Universidad de las Ciencias Informáticas

Facultad 7

Título: Desarrollo de funcionalidades para fortalecer la seguridad del módulo Admisión aplicando los perfiles de seguridad IHE.

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.

Autor(es): Yasmín Galvez Cuza

Edder Monzón Rodríguez

Tutor: Ing. Juan Manuel García Orduñez

Cotutor: Ing. Juan Manuel Rey Alvarez

La Habana, Junio 2013

"Año 55 de la Revolución."

Pensamiento.

¨…lo fundamental es hacer algo nuevo cada día y luego perfeccionar lo que se ha hecho el día anterior…¨

Ernesto Guevara de la Serna

Declaración de autoría

Declaramos ser los únicos autores del presente trabajo y autorizamos al Departamento de Gestión Hospitalaria (GEHOS) de la Universidad de las Ciencias Informáticas (UCI); así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ___ días del mes de _____ del año 2013.

_____ _____

Yasmín Galvez Cuza

Autor

Edder Monzón Rodríguez

Autor

_____ _____

Ing. Juan Manuel García Orduñez

Tutor

Ing. Juan Manuel Rey Alvarez

Cotutor

Datos de los contactos

Datos del tutor

Nombre: Ing. Juan Manuel García Orduñez

Ingeniero en Ciencias Informáticas, graduado en la Universidad de las Ciencias Informáticas en el año 2009. Posee la categoría docente de Profesor Instructor. Ha impartido la asignatura de programación 3. Además participado en varios proyectos de desarrollo vinculados al perfil de salud. Actualmente labora en el Departamento de Sistemas de Gestión Hospitalaria del Centro de Informática Médica (CESIM), desempeñándose como líder de desarrollo del módulo Emergencias.

Correo electrónico: jmgarcia@uci.cu

Datos del cotutor

Nombre: Ing. Juan Manuel Rey Alvarez

Recién Graduado en Adiestramiento, graduado en el año 2011 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Ha participado en proyectos de desarrollo vinculados al perfil de salud. Se desempeña como líder de desarrollo del módulo Archivo.

Correo electrónico: jmrey@uci.cu

Dedicatoria.

Yasmín Galvez Cuza.

Dedico este trabajo a mi familia, quienes me ayudaron con su apoyo incondicional a estar más cerca de mis metas profesionales. En especial a mis padres por su infinito amor y comprensión. A mi hermano y mi tía quienes sé que me quieren mucho y me llenaron de esperanza en todo momento para concluir mis estudios.

Agradecimientos

Gracias a nuestros profesores por su entrega durante los cinco largos años de estudio en la UCI. En especial a Suleydis Suarez y Yoandy Gonzalez por su colaboración en la confección de este trabajo de diploma. A nuestras familias por el esfuerzo que han realizado para darnos lo mejor y educarnos como personas de bien. Especial agradecimiento a nuestros padres por dedicarnos todo su amor. A nuestros compañeros por ayudarnos, de una forma u otra, a enfrentar todas las vicisitudes que emanan de esta carrera llena de retos y enseñanzas. A nuestros tutores por su apoyo en la revisión y perfeccionamiento de este trabajo de diploma.

Resumen

El creciente uso de las Tecnologías de la Información y las Comunicaciones (TIC) en el sector de la salud ha potenciado el mejoramiento de los servicios sanitarios, fundamentalmente con el empleo de los Sistemas de Información Sanitarios (SIS). Estos han facilitado el acceso e intercambio de los datos clínicos de los pacientes. Este hecho hace necesaria la implantación de medidas de seguridad con el objetivo de garantizar la integridad, confidencialidad y disponibilidad de la información gestionada. Actualmente el sistema de información hospitalaria alas HIS presenta algunas brechas de seguridad informática con potencialidad a convertirse en ataques informáticos, debido a que la información del paciente se intercambia y almacena de forma legible, lo que se presta a la intervención de agentes externos. Además no existen mecanismos que certifiquen la autenticidad de los diferentes nodos que intercambian datos por la red. Una vía para la solución de estos problemas es el desarrollo de un componente de seguridad para el módulo Admisión guiándose por los perfiles de seguridad propuesto por el conjunto de estándares Integrando Empresas Sanitarias (IHE, por sus siglas en inglés). El mismo garantizará la protección (confidencialidad) de los datos almacenados, y la seguridad del flujo de información que se transmite por la red. Su desarrollo está basado en tecnologías libres o de código abierto. Se utilizó Java como lenguaje de programación, PostgreSQL como gestor de base de datos, el framework Hibernate para el acceso a datos, los protocolos SSL y HTTPS para la transferencia segura de información y el algoritmo simétrico Estándar Avanzado de Cifrado (AES, por sus siglas en inglés) para la encriptación de los datos.

Palabras clave:

AES, CDA, confidencialidad, HTTPS, integridad, IHE, SSL, seguridad, seguridad informática y TIC.

Índice.

Declaración de autoría II

Datos de los contactos III

Dedicatoria IV

Agradecimientos V

Resumen VI

Introducción 1

Capítulo 1: Fundamentación teórica del desarrollo del componente de seguridad 6

1.1 Antecedentes de la seguridad de los sistemas sanitarios 6

1.2 Perfiles de seguridad de IHE y su uso práctico 7

1.3 Herramientas para la implementación de un RDE 9

1.4 Algoritmos de encriptación simétricos 10

1.5 Metodología, tecnologías y herramientas 13

1.5.1 SSL 13

1.5.2 HTTPS 14

1.5.3 Enterprise JavaBeans (EJB) 14

1.5.4 Java Persistence API (JPA) 15

1.5.5 Hibernate 15

1.5.6 Java 1.6 15

1.5.7 Java 16

1.5.8 Drools 16

1.5.9 Eclipse 3.2.4 16

1.5.10 Apache Commons 16

1.5.11 Open SSL 17

1.5.12 Keytool 17
1.5.13 VSFTPD 17
1.5.14 Pgadmin 3 17
1.5.15 Servidor de aplicaciones Jboss 4.2 17
1.5.16 PostgreSQL Server 9.1 17
1.5.17 RUP 18
1.5.18 Visual Paradigm 6.4 18
Capítulo 2: Características de la solución propuesta 20
2.1 Flujo actual de los procesos 20
2.2 Modelo de dominio 20
2.2.1 Conceptos fundamentales del dominio 20
2.2.2 Diagrama del modelo de dominio 21
2.4 Propuesta de solución 22
2.3 Especificación de los requerimientos de software 22
2.3.1 Requisitos funcionales 22
2.3.2 Requisitos no funcionales 23
2.4 Modelo de casos de uso de la solución propuesta 25
2.4.1 Descripción textual de los casos de uso 25
Capítulo 3: Diseño y desarrollo de la solución propuesta 32
3.1 Descripción de la arquitectura 32
3.2 Modelo de diseño 33
3.2.1 Diagrama de clases del diseño 33
3.2.2 Diagrama de paquetes 33
3.2.3 Diagrama de clases del diseño pertenecientes al paquete Seguridad 34
3.2.4 Descripción de las clases 35

3.2.5 Diagrama de clases del diseño pertenecientes al paquete Capturador 36
3.2.6 Diagrama de clases del diseño pertenecientes al paquete Cifrado 37
3.2.7 Diagrama de clases del diseño pertenecientes al paquete Gestión CDA 37
3.3 Implementación 38
3.3.1 Diagrama de componentes 38
3.3.2 Estrategias de codificación. Estándares y estilos a utilizar 38
3.3.3 Configuración 39
3.3.4 Diagrama de despliegue 41
Conclusiones 43
Recomendaciones 44
Referencia bibliográfica 45
Bibliografía 45
Anexos 51
Glosario de término 58

Índice de tablas.

Tabla 1 Características de los algoritmos DES, IDEA y AES. 12

Tabla 2 CU guardar en FTP. 25

Tabla 3 CU descargar del FTP. 26

Tabla 4 CU capturar evento persistir. 27

Tabla 5 CU capturar evento cargar. 28

Tabla 6 CU encriptar ficheros. 29

Tabla 7 CU desencriptar datos. 30

Tabla 8 Actores y transacciones del perfil ATNA. 51

Tabla 10 Actores y transacciones del perfil EUA. 53

Tabla 11 Actores y transacciones del perfil XDS. 55

Índice de ilustraciones.

Ilustración 1 Diagrama del modelo de dominio. 21
Ilustración 2 Modelo de caso de uso de la solución propuesta. 25
Ilustración 3 Arquitectura de la solución propuesta de software. 33
Ilustración 4 Diagramas de paquetes. 34
Ilustración 5 Diagrama de clases del diseño pertenecientes al paquete Seguridad. 34
Ilustración 6 Clase OyenteLoad. 35
Ilustración 7 Clase OyentePersis. 35
Ilustración 8 Clase AESEncriptacion. 35
Ilustración 9 Clase GestionCDA. 36
Ilustración 10 Diagrama de clases del diseño pertenecientes al paquete Capturador. 37
Ilustración 11 Diagrama de clases del diseño pertenecientes al paquete Cifrado. 37
Ilustración 12 Diagrama de clases del diseño pertenecientes al paquete Gestión CDA. 37
Ilustración 13 Diagrama de componentes. 38
Ilustración 14 Generación de un certificado digital desde la consola. 40
Ilustración 15 Configuración del archivo server.xml. 40
Ilustración 16 Diagrama de despliegue. 42
Ilustración 17 Proceso de encriptación con llave simétrica. 57

Introducción

La seguridad informática es el conjunto de medidas administrativas, organizativas, físicas, técnicas, legales y educativas dirigidas a prevenir, detectar y responder a las acciones que puedan poner en riesgo la confidencialidad, integridad y disponibilidad de la información que se procesa, intercambia, reproduce o conserva, por medio de las tecnologías de la información. (1) Así mismo la seguridad de la información es un factor fundamental en cualquier institución, donde la pérdida, desvío o mala manipulación de los datos pueden ocasionar daños económicos y sociales irreparables. Por tanto se requiere una buena administración y control de la información, así como una adecuada seguridad acorde a los intereses exigidos.

El creciente uso de las Tecnologías de la Información y las Comunicaciones (TIC), en gran parte de las esferas del desarrollo humano, ha tenido innumerables beneficios en los procesos de producción. Al mismo tiempo, existen riesgos que pueden amenazar la seguridad de los datos gestionados por las instituciones que hacen uso de las mismas, por lo que, se hace imprescindible conocer e implementar las medidas de prevención que ofrece la seguridad informática, para evitar que dicha información sea comprometida.

Con respecto a los riegos mencionados la salud pública es una de las ramas más vulnerables, pues la documentación clínica generada, asociada al paciente, es altamente sensible. Por ello existen normas y estándares que definen los requisitos mínimos que deben cumplir los sistemas en este sector para garantizar niveles aceptables de protección.

Entre ellos, está la recomendación de uso de estándares como IHE. Este es aplicado particularmente en el ámbito médico para garantizar la integración de los Sistemas de Información Sanitarios (SIS). Su empleo en este sentido pretende facilitar el acceso por parte del personal de salud a la información de los pacientes y mejorar la calidad y costo de los servicios sanitarios. El IHE además, propone perfiles de integración, que se localizan dentro de su marco técnico, para el cumplimiento de políticas de seguridad informática en las aplicaciones médicas.

Uno de los tipos de SIS más extendidos en el ámbito sanitario es el de los Sistemas de Información Hospitalaria (HIS, por sus siglas en inglés). Estos además permiten la recolección, almacenamiento, recuperación y transmisión de los datos generados durante la atención a los pacientes en las instituciones hospitalarias; así como la generación de reportes estadísticos fundamentales para la toma de decisiones clínicas, administrativas y epidemiológicas.

La Universidad de Ciencias Informáticas (UCI), se encuentra actualmente desarrollando el sistema de información hospitalaria alas HIS en el Centro de Informática Médica (CESIM). El alas HIS es una solución integral para la gestión hospitalaria. Además tiene como mayor potencialidad la posibilidad de gestionar toda la documentación e imágenes que se generen en torno a un paciente en una única Historia Clínica Electrónica (HCE). La misma es almacenada en documentos que cumplen las especificaciones del estándar Arquitectura de Documentos Clínicos (CDA, por sus siglas en inglés), con el objetivo de facilitar la interoperabilidad con futuros sistemas que intercambien información con este.

El sistema alas HIS permite gestionar estadísticas más rápidas y eficientes de los casos de estudios médicos. Está compuesto por los módulos Citas, Consulta Externa, Emergencias, Hospitalización, Laboratorio, Salud Ocupacional, Visor de Historia Clínica (en adelante Visor HC), Admisión entre otros que, representan las distintas áreas que conforman un centro asistencial. Específicamente el módulo Admisión, entre otras funcionalidades, facilita la gestión de HCE de los pacientes registrados en el sistema.

Las HCE gestionadas en el alas HIS, presentan algunas brechas de seguridad informática que pueden llegar a convertirse en ataques informáticos. Una de ellas es que los datos manipulados viajan a través de la red en texto plano, desde el navegador web hacia el servidor de aplicaciones, así como desde este hasta el gestor de base de datos. Esto constituye una fisura significativa, que pudiese dar lugar a que un atacante potencial intercepte el tráfico de datos, obteniendo información que pueda ser analizada, distribuida a terceros, distorsionada o usada con otros fines.

Otra vulnerabilidad está presente en el gestor de base de datos. Este almacena la información de forma legible y la misma puede ser accedida por el administrador, el cual posee todos los permisos sobre ésta. Lo antes descrito posibilita asociar con relativa facilidad, la información de identificación de un paciente con los resultados clínicos del mismo. Dicho administrador puede ser objeto de soborno o amenaza pudiendo llegar a comprometer los datos almacenados.

El servidor de aplicaciones además de las funciones que realiza, guarda localmente los ficheros CDA gestionados por el módulo Visor de Historia Clínica. Aunque dichos documentos son firmados digitalmente por el personal de salud al emitirlos, hecho que garantiza su integridad, autenticidad y no repudio, su confidencialidad no es total pues están escritos de forma explícita. Dada la veracidad que le imprime el uso de la firma digital en dichos archivos, su divulgación pudiera traer consigo complicaciones en asuntos médico-legales.

Actualmente el alas HIS no cuenta con mecanismos que certifiquen la autenticidad de los diferentes nodos que intercambian datos por la red. Esto pudiese causar que los usuarios sean redirigidos a sistemas falsos, convirtiéndose en víctimas de robo de cuenta y/o información personal, estadística o médica. En este caso no es posible garantizar que la información consultada sea generada por agentes autorizados.

Por todo lo anteriormente descrito, se determina como **problema a resolver**: ¿Cómo incrementar la confidencialidad de la información y la autenticidad de los agentes gestores de la misma, en el sistema alas HIS? El **objeto de estudio** de la presente investigación lo constituyen los perfiles de integración de IHE y el **campo de acción** se centra específicamente en los perfiles de seguridad propuestos por IHE.

Se define para la investigación el siguiente **objetivo general**: desarrollar funcionalidades que permitan fortalecer la confidencialidad de la información que se intercambia y almacena en el sistema alas HIS, aplicando los perfiles de seguridad de IHE. Para dar cumplimiento al objetivo general, se plantean las siguientes **tareas de investigación**:

- Estudiar las tendencias actuales de la seguridad informática en aplicaciones web.
- Investigar los perfiles de seguridad que propone IHE.
- Analizar los algoritmos de encriptación simétricos.
- Estudiar la forma de implementar un Repositorio de Documento Electrónico.
- Asimilar las tecnologías y arquitectura definidas para el sistema de información hospitalaria alas HIS.
- Desarrollar las funcionalidades del componente de seguridad para el sistema de información hospitalaria alas HIS.

Para llevar a cabo la investigación propuesta se han utilizado los siguientes métodos científicos de investigación:

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 más 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. Estos

métodos permiten el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada, su relación con otros fenómenos, así como su aislamiento como objeto estudiado. (2) Siendo esta una de las razones fundamentales por la cual son utilizados en la presente investigación. Los siguientes métodos teóricos sustentan la investigación.

- Histórico-Lógico: Su empleo permitió el desarrollo evolutivo y coherente en el estudio de los patrones de diseño, herramientas y sistemas para el desarrollo de los artefactos que proponen los flujos estudiados.
- Analítico-Sintético: Permitió integrar y descomponer el conocimiento, descubriendo las relaciones para la utilización de artefactos propuestos, determinando los aspectos esenciales y arribando a conclusiones prácticas y teóricas.
- Inductivo-deductivo: Permitió pasar de lo particular a lo general y viceversa favoreciendo objetivamente el enlace que se establece en la realidad entre lo singular y lo general ya que ambas se complementan mutuamente en el proceso de desarrollo científico. (2)
- Modelación: Pues se crean abstracciones que explican la realidad, por ejemplo, todos los modelos y diagramas presentados. (2)En la presente investigación se utiliza este método científico para la representación de: los modelos de casos de uso de la solución propuesta, de requerimientos y de despliegue.

Métodos Empíricos:

Como parte de la investigación es necesario determinar el método de recolección de datos y tipo de instrumento que se utilizará, por lo cual deberán tomarse en cuenta especialmente los objetivos y las categorías del estudio expuestos con anterioridad.

- Método Entrevista: Es una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o características personales del entrevistado, puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación. (2) Para la presente investigación se utilizó el método de la entrevista no estructurada para la recolección de información, proporcionando los datos necesarios y de interés para las posteriores etapas de la misma.

Una vez concluidas la investigación y el desarrollo de la propuesta de solución se espera obtener beneficios como:

- Aumentar la confidencialidad de la información almacenada del paciente.

- Garantizar la trasmisión segura de datos médicos mediante técnicas de cifrados.
- Brindar certificados autofirmados para avala la autenticidad de los agentes gestores.
- Contar con un Repositorio de Documentos Electrónicos para almacenar las HCE de todos los episodios médicos realizados como parte de la atención de salud que recibe el paciente.

El presente documento se encuentra estructurado en tres capítulos, el primero de ellos, **capítulo 1: Fundamentación teórica del desarrollo del componente de seguridad,** donde se realiza un estudio preliminar de sistemas y funcionalidades que puedan dar respuesta al problema planteado. Igualmente muestra las tecnologías, metodologías y herramientas que fueron utilizadas en el desarrollo de la solución propuesta.

Seguidamente el **capítulo 2: Características de la solución propuesta**, contiene el flujo actual de los procesos, el modelo de dominio, la propuesta de solución, los requisitos de software y el modelo de caso de uso. Por último el **capítulo 3: Diseño y desarrollo de la solución propuesta**, se centra en la modelación detallada y la construcción de la estructura del componente. Además se implementan las clases y subsistemas en términos de componentes de la solución propuesta.

Capítulo 1: Fundamentación teórica del desarrollo del componente de seguridad

El presente capítulo aborda brevemente algunos sistemas sanitarios que siguen las especificaciones de los perfiles de IHE, así como ejemplos de algoritmos de encriptación y Repositorios de Documentos Electrónicos. Estos pueden ayudar a la solución de las brechas de seguridad que presenta el módulo Admisión del sistema alas HIS. Además, se realiza una descripción de las tecnologías, metodologías y herramientas de software con las que se proponen llevar a cabo el proceso de desarrollo de dicha solución.

1.1 Antecedentes de la seguridad de los sistemas sanitarios

La consolidación de Internet como medio de interconexión y comunicación mundial, la aparición de la Informática y el uso masivo de las comunicaciones digitales han traído consigo un número creciente de problemas de seguridad. Cada vez se hace mayor el número de atacantes que adquieren habilidades más especializadas, convirtiéndose, por tanto, en una amenaza preocupante. Este hecho trae aparejado una creciente necesidad de implantar mecanismos con la finalidad de proteger o reducir al mínimo los riegos de la seguridad informática.

La seguridad de la información es el conjunto de protocolos y mecanismos que aseguren que la comunicación entre los sistemas, esté libre de intrusos. (3) Los sistemas que procesen, almacenen, recuperen o transmitan los datos deben preservarlos de modo que su contenido no sea modificado o conocido por personas no autorizadas y al mismo tiempo se garantice que estén disponibles cuando sean necesarios. Si alguno de estos pilares fallara dicha información dejaría de ser segura.

Hasta la aparición y difusión del uso de los sistemas informáticos, los datos de interés de una organización quedaban registrados en formato duro; lo que acarreaba problemas para su transporte, acceso, procesamiento y almacenamiento. Este último precisaba la utilización de grandes espacios y condiciones especiales de los mismos, imprescindibles para lograr la conservación de la documentación por períodos de tiempo prolongado.

Los sistemas informáticos resuelven las desventajas del almacenamiento de la información en formato duro, permitiendo entre otras ventajas, digitalizar grandes volúmenes de información reduciendo el espacio ocupado, pero sobre todo, facilitando su análisis y gestión. Un ejemplo de ello, lo constituyen los HIS, que son capaces de gestionar las diferentes áreas de un hospital y permiten la interrelación de todos los datos de un modo eficiente y sencillo, para facilitar la eficaz gestión de un centro hospitalario.

(4) Los mismos proporcionan beneficios tales como: organización en los flujos de trabajo de la institución de salud, reducción de tiempo de espera del paciente, facilidades en la toma de decisiones, entre otras.

Al mismo tiempo, aparecen vulnerabilidades asociadas a estas facilidades. El acceso a la información a través de redes informáticas brinda inmediatez pero además, en cierto sentido, anonimato e impunidad para realizar actos vandálicos en contra de los repositorios y proveedores de la misma. No menos significativo resulta el hecho de la facilidad de copia, replicación y transmisión que caracteriza a los archivos digitales. Esta característica eleva las probabilidades de que dichos ficheros lleguen a manos de terceros no autorizados, comprometiéndose la confidencialidad de los datos gestionados por los sistemas informáticos.

Todas estas vulnerabilidades pueden poner en riesgo la seguridad de la información dentro de cualquier institución sanitaria. La relevancia de estos peligros está estrechamente relacionada con la importancia de los datos gestionados, así como el impacto que produciría su divulgación.

En este sentido los sistemas del ámbito sanitario son reservorios de datos muy sensibles. Esto ha motivado que en numerosos países se establezcan normativas para el manejo de la información clínica de los pacientes, partiendo del principio de que la privacidad de la HCE constituye un derecho de los mismos. Por todos estos elementos IHE propone varios de sus perfiles de integración dirigidos específicamente a garantizar niveles de seguridad elevados para los sistemas de gestión en el sector de la salud.

1.2 Perfiles de seguridad de IHE y su uso práctico

IHE es una iniciativa de profesionales de la sanidad y empresas proveedoras cuyo objetivo es mejorar la comunicación entre los sistemas de información que se utilizan en la atención al paciente. El mismo en el marco técnico de infraestructura define perfiles de integración. Este utiliza estándares ya existentes para la unificación de sistemas, de manera que proporcionen una interoperabilidad efectiva y un flujo de trabajo eficiente. IHE permite alcanzar el nivel de integración exigible en la era de la historia clínica electrónica.

Cada Perfil de Integración IHE describe una necesidad clínica de integración de sistemas y la solución para llevarla a cabo. Define también los componentes funcionales, a los llamados actores IHE, y específica, con el mayor grado de detalle posible, las transacciones que cada actor deberá llevar a cabo, basadas siempre en estándares existentes. (5) Los perfiles de integración propuestos por IHE que aportarán a la seguridad del sistema alas HIS son:

Autenticación de Usuarios (6) (**EUA**, por sus siglas en inglés): este perfil establece como debe realizarse la autenticación dentro de una institución mediante la utilización de certificados digitales tanto para la verificación de identidad entre subsistemas como entre clientes y servidores. (7)

Registro para Auditoría y Autenticación de Nodos (**ATNA**, por sus siglas en inglés): este proporciona una infraestructura básica para implementar las políticas de seguridad de cada organización. Describe una forma de autenticación de los actores (sistemas) usando certificados, y cómo transmitir eventos relacionados con la información personal a un repositorio para auditoría. (6)

Marca de Tiempo Consistente (**CT**, por sus siglas en inglés): garantiza que los relojes y marcas de tiempo de todos los equipos en una red estén sincronizados (error menor de 1 sec.). (6)

Directorio de Personal (6) (**PWP**, por sus siglas en inglés): establece como debe realizarse el registro de credenciales de los usuarios. (7)

Compartir Documentos entre Organizaciones (6) (**XDS**, por sus siglas en inglés): proporciona soluciones para el intercambio de documentos entre instituciones afiliadas. Permite a varias organizaciones de prestación de asistencia sanitaria que pertenezcan a un dominio de afinidad, cooperar en el cuidado de un paciente mediante el intercambio de historias clínicas en forma de documentos a medida que avanzan las actividades de atención a sus pacientes. (7)

Consentimiento de Privacidad Básica del Paciente (**BPPC**, por sus siglas en inglés): establece el registro de consentimientos de privacidad básica del paciente, donde cada uno de ellos podrá definir para cada estudio realizado que información compartirá y cual no y si está de acuerdo en que su información de salud sea usada en proyectos de investigación o no. (7)

Documento de Firma Digital (**DSG**, por sus siglas en inglés): norma el uso de la firma digital, como vía para garantizar la integridad y autenticidad de los documentos clínicos. (7)

Afirmación de Usuarios entre Empresas (**XUA**, por sus siglas en inglés): define cómo debe realizarse la autenticación entre instituciones mediante intercambio de credenciales entre dominios. (7)

Uno de los sistemas analizados que siguen las especificaciones de los perfiles de IHE es ALERT® EDIS. El cual fue desarrollado por la compañía ALERT que se fundó en 1999 por M. Jorge Guimarães. (8) Este se basa en una concepción de flujos de trabajo, para fácil registro y acceso a la documentación. Con ALERT® EDIS, toda la información puede ser documentada y compartida con otros usuarios de forma inmediata, facilitando el acceso al historial del paciente y resolviendo problemas de comunicación.

ALERT® EDIS brinda como funcionalidades: la documentación clínica para los pacientes de urgencias, la codificación del episodio de urgencia del paciente, las alertas para tareas que requieren atención urgente, la eliminación del uso del papel, las interfaces HL7 con todos los servicios que interactúan con Urgencias, la estandarización, codificación e interoperabilidad de acuerdo con normas internacionales, las herramientas para documentación, gestión de personal, procesos y recursos técnicos, el análisis objetivo y evaluación de las operaciones del servicio utilizando herramientas de asistencia para la toma de decisiones. Algunos de los perfiles IHE que se identifican en ALERT EDIS son: PIX, ATNA, CT, XDS y BPPC.

Otro software que sigue ciertos perfiles de seguridad es GALILEO. El cual es un sistema de HCE desarrollado por NoemaLife S.p.a. Este ayuda a las organizaciones sanitarias a mejorar la calidad, reduciendo los costos y limitando los riesgos durante todo el proceso de atención del paciente. (9)

GALILEO ha sido específicamente diseñado para asegurar una correcta integración entre sus diferentes componentes. Al mismo tiempo, brinda una arquitectura abierta, que permite una flexible y fácil integración con sistemas de terceros. Gracias a su potente capacidad de integración, GALILEO ha sido implementado y probado en el *Connect-a-thon* en 16 diferentes perfiles IHE desde 2003. (9)

Algunos de los perfiles IHE implementados por GALILEO son: BPPC, ATNA, PWP, DSG y XDS. Este último, XDS establece la necesidad de un intercambio seguro de archivos digitales, mediante un Repositorio de Documentos Electrónicos (RDE) donde cada documento tendrá metadatos asociados. Los RDE funcionan como un servidor donde se almacenan dichos datos para ser accedidos luego por los usuarios.

1.3 Herramientas para la implementación de un RDE

En la creación de repositorios de documentos electrónicos se pueden utilizar una serie de herramientas, entre ellas está el software Alfresco. El mismo tiene un sistema de administración de contenido web que posee un potente motor de búsquedas para los documentos almacenados. No depende de una única plataforma. Además es de código libre y está desarrollado en Java, basado en estándares abiertos y de escala empresarial para sistemas operativos como: Windows, Mac y Linux. La principal estrategia de Alfresco es proporcionar un conjunto de funciones, a un coste económico reducido. Proporciona funciones de versionado, control de acceso, transformación de documentos, gestión de los flujos de trabajo y muchas otras que facilitan enormemente el trabajo con grandes volúmenes de documentación. (10) Aunque es un software libre, no significa que sea gratuito en general, implica un costo de implementación y licencias menores.

Al mismo tiempo existen servidores especializados en el almacenamiento y transferencia de ficheros. Estos pudieran ser útiles para guardar los ficheros generados. Un ejemplo de los mismos es el Protocolo de Transferencia de Archivos (FTP, por sus siglas en inglés). Este es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red de Protocolo de Control de Transmisión (TCP, por sus siglas en inglés). Está basado en la arquitectura cliente-servidor. (11) Desde un equipo cliente se establece una conexión a un servidor de este tipo para descargar o subir archivos al mismo. Dicho servidor utiliza los puertos 20 y 21, exclusivamente sobre TCP.

Un problema básico de FTP es que está pensado para ofrecer la máxima velocidad en la conexión, pero no la máxima seguridad. Todo el intercambio de información, desde la contraseña del usuario en el servidor hasta la transferencia de cualquier archivo, se realiza en texto plano sin ningún tipo de cifrado. Esto posibilita que un atacante pueda capturar este tráfico, acceder al servidor y/o apropiarse de los archivos transferidos.

Para solucionar este problema surgen versiones de FTP como: el Protocolo Seguro de Transferencia de Archivos (FTPS, por sus siglas en inglés), el Protocolo de Transferencia de Archivos Seguros (SFTP, por sus siglas en inglés) y el Demonio FTP Muy Seguro (VSFTPD, por sus siglas en inglés), que permiten cifrar el tráfico de archivos. El FTPS también llamado como FTP/SSL, es una extensión de FTP mediante SSL para el cifrado de los datos. El mismo utiliza dos canales que, envían y reciben los mensajes en formato texto. FTPS normalmente es el más conocido ya que usa los mismos comandos que FTP. (12)

El SFTP es completamente diferente del protocolo FTP. SFTP fue construido desde cero y añade la característica de FTP a SSH. Sólo usa un canal de comunicación, envía y recibe los mensajes en binario (y no en formato texto como hace FTP). (12) También utiliza el puerto 22 de TCP.

Por último VSFTPD es un equipamiento lógico utilizado para implementar servidores de archivos a través del protocolo FTP. Se distingue principalmente porque sus valores predeterminados son muy seguros y por su sencillez en la configuración, comparado con otras alternativas. Actualmente se presume que VSFTPD podría ser, quizás, el servidor FTP más seguro del mundo. (13) Entre sus características más interesantes se encuentran: que es un servidor rápido, estable y seguro; de configuración sencilla; que establece límites por usuario, conexión y ancho de banda; entre otras.

1.4 Algoritmos de encriptación simétricos

Una vez que se haya establecido la herramienta más adecuada para el almacenamiento de los documentos clínicos, resulta indispensable asegurar su confidencialidad. La misma puede garantizarse

con el empleo de métodos criptográficos. Estos están basados en algoritmos matemáticos tales que aplicados sobre ciertos datos más un argumento variable (clave) producen un determinado resultado no legible (texto cifrado). (14) Los mismos se clasifican en tres grupos: criptografía simétrica o de clave secreta, criptografía asimétrica o de clave pública y hash o de resumen. Teniendo en cuenta que no es necesario el intercambio de llaves entre componentes de la red, los algoritmos de encriptación más apropiados son los simétricos, debido a que su aplicación es mucho más sencilla que la del resto.

Uno de estos algoritmos es el Estándar de Encriptación de Datos (DES, por sus siglas en inglés). Es un algoritmo de cifrado de datos desarrollado originalmente por la compañía IBM (International Business Machines) y posteriormente modificado y adoptado por los Estados Unidos de América (EE.UU) en 1977, como estándar de cifrado de todas las informaciones sensibles no clasificadas. El mismo realiza combinaciones, sustituciones e intercambios entre el texto a cifrar y la clave, permitiendo que las operaciones puedan realizarse en ambas direcciones. Se trata de un sistema de cifrado simétrico por bloques de 64 bits, de los cuales 8 bits (un byte) se utilizan como control de paridad (para la verificación de la integridad de la clave). (15)

Tiene una clave de longitud útil de 56 bits, que son utilizados en el algoritmo. Aunque se demostró que es posible un ataque por fuerza bruta, por la escasa longitud que emplea en su clave, no presenta ninguna debilidad grave desde el punto de vista teórico, por lo que se ha convertido en el algoritmo simétrico más extendido mundialmente.

Por otra parte, se encuentra el Algoritmo Internacional de Cifrado de Datos (IDES, por sus siglas en inglés). Este algoritmo data de 1992. Es más joven que DES y al igual que éste usa el mismo algoritmo simétrico tanto para cifrar como para descifrar. Opera con bloques de 64 bits usando una clave de 128 bits. (16)

Es un algoritmo seguro que ha mostrado ser resistente a multitudes de ataques. Para muchos constituye el mejor y más seguro de los simétricos disponibles en la actualidad. Su fortaleza se basa en que dada la longitud de su clave es imposible en la práctica atacar mediante la fuerza bruta, ya que sería necesario probar 10^38 claves. (16)

Por último, está el Estándar Avanzado de Cifrado (AES, por sus siglas en inglés), el cual, en octubre de 2000 fue adoptado por el Instituto Nacional de Normas y Tecnología de los EE.UU, el algoritmo Rijndael, acrónimo derivado de los nombres de sus dos autores; los belgas Joan Daemen y Vincent Rijmen. Este fue empleado en aplicaciones criptográficas no militares por ser un algoritmo de cifrado potente, eficiente, y fácil de implementar. (17)

AES es un sistema de cifrado por bloques, que soporta diferentes tamaños de bloque y clave. Utiliza una de las tres fortalezas de clave de cifrado: una clave de encriptación (contraseña) de 128, 192, o 256 bits. Cada tamaño de la clave de cifrado hace que el algoritmo se comporte ligeramente diferente, por lo que el aumento de tamaño de clave no sólo ofrece un mayor número de bits con el que se pueden cifrar los datos, sino que también aumenta la complejidad del algoritmo de cifrado. (17)

El proceso de selección, revisión y estudio de este algoritmo se efectúa de forma pública y abierta por toda la comunidad criptográfica mundial, lo cual lo convierte en un algoritmo perfectamente digno de la confianza de todos. Se ha comprobado que es resistente al criptoanálisis y se considera como uno de los más seguros en la actualidad. AES es rápido tanto en software como en hardware y requiere poca memoria. Como nuevo estándar de cifrado, se está utilizando actualmente a gran escala. (18)

Los algoritmos anteriores enunciados se relacionan debajo para alinear sus características y establecer una comparación.

Tabla 1 Características de los algoritmos DES, IDEA y AES.

Algoritmo	Técnicas de cifrado	Tamaño de la clave	Resistencia	Ventaja
DES	Por bloque (64 bits)	56 bits(8 paridad)	Susceptible a ataques de fuerza bruta debido a la escasa longitud de su clave.	Rápido y fácil de implementar. Usa el mismo algoritmo para cifrar como para descifrar.
IDEA	Por bloque (64 bits)	128 bits	Resistente a ataques por fuerza bruta y al criptoanálisis diferencial.	Usa el mismo algoritmo para cifrar como para descifrar.
AES	Por bloque (de longitud variable y si es Rinjdael entonces	De longitud variable y si es Rinjdael entonces de	Resistente al criptoanálisis diferencial y lineal.	Uno de los algoritmos más seguros en la

	de 128 bits)	128, 192 ó 256 bits.		actualidad.

Después de describir los distintos sistemas mencionados anteriormente, se demostró que los perfiles de seguridad IHE son implementados por un importante grupo de desarrolladores de aplicaciones sanitaras. Para el desarrollo de las funcionalidades que fortalezcan la seguridad del módulo Admisión, se propone seguir, además las especificaciones de los perfiles de integración ATNA, EUA y XDS propuesto por IHE.

Por otro lado, la herramienta VSFTPD es una alternativa viable para el desarrollo de un RDE, dado que es un servidor rápido, ligero, seguro y de fácil configuración. Así mismo el algoritmo AES es el más adecuado para la encriptación de datos por ser el más seguro de los algoritmos simétricos en la actualidad y el más usado.

Todos estos elementos constituyen la base para el desarrollo de una solución que resuelva las actuales fisuras en la seguridad del módulo Admisión del sistema alas HIS. A continuación se describen las tecnologías, herramientas y metodología empleadas para realizar las configuraciones necesarias y la implementación de las funcionalidades requeridas (Para ver una descripción más detallada de estos perfiles consultar los anexos del 1 al 3).

1.5 Metodología, tecnologías y herramientas

El desarrollo de funcionalidades para el componente de seguridad debe insertarse de manera que tenga un mínimo impacto en la actual implementación del módulo Admisión; así como en el resto de los módulos del sistema alas HIS. Estas deben estar libres de costos asociados a licencias de software. Por lo antes descrito se hace necesario asimilar las tecnologías, herramientas y metodología utilizadas en el sistema alas HIS, así como definir las específicas a utilizar para el desarrollo de la solución propuesta.

Esto se realiza siguiendo las especificaciones de los perfiles de seguridad propuestos por IHE relacionados con el almacenamiento y la transmisión de los datos. Todos estos aparecen relacionados y se describen a continuación, para facilitar la comprensión de los elementos que motivaron su elección.

1.5.1 SSL

El protocolo Capa de Conexión Segura (SSL, por sus siglas en inglés), permite conectarse de forma segura entre dos extremos de la red, mediante técnicas de encriptación y criptografía. Además para el

funcionamiento de este, se utilizan certificados que permiten asegurar la identidad de una persona y un nodo en la red. Estos son emitidos por las entidades certificadoras. Las mismas validan los datos del solicitante, y adjuntan su certificado para probar la veracidad.

El certificado tiene dos claves una privada que solo la conoce el propietario del certificado y una pública que la conocen todos los usuarios. Esta última se encripta y se arma en base a la privada. Dichas llaves pueden tener 128, 512, o 1024 bits de datos. Todos estos elementos constituyen a base del funcionamiento de SSL.

En resumen, un certificado sirve como un "pasaporte" electrónico que establece las credenciales de la entidad en línea al hacer negocios en la Web. Cuando un usuario de Internet intenta enviar información confidencial a un servidor Web, el navegador del usuario accede al certificado digital del servidor y establece una conexión segura. (19)

1.5.2 HTTPS

El Protocolo de Transferencia de Hipertexto Seguro (HTTPS, por sus siglas en inglés) se utiliza para garantizar la privacidad y la seguridad en la transmisión de la información del usuario hacia el servidor de aplicaciones. Este es un protocolo ubicado en la Capa de Aplicación del Modelo OSI basado en HTTP. El mismo es destinado a la transferencia segura de datos de Hipertexto. Es decir, es la versión segura de HTTP, que es el método más común de intercambio de información en Internet a través de páginas web.

HTTPS es un subconjunto del protocolo SSL, que funciona utilizando HTTP, pero en vez de utilizar conexiones tradicionales de *TCP*/IP, utiliza conexiones de SSL como base. Se cifra un mensaje HTTP previo a la transmisión y se descifra una vez recibido. Estrictamente hablando, HTTPS no es un protocolo separado porque refiere el uso de HTTP sobre SSL.

1.5.3 Enterprise JavaBeans (EJB)

Los EJB3 son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE. Estos proporcionan un modelo de componentes distribuidos estándar del lado del servidor. El objetivo de los EJB3 es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial. La concurrencia, las transacciones, la persistencia, la seguridad, entre otros problemas clásicos, son manejados por estos permitiendo al desarrollador centrarse en la implementación de la lógica del negocio. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables. (20)

1.5.4 Java Persistence API (JPA)

JPA es la API de persistencia desarrollada para la plataforma Java EE e incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue su diseño es no perder las ventajas de la orientación a objetos al interactuar con una base de datos y permitir usar objetos regulares conocidos como POJOs. (20)

1.5.5 Hibernate

Es una herramienta de mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación. Esto se realiza mediante archivos declarativos XML que permiten establecer estas relaciones. Hibernate es un software libre, distribuido bajo los términos de la licencia GNU LGPL. (21)

Este busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información, Hibernate posibilita a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la Programación Orientada a Objetos. (21)

Hibernate convierte los datos entre los tipos utilizados por Java y los definidos por SQL. Genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias. Todos ellos se realizan manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución. Está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente.

Además ofrece un lenguaje de consulta de datos llamado Lenguaje de Consultas de Hibernación (HQL, por sus siglas en inglés), al mismo tiempo que una API para construir las consultas. Para Java, Hibernate puede ser utilizado en aplicaciones independientes o en aplicaciones de Edición Empresarial de la Plataforma Java (Java EE, por sus siglas en inglés). Esto se logra mediante el componente Hibernate Annotations que implementa el estándar JPA, el cual es parte de esta plataforma. (22)

1.5.6 Java 1.6

Es una plataforma de programación (parte de la Plataforma Java) para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con arquitectura de N niveles distribuida. Se basa

ampliamente en componentes de software modulares y se ejecuta sobre un servidor de aplicaciones. (23)

1.5.7 Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. (24)

1.5.8 Drools

Drools es una implementación del JSR 94 (Java Rule Engine API), una especificación que define una interfaz común para un motor de reglas estándar dentro de la plataforma Java. Para definir las reglas emplea XML y permite adaptarse a la semántica de un determinado dominio definiendo un esquema que la represente. Su licencia es BSD (Berkeley Software Distribution) que poco después de la liberación de la versión 2.0, se unió a la compañía JBoss, la cual ofrece servicios de consultoría, formación y soporte sobre el producto bajo el nombre JBoss Rules.

1.5.9 Eclipse 3.2.4

Eclipse es un entorno de desarrollo integrado de código abierto y multiplataforma para desarrollar lo que el proyecto alas HIS llama Aplicaciones de Cliente Enriquecido, opuesto a las aplicaciones Cliente liviano basadas en navegadores web. Esta plataforma, ha sido típicamente usada para desarrollar Entornos de Desarrollo Integrados (IDE, por sus siglas en inglés). Como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se integra como parte de Eclipse y que son usados también para desarrollar al mismo. Sin embargo, también se puede usar para otros tipos de aplicaciones cliente. (24)

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. El mismo es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. (24)

1.5.10 Apache Commons

Las librerías de Apache Commons brinda la posibilidad de conectarse a un FTP con Java. Además implementa el lado cliente de muchos protocolos básicos de Internet. El propósito de la biblioteca es proporcionar acceso de protocolo fundamental, no abstracciones de nivel superior. Por lo tanto,

algunos de los diseños violan los principios del diseño orientado a objetos. La filosofía es hacer la funcionalidad global de un protocolo de acceso (por ejemplo, FTP enviar y recibir archivos de archivos) cuando sea posible, sino que también proporcionan acceso a los protocolos fundamentales en su caso para que el programador puede construir sus propias implementaciones personalizadas. (25)

1.5.11 Open SSL

Usando la biblioteca Open SSL, los desarrolladores pueden escribir aplicaciones que usan SSL y librería de criptografía. Esta incluye una multitud de algoritmos de codificación. Los algoritmos de codificación simétrica o de clave privada que incluyen Rijndael y los algoritmos asimétricos o de clave privada y pública. Esta herramienta junto a Keytool permite crear certificados de servidores. (26)

1.5.12 Keytool

Keytool es una herramienta de criptografía con interfaz gráfica de usuario que permite crear, gestionar claves y certificados autofirmados, así como verificar, firmar, cifrar y descifrar los archivos. (26)

1.5.13 VSFTPD

Es un servidor FTP, el cual es de código abierto y está diseñado desde la base para ser rápido, estable y seguro. Su habilidad para manejar grandes números de conexiones de forma eficiente y segura es lo que lo hace el único FTP independiente distribuido. (27)

1.5.14 Pgadmin 3

Es una herramienta de código abierto para la administración de bases de datos PostgreSQL y derivados. El mismo incluye una interfaz gráfica administrativa y herramientas de consultas SQL. Además está diseñado para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. (28)

1.5.15 Servidor de aplicaciones Jboss 4.2

Es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Por ser una plataforma certificada Java EE, soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como agrupar, capturar y persistir. JBoss es ideal para aplicaciones Java y aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0, esto hace que el desarrollo de las aplicaciones empresariales sean mucho más simples. (24)

1.5.16 PostgreSQL Server 9.1

PostgreSQL es un sistema de gestión de bases de datos de objeto relacional. Considerado estable y robusto PostgreSQL encabeza los sistemas de gestión de bases de datos objeto relacional desarrollado bajo licencia de código abierto. El mismo implementa una gran parte del lenguaje de consultas SQL y permite realizar consultas complejas. Además soporta el uso de llaves foráneas, eventos disparadores, vistas e integridad transaccional. (24)

1.5.17 RUP

El Proceso Unificado de Desarrollo de Software (RUP, por sus siglas en inglés), es un proceso de ingeniería de software orientado a objetos. Consiste en un conjunto de actividades necesarias para transformar los requerimientos del usuario en el sistema de software. Está especializado para diversos tipos de software de sistemas, diversas áreas de aplicación, diferentes tipos de organizaciones y diferentes tamaños de proyectos. (29)

Además se basa en componentes, lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas. Igualmente usa el Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) en la preparación de todos los planos del sistema. De hecho, UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Permite la modelación de sistemas con tecnología orientada a objetos.

1.5.18 Visual Paradigm 6.4

Visual Paradigm es una herramienta para el modelado de software, que ayuda a que el mismo sea más fácil y rápido de realizar. Permite diseñar todos los diagramas necesarios y contiene una buena cantidad de productos o módulos para facilitar el trabajo, durante la confección de un software. (18)

Desde Visual Paradigm es posible generar códigos para plataformas como .NET, Java y PHP, así como obtener diagramas a partir del código. Esto reduce la posibilidad de cometer errores y ahorra tiempo a los desarrolladores. El mismo integra con numerosos ambientes de desarrollo integrados, lo que permite pasar del código al modelado y viceversa. Soporta UML 2.1 y permite documentar todo el trabajo y las especificaciones de los casos de uso, sin necesidad de utilizar herramientas externas. (30)

Conclusiones

Como resultado del estudio realizado en este capítulo, se puede concluir que los perfiles de seguridad de IHE, ATNA, EUA y XDS; constituyen una guía para el desarrollo de la solución propuesta. El

servidor VSFTPD es una alternativa viable a utilizar para el desarrollo de un RDE donde se almacenan los documentos clínicos generados durante la atención de los pacientes. Además debe realizarse con herramientas no privativas y libres de costo.

Capítulo 2: Características de la solución propuesta

En este capítulo se describe la propuesta de solución para el desarrollo del componente de seguridad. Para ello se comienza con una breve explicación de los procesos fundamentales del módulo Admisión. Se expone el marco conceptual en el que se debe desarrollar y se establecen las relaciones entre los principales conceptos que lo conforman. Por otra parte se especifican los requisitos de software que debe cumplir la propuesta de solución.

2.1 Flujo actual de los procesos

Cumpliendo con la metodología seleccionada resulta útil el estudio de la forma en que se desarrollan actualmente los procesos relacionados con los problemas identificados en el almacenamiento y transmisión de las HCE. Durante dicho análisis se encontró que el usuario accede al módulo Admisión mediante un navegador web, donde introduce la información asociada al paciente, la cual es enviada a través de la red al servidor de aplicaciones, utilizando el protocolo HTTP.

Una vez que la información está en el servidor de aplicaciones, Hibernate se encarga de su almacenamiento y transmisión hacia el servidor de base de datos utilizando el protocolo TCP/IP. Por otra parte, el módulo Visor HC genera ficheros en formato CDA con los datos clínicos del paciente, de los cuales se desea conservar un historial, estos son firmados digitalmente para garantizar su autenticidad e integridad, y quedan almacenados localmente en el servidor web.

Para el desarrollo de dichas funcionalidades, resulta realmente útil realizar un modelado que refleje los principales conceptos relacionados con el funcionamiento actual del sistema alas HIS. El mismo representará las relaciones entre dichos conceptos y está enmarcado específicamente en el módulo Admisión.

2.2 Modelo de dominio

La utilización del modelo de dominio o modelo conceptual brinda una representación gráfica y un mejor entendimiento de cómo interactúan los conceptos fundamentales que intervienen en el flujo de los procesos anteriormente expuestos. Este describe entidades o conceptos del mundo real que están asociados al problema en cuestión. Dicho modelo se utiliza como una base de las abstracciones relevantes en el proceso de construcción de la solución propuesta.

2.2.1 Conceptos fundamentales del dominio

Para la comprensión del diagrama del modelo de dominio a continuación se realiza una breve descripción de los conceptos encontrados en el ámbito del problema.

alas HIS: Sistema de Información Hospitalaria.

Módulos: Subsistemas en que se encuentra dividido el sistema alas HIS.

Persona: Entidad que contiene los datos personales.

HC local: Entidad que contiene los datos de la HCE.

Módulo Admisión: Módulo del sistema alas HIS encargado de registrar los pacientes que ingresan a la institución de salud.

Módulo Visor HC: Módulo que gestiona las HCE de los pacientes registrados en el sistema.

CDA: Documento clínico electrónico del paciente.

Hibernate: Capa de persistencia y acceso a datos.

Gestor de Base de Datos: Es el que se encarga de la manipulación de la información almacenada.

2.2.2 Diagrama del modelo de dominio

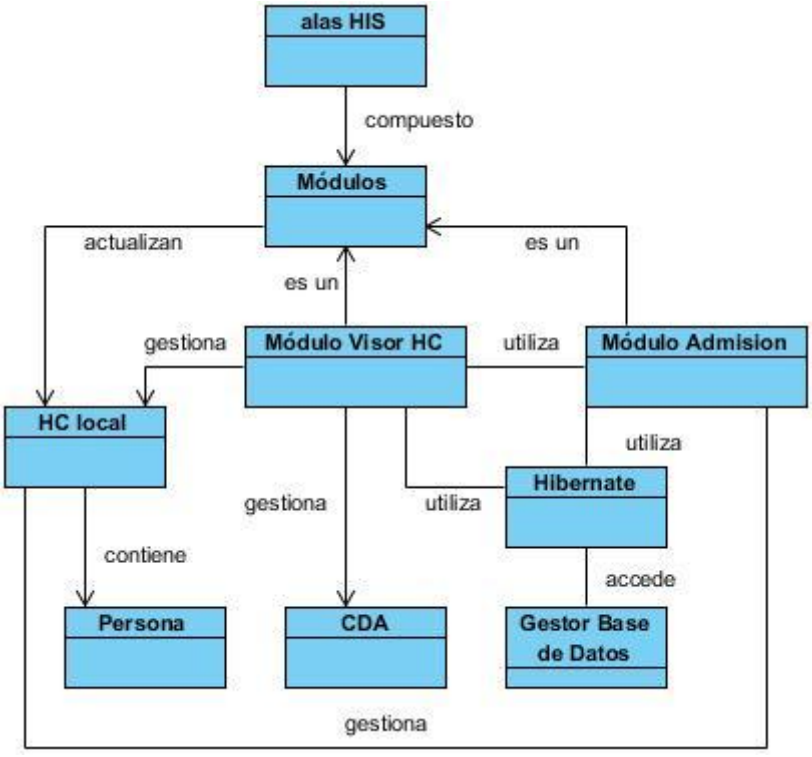

Ilustración 1 Diagrama del modelo de dominio.

2.4 Propuesta de solución

El modelo de dominio expuso los elementos que interactúan para ejecutar las actuales funcionalidades que brindan los módulos Admisión y Visor HC, insertados en la arquitectura del sistema alas HIS. El análisis del mismo evidenció que los principales problemas existentes pueden ~~concentrarse~~ dividir en dos grupos.

El primer grupo reúne los elementos relacionados con la transmisión legible de la información. Para dar solución se siguen los perfiles de seguridad EUA y ATNA que proponen crean certificados autofirmados para avalar que un sistema o nodo sea quien dice ser y se sustituyan los protocolos HTTP y TCP/IP por ~~los~~ HTTPS y SSL respectivamente, para garantizar la confidencialidad de los datos mediante su encriptación.

El segundo grupo engloba los elementos asociados al desarrollo de funcionalidades que solucionen las problemáticas de confidencialidad de la información clínica. Dichas funcionalidades estarán orientadas a los datos guardados en el gestor de base de datos, y en los documentos clínicos almacenados en el servidor de aplicaciones. Para solucionar esta situación se encripta y desencripta la información mediante el algoritmo simétrico AES con una llave de 256 bits, para lograr una mayor resistencia a ataques informáticos. De esta forma se asegura que los datos y ficheros se guarden de forma ilegible. Además se crea un servidor VSFTPD para almacenar todas las HCE generadas en el módulo Visor HC como recomienda el perfil de seguridad XDS.

2.3 Especificación de los requerimientos de software

La especificación de los requerimientos de software constituye un elemento de vital importancia para la elaboración de un software de calidad superior. Los que se definen como condiciones o capacidades que deben estar presentes en un sistema o componentes de este, para satisfacer un contrato, estándar, especificación u otro documento formal. Es necesario hacer énfasis en la precisión con que se debe realizar esta tarea, por que cumple un papel primordial en el proceso de producción de software, pues se enfoca en un área fundamental: las exigencias que debe cumplir y los procesos a automatizar, permitiendo describir con mayor claridad el comportamiento del sistema minimizando los problemas derivados de su desarrollo. (31)

2.3.1 Requisitos funcionales

Los requisitos funcionales (**RF**) se definen como las condiciones o capacidades que el sistema debe cumplir. Luego del análisis de las funcionalidades a desarrollar para cumplir con las necesidades de seguridad del módulo Admisión se determinan como requisitos funcionales los siguientes:

RF1: Capturar eventos.

La solución propuesta debe permitir que mediante los eventos de Hibernate se puedan obtener y modificar los valores que contenga una entidad al actuar sobre ella.

RF2: Encriptar datos.

El componente debe brindar la posibilidad de encriptar los datos de identidad del paciente, salvados al registrarse uno nuevo en el sistema, mediante un algoritmo simétrico.

RF3: Desencriptar datos.

El componente debe permitir la posibilidad de desencriptar, mediante un algoritmo simétrico, los datos personales de un paciente cuando son consultados por parte de cualquier módulo del sistema alas HIS.

RF4: Encriptar fichero.

Debe brindarse la posibilidad de encriptar usando un algoritmo simétrico los documentos generados por el módulo Visor HCE.

RF5: Desencriptar fichero.

Debe ser posible desencriptar los documentos antes de ser visualizados por el módulo Visor HCE.

Para dar cumplimiento a los requisitos **RF2**, **RF3, RF4 y RF5**, se utiliza el algoritmo simétrico AES por ser el más seguro y fácil de implementar. Además se emplea con una llave de 256 bits para lograr una mayor complejidad de dicho algoritmo.

RF6: Guardar en FTP.

La solución propuesta debe permitir guardar los documentos clínicos en el RDE.

RF7: Descargar del FTP.

El componente debe brindar la posibilidad de descargar los ficheros del RDE.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales (**RNF**) son cualidades o propiedades con las que debe de cumplir la solución. Son las características que logran que la solución sea más segura, rápida y confiable. El componente de seguridad debe tener las características que han sido agrupadas en las siguientes secciones:

Seguridad

RNF1: Debe resolver, de manera satisfactoria, los problemas de seguridad de la información de los pacientes, garantizando que el texto quede realmente ilegible.

RNF2: Los documentos CDA deben quedar encriptados y protegidos en un repositorio de documentos electrónicos, con las políticas de seguridad definidas por los perfiles propuestos por IHE.

RNF3: El servidor web se comunicará mediante el protocolo SSL con el servidor de base de datos, garantizando de esta manera la seguridad en el flujo de datos.

Fiabilidad

RNF4: La información intercambiada entre el cliente web y el servidor de aplicaciones, debe viajar cifrada para evitar accesos o modificaciones no autorizadas.

RNF5: Deben utilizarse certificados digitales que permitan asegurar la identidad entre los nodo en la red.

Para asegurar estos requisitos no funcionales se utiliza el protocolo de comunicación segura HTTPS. Este garantiza la autenticidad de los nodos que intercambian información a través de la red mediante un certificado auto firmado. Además de proteger los datos durante la transferencia mediante la creación de un canal cifrado para comunicaciones privadas.

Configuración

RNF6: Se deben crear certificados autofirmados, el navegador debe aceptar conexiones seguras SSL.

RNF7: Se debe configurar el servidor de aplicaciones para responder solo peticiones HTTPS y realizar conexiones seguras a la bases de datos utilizando SSL.

RNF8: Es necesario crear un VSFTPD como repositorio para los documentos clínicos aplicando las políticas de seguridad pertinentes.

Software

RNF9: La aplicación debe correr en sistemas operativos Unix y Linux, utilizando la plataforma JAVA (Java Virtual Machine, VSFTPD, JBoss AS y PostgreSQL).

2.4 Modelo de casos de uso de la solución propuesta

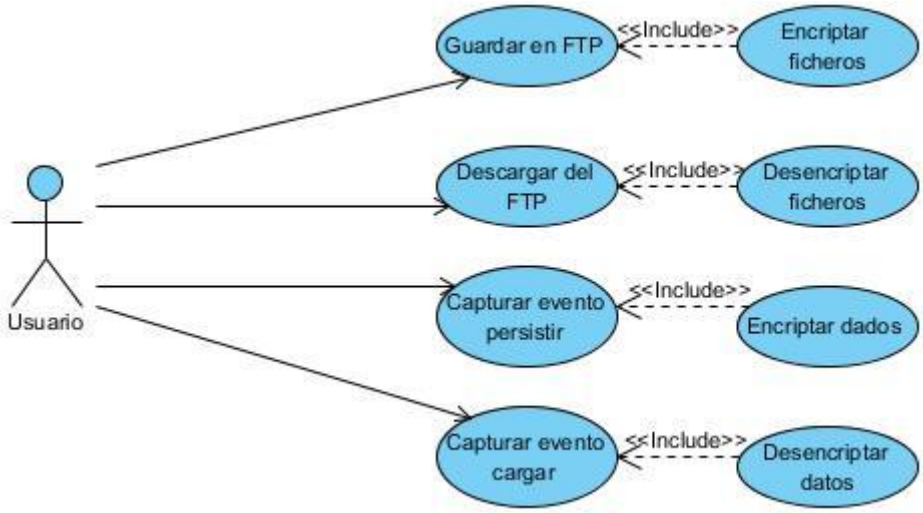

Ilustración 2 Modelo de caso de uso de la solución propuesta.

2.4.1 Descripción textual de los casos de uso

Tabla 2 CU guardar en FTP.

CASO DE USO:	Guardar en FTP.
Resumen:	El caso de uso inicia cuando el actor crea o modifica un documento clínico, el sistema brinda la posibilidad de guardar los ficheros en el RDE y el caso de uso termina.
Complejidad:	Media.
Prioridad:	1
REFERENCIAS	
Actores:	Usuario.
Requisitos:	**RF7**

Casos de Uso:	Encriptar ficheros.
FLUJO NORMAL DE EVENTOS	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso inicia cuando el actor accede a la opción ver documentos clínicos del paciente.	
	2. El sistema se conecta al FTP proporcionando dirección, usuario y contraseña. El fichero es guardado en el RDE.
	3. El caso de uso termina.
Poscondiciones	El fichero queda almacenado de forma segura.

Tabla 3 CU descargar del FTP.

CASO DE USO:	Descargar del FTP.
Resumen:	El caso de uso inicia cuando el actor accede a la opción ver documento clínico, el sistema brinda la posibilidad de descargar los ficheros del RDE y el caso de uso termina.
Complejidad:	Media.
Prioridad:	1
REFERENCIAS	
Actores:	Usuario.
Requisitos:	**RF6**
Casos de Uso:	Desencriptar ficheros.

FLUJO NORMAL DE EVENTOS	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso inicia cuando el actor accede a la opción ver documento clínico del paciente.	
	2. El sistema se conecta al FTP proporcionando dirección, usuario y contraseña. El fichero seleccionado es descargado del RDE.
	3. El caso de uso termina.
Poscondiciones	El fichero es descargado.

Tabla 4 CU capturar evento persistir.

CASO DE USO:	Capturar evento persistir.
Resumen:	El caso de uso inicia cuando el actor guarda o modifica los datos del paciente, el sistema captura la acción de persistir sobre una entidad, lo que hace posible obtener los valores pertenecientes a la misma para su modificación, antes de que termine el proceso y el caso de uso termina.
Complejidad:	Media.
Prioridad:	1
REFERENCIAS	
Actores:	Usuario.
Requisitos:	**RF1**
Casos de Uso:	Encriptar datos.

FLUJO NORMAL DE EVENTOS	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso inicia cuando el actor guarda o modifica los datos del paciente.	
	2. El sistema captura la acción de persistir sobre la entidad dada.
	3. El caso de uso termina.

Tabla 5 CU capturar evento cargar.

CASO DE USO:	Capturar evento cargar.
Resumen:	El caso de uso inicia cuando el actor solicita la información de una entidad, esto hace posible obtener los valores pertenecientes a la misma para su modificación, antes de que termine el proceso y el caso de uso termina.
Complejidad:	Media.
Prioridad:	1
REFERENCIAS	
Actores:	Usuario.
Requisitos:	**RF1**
Casos de Uso:	Desencriptar datos.
FLUJO NORMAL DE EVENTOS	
Acción del Actor	**Respuesta del Sistema**

1. El caso de uso inicia cuando el actor solicita los datos del paciente.	
	2. El sistema captura la acción sobre la entidad dada.
	3. El caso de uso termina.

Tabla 6 CU encriptar ficheros.

CASO DE USO:	Encriptar ficheros.
Resumen:	El caso de uso inicia cuando el actor crea o modifica un documento clínico, el sistema brinda la posibilidad de encriptar ante de ser almacenado en el RDE y el caso de uso termina.
Complejidad:	Alta.
Prioridad:	1
REFERENCIAS	
Actores:	Usuario.
Requisitos:	**RF4**
Casos de Uso:	Guardar en FTP.
FLUJO NORMAL DE EVENTOS	
Acción del Actor	**Respuesta del Sistema**
1. El caso de uso inicia cuando el actor solicita los datos del paciente.	
	2. El sistema encripta el fichero.

	3. El caso de uso termina.

Tabla 7 CU desencriptar datos.

CASO DE USO:	Desencriptar datos.	
Resumen:	El caso de uso inicia cuando el actor solicita la información de una entidad, el sistema desencripta los datos del paciente y el caso de uso termina.	
Complejidad:	Alta.	
Prioridad:	1	
REFERENCIAS		
Actores:	Usuario.	
Requisitos:	**RF3**	
Casos de Uso:	Capturar evento cargar.	
FLUJO NORMAL DE EVENTOS		
Acción del Actor		**Respuesta del Sistema**
1. El caso de uso inicia cuando el actor solicita los datos del paciente.		
		2. El sistema desencripta la información del paciente.
		3. El caso de uso termina.

Conclusiones.

En este capítulo se especifica los principales procesos asociados al dominio de la aplicación. Con el apoyo de diagramas y fichas se elaboró una representación detallada de cada proceso. Además para lograr la seguridad requerida se propuso la automatización de nuevas funcionalidades que

conformarán el componente. Se realizaron las actividades correspondientes a la especificación de los requisitos de software, las cuales permitieron conocer las necesidades reales y obtener los artefactos correspondientes. De esta manera ha sido posible alcanzar un mejor entendimiento de la solución propuesta.

Capítulo 3: Diseño y desarrollo de la solución propuesta

El objetivo que se persigue con el desarrollo de este capítulo, es llevar a cabo la descripción de la solución propuesta. Dentro de los temas que serán mostrados se encuentran el Modelo de Diseño, especificando la estructura y la definición de los elementos de la arquitectura; Diagrama de Clases y descripción de las clases del diseño. Así como los aspectos necesarios para el diagrama de despliegue y las estrategias de codificación.

3.1 Descripción de la arquitectura

La Arquitectura de Software es un diseño de alto nivel de la estructura de un sistema. En la misma se describen los patrones y diagramas más relevantes, estableciendo un modelo o línea base de desarrollo. De esta forma se crea un sistema de abstracciones coherentes que proporciona el marco de referencia necesario y erige las pautas a seguir en la construcción del sistema. (32)
Para el desarrollo del componente de seguridad, se asimila la arquitectura en capas presentes en el sistema alas HIS, que es donde se enmarca el dominio de la aplicación. La misma está compuesta por las capas de presentación, negocio y acceso a datos.

La capa de negocio contiene la propuesta de solución, la que está formada por funcionalidades que cumplen con las necesidades del sistema. Se utilizan pautas establecidas para guiar el cumplimiento de los requisitos y asegurar un resultado deseado. Además se utiliza el patrón Observador, el cual define una dependencia del tipo uno a muchos entre objetos, de tal forma que cuando un objeto cambia su estado, todos los objetos dependientes de este, son notificados y actualizados automáticamente. El empleo de este patrón garantiza que el componente desarrollado sea notificado al realizarse alguna acción sobre la entidad Persona, esta es la que almacena la información de identidad del paciente.

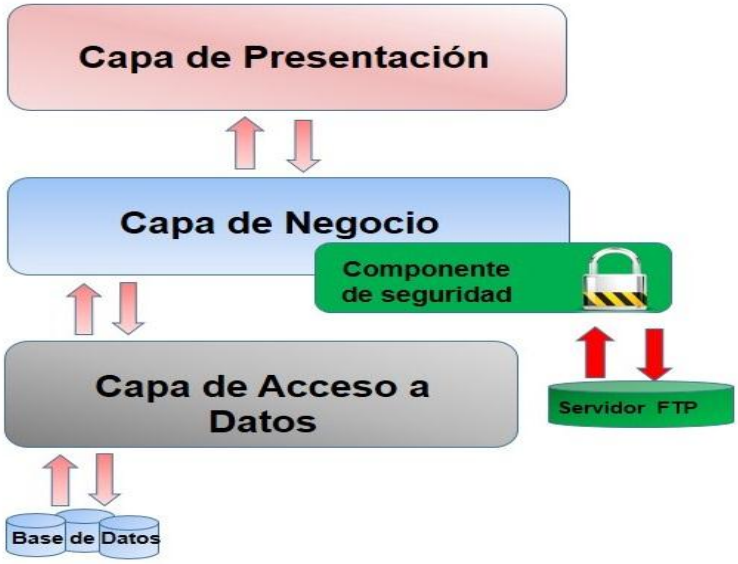

Ilustración 3 Arquitectura de la solución propuesta de software.

3.2 Modelo de diseño

Un modelo de diseño describe la realización física de los casos de uso, centrándose en cómo los requerimientos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación; tienen impacto en el sistema. Este artefacto constituye la entrada fundamental utilizada para el correcto desarrollo de la implementación. (31)

3.2.1 Diagrama de clases del diseño

Mediante los diagramas de clases del diseño es posible obtener una abstracción de cómo quedará la implementación del sistema, los diagramas especifican la relación entre los distintos elementos, lo que hace que sea más fácil la implementación y proveen una abstracción de lo que será el producto final. En este epígrafe se presenta el diseño de las clases del sistema. En el mismo se muestra primeramente de forma general y luego se expone dividido por las clases correspondientes a cada paquete. El estudio de este diagrama ayuda a un mejor entendimiento de cómo fue diseñada la propuesta de solución.

3.2.2 Diagrama de paquetes

En el diagrama de paquetes se muestra cómo el sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. En la imagen se representa el paquete alas HIS que está dividido por los paquetes módulos Admisión, módulo Visor HC y el paquete Seguridad que se

relacionan con los dos mencionados anteriormente. Dentro de este último se relacionan los paquetes Capturar y Gestión CDA con el paquete cifrado. Además Gestión CDA utiliza el paquete Apache.commos.net como librería.

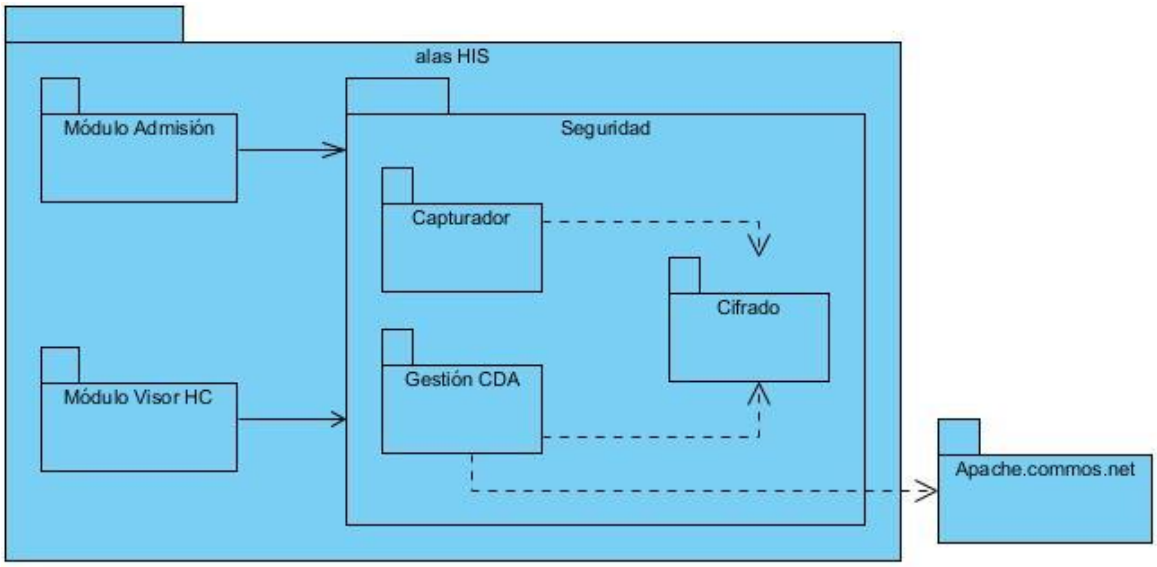

Ilustración 4 Diagramas de paquetes.

3.2.3 Diagrama de clases del diseño pertenecientes al paquete Seguridad

El diagrama muestra cómo están relacionadas y distribuidas las clases del diseño pertenecientes al paquete donde se localizan los componentes de la solución propuesta.

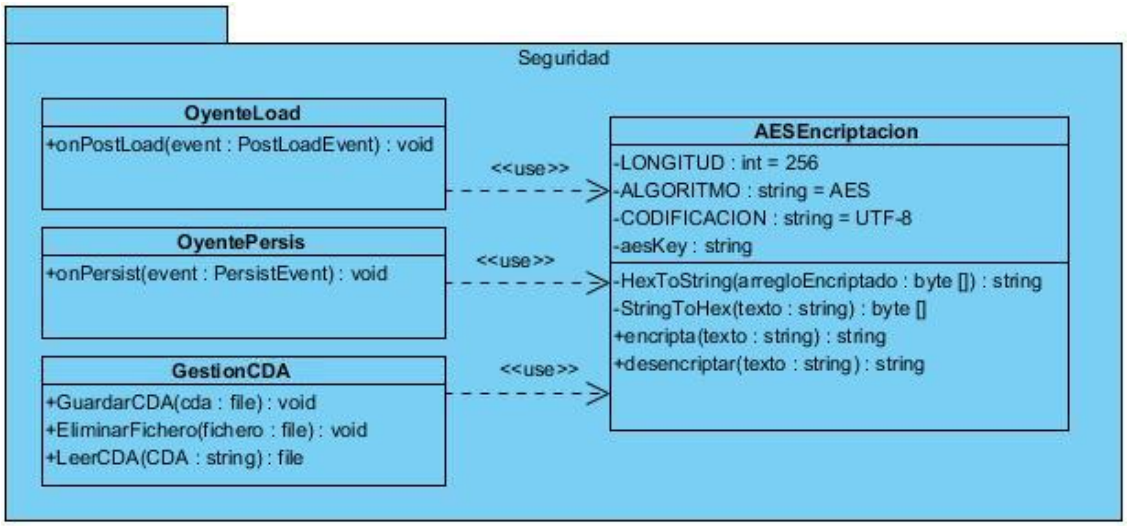

Ilustración 5 Diagrama de clases del diseño pertenecientes al paquete Seguridad.

3.2.4 Descripción de las clases

OyenteLoad: clase que se encarga de capturar el evento de Hibernate PostLoad sobre la entidad HojaFrontal_admision.

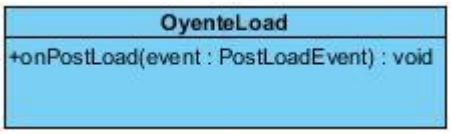

Ilustración 6 Clase OyenteLoad.

En la clase OyenteLoad se encuentra el método onPostLoad. El método onPostLoad se encarga de capturar el evento de Hibernate PostLoad lanzado después de cargar la entidad HojaFrontal_admision, este obtiene una instancia de la misma para su modificación.

OyentePersis: clase que se encarga de capturar el evento de Hibernate PersistEvent sobre la entidad HojaFrontal_admision.

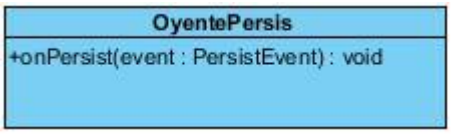

Ilustración 7 Clase OyentePersis.

En clase OyentePersis se encuentra el método onPersist. Este se encarga de capturar el evento de Hibernate PersistEvent capturando la instancia antes de ser persistida la entidad HojaFrontal_admision, para su modificación.

AESEncriptacion: clase que se encarga de encriptar y desencriptar los datos del paciente.

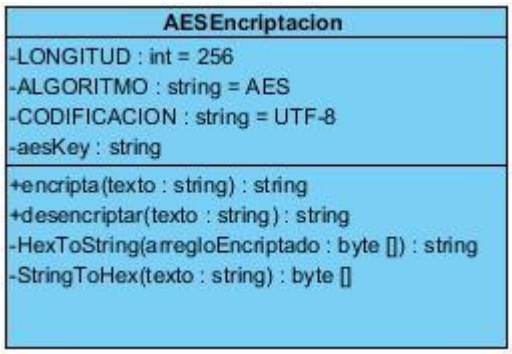

Ilustración 8 Clase AESEncriptacion.

En la clase AESEncriptacion se compone de los atributos LONGITUD, ALGORITMO, CODIFICACION, aesKey además de los métodos encripta, desencripta, HexToString, StringToHex.

El atributo LONGITUD: es la longitud que tendrá la llave que se utilizará para encriptar y desencriptar.

El atributo ALGORITMO: es el nombre del algoritmo que se utilizará.

El atributo CODIFICACION: este contendrá el tipo de codificación.

El atributo aesKey: contendrá la llave con la que se codificara y decodificara.

El método encripta: es el encargado de encriptar el texto, devolviendo como resultado el texto encriptado.

El método desencriptar: recibe como parámetro el texto encriptado y devuelve el texto desencriptado.

El método HexToString: el mismo recibe un arreglo de byte encriptado y retorna un texto.

El método StringToHex: recibe un texto y retorna un arreglo de byte.

GestionCDA: clase que se encarga de almacenar los ficheros CDA encriptados en el FTP así como obtenerlos una vez que estos sean desencriptados para su lectura.

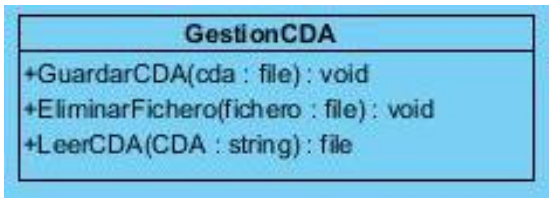

Ilustración 9 Clase GestionCDA.

La clase se compone de los siguientes métodos GuardarCDA el cual recibe un fichero CDA el que será encriptado y almacenado en el FTP.

El método LeerCDA: a este se le pasa el nombre del fichero que se quiere obtener, este lo devuelve ya desencriptado.

El método EliminarFichero: se encargará de borrar el fichero de la dirección local una vez que este haya sido guardado en el FTP.

3.2.5 Diagrama de clases del diseño pertenecientes al paquete Capturador

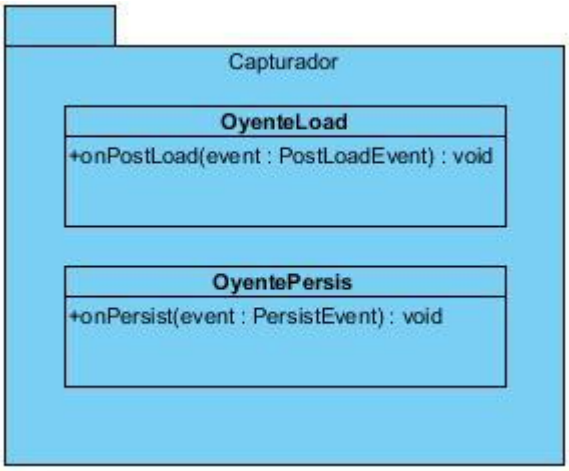

Ilustración 10 Diagrama de clases del diseño pertenecientes al paquete Capturador.

3.2.6 Diagrama de clases del diseño pertenecientes al paquete Cifrado

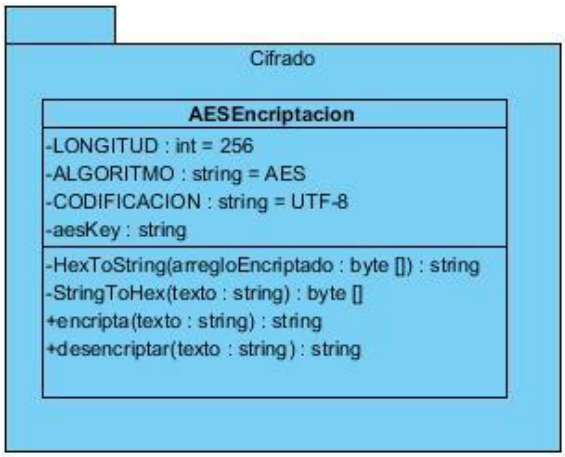

Ilustración 11 Diagrama de clases del diseño pertenecientes al paquete Cifrado.

3.2.7 Diagrama de clases del diseño pertenecientes al paquete Gestión CDA

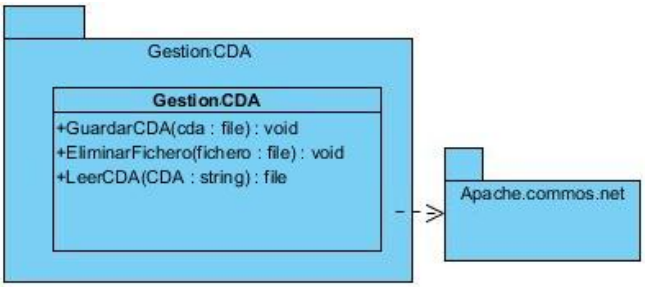

Ilustración 12 Diagrama de clases del diseño pertenecientes al paquete Gestión CDA.

3.3 Implementación

Una vez obtenidos los elementos del modelo del diseño, esto se implementan en términos de componentes, reflejado en el diagrama de componentes.

3.3.1 Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean éstos de código fuente, binarios o ejecutables. Además tiene en consideración los requisitos relacionados con la facilidad de desarrollo, la reutilización y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

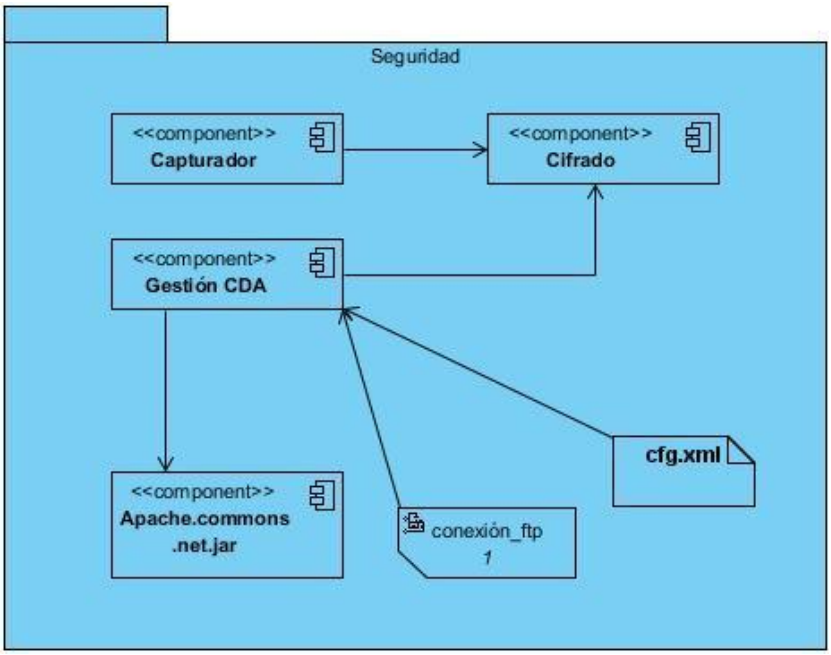

Ilustración 13 Diagrama de componentes.

3.3.2 Estrategias de codificación. Estándares y estilos a utilizar

Los estándares de código son importantes y de obligatorio cumplimiento para los proyectos en nuestra Universidad. Estos comprenden los aspectos de la generación de código y repercuten directamente en la legibilidad y la extensibilidad del código de cualquier software, haciendo que los desarrolladores se acoplen rápidamente al proceso de desarrollo. A continuación se encuentran detallados todos los estándares de codificación que se definieron para la creación de la presente solución teniendo en cuenta el lenguaje de programación. (31)

Clases: los nombres de clases comienzan con mayúsculas y las palabras que la forman en minúsculas, en caso de que sea nombre compuesto siempre comenzarán con mayúscula. Algunas clases comienzan con un identificativo para ubicarlas lógicamente en el paquete. (31)

Atributos de las clases: las variables miembros de clase se escriben con minúsculas, las variables con nombres compuestos, comienzan con la primera palabra enteramente en minúscula y el resto comenzando con mayúscula. Cuando se definan nombres de variables deben hacerse de forma representativa con el propósito de facilitar el entendimiento. (31)

Métodos coherentes: cumplen con las mismas restricciones que las variables. No debe añadírsele demasiada complejidad a los procedimientos, pues si manejan muchas tareas resulta natural que sean difíciles de entender y tengan una alta probabilidad de ocurrencia de errores. (31)

Comentarios: se comentarán los métodos al principio para facilitar la comprensión y entendimiento del código. (31)

Saltos de línea: añadir un salto de línea después del cierre de los paréntesis de los parámetros y después de un punto y coma cuando termina la sentencia. (31)

Longitud de la línea: evitar las líneas de más de 80 caracteres, cuando se supere esta cifra se debe reorganizar el código usando algún principio descrito anteriormente. (31)

3.3.3 Configuración

La solución propuesta precisa la configuración de varios elementos representados en el diagrama de despliegue anterior. La necesidad de asegurar la transmisión de los datos desde un nodo a otro, a través de la red, hace necesario el uso de protocolos de comunicación más seguros a los actualmente utilizados. Esto exigirá la configuración de los servidores de aplicación y base de datos, Jboss Server y PosgreSQL respectivamente, para que soporten dichos protocolos.

Los nuevos protocolos a utilizar son HTTPS y SSL. El empleo de estos requiere del uso de certificados digitales, como ya se había comentado en la descripción de la propuesta de solución. Con el fin de crear dichos certificados, se utiliza la herramienta Keytool.

Generar certificado digital utilizando Keytool:

Para generar un certificado digital auto-firmado en sistemas Unix se accede a través de la consola de comandos a la herramienta Keytool, la cual pedirá una serie de datos necesarios para la creación del mismo. En el proceso es introducido el nombre de la unidad de organización, la organización, ciudad o

localidad, estado o provincia y código del país. Además de una contraseña para el certificado digital, como también la del almacén de llaves. Es imprescindible que dichas contraseñas sean idénticas.

```
root@edder-pc:/home/edder# keytool -genkey -alias certificados -keyalg RSA -keystore almacen_de_llaves -validity 365
Escriba la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
  [Unknown]:  His
¿Cuál es el nombre de su unidad de organización?
  [Unknown]:  Hospitales
¿Cuál es el nombre de su organización?
  [Unknown]:  gehos
¿Cuál es el nombre de su ciudad o localidad?
  [Unknown]:  Habana
¿Cuál es el nombre de su estado o provincia?
  [Unknown]:  Habana
¿Cuál es el código de país de dos letras de la unidad?
  [Unknown]:  cu
¿Es correcto CN=His, OU=Hospitales, O=gehos, L=Habana, ST=Habana, C=cu?
  [no]:  si

Escriba la contraseña clave para <certificados>
        (INTRO si es la misma contraseña que la del almacén de claves):
Volver a escribir la contraseña nueva:
root@edder-pc:/home/edder#
```

Ilustración 14 Generación de un certificado digital desde la consola.

Configuración del server para escuchar peticiones HTTPS:

El certificado digital creado debe ser copiado al directorio jboss/server/defaul/conf. Por defecto el servidor escucha peticiones HTTP por el Puerto 80, y no las conexiones SSL por el puerto 443. Lo que hace necesario habilitar el soporte SSL. Mediante la modificación del archivo server/defaul/ que se encuentra en la dirección deploy/jboss-web.deployer/server.xml el cual debe quedar de la siguiente forma:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    maxThreads="150" scheme="https" secure="true"
    clientAuth="false"
    strategy="ms"
    address="${jboss.bind.address}"
    keystoreFile="${jboss.server.home.dir}/conf/almacen_de_llaves"
    keystorePass="contraseña"
    trustStoreFile="${jboss.server.home.dir}/conf/almacen_de_llaves"
    trustStorePass="contraseña"
    sslProtocol="TLS"/>
```

Ilustración 15 Configuración del archivo server.xml.

PostgreSQL y SSL

Para realizar conexiones seguras a las bases de datos se utiliza generalmente SSL. En la activación de este soporte, el sistema debe cumplir los requerimientos no funcionales RNF6 y RNF9. Además de definir el parámetro SSL en la configuración de PosgreSQL postgresql.conf. Al mismo tiempo, debe ser instalado el certificado digital autofirmado del servidor de base de datos, que contará con la clave privada correspondiente.

Una vez instalado y activado el soporte SSL se debe configurar en el fichero pg_hba.conf para cifrar el tráfico y/o autenticación de los nodos con certificados digitales. Como las conexiones son internas y locales, no hace falta incurrir en gastos por pago a una Autoridad de Certificación (CA, por sus siglas en inglés), para que emita y firme nuevamente los certificados. Este puede ser generado y firmado por una CA local.

Para completar lo antes descrito es necesario generar un certificado y llave que se utilizarán para firmar los certificados que se vayan a utilizar en la solución. También es importante generar el certificado y la llave que el servidor PostgreSQL 8.4 necesita. El certificado debe ser firmado por la CA local.

3.3.4 Diagrama de despliegue

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. (19) Muestra las relaciones físicas entre los componentes de hardware y software, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software.

La aplicación está distribuida en cuatro nodos: uno de ellos es el servidor de aplicaciones JBoss 4.2, al cual estarán conectados los tres restantes. Las PCs clientes, las cuales pueden tener instalado cualquier sistema operativo, se conectan a través del protocolo de comunicación segura HTTPS. Los otros dos, el servidor de base de datos Postgresql-server 8.4 y el servidor FTP en su modalidad VSFTPD el que estará desplegado sobre Linux lo harán mediante el protocolo SSL.

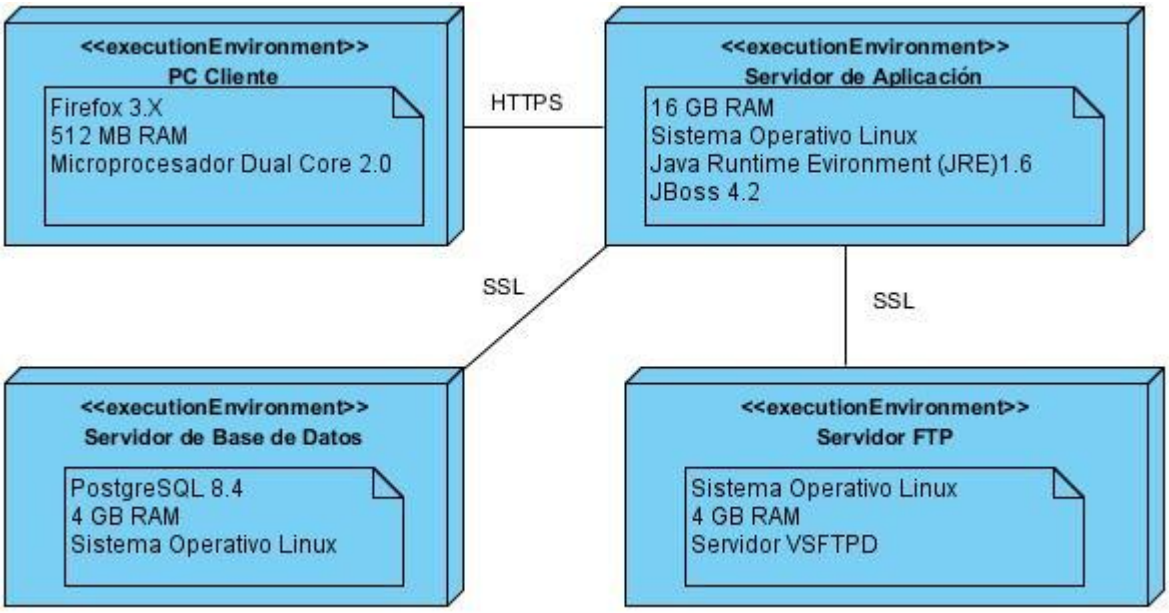

Ilustración 16 Diagrama de despliegue.

Conclusiones.

Durante el transcurso del capítulo se argumentaron los fundamentales aspectos que se llevan a cabo en el proceso de análisis de la solución propuesta. Además se describió la arquitectura, así como ficheros de configuración utilizados en el entorno de la implementación para garantizar una ejecución eficaz del componente. También se plasmó el diseño de clases para una mejor comprensión de cómo está estructurada la propuesta de solución.

Conclusiones

El estudio de las tendencias actuales de la seguridad informática en aplicaciones web, permitió identificar a HTTPS y SSL como protocolos que incrementen la seguridad durante la transmisión de los datos a través de la red. Además se determinó que para su aplicación en el sistema alas HIS lo mas apropiado es el empleo de certificados autofirmados.

La investigación realizada sobre los perfiles de seguridad que propone IHE, permitió conocer las especificidades de los mismos. Así como identificar algunos sistemas sanitarios que implementan dichos perfiles, demostrando que el uso de estos es viable y necesario en el desarrollo de aplicaciones para la salud. Además se puede concluir que los perfiles ATNA, EUA XDS aportan confidencialidad en la transmisión y almacenamiento de los datos.

El análisis hecho entre los algoritmos de encriptación simétricos estudiados, dió como resultado que el algoritmo más apropiado para el cifrado de datos es el AES, por ser el más seguro, de los de su tipo, usado internacionalmente.

El estudio que se hizo sobre la forma de implementar un RDE, permitió detectar el servidor VSFTPD como el más fácil y rápido de desarrollar.

El empleo de la arquitectura definida permitió el desarrollo de un componente de seguridad con un impacto mínimo en la actual implementación del módulo Admisión del sistema alas HIS.

Se logró desarrollar el componente de seguridad para el módulo Admisión del sistema alas HIS, el cual brinda los requerimientos de seguridad necesarios para garantizar la confidencialidad de la información almacenada y trasmitida.

Recomendaciones

Rediseñar la estructura de clases del sistema alas HIS, con el objetivo de eliminar las repeticiones de la entidad Persona en los distintos módulos del sistema.

Desarrollar una herramienta que permita encriptar los datos almacenados antes de la implantación de la solución propuesta.

Desarrollar mecanismos que certifiquen la autenticidad de un usuario que accede al sistema desde una computadora no acostumbrada.

Referencia bibliográfica

1. Ministerio de Administraciones Públicas. Metodología de Análisis y Gestión de Riesgos de los Sistemas Informáticos. Madrid : s.n. Vol. 1.1 : s.n., 2006.

2. slideshare. [En línea] http://es.slideshare.net/jmorenol/introduccin-a-la-seguridad-informtica.

3. kanteron . [En línea] http://www.kanteron.com/blog/es/products/kanteron-his/.

4. IHE. [En línea] http://www.ihe-e.org/intro.htm.

5. IHE. [En línea] http://www.ihe-e.org/sc/ITI.htm.

6. Vidal Sintes, Ana Elda y Rodriguez Venegas, Yanet. Diseño del Módulo de Economía para la Empresa Nacional. Cuba. La Habana : s.n., 2010.

7. Zayas-Bazán Mola, Yaquelin. La seguridad de la información clínica. Perfiles de seguridad IHE. Cuba. La Habana : s.n., 2012.

8. ALERT Life Sciences Computing. [En línea] http://www.alert-online.com/presentation.

9. GALILEO. [En línea] http://www.dominion.es/Portals/0/images/new/Galileo_Brochure_02_09_ESP.pdf.

10. Rodríguez Espinosa, Diancy. Personalización del sistema eXcriba para el control de los documentos administrativos de la Facultad 7. Cuba. La Habana : s.n., 2012.

11. Pequeño diccionario de términos de informática. [En línea] sites.google.com/site/pepevane/Home/diccionario.doc.

12. dsi servicios. [En línea] 2012. http://www.hardwareyredes.es/2012/servidores-sftp-y-ftps/.

13. alcancelibre . [En línea] http://www.alcancelibre.org/staticpages/index.php/09-como-vsftpd.

14. NORMA DE CONTROLES CRIPTOGRÁFICOS. [En línea] http://www.google.com.cu/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCsQFjAA&url=http%3A%2F%2Fwww2.minedu.gob.pe%2Fseginfo%2Fwp-content%2Fuploads%2F2011%2F09%2FNO-010-Norma-de-controles-criptogr%25C3%25A1ficos-VF.doc&.

15. Descripción del algoritmo DES. [En línea] 1990. http://www.tierradelazaro.com/public/libros/des.pdf.

16. DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO DE LABORATORIO DE CRIPTOGRAFÍA . [En línea] 2005. http://rd.udb.edu.sv:8080/jspui/bitstream/123456789/278/1/036199_tesis.pdf.

17. Investigación de criptografía. [En línea] 2011. http://es.slideshare.net/CesarCuamatzi/criptografia-11362555.

18. Menéndez Samé, Caridad y Martínez Ortega, Heileen. SISTEMA OFIMÁTICO ENCRIPTADO. . Cuba. La Habana : s.n., 2010.

19. Help Center. [En línea] http://help.webmiral.com/article/542?prog_id=484898.

20. Oracle. [En línea] http://www.oracle.com/technetwork/java/index-jsp-140203.html.

21. Hibernate in Action . [En línea] http://www.cpe.ku.ac.th/~plw/oop/e_book/Hibernate_in_action.pdf . ISBN 1932394-15-X .

22. Bauer, Cristian y King, Gavin. Java Persistence with Hibernate. 2005. 1-932394-88-5.

23. Franky, María Consuelo. [En línea] 2007. http://www.acis.org.co/fileadmin/Conferencias/ConfConsueloFranky_Abr19.pdf.

24. Ledo Báster, David Ricardo. Infraestructura de firma y validación digital de los documentos clínicos electrónicos generados por el sistema alas HIS. Cuba. La Habana :s.n : s.n., 2011.

25. Apache Commons. [En línea] http://commons.apache.org/net/.

26. Softpedia . [En línea] http://keytool-iui.softpedia.com/.

27. Servidor FTP con vsftpd . [En línea] http://www.colombialinux.org/?q=node/18.

28. ArPUG Grupo de usuarios PostgreSQL de Argentina. [En línea] http://www.arpug.com.ar/trac/wiki/PgAdmin.

29. ING SOFTWARE. [En línea] http://ingjesussoft.blogspot.com/2012/09/rup-proceso-unico-de-desarrollo-de.html.

30. Desarrollo de funcionalidades que faciliten al docente su preparación y el control del aprendizaje de los estudiantes en la plataforma educativa Zera . [En línea] 2011. http://publicaciones.uci.cu/index.php/SC| seriecientifica@uci.cu.

31. Centeno Díaz, Karina, Rojas Ríos, Danisbel y Solis Mulet, Héctor Manuel. Componente de Seguridad para aplicaciones del Área Temática Sistemas de Apoyo a la Salud. . Cuba. La Habana : s.n., 2008.

32. Torres, Y. M. G. B. y. F. R. Módulo Laboratorio del Sistema de Información Hospitalaria alas HIS. Cuba. La Habana : s.n., 2009.

33. IHE y la historia clínica compartida: XDS. [En línea] 2009. http://www.ihe-e.org/docweb/presentaciones/IntroXDS.pdf.

34. HISTORIA CLÍNICA ELECTRONICA CON CDA. [En línea] http://cgallego.es/resources/cda.pdf.

35. Scielo. [En línea] ACIMED v.14 n.5 , 2006 . http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352006000500009. ISSN 1024-9435.

36. Manual de Usuario IHE-Radiología . [En línea] 2005. http://www.ihe-e.org/sc/radio/Manual%20de%20Usuario%20IHE-Radiologa.pdf.

37. IHE y la historia clínica compartida: XDS. [En línea] 2009. http://www.ihe-e.org/docweb/presentaciones/IntroXDS.pdf.

38. Menéndez Rizo, Javier y Cuadrado Sospedr, Abel Ernesto. Integración de servicios a través de un Sistema de Perfiles Centralizados . Cuba. La Habana : s.n., 2010.

39. Scribd. [En línea] http://es.scribd.com/doc/27519905/19/Desventajas.

40. AUTENTIFICACIÓN. [En línea] http://www.eumed.net/cursecon/ecoinet/seguridad/autentificacion.htm.

41. MEDCICLOPEDIA: DICCIONARIO ILUSTRADO DE TÉRMINOS MÉDICOS. [En línea] http://www.iqb.es/diccio/a/au.htm.

42. JBercero.com . [En línea] 2009 . http://jbercero.com/index.php?option=com_content&view=article&id=68:criptografia-iv-conceptos-basicos&catid=41:criptografia&Itemid=67..

43. Roberto Gómez Cárdenas . [En línea] http://www.cryptomex.org.

44. Criptoanálisis diferencial . [En línea] https://www.ccn-cert.cni.es/publico/serieCCN-STIC401/es/c/differential_cryptanalysis.htm 50.

45. Criptoanálisis lineal. [En línea] https://www.ccn-cert.cni.es/publico/serieCCN-STIC401/es/c/linear_cryptanalysis.htm 50.

46. SISTEMAS OPERATIVOS UNIX . [En línea] http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/MonogSO/COMUNUNIX99.htm.

47. ¿Que es el TCP/IP? - Definición de TCP/IP . [En línea] http://www.masadelante.com/faqs/tcp-ip..

48. Protocolo SSH . [En línea] http://www.gb.nrao.edu/pubcomputing/redhatELWS4/RH-DOCS/rhel-rg-es-4/ch-ssh.html.

49. Kioskea.net . [En línea] http://es.kioskea.net/contents/300-lenguajes-de-programacion-ndash-api.

50. Webopedia . [En línea] http://www.webopedia.com/TERM/J/J2EE.html.

53. VISUAL AGE for JAVA . [En línea] http://www.ub.edu.ar/catedras/ingenieria/ing_software/ubftecwwwdfd/java/java.htm.

52. Desarrolloweb.com. [En línea] http://www.desarrolloweb.com/wiki/programacion-orientada-a-objetos.html.

Bibliografía

- Zayas-Bazán Mola, Yaquelin. *La seguridad de la información clínica. Perfiles de seguridad IHE.* Cuba. La Habana : s.n., 2012.
- Ledo Báster, David Ricardo. *Infraestructura de firma y validación digital de los documentos clínicos electrónicos generados por el sistema alas HIS. .* Cuba. La Habana : s.n., 2011.
- Introducción a IHE. [En línea] Lamata, Pablo, 6 de octubre de 2006. http://www.ihe-e.org/docweb/presentaciones/IntroIHE.ppt.
- Recuperación de información . [En línea] http://recuperandoinformacion-jupa.blogspot.com/2011/05/seguridad-de-documentos-electronicos.html.
- Introducción . [En línea] http://www.ihe-e.org/intro.htm.
- Historia de la computación . [En línea] 2008. http://hcprogramasdecomputo.blogspot.com/2008/06/evolucin-de-la-seguridad-informtica.html.
- Griensu, tecnologia y servicios para la salud. [En línea] http://www.griensu.com.
- Galileo. [En línea] http://www.dominion.es/Portals/0/images/new/Galileo_Brochure_02_09_ESP.pdf.
- Seguridad en la Web. [En línea] Espinosa, Eduardo. http://www.espina.info/papers/seguridadenlaWeb.pdf.
- IEEE Xplore. [En línea] http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4768661&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4768661.
- ALERT Life Sciences Computing. [En línea] http://www.alert-online.com.
- Protocolo de tranferencia de archivos. (FTP) . [En línea] Ángel Luis Cobo Yera, Diviembre de 2009.http://www.csi-csif.es/andalucia/modules/mod_ense/revista/pdf/Numero_25/ANGEL_LUIS_COBO_2.pdf. ISSN 1988-6047 .

- Mecanismo de seguridad de la informática en aplicaciones web. (2006). Obtenido de http://zarza.usal.es/~fgarcia/doctorado/iweb/05-07/Trabajos/SeguridadAppWeb.pdf

Anexos

Anexo 1. Descripción del perfil ATNA.

El registro de auditoría y autenticación del nodo (Atna) Perfil de Integración establece las medidas de seguridad que, junto con la política de seguridad y procedimientos, la confidencialidad de la información del paciente, integridad de datos y la rendición de cuentas de usuario. Contribuye al control de acceso mediante la limitación del mismo a la red entre los nodos, y la limitación del acceso a cada nodo de los usuarios autorizados. La red de comunicaciones entre los nodos de seguridad en un dominio seguro se limita sólo a los otros nodos seguros en ese dominio. Los nodos seguros limitan el acceso a usuarios autorizados según lo especificado por la autenticación local y la política de control de acceso. (7)

Descripción Funcional: Define las medidas de seguridad que garantizan la integridad, confidencialidad y registro de accesos a la información de los pacientes. (7)

En el perfil ATNA se detallan las medidas de seguridad básicas que deben implementar los nodos de una red perteneciente a una institución hospitalaria, así como las garantías que deben brindar para ser utilizados como parte de un ambiente médico seguro y privado. El objetivo de ATNA es asistir a los administradores de cada sistema a implementar las políticas de seguridad y confidencialidad necesarias. (7)

Tabla 8 Actores y transacciones del perfil ATNA.

Actores	Transacciones	Opcionalidad
Audit Record Repository	Record AuditEvent	Requerido
SecureNode	AuthenticateNode	Requerido
	Record AuditEvent	Requerido
	Maintain Time	Requerido
SecureApplication	AuthenticateNode	Opcional
	Maintain Time	Opcional
	Record AuditEvent	Opcional

Los sistemas que implementen el perfil deberán tener las siguientes características: (7)

- Autenticación de usuario. ATNA solo requiere autenticación de usuario local, permitiendo utilizar la tecnología de control de acceso que el nodo defina, por ejemplo EUA y XUA son candidatos.
- Control de acceso entre nodos de una red limitando el acceso a nodos únicamente a usuarios autorizados.
- Autenticación bidireccional de nodos en la comunicación entre ambos, mediante la utilización de certificados digitales.
- Integridad y confidencialidad de los datos intercambiados entre los nodos.
- Seguimiento de usuarios para determinar acciones mal intencionadas sobre la información protegida.
- TCP/IP Transport Layer Security Protocol (TLS) para la autenticación del nodo y cifrado (opcional). Si se dispone de comunicaciones entre servicios web, ATNA permite la implementación de mecanismos de seguridad punto a punto definidos en el estándar WS-Security que cumplan con el perfil básico WS-I.
- El actor debe ser capaz de configurar la lista de certificados de nodos autorizados.
- El sistema deberá registrar eventos de aplicación en un repositorio centralizado.
- La comunicación con el repositorio de eventos podrá ser mediante el protocolo Syslog sobre UDP.

Anexo 2. Descripción del perfil EUA.

Define un medio para establecer un nombre de usuario que se puede utilizar en todos los dispositivos y software que participan en este perfil de integración. El mismo facilita enormemente la gestión centralizada de autenticación de usuario y proporciona a los usuarios la comodidad y la velocidad de un inicio de sesión único. Este perfil aprovecha Kerberos y el estándar HL7 CCOW. La autenticación de usuarios es un paso necesario para la mayoría de las aplicaciones y las operaciones de acceso a datos y constituye una mejora del flujo de trabajo para los usuarios. El perfil IHE EUA agrega valor a la especificación CCOW por el tema de usuario por indicación del objeto de usuario y CCOW sufijo de usuario. Este perfil no tiene en cuenta las características de seguridad tales como pistas de auditoría, control de accesos, gestión de autorizaciones y PKI. El medio ambiente se supone que es una sola empresa, que se rige por una política de seguridad única y con un dominio de red común. (7)

Descripción Funcional: Define un método para establecer un único identificador por usuario que le permita acceder de forma transparente a cualquier sistema que participa en este perfil de integración. Estándares principales (7)

- Kerberos v5 standard
- HL7 CCOW

Actores y transacciones Un número de transacciones utilizadas en este perfil se ajustan al estándar Kerberos v5, norma que se ha mantenido estable desde 1993, es ampliamente aplicada en los sistemas operativos actuales, ha soportado con éxito los ataques en sus 10 años de historia, y es totalmente interoperable entre plataformas. Por ejemplo, Sun Solaris, Linux, AIX, HPUX, IBM-z/OS, IBM OS400, Novell, MAC OS X y Microsoft Windows 2000/XP todos aplican Kerberos de forma interoperable. (7)

Tabla 9 Actores y transacciones del perfil EUA.

Actores	Transacciones	Opcionalidad
KerberosAuthentication Server	GetUserAuthentication	Requerido
	GetService Ticket	Requerido
ClientAuthenticationAgent	GetUserAuthentication	Requerido
	ClientAuthenticationAgent	GetUserAuthentication

	ClientAuthenticationAgent	GetUserAuthentication
	ClientAuthenticationAgent	GetUserAuthentication
	ClientAuthenticationAgent	GetUserAuthentication
	ClientAuthenticationAgent	GetUserAuthentication
Kerberized Server	KerberizedCommunication	Requerido
UserContextParticipant	JoinContext	Requerido
	FollowContext	Requerido
	LeaveContext	Requerido
Context Manager	JoinContext	Requerido
	FollowContext	Requerido
	LeaveContext	Requerido
	ChangeContext	Requerido

Anexo 3. Descripción del perfil XDS.

Facilita el registro, distribución y acceso a través de instituciones de salud a registros electrónicos de salud del paciente. El intercambio de documentos se centra en proporcionar una especificación basada en estándares para la gestión de la distribución de los documentos entre una empresa de salud. El Perfil de Integración XDS supone que las instituciones de salud pertenecen a uno o más dominios de afinidad XDS. Un dominio de afinidad XDS es un grupo de empresas de salud que han acordado colaborar utilizando un conjunto común de políticas y compartir una infraestructura común. (33)

Dentro de un dominio de afinidad XDS, ciertas políticas comunes y reglas de negocio deben ser definidas. Estas incluyen: cómo se identifican los pacientes, se obtiene el consentimiento, y el control de acceso, así como el formato, contenido, estructura, organización y representación de la información clínica. Este perfil de integración no define las políticas y normas específicas de negocio, sin embargo, ha sido diseñado para dar cabida a una amplia gama de estas políticas para facilitar el despliegue de infraestructuras basadas en estándares para el intercambio de documentos clínicos del paciente. Todo esto a través de repositorios de documentos y un registro de documentos para crear un registro longitudinal de información sobre un paciente dentro de un determinado dominio de afinidad XDS. Estas son entidades distintas con responsabilidades separadas: (33)

- Un repositorio de documentos es responsable de almacenar documentos de manera transparente, segura, confiable y persistente y la respuesta a documentar las solicitudes de recuperación.
- Un registro de documentos es el responsable de almacenar la información sobre esos documentos. Así, los documentos de interés para el cuidado de un paciente pueden ser fácilmente encontrados, seleccionados y recuperados, independientemente del repositorio donde se almacenan realmente.

Definición Funcional: Permite intercambiar información clínica de diferentes tipos entre distintas organizaciones. (33)

Tabla 10 Actores y transacciones del perfil XDS.

Actores	Transacciones	Opcionalidad
DocumentConsumer	RegistryStoredQuery	Requerido
	RetrieveDocument Set	Requerido

DocumentSource	Provide and RegisterDocument	Requerido
DocumentRepository	Provide and RegisterDocument	Requerido
	RegisterDocument	Requerido
	RetrieveDocument Set	Requerido
DocumentRegistry	RegisterDocument	Requerido
	RegistryStoredQuery	Requerido
	PatientIdentityFeed	Opcional
	PatientIdentityFeed HL7v3	Opcional
IntegratedDocumentSource / Repository	RegisterDocument	Requerido
	RetrieveDocument Set	Requerido
PatientIdentitySource	PatientIdentityFeed	Opcional
	PatientIdentityFeed HL7v3	Opcional

Estándares principales

Electronic Business Usinge Xtensible Markup Language (ebXML): Estándar que especifica una arquitectura de registro/repositorio destinada a publicar y permitir el descubrimiento de productos y servicios en cualquier tipo de negocio donde el repositorio almacena cualquier tipo de contenido digital, mientras que el registro almacena los metadatos que describen el contenido digital. Para que XDS funcione se deben consensuar distintas políticas en las entidades que van a inter-operar, estas políticas que deben incluir al menos: (7)

- Definición de la identificación común de pacientes
- Definición de los mecanismos de obtención de consentimiento
- Definición del control de acceso a datos
- Formato, contenido, estructura, organización y representación de la información.

Anexo 5. Proceso de encriptación con llave simétrica.

Ilustración 17 Proceso de encriptación con llave simétrica.

Glosario de término

CDA: La Arquitectura de Documentos Clínico, de HL7, es un estándar basado en XML para el manejo de documentos que especifica la estructura y semántica de documentos clínicos para el propósito de facilitar su intercambio en un entorno de interoperabilidad. (34)

Metadatos: Son los datos que describen o caracterizan a un libro, recurso o página Web. (35)

Actor [IHE]: Sistema o aplicación responsable de cierta información o tareas. Cada Actor soporta un conjunto determinado de transacciones IHE para comunicarse con otros Actores. Un producto de un proveedor puede incluir uno o varios Actores. (36)

Dominio [IHE]: Es un grupo de trabajo de IHE que se dedica a un área clínica en particular (ejemplo: radiología, cardiología, laboratorio o infraestructura de TI). Cada dominio publica un Marco Técnico (Technical Framework, TF). (36)

Connect-a-thon: Es un evento que se celebra anualmente, y que reúne a empresas de toda Europa que quieren validar el correcto funcionamiento de los Perfiles de Integración IHE implementados en sus productos sanitarios. (36)

Transacción [IHE]: Es un intercambio de información entre Actores. El Marco Técnico describe cómo utilizar los estándares establecidos (HL7, DICOM, W3C) para cada transacción en el intercambio de información. (36)

Marco Técnico (TF, por sus siglas en inglés) [IHE]: Es el documento que define los Perfiles de Integración, los problemas y los casos de uso a los que se refieren los Actores y Transacciones que intervienen en ellos. Proporciona también instrucciones precisas de implementación para cada transacción (se utiliza principalmente como guía para empresas proveedoras). (36)

Electronic Business usinge Xtensible Markup Language (ebXML): Estándar que especifica una arquitectura de registro/repositorio destinada a publicar y permitir el descubrimiento de productos y servicios en cualquier tipo de negocio. (37)

XML: son las siglas en inglés de eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el World Wide Web Consortium (W3C). Es un estándar para describir datos y crear etiquetas. Las características especiales son la independencia de datos, o de la separación de los contenidos de su presentación. Es un metalenguaje que permite diseñar un lenguaje propio de etiquetas para múltiples clases de documentos. Los documentos XML se componen de

unidades de almacenamiento llamadas entidades (entities), que contienen datos analizados (parsed) o sin analizar (unparsed). (38)

Servidor: computadora central de un sistema de red que provee servicios y recursos (programas, comunicaciones, archivos, etc.) a otras computadoras (clientes) conectadas a ella. (38)

Cliente: Sistema que establece un intercambio de datos con un servidor. (38)

Servidor de aplicaciones: o servidor web es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML. El programa implementa el protocolo HTTP que pertenece a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa. (39)

Autentificacion: es la comprobación de la identidad de una persona o de un objeto. (40)

Autenticidad: calidad de ser genuino y fiable. (41)

Criptoanálisis: Consiste en comprometer la seguridad de un criptosistema. Esto se puede hacer descifrando un mensaje sin conocer la llave, o bien obteniendo a partir de uno o más criptogramas la clave que ha sido empleada en su codificación. (42)

Criptosistema: Es el conjunto de procedimientos que garantizan la seguridad de la información y utilizan técnicas criptográficas. (43)

Criptografía: Es el conjunto de técnicas o procedimientos que alteran los símbolos de información sin alterar el contenido, convirtiendo a la información modificada en un conjunto de símbolos sin contenido para las partes que no disponen de las técnicas. (43)

Criptoanálisis diferencial: Técnica critoanalítica de tipo estadístico, consistente en cifrar parejas de texto en claro escogidas con la condición de que su producto o-exclusivo obedezca a un patrón definido previamente. Los patrones de los correspondientes textos cifrados suministran información con la que conjeturar la clave criptográfica. (44)

Criptoanálisis lineal: Técnica criptoanalítica de tipo estadístico, consistente en operar o-exclusivo dos bits del texto en claro, hacer lo mismo con otros dos del texto cifrado y volver a operar o-exclusivo los dos bits obtenidos. Se obtiene un bit que es el resultado de componer con la misma operación dos bits de la clave. Si se usan textos en claro recopilados y los correspondientes textos cifrados, se pueden conjeturar los bits de la clave. Cuantos más datos se tengan más fiable será el resultado. (45)

Criptografía de clave privada o simétrica: Son las funciones más clásicas, es decir, se utiliza una determinada clave para encriptar y desencriptar, el problema reside en la necesidad de que todas las partes (receptor y emisor) conozcan la llave. (24) Ver el anexo 5.

Sistema Unix: UNIX es un sistema operativo multitarea y multiusuario, lo cual significa que puede ejecutar varios programas simultáneamente, y que puede gestionar a varios usuarios simultáneamente. (46)

TCP/IP: El Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP, por sus siglas en inglés), es un sistema de protocolos que hacen posibles servicios FTP, E-mail, y otros entre ordenadores que no pertenecen a la misma red. (47)

SSH: El Intérprete de Órdenes Seguras es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente/servidor y que permite a los usuarios conectarse a un host remotamente. Este encripta la sesión de conexión, haciendo imposible que alguien pueda obtener contraseñas no encriptadas. (48)

API: Una API (Interfaz de Programación de Aplicaciones) es un conjunto de funciones que permite al programador acceder a servicios de una aplicación a través del uso de un lenguaje de programación. (49)

J2EE: La Edición Empresarial de Plataforma Java 2 es para desarrollar, construir y desplegar aplicaciones empresariales basadas en Web en línea. Este consta de un conjunto de servicios, APIs y protocolos que proporcionan la funcionalidad para desarrollar aplicaciones de múltiples niveles, basados en la Web. (50)

VisualAge: Es una herramienta de desarrollo rápida de aplicaciones creada por IBM que posee un ambiente de desarrollo integrado a la Programación Orientada a Objetos. Este genera código Java a partir de especificaciones. (51)

Entidad: Es un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos) los mismos que consecuentemente reaccionan a eventos. Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase. (52)