Sistema prototipo para la gestión de recetas médicas

Figueroa Romero Luis Manuel, Martínez San Román Aarón Hazel, Olea Zúñiga Jonathan,
Dra. Sandra Díaz Santiago, M. en C. Axel Ernesto Moreno Cervantes
Escuela Superior de Cómputo I.P.N. Ciudad de México
Tel. 5729-6000 ext. 52000 y 52021. E-mail: luismachivas@hotmail.com, hazelmsr@gmail.com,
jonathanoleaz@hotmail.com

***Resumen*—En este trabajo terminal se plantea el desarrollo de un sistema web para digitalizar los procesos involucrados en la prescripción y dispensación de medicamentos. Adicionalmente, se integrarán mecanismos criptográficos para proveer autenticidad a la receta que prescribe el médico. El propósito de lo anterior es tener mayor control en los medicamentos que el paciente recibe y al mismo tiempo, proveer seguridad a los procesos antes mencionados.**

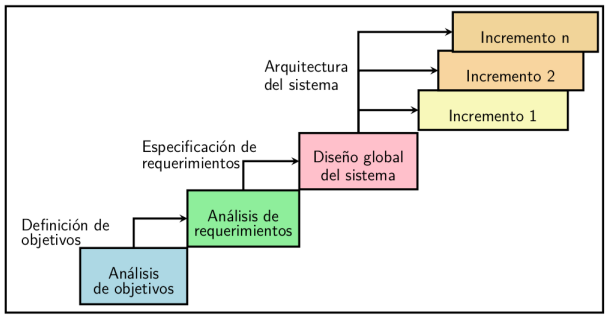
***Index Terms*—Criptografía, Firma electrónica, Sistema web, Receta médica.**

I. Introducción

Una receta médica es un documento en el que se expresa una orden para la dispensación de medicamentos y demás insumos con el objetivo de llevar a cabo un tratamiento o la prevención de algún padecimiento. El sustento de los datos en la receta médica (principalmente el medicamento, dosis, duración y vía de administración) es dado por el diagnóstico médico.

II. Metodología

En el modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia define qué funcionalidades serán entregadas en ese incremento; éstos incrementos pueden ser puestos a producción a partir del momento de la entrega, siendo independiente del desarrollo y de las entregas posteriores [9]. Ya que éste modelo se adecua con la secuencia del cronograma del trabajo terminal y con nuestra forma de trabajar realizando "incrementos" al sistema, es la razón por la que lo utilizaremos.



Este modelo involucra cinco fases principales en cada iteración las cuáles son:

- Comunicación: con esta etapa se inicia el incremento recopilando los requerimientos.
- Planeación: se estiman los tiempos de finalización, se realiza un itinerario conforme a éstos y se da un seguimiento a las anteriores fases.
- Modelado: se realiza el análisis de los requerimientos y se buscan posibles soluciones. También se realiza un diseño del módulo del incremento en cuestión.
- Desarrollo: durante esta fase se codifican las soluciones propuestas en el modelado siguiendo el diseño además de realizarse las pruebas debidas.
- Despliegue: se entrega el incremento, se realiza el soporte y se da una retroalimentación con los anteriores incrementos.

Los incrementos principales consisten en los siguientes módulos:

- Médico: registra al médico, realiza la autentificación del médico, gestiona el perfil del médico, realiza una relación de las recetas que ha expedido y firma digitalmente la receta.
- Farmacia: registra a la farmacia en el sistema, permite consultar las recetas a la farmacia y gestiona la receta con los datos de los medicamentos entregados.
- Receta: crea, consulta, modifica y cancela la receta, así como también genera un código QR que se incorporará a la receta y posteriormente se enviará un correo electrónico al paciente o en su defecto se imprimirá.
- Medicamento: gestión de medicamentos para prescribirse en las recetas médicas.

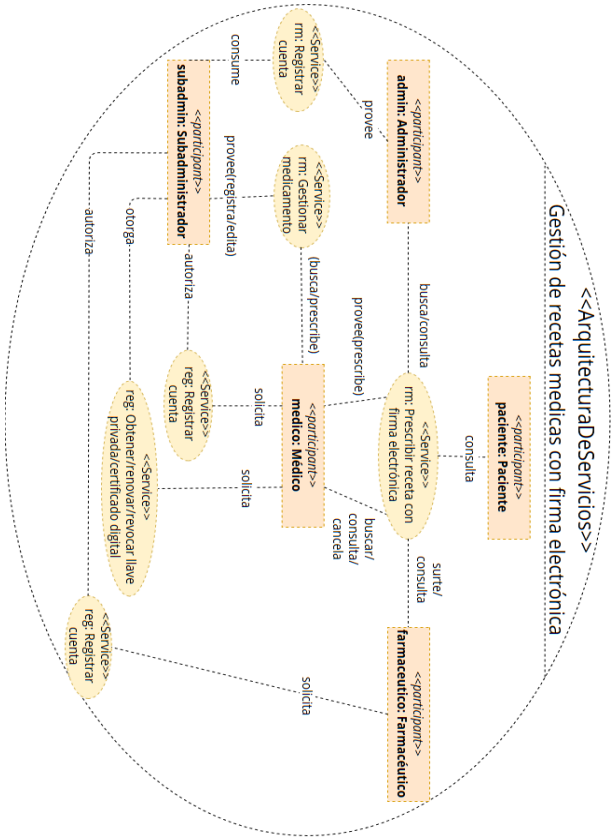
Éstos a la vez tendrán submódulos para realizar un desarrollo holístico del sistema.

La arquitectura orientada a servicios (SOA) es un estándar del sector de definición abierta, el cual presenta todos los procesos de negocio de un modo orientado a servicios. El objetivo de la arquitectura orientada a servicios es separar la lógica de integración de negocio de la implementación, para que el desarrollador de integración pueda centrarse en ensamblar una aplicación integrada en lugar de hacerlo en los detalles de la implementación. [10]

Los componentes de servicio también ofrecen la opción de permitir que el desarrollador añada la implementación en una etapa posterior al desarrollo del sistema. Finalmente

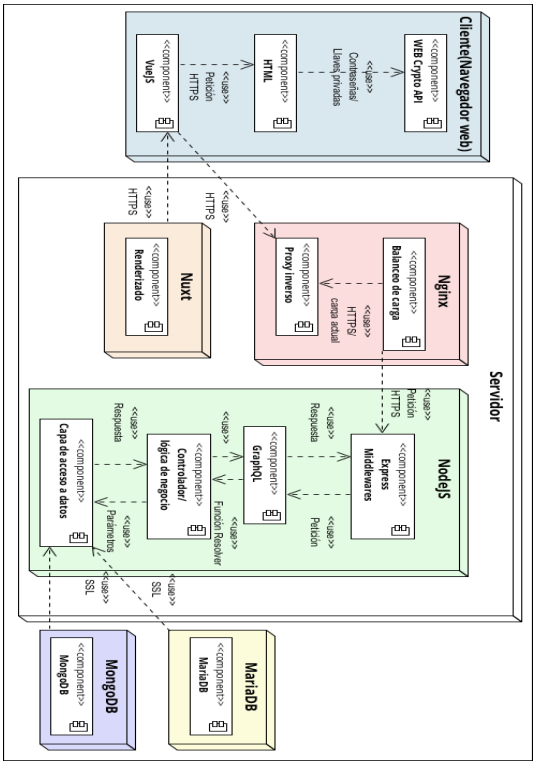
el propósito de usar la arquitectura orientada a servicios se resume en:

- Una arquitectura orientada a servicios permite reaccionar con agilidad ante los cambios delas condiciones de las empresas y beneficiarse de los mismos.
- Permite que las aplicaciones sean independientes de plataforma en las que están montadas.
- Un componente que accede a otro no necesita conocer las estructuras de datos, las llamadas a otros componentes, la gestión de transacciones, etc.
- Cualquier componente puede trabajar con otro, incluidos los componentes creados por entornos de desarrollo de proveedores diferentes.
- Los componentes son independientes de la ubicación.



La arquitectura de software está separada en los siguientes módulos, se puede apreciar que para brindar una mejor seguridad para nuestro cliente (el médico), se tienen que implementar tecnologías en el navegador web. También se puede observar el servidor está dividido en dos grandes entornos, los cuáles se encargan de proporcionar una interfaz gráfica al usuario, la que llamaremos Frontend. Y la que se encarga de la lógica del sistema: Backend. También podemos observar que en el Backend se encuentra las bases de datos, las cuáles van a alojar nuestros datos estructurados y no estructurados.

Siguiendo con el enfoque de nuestra base de datos, se decidió utilizar un gestor de bases de datos no estructurado llamado MongoDB para la gestión de recetas médicas. Ya que es más práctico darle éste enfoque para un análisis de Big Data. Los demás datos están modelados en el gestor



MariaDB el cuál es una bifurcación del proyecto MySQL.

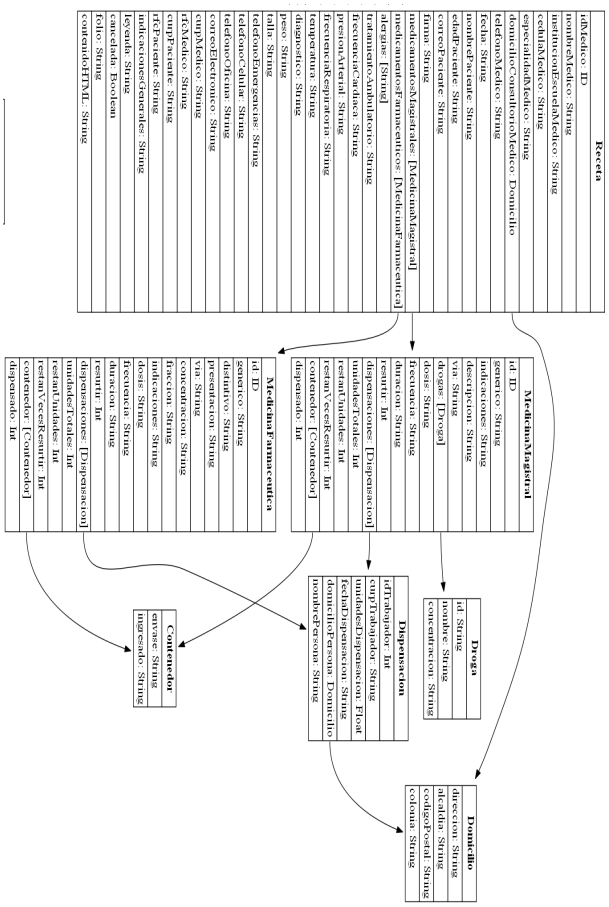
Para proveer los servicios criptográficos de no repudio, integridad de los datos y autenticación, se usará ECDSA configurado con la curva secp256r1 y como función hash SHA-256. Se eligió ECDSA por ofrecer ventajas sobre otros algoritmos de firma electrónica como RSA (ambos aprobados por el NIST en el Federal Information Processing Standards Publication 186: DigitalSignature Standard[11]), como [12]:

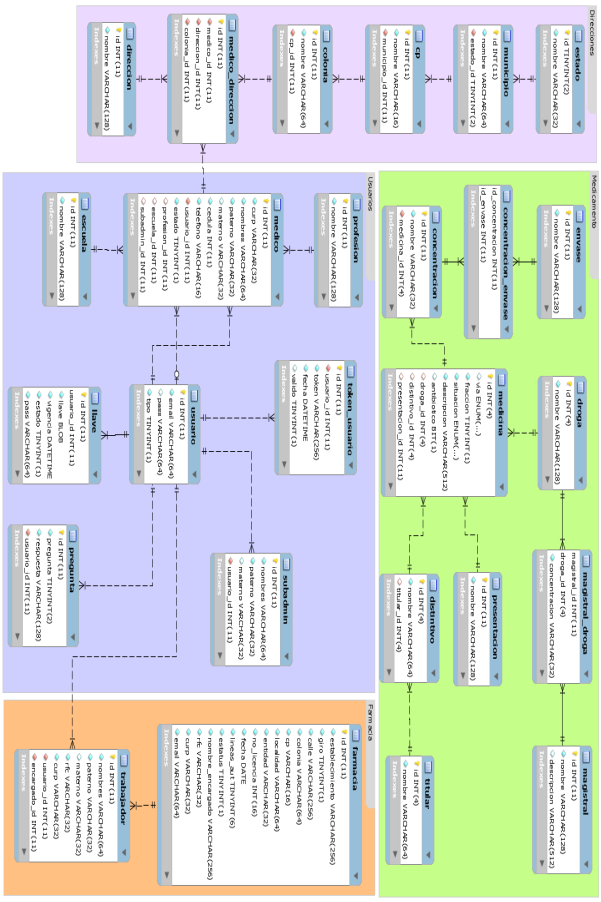
- El tamaño de llaves es menor en ECDSA: una llave con una longitud de 256 bits, ofrece un nivel de seguridad equivalente (128 bits) que una llave de RSA de 3072 bits. Así, el principal motivo para usar ECDSA es que se reduce el almacenamiento requerido para guardar las llaves públicas empleadas en la firma electrónica dentro del sistema (las cuales estarán contenidas en los certificados digitales de cada médico). Un aspecto que debe destacarse es el tamaño del certificado digital en ambos esquemas de firma digital, el cual esta directamente relacionado con el tamaño de llaves señalado en la tabla anterior, pues contiene la llave pública. En OpenSSL, un certificado digital con el formato X 509 con un nivel de

Cuadro I
TAMAÑO DE LLAVES DE ECDSA Y RSA

Nivel de seguridad (bits)	RSA (bits)	ECDSA (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Fuente: [12].





seguridad de 128 bits, en RSA ocupa 1590 bytes mientras que en ECDSA ocupa 709 bytes (cerca de la mitad).

- ECDSA ofrece un mejor rendimiento en los procesos de generación de llaves, y generación de firma (este último, cuando el tamaño de llaves es mayor a 256 bits), los cuales si bien no son los procesos mas comunes respecto a la firma electrónica dentro del sistema prototipo, cuyo rendimiento podría impactar en el rendimiento general del sistema, es importante mencionar que son los procesos en los que se emplea la llave privada del médico, la cual por seguridad de la misma se debe evitar su comunicación por cualquier medio aun cuando este sea seguro, y por ende, solo emplearse de lado del cliente (navegador web), cuyo rendimiento no se podrá garantizar pues dependerá directamente de las capacidades de la computadora que use el médico y personal de la autoridad certificadora.
- Respecto a RSA y DSA, en ECDSA la verificación de la firma es mas lenta, sin embargo, en este proceso, no se necesita emplear la llave privada del médico, por lo que esta tarea podría ejecutarse de lado del servidor, y en caso de que el rendimiento del servidor se vea degradado, delegar tal procesamiento al lado del cliente.

En el artículo de investigación *Performance Comparison of Elliptic Curve and RSA Digital Signatures* [13], se realizó una comperación de rendimiento entre ECDSA y RSA. Las pruebas fueron hechas en una computadora con procesador Intel P4 2.0 GHz con 512MB de RAM, implementadas en C++. En la generación y verificación de firma se usó un archivo de texto plano (.txt) de 100 KB. Los resultados son los siguientes:

Cuadro II
PRUEBA DE RENDIMIENTO CON RSA

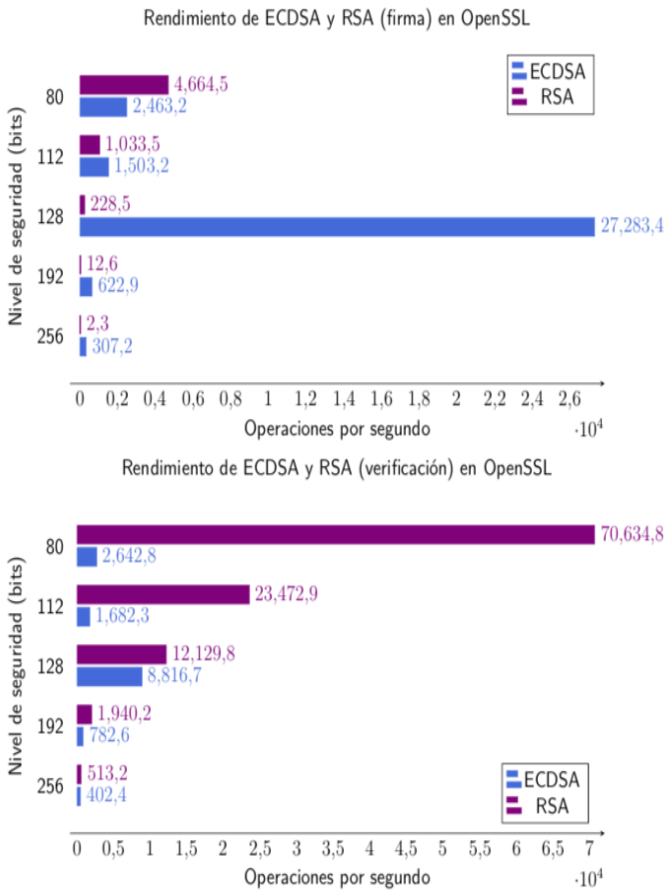
Nivel de seguridad (bits)	Generación de llave (s)	Firma (s)	Verificación (s)	Total (s)
80	0.16	0.01	0.01	0.2
112	7.47	0.15	0.01	7.6
128	9.8	0.21	0.01	10.1
192	133.9	1.53	0.01	135.4
256	679.06	9.2	0.03	688.3

Cuadro III
PRUEBA DE RENDIMIENTO CON ECDSA

Nivel de seguridad (bits)	Generación de llave (s)	Firma (s)	Verificación (s)	Total (s)
80	0.08	0.15	0.23	0.46
112	0.18	0.34	0.51	1.03
128	0.27	0.59	0.86	1.72
192	0.64	1.18	1.8	3.62
256	1.44	3.07	4.53	9.04

Cabe destacar que la generación de llaves en ECDSA crece linealmente con el tamaño de llaves mientras que con RSA el crecimiento es exponencial.

Adicionalmente, se realizó una prueba por parte del equipo, con la herramienta OpenSSL versión 1.1.1a, a través del comando *openssl speed* el cual permite medir el rendimiento de los algoritmos criptográficos que soporta. Las pruebas fueron hechas en una computadora con procesador Intel Core i7 3.2 GHz con 16 GB de RAM en el sistema operativo Windows 10. Cabe mencionar que OpenSSL no muestra información sobre el rendimiento en generación de llaves (proceso que será el menos común dentro del sistema). Los resultados son:



Otra prueba realizada fue con la Web Cryptography API, firmando y verificado una cadena de texto de 4.64 KB bytes (una estimación a un archivo JSON cuya estructura es la de una receta médica con los datos requeridos por la Comisión Federal para la Protección contra Riesgos Sanitarios). Se muestra el promedio que de una muestra de diez ejecuciones en Chrome con una computadora con procesador Core i7 con 16GB de RAM:

Cuadro IV
PRUEBA DE RENDIMIENTO EN WEB CRYPTO API CON ECDSA. (NIVEL DE SEGURIDAD DE 128 BITS)

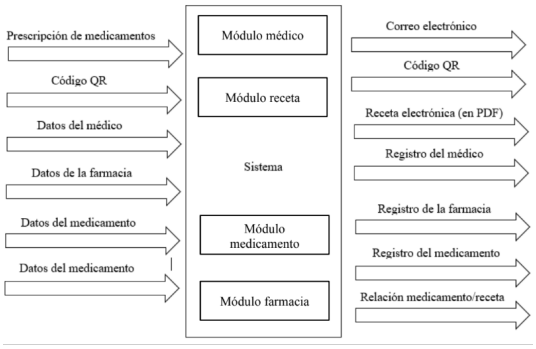
Algoritmo	Firma (ms)	Verificación (ms)	Total (ms)
ECDSA	2.545	1.841	4.386
RSA	4.785	1.916	6.701

III. RESULTADOS

Los productos finales serán un sistema web (de manera evidente a través de las páginas web), la documentación del proyecto, el repositorio con los archivos de éste y el manual de usuario.

Los resultados que se producen en el sistema se pueden ver junto en conjunto con sus entradas.

- Prescripción del medicamento: cuando el médico prescriba medicamentos en una receta tendrá posibles salidas como consecuencia. La primera es el correo electrónico: a través de éste se enviará al paciente un PDF que contará con la receta médica y dentro de ésta el código QR, el cual se mostrará en la farmacia para que puedan consultar la receta previamente firmada por el médico, de manera digital. También está la opción de permitir imprimir la receta.
- Código QR: a través del sistema se podrá escanear el código QR del paciente (A través de un dispositivo no incluido) para que el personal de la farmacia obtenga la receta digital, y así no exista necesidad de gastar papel para su distribución.
- Datos del paciente: Se registrará al paciente para llevar una relación de las recetas medicadas y entregadas.
- Datos de la farmacia: para tener una mejor gestión de las farmacias que distribuyen los medicamentos se tendrá un registro con sus datos más relevantes.
- Datos del medicamento: si se quiere registrar un nuevo medicamento que no esté en nuestra base de datos se procederá a llenar un registro con su información.
- Datos del medicamento otorgado: cuando el farmacéutico entregue un fármaco al paciente deberá especificar la cantidad que se le entregó, ya que puede haber la posibilidad de solicitar sólo alguna parte del tratamiento.



IV. CONCLUSIONES

Se concluye dado el trabajo realizado documentado que los módulos desarrollados cumplen con la funcionalidad especificada en los requerimientos funcionales, además que se realizó una implementación aún de manera local con los equipos de cómputo personales.

La curva de aprendizaje de las tecnologías empleadas (principalmente Javascript) en el desarrollo del sistema contribuyó

bastante el cumplir con el cronograma planteado. Existieron complicaciones a lo largo del desarrollo principalmente en el modulo dedicado a generar la infraestructura de llave privada debido a que se tuvo que buscar una librería más adecuada que permitiera emplearla del lado del cliente pero que ofreciera interoperabilidad con el servidor, la cual fue la Web Cryptography API. El acceso a información oficial (de forma automatizada) contribuiría en gran medida la viabilidad de implementar este sistema. Se cumplieron los objetivos específicos. Se realizaron mejoras sobre el desarrollo para mejorar rendimiento y proveer funcionalidad en más navegadores web. Se logró una configuración correcta para establecer una interoperabilidad entre el navegador y el servidor en el caso de la generación del par de llaves.

El propósito de éste trabajo terminal es crear un sistema cuya arquitectura permita el no repudio y la integridad de las recetas médicas. Sin embargo, incorporar éste modulo a otros sistemas ya implementados tanto en el sector privado como en el público, como sistemas de ventas, inventarios, citas, entre otros, en un ambiente productivo tendría cierta complejidad para los técnicos e ingenieros encargados de dichos sistemas dependiendo de las tecnologías y protocolos de red que éstos usen.

Por lo que el trabajo a futuro se puede considerar como un sistema holístico que contemple todos los posibles ámbitos involucrados en el área médica, contemplando en dichos procesos al paciente y farmacia. De hecho en éste trabajo terminal en un planteamiento inicial se buscó desarrollar la dispensación de las recetas médicas por parte de las farmacias, sin embargo, al contemplar que implica un monopolio de dispensación; un caso puntual de ello es un lugar aislado tecnológicamente pero que cuente con servicio médico, en éste caso puede que el paciente vaya a una farmacia que no cuente con el sistema, o en todo caso una que sí cuente con el sistema y ésta receta no esté firmada por el éste, pero el doctor si dio consentimiento de lo prescrito en ésta. Lo que se puede hacer en éste caso es que el sistema tenga en consideración las recetas médicas que no se expidan para éste, sin embargo, esto conlleva muchos problemas aunque no es imposible de realizar.

Para complementar el módulo de farmacia, que puede ser el que más está relacionado con el sistema actual, se puede añadir el submódulo del inventario, el cuál el responsable de la farmacia podrá ingresar la cantidad de los medicamentos con los que cuenta actualmente y que se actualicé en conjunto al evento de dispensar receta; la dispensación de recetas médicas debe poder realizar parcial o totalmente considerando todos los posibles envases externos e internos para la medicina, sin embargo, para poder conseguir ésta información es muy difícil, ya que incluso en el portal de la COFEPRIS, no brindan la información en el formato adecuado o incluso existe una ausencia de información.

Otro módulo importante que se puede considerar en un futuro sería un expediente digital, aunque en el sistema pareciera que el paciente si tiene un expediente, no es así, sólo es el conjunto de recetas médicas que ha expedido los médicos a un paciente. Sin embargo, hay más puntos importantes que considerar para éste actor, por ejemplo: si ha tenido enfermedades antes, si ha sido curado de éstas, si están recesivas, si necesita estudios de laboratorio, si necesita alguna operación en el quirófano. Así como almacenar sus radiografías, ecografías, ultrasonidos, entre otros.

También se puede agregar a la arquitectura una infraestructura de lector de códigos QR a través del celular; actualmente solamente cuenta con una arquitectura que permita leer el código QR o de barras a través de un lector de dichos códigos lo cual resultaría muy útil para las farmacias que no cuenten con el presupuesto o viabilidad para comprar éste hardware.

V. Reconocimientos

Los autores agradecen a la Escuela Superior de Cómputo del Instituto Politécnico Nacional por el apoyo recibido y las facilidades otorgadas para el desarrollo del presente trabajo terminal, en particular a los profesores que participaron en calidad de directores y sinodales del mismo por el tiempo, atención y disposición para buscar de la mejor manera que este trabajo contribuyera a nuestra formación.

Referencias

[1] COFEPRIS, *Guía para comercialización de medicamentos controlados en farmacias*, https://www.gob.mx/cms/uploads/attachment/file/305082/Gu_aREyPF_mayo2017.pdf, 2017.

[2] Sandoval Morales, Miguel, *Notas sobre Criptografía*, https://www.tamps.cinvestav.mx/ mmorales/documents/Criptograf.pdf, 2013.

[3] Rello, Maricarmen, *Alertan sobre peligros de la automedicación*, http://www.milenio.com/estados/alertan-sobre-peligros-de-la-automedicacion, 2017

[4] OMS, *Antibiotic Resistance*, : http://www.who.int/news-room/fact-sheets/detail/antibiotic-resistance, 2018.

[5] INEGI, *Estadísticas a propósito de la industria farmacéutica y sus proveedores*, http://codigof.mx/wp-content/uploads/2017/07/Farma11julio2017_x.pdf, 2007

[6] Presidencia de la República Mexicana. *Reglamento de Insumo para la Salud*, http://www.salud.gob.mx/unidades/cdi/nom/compi/ris.html, 2014.

[7] Sánchez Onofre, Miguel. *Prescrypto lleva el blockchain a las recetas médicas electrónicas*, https://www.eleconomista.com.mx/tecnologia/Prescrypto-lleva-el-blockchain-a-las-recetasmedicas-electronicas-20170705-0073.html, 2017.

[8] Sistema Integral Médico Farmacéutico. *trabajo terminal2012-A021*, https://tesis.ipn.mx/handle/123456789/20037, 2013.

[9] Arias, Angel. *Aprende sobre la Ingeniería de Software: 2da Edición*, 2015.

[10] IBM Knowledge Center. *Arquitectura orientada a servicios*, https://www.ibm.com/support/knowledgecenter/es/SSFTN5_8.5.7/com.ibm.wbpm.wid.main.doc/prodoverview/topics/csoa.html 2017.

[11] NIST. *Federal information processing standards publication: Digital signature standard (DSS)*, http://dx.doi.org/10.6028/NIST.FIPS.186-4 2013.

[12] A. I. Ali. *Comparison and evaluation of digital signature schemes employed in ndn network.*, https://arxiv.org/ftp/arxiv/papers/1508/1508.00184.pdf 2015.

[13] Sharon Levy. *Performance and Security of ECDSA.*, https://pdfs.semanticscholar.org/7b20/33159cc8ef7f50619b1592eeea639e2807a3.pdf"2018.