Universitat Oberta de Catalunya

uoc.edu

Plataforma para la gestión de facturas electrónicas

Factura-Web

Nombre Estudiante: Juan José Palacios de la Flor

Plan de Estudios del Estudiante: Máster Universitario en Seguridad de las Tecnologías de la Información y las Comunicaciones (MISTIC)

Área del trabajo final: Protocolos y aplicaciones de seguridad

Nombre Consultor/a: Juan Carlos Fernández Jara

Nombre Profesor/a responsable de la asignatura: Víctor García Font

Centro: Universitat Oberta de Catalunya

Fecha Entrega: Junio 2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	Plataforma para la gestión de facturas electrónicas- Factura-web
Nombre del autor:	Juan José Palacios de la flor
Nombre del consultor/a:	Juan Carlos Fernández Jara
Nombre del PRA:	Víctor García Font
Fecha de entrega (mm/aaaa):	06/2019
Titulación::	Máster en Seguridad de las tecnologías de la información y las comunicaciones (MISTIC)
Área del Trabajo Final:	Protocolos y aplicaciones de Seguridad
Idioma del trabajo:	Castellano
Palabras clave	Factura, FacturaE, Firma electrónica.

Resumen del Trabajo (máximo 250 palabras): Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.

Las aplicaciones web cada vez son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones sin distribuir e instalarlas en los ordenadores de sus usuarios. Este hecho unido con la entrada en vigor de la regulación de la firma digital reconocida en los estados de la unión europea, ha abierto las puertas a los nuevos servicios de firma digital que aprovechan las utilidades de las aplicaciones web.

El objetivo de este TFM es el estudio del reglamento de la generación de facturas electrónicas, e integrar los servicios de firma digital y verificación de firmas de TrustedX, en una aplicación web funcional. Para garantizar la confidencialidad, la integridad, la disponibilidad y el no repudio que son algunos aspectos principales de seguridad es necesario recurrir a soluciones de certificación digital que excedan en la confianza de las transacciones de cara al remitente (consumidor) como al destinatario (empresa).

Abstract (in English, 250 words or less):

Web applications are becoming popular due to the practicality of the web browser as a thin client, the independence of the operating system, as well as the ease to update and maintain applications without distributing and installing them on the computers of its users. This fact, together with the entry into force of the regulation of the digital signature recognized in the European Union states, has opened the doors to new digital signature services that take advantage of the utilities of web applications.

The objective of this TFM is the study of the regulation of the generation of electronic invoices, and to integrate the services of digital signature and verification of TrustedX signatures, in a functional web application. To guarantee confidentiality, integrity, availability and non-repudiation, which are some of the main aspects of security; it is necessary to resort to digital certification solutions that exceed the confidence of the transactions facing the sender (consumer) as well as the recipient (company).

Índice

1. INTRODUCCIÓN

1.1 CONTEXTO Y JUSTIFICACION DEL TRABAJO

Implementación de una plataforma web que permita a los usuarios de esta herramienta la generación, recepción y gestión completa de facturas en formato electrónico. Cumpliendo con los requisitos de la normativa legal vigente.

Solución final completa para la emisión, recepción y firma electrónica de las distintas versiones del formato FacturaE

Estudio de diferentes estándares implicados en la generación y verificación de facturas (XAdES, FacturaE, etc.).

Junto al presente documento se adjuntará un video en el que se mostrará el funcionamiento completo de la aplicación con todos sus casos de uso implementados

1.2 OBJETIVOS DEL TRABAJO

El objetivo principal es desarrollar una aplicación Web capaz de generar, recibir y gestionar facturas electrónicas, almacenando la información en una BBDD.

El sistema entonces tendrá tres partes diferenciadas una que se encargara de la generación y firma de facturas electrónicas, documentos en formato XML, otra que elegirá verificara las facturas electrónicas firmadas, para estas dos primeras partes el sistema se utiliza servicios externos, y una última parte que se encargara de almacenar la información para su visualización y gestión. La aplicación se integra con servicios de terceros parar poder ofrecer un servicio para la custodia de las claves para poder firmar.

El reto que plantea este trabajo no es tanto la complejidad funcional de la aplicación sino la posibilidad de componer un trabajo completo con varios módulos y obtener como resultado una aplicación original que pueda sentar las bases de a varias ampliaciones en este campo.

Así pues, otro de los objetivos del presente trabajo es el estudio de normativas vigentes sobre facturas electrónicas, el estudio de los formatos de firma digital de documentos, el estudio de las plataformas y mecanismos de generación, verificación de firma digital, la custodia de firmas en la nube.

1.3 ENFOQUE Y METODO SEGUIDO

Se ha decidido desarrollar un nuevo producto, aunque para ello se emplearán librerías existentes, especialmente para la comunicación con el servidor o la presentación de gráficos. Para la custodia de claves la firma de documentos y la validación de firmas se utilizan servicios que nos ofrece la plataforma TrustedX eIDAS y el servicio TrustedX Digital Signature Verification (TWS-DSV)

1.4 METODOLOGIA

Los lenguajes de programación que utilizaran son Java, HTML, XML, y algo de JavaScript para desarrollar la aplicación Web y SQL para las consultas a la base de datos.

El modelo que se utilizará para la programación es el de orientado a objetos. Este es el paradigma que utiliza objetos y sus interacciones para diseñar aplicaciones. Está basado en varias técnicas, entre las que se incluyen la herencia, la abstracción, el polimorfismo y los encapsulamientos, todas ellas utilizadas en el proyecto. El lenguaje utilizado será Java, orientado a objetos y el principal lenguaje en el que se realizan las aplicaciones

El modelo de desarrollo que se seguirá será el "Modelo en cascada" es un proceso de desarrollo secuencial, en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada.

A continuación, explico brevemente las fases:

- Análisis: En esta fase se determinan (o amplían) los requisitos software del sistema, especificando a un alto nivel la arquitectura de la solución que se propone para dichos requisitos.
- Diseño: En esta segunda fase, se diseñan tanto los interfaces de usuario de la aplicación, como la arquitectura a un nivel más bajo de especificación, detallando los procesos del sistema.
- Implementación: En esta etapa se codifican los interfaces de usuario, se codifican los procesos, y se documenta el manual de usuario.
- Pruebas: En esta última fase, se definen las pruebas a realizar por el prototipo en cuestión y se llevan a cabo.

En el aspecto de diseño, se ha utilizado el lenguaje de modelado UML, para especificar y describir objetos, métodos y procesos. Este lenguaje de modelado se

utiliza para definir un sistema, para representar la realidad de una utilización de un requerimiento.

1.5 ESTUDIO DE VIABILIDAD DEL SISTEMA

El objetivo del estudio de viabilidad del sistema es realizar un análisis de tallado de los objetivos con el fin de proponer una solución en un plazo. El estudio debe de tener en cuenta restricciones de diversa índole: económicas, temporales, legales, técnicas y operativas.

Estudio de la solicitud: En la solicitud del trabajo se especifica la necesidad del desarrollar una aplicación web que permita al usuario gestionar completamente facturas electrónicas.

Con el desarrollo de este proyecto se pretende desarrollar una aplicación web que cumpla con las siguientes características:

- Generar y firmar facturas electrónicas.
- Recibir y validar facturas electrónicas.
- Almacenar y gestionar estas facturas electrónicas.

Asimismo, de forma complementaria a dicha aplicación se pretende desarrollar un almacén de claves y certificados en un servidor centralizado con control de acceso, para que el usuario los tenga siempre disponible para poderlo utilizar desde cualquier lugar/ dispositivo.

Valoración del estudio de la situación actual

En la actualidad las aplicaciones que existen para la gestión de facturas electrónicas son aplicaciones de escritorio en la que la información está centrada en un solo equipo y trabaja con ella de manera local. Por lo que no tendríamos acceso a ella desde otros dispositivos. Y requiere un sistema de firma avanzada con PKCS#12 o SmartCard que requiere de hardware especial.

Una vez realizado el estudio de las técnicas y herramientas existentes en la actualidad, se puede concluir que no existe un sistema que proporcione la misma funcionalidad que se pretende alcanzar con este proyecto.

2. CONCEPTOS BASICOS

2.1. FACTURAS

La obligación de generar una constancia documental de las operaciones sujetas al Impuesto sobre el Valor Añadido hace que se normalice y regule el formato que han de contener dichos documentos. Así; el documento que soporta la operación que devenga la liquidación del impuesto no es otro que la factura, documento que ha de cumplir con una serie de requisitos formales para ser considerada como tal. La normativa que regula los requisitos formales de una factura es la siguiente: Real Decreto 1619/2012, de 30 de noviembre, que sustituya al anterior Real Decreto 1496/2003.

Toda factura y sus copias contendrán los siguientes datos o requisitos

- Número y, en su caso, serie. La numeración de las facturas dentro de cada serie será; correlativa.
- Fecha de su expedición.
- Nombre y apellidos, denominación social completa, tanto del obligado a expedir factura como del destinatario de las operaciones.
- Número de identificación fiscal atribuido por la Administración española o, en su caso, por la de otro Estado miembro de la Comunidad Europea, con el que ha realizado la operación el obligado a expedir la factura
- Domicilio, tanto del obligado a expedir factura como del destinatario de las operaciones. Cuando dispongan de varios lugares fijos de negocio, deberá indicarse l ubicación de la sede de actividad

- Descripción de las operaciones, consignándose todos los datos necesarios para la determinación de la base imponible del impuesto; su importe, incluyendo el precio unitario sin impuesto de dichas operaciones, as í como cualquier descuento o rebaja que no esté incluido en dicho precio unitario
- El tipo impositivo o tipos impositivos aplicados a las operaciones
- La cuota tributaria que se repercuta deberá consignarse por separado
- La fecha en que se hayan efectuado las operaciones que se documentan o en la que se haya recibido el pago anticipado, siempre que se trate de una fecha distinta a la de expedición de la factura
- En el supuesto de que la operación esté exenta o no sujeta al IVA o de que el sujeto pasivo del IVA sea su destinatario, se deberá incluir en ella una referencia o indicación de que la operación está exenta o no sujeta o de que el sujeto pasivo del impuesto es el destinatario de la operación.

Todas las facturas deben garantizar:

A. La legibilidad de la factura.
B. La autenticidad del origen de la factura (es decir, garantizar la identidad del obligado a su expedición y del emisor de la factura, que pueden ser la misma persona).
C. La integridad del contenido de la factura (es decir, garantizar que su contenido no ha sido modificado).

2.2. FACTURAS ELECTRONICAS

Con el fin de ganar en eficiencia y ahorrar costes las empresas están sustituyendo las facturas en papel por facturas electrónicas. La factura electrónica es una factura que, ajustándose a los requisitos establecidos en el Real Decreto 1619/2012, de 30 de noviembre, por el que se aprueba el Reglamento de facturación, ha sido expedida y recibida en formato electrónico.
En todo caso la expedición de la factura electrónica estará condicionada a que su destinatario haya dado su consentimiento.
El Reglamento de facturación establece una igualdad de trato entre la factura en papel y la electrónica, siempre que se garantice al obligado a su expedición la autenticidad de su origen, la integridad de su contenido y su legibilidad, desde su fecha de expedición y durante todo el periodo de conservación. En España, la firma electrónica es el mecanismo más generalizado para garantizar la autenticidad de origen y la integridad del contenido de la factura electrónica.

Las facturas electrónicas se pueden emitir en distintos formatos (doc, PDF, EDIFACT, etc.), siempre que se respeten los aspectos señalados en el párrafo anterior. No obstante, tras la publicación de la Orden PRE/2971/2007 [3] el formato XML FacturaE se convirtió en el formato obligatorio para las facturas cuyo destinatario fuera uno de los organismos de la Administración General del Estado (AGE).

La facturación electrónica, que inicialmente era utilizada por la Administración y por las grandes empresas, se va imponiendo y empieza a sustituir a la facturación en papel en todos los ámbitos. La mayoría de las empresas de cierto tamaño ofrecen a sus clientes la posibilidad de emisión de facturas electrónicas.

No obstante, las facturas electrónicas no se pueden validar sin un ordenador o dispositivo que cuente con la aplicación adecuada.

En el caso de la factura electrónica:

La legibilidad la facilita el programa informático que la crea o recibe.

La autenticidad y la integridad se pueden garantizar de diversas formas:

- Mediante firma electrónica avanzada basada en un certificado reconocido.
- Mediante intercambio electrónico de datos EDI.
- Mediante otros medios que los interesados hayan comunicado a la Agencia Estatal de Administración Tributaria con carácter previo a su utilización y hayan sido validados por la misma.
- Mediante los controles de gestión usuales de la actividad empresarial o profesional del sujeto pasivo, siempre que permitan crear una pista de auditoría fiable que establezca la necesaria conexión entre la factura y la entrega de bienes o prestación de servicios que la misma documenta.

2.2.1. Tipos de facturas Electrónicas

Conviene distinguir dos tipos fundamentales de factura electrónica: la factura electrónica con formato estructurado y la factura electrónica con formato no estructurado.

Ambas son documentos electrónicos susceptibles de ser transmitidos mediante redes de comunicaciones electrónicas, como Internet. La diferencia estriba en que el formato estructurado facilita su tratamiento automatizado mientras que el no estructurado no lo facilita.

Facturas en formato estructurado

Las facturas en formato estructurado contienen datos y pueden ser generadas automáticamente por los sistemas informáticos de facturación del emisor y ser tramitadas de forma igualmente automatizada por los sistemas informáticos de pago y contabilidad del receptor. Ejemplos de formatos estructurados son los que utilizan el lenguaje XML (como UBL o FacturaE), EDIFACT, etc.

Facturas en formato no estructurado

Las facturas en formato no estructurado consisten esencialmente en una imagen, lo que implica que su procesamiento para poder ser introducidas en los sistemas informáticos del receptor requiere una intervención manual o un proceso costoso que no suele estar completamente automatizado, como el reconocimiento óptico de caracteres (OCR). Entre estas tenemos las facturas en papel escaneadas y los ficheros PDF.

2.3. FORMATO FACTURAE

A partir del 15-1-15, conforme a la Ley 25/2013, de 27 de diciembre, de impulso de la factura electrónica y creación del registro contable de facturas en el Sector Público, las facturas que se remitan a las Administraciones Públicas serán electrónicas y se ajustarán al formato estructurado de factura electrónica FacturaE versión 3.2.x con firma electrónica XAdES.

2.4 FIRMA ELECTRONICA

La firma electrónica es un conjunto de datos electrónicos que acompañan o que están asociados a un documento electrónico y cuyas funciones básicas son:

- Identificar al firmante de manera inequívoca
- Asegurar la integridad del documento firmado. Asegura que el documento firmado es exactamente el mismo que el original y que no ha sufrido alteración o manipulación
- Asegurar el no repudio del documento firmado. Los datos que utiliza el firmante para realizar la firma son únicos y exclusivos y, por tanto, posteriormente, no puede decir que no ha firmado el documento

La base legal de la Firma electrónica está recogida en la Ley 59/2003 de Firma Electrónica y se desarrolla en más profundidad en la sección Base legal de las Firmas. La sección también explora, bajo qué circunstancias la ley equipara la firma electrónica a la firma manuscrita.

2.4.1. Tipos de Firma

Según la relación entre la localización de los datos firmados y la firma se pueden distinguir tres tipos distintos:

- **Detached** (separada) La firma se encuentra separada del documento firmado.
- **Enveloping** (envolvente) La firma resultante contiene también los datos firmados
- **Enveloped** (incrustada) La firma se encuentra en los datos firmados.

En agosto de 2014 el Parlamento Europeo aprobó un nuevo marco regulatorio para los servicios de identificación y confianza de las transacciones electrónicas en el mercado interior: El nuevo Reglamento (UE) Nº 910/2014**,** conocido como eIDAS, tiene el objetivo de crear un clima de confianza que haga posible y refuerce el comercio electrónico y las transacciones digitales en la UE. Lo que se pretende es eliminar todas las barreras a realizar transacciones electrónicas que existen entre países miembros. Para ello, el eIDAS establece sistemas comunes de identificación de ciudadanos y de validez

de sus firmas electrónicas, para que se pueda operar online con mayor seguridad, agilidad, y eficiencia a nivel europeo.

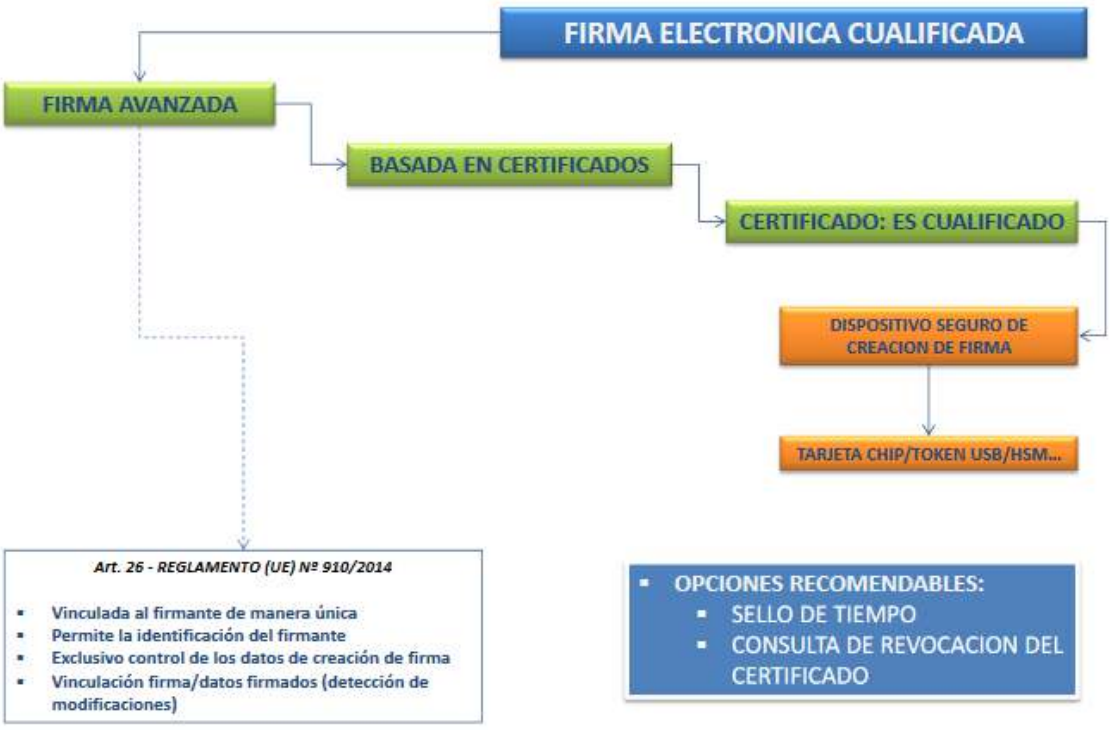
Según el nivel de seguridad de la firma que se regula en el Reglamento (UE) Nº 910/2014 se clasifican de la siguiente manera:

Firma electrónica, comúnmente denominada como simple, es la que tiene un nivel más bajo de seguridad. Ofrece tanto al firmante como a quien solicita la firma es muy limitada, su uso ha sido generalizado.

La firma electrónica avanzada tiene un nivel de seguridad superior al de la firma simple: las dos primeras condiciones definidas por el Reglamento eIDAS aseguran que el firmante sólo puede haber sido aquel a quien se solicitó la firma, mientras que las dos últimas limitan extraordinariamente el riesgo de suplantación de identidad. Además, es una evidencia jurídica admisible como prueba en un juicio.

Firma electrónica Cualificada: es la que ofrece un nivel de seguridad más alto, su uso se ve entorpecido por la necesidad de disponer de un certificado cualificado de firma electrónica (DNIe) y de un dispositivo seguro de creación de firma cualificado. Un Certificado Cualificado de Firma Electrónica será Cualificado si ha sido expedido por un prestador cualificado de servicios de confianza y cumple con una serie de requisitos que se establecen en la Regulación (UE) Nº 210/2014. Condiciones que no se exigen a la firma electrónica avanzada, tal y como hemos explicado.

Por este motivo, el uso de la firma electrónica cualificada suele limitarse a trámites que se realizan con las administraciones públicas, como hacienda o la seguridad social. Su complejidad operativa no la hace una opción recomendable para empresas o personas que deban solicitar firmas a distancia de forma frecuente.



2.4.2 XMLDSig

El estándar XMLDSig recoge las reglas básicas de creación y procesamiento de firmas electrónicas de documentos XML. Es muy similar al PKCS#7 pero mucho más extensible y está pensado especialmente para firmar documentos XML, aunque puede usarse para cualquier otro tipo de datos. Dicho estándar se amplía con las especificaciones de XADES, donde se definen estructuras que permiten incorporar información adicional a la firma que facilita su validación. El cumplimiento de los estándares permite el reconocimiento de la firma por toda la comunidad electrónica, si bien su flexibilidad permite distintos grados de libertad que desde FacturaE se precisa acotar para su aplicación a la factura electrónica.

2.4.3 XADES

Las firmas XAdES son una evolución de las firmas XMLDSig a la que añaden ciertas extensiones y en la que se concretan con más definición algunas operaciones, como las contrafirmas.

El formato XAdES admite múltiples variantes, con distintas aplicaciones (desde sellos de tiempo hasta archivo longevo).

La firma en formato de factura electrónica es en realidad una firma XAdES Enveloped particular. La firma de facturas se realiza de acuerdo a la versión 3.1 del esquema de factura electrónica.

Detalles a tener en cuenta del formato de firma de factura son:

- Los datos de entrada deben ser una factura electrónica conforme a las normas FacturaE.
- No contempla las operaciones de confirma ni contrafirma.
- Las firmas son siempre implícitas, por lo que no se atenderá a la configuración de modo del cliente.
- No es necesario configurar la política de firma de factura electrónica. La política se establece automáticamente cuando se utiliza este formato.

Se definen dos formatos de firma electrónica.

2.4.3.1. Formato de firma electrónica avanzada básico.

Contiene los elementos mínimos y necesarios para que la firma se considere firma electrónica avanzada acorde con la Ley 59/2003, de 19 de diciembre, de firma electrónica. Incluye los campos que a continuación se detallan.

La firma se considera un campo más a añadir en el documento de factura y además se debe aplicar a: Todos los elementos de la factura, Los elementos de firma ubicados en el contenedor “SignedProperties” y El certificado digital con el que se ha firmado incluido en el elemento “KeyInfo”.

Es necesario utilizar el elemento ds:KeyInfo, conteniendo, al menos, el certificado firmante codificado en base64. Además, dicha información precisa ser firmada con objeto de evitar la posibilidad de sustitución del certificado.

El formato básico de firma electrónica avanzada no incluye ninguna información de validación más allá del certificado firmante, que está incluido en el campo ds:KeyInfo.

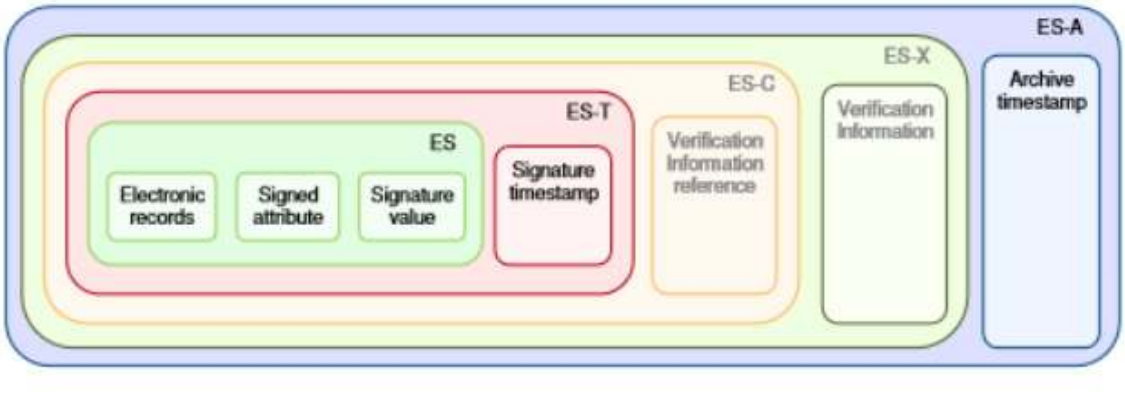
```
<ds:Signature >
        <ds:SignedInfo/>
        <ds:SignatureValue/>
        <ds:KeyInfo>
          <ds:X509Data>
              <ds:X509Certificate/>
          </ds:X509Data>
        </ds:KeyInfo>
        <ds:Object>
            <XAdES:QualifyingProperties>
            <XAdES:SignedProperties>
                <XAdES:SignedSignatureProperties>
                    <XAdES:SigningTime />
                    <XAdES:SigningCertificate/>?
                    <XAdES:SignaturePolicyIdentifier/>
                    <XAdES:SignerRole/>?
                </XAdES:SignedSignatureProperties>
            </XAdES:SignedProperties>
            <XAdES:UnSignedProperties/>
          </XAdES:QualifyingProperties>
        </ds:Object>
    </ds:Signature>
```

2.4.3.2. Formato de firma electrónica avanzada con información de validación

El formato incorpora, al formato básico, información adicional a la firma necesaria para su validación. Por ejemplo, sellado de tiempo, evidencia de que la firma existía antes de un determinado momento en el tiempo, información sobre la cadena de certificación y el estado de revocación de los mismos. La inclusión de esta información a la firma proporciona evidencias ante terceros o potenciales arbitrajes. Asimismo, asegura la validación de la firma a largo plazo, independientemente de que exista o no la Autoridad de Certificación.

A continuación, especifico los campos adicionales:

```
<XAdES: UnSignedSignatureProperties>
    <XAdES: SignatureTimeStamp />*
    <XAdES: CompleteCertificateRefs/>
    <XAdES: CompleteRevocationRefs/>
    (<SigAndRefsTimeStamp>*| <RefsOnlyTimeStamp>*)
    <XAdES: CertificateValues>
    <XAdES: RevocationValues/>
</XAdES: UnSignedSignatureProperties>
```



Esquema de los distintos formatos de firma.

2.4.4. Archivado y conservación

El archivado de firmas conforme a la presente política consiste en almacenar el documento firmado según la definición del formato de firma electrónica avanzada con información de validación. Si como método de verificación del estado del certificado firmante, se utilizó consultas a CRLs, será necesario almacenar la CRL consultada, que, además, deberá incluir fecha y firma del responsable de su expedición. Opcionalmente, se podrá almacenar información sobre el estado de los certificados de las Autoridades de Certificación pertenecientes a la cadena de confianza del certificado firmante.

Las obligaciones de conservación de facturas se establecen en los artículos 1, 19, 20, 21, 22 y 23 del Reglamento por el que se regulan las obligaciones de facturación, aprobado por el Real Decreto 1619/2012, de 30 de noviembre. De los artículos citados, los artículos 20 y 21 se refieren específicamente a la factura electrónica.

También mantiene su vigencia en lo que no se oponga a este Real Decreto 1691/2012, la Orden EHA/962/2007, de 10 de abril, por la que se desarrollan determinadas disposiciones sobre facturación telemática y conservación electrónica de facturas

2.4.5. Certificados Electrónicos

El certificado electrónico (o certificado digital) reconocido es un documento emitido y firmado por una Autoridad Certificadora Reconocida que dispone de los medios para emitir un certificado electrónico que vincula de forma inequívoca al suscriptor (personas físicas o jurídicas) de dicho certificado con unos datos de emisión y verificación de firma electrónica y confirma de forma inequívoca su identidad "digital".

Se consideran válidos para ejecutar la firma conforme a la presente política, todos aquellos certificados que cumplan con lo indicado en los apartados a) o c) del artículo 18 del Reglamento por el que se regulan las obligaciones de facturación y que está recogido en el R. D. 1496/2003 de 28 de noviembre.

Atendiendo a la normativa, existen dos tipos de certificados electrónicos fundamentales:

- Certificado electrónico: es un documento firmado electrónicamente por un prestador de servicios de certificación que vincula unos datos de verificación de firma a un firmante y confirma su identidad.
- Certificado reconocido: es un certificado electrónico que cumple con los requisitos recogidos en la Ley de Firma Electrónica 59/2003 tanto en cuanto a su contenido, como en ciertas condiciones que debe cumplir el prestador de servicios de certificación.

Atendiendo a la normativa actual podemos diferenciar entre certificados de persona física, de persona jurídica, de entidad sin personalidad jurídica y certificado de Administración Pública.

A partir del 1 de julio de 2016 deben dejar de emitirse certificados de firma electrónica a favor de personas jurídicas o entidades sin personalidad jurídica, si bien estos certificados (conforme a las directrices del Ministerio de Industria, Energía y Turismo) podrán seguir utilizándose hasta su caducidad o revocación. Para sustituir a dichos certificados, se podrán utilizar certificados de firma electrónica de representante de personas jurídicas o entidades sin personalidad jurídica.

De acuerdo con esta nueva normativa los tipos de certificados son:

- Certificado de persona física: es el que identifica a una persona individual.
- Certificado de representante de persona jurídica: se expide a las personas físicas como representantes de las personas jurídicas.
- Certificado de representante entidad sin personalidad jurídica: se expide a las personas físicas como representantes de las entidades sin personalidad jurídica en el ámbito tributario y otros previstos en la legislación vigente.

- Certificados AP (Administración Pública).

2.4.5.1 Certificado Electrónico X.509

La definición del certificado digital se encuentra contenida en la especificación ISO/IEC 9594-8 (o su equivalente Recomendación X.509 de ITU-T)

Versiones 1 y 2 del certificado X.509

La estructura del certificado lo componen 7 campos que según la especificación (IETF, 2008) son definidas así:

A. Versión. hace referencia a la versión del certificado conforme a la cual están definidos formalmente sus distintos campos. El valor 0 indica la versión 1 y el valor 1 a la versión 2.
B. Número de serie: cada CA debe numerar correlativamente todos los certificados que emita, de forma número de serie sirve de identificador único para este certificado.
C. Algoritmo de firma del certificado (signature): especificación del tipo de algoritmo utilizado para cifrar la firma del certificado. Normalmente será RSA o DSA, pero tal y como está concebido el certificado X.509 puede ser cualquier otro algoritmo que sean capaces de manejar las entidades que se fían de los certificados emitidos por esa CA.
D. Nombre de la CA emisora (issuer): en la especificación del certificado está previsto que este campo recoja el nombre el nombre X.500 de la CA.
E. Validez. Indica el comienzo y el final del período de tiempo durante el cual el certificado es válido.
F. Nombre del usuario o titular (subject): es el nombre x.500 de la entidad adscrita a esta CA a la que se le ha expedido el certificado. Puede tratarse de una CA a la que otra CA le haya generado un certificado.
G. Información sobre la clave del usuario: es evidentemente, el componente principal del certificado. Consta a su vez de dos elementos:
    a. Algoritmo con que será usada: identificador del algoritmo con el que se ha previsto que la clave pública sea usada.
    b. Valor de la clave pública (subjecgt public key): es el valor de la clave pública de la que es propietaria la entidad comunicante para la que se ha emitido el certificado.

Para la versión 2 aprobada en 1993 se incluyen los siguientes campos optativos:

Identificador único de CA emisora. Se trata de una cadena de bits, sin formato específico, que de forma opcional sirve para contener información adicional sobre la CA emisora del certificado.

2.4.5.2. Autoridad de Certificación

Una autoridad de certificación o CA es la entidad de confianza para todos los participantes que se encarga de firmar y emitir los certificados digitales.

Un certificado firmado por una CA es un certificado válido ya que las CA son de confianza, por lo tanto, su firma tiene validez.

La confianza de la CA se basa en dos factores:

1. La CA sigue suficientes procedimientos de verificación de los datos del titular y verifica que su clave pública sea realmente suya.
2. La clave privada de la CA se mantiene bajo suficientes medidas de seguridad.

Existe una jerarquía de verificación que consiste en que las CA utilizan su clave privada para firmar los certificados de otras CA, por lo que la firmante es superior en la jerarquía. Las CA iniciales en las jerarquías se llaman CA raíz, y sus certificados son auto-firmados.

Dado que la PKI se basa en la confianza si se decide confiar un una determinada CA raíz, se confiará en todas las CA cuyos certificados hayan sido firmados por la CA raíz y así sucesivamente.

2.4.5.3. Autoridad de registro

El concepto de Autoridad de Certificación se ha tratado anteriormente: es un sistema que dispone de un par de claves que se aplican sobre las peticiones de certificación, emitiendo certificados.

No obstante, los datos que llenan la CA (nombres de titular, su clave pública, etc.) están bajo control. Si la validación de una petición es un proceso automático –la petición de certificación posee una firma verificable con una clave pública incorporada-, el nombre del titular debe verificarse.

La Autoridad de registro (RA) comunica las entidades que solicitan certificados con las CAs. Controla la generación de certificados verificando los datos proporcionados por las entidades y emitiendo una petición del certificado a la CA correspondiente, que incluye información del solicitante, su clave pública y un fragmento firmado con su clave privada. Cuando el certificado es generado lo devuelve a la entidad solicitante.

Repositorio de certificados

Los repositorios de certificados son almacenes de certificados válidos, es decir, que no han caducado ni han sido revocados, que proporcionan servicios a las entidades para que puedan descargárselos. Suelen estar en directorios X.5002 accesibles mediante LDAP3.

2.4.5.4. Repositorio de listas de certificados revocados

En la sección de certificados digitales se ha mencionado que los certificados tienen fecha de caducidad. Cuando un certificado caduca deja de ser válido, por lo tanto, cualquier firma que haya realizado el propietario de certificado caducado deja de ser válida.

Para saber que un certificado ha caducado basta con comprobar su fecha de caducidad, pero existe otro motivo por el cual un certificado tiene que dejar de ser válido y es que la seguridad de la clave privada haya sido comprometida.

Cuando la entidad propietaria de un certificado sospecha que su clave privada puede haber sido comprometida, informa a la CA que la ha firmado para que incluya el certificado en su repositorio de listas de certificados revocados o CRL.

Una CRL contiene los números de serie de los certificados que ya no son válidos por motivos ajenos a su fecha de caducidad. Las listas son emitidas por las CA periódicamente y firmadas por estas para asegurar su validez.

Como el tiempo entre la revocación del certificado y la emisión de una nueva lista de certificados revocados no es inmediato, las operaciones de verificación de firmas digitales no se pueden realizar hasta pasado un período de gracia desde que fue realizada la firma. Así se asegura que el certificado estaba vigente cuando se realizó dicha firma. Los puntos de distribución de CRL se llaman CRL Distribution Point, usualmente están implementados como los repositorios de certificado.

2.4.6. Sellado de Tiempo

El sellado de tiempo puede garantizar la integridad del conjunto de datos electrónicos que conforman la firma electrónica. Es decir, el sello de tiempo garantiza que una firma llevada a cabo en un momento dado no puede modificarse.

Además, el sello de tiempo también garantiza la no alteración de una serie de datos asociados con la firma electrónica, como la fecha, hora y lugar de realización de la firma, la dirección de correo del emisor del documento a firmar, la dirección de correo del firmante, etc.

En este sentido, el sellado de tiempo aporta un valor extra a la firma electrónica, no sólo por las garantías que da respecto a la integridad de los datos que conforman la firma, sino también porque incluye información relevante acerca del momento de su creación.

Un sellado de tiempo debe de ser aportado una tercera de confianza, que se conoce como Autoridad de Sellado de Tiempo.

2.4.6.1 Autoridades de sellado de tiempo

Se admiten sellos de tiempo expedidos por aquellas Autoridades de Sellado de Tiempo que cumplan con la norma ETSI TS 102 023 “Policy requirements for time-stamping authorities”.

Una Autoridad de Sellado de Tiempo (la traducción al castellano del término inglés *Time Stamp Authority*, abreviado a menudo como TSA), es un prestador de servicios de certificación, cuyo servicio es precisamente actuar como tercero de confianza ofreciendo sellos de tiempo oficiales.

Las autoridades de sellado de tiempo utilizan una tecnología llamada infraestructura de clave pública o PKI (del inglés, *public key infrastructure),* que permite ejecutar varios tipos de operaciones criptográficas, como por ejemplo el cifrado y descifrado de comunicaciones.

En el caso de aplicar un sello de tiempo a una firma electrónica, el funcionamiento es el siguiente:

1. Cuando se firma un documento, enviamos a la Autoridad de Sellado de Tiempo un valor hash que representa los datos del documento firmado, incluidos los datos de la firma (firma encriptada).
2. La Autoridad de Sellado de Tiempo nos devuelve un hash distinto, que es el sellado de tiempo (que no deja de ser un certificado de tiempo).
3. Con este sello de tiempo se garantiza la integridad de todos los datos que forman parte del documento. Si se modificasen estos datos en algún momento posterior al sellado, éste se rompería.

2.4.7. Algoritmos

Para los entornos de seguridad genérica se tomará la referencia a la URN en la que se publican las funciones de hash y los algoritmos de firma utilizados por las especificaciones XAdES, como formatos de firma adoptados, de acuerdo con las especificaciones técnicas ETSI TS 102 176-1 sobre “Electronic Signatures and Infraestructures (ESI); Algorithms and parameters for secure electronic signature”. Todo ello sin perjuicio de los criterios que al respecto pudieran adoptarse en el Esquema Nacional de Seguridad, desarrollado a partir del artículo 42 de la Ley 11/2007.

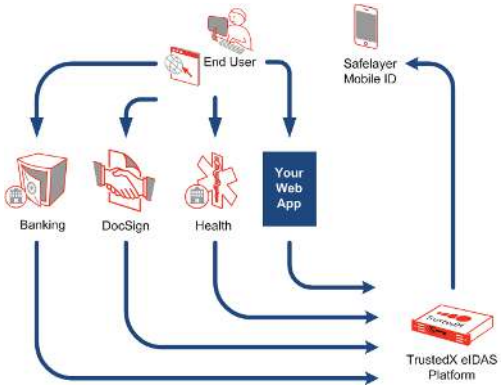
Para los entornos de alta seguridad, de acuerdo con el criterio del Centro Criptológico Nacional, CCN, serán de aplicación las recomendaciones revisadas de la CCN-STIC 405.

Se podrán utilizar cualquiera de los siguientes algoritmos para la firma electrónica: RSA/SHA1 (formato que se recomienda reemplazar en el medio plazo por algoritmos más robustos), RSA/SHA256 y RSA/SHA512 que es recomendado para archivado de documentos electrónicos (very long term signatures).

Se admitirán como válidos los algoritmos de generación de hash, codificación en base64, firma, normalización y transformación definidos en el estándar XMLDSig.

2.5. TRUSTEDX

TrustedX es una plataforma de identificación, autenticación y firma electrónica centrada en el usuario para entornos Web. Basada en SOA, desarrollada por la empresa Safelayer Secure Communications, S.A cuyo objetivo es simplificar el uso de servicios de confianza basados en Infraestructuras de Clave Pública para firma electrónica, protección de datos y cualquier tipo de gestión de identidad electrónica.



Dada su arquitectura orientada a servicios la integración con otras plataformas o aplicaciones resulta muy sencilla, reduciendo la complejidad que hasta la fecha suponía el dotar a cualquier aplicación de mecanismos de seguridad y *PKI*.

2.5.1 Autenticación

La autenticación es el acto de confirmar que algo es auténtico, en el contexto de este proyecto sería confirmar que la identidad del usuario es auténtica.

Existen diferentes métodos para confirmar que una persona afirma ser quien dice ser y se pueden basar en los siguientes factores:

- Algo conocido: por ejemplo, una contraseña, que es sin duda el método de autenticación más popular, dada su simplicidad ya que el usuario únicamente tiene que recordar la secuencia de datos que componen su contraseña.
- Algo poseído: este método de autenticación consiste en poseer algo único y que sólo puedes tener tú, por ejemplo, una tarjeta o un token numérico que cambia según demanda. La autenticación en dos pasos utiliza este factor de autenticación. Consiste en después de confirmar la contraseña del usuario, solicitar un el valor numérico de un token de autenticación que está en su poder.
- Algo físico: también llamada autenticación por biometría, consiste en verificar alguna característica biológica, como puede ser la sangre, la retina, las huellas dactilares, etc.

En este proyecto se mencionarán los métodos de autenticación de usuario y contraseña y la autenticación en dos pasos por OAuth2.

2.5.2 Servidor IdP (proveedor de identidad)

Un proveedor de identidad (IdP de las siglas en inglés Identity Provider) es un servicio externo que permite autenticar a un usuario en otro servicio sin necesidad de que este cree unas credenciales nuevas.

TrustedX incluye un proveedor de identidad propio para autenticar a los usuarios que deseen acceder a los servicios ofrecidos por TrustedX. Este proveedor de identidad puede federarse con Google, Facebook, etc. Para este TFM utilizaremos un dominio de identidad. Esto significa que los usuarios deberán autenticarse también contra el dominio de la plataforma TrustedX.

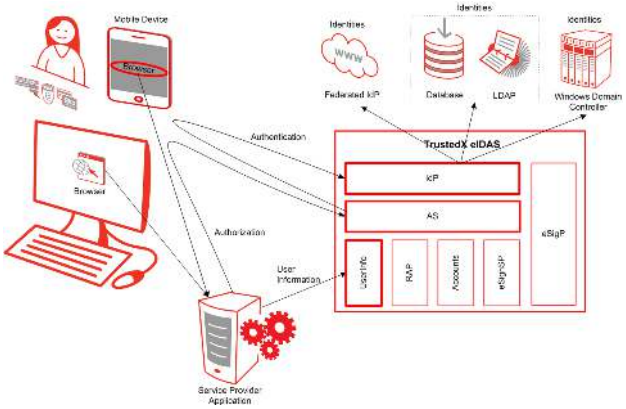
La aplicación cliente no deberá integrarse directamente contra el IdP ya que las sesiones de los usuarios estarán controladas a través del servidor de autorización.

2.5.3. Servidor de Autorización

TrustedX ofrece un servidor de autorización compatible con el protocolo OAuth 2.0 el cual controlará el acceso a los recursos del servidor principal (identidades de firma).

Este servidor de autorización está directamente relacionado con el IdP del dominio de TrustedX con lo que toda autorización OAuth2 (Authorization Code Grant) redireccionará al usuario automáticamente al IdP en caso que el usuario no esté autenticado en nuestro dominio.

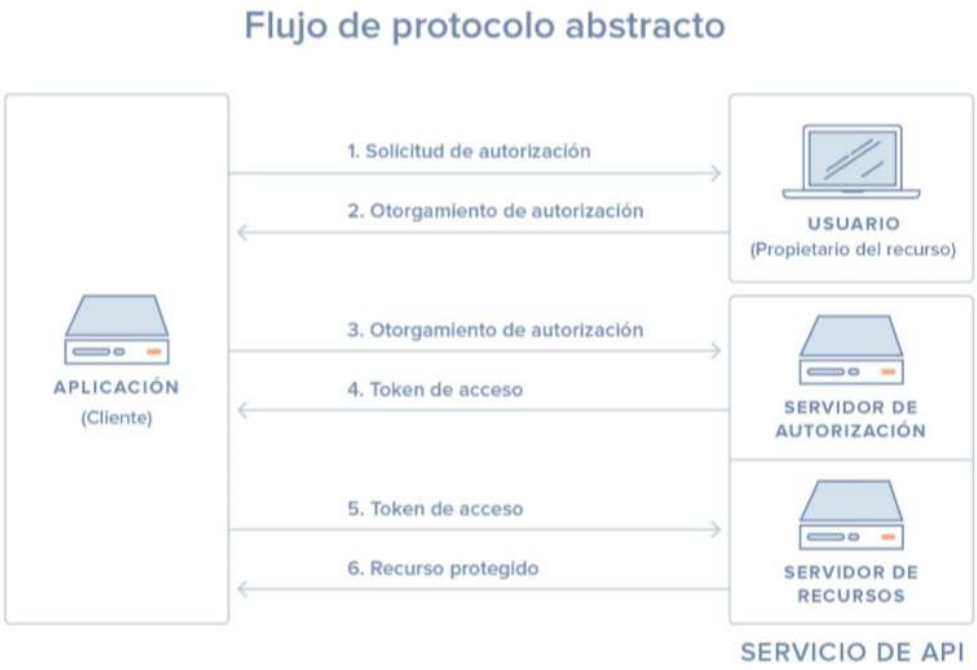
Los recursos que se protegerán son las Identidades de Firma. Para poder crear, administrar o usar estas identidades de firma, la aplicación cliente deberá solicitar autorización al usuario para poder operar en su nombre desde la aplicación.



2.5.3.1 OAuth

OAuth 2.0 es un protocolo de autorización que permite a aplicaciones de terceros obtener acceso limitado a un servicio HTTP, tanto en nombre del propietario del recurso (resource owner) mediante una interacción de aprobación entre el resource owner y el servicio HTTP como dando permisos a una aplicación de terceros para obtener acceso en su nombre.

El RFC del protocolo OAuth 2.0 fue aprobado en octubre de 2012 dejando obsoleto su predecesor, OAuth 1.0 descrito en el RFC 5849.



2.5.4. Proveedor de Firma eSigP

Este componente es el servidor de recursos protegido por el servidor de autorización comentado anteriormente, que proporciona información confiable sobre las identidades de firma de los usuarios y el acceso a la generación de firmas digitales mediante las claves de estas identidades. La responsabilidad principal es custodiar las identidades de firma de los usuarios.

Tal como se ha visto con anterioridad, toda operación que realice la aplicación cliente sobre este proveedor de firma deberá estar autorizada por el usuario mediante un Token OAuth2 obtenido con anterioridad.

2.5.4.1. Identidad de Firma

Una identidad de firma es un conjunto de datos manejados por TrustedX eIDAS para administrar la selección y el uso de una clave de firma específica. Cada identidad de firma contiene información sobre el usuario que posee la clave, el certificado X.509 asociado a esta clave y las etiquetas que permiten seleccionarlos.

Dependiendo de dónde se encuentre la clave de firma a la que se refiere una identidad de firma, se hace una distinción entre:

- Una identidad de firma en un dispositivo móvil, cuando la clave de firma está en un móvil en posesión del usuario.
- Una identidad de firma en un servidor, en cuyo caso la clave de firma está en un HSM o en una base de datos a la que el eSigP tiene acceso. Cuando la clave de

firma está en un dispositivo HSM de Thales acreditado como dispositivo seguro de creación de firma, se dice que la identidad de firma correspondiente a la clave en cuestión es una identidad de firma de servidor calificada.

2.5.4.2 Habilitar identidad de firma en el servidor

Es la forma en que la clave de firma asociada a la identidad está disponible para generar una firma electrónica. El mecanismo de habilitación de identidad de firma del servidor TrustedX eIDAS requiere autenticación a través del envío de un token de acceso. Este token de acceso debe contar con la autorización del propietario de la identidad para utilizar las claves de firma de todas las identidades de firma del servidor que posee el sujeto en un esquema particular. Si la identidad de firma del servidor es una identidad de firma calificada,

2.5.5. Servicio de Firma eSignSP

Este componente es el servidor encargado de realizar las firmas sobre los diferentes tipos de documentos (PDF, XML, binario) en sus diferentes formatos estándar (PAdES, XAdES, CAdES). eSignSP hace uso de las identidades de firma registradas a través del servicio eSigP para la firma electrónica del documento. Así pues, eSigP custodia como recursos las claves de los usuarios y eSignSP se encarga de realizar firmas bajo formatos estándares de documentos utilizando estas claves de los usuarios.

2.5.6. TrustedX DSV (Digital Signature Verification)

Servicio de verificación de firmas (incluidas firmas avanzadas o longevas) independiente del prestador, del mecanismo de verificación de certificados y del formato de firma. Este servicio utiliza una interfaz SOAP (XML) a diferencia de los servicios vistos con anterioridad. Este servicio es capaz de recibir una firma en formato XAdES, CAdES, PAdES y verificar la firma y devolver la máxima información relacionada con la firma posible.

3. ANÁLISIS

3.1. OBEJTIVO DEL ANALISIS

Al realizar el Análisis del Sistema se pretende obtener una colección completa y detallada de los requisitos del sistema, tomando como punto de partida los requisitos identificados en el Estudio de Viabilidad del Sistema.

El documento generado en esta fase será la base para el desarrollo de la fase de Diseño del Sistema, en el que se especificará el diseño completo del sistema de información.

3.2. ALCANCE DEL ANALISIS

Esta fase tiene como objetivo obtener una especificación detallada del sistema que se va a diseñar. Mediante su producto, el Documento de Análisis del Sistema, se pretende captar cuales son las necesidades que tiene el cliente.

En primer lugar, se definirá cual es el alcance del sistema que se desea desarrollar, así como el entorno tecnológico asociado al proyecto. Además, se identificarán los diferentes participantes que aparecen a lo largo de la vida del proyecto, así como los usuarios finales.

A continuación, se definirán cuáles son los requisitos software que debe cumplir el sistema a desarrollar, tomando como punto de partida los casos de uso y los requisitos de usuario.

Por último, se definirán las interfaces de usuario que se utilizarán, dejando como última tarea las comprobaciones de calidad sobre los distintos modelos y requisitos software que se han generado durante la fase de análisis.

3.3. DEFINICION DEL SISTEMA

En este apartado se determina el alcance del sistema a desarrollar, para que permita satisfacer las necesidades planteadas.

El sistema a desarrollar consiste en una aplicación Web que permita a los usuarios de esta herramienta la generación, recepción y gestión completa de facturas en formato electrónico. Cumpliendo con los requisitos de la normativa legal vigente.

Solución final completa para la emisión, recepción y firma electrónica de las distintas versiones del formato FacturaE

Estudio de diferentes estándares implicados en la generación y verificación de facturas (XAdES, FacturaE, etc.).

3.4 RESTRICCIONES GENERALES

A continuación, se detallan las restricciones que deberá cumplir el sistema a diseñar:

En un primer lugar la aplicación funcionará en cualquier S.O. solo será necesario un navegador para poder utilizarla.

La interfaz, de la aplicación Web, con la que interaccionará el usuario final, debe proporcionar todas las funcionalidades descritas en los requisitos software del sistema, de una forma amigable e intuitiva, evitando posibles ambigüedades que puedan ocasionar confusión al usuario.

El idioma usado en el desarrollo del sistema será el castellano.

3.5 IDENTIFICACION DE LOS USUARIOS

En este apartado se van a detallar los usuarios que participan en el proceso de análisis del sistema, así como los usuarios que lo validarán y aceptarán finalmente.

En primer lugar, en el proceso de análisis van a participar:

- El desarrollador: La persona encargada de llevar a cabo el desarrollo del presente proyecto
- Cliente o tutor: En este caso tutor, que espera un correcto desarrollo del producto final en términos de calidad tiempo y coste, Además, son parte importante a la hora de la obtención de los requisitos del sistema a diseñar, con el objetivo de obtener un producto que cumpla con las necesidades que le han llevado a solicitar el sistema.

En cuanto a los usuarios finales del sistema, éstos serán los que se describen a continuación:

- Usuarios: Son aquellas personas que harán uso del sistema desarrollado. Se encargarán tanto de la configuración del sistema como de su puesta en producción.

3.6. REQUISITOS

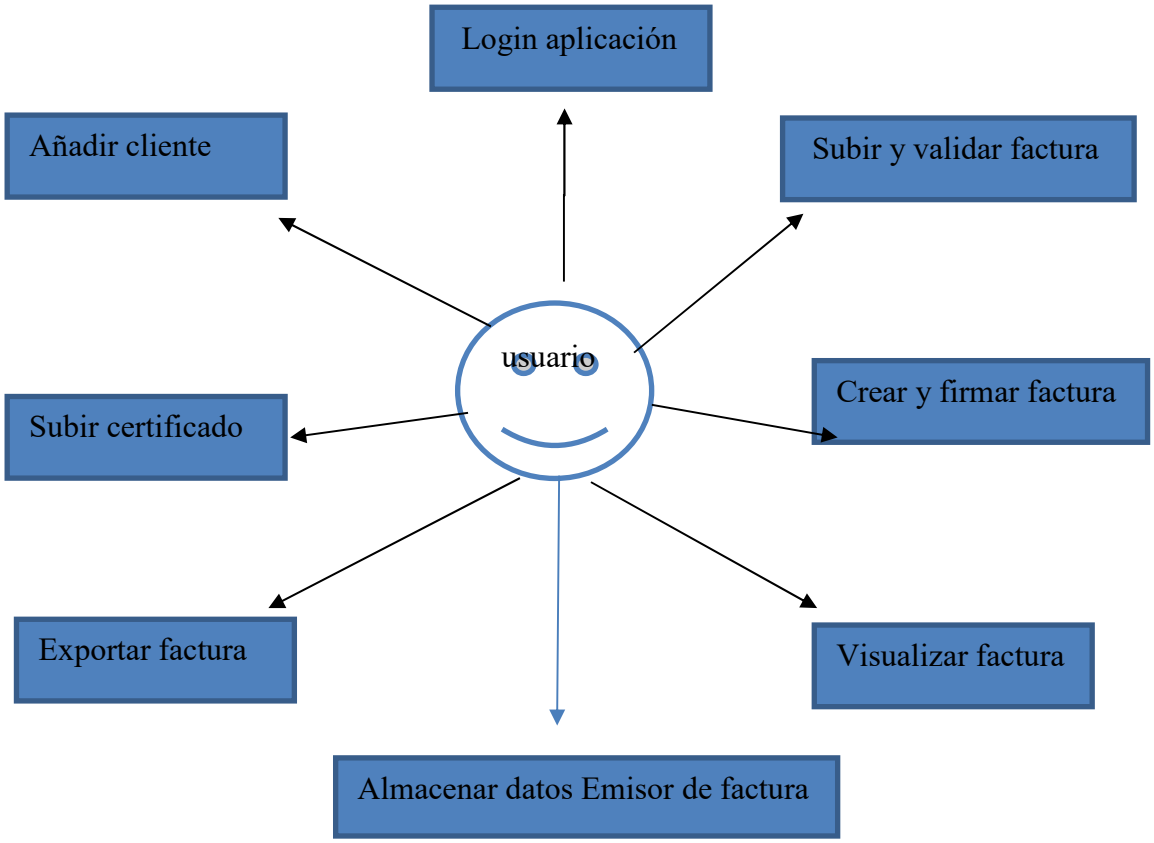
Se realiza la extracción de requisitos con el fin de cumplir con las funcionalidades debe proporcionar la aplicación.

Esta extracción de requisitos se realiza también con el objetivo de orientar al cliente, permitiendo al equipo de desarrollo obtener una lista de requisitos detallada, completa y sin ambigüedades. Estos requisitos proporcionarán una visión general de la aplicación, sin ahondar en aspectos técnicos, estableciendo las principales funcionalidades y restricciones, sirviendo de base a posteriores procesos del ciclo de vida.

Previamente a la redacción de los requisitos han sido identificados los casos de uso con el fin de lograr una mejor definición.

3.6.1. ESPECIFICACION DE LOS CASOS DE USO

A continuación, se muestran los diagramas de casos de uso para cada uno de los módulos del sistema que se van a desarrollar en este proyecto:



3.6.1.1.ESPECIFICACION DE LOS CASOS DE USO DETALLADO

En este apartado se realiza la especificación de los casos de uso. Cada caso de uso estará especificado por los siguientes atributos:

- **Identificador:** Identifica al caso de uso de forma única. Debe seguir el formato: *CU-XX*, siendo XX un valor numérico único para cada caso de uso.
- **Nombre:** Breve especificación textual del caso de uso.
- **Actores:** Tipo de usuario del sistema que inicia el caso de uso.
- **Objetivo:** Finalidad del caso de uso.
- **Precondiciones:** Estado previo que se debe cumplir para poder realizar una operación.
- **Post-condiciones:** Estado en el que queda el sistema tras realizar una operación.
- **Escenario básico:** Especifica la manera en la que interactúa un actor con el sistema y cuál es la respuesta que el sistema le ofrece.

Identificador	*CU-01*
Nombre	Login aplicación
Actor	Usuario
Objetivo	Autenticarse en la aplicación con los credenciales
Precondiciones	Se debe tener conexión a internet. Debes conocer un usuario y contraseña
Post-Condiciones	Acedes a los recursos de la aplicación
Escenario Básico	El usuario introduce en el formulario las credenciales .Accedes a la página principal de la aplicación.

TABLA Caso de uso CU-01

Identificador	*CU-02*
Nombre	Añadir cliente
Actor	Usuario
Objetivo	Almacenar la información sobre un cliente en la base de datos.
Precondiciones	El usuario se debe haber loggeado en la aplicación
Post-Condiciones	La información se almacena en la base de datos.

Escenario Básico	La aplicación muestra un formulario. El usuario introduce información en el formulario y pulsa guardar La aplicación almacena la información en BBDD

TABLA Caso de uso CU-02

Identificador	*CU-03*
Nombre	Subir certificado
Actor	Usuario
Objetivo	Almacenar en el servidor de internet un certificado
Precondiciones	El usuario debe estar logado en el sistema El usuario debe poseer un certificado, en el dispositivo.
Post-Condiciones	Las claves/certificado se almacenan en el servidor de internet.
Escenario Básico	El usuario accede a la página de subir certificado El usuario selecciona el archivo que contiene el certificado en el equipo local La aplicación almacena el certificado en el servidor centralizado por TrustedX eIDAS.

TABLA Caso de uso CU-03

Identificador	*CU-04*
Nombre	Subir y validar factura
Actor	Usuario
Objetivo	Subir al servidor una factura que posee el usuario y validarla
Precondiciones	El usuario se a loggeado en el sistema. El usuario posee una factura en el dispositivo en formato PDF o XML
Post-Condiciones	La aplicación comprueba el documento y almacena información en la BBDD
Escenario Básico	El usuario accede a la página de subir factura El usuario selecciona el archivo en formato XML o PDF que contiene la factura en el equipo local. La aplicación utiliza el servicio TrustedX digital signature verification para validar la firma de la factura introducida. Y almacena en la BBDD la información de la factura

TABLA Caso de uso CU-04

Identificador	*CU-05*
Nombre	Crear y firmar factura
Actor	Usuario
Objetivo	Introducir información sobre una factura que se firmara con un certificado después.
Precondiciones	El usuario debe haberse logado en el sistema Se debe haber subido al sistema anteriormente el certificado Se debe haber introducido en el sistema el cliente anteriormente
Post-Condiciones	Se almacena en BBDD la información de la factura Se obtiene una documento firmado.
Escenario Básico	El usuario entra en la página de alta de factura. El usuario selecciona el cliente al que se va a generar la factura El usuario inserta la información de la factura

	El sistema almacena la información. El usuario selecciona el certificado. Se descarga en el dispositivo local el archivo..

TABLA Caso de uso CU-05

Identificador	*CU-06*
Nombre	Visualizar factura
Actor	Usuario
Objetivo	Se muestra en el dispositivo una factura almacenada en el sistema
Precondiciones	El usuario debe haberse logado en el sistema Se debe haber subido al sistema o creado anteriormente la factura.
Post-Condiciones	Muestra en pantalla la información de la factura debidamente formateada
Escenario Básico	El usuario selecciona la factura. El sistema muestra en pantalla la factura

Tabla Caso de uso CU-06

Identificador	*CU-07*
Nombre	Exportar factura
Actor	Usuario
Objetivo	Se descarga en el dispositivo un archivo en formato XML o PDF con la factura firmada
Precondiciones	El usuario debe haberse logado en el sistema Se debe haber subido o creado en el sistema la factura
Post-Condiciones	Se descarga en el dispositivo un archivo en formato XML o PDF con la factura firmada Se obtiene una documento firmado.
Escenario Básico	El usuario selección la factura Se descarga en el dispositivo local el archivo..

Tabla Caso de Uso CU-07

Identificador	*CU-08*
Nombre	Almacenar datos Emisor de factura
Actor	Usuario
Objetivo	Almacenar en BBDD la información necesaria para emitir facturas
Precondiciones	Se debe tener conexión a internet. El usuario se debe de haber loggeado en la aplicación
Post-Condiciones	Información almacenada en la BBDD
Escenario Básico	El usuario introduce en el formulario toda la información necesaria del emisor de la factura para poder realizar una factura La información se almacena en nuestra BBDD

TABLA Caso de Uso CU-08

3.7. REQUISITOS DE FIRMA

Uno de los aspectos más importantes de este trabajo se centra en la calidad de la firma con la que se firmarían nuestras facturas.

Cabe destacar que en el caso de uso CU-05 Firmar Factura la firma que se creará será de tipo avanzada, ya que las firmas se realizarán a partir de la importación de archivos PKCS12 con las llaves ya generadas.

TrustedX está siendo certificado como Qualified Signature Creation Device (QSCD) para ser utilizado por Proveedores de Servicios de Confianza (TSP). Para que desde la aplicación se pudiera realizar firma calificada deberían crear las claves dentro del propio TrustedX mediante un HSM. De todas formas, el proceso de firma es el mismo tanto para firma cualificada como para la avanzada, por lo que el resultado final es un buen ejemplo de lo que sería un proceso de firma cualificada.

3.8. DEFINICION DE REQUISITOS DEL SISTEMA

En este apartado se realiza una extracción de requisitos del sistema con el fin de presentar las principales funcionalidades deseadas para el proyecto, sirviendo de base a posteriores fases del ciclo de vida del proyecto. Los requisitos identificados proporcionarán al tribunal una visión general de la aplicación, de forma completa y sin ambigüedades.

3.8.1 Identificación de los Requisitos (ver anexo)

3..2 Formato FacturaE (ver anexo).

4.DISEÑO

4.1 DISEÑO DE LA ARQUITECTURA DE SOPORTE

El objetivo principal de este documento es presentar el diseño del sistema realizado de forma detallada. También se estudiará la tecnología que será de utilidad para llevar a cabo esta actividad.

Se realizará una especificación detallada de los componentes en los que se dividirá el sistema, con el fin de cubrir todas las decisiones de diseño correspondientes a la fase de construcción. De esta manera, sólo quedarán posibles decisiones a tomar relacionadas con el lenguaje de programación seleccionado para la codificación del sistema.

El Documento de Diseño del Sistema es de vital importancia en el desarrollo de un proyecto software, ya que marca las pautas para las posteriores fases de construcción y de implantación final del sistema, siendo un documento básico para los programadores encargados de la implementación.

4.1.1 ALCANCE

El presente documento presenta una especificación detallada de los componentes en los que se dividirá el sistema, permitiendo cubrir todas las decisiones de diseño que se verán reflejadas en la fase construcción.

Por otro lado, se especificará el entorno tecnológico necesario para que el sistema pueda entrar en ejecución. Contendrá además la planificación de capacidades, los requisitos de administración, el control de accesos, la seguridad y la operación.

4.1.2.DISEÑO DE LA ARQUITECTURA DE SOPORTE

En este apartado se presentan los elementos más significativos del sistema: los componentes de la arquitectura.

El sistema sigue una arquitectura cliente-servidor en 3 niveles (o 3 capas).

Componentes de esta arquitectura:

- Servidor: ROL que desempeña un equipo ofreciendo un conjunto de servicios a los clientes, tales como manejo de archivos, páginas web, control de acceso….

- Cliente: ROL que desempeña un equipo demandando servicios de los servidores, pero también puede hacer ciertas tareas de procesamiento local, como desplegar páginas Web, mostrar ventanas….

Los tres niveles corresponderían a:

  - Presentación (capa 1): Software que permite presentar de forma adecuada los resultados de una aplicación mediante las páginas Web. Lo que en el caso de nuestra aplicación serán las páginas HTML y jsp.
  - Aplicación o lógica de negocio (capa 2): Software que entrega un resultado útil para el usuario en nuestro caso los servlets y las clases .java.
  - Administración de datos (capa3): El manejo del os datos en una Base de Datos, que sirven a las aplicaciones de la lógica de negocio. En el caso de nuestro TFG esta capa se desarrolla en un servidor de BBDD.

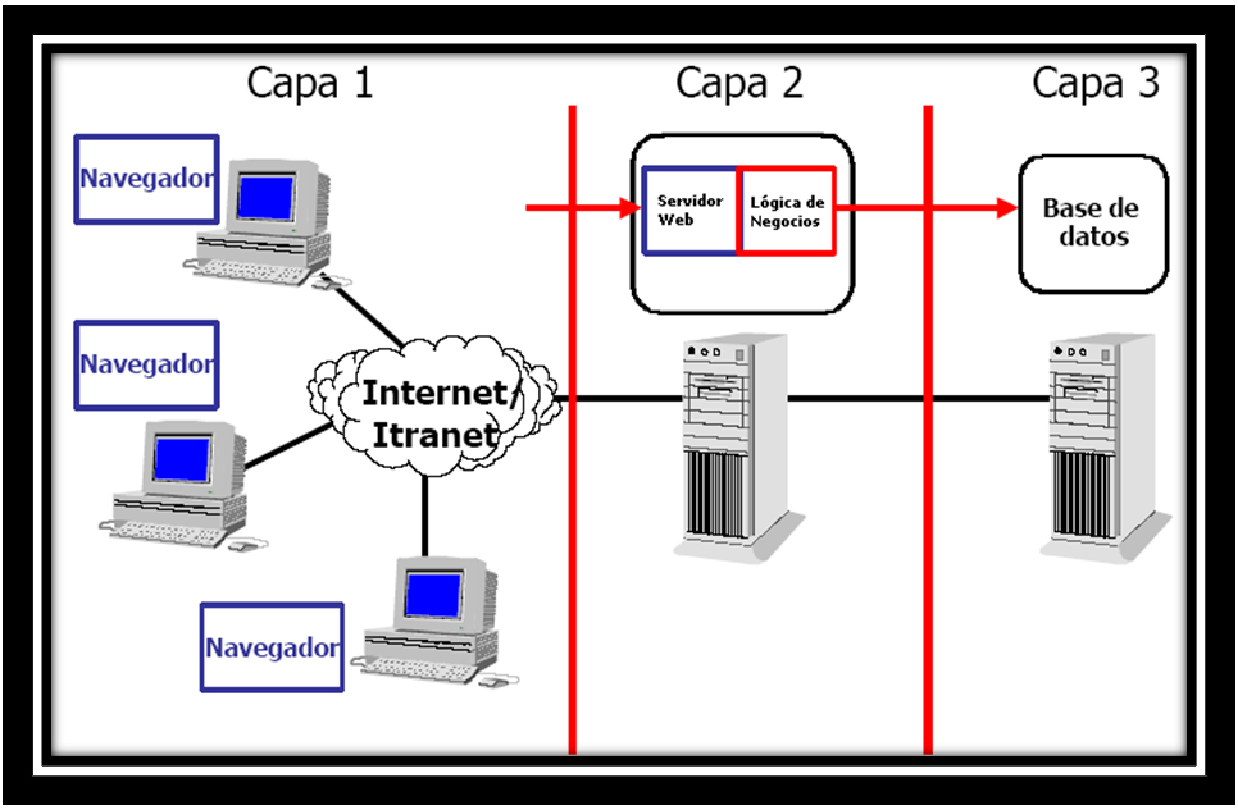


Diagrama ARQUITECTURA CLIENTE SERVIDOR 3 capas

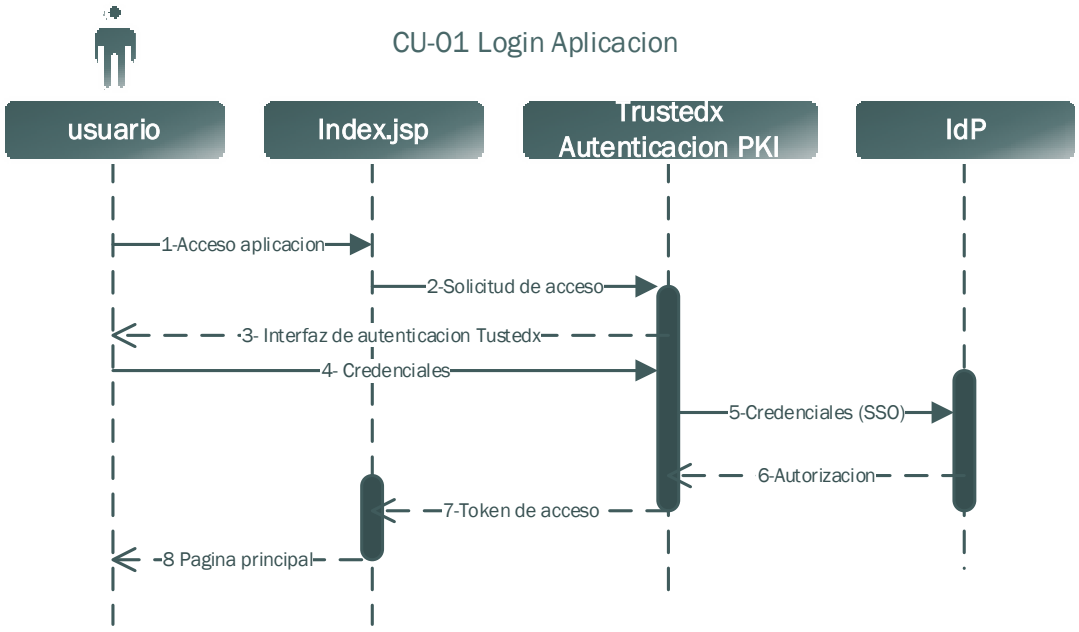
4.2. DIAGRAMAS DE SECUENCIA

El diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML.

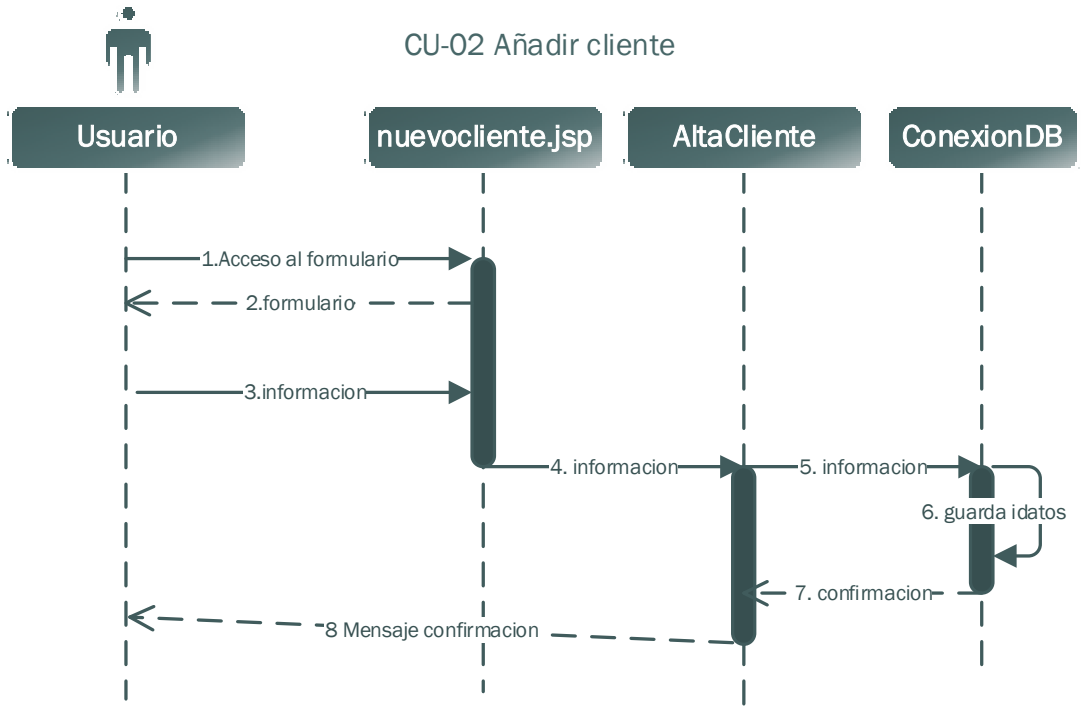
Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. El diagrama de secuencia contiene detalles de implementación, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede atender a esos pasos para descubrir qué objetos son necesarios para seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

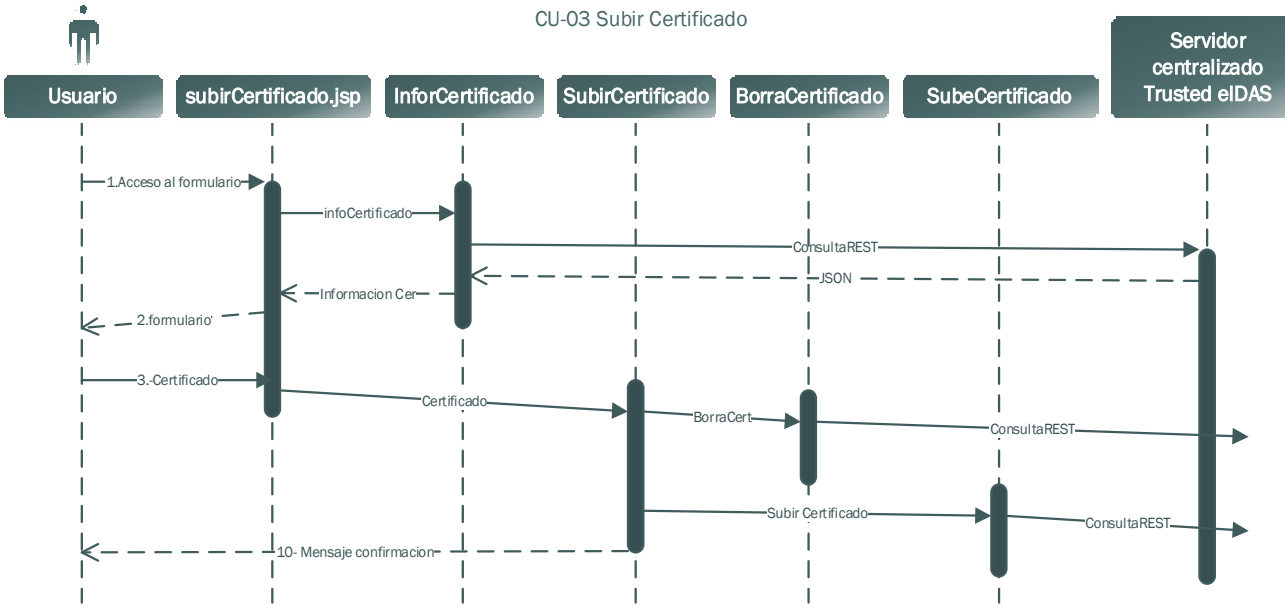
A continuación, se muestran los diagramas de secuencia para los casos de uso vistos anteriormente.



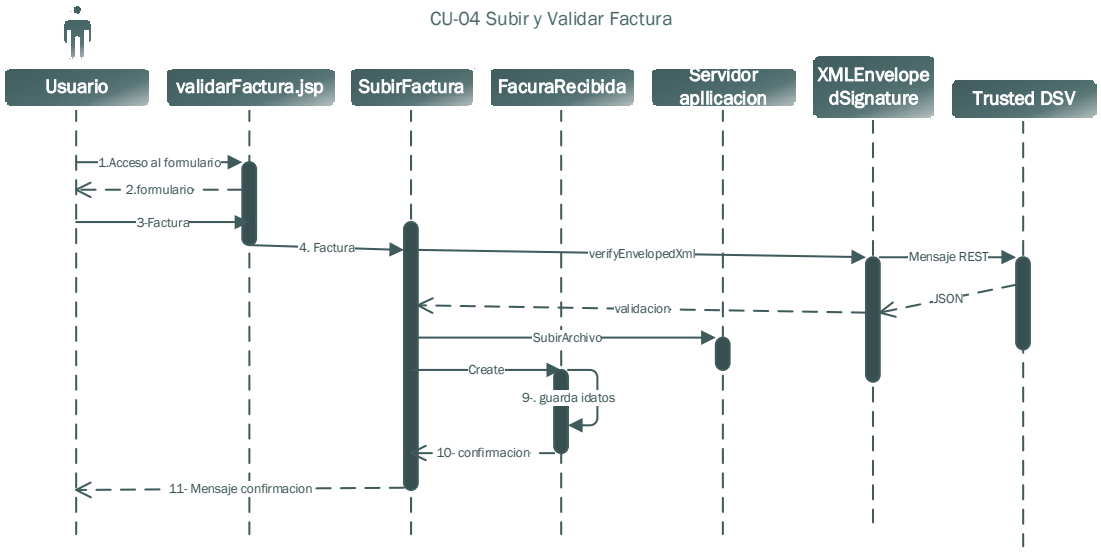
CU-01 LoginAplicacion



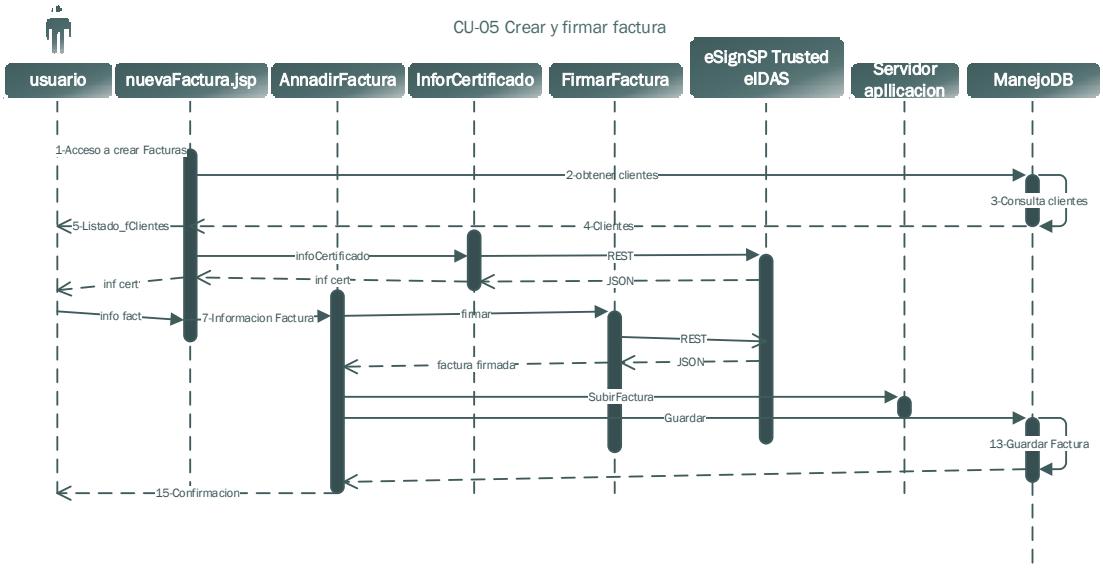
CU-02-AñadirCliente



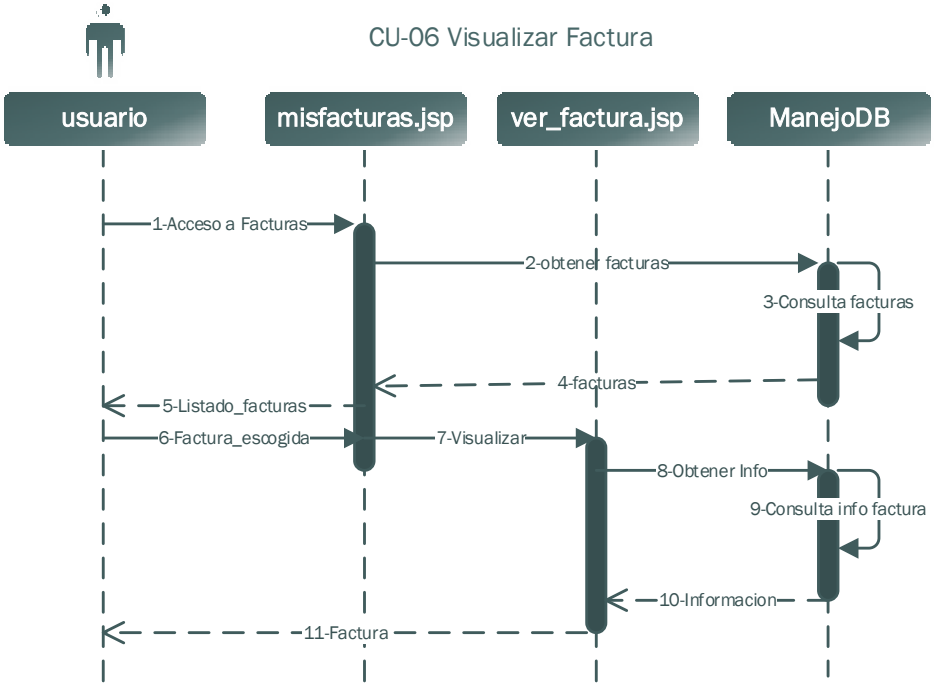
CU-03-Subir Certificado



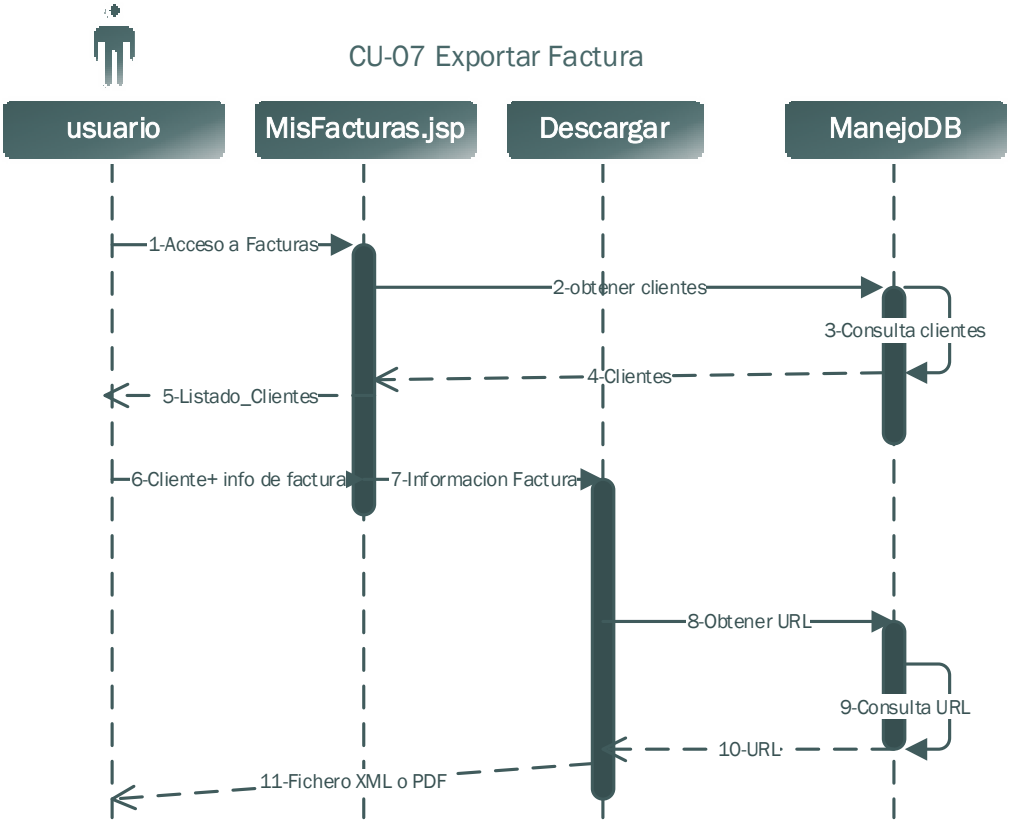
CU-04 Subir y validar Factura



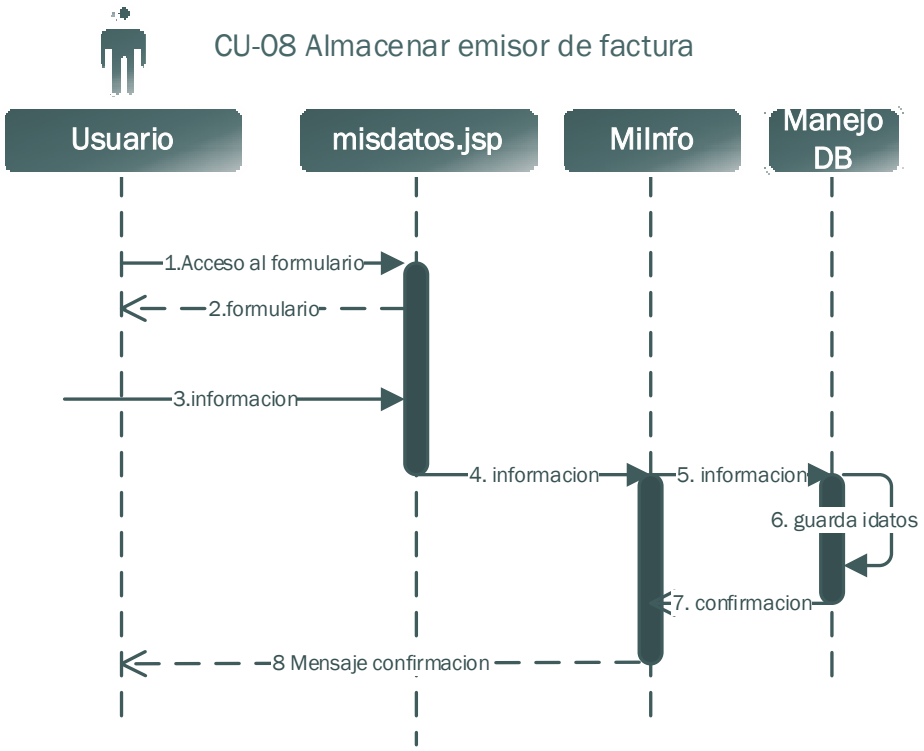
Cu-05 Crear y Firmar Factura



CU-06 Visualizar Factura



CU-07 Exportar Factura



CU-08 Almacenar emisor Factura

4.3 DISEÑO DE CLASES

En esta sección se desarrolla el modelo de clases del sistema que se está diseñando.

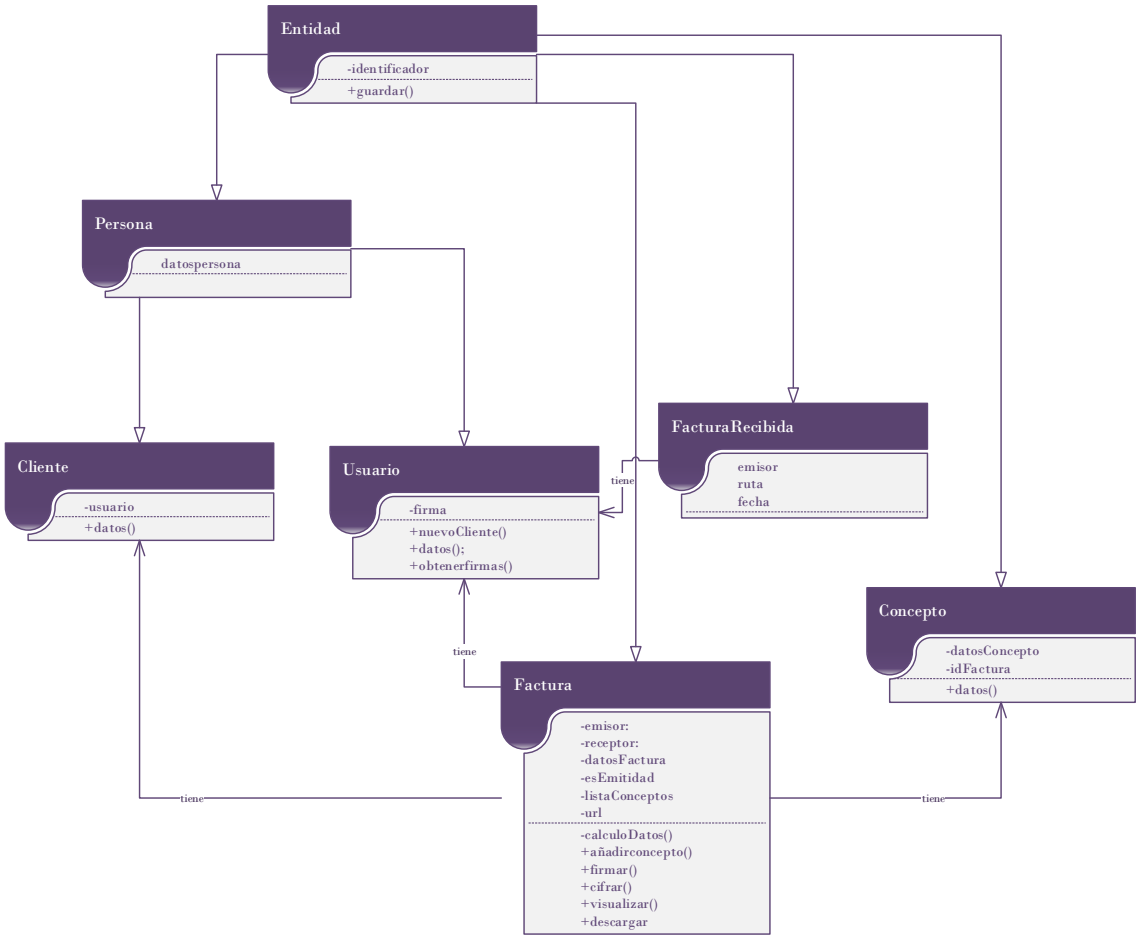
Los modelos aquí presentados servirán de guía al programador, por lo que los identificadores, relaciones, atributos y funciones son orientativos, dejando al programador libertad para programar bajo su propio criterio.

4.3.1 DIAGRAMA DE CLASES

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientadas a objetos.

El diagrama de clases incluye mucha más información como la relación entre un objeto y otro, la herencia de propiedades de otro objeto, conjuntos de operaciones/propiedades que son implementadas para una interfaz gráfica.

Presenta las clases del sistema con sus relaciones estructurales y de herencia.



4.3.2. CLASES, ATRIBUTOS Y METODOS ESTRUCTURALES (Ver anexo)

4.3.3 DESCRIPCION DE CLASES DE CAPA INTERMEDIA (SERVLETS) (ver Anexo)

4.3.4. DESCRIPCION DE CLASES DE COMUNICACIONES CON LA PLATAFORMA TRUSTEDX (ver Anexo)

4.3.5 Descripción de la Clases de la capa Cliente (jsp) (ver Anexo)

4.4 DISEÑO FISICO DE DATOS

4.4.1. DIAGRAMA ENTIDAD- RELACIÓN (E-R)

El modelo Entidad-Relación (E-R) representa un determinado dominio utilizando entidades y relaciones entre ellas.

Entidades:

- Cada entidad se caracteriza por el valor de sus atributos.
- Un atributo (clave) identifica unívocamente a cada entidad.

Relaciones:

- Una relación describe una asociación entre varias entidades.
- Esta asociación determina el número de entidades (cardinalidad) relacionadas en la misma.

A continuación, se muestra el modelo E/R de la base de datos:

DIAGRAMA ENTIDAD RELACION

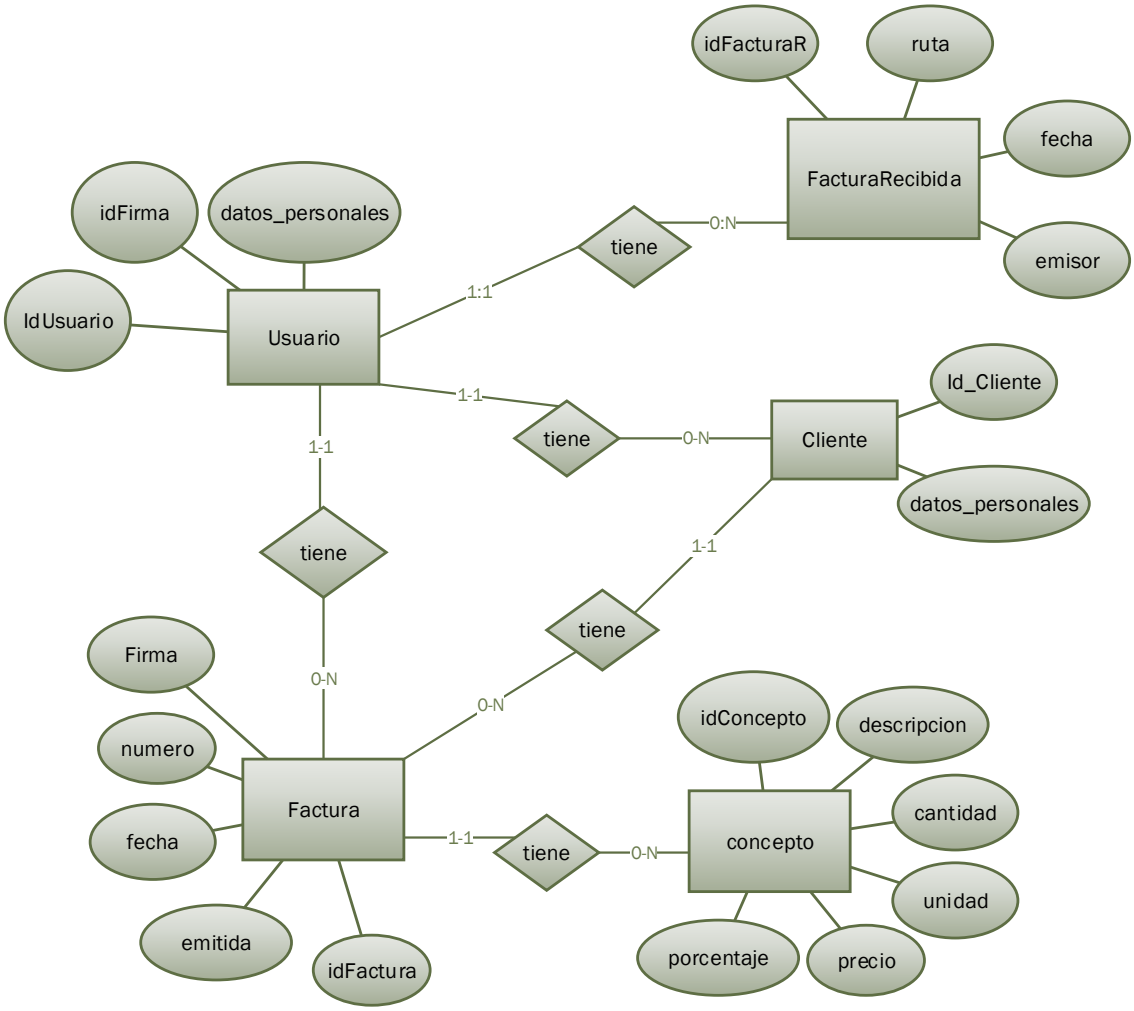


Diagrama Entidad Relación

Detalles del diagrama:

La entidad Usuario representa a los usuarios de la aplicación, y almacena la información asociada a estos. Tiene los siguientes datos

- Un identificador único.
- Un certificado, que indica si tiene un certificado en la plataforma asociado o no.
- Datos personales que son un conjunto de datos que figurarán en las facturas que el usuario emita, son estos:
    - Tipo Persona (persona física o persona jurídica)
    - Tipo de residente (español, extranjero, o residente en la unión europea)
    - Identificación Fiscal (DNI, CIF, NIF…)
    - Razón social

    - Nombre comercial
    - Nombre
    - Primer apellido
    - Segundo apellido
    - Dirección
    - Población
    - Provincia
    - País

La entidad Cliente representa a los receptores de las facturas, en ella se almacena la información asociada a estos que se indican en las facturas. En ella se almacenan los siguientes datos:

- Un identificador único de cliente
- Datos personales que son un conjunto de datos que figurarán en las facturas que el usuario emita a dicho cliente, son estos:
    - TipoPersona (persona física o persona jurídica)
    - Tipo de residente (español, extranjero, o residente en la unión europea)
    - Identificación Fiscal (DNI, CIF, NIF…)
    - Razón social
    - Nombre comercial
    - Nombre
    - Primer apellido
    - Segundo apellido
    - Dirección
    - Población
    - Provincia
    - País

La entidad usuario y la entidad Cliente se relacionan entre sí, de modo que un usuario tiene 0 y varios clientes y un cliente solamente pertenece a un usuario. Por lo que la relación será de 1:N.

La entidad que se ha nombrado como factura representa a las facturas creadas por la aplicación.

- Un identificador único de la factura.
- Fecha: la fecha de la factura
- El número de la factura
- Si está firmada
- La url donde se ha guardado

Entre la entidad Factura y la entidad usuario existe una relación en la que cada factura tiene solo un usuario (que se refiere al emisor de la factura) y Cada usuario puede tener varias facturas. Por lo tanto, la relación existente es de tipo 1:N.

Entre la entidad Factura y la entidad cliente también hay una relación en la que cada factura tiene un cliente (que es el receptor de la factura) y cada cliente puede aparecer en varias facturas. Por lo tanto, la relación existente es de tipo 1:N.

Entidad concepto es la que representara los conceptos que existen en cada factura su valor y tributación. Y tiene los siguientes datos:

- Un identificador de Concepto
- La descripción del concepto
- La cantidad
- El tipo de unidades (kg, hora....)
- El precio por unidad
- Y el porcentaje de impuesto que se aplica

Entre la entidad Factura y la entidad línea de detalle también hay una relación en la que cada línea de detalle tiene una factura y cada factura puede tener en varias líneas de detalle. Por lo tanto, la relación existente es de tipo 1:N.

La entidad FacturaRecibida corresponde a las facturas que hemos validado con nuestra aplicación y hemos subido al servidor. Se compone de los siguientes atributos:

- El identificador de la factura recibida
- La fecha de la factura
- La ruta donde esta almacenada la factura en nuestro servidor
- El nombre del emisor de la factura.

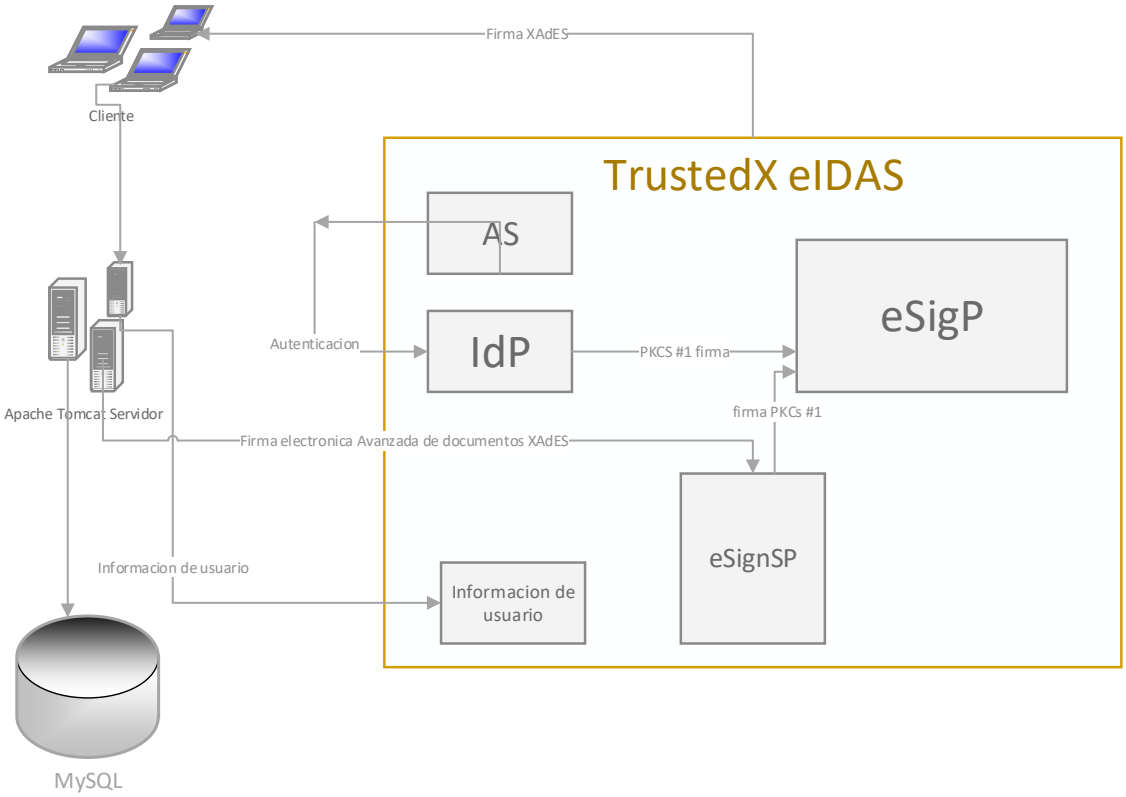
Entre la entidad FacturaRecibida y la entidad usuario existe una relación en la que cada factura tiene solo un usuario (que se refiere al receptor de la factura) y Cada usuario puede tener varias factures que hubiera recibido y validado. Por lo tanto, la relación existente es de tipo 1:N.

4.4.4 DICCIONARIO DE DATOS (VER ANEXO)

4.5. INTERFAZ DE USUARIO. (VER ANEXO)

4.6. ESQUEMA GENERAL DE LA APLICACION

En la siguiente figura se muestran los componentes necesarios para que los usuarios se autentiquen y firmen documentos.



5.IMPLEMENTACION DEL PROYECTO

5.1. INTRODUCCION

En este capítulo se pondrá en detalle el proceso de implementación del proyecto. Empezando por los entornos de desarrollo donde se ha llevado a cabo, las principales tecnologías que han tomado partido, tanto en el cliente como en el servidor y se mostrarán los diagramas UML más relevantes para comprender mejor la arquitectura interna del sistema.

5.2. IMPLEMENTACION DE LA AUTENTICACION OAUTH 2.0 CON TRUSTEDX

Para realizar la autenticación OAuth 2.0 hay que bajarse un formulario web del servidor que permita al usuario rellenar con sus datos (usuario y contraseña) y mandarlo al servicio correspondiente.

5.2.1 Aplicación de Flujo OAuth 2.0.

Para que la aplicación pueda realizar tareas en *TrustedX* en nombre del usuario es necesario el *token* OAuth 2.0. En esta sección se detalla la implementación del flujo.

Se trata de una autenticación basada en el flujo *Authorization Code*, como la explicada en la sección 2 del capítulo CONCEPTOS BÁSICOS, con una autorización implícita. Esto es, no se le pide explícitamente al usuario si autoriza que el servidor intermedio utilice *TrustedX* para firmar en su nombre.

El navegador abre una URL que muestra el formulario de autenticación (usuario y contraseña) de TrustedX. Especificando también la URL del servicio que va a actuar en su nombre, en este caso, el *Signature Server*, así como el *client id* para identificar qué aplicación es la que quiere acceder.

1. El usuario introduce su nombre de usuario y contraseña desde su navegador.
2. El navegador manda los campos de autenticación a *TrustedX* y éste los valida.

3. *TrustedX* manda el *code* al navegador junto con una orden de redirección a la URL del *Signature Server* especificada.
4. Cuando el *Signature Server* recibe el *code* lo manda de vuelta a *TrustedX* mediante una solicitud autenticada de token. *TrustedX* valida la identidad del *Signature Server* a partir de su *client id* y *client secret* y comprueba que el *code* es el mismo que ha mandado él y que viene de la misma URL que se ha especificado en el punto 1.
5. Después la aplicación comprueba si el usuario existe en nuestra BBDD y en caso negativo se crea uno nuevo con el client id recibido.

5.3. IMPLEMENTACION DE LA SUBIDA DE CERTIFICADOS TRUSTEDX

Tras que el usuario esta autenticado en el sistema, tiene un token de sesión, este puede subir un certificado propio con el que más adelante puede firmar las facturas que cree en la aplicación. Para ello utilizara el servicio eSigP de TrustedX, ese servicio se encarga de custodiar las identidades de firma del usuario.

La aplicación solamente permite que cada usuario tenga una identidad en el servidor, en el caso de querer subir una nueva identidad teniendo ya alojada otra anteriormente, la que estaba primero se eliminara antes de subir la segunda.

Las operaciones a realizar son:

Partiendo con el requisito que tenemos un token de autenticación OAuth2 valido. La aplicación enviara la siguiente información mediante REST.

```
POST /TrustedX-resources/esigp/v1/sign_identities/server/pki_x509/pkcs12 HTTP/1.1
Host: uoc.safelayer.com:8082
Authorization: Bearer TOKEN
Content-Type: application/json
cache-control: no-cache
{
   "labels" : ["uoc", "student"],
 "pkcs12" : "Certificado de identidad codificado en Base64”  "password" : "contraseña del certificado"
}
```

5.4. IMPLEMENTACION DE LA FIRMA DE FACTURAS TRUSTEDX

En la aplicación web se firmarán los documentos XML que son facturas con formato FacturaE, con una firma XAdES. Se utilizará el servicio eSignSP.

Este servicio hace uso del servicio eSigP para la firma electrónica de la factura.

Se generará una firma con el formato XAdES.

A.- La aplicación cliente deberá primero de todo obtener una autorización conforme puede generar procesos de firma contra eSignSP. Este servicio utiliza Client Credentials Grant del protocol OAuth2 y por lo tanto podrá solicitar un token administrativamente utilizando las credenciales de la aplicación:

Esta es la información:

```
POST /TrustedX-authserver/oauth/main/token HTTP/1.1
Host: uoc.safelayer.com:8082
Authorization: Basic ZmFjdHVyYWU6ZmFjdHVyYWU=
cache-control: no-cache

grant_type=client_credentials&scope=urn%3Asafelayer%3Aeidas%3Asign%3A
process%3Adocument
```

B.- Esto nos devolverá un “access_token” de respuesta que nos permite generar proceso de firma de documentos. Con el uso este token obtenido, podemos lanzar una petición de forma administrativa para enviar el documento XML a firmar y sus características de firma.

```
POST /TrustedX-resources/esignsp/v2/signer_processes HTTP/1.1
Host: uoc.safelayer.com:8082
Authorization: Bearer TOKEN
cache-control: no-cache
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="process"
{
 "process_type": "urn:safelayer:eidas:processes:document:sign",
  "signer" : {
     "signature_policy_id" : "urn:safelayer:eidas:policies:sign:document:xml",
     "parameters": {
       "type" : "xades-epes",
       "signature_target" : {
          "signature_packaging" : "enveloped"
       },
       "policy_identifier" : {
          "policy_id" : {
             "identifier" : {
                "uri": "urn:oid:2.16.724.1.3.1.1.2.1.9",
                "qualifier" : "OIDAsURN"
             },
             "description" : "Política de firma electrónica para la Administración
General del Estado"
          },
          "policy_hash" : {
             "digest_algorithm_identifier" : {"id" : "sha1"},
             "digest_value" : "G7roucf600+f03r/o0bAOQ6WAs0="
          },
          "policy_qualifiers" : [
             {
                "type" : "spuri",
                "uri" : "https://sede.060.gob.es/politica_de_firma_anexo_1.pdf"

             }
          ]
       }
     }
   },

  "finish_callback_url": "https://pagina_de_respuesta

}
```

C.- Una vez enviamos la petición obtendremos una respuesta JSON satisfactoria de la que almacenaremos la información:

Tasks.pending.url: URL a la que la aplicación cliente deberá redirigir el navegador del usuario para que complete el proceso de firma XAdES-EPES.

Documents.id: identificador de documento para el proceso de firma creado para poder recuperar el documento firmado.

Seguidamente la aplicación redirigirá el navegador a la url que hemos obtenido donde se nos solicitará permiso para obtener información del usuario y las identidades que tiene disponibles para poder continuar con el proceso de firma.

Una vez autorizado el acceso a esta información se nos presentará una pantalla donde permitirá al usuario pre visualizar el documento que va a firmar.

D.- Una vez continuemos el proceso de firma y pasemos por varias pantallas que nos indica que está procesando la firma, nos solicitará de nuevo autorización a utilizar la identidad de firma.

Una vez autorizado, el proceso finalizará y redirigirá a la url que se indicó en "finish_callback_url"

E.- Una vez finalizado el proceso de firma, a través del identificador de documento que se almacenó se podrá acceder al contenido del documento firmado.

Para ello, sustituir el documents.id que se almacenó en la url siguiente:

```
GET /TrustedX-resources/esignsp/v2/documents/document.id/content HTTP/1.1
Host: uoc.safelayer.com:8082
Authorization: Bearer TOKEN
cache-control: no-cache
```

F.- Tras finalizar un proceso de firma y recuperar su documento, la aplicación cliente deberá eliminar el proceso de firma para liberar recursos en el servidor.

5.5. ENTORNOS DE DESARROLLO

Para la implementación del servidor y de la aplicación Web se ha utilizado NetBeans IDE 7.4 Java: 1.8.0_151 en una maquina Windows 10 versión 10.0 running on amd64. Y un servidor de MySQL 5.5.24 he utilizado Wampserver 2.2 El servidor de aplicaciones Web que he utilizado es Apache tomcat 7.0.93

5.6. TECNOLOGIAS UTILIZADAS

Para implementar la capa multiplataforma de los clientes ha sido necesario utilizar las tecnologías web que se describen en este apartado.

5.6.1. HTML

HTML es el lenguaje principal con el que se construyen las páginas web. Las siglas significan *HyperText Markup Language* o Lenguaje de Marcado de HiperTexto.

El HTML se escribe en forma de etiquetas y puede describir la estructura principal de un documento. También puede incluir *scripts* que afecten el comportamiento de los navegadores web u otros intérpretes.

5.6.2. CSS

El *Cascading Style Sheet* (CSS), o Hoja de Estilo en Cascada, es un lenguaje de hoja de estilo usado para definir las semánticas de presentación (es aspecto y formateo) de un documento escrito en un lenguaje de marcado, como HTML.

5.6.3. JavaScript

JavaScript es un lenguaje de programación interpretado con propiedades de la programación orientada a objetos (POO). La sintaxis es parecida a la de C++, C y Java, sin embargo, esta similitud termina en el ámbito sintáctico.

No existe tipado en JavaScript, es decir, las variables no necesitan ser especificadas de un tipo u otro. Es el propio intérprete quien se encarga, en tiempo de ejecución, de asignar tipos a las variables.

JavaScript se usa comúnmente en navegadores web y, en ese contexto, el núcleo de propósito general se extiende con objetos que permiten a los *scripts* interactuar con el usuario, controlar el navegador web y alterar el contenido del documento web. Ésta versión incrustada de JavaScript ejecuta *scripts* incrustados dentro del código HTML de las páginas web. Comúnmente se denomina *client-side* JavaScript para resaltar que los *scripts* son ejecutados en la máquina cliente en lugar de en el servidor web.

Estas librerías también ofrecen métodos para realizar llamadas AJAX al servidor, pudiendo así mantener una comunicación asíncrona sin interferir con la visualización o el comportamiento de la aplicación. Todas las llamadas al servidor, por ejemplo, obtener los documentos, se hacen mediante AJAX.

5.6.4 JSON

JSON es el acrónimo para *JavaScript Object Notation*, y aunque su nombre lo diga, no es necesariamente parte de JavaScript, de hecho es un estándar basado en texto plano para el intercambio de información, por lo que se usa en muchos sistemas que requieren mostrar o enviar información para ser interpretada por otros sistemas, la ventaja de JSON al ser un formato que es independiente de cualquier lenguaje de programación, es que los servicios que comparten información por éste método, no necesitan hablar el mismo idioma, es decir, el emisor puede ser Java y el receptor PHP, cada lenguaje tiene su propia librería para codificar y decodificar cadenas de JSON.

5.6.5 SOAP

SOAP (Simple Object Access Protocol), es un protocolo que nos permitirá realizar servicios web sin estado, a través de TCP y con un formato XML.

Entre sus ventajas podemos encontrar que al funcionar a través del protocolo de transporte TCP, se pueden utilizar diferentes protocolos de aplicación como: HTTP, SMTP o JMS. También nos brinda la posibilidad de generar cliente/servidor en distintos lenguajes de programación. Y está ampliamente estandarizado, por lo cual hay reglas concretas para formar el mensaje, el contrato entre cliente/servidor o el formato de los datos a enviar, siempre XML.

5.6.6 REST

REST es el acrónimo de REpresentational State Transfer. A diferencia de SOAP, más que un protocolo es una definición de arquitectura sé dónde nos indica cómo realizar el intercambio y manejo de datos a través de servicios web. A aquellos servicios web que siguen su definición se les conocen como RESTful Web services.

Las APIs REST se distinguen por que se basan fuertemente en el protocolo de aplicación HTTP. Es decir, usan los métodos y códigos de respuesta HTTP para una

función específica y ampliamente reconocida por todos. Y nos permite a través de la URI, la estructuración de los recursos disponibles.

Entre sus ventajas se encuentran la posibilidad de crear cliente/servidor en distintos lenguajes. Nos da la posibilidad de enviar la información en distintos formatos, aunque habitualmente se usa JSON.

5.6.7 JAVA

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

Es un lenguaje de alto nivel, de propósito general, concurrente, basado en clases y orientado a objetos que esta específicamente diseñado para tener el menor número de dependencias de implementación posible. Se pretende que los desarrolladores escriban el código una vez y lo ejecuten en cualquier sitio, por lo que el código que se ejecuta en una plataforma, no necesita ser recompilado para ejecutarse en otra.

Las aplicaciones Java son compiladas en *bytecode*, que puede ser ejecutado únicamente en una máquina virtual de Java (JVM) independientemente de la arquitectura del sistema.

La sintaxis del lenguaje deriva mucho de C y C++, pero tiene muchas menos funcionalidades de bajo nivel. Por ejemplo, no se permite el acceso directo a memoria. Esto permite disminuir el riesgo de errores por parte del desarrollador.

El propio lenguaje se encarga de la gestión de memoria del ciclo de vida de los objetos. El programador determina cuando un objeto es creado y el motor de Java se encarga de liberar la memoria una vez estos ya no están en uso.

5.6.7.1 Java EE

Java Enterprise Edition, traducido informalmente como Java Empresarial, es una plataforma de programación que forma parte de la plataforma Java. La plataforma ofrece una API y un entorno de ejecución para desarrollar y ejecutar software empresarial, incluyendo servicios web y otras aplicaciones en red a gran escala, multicapa, escalables, fiables y seguras. Java EE hereda de la plataforma Java, Stantard Edition, ofreciendo una API para mapeo de objetos relacionales, arquitecturas multicapa distribuidas y servicios web. La plataforma incorpora un diseño basado mayormente en componentes modulares que se ejecutan en un servidor de aplicaciones. El software desarrollado para Java EE esta mayormente desarrollado mediante el lenguaje de programación Java.

5.6.8.Apache Tomcat

Tomcat es un servidor web desarrollado también por la Apache Software Foundation, por lo que su nombre oficial es Apache Tomcat. También es un servidor HTTP, sin embargo, está hecho para aplicaciones Java en lugar de sitios web estáticos. Tomcat puede ejecutar varias especificaciones diferentes de Java, como Java Servlet, JavaServer Pages (JSP), Java EL y WebSocket.

5.6.9 MySQL

Para implementar la persistencia de los datos de usuarios que el servidor tiene que almacenar se ha decidido utilizar la base de datos MySQL.

MySQL permite la creación de bases de datos relacionales gráficamente mediante su herramienta MySQL Workbech. El usuario puede diseñar las tablas y sus relaciones en un entorno gráfico y luego exportar un *script* que crea las instancias de la base de datos diseñada.

A su vez MySQL es una de las bases de datos más usadas entre los desarrolladores por su fiabilidad manejando grandes cantidades de datos y consultas.

5.7. LIBRERIAS

A continuación, se listan y describen las diferentes librerías que se han utilizado para el desarrollo del programa

- **Apache-loggin-logj4**: Log4j es una biblioteca open source desarrollada en Java por la Apache Software Foundation que permite a los desarrolladores de software escribir mensajes de registro, cuyo propósito es dejar constancia de una determinada transacción en tiempo de ejecución.
- **Axis**: Axis es una implementación OpenSource de SOAP que proporciona un entorno de ejecución para Servicios Web implementados en Java. A grandes rasgos, un Servicio Web es un conjunto de métodos que realizan una funcionalidad que se exponen al resto de las aplicaciones. Cualquier aplicación sea cual sea su plataforma o lenguaje en la que está implementada podrá invocar los métodos que expone el Servicio Web.
- **Commons Discovery 0.5**: Librería que localiza clases que implementan una interfaz Java determinada. El patrón de descubrimiento, aunque no necesariamente este paquete, se usa en muchos proyectos, incluidos JAXP.
- **Commons FileUpload 1.3:** El paquete Commons FileUpload facilita agregar capacidades de carga de archivos robustas y de alto rendimiento a sus servlets y aplicaciones web. FileUpload analiza las solicitudes HTTP que cumplen con RFC 1867, "Carga de archivos basada en formulario en HTML". Es decir, si se envía una solicitud HTTP utilizando el método POST y con un tipo de contenido de "multipart / form-data", entonces

FileUpload puede analizar esa solicitud y hacer que los resultados estén disponibles de una manera fácil para el llamante.

- **Javax.xml.rpc:** API javax.XML es una colección de APIs para el parseo y tratamiento de documentos XML que parte del JDK de Sun Microsystems. RPC: permite invocar un servicio web desde una aplicación Java. Puede considerarse como protocolo RMI sobre servicios web.
- **Json:** Los archivos en este paquete implementan codificadores / decodificadores JSON en Java. También incluye la capacidad de convertir entre JSON y XML, encabezados HTTP, cookies y CDL.
- **Kotlin-STDLib:** La biblioteca estándar de Kotlin proporciona elementos esenciales para el trabajo diario con Kotlin. Éstos incluyen: Funciones de orden superior que implementan patrones idiomáticos (dejar, aplicar, usar, sincronizar, etc.). Funciones de extensión que proporcionan operaciones de consulta para colecciones (eager) y secuencias (perezoso). Varias utilidades para trabajar con cadenas y secuencias de caracteres. Extensiones para las clases de JDK por lo que es conveniente trabajar con archivos, IO y subprocesos.
- **Okhttp:** Es una librería de código abierto la cual permite realizar operaciones tanto en HTTP como en SPDY de manera sencilla y eficiente en ambientes Java (versión 1.7 como mínimo) y Android (2.3 como mínimo), sin necesidad de cambiar el código de la aplicación entre ambas plataformas, con una interfaz fluida. Entre las virtudes que tiene esta librería es que automáticamente evalúa las características de la conexión y determina cómo operar de mejor manera, por ejemplo, si se tiene soporte para SPDY, contenido HTTP comprimido (GZIP), recuperación de algunos errores en la conexión.
- **Okio:** es una biblioteca que complementa java.io y java.nio para que sea mucho más fácil acceder, almacenar y procesar sus datos. Comenzó como un componente de OkHttp, el cliente HTTP capaz incluido en Android. Está bien ejercitado y listo para resolver nuevos problemas.
- **org.apache.commons.httpclient:** es una librería con un conjunto de herramientas de componentes Java de bajo nivel enfocados en HTTP y protocolos asociados.
- **Org.apache.commons.loggin:** Una biblioteca que utiliza la API de registro de bienes comunes puede usarse con cualquier implementación de registro en tiempo de ejecución. Commons-logging viene con soporte para varias implementaciones populares de registro, y escribir adaptadores para otros es una tarea bastante simple.
- **SmartWrapper:** es una API desarrollada por Safelayer sobre Axis para crear aplicaciones cliente que utilicen los servicios de TrustedX. SmartWrapper permite generar aplicaciones Java de manera más sencilla ya que evita la complejidad de programar utilizando directamente Axis. Aun así es el acceso a las estructuras Axis para crear llamadas avanzadas.
  Trustedx.Client.axis
- **Trustedx.Provider**
- **Wsdl4j** librería que permite trabajar con documentos WSDL
- **Xerces:** es una biblioteca para el análisis sintáctico, validación, serialización y manipulación de documentos XML de la Apache Software Foundation.

- **XML-apis-2.0** Es una librería que utiliza estructura de nodos para crear/leer un XML.

6. Conclusiones y Posibles Ampliaciones

6.1. CONCLUSIONES

Con la realización de este trabajo se pretendía alcanzar una serie de objetivos marcados al inicio del mismo, tarea que se ha conseguido satisfactoriamente.

En primer lugar, se ha creado un sistema que proporciona la funcionalidad deseada.

Se ha seguido la planificación establecida al principio del trabajo, salvo en ligeras modificaciones porque en algunas tareas se han invertido más tiempo que el planeado y en otras menos tiempo, cumpliendo con los plazos de entrega establecidos.

Se han puesto en práctica algunos de los conocimientos adquiridos a lo largo del master, sobre todo en cuanto al uso de facturas electrónicas, estándares de firmas electrónicas, certificados electrónicos, integridad, autenticidad.

Realizar este trabajo ha supuesto una adquisición de nuevos conceptos tecnológicos y metodologías. Ha sido necesario un amplio estudio de los conceptos de la PKI, OAuth2 y de las tecnologías de Safelayer. También se han adquirido conocimientos y experiencia en los lenguajes de programación, Java y de las tecnologías web JavaScript, HTML, CSS, JSON, REST y SOAP.

Se puede concluir finalmente que se ha logrado alcanzar el objetivo de desarrollar un proyecto software completo, adquiriendo nuevos conocimientos y comprobando la dificultad existente, en cuanto a esfuerzo dedicación y coste de desarrollar un proyecto de manera correcta.

Se ha desarrollado una aplicación web multidispositivo que permite la gestión y creación de facturas electrónicas en formato FacturaE. Mediante la aplicación se pueden crear, descargar y validar facturas en un formato legalmente correcto, con la misma validez que la factura tradicional en papel.

6.2. POSIBLES AMPLIACIONES.

Algunas de las posibles ampliaciones que se podrían realizar en el futuro se han ido comentando a lo largo del documento, pero se resumen en las siguientes:

- La aplicación las facturas que se pueden crear solamente es posible añadir el tipo de impuesto IVA, en una futura ampliación se podría permitir crear facturas con otros tipos de impuesto como IS, RLEA, IRPF, RA, etc.

- Las facturas electrónicas que se pueden crear son bastante simples una posible ampliación seria dar la posibilidad de hacer facturas más complejas en las que se puedan añadir a cada concepto otra información como descuentos, cargos, albaranes Referencias de contratos expedientes, números de albaranes, realizar, conceptos sin impuestos etc.
- Se podría realizar una ampliación permitiendo que desde la aplicación se enviaran facturas a la Administración general del estado, a través de FACe (Punto general de entrada de facturas electrónicas de la AGE).
- Ampliar la aplicación realizando facturas rectificativas.
- Se puede añadir a la aplicación un buscador de facturas en nuestra base de datos con los filtros como emisor de la factura, receptor, fecha importe, etc.
- Se podría dar la posibilidad de eliminar facturas de la aplicación.
- Otra posible ampliación sería un módulo de envío semiautomático mediante correo electrónico de las facturas.
- Se podría implementar una firma cualificada: Ya se ha comentado con anterioridad que realiza forma avanzada, debido a que las firmas se realizarán a partir de la importación de archivos PKCS12 con las llaves ya generadas. Sería interesante añadir un proceso para generar claves a TrustedX, lo que nos permitiría realizar firma cualificada.

6.3. OPINIÓN PERSONAL

Al iniciar este TFM tenía como objetivo profundizar en los conocimientos adquiridos en las asignaturas del Master, sobre todo facturas y firmas electrónicas.

Personalmente me ha gustado trabajar en este proyecto, por la experiencia de crear algo nuevo, original y ofrecer facilidades a futuros usuarios.

Pienso que he alcanzado mis objetivos personales en el desarrollo del TFM. Me supuso gran esfuerzo, ya que he tenido que adquirir ciertos conocimientos imprescindibles, a la vez que iba avanzando en el desarrollo, pero ha merecido la pena ya que he disfrutado durante este proceso de investigación y de aprendizaje.

7. BIBLIOGRAFÍA

- Programación en JAVA-J2EE: Antonio Martin Sierra, Ramón Egido García. 2ª Edición Grupo Syncrom
- Adecuación de los sistemas informáticos a la LOPD RD1720 y LSSI.
- Diseño y programación de Bases de Datos 2005: Pedro López-Belmonte Eraso. Alambra eidos
- Manual de Dirección de operaciones: Francisco Javier Miranda González, Sergio Rubio Lacoba, Antonio Chamorro Mera, Tomas Manuel Bañegil Palacios
- http://blog.fundaciobit.org/
- http://blog.isigma.es/
- http://forum.wampserver.com
- http://help.eclipse.org
- http://jafma.net
- http://openaccess.uoc.edu
- http://sedeaplicaciones2.minetur.gob.es
- http://tomcat.apache.org/
- http://www.allitebooks.org/
- http://www.automationexchange.com
- http://www.coaa.es
- http://www.cyberhades.com
- http://www.echoecho.com
- http://www.mysqltutorial.org/
- http://www.sinadura.net
- http://www.wampserver.com
- http://xerces.apache.org/
- https://administracionelectronica.gob.es
- https://auth0.com
- https://blog.signaturit.com
- https://boe.es
- https://chrome.google.com/
- https://cloud.google.com
- https://demo.safelayer.com
- https://developer.mozilla.org
- https://firmaelectronica.gob.es/
- https://httpd.apache.org/
- https://nationbuilder.com
- https://openclassrooms.com
- https://raygun.com
- https://searchoracle.techtarget.com
- https://sede.andujar.es/
- https://sede.mjusticia.gob.es
- https://sede.sepe.gob.es
- https://sede.uco.es
- https://smartbear.com
- https://sourceforge.net

- https://stackify.com/
- https://stackoverflow.com/
- https://support.eset.com/
- https://support.safelayer.com
- https://techedgespain.atlassian.net/
- https://thehackernews.com
- https://valide.redsara.es
- https://www.accv.es
- https://www.accv.es/
- https://www.adictosaltrabajo.com/
- https://www.baeldung.com
- https://www.baeldung.com
- https://www.bikewale.com
- https://www.blazemeter.com
- https://www.blog.andaluciaesdigital.es
- https://www.crunchbase.com
- https://www.cvedetails.com
- https://www.edureka.co
- https://www.FacturaE.gob.es
- https://www.getpostman.com/
- https://www.guru99.com
- https://www.ibm.com
- https://www.idia.es
- https://www.incibe.es/
- https://www.informatica.us.es
- https://www.ius.edu/
- https://www.javatpoint.com
- https://www.journaldev.com
- https://www.katalon.com
- https://www.learn-js.org/
- https://www.logicbig.com/
- https://www.lynda.com
- https://www.mysql.com/
- https://www.oasis-open.org
- https://www.percona.com/
- https://www.programmableweb.com
- https://www.quora.com
- https://www.redseguridad.com
- https://www.safelayer.com
- https://www.siteground.com
- https://www.theserverside.com
- https://www.tutorialspoint.com
- https://www.upwork.com
- https://www.uva.es
- https://www.vogella.com
- https://www.w3schools.com
- https://www.xolido.com