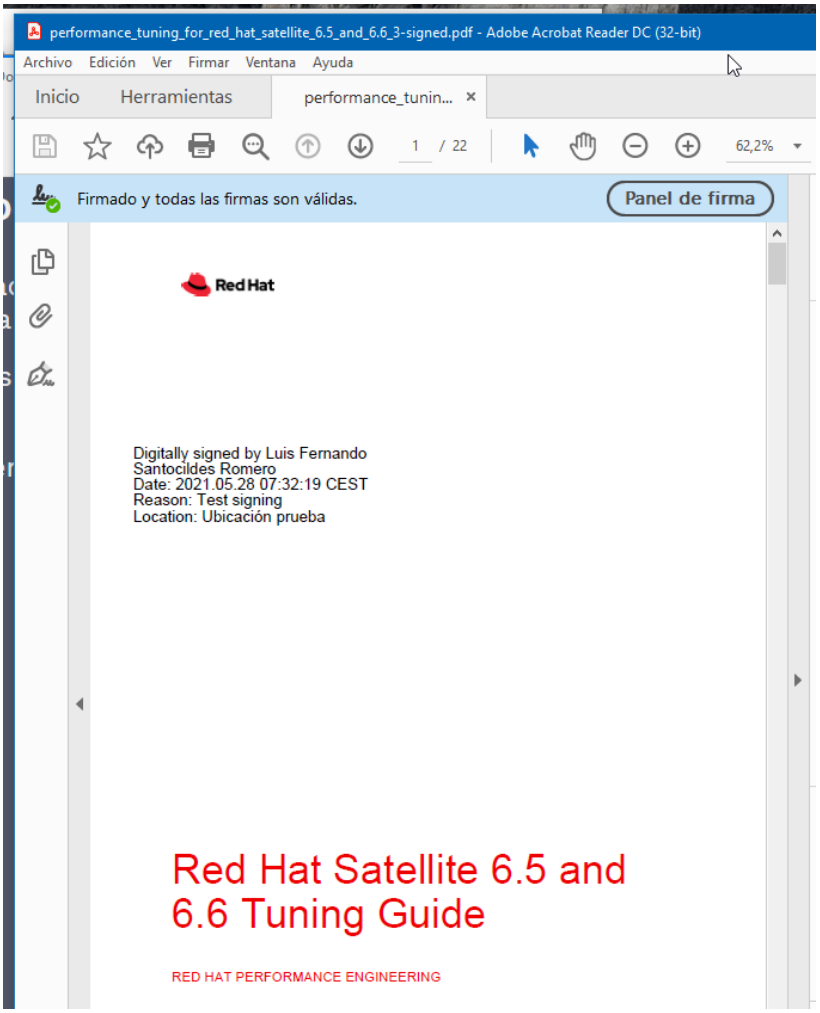
Una vez autorizado el uso de la firma se le solicita confirmación al usuario de la operación que se va a realizar.

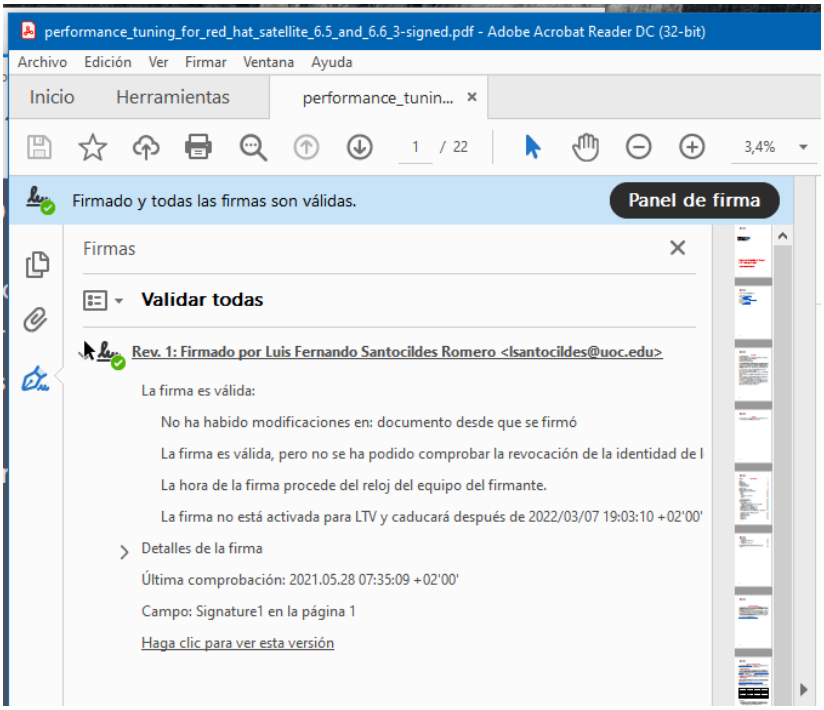


Una vez firmados los documentos, se le avisa al usuario. Los documentos firmados se guardan en las mismas carpetas que los originales.
Se proporciona al usuario un enlace para descargar el fichero firmado



Visualización de un fichero firmado y validación del mismo a través de Acrobat Reader.





3.7. Puesta en producción

Para la puesta en producción de la aplicación es necesario empaquetar la aplicación en un fichero de tipo *war*, Este fichero se generará desde el propio entorno de desarrollo Eclipse, usando la opción de exportar del entorno. Cuando se haya generado el fichero *war* que contiene la aplicación, se desplegará en un servidor **Apache Tomcat**. El despliegue se realizará copiando el fichero a la carpeta *webapps* que se encuentra dentro de la instalación de Apache Tomcat.

Realizado el despliegue, se podrá acceder a la aplicación a través de un navegador web, utilizando para ello la dirección URL que apunta al servidor de aplicaciones y a CloudDocs en particular.

4. Conclusiones

Durante el desarrollo de este trabajo se ha observado la necesidad de una regulación de la firma electrónica que permita un uso más flexible de la firma electrónica, siempre sin perder de vista la seguridad, la protección de los datos y una facilidad de uso. La creación del reglamento eIDAS, y su trasposición a las diferentes legislaciones nacionales de los miembros de la Unión Europea, permite un mejor uso de la firma electrónica, acercando esta a la realidad tecnológica existente.

El estudio previo al desarrollo del trabajo muestra que existen bastantes soluciones de firma electrónica que implican el uso de dispositivos de firma y aplicaciones específicas, lo que conlleva una dificultad de instalación y uso de la firma electrónica, encontrándose casos en los que, cuando es necesario presentar documentos firmados electrónicamente a diferentes organismos, es necesario el uso de diferentes aplicaciones para firmar ya que cada organismo tiene su aplicación específica. Además, se ha visto que, en ocasiones, los dispositivos de firma electrónica no estan soportados en todos los sistemas que pueden utilizar los usuarios.

La solución estudiada y desarrollada en el presente trabajo ha hecho uso de las tecnologías de desarrollo actuales y de las facilidades introducidas por el reglamento eIDAS para evitar los problemas estudiados. Para ello se ha implementado una solución web que utiliza los servicios de firma electrónica ofrecidos por TrustedX para así firmar los documentos almacenados en la nube. La propia estructura de la aplicación mantiene también la privacidad de la información de los usuarios, ya que los documentos a firmar nunca se envían al servicio de firma, y solo se almacenan en la propia aplicación el tiempo necesario para generar la firma.

Durante el desarrollo del producto no se ha podido cumplir con uno de los objetivos planteados inicialmente, en particular la validación de firmas electrónicas desde dentro de la propia aplicación. Esta línea de trabajo queda pendiente de explorar y desarrollar en futuros desarrollos, así como un mejor desarrollo de la modularidad de la aplicación. Quedan pendientes cambios en la autenticación del usuario y en la creación de las identidades de firma para poder cumplir al completo con el reglamento.

Queda la satisfacción de haber conseguido el desarrollo e implantación de un producto que demuestra la utilidad del reglamento eIDAS, facilitando la creación y uso de la firma digital.

5. Glosario

eIDAS: Reglamento Europeo que regula la identificación electrónica y los servicios de confianza para las transacciones electrónicas, dentro del ámbito del mercado interior de la Unión Europea.

Firma electrónica: es un concepto jurídico, equivalente electrónico al de la firma manuscrita, donde una persona acepta y da por validado el contenido de un mensaje electrónico a través de cualquier medio electrónico que sea legítimo y permitido.

Firma electrónica avanzada: firma electrónica en la que se identifica de forma única al firmante del mensaje o documento.

Firma electrónica cualificada: firma similar a la firma electrónica avanzada, creada a través de un dispositivo cualificado de firma electrónica. Debe cumplir con los requisitos del reglamento eIDAS.

Prestador de servicios de confianza: prestador de servicios de creación y/o validación de identidad electrónica, firma electrónica, sello electrónico o sello electrónico de tiempo.

Prestador de servicios de confianza: prestador de servicios de confianza que cumple con los requisitos del reglamento eIDAS.

Nube/Cloud: Conjunto de servidores remotos, interconectados y configurados para funcionar como un único ecosistema. Pueden almacenar datos, ejecutar aplicaciones, servir contenidos, etc. Los usuarios de la nube no acceden a los servidores individuales, sino a los servicios que prestan estos servidores.

PDF: Portable Document Format. Estándar bastante utilizado para representar información contenida en documentos legibles.

PKCS: Public-Key Cryptography Standars. Conjunto de estándares de criptografía de clave pública.

HTML: HyperText Markup Language. Lenguaje que se utiliza para especificar la estructura y el contenido de las páginas web.

CSS: Cascading Style Sheets u Hojas de Estilo en Cascada. Lenguaje de diseño que permite definir como se visualizará el contenido de las páginas web.

Javascript: lenguaje de programación, mayormente utilizado en las páginas web para dotar de dinamismo a las mismas. Se interpreta en el navegador del usuario.

IdP: Proveedor de Identidad. Servicio o entidad que almacenan los datos de identidad de los usuarios de uno o más sistemas. Son proveedores de identidad, por ejemplo, Google, Microsoft, Yahoo, etc.

API: Interfaz de programación de aplicaciones. Son el conjunto de subrutinas, funciones y procedimientos que provee los servicios y librerías para poder ser utilizados desde otras aplicaciones.

iText: Librería para crear y manipular documentos PDF desde otras aplicaciones.

JSP: Java Server Pages. Tecnología para crear fácilmente páginas web dinámicas. Basada en el lenguaje de programación Java.

6. Bibliografía

- [1] INE (INSTITUTO NACIONAL DE ESTADÍSTICA). *Encuesta sobre el uso de TIC y del comercio electrónico en las empresas Año 2019 – Primer trimestre de 2020.* Nota de prensa del 20/10/2020. Formato PDF. Publicada en https://www.ine.es/prensa/tic_e_2019_2020.pdf.
- [3][12] UNION EUROPEA. *REGLAMENTO (UE) Nº 910/2014 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 23 de julio de 2014 relativo a la identificación electrónica y los servicios de confianza para las transacciones electrónicas en el mercado interior y por la que se deroga la Directiva 1999/93/CE.* Diario Oficial de la Unión Europea del 28/8/2014. Formato PDF. Publicado en https://www.boe.es/doue/2014/257/L00073-00114.pdf.
- [4] COMISIÓN EUROPEA. *Tipos de Derecho de la UE*. <https://ec.europa.eu/info/law/law-making-process/types-eu-law_es#tipos-de-actos-jurdicos-de-la-ue> [Enlace revisado el 30/5/2021]
- [6][7] INE. *Encuesta sobre Equipamiento y Uso de Tecnologías de la Información y Comunicación en los hogares 2010.* <https://www.ine.es/jaxi/Datos.htm?path=/t25/p450/a2010/l0/&file=08028.px> y <https://www.ine.es/jaxi/Tabla.htm?path=/t25/p450/a2010/l0/&file=08028.px&L=0> [Revisado el 25/2/2021].
- [10] España. *Ley 6/2020, de 11 de noviembre, reguladora de determinados aspectos de los servicios electrónicos de confianza.* «BOE» núm. 298, de 12/11/2020. Formato PDF. Publicado en https://www.boe.es/buscar/pdf/2020/BOE-A-2020-14046-consolidado.pdf.
- [11] Pradel Miquel, J., Raya Martos, J. *Ingeniería del software*. Universitat Oberta de Catalunya. 3ª edición. 2016. Versión en PDF. Disponible en https://discovery.uoc.edu/iii/encore/record/C__Rb1044938
- [14][15]Wikipedia. *Hypertext Transfer Protocol*. <https://en.wikipedia.org/wiki/HTTP > [Enlace revisado el 15/3/2021].
- [14][15]Wikipedia. *Hypertext Transfer Protocol*. <https://en.wikipedia.org/wiki/HTTP > [Enlace revisado el 15/3/2021].
- [16]Red Hat. *¿Qué es una API de REST?*. <https://www.redhat.com/es/topics/api/what-is-a-rest-api> [Enlace revisado el 25/3/2021].
- [17] OAuth 2.0. <https://oauth.net/2/> [Enlace revisado el 25/3/2021].
- [19][21][22]Wikipedia. *OpenID*. < https://en.wikipedia.org/wiki/OpenID > [Enlace revisado el 25/3/2021].

- [24] Wikipedia. *PKCS.* <https://en.wikipedia.org/wiki/PKCS> [Enlace revisado el 30/5/2021].
- [25] IETF. *PKCS #1: RSA Cryptography Specifications Version 2.2.* Moriarty, K., Ed., EMC Corporation, Kaliski, B., Verisign, Jonsson, J., Subset AB, Rusch, A., RSA. Publicado en https://datatracker.ietf.org/doc/html/rfc8017
- [26] IETF. *PKCS #12: Personal Information Exchange Syntax v1.1.* Moriarty, K., Ed., EMC, Nystrom, M., Microsoft Corporation, Parkinson, S., Rusch, A., Scott, M., RSA. Publicado en https://datatracker.ietf.org/doc/html/rfc7292
- Wikipedia. *Modelo–vista–controlador*. <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador> [Enlace revisado el 21/3/2021].
- DropBox, *Dropbox for HTTP Developers*. <https://www.dropbox.com/developers/documentation/http/overview>
- Itext, *How to use a Digital Signing Service (DSS) such as GlobalSign, with iText 7*. <https://kb.itextpdf.com/home/it7kb/examples/how-to-use-a-digital-signing-service-dss-such-as-globalsign-with-itext-7>
- UOC, *TrustedX Integration – CloudDocs Signature Platform*. Documento en formato PDF.
- MICROSOFT*. ¿Qué es la nube?*. < https://azure.microsoft.com/es-es/overview/what-is-the-cloud/ > [Enlace revisado el 1/6/2021]
- D. Hardt, Ed. *RFC 6749, The OAuth 2.0 Authorization Framework.* Internet Engineering Task Force (IETF). Versión en PDF. Disponible en https://tools.ietf.org/pdf/rfc6749.pdf.
- Junta de Andalucia. *Guía para la redacción de casos de uso*. <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416 > [Enlace revisado el 16/3/2021].