Universidad de las Ciencias Informáticas.

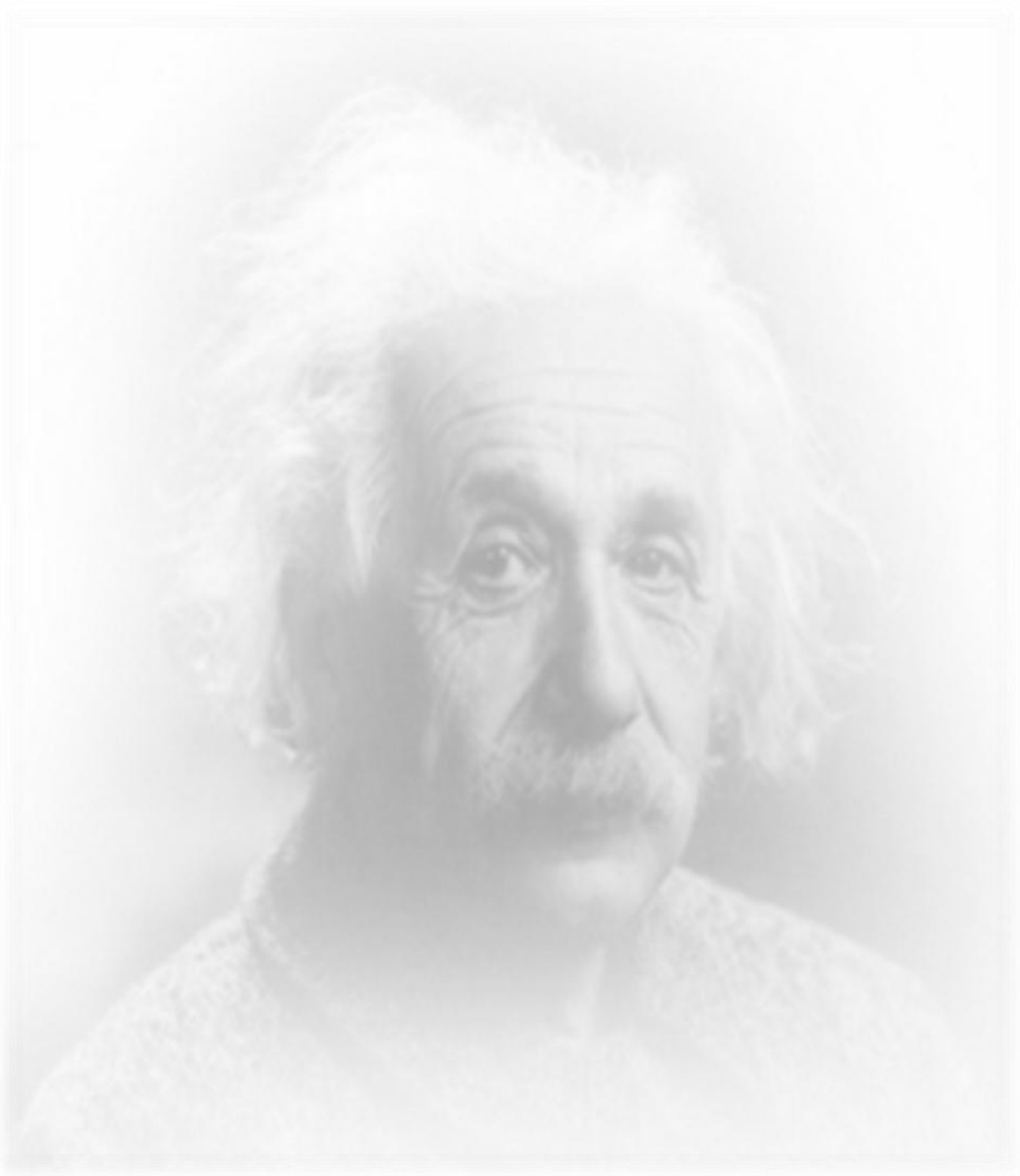
Facultad 3.

Título: Componente de software para la firma digital de documentos jurídicos tratados en formato electrónico en el proyecto Tribunales.

Trabajo de Diploma para optar por el título de
Ingeniero Informático.

Autor: Gianny Miranda Ramos.
Tutor: Ing. Maikel Navarro Díaz.
Cotutor: Ing. Dusniel Horta Centeno.
Asesor: Ing. Daimi Lamorú Marciel.

"Ciudad de La Habana. Junio, 2010"



"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber".

Albert Einstein.

DECLARACIÓN DE AUTORÍA.

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2010.

Gianny Miranda Ramos.

_____

Ing. Maikel Navarro Díaz.

_____

Ing. Dusniel Horta Centeno.

_____

Ing. Daimi Lamorú Marciel.

_____

Datos de Contacto.

Nombre y Apellidos: Maikel Navarro Díaz.
Edad: 26 años.
Ciudadanía: Cubano.
Institución: Universidad de las Ciencias Informáticas (UCI).
Título: Ingeniero Informático.
Categoría Docente: Profesor Asistente.
E-mail: mndiaz@uci.cu

Nombre y Apellidos: Dusniel Horta Centeno.
Edad: 26 años.
Ciudadanía: Cubano.
Institución: Universidad de las Ciencias Informáticas (UCI).
Título: Ingeniero Informático.
Categoría Docente: Profesor Asistente.
E-mail: dhorta@uci.cu

Nombre y Apellidos: Daimí Lamorú Marciel.
Edad: 26 años.
Ciudadanía: Cubano.
Institución: Universidad de las Ciencias Informáticas (UCI).
Título: Ingeniero Informático.
Categoría Docente: Profesor Asistente.
E-mail: dlamorú@uci.cu

AGRADECIMIENTOS.

Debo agradecer la terminación exitosa de este Trabajo de Diploma:

A Pepa, ya que si ti todo esto no hubiese sido posible.

A Mi familia, por apoyarme en todo momento y confiar en mis posibilidades.

A Nemo, mi mascota galáctica, por impedir que me molestaran cuando estaba trabajando.

A Carlos, tu ayuda fue fundamental y llegó en un momento clave para terminar la implementación.

A Osmel, por ayudarme desinteresadamente.

A Daimí, por incluirme en el proyecto y darme el tema de tesis.

A Lázaro, Migue, Juanito y todo aquellos que me ayudaron desinteresadamente.

A la Revolución, por permitirme estudiar y contribuir a mi superación profesional.

DEDICATORIA.

Este trabajo diploma está dedicado específicamente a todas las personas que siempre confiaron en mis capacidades y dedicación al estudio, que me proporcionaron todos los medios necesarios para superarme profesionalmente y que hoy en día disfrutan de los logros de tanto esfuerzo y sacrificio.

Esta tesis está especialmente dedicada:

A PEPA.

RESUMEN.

En el presente trabajo se hace un estudio de la **firma digital** como mecanismo de seguridad y se desarrolla un **componente web** para la **firma digital** de documentos en formato **PDF**, que garantice la **autenticidad**, **integridad** y **no repudio** de la información almacenada en dichos documentos. Por lo que se realiza un estudio del arte, en el cual se abordan conceptos claves como **criptografía**, **firma digital** y **certificados digitales y** se describen las herramientas, tecnologías utilizadas y artefactos generados en el proceso de desarrollo. Para el desarrollo de la solución propuesta se tuvieron en cuenta los requerimientos del cliente, que finalmente se probaron mediante las pruebas de aceptación realizadas al término de cada iteración planificada. Por último, se hace referencia a las recomendaciones y seguidamente se presentan las conclusiones finales, así los materiales anexos y las citas bibliográficas empleadas.

Palabras Claves: firma digital, PDF, autenticidad, integridad, no repudio, criptografía, certificados digitales.

Índice de Contenidos.

ÍNDICE DE TABLAS.

ÍNDICE DE FIGURAS.

INTRODUCCIÓN.

Con el desarrollo alcanzado por las redes telemáticas e Internet es cada vez más común el intercambio de información de cualquier tipo, ya sea de contenido financiero, administrativo o judicial entre personas distantes geográficamente. Sin embargo, este marco de relaciones está expuesto a todo tipo de amenazas, lo cual constituye un grave problema en la seguridad de los datos.

Esto ha permitido que muchos estados se acerquen a los ciudadanos transformando sus relaciones, prestando mejores servicios, garantizando transparencia y celeridad en la gestión pública, mejorando la eficiencia y eficacia de los gobiernos.

La seguridad en las transacciones electrónicas de la administración pública es un aspecto de vital importancia, que permite a los ciudadanos confiar en el intercambio de información con el gobierno. El empleo de la firma digital ha sido reconocido en muchos países, por su eficacia jurídica y valor probatorio.

El fraude, el robo de identidad y falta de confianza entre las partes que intervienen en la comunicación son problemas comunes en el mundo electrónico. La firma digital, como aplicación de la criptografía, provee los mecanismos para evitar dichos problemas. Al igual que la firma manuscrita al papel, la firma digital, otorga a los documentos electrónicos las propiedades de autenticación, no repudio e integridad.

La firma digital es una secuencia de datos electrónicos que se obtienen de la aplicación a un mensaje determinado, un algoritmo de cifrado asimétrico o de clave pública. Esta equivale funcionalmente a la firma autógrafa en orden a la identificación del autor del que procede el mensaje. Desde un punto de vista material, la firma digital es una simple cadena o secuencia de caracteres que se adjunta al final del cuerpo del mensaje firmado digitalmente.

El uso de esta técnica ha crecido a nivel mundial. En sus inicios fue asumida por empresas norteamericanas como Dell, Intel, Hewlett-Packard y AT&T (1) en sus proyectos pilotos y hoy la firma digital es parte de la vida diaria de muchos habitantes del planeta. Gobiernos como el de Estados Unidos y los de los países de la Unión Europea fueron los primeros en aprobar el valor legal de la misma, luego otros países se fueron sumando a la lista, incluso muchos países de Latinoamérica. Empresas como Gemalto y Verisign proveen productos y soluciones de alto nivel para sectores públicos, empresas e Internet, basados en esta y otras tecnologías.

Muy ligado al empleo de la firma digital, están los documentos electrónicos, por ser estos los que guardan la información a proteger. Dichos documentos almacenan cualquier tipo de datos: texto, imagen, audio y video son los más usuales. Dentro de los formatos más conocidos se encuentra el PDF el cual constituye un estándar abierto y combina los tipos de datos antes mencionados.

Además este formato es independiente de la plataforma o sistema operativo, puede ser generado utilizando varios lenguajes de programación y ser creado con características especiales, como la accesibilidad para personas discapacitadas. El PDF es empleado en facturación y publicaciones electrónicas, emisión de contratos digitales y licencias en Internet.

En Cuba, según (2), son innumerables las instituciones que deberían usar la firma digital como un mecanismo de seguridad, uno de ellos son los Tribunales Populares de la República de Cuba, que son el Órgano del Estado al que corresponde, como objetivos fundamentales, el control y la preservación de la legalidad, sobre la base de la vigilancia del estricto cumplimiento de la Constitución, las leyes y demás disposiciones legales, por los organismos del Estado, entidades económicas y sociales y por los ciudadanos, y la promoción y el ejercicio de la acción penal pública en representación del Estado.

La Universidad de las Ciencias Informáticas (UCI) ha estado llevando a cabo el desarrollo de proyectos de identificación y seguridad con el uso de la tecnología de tarjetas inteligentes en pasaportes electrónicos. Además en el marco de los Acuerdos del ALBA, la universidad desarrolló una solución informática para las oficinas de Registros y Notarías de la República Bolivariana de Venezuela, dicha solución contó con un componente para generación y verificación de firma digital de documentos.

Soluciones para la administración pública, comercio y gobierno electrónico están dentro de las líneas investigativas y posibles líneas de producción de proyectos en la UCI. Dichas soluciones son impensables sin la gestión eficiente de la seguridad. Es por ello que se hace necesario el estudio y desarrollo de productos informáticos que incorporen elementos como la firma digital y el uso de dispositivos seguros como las tarjetas inteligentes.

Todo esto que hemos tratado nos lleva a la siguiente **situación problemática**:

El procedimiento actual que se llevará a cabo en el proyecto TPC para el tratamiento de documentos jurídicos en formato electrónico no garantizaría el valor probatorio ni la seguridad legal que salvaguarda a estos cuando son tratados en papel como método tradicional a través de los procedimientos y ordenamientos jurídicos existentes.

En estos procesos se genera y manipula un gran número de información, como son: expedientes de acusados, documentos de pruebas y testimonios. Esta información es clasificada y sensible, solo puede ser modificada por los jueces, fiscales y abogados que trabajen directamente con el proceso que se esté llevando a cabo.

Por lo que no se puede alterar salvo por el personal autorizado, de manera que se necesita un componente que permita darle integridad y autenticidad a toda la información que se genere digitalmente, garantizando con esto que se pueda detectar cualquier modificación realizada en estos documentos.

Con el análisis de lo antes expuesto y con el fin de darle solución a la problemática existente, este trabajo plantea la siguiente **interrogante**: *¿Cómo garantizar la **autenticidad**, **integridad** y **no repudio** de los documentos electrónicos?*

Para dar solución a la interrogante se asume como **objeto de estudio**: *La Seguridad Informática* y como **campo de acción**: *La firma digital avanzada de documentos*.

Esta investigación plantea la **hipótesis** de que: *Si se desarrolla un componente para la firma digital de documentos usando tecnologías **PKI** se garantizaría la integridad, no repudio y autenticidad de la información almacenada en dichos documentos.*

Por lo que el **objetivo general** de esta investigación es: *Desarrollar un componente de software para la firma digital de documentos legales.*

Para dar cumplimiento al **objetivo general** se han definido los siguientes **objetivos específicos**:

- Identificar y valorar diferentes aplicaciones de firma digital. Analizar los estándares para el trabajo con infraestructuras PKI y firmas digitales.
- Diseñar un componente para la firma digital de documentos.
- Implementar el componente para la firma digital de documentos.
- Evaluar la calidad y cumplimiento con estándares del componente para la firma digital de documentos.

Con esta investigación se pretende obtener como **posible resultado**: *Un componente para la firma digital de documentos en formato PDF que pueda ser comercializado y utilizado en otros proyectos de desarrollo de software con necesidades similares.*

Además, haciendo una **valoración económica** y calculando el **impacto social** que significaría la creación de dicho componente, se ha llegado a las siguientes conclusiones:

Con esta solución las empresas podrán firmar todos los documentos digitales que considere importantes, estos documentos una vez firmados tendrán el mismo valor legal que los documentos impresos.

Esta herramienta permitiría a la entidad que decida utilizarla:

- Ahorro de insumos (papel), ya no es necesario imprimir un documento para que tenga valor legal.

- Ahorro de largas, complejas e inciertas etapas probatorias en trámites judiciales y administrativos para demostrar la validez de documentos digitales.

- Manipulación de contratos, expedientes de acusados, atestados, etc. en su estado digital por las partes interesadas, sin perder la autenticidad e integridad de los documentos.

- Disminuir los gastos por concepto de envío de paquetes de correo o viajes. Actualmente cuando una empresa necesita enviar documentos legales a otros municipios, provincias o países es necesario enviar estos documentos por correo postal, DHL (por sus siglas en inglés), enviarlos en un automóvil o avión con representante de la empresa.

Toda investigación debe seguir **métodos científicos** que permitan llevar a cabo la misma en pos de cumplimentar los objetivos propuestos para obtener los resultados esperados.

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones.

El método científico se puede clasificar en **teóricos** y **empíricos**, los cuales están dialécticamente relacionados.

Los **métodos teóricos:** Permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir mas allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas y para su ejecución se apoyan en el proceso de análisis y síntesis.

Clasificación de los métodos teóricos:

Métodos históricos: Analizan la trayectoria completa del fenómeno, su condicionamiento a los diferentes periodos de la historia, revela las etapas principales de su desenvolvimiento y las conexiones históricas fundamentales.

Métodos lógicos: Se basan en el estudio histórico del fenómeno, ponen de manifiesto la lógica interna de su desarrollo, de su teoría y definen el conocimiento mas profundo de su esencia. Estos métodos expresan en forma teórica la esencia del objeto, explican la historia de su desarrollo, reproducen el objeto en su forma superior y permiten unir el estudio de la estructura del objeto de investigación con su concepción histórica.

Por otra parte, los **métodos empíricos:** Describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional.

Aunque existen diversas opiniones la mayoría de los autores concuerdan que los métodos empíricos generales son: la **observación**, la **medición** y la **experimentación**.

En cuanto a la **observación,** se puede definir que: Es la percepción planificada dirigida a un fin y relativamente prolongada de un hacho o fenómeno. Es el instrumento universal del científico, se realiza de forma consciente y orientada a un objetivo determinado.

Por otro lado, la **medición** es: El procedimiento que se realiza con el objetivo de obtener información numérica acerca de una propiedad o cualidad del objeto, donde se comparan magnitudes medibles y conocidas.

Por último, haremos referencia a la **experimentación,** que no es más que: La demostración del vínculo causal entre dos fenómenos, llegando a considerarse solamente como científicas las demostraciones que se realizaban por vía experimental.

Además, pudiéramos agregar los **métodos particulares**: Los cuales son más específicos, ya que están desarrollados en base a las características propias de cada ciencia y para su aplicación están vinculados a técnicas de recolección de datos característicos de ese tipo de investigación.

A continuación se describen algunos de estos métodos aplicados en las ciencias sociales:

La **entrevista:** Es una conversación planificada entre el investigador y el entrevistado para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación.

La **encuesta**: Se realiza cuando la información que se realiza puede ser obtenida a partir de la respuesta que una persona o varias puedan dar a un cuestionario pre elaborado, y las mismas están dispuestas a colaborar con la investigación.

A continuación presentamos una propuesta de la estructura que tendrá el presente **Trabajo de Diploma**:

Capítulo 1. **Fundamentación Teórica*:*** *Se presentan los conceptos fundamentales relacionados con la firma digital. Se seleccionan las herramientas y tecnologías para desarrollar el componente. Al finalizar el capítulo se realiza un estudio de aplicaciones de software para la firma digital.*

Capítulo 2. Propuesta de Solución: *Se presentan las fases de exploración y planificación definidas por la metodología **XP** para dar solución al problema científico. Se identifican las diferentes historias de usuarios y los requerimientos no funcionales, se realiza el plan de iteraciones y el plan de entregas.*

Capítulo 3. Diseño, Implementación y Validación de la Solución: *Primeramente se realiza el diseño, posteriormente se da cumplimiento a los planes trazados a través de las fases de iteraciones a primera liberación y producción, se codifica la solución diseñada y finalmente se realizan las pruebas de aceptación con el cliente.*

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción.

El objetivo principal que se persigue en este capítulo es la comprensión de las bases teóricas de la firma digital como mecanismo para garantizar la autenticidad de documentos electrónicos. Además se hace un estudio del estado del arte de los algoritmos fundamentales que dan soporte a la firma digital. También se seleccionan las herramientas y tecnologías para desarrollar el componente, así como la arquitectura base. Finalmente se analizan otras aplicaciones para la firma digital de documentos en formato PDF y se dan las conclusiones del capítulo.

1.2 Criptografía.

Desde muy antiguo, la Criptografía ha sido una ciencia asociada con la protección de información de carácter militar o político. La historia de la Criptografía se divide en tres períodos. Hasta finales de los cuarenta era más un arte que una ciencia: aunque algunos de los algoritmos de cifrado ya poseían una consistente formulación algebraica, carecían globalmente de una sólida base matemática. Esta se conoce como la época de la Criptografía pre científica. Es a partir de 1948, con el surgimiento de la Teoría de la Comunicación de Shannon cuando comienza el período de la Criptografía científica, más elaborada y con una amplia base matemática.

Actualmente, la importancia de las telecomunicaciones en el mundo de la informática ha convertido a la criptografía en una necesidad, y hoy en día es muy elevado el número de investigadores que se dedican a desarrollar algoritmos y protocolos de cifrado de la información. Su uso es generalizado en todos aquellos campos de la vida moderna en los que se necesita mantener la información en secreto o en los que se vela por la integridad de la misma evitando posible ataques para destruirla o manipularla.

1.2.1 Qué es cifrar.

Es el proceso por el cual un mensaje (**texto en claro**) es transformado en otro ininteligible (**criptograma**) usando una función matemática y una clave de cifrado. Por descifrar se conoce al proceso inverso, es decir, transformar el criptograma en el mensaje original (texto en claro) usando una función matemática y una clave de descifrado.

Un sistema de cifrado o **criptosistema** (figura 1) está constituido por un **emisor** que genera el mensaje original, un dispositivo cifrador que transforma el texto en claro en texto cifrado o criptograma, un **canal** de almacenamiento o transmisión, un dispositivo descifrador que recupera el mensaje original, y una **receptora** del mensaje.

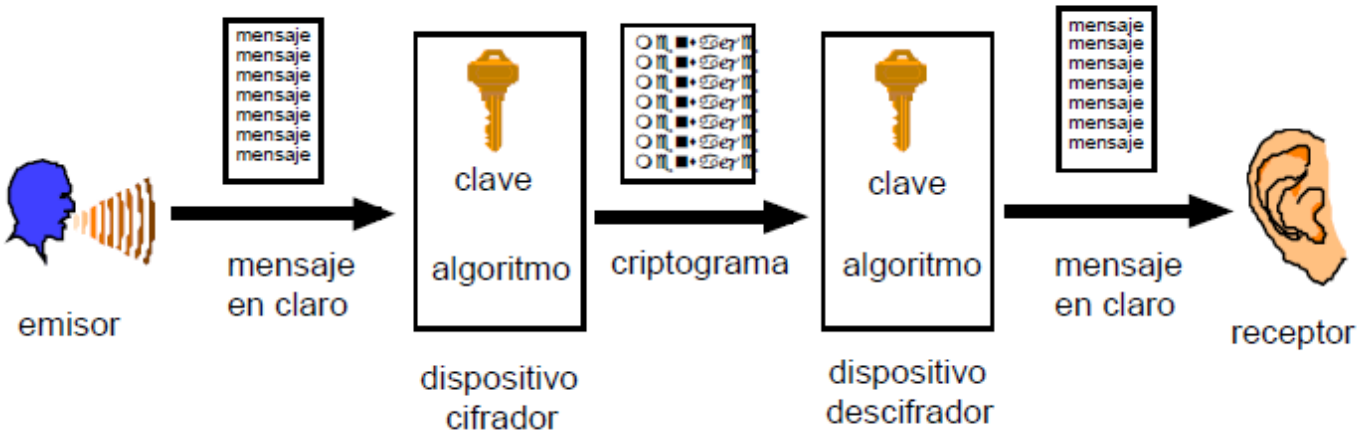


Figura 1. Estructura de un criptosistema.

1.3 Criptografía de Clave Secreta.

También llamados criptosistemas de de clave privada, de clave única, simétricos o convencionales. Son los herederos de la criptografía clásica, y utilizan un esquema de cifrado con una única clave, que debe ser usada y compartida en secreto por emisor y receptora. La principal ventaja que presentan es su simetría. Los papeles del emisor y la receptora son intercambiables dado que ambos emplean la misma clave.

Por otro lado son los criptosistemas más estudiados, y por tanto los más seguros. Además suelen estar implementados por algoritmos muy eficientes, a menudo directamente sobre hardware. No tienen alternativa cuando se trata de utilizar la criptografía para proteger información que no interviene en protocolos de comunicación.

Estos sistemas, además de asegurar la confidencialidad de la información, permiten autenticación de la emisora: siempre que la clave permanezca en secreto, el receptor tiene la seguridad de que la emisora es quien dice ser y no un suplantador, dado que hipotéticamente son ellos dos los únicos que conocen la clave.

Figura 2. Esquema de un sistema simétrico de cifrado.

En este tipo de criptografía se usan dos clases de algoritmos: algoritmos de cifrado por bloques y algoritmos de cifrado por flujo, a continuación se explicará cada tipo.

1.3.1 Cifrado en Flujo.

Según (3), el cifrado en flujo el algoritmo va cifrando el mensaje a medida que avanza. Idealmente cada bit de información es codificado sin esperar al siguiente, y además la forma de cifrar va cambiando de manera dinámica. Ese cambio no debe obedecer a un esquema cíclico (como en el método de Vigenère), pues entonces el criptograma podría dividirse en unidades funcionales menores para su análisis. La forma más segura de conseguir esto es disponiendo de una clave arbitrariamente larga (idealmente infinita), que consiga que dos partes diferentes del mensaje se codifiquen de forma distinta e independiente, en función de la porción de clave utilizada.

1.3.2 Cifrado por Bloques.

El sistema de cifrado en flujo (4) resulta excesivamente complicado para la mayor parte de las aplicaciones prácticas de la criptografía, y suele reservarse para casos en los que el volumen de la información transmitida es bajo, pero cuya confidencialidad es cuestión de vital importancia. Alternativamente, la mayor parte de los algoritmos en uso utilizan el cifrado por bloques, mucho más económico y sencillo de gestionar.

1.3.3 Algoritmos de Cifrado Simétrico.

1.3.3.1 Data Encryption Standard (DES).

Es un esquema de encriptación simétrico desarrollado en 1977 por el Departamento de Comercio y la Oficina Nacional de Estándares de EEUU en colaboración con la empresa IBM, que se creó con objeto de proporcionar al público en general un algoritmo de cifrado normalizado para redes de ordenadores (3). En general, DES utiliza una clave simétrica de 64 bits, de los cuales 56 son usados para la encriptación, mientras que los 8 restantes son de paridad, y se usan para la detección de errores en el proceso.

1.3.3.2 Triple DES.

Como hemos visto, el sistema **DES** se considera en la actualidad poco práctico, ya que el desarrollo de la computación demostró que era viable un ataque por fuerza bruta contra este algoritmo, y en una conferencia de la RSA Data Security en 1999, la llave de un mensaje cifrado con DES fue rota en menos de 24 horas. Para solventar este problema y continuar utilizando **DES** se creó el sistema **Triple DES** (**TDES**), basado en tres iteraciones sucesivas del algoritmo **DES**, con lo que se consigue una longitud de clave de 128 bits, y que es compatible con **DES** simple.

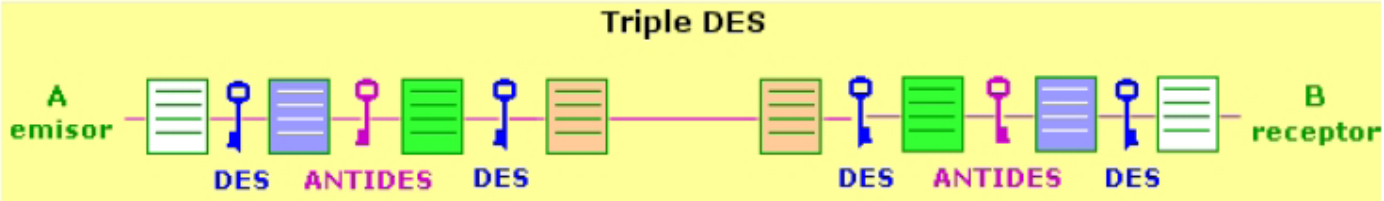


Figura 3. Esquema de funcionamiento del algoritmo **TDES**.

1.3.3.3 Advanced Encryption Standard (RSA).

También conocido como **Rijndael** (5), es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. Se espera que sea usado en el mundo entero y analizado exhaustivamente, como fue el caso de su predecesor, el Data Encryption Standard (**DES**).

1.4 Criptografía de Clave Pública.

También llamados de doble clave o **asimétrico**, están basados en la existencia de dos claves distintas: una para cifrar y la otra para descifrar (figura 4). A pesar de la estrecha relación entre ambas claves, disponer de una de ellas no permite recuperar la otra. Esto conlleva que la clave de cifrado pueda hacerse pública sin peligro de que nadie averigüe la de descifrado, cuyo secreto puede ser mantenido.

Las claves pública y privada tienen características matemáticas especiales, de tal forma que se generan siempre a la vez, por parejas, estando cada una de ellas ligada intrínsecamente a la otra, de tal forma que si dos llaves públicas son diferentes, entonces sus llaves privadas asociadas también lo son, y viceversa (6).

Mientras que la clave privada debe mantenerla en secreto su propietario, ya que es la base de la seguridad del sistema, la clave pública es difundida ampliamente por Internet, para que esté al alcance del mayor número posible de personas, existiendo servidores que guardan, administran y difunden dichas claves.

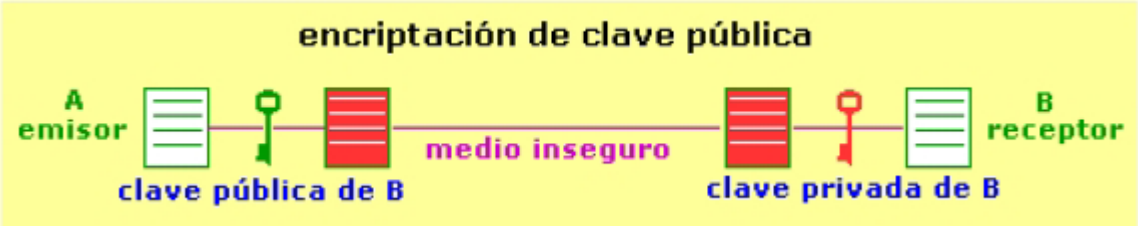


Figura 4. Esquema de un sistema asimétrico de cifrado.

1.4.1 Algoritmos de Cifrado Asimétrico.

1.4.1.1 Diffie Hellman.

Es un algoritmo asimétrico que se emplea fundamentalmente para acordar una llave común entre dos entidades (llave de sesión), a través de un canal de comunicación inseguro. La ventaja de este sistema es que no son necesarias llaves públicas en el sentido estricto, sino una información compartida por los dos comunicantes (3). Este procedimiento es conocido como “acuerdo de llaves”. La seguridad del algoritmo está dada por la dificultad de computar logaritmos discretos (7), problema muy parecido al planteado por **RSA**. Se aconseja que las llaves sean de una longitud no menor que 1024bit.

1.4.1.2 RSA.

Este algoritmo fue propuesto por *Ron **R**ivest*, *Adi **S**hamir*, y *Len **A**dleman* en 1978, es el más conocido y versátil de los algoritmos de llave pública usados actualmente. Es apropiado para las operaciones de cifrado y firma digital (6).Su seguridad está basada en la dificultad de factorizar números enteros muy grandes y por el grado de avance en esta área de las matemáticas, se sugiere el empleo de llaves de al menos 1024bit.

1.4.1.3 ElGamal.

Algoritmo basado en el protocolo **Diffie Hellman** y descrito en 1984 por *Taher ElGamal*. Es usado en software libre en GNU Privancy Guard, versiones recientes del PGP y otros sistemas. La seguridad se basa en la actual incapacidad computacional de solucionar el problema discreto del logaritmo. Diseñado previamente para producir firmas digitales pero se extendió posteriormente para codificar mensajes.

1.5 Firma Digital.

La firma digital es una herramienta tecnológica que permite garantizar la autoría e integridad de los documentos digitales, posibilitando que estos presenten una característica que únicamente era propia de los documentos en papel. Se trata de un conjunto de datos asociados a un mensaje digital que permite garantizar la identidad del firmante y la integridad del mensaje. La firma digital se basa en el empleo de dos técnicas muy distintas. La primera es la criptografía asimétrica o de clave pública y la otra es el uso de funciones hash o resumen (6).

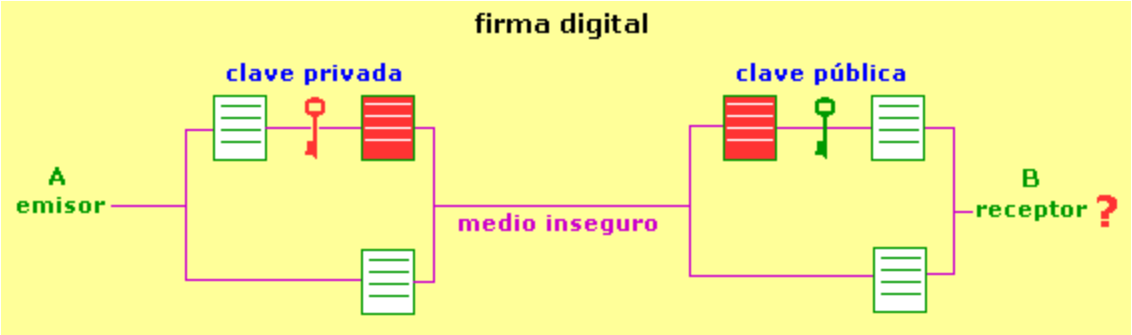


Figura 5. Esquema básico de una Firma Digital.

Las normas **TS733** y **TS903** propuestas por el **ETSI** (European Telecommunications Standards Institute) definen los formatos técnicos de la firma electrónica. La primera se basa en el formato clásico **PKCS#7** (Public Key Cryptography Standards) y la segunda en XMLDsig (Extensible Markup Language Digital Signature).

Bajo estas normas se definen tres modalidades de firma:

- **Firma básica.** Incluye el resultado de operación de hash y clave privada, identificando los algoritmos utilizados y el certificado asociado a la clave privada del firmante.

- **Firma fechada o avanzada.** A la firma básica se añade un sello de tiempo calculado a partir del hash del documento firmado por una TSA (Time Stamping Authority).

- **Firma reconocida o completa.** A la firma avanzada, se añade información sobre la validez del certificado procedente de una consulta de CRL (Certificate Revocation List) o de OCSP (Online Certificate Status Provider) realizada a la Autoridad de Certificación.

La firma digital se utiliza para identificar la identidad del firmante, para autentificar que el que rubrica dice ser quien es. Además es imposible que sea suplantada, como ocurre con la firma tradicional que es fácil de falsificar.

El proceso de firma digital consta de dos partes bien diferenciadas:

1.5.1 Proceso de Firma Digital.

El proceso de firmado consiste en alimentar un programa de cómputo con el documento a firmar y su clave privada (que solo el conoce). El programa encripta con la clave privada el documento produciendo como resultado un mensaje encriptado denominado firma digital. Juntos, el documento y la firma constituyen el documento firmado.

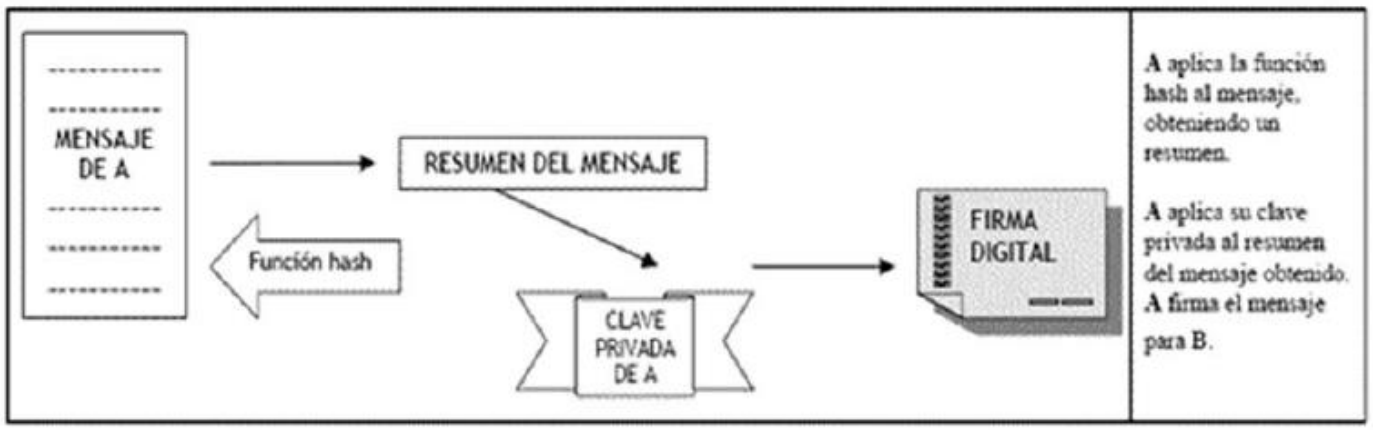


Figura 6. Proceso para firmar digitalmente un documento.

1.5.2 Proceso de Verificación de la Firma.

El proceso de verificación (algunos lo llaman proceso de autenticación) consiste en alimentar un programa mediante el documento firmado y la clave pública del supuesto firmante, el programa desencripta con la clave publica la firma y lo compara con el documento indicando si es autentico o no.

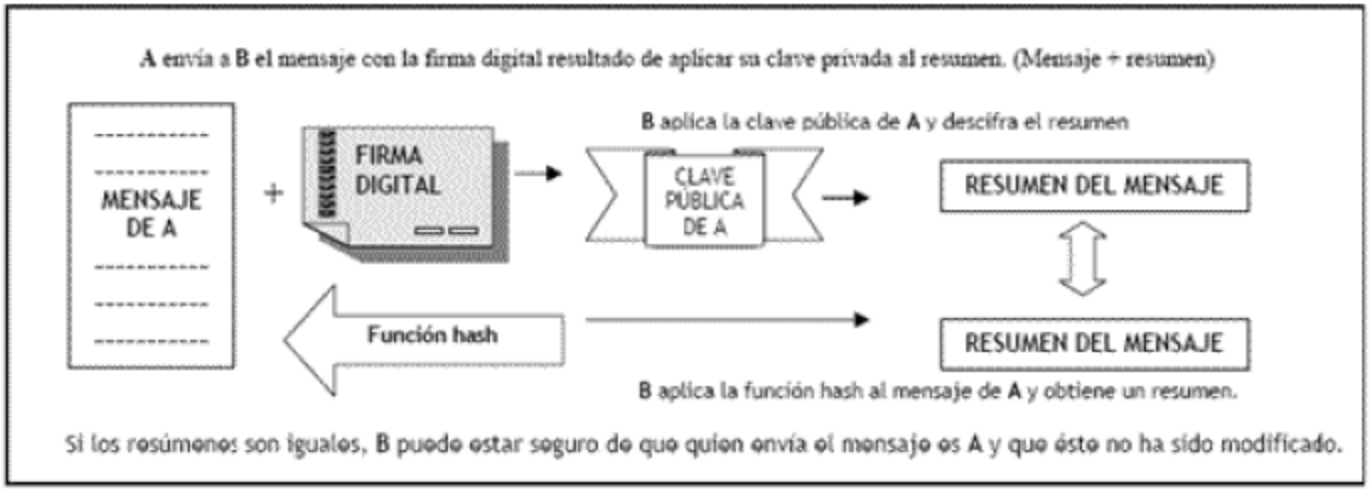


Figura 7. Proceso para verificar la firma digital de un documento.

1.5.3 Características de la Firma Digital.

Las fuentes consultadas coinciden en que la firma digital provee a los documentos electrónicos de las siguientes características:

1.5.3.1 Integridad.

Es necesario estar seguro de que los datos que enviamos llegan íntegros, sin modificaciones, a su destino final. La *integridad* se consigue combinando Criptografía, funciones hash y firmas digitales.

1.5.3.2 Autenticidad.

Todas las entidades participantes en la transacción deben estar perfecta y debidamente identificadas antes de comenzar la misma. La a*utenticidad* se consigue mediante el uso de los certificados y firmas digitales.

1.5.3.3. No Repudio.

Debemos estar seguros de que una vez enviado un mensaje con datos importantes o sensibles el destinatario de los mismos no pueda negar el haberlos recibido. El *no repudio* se consigue mediante los certificados y la firma digital.

1.5.3.4. Confidencialidad.

Debemos estar seguros de que los datos que enviamos no puedan ser leídos por otra persona distinta del destinatario final deseado, o que si ocurre esto, el espía no pueda conocer el mensaje enviado. La *confidencialidad* de consigue en las transacciones electrónicas con el uso de la Criptografía.

1.5.4 Algoritmos de Firma Digital.

Actualmente muchas aplicaciones de seguridad que implementan la firma digital soportan varios de los mecanismos existentes, aunque no todos sean utilizados con igual frecuencia. A continuación alguna de las características según (4) de estos mecanismos:

1.5.4.1 SHA-1/RSA.

Este método combina **SHA-1** como función hash y **RSA** como algoritmo de cifrado, y el proceso de firma y verificación es de manera semejante al descrito en este epígrafe. Además el proceso de verificación resulta más rápido que el proceso de firmado, característica que lo hace superior a otros métodos como **DSA**, debido que lo más común es que un mensaje o documento electrónico sea firmado una vez y verificado varias veces. Actualmente es el método de firma digital más utilizado por las aplicaciones criptográficas y es soportado por muchos protocolos de seguridad.

1.5.4.2 Digital Signature Algorithm (DSA).

Este algoritmo fue publicado en 1994 formando parte del estándar **DSS** (Digital Signature Standard) adoptado por el Gobierno de Estados Unidos como estándar para el proceso de firma digital. Está basado en el problema de los logaritmos discretos y en el criptosistema **ElGamal** y a diferencia del algoritmo **RSA** que se utiliza en las operaciones de cifrado y firma digital, **DSA** solo puede realizar la firma digital. Dentro de las características de este algoritmo están la rapidez del proceso de firma respecto al proceso de verificación, el uso de **SHA-1** como función hash y las llaves de 1024bit.

1.5.4.3 Elliptic Curve DSA (ECDSA).

Es una versión, basada en la teoría de curvas elípticas, del algoritmo **DSA**. Es más rápido que **RSA** y **DSA** pero actualmente muy pocas aplicaciones criptográficas lo incorporan entre sus algoritmos. Al igual que **DSA** forma parte del estándar **DSS** (Digital Signature Standard). Por ser un algoritmo de curva elíptica, la longitud de las llaves son mucho más cortas que en **RSA** y **DSA**, y es aconsejable utilizar al menos 192 bits.

1.5.5 Funciones Criptográficas de Resumen.

Son unas funciones matemáticas que realizan un resumen del documento a firmar. Su forma de operar es comprimir el documento en un único bloque de longitud fija, bloque cuyo contenido es ilegible y no tiene ningún sentido real. Tanto es así que por definición las funciones hash son irreversibles, es decir, que a partir de un bloque comprimido no se puede obtener el bloque sin comprimir, y si no es así no es una función hash.

Estas funciones son además de dominio público. A un mensaje resumido mediante una función hash y encriptado con una llave privada es lo que en la vida real se denomina **firma digital**.

El esquema de firma digital mediante una función hash es el siguiente:

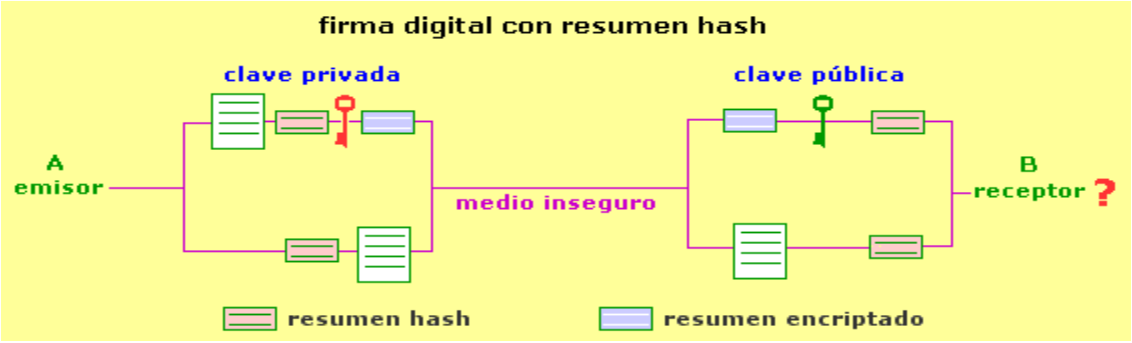


Figura 8. Esquema básico de una función de resumen hash.

Las funciones hash y la firma digital son elementos indispensables para el establecimiento de canales seguros de comunicación, basados en los Certificados Digitales.

Las funciones hash más conocidas y usadas según (4) son:

1.5.5.1 Message Digest 5 (MD5).

Fue diseñado en 1992 por Ron Rivest como versión mejorada de otros algoritmos suyos, buscando su eficacia incluso en arquitecturas de usuario final modestas. De ahí procede la simplicidad de su diseño. Tiene la propiedad de que cada bit de la entrada tiene influencia en cada bit del resultado.

1.5.5.2 Secure Hash Algorithm (SHA1).

Desarrollado como parte integrante del Secure Hash Standar (SHS) y el Digital Signature Standar (DSS) por la Agencia de Seguridad Nacional Norteamericana, NSA. Sus creadores afirman que la base de este sistema es similar a la de MD4 de Rivest, y ha sido mejorado debido a ataques nunca desvelados. La versión actual se considera segura (por lo menos hasta que se demuestre lo contrario) y es muy utilizada algoritmo de firma, como en el programa PGP en sus nuevas claves **DH/DSS** (Diffie-Hellman/Digital Signature Standar).

1.5.5.3 RIPEMD 160.

Desarrollado por un grupo de investigadores europeos, entre los que se encuentra Hans Dobbertin y otros investigadores incluidos en el proyecto RIPE (RACE Integrity Primitives Evaluation). Su primera versión adolecía de las mismas debilidades que MD4, produciendo colisiones, pero las versiones mejoradas actuales son consideradas seguras. Maneja claves muy robustas, normalmente de 160 bits, aunque existen versiones de 128 y se están planteando nuevas de 256 y 320 bits. Es muy rápido, no está patentado y su código fuente es abierto, de libre acceso.

1.6 Certificados Digitales.

Un **Certificado Digital** es un documento electrónico que contiene datos identificativos de una persona o entidad (empresa, servidor web, etc.) y la llave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza, denominada **Autoridad Certificadora**. Las principales Autoridades Certificadoras (AC) actuales son **VeriSign** (filial de RSA Data Security Inc.) y **Thawte** (8).

El certificado Digital vincula indisolublemente a una persona o entidad con una llave pública, y mediante el sistema de firma digital se asegura que el certificado que recibimos es realmente de la persona que consta en el mismo. El sistema de firma digital liga un documento digital con una clave de cifrado.

1.6.1 Especificaciones PKCS para certificados digitales.

Los documentos de la serie PKCS son especificaciones producidas por los laboratorios RSA. Fueron publicadas por primera vez en 1991 y son ampliamente referenciados e implementados. Las contribuciones de la serie de PKCS forman parte de muchos estándares formales y de facto, incluyendo los documentos del ANSI X9, PKIX, SET, S/MIME, y SSL (8).

La siguiente es la lista de documentos PKCS que se van a utilizar en este trabajo de diploma:

- **PKCS #7. Cryptographic Message Syntax Standard.** Es un estándar para firmar o cifrar datos (ellos lo llaman "sobreado"). Dado que el certificado es necesario para verificar datos firmados, es posible incluirlos en la estructura SignedData. Un archivo **.P7C** es simplemente una estructura SignedData, sin datos para firmar (8).

- **PKCS #12. Personal Information Exchange Syntax Standard.** Evolucionó del estándar **.PFX** (Personal inFormation eXchange) y se usa para intercambiar objetos públicos y privados dentro de un archivo. Un archivo **.PEM** puede contener certificados o claves privadas, encerrados entre las líneas **BEGIN/END** apropiadas (8).

1.7 Herramientas y tecnologías a utilizar.

1.7.1 Metodologías de Desarrollo de Software.

De acuerdo con el análisis de algunos conceptos se pudo concluir que las metodologías de desarrollo son el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a producir un nuevo producto. Imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo crean ejecutando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería. Tienen el propósito de garantizar la eficacia a la hora de cumplir los requisitos y de disminuir la pérdida de tiempo empleado durante el desarrollo del mismo.

Dentro de las metodologías de desarrollo existen dos grandes grupos, las conocidas **Metodologías Tradicionales** y las **Metodologías Ágiles**. Según (9), las primeras enfatizan en el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto, mientras que las segundas dan mayor importancia a la capacidad de respuesta a los cambios y a mantener una buena relación con el cliente para llevar al éxito el proyecto.

1.7.1.1 Metodologías Tradicionales.

Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un software más eficiente y predecible. Para ello, se hace un especial hincapié en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto software. Este planteamiento está basado en el resto de disciplinas de ingeniería, a pesar de que el software no pueda considerarse como la construcción de una obra clásica de ingeniería. La más empleadas a nivel mundial son: **RUP, MSF y MÉTRICA 3.**

1.7.1.2 Metodologías Ágiles.

Las metodologías ágiles son un marco de trabajo conceptual de la ingeniería de software permitiendo promover iteraciones a lo largo de todo el ciclo de vida de un proyecto. Minimiza riesgos desarrollando productos en cortos plazos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración y cada una incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Las más utilizadas son **XP, SCRUM y FDD.**

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Tabla 1. Diferencias entre metodologías ágiles y las tradicionales.

De acuerdo con (tabla 1), seleccionamos las metodologías ágiles porque sus características son afines a los intereses que se persiguen en este trabajo de diploma, además de que facilitan y agilizan todo el trabajo de planificación y diseño, dando prioridad casi total a la implementación.

A continuación analizaremos a SCRUM y Extreme Programming, ya que son las metodologías ágiles de más uso a nivel para posteriormente decidirnos por la más adecuada a nuestros intereses.

1.7.1.3 PROGRAMACIÓN EXTREMA (EXTREME PROGRAMMING, XP).

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (10).

XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

El ciclo de trabajo de esta metodología consta de 4 fases de trabajo:

- **Planificación:** en la cual la tarea más importante es la creación de las Historias de Usuario (HU) la cuales son escritas por el cliente para expresar las funcionalidades que desea y son usadas para estimar el tiempo de desarrollo de la funcionalidad que describen.
- **Diseño:** en esta fase se crean las Tarjetas CRC las cuales definen una clase expresando las funcionalidades de esta y las otras clases con las que colabora.

- **Codificación:** donde se definen las tareas de desarrollo para que los desarrolladores tengan una guía para implementar todas las HU.
- **Pruebas:** en esta fase se le realizan diferentes test a cada una de las HU para probar que cumplen con las funcionalidades que desea el cliente.

XP se basa en cinco valores, la raíz de los elementos básicos que, a juicio de Beck (10) son los realmente importantes para el éxito de desarrollo de software. Estos valores son la orientación para el propio desarrollo y la inspiración de toda la metodología. Cuatro de ellos son los mismos que XP original, y se añade el respeto como el quinto valor.

Estos valores son:

- **Comunicación:** la mayoría de los problemas y los errores son causados por la falta de comunicación. Por esta razón, debe existir una gran comunicación entre los miembros del equipo y entre el equipo y los clientes.
- **Simplicidad:** este es el más intelectual de los valores XP. "Las cosas más simples en que podría trabajar". Sin embargo, la simplicidad no simplista es muy difícil.
- **Comentarios:** siempre debe ser capaz de medir el sistema y saber hasta qué punto está terminado a partir de las características necesarias.
- **Valor:** todos los procesos y las metodologías son herramientas para manejar y reducir nuestros miedos.
- **Respeto:** Si los miembros de un equipo no se preocupan por sí y su trabajo, la metodología no puede funcionar.

Estos cinco valores no dan consejos específicos sobre cómo gestionar un proyecto, o cómo escribir programas. Para este fin, son necesarias prácticas, y antes de las prácticas, se necesitan principios. Los principios son el puente entre los valores, que es lo sintético y abstracto, y las prácticas, que dicen en realidad cómo desarrollar software.

Los principios fundamentales que rigen XP, según (11) son:

- **Beneficio Mutuo:** cada actividad debe beneficiar a todas las personas y organizaciones interesadas. Esto es quizás el más importante principio de XP, y de los más difíciles de adherir.
- **Mejora:** la mejora continua es fundamental para XP. La perfección no existe, pero debemos esforzamos para la perfección.
- **Oportunidad:** los problemas deben verse como una oportunidad de mejora. Usted debe experimentar problemas, pero para obtener la excelencia, no puede simplemente corregir los problemas. Necesita convertirlos en oportunidades de aprendizaje y mejora.
- **Calidad:** la calidad debe estar siempre al máximo. Aceptar una menor calidad no afecta ni al ahorro, ni al rápido desarrollo.

Además, podemos decir que XP se basa en trece prácticas primarias, y once corolarios de prácticas. Las prácticas primarias deben aplicarse en primer lugar, y cada uno de ellos puede producir una mejora en el proceso de desarrollo de software.

1.7.1.4 SCRUM.

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos.

Scrum debe todas sus prácticas desde un proceso iterativo e incremental. El esqueleto de Scrum se muestra en la figura 6. El círculo inferior representa una iteración del desarrollo de las actividades que ocurren una tras otra. El producto de cada iteración es un incremento en el producto (12).

El círculo superior representa la reunión diaria que ocurre durante la iteración, en la cual los miembros individualmente del grupo conocen, inspeccionan las actividades y hacen los cambios apropiados. Como resultado de la iteración queda una lista de requerimientos. Este ciclo se repite durante todo el proyecto.

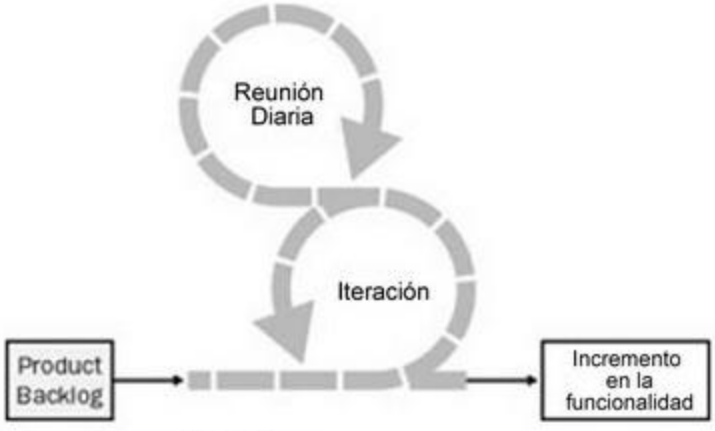


Figura 9. Esqueleto de Scrum.

El corazón de Scrum no es válido en la iteración. El equipo revisa los requerimientos, considerando la tecnología disponible, evaluando sus habilidades y capacidades. Luego, determina colectivamente como van a construir la funcionalidad, mientras que encuentran y discuten nuevas complejidades, dificultades y sorpresas. El equipo muestra cuales son las necesidades y cual es la mejor forma de satisfacerlas. Este proceso de creatividad es el corazón de la productividad de Scrum (13).

Hay solo tres roles en Scrum: el **Product Owner** (dueño del producto), ***The Team*** *(*el equipo*)*, y el **ScrumMaster** *(*Maestro Scrum).

Todas las responsabilidades de manejo de un proyecto se dividen entre estos tres papeles:

- El **Product Owner** es el responsable de cuidar los intereses de cada uno de los participantes, recalcándolos con una “estaca”, lo cual se transformara en el producto final.

- **The Team:** Los miembros del equipo son responsables en conjunto del éxito de cada iteración y del proyecto en su totalidad.

- El **ScrumMaster** es responsable del proceso Scrum, debe enseñar la metodología Scrum a cada integrante implicado en el proyecto.

En cuanto a los principales artefactos que se generan con SCRUM es necesario describir a continuación los más importantes:

- **Sprint Backlog.** Define el trabajo, o las tareas, que el Team desarrollará para poder convertir el Product Backlog seleccionado para ese Sprint, en un incremento potencialmente funcional del producto.

- **Carta Burndown.** Demuestra la cantidad de trabajo restante a través de tiempo. La carta burndown es una manera excelente de visualizar la correlación entre la cantidad de trabajo restante en cualquier punto y el progreso de los equipos de proyecto en la reducción de este trabajo.

- **Product Backlog.** La Lista de Requerimientos se llama *the Product Backlog* (Reserva del Producto*).*

1.9.1.5 Criterio de selección de Extreme Programming como metodologías de desarrollo a utilizar.

El estudio de las dos metodologías de desarrollo ágiles más usadas en la actualidad ha demostrado que XP es la mejor opción y la que más se adecua a las necesidades del sistema a implementar ya que es una de las metodologías de desarrollo de software más exitosas utilizadas en la actualidad para proyectos de corto plazo y para equipos de desarrollo pequeños. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final.

1.7.2 PHP como Lenguaje de Programación a utilizar.

PHP es considerado como un puntal en el desarrollo de aplicaciones Web, es un lenguaje interpretado, sencillo, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Es además software libre, multiplataforma y rápido. Contiene varias bibliotecas para funciones matemáticas, de bases de datos e inclusive trabajo con redes. Es un lenguaje fácil de usar que cuenta, desde la versión 5.0, con un magnífico trabajo con el paradigma orientado a objetos.

Se decide PHP como lenguaje para desarrollar la aplicación Web debido a que viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envío de correo, gestión de un e-commerce, XML, creación de PDF), es multiplataforma, o sea, funciona en todas las plataformas que soporten Apache, es software libre. Se puede obtener en la Web y su código está disponible bajo la licencia GPL (14).

1.7.3 NeatBeans 6.8 beta como Entorno Integrado de Desarrollo.

NetBeans IDE es un entorno de desarrollo integrado (IDE) modular y basado en estándares, escrito con el lenguaje de programación Java. El proyecto de NetBeans consta de un IDE de código abierto con gran variedad de funciones escrito con el lenguaje de programación Java y una plataforma para aplicaciones de cliente enriquecidas que se puede utilizar como marco genérico para crear cualquier tipo de aplicación.

NetBeans IDE 6.8 es el primer IDE en ofrecer compatibilidad para todas las especificaciones de Java EE 6, con compatibilidad mejorada para JSF 2.0/Facelets, Java Persistence 2.0, EJB 3.1 (incluido el uso de EJB en aplicaciones web), servicios web RESTful y GlassFish v3. También lo recomendamos para desarrollar con el último JavaFX SDK 1.2.1 y para crear aplicaciones web PHP con la nueva versión PHP 5.3 o con la estructura Symfony (15).

1.7.4 OPENSSL como Biblioteca básica de soporte para PHP.

OpenSSL es un proyecto de software desarrollado por los miembros de la comunidad *Open Source* para libre descarga y está basado en *SSLeay*, desarrollado por Eric Young y Tim Hudson. Consiste en un robusto paquete de herramientas de administración y librerías relacionadas con la criptografía, que suministran funciones criptográficas a otros paquetes como *OpenSSH* y navegadores web (para acceso seguro a sitios HTTPS).

Estas herramientas ayudan al sistema a implementar el *Secure Sockets Layer (SSL)*, así como otros protocolos relacionados con la seguridad, como el *Transport Layer Security (TLS)*. Este paquete de software es importante para cualquiera que esté planeando usar cierto nivel de seguridad en su máquina con un sistema operativo libre basado en *GNU/Linux*. *OpenSSL* también permite crear certificados digitales que pueden aplicarse a un servidor, por ejemplo *Apache* (16).

1.8 Estado del Arte.

Cada vez más administraciones y empresas requieren el envío de documentos sobre soporte electrónico. Con el uso aplicaciones para la **firma digital** estas organizaciones pueden enviar a través de Internet documentos importantes en formato **PDF**, como ofertas de negocios, facturas electrónicas y formularios de una forma rápida y sin que estos sean alterados.

Existen múltiples aplicaciones que permiten firmar digitalmente documentos **PDF**, a continuación algunas características de las más usadas:

1.8.1 Ascertia PDF Sign&Seal:

Herramienta de software especializada en **firma digital** y seguridad de documentos en formatos **PDF**, posee su propio visor de documentos y se integra con el almacén de llaves de Windows, a través del cual puede hacer uso de las **tarjetas inteligentes**. Incluye **sellado de tiempo** y tiene una interfaz gráfica muy amigable.

1.8.2 Adobe Acrobat 9.0.

Es la aplicación más completa para edición y creación de documentos en formato **PDF** y permite la creación de este tipo de documento a partir de múltiples formatos. Además se pueden crear formularios en los documentos y darles seguridad a través de la **firma digital** y el cifrado. Se caracteriza por ser una aplicación muy robusta y se integra con el almacén de llaves de Windows, a través del cual puede hacer uso de las **tarjetas inteligentes.** La desventaja está dada porque al ser una aplicación profesional su costo es elevado.

1.8.3 PortableSigner:

Está desarrollado en lenguaje java y permite el uso de certificados en diferentes formatos sin necesidad de importarlos en el sistema operativo. Entre las ventajas de **PortableSigner** están la sencillez de su interfaz de usuario y su característica multiplataforma, y como desventaja, que no incorpora la tecnología de **tarjetas inteligentes** en el proceso de **firma digital**.

1.9 Conclusiones Generales del Capítulo.

A partir de la revisión bibliográfica realizada para el desarrollo de este capítulo, se concluye:

1. La criptografía provee los métodos y mecanismos para garantizar la integridad, autenticidad y no repudio de la información.

2. El algoritmo SHA-1/RSA es el estándar de facto para la realización de la firma digital.

3. Se seleccionaron las herramientas y tecnologías necesarias para el desarrollo del componente.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.

2.1 Introducción.

La propuesta de solución al problema científico se abordará utilizando la metodología de desarrollo de software XP (Extreme Programming). Esta metodología propone seis fases y el objetivo que se persigue con la elaboración de este capítulo es mostrar la evolución de la solución durante las fases de Exploración y Planificación, además de presentar los diferentes artefactos generados en dichas fases.

2.2 Metodología XP. Fases.

La metodología XP, como metodología ágil, enfatiza en el carácter iterativo e incremental del desarrollo. Una iteración es un período de una a cuatro semanas, en el cual el cliente selecciona las funcionalidades que desea que se implementen en dicha iteración. Al final de una iteración el cliente puede ejecutar las pruebas funcionales relativas a la iteración. Estas iteraciones tienen lugar a lo largo de las diferentes fases (17).

Existe una fase de análisis inicial orientada a planificar las iteraciones de desarrollo y cada iteración incluye diseño, codificación y pruebas, sub fases superpuestas de tal manera que no se separen en el tiempo.

La siguiente figura muestra las fases en las que se subdivide el ciclo de vida de un proyecto de software con XP:

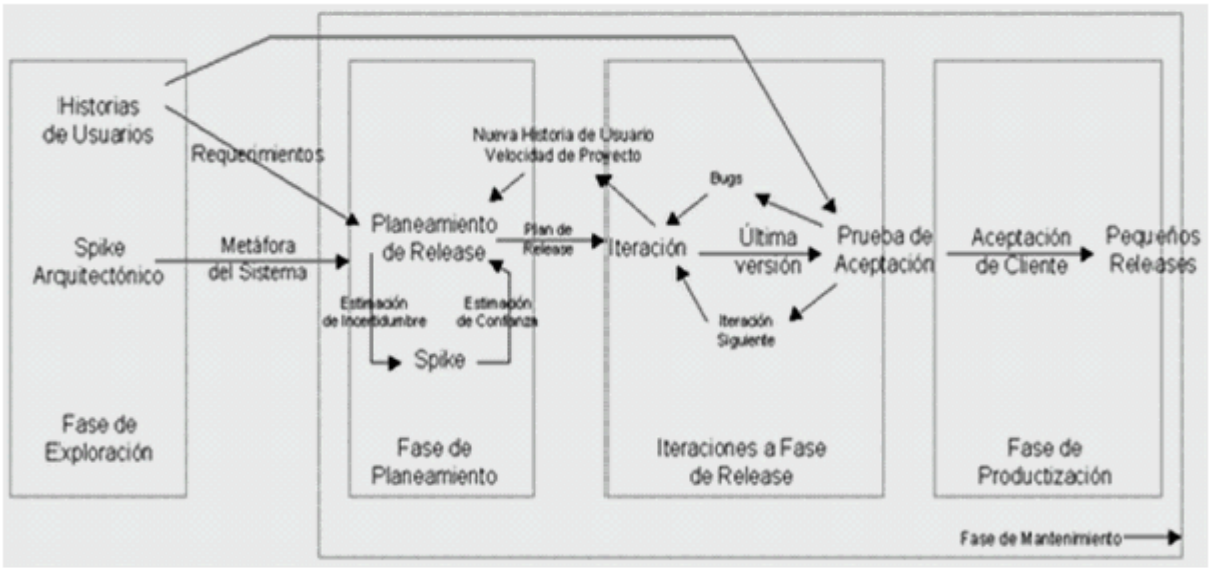


Figura 10. Fases de un proyecto con XP.

A continuación se verán algunos detalles de las fases de Exploración y Planeamiento. El resto de las fases serán tratadas en el próximo capítulo.

2.2.1 Fase de Exploración.

En esta fase, los clientes plantean qué necesitan que haga el sistema, y dichas necesidades se recogen en lo que XP define como *Historias de Usuarios*. Las *Historias de Usuarios* (ver tabla 1) sirven como lista de requerimientos del sistema; además son utilizadas para estimar el tiempo en la planificación de las liberaciones (18).

Historia de Usuario	
Número: <Número identificador de la Historia de Usuario>	**Nombre:** <Nombre de la Historia de Usuario>
Usuario:	
Prioridad del Negocio: <Alta, Media, Baja>	**Riesgo de Desarrollo:** <Alta, Media, Baja>
Puntos Estimados: <0,1,2,3>	**Iteración Asignada:** <1, …, n>
Descripción:	
Observaciones:	

Tabla 2. Plantilla de una Historia de Usuario.

Al mismo tiempo que se definen las *Historias de Usuarios,* el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Además se exploran las posibilidades de la arquitectura del sistema.

El tiempo invertido en la fase de exploración, toma de unas semanas a unos pocos meses, dependiendo del tamaño del problema en cuestión y el grado de familiarización que hayan alcanzado los miembros del equipo de desarrollo con las tecnologías a utilizar.

2.2.2 Fase de Planeamiento.

El propósito de la fase de Planeamiento es que los clientes y programadores se pongan de acuerdo en la fecha de las liberaciones, además los clientes priorizan las *Historias de Usuario* que deberán ser desarrolladas en cada liberación. Con una buena fase de Exploración la fase de Planeamiento debería tomar entre uno y dos días. En el caso del planeamiento de la primera liberación puede tomar entre dos y seis meses porque es donde se tienen en cuenta los problemas más significativos del negocio, además los programadores estiman cuánto esfuerzo requiere cada historia y a partir de allí se define el cronograma.

2.3 Propuesta de Solución.

Después de haberse precisado algunos conceptos básicos de la metodología XP, específicamente de las fases de Exploración y Planificación, se tratarán los aspectos específicos de la solución propuesta.

2.3.1 Historias de Usuario.

Durante la fase de Exploración se definieron las diferentes *Historias de Usuarios*, las tablas a continuación describen cada una de ellas.

Historia de Usuario.	
Número: HU1.	**Nombre:** Gestionar Archivos.
Usuario: Desarrollador.	
Prioridad del Negocio: Alta.	**Riesgo de Desarrollo:** Baja.
Puntos Estimados: 2.	**Iteración Asignada:** 1.
Descripción: Permitirá cargar los archivos que serán firmados digitalmente, seleccionando la ruta del archivo. Además permitirá definir el nombre que se desea tenga el archivo firmado o el directorio con los archivos firmados.	
Observaciones:	

Tabla 3. HU1. Gestionar Archivos.

Historia de Usuario.	
Número: HU2.	**Nombre:** Gestión de Certificados Digitales.
Usuario: Desarrollador.	
Prioridad del Negocio: Alta.	**Riesgo de Desarrollo:** Alta.
Puntos Estimados: 2.	**Iteración Asignada:** 1.
Descripción: Permitirá cargar el contenedor de la llave privada destinada a la firma digital y el certificado digital correspondiente. El contenedor de llaves podrá ser una tarjeta inteligente o un archivo.	
Observaciones:	

Tabla 4. HU2. Gestión de Certificados Digitales.

Historia de Usuario.	
Número: HU3.	**Nombre:** Firmar Documento.
Usuario: Desarrollador.	
Prioridad del Negocio: Alta.	**Riesgo de Desarrollo:** Alta.
Puntos Estimados: 2.	**Iteración Asignada:** 1.
Descripción: Firmará digitalmente un documento utilizando un certificado digital que estará almacenado en un archivo y como resultado de la operación creará un nuevo documento firmado con el certificado seleccionado.	
Observaciones:	

Tabla 5. HU3. Firmar Documento.

Historia de Usuario	
Número: HU4.	**Nombre:** Validar Firma de Documento.
Usuario: Desarrollador.	
Prioridad del Negocio: Alta.	**Riesgo de Desarrollo:** Alta.
Puntos Estimados: 2.	**Iteración Asignada:** 1.
Descripción: Realizará la validación del Documento que fue firmado, el cual puede estar almacenado en un archivo o en un servidor de datos, garantizando que los ficheros firmados no se han modificado y el “no repudio” del documento por el poseedor del certificado electrónico que firmo el documento.	
Observaciones:	

Tabla 6. HU4. Validar Firma de Documento.

2.3.2 Metáfora del Sistema.

Después de haberse definido las funcionalidades que el sistema debe cumplir, los requerimientos no funcionales y las herramientas de desarrollo, el equipo procede a la creación de la *Metáfora*. Esta es una breve descripción de cómo debe funcionar el sistema en su totalidad. La *Metáfora* guía todo el desarrollo como una gran *Historia de Usuario* ayudando al equipo a entender los elementos básicos del sistema y sus relaciones (18).

Metáfora.
Para firmar un documento PDF se seleccionará el documento en formato a firmar. Luego el nombre que tendrá el documento firmado o la carpeta que contendrá los documentos firmados. Será necesario seleccionar el certificado digital para realizar la firma digital. Después de estas acciones se podrá realizar la firma digital.

Tabla 7. Metáfora del Sistema.

2.3.3 Estimación de Tiempo.

Concluida la fase de Exploración los programadores estiman el tiempo que necesitan para desarrollar cada Historia de Usuario, este valor se expresa en semanas. Una Historia de Usuario no debe desarrollarse en menos de una, ni en más de dos semanas. En otro caso, será necesario acoplar o dividir las Historias de Usuarios. Como se ha dicho anteriormente, este valor es estimado y se irá acercando a la realidad con el transcurso de las iteraciones.

Historia de Usuario.	Estimación.
Gestionar Archivos.	**2**
Gestión de Certificados Digitales.	**2**
Firmar Documento.	**2**
Validar Firma de Documento.	**2**
Total.	**8**

Tabla 8. Estimación de Tiempo.

2.3.4 Plan de Iteraciones.

Luego a partir de las prioridades definidas por el cliente se crea el *Plan de Iteraciones*. En la presente solución se han identificado cuatro *Historias de Usuarios* y se definió una sola iteración, por considerarse más que suficiente para desarrollar el componente de firma digital (18).

Iteración.	Nro. HU	Historia de Usuario.	Duración estimada.
Iteración 1.	HU1	Gestionar Archivos.	8 Semanas.
	HU2	Gestión de Certificados Digitales.	
	HU3	Firmar Documento.	
	HU4	Validar Firma de Documento.	

Tabla 9. Plan de Iteraciones.

2.3.5 Plan de Entregas.

Para finalizar la fase de Planeamiento se acuerda con el cliente el *Plan de Entregas* que define las fechas en que serán liberadas las versiones funcionales del producto. Este artefacto cumple con el principio de la metodología XP de las "liberaciones frecuentes", generalmente asociadas al fin de un grupo de iteraciones (18).

A continuación se muestra el *Plan de Entrega*:

ENTREGABLE	FIN DE ITERACIÓN 1.
FIRMADIGITAL.php	07/04/2010
VERIFICARFIRMADIGITAL.php	15/04/2010

Tabla 10. Plan de Entrega.

2.4 Conclusiones del Capítulo.

Luego de terminadas las fases de Exploración y Planificación de la solución propuesta, se concluye:

1. Se definieron cuatro Historias de Usuario:

    a. Gestionar Archivos.
    b. Gestionar Certificados Digitales.
    c. Firmar Documento.
    d. Validar Firma de Documento.

2. Se llevó a cabo la estimación total del tiempo de desarrollo en base al Plan de Estimación, así como se fijaron plazos de entrega de versiones del componente plasmados en el Plan de Entregas.

3. Se definieron dos iteraciones con el objetivo de planificar el trabajo del equipo de desarrollo.

CAPÍTULO 3: DISEÑO, IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN.

3.1 Introducción.

En el presente capítulo se abordan las fases diseño, implementación y pruebas de la Metodología XP. Uno de los artefactos fundamentales es la creación de las tarjetas **CRC** (Clase, Responsabilidades y Colaboración) las cuales permiten brindar un mayor enfoque orientado a objetos. Por otra parte se describen cada una de las tareas planificadas para llevar a cabo el desarrollo de cada una de las historias de usuario detectadas. Además se muestran las pruebas de aceptación creadas por el cliente para verificar el buen funcionamiento de la aplicación. Las mismas fueron verificadas en cada entrega que se realiza del producto en la planificación establecida.

3.2 Diseño.

A continuación, abordaremos la Fase de Diseño de la metodología XP. Uno de los artefactos fundamentales es la creación de las tarjetas CRC (Clase-Responsabilidades-Colaboración) las cuales permiten brindar un mayor enfoque orientado a objetos.

3.2.1 Tarjetas CRC.

XP propone realizar diseños simples y sencillos, hacerlo todo lo menos complejo posible para lograr que sea entendible e implementable. Realizar una correcta especificación de los nombres de métodos y clases ayuda a comprender mejor lo diseñado y facilita las posteriores ampliaciones y la reutilización del código. Nunca se debe añadir funcionalidades extras al software aunque se piense que serán factibles en el futuro. El uso de la técnica de las tarjetas CRC para diseñar con la máxima simplicidad posible permite al programador centrarse en el desarrollo orientado a objetos (19).

Estas tarjetas representan objetos, la clase a la que pertenece el objeto se escribe en la parte superior de la tarjeta, en una columna a la izquierda se escriben las responsabilidades u objetivos que debe cumplir el objeto y a la derecha las clases que colaboran con cada responsabilidad como se muestra en la siguiente plantilla:

Clase	
Responsabilidades	Colaboraciones

Tabla 11. Plantilla Tarjeta CRC.

Esta nueva técnica de diseño es adoptada como alternativa a los diagramas UML de las clases, pues en estas se plasman las responsabilidades que tienen cada objeto y las clases con las que tienen que interactuar para darles respuesta brindando así la información que se necesita a la hora de implementar.

Firmar Documento.	
Responsabilidades	**Colaboraciones**
Firmar Documento.	*Subir Archivo a Firmar.*
Generar Archivo Firma Digital.	
Codificar Firma Digital.	
Salvar Archivo de Firma Digital.	

Tabla 12. Tarjeta CRC Firmar Documento.

Validar Firma Documento.	
Responsabilidades	**Colaboraciones**
Decodificar Firma Digital.	*Subir Archivo Firmado.*
Validar Firma Digital.	

Tabla 13. Tarjeta CRC Validar Firma Documento.

Subir Archivo a Firmar.	
Responsabilidades	**Colaboraciones**
Cargar Documento a Firmar.	Firmar Documento.
Cargar Certificado Digital.	
Salva Archivo de Firma Digital.	

Tabla 14. Tarjeta CRC Subir Archivo a Firmar.

Subir Archivo Firmado.	
Responsabilidades	**Colaboraciones**
Cargar Documento Firmado.	Validar Archivo Firmado.
Cargar Certificado Digital.	
Cargar Firma Digital.	

Tabla 15. Tarjeta CRC Subir Archivo Firmado.

3.3 Implementación.

De las historias de usuarios creadas por el cliente se generan **Tareas de Ingeniería** (**TI**) o tareas de programación como también se les conoce, algunas historias no se van a dividir en tareas pues se les dará solución conjuntamente como propone XP en sus principios, la programación en parejas para una mejor optimización del código.

3.3.1 Iteraciones a Primera Liberación.

En la fase “Iteraciones a primera liberación” es donde se da cumplimento al “Plan de Iteraciones” (ver epígrafe 2.3.6). En cada iteración se desarrollan las sub fases de “Diseño”, realización de “Pruebas Unitarias”, “Codificación de la Solución” y “Refactorización”. Dada la naturaleza ágil de XP la sub-fase más importante del ciclo de vida del proyecto es la “Codificación de la Solución”.

Al finalizar la fase de “Iteraciones a Primera Liberación” el cliente estará apto para realizar las pruebas de aceptación.

3.3.2 Tareas de la ingeniería.

Antes de comenzar a codificar la solución es necesario saber qué codificar. Las historias de usuarios no ofrecen el nivel de detalles requerido para llevar a cabo esta acometida, es por eso que son divididas en tareas de la ingeniería. Una historia de usuario generalmente se divide en más de una tarea de la ingeniería y a partir de estas tareas comienza el ciclo de la fase de Iteraciones a primera liberación (20).

Según el Plan de Iteraciones (ver epígrafe 2.3.6) las historias de usuario se agruparon en dos iteraciones. A continuación se muestran las tareas de la ingeniería derivadas de cada historia de usuario por iteración.

3.3.2.1 Tareas de la Ingeniería. Iteración 1.

3.3.2.1.1 HU1 Gestionar Archivos PDF.

TAREA	
Número. HU1_T1.	**Historia de Usuario.** Gestionar Archivos.
Nombre. Diseño de Interfaz para Gestión de Archivos.	
Tipo. Desarrollo.	**Puntos Estimados.** 1
Fecha Inicio.	**Fecha Fin.**
Responsable: Gianny Miranda Ramos.	
Descripción. La interfaz debe contar con campos de texto para entrar la dirección del documento y botones que muestren un selector de ficheros para seleccionar el documento visualmente.	

Tabla 16. HU1_T1. Diseño de Interfaz para Gestión de Archivos.

TAREA	
Número. HU1_T2.	**Historia de Usuario.** Gestionar Archivos.
Nombre. Cargar Archivos PDF.	
Tipo. Desarrollo.	**Puntos Estimados.** 1
Fecha Inicio.	**Fecha Fin.**
Responsable: Gianny Miranda Ramos.	
Descripción. El usuario podrá introducir en un campo de texto la ruta del documento que desee firmar. El sistema debe validar que la ruta del documento o directorio sea correcta, en caso contrario debe mostrar un mensaje de error informando al usuario.	

Tabla 17. HU1_T2. Cargar Archivos.

TAREA	
Número. HU1_T3	**Historia de Usuario.** Gestionar Archivos.
Nombre. Salvar Archivos.	
Tipo. Desarrollo.	**Puntos Estimados.** 1
Fecha Inicio.	**Fecha Fin.**
Responsable: Gianny Miranda Ramos.	
Descripción. El usuario podrá introducir en un campo de texto la ruta donde desea guardar el documento firmado. El sistema debe validar que la ruta del documento o directorio sea correcta, en caso contrario debe mostrar un mensaje de error informando al usuario.	

Tabla 18. HU1_T3. Salvar Archivos.

3.3.2.1.2 HU2 Gestionar Certificados Digitales.

TAREA	
Número. HU2_T1**.**	**Historia de Usuario.** Gestionar Certificados Digitales.
Nombre. Diseño de Interfaz para Gestión de Certificados Digitales.	
Tipo. Desarrollo.	**Puntos Estimados.** 1
Fecha Inicio.	**Fecha Fin.**
Responsable: Gianny Miranda Ramos.	
Descripción. La interfaz debe contar con un campo de selección que contendrá los certificados existentes y un botón para mostrar los detalles del certificado seleccionado.	

Tabla 19. HU2_T1. Diseño de Interfaz para Gestión de Certificados Digitales.

TAREA	
Número. HU2_T2**.**	**Historia de Usuario.** Gestionar Certificados Digitales.
Nombre. Cargar Certificados Digitales.	
Tipo. Desarrollo.	**Puntos Estimados.** 1
Fecha Inicio.	**Fecha Fin.**
Responsable: Gianny Miranda Ramos.	
Descripción. El usuario podrá introducir en un campo de texto la ruta del archivo contenedor del certificado o seleccionarlo de manera visual. Además debe soportar las extensiones de archivo **.pfx** y **.p12**.	

Tabla 20. HU2_T2. Cargar Certificados Digitales.

3.3.2.1.3 HU3 Firmar documento PDF.

TAREA	
Número. HU3_T1.	**Historia de Usuario.** Firmar Documento.
Nombre. Firmar Documento.	
Tipo. Desarrollo.	**Puntos Estimados.** 1
Fecha Inicio.	**Fecha Fin.**
Responsable: Gianny Miranda Ramos.	
Descripción. Se debe cargar el documento a partir de la ruta del mismo. Para realizar esta operación será necesario haber seleccionado un certificado digital.	

Tabla 21. HU3_T1. Firmar Documento.

3.3.2.1.4 HU4 Validar Firma de Documento PDF.

TAREA	
Número. HU4_T1.	**Historia de Usuario.** Validar Firma de Documento.
Nombre. Cargar Documento Firmado.	
Tipo. Desarrollo.	**Puntos Estimados.** 1
Fecha Inicio.	**Fecha Fin.**
Responsable: Gianny Miranda Ramos.	
Descripción. Se debe cargar el documento firmado a partir de la ruta del mismo. Para realizar esta operación será necesario haber seleccionado un certificado digital.	

Tabla 22. HU4_T1. Cargar Documento Firmado.

TAREA	
Número. HU4_T2.	**Historia de Usuario.** Validar Firma de Documento.
Nombre. Validar Firma de Documento.	
Tipo. Desarrollo.	**Puntos Estimados.** 1
Fecha Inicio.	**Fecha Fin.**
Responsable: Gianny Miranda Ramos.	
Descripción. Se llevará a cabo la validación del documento previamente cargado en HU4_T1, en caso de no ser válida la firma o no estar firmado, se mostrará el correspondiente mensaje de advertencia.	

Tabla 23. HU4_T2. Validar Firma de Documento.

3.4 Pruebas de Aceptación (PA).

Uno de los pilares de la metodología XP es el proceso de pruebas. La metodología propone probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos no deseados a la hora de realizar modificaciones y refactorizaciones.

XP propone la realización de pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente.

A la hora de codificar no se sigue la regla de XP que aconseja crear pruebas unitarias con entornos de desarrollo antes de programar. Las pruebas se obtienen de la descripción de requisitos plasmados en las historias de usuarios, y estas especifican las barreras que deben pasar las distintas funcionalidades del programa, procurando codificar teniendo en cuenta las pruebas que se deben vencer (21).

Para realizar las pruebas de aceptación el cliente utiliza la siguiente plantilla:

Prueba de aceptación
HU: Nombre de la historia de usuario que va a comprobar su funcionamiento.
Nombre: Nombre del caso de prueba.
Descripción: Descripción del propósito de la prueba.
Condiciones de ejecución: Precondiciones para que la prueba se pueda realizar.
Entrada/Pasos de ejecución: Pasos para probar la funcionalidad.
Resultado esperado: Resultado que se desea de la prueba.
Evaluación de la prueba: Aceptada o Denegada.

Tabla 24. Plantilla Prueba de Aceptación.

A continuación se muestran las pruebas realizadas para verificar el funcionamiento de cada historia de usuario en las entregas que se le hacen al cliente cumpliendo con lo establecido en el cronograma de entregas.

Prueba de Aceptación.
HU: Gestionar Archivos.
Nombre: Cargar Archivos.
Descripción: Se deben seleccionar tanto el documento PDF a firmar como el certificado digital. Estos serán cargados al navegador para ser utilizados en el proceso de firma.
Condiciones de ejecución: Seleccionar correctamente el documento PDF a firmar y el certificado digital.
Entrada/Pasos de ejecución: Se intenta cargar el documento PDF y el certificado digital.
Resultado esperado: Se han cargado correctamente los archivos seleccionados.
Evaluación de la prueba: Satisfactoria.

Tabla 25. PA Cargar Archivos.

Prueba de Aceptación.
HU: Gestionar Archivos.
Nombre: Salvar Archivos.
Descripción: Se deben salvar en la dirección predeterminada tanto el archivo de firma digital como la clave pública extraída del certificado digital.
Condiciones de ejecución: Se ha realizado correctamente el proceso de firma digital.
Entrada/Pasos de ejecución: Se intenta salvar la firma digital y la clave pública.
Resultado esperado: Se han salvado correctamente los archivos generados.
Evaluación de la prueba: Satisfactoria.

Tabla 26. PA Salvar Archivos.

Prueba de Aceptación.
HU: Firmar Documento.
Nombre: Firmar Documento.
Descripción: Se debe firmar el documento seleccionado utilizando el certificado digital previamente escogido.
Condiciones de ejecución: Se ha realizado correctamente el proceso de firma digital.
Entrada/Pasos de ejecución: Se realiza la firma digital del documento seleccionado.
Resultado esperado: Se ha firmado correctamente el documento.
Evaluación de la prueba: Satisfactoria.

Tabla 27. PA Firmar Documento.

Prueba de Aceptación.
HU: Validar Firma de Documento.
Nombre: Validar Firma de Documento.
Descripción: Se debe validar el documento seleccionado utilizando los archivos de firma digital y clave públicas generados en el proceso de firma.
Condiciones de ejecución: Se ha realizado correctamente el proceso de validación de la firma digital del documento.
Entrada/Pasos de ejecución: Se realiza la validación de la firma digital del documento seleccionado.
Resultado esperado: Se ha validado correctamente el documento firmado.
Evaluación de la prueba: Satisfactoria.

Tabla 28. PA Validar Firma de Documento.

3.5 Conclusiones del Capítulo.

Luego de terminadas las fases de Iteraciones a Primera Liberación y Producción de la solución propuesta, se concluye:

1. A partir del desglose de las historias de usuario en tareas de la ingeniería fue posible la realización del diseño y codificación.

2. El desarrollo guiado por pruebas aseguró la ejecución correcta de la solución en todo el período de desarrollo, aminorando el tiempo invertido en el ciclo de compilación y ejecución.

3. Las pruebas de aceptación concluyeron de manera exitosa demostrando la satisfacción del cliente con la solución.

CONCLUSIONES GENERALES.

Al término de esta investigación se concluye:

1. Se desarrolló un componente para la firma digital de documentos en formato PDF, que garantiza la autenticidad, integridad y no repudio de información almacenada en dichos documentos.

2. Se garantizó la validez a largo tiempo de la firma digital mediante el empleo del estándar **PKCS12**.

3. Con el empleo del lenguaje de programación PHP el componente se integra de manera satisfactoria a la aplicación principal del proyecto TPC y a su vez puede ser empleado por otras instituciones que requieran de su uso.

RECOMENDACIONES.

Como producto de la culminación de la presente investigación se tiene en cuenta las siguientes recomendaciones:

1. Incluir verificación de documentos PDF firmados, haciendo uso de CRL y OCSP.

2. Explotar las características del formato PDF para garantizar la confidencialidad de la información a través del cifrado simétrico.

3. Incluir soporte para firmar digitalmente varios tipos de documentos electrónicos.

4. Agregar servicio de sellado de tiempo mediante el uso de una TSA (Time Stamp Authority).

5. Desarrollar una versión para servidores.

TRABAJOS CITADOS.

[1] *Digital signature bill enables e-commerce.* **Jones, Jennifer and Johnston, Margret.** 25, 6 19, 2000, InfoWorld, Vol. 22.

[2] **Valdés, Marisol.** BetSime - La Revista del Empresario Cubano. [Online] Julio 2003. http://www.betsime.disaic.cu/secciones/fin_ja_03.htm.

[3] **Burnett, Steve and Paine, Stephen.** *RSA Security's Official Guide to Cryptography.* s.l.: McGraw-Hill, 2001.

[4] **Lucena, Manuel.** *Criptografía y Seguridad en Computadores.* 2004.

[5] **Mogollón, Manuel.** *Cryptography and Security Services: Mechanisms and Aplications.* s.l.: Cybertech Publishing, 2007.

[6] **Rankl, Wolfgang and Effing, Wolfgang.** *Smart Card Handbook.* 3ra. s.l.: John Wiley & Sons Ltd, 2002.

[7] **Laboratories, RSA.** *RSA Laboratories Frequently Asked Questions About Today's Cryptography, Version 4.1.* s.l.: RSA Security Inc., 2000.

[8] **Adams, C., et al.** RFC: 3161. *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP).* 2001.

[9] **Solís, Camilo J. and Figueroa, Roberth G.** *Metodologías Tradicionales vs. Metodologías Ágiles.* s.l.: Universidad Técnica Particular de Loja, Escuela de Ciencias de la Computación.

[10] **Beck, K.** "Extreme Programming Explained. Embrace Change", Pearson Education, 1999. Traducido al español como: "Una explicación de la programación extrema. Aceptar el cambio", Addison Wesley, 2000.

[11] Wells, J. Donovan. Extreme Programming: A gentle introduction. [En línea] 1999. http://www.extremeprogramming.org/.

[12] **Schwaber K., Beedle M., Martin R.C.** "Agile Software Development with SCRUM". Prentice Hall. 2001.

[13] **Jeff Sutherland,** http://jeffsutherland.com/scrum/index.html [En línea].

[14**] Dondo, Agustín.** ¿Por qué elegir PHP? PHP en Castellano.

[En línea] http://www.programacion.com/php/articulo/porquephp/.

[15] [En línea] http://netbeans.org/community/releases/68/relnotes_es.html

[16] [En línea] http://www.php.net/manual/en/openssl.official_guide.php

[17] **Mendoza Sánchez, María A.** informatizable. [En línea] junio 7, 2004. [Cited: febrero 11, 2008.] http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

[18] [En línea] http://www.wikipedia.com.es/extremmeprograming/historias de usuario.

[19] **Fuentes, Isidro.** Edib. *Tarjetas CRC.* [En línea] 28 de octubre de 2008. http://informatica.escuelaedib.com/index.php?option=com_myblog&show="Tarjetas-CRC".html&Itemid=61

[20] [En línea] http://www.wikipedia.com.es/extremmeprograming/tareas de ingeniería.

[21] **J.J Gutiérrez, M.J Escalona, M. Mejías, J.Torres.** *Pruebas del sistema en programación extrema.* [En línea] http://www.lsi.us.es/~javierj/investigacion_ficheros/PSISEXTREMA.pdf

GLOSARIO DE TÉMINOS.

ALBA: La **AL**ternativa **B**olivariana para los Pueblos de Nuestra **A**mérica es una propuesta de integración enfocada para los países de América Latina y el Caribe que pone énfasis en la lucha contra la pobreza y la exclusión social.

CLR (**C**ertificates **R**evocation **L**ist): Lista con los números de serie de los certificados revocados por una entidad certificadora.

DSS (**D**igital **S**ignature **S**tandard). Fue adoptado por el gobierno de los Estados Unidos en 1994 como estándar para la firma digital. La especificación incluye los algoritmos DSA, SHA-1, RSA y ECDSA.

ISO (**I**nternational **O**rganization for **S**tandardization): La Organización Internacional para la Normalización es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.

OCSP (**O**nline **C**ertificate **S**tatus **P**rotocol): Protocolo especificado por RFC 2560, que permite determinar si un certificado está revocado o no, a través de un servicio expuesto por la autoridad certificadora que lo emitió.

PDF (Portable Document Format): Creado pensando en la oficina con información 100% digital. Este nuevo formato además de texto, incorporaba imágenes y gráficos vectoriales entre otros.

PHP (Hy**P**ertext **P**re **P**rocessor.): Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas.

PKCS (**P**ublic **K**ey **C**ryptography **S**tandards): Se refiere a un grupo de estándares de criptografía de clave pública concebidos y publicados por los laboratorios de RSA.

PKCS#12. Este estándar describe la sintaxis que deben tener los datos de identidad personal (llaves privadas, certificados, información secreta y otros) al ser transportados.

OPENSSL (**O**pen **S**ecurity **S**ocket **L**ayer): Consiste en un robusto paquete de herramientas de administración y librerías relacionadas con la criptografía, que suministran funciones criptográficas.

SSL (Secure **S**ockets **L**ayer**).** Es un protocolo de comunicación segura a través de una red de computadoras. Utiliza algoritmos criptográficos para garantizar la integridad i confidencialidad de la comunicación.

TSP (Time **S**tamp **P**rotocol**)** Protocolo de intercambio de información entre aplicaciones y la autoridad de sellado de tiempo. Define la estructura de los pedidos y respuestas de la autoridad de sellado de tiempo.