Universidad de las Ciencias Informáticas

Facultad 2

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

Desarrollo de un componente para la autenticación unificada de la solución PACS-RIS

Autores: Hector Manuel Valcarcel Blanco

Yuliet Fernández Lavalle

Tutores: Ing. Luis Eduardo González Abreu

Ing. Yania Junco López

Ciudad de La Habana, 25 de junio de 2015
"Año 57 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los 25 días del mes de junio del año 2015.

Autores:

Hector Manuel Valcarcel Blanco	Yuliet Fernández Lavalle

Tutores:

Ing. Luis Eduardo González Abreu	Ing. Yania Junco López

DATOS DE CONTACTO

Tutores:

Ing. Luis Eduardo González Abreu:

Graduado de Ingeniero en Ciencias Informáticas, egresado de la UCI en el año 2010. Es especialista del Departamento de Desarrollo de Aplicaciones del Centro de Informática Médica de la Universidad de las Ciencias Informáticas. Se desempeña como líder del equipo de desarrollo del sistema RIS. Correo electrónico: legonzalez@uci.cu

Ing. Yania Junco López:

Graduada de Ingeniera en Ciencias Informáticas, egresada de la UCI en el año 2012. Se desempeña como especialista del Departamento de Desarrollo de Componentes del Centro de Informática Médica de la Universidad de las Ciencias Informáticas. Es analista de la solución PACS-RIS. Correo electrónico: yjlopez@uci.cu

RESUMEN

Con el objetivo de mejorar tanto la atención al paciente como las condiciones de trabajo del especialista en la esfera de salud, en la Universidad de las Ciencias Informáticas, el Centro de Informática Médica, desarrolla componentes de una solución PACS-RIS. La solución PACS-RIS es la integración del Sistema de Almacenamiento y Transmisión de Imágenes Digitales con el Sistema de Información Radiológica, estos son capaces de gestionar la información imagenológica del paciente. Dicha solución presenta diferentes mecanismos de autenticación para acceder a cada uno de sus componentes, lo cual dificulta la evolución tecnológica con respecto a la nube, pues la existencia de varias autenticaciones no permite controlar la forma y el lugar de autenticación. Existe un identificador del cliente que tiene un valor fijo para cada componente que lo utilice y es difícil restringir el acceso de un usuario a una determinada aplicación. Además al autenticarse en uno de los componentes el usuario y la contraseña viajan en texto plano, ocasionando problemas graves principalmente de seguridad. Se decide desarrollar un componente para la autenticación unificada de la solución PACS-RIS. Este posibilita unificar la forma de autenticación aplicando un mecanismo híbrido entre los protocolos OAuth 2.0 y OpenID Connect, evitando múltiples cuentas y brindando seguridad con respecto a los mecanismos de autenticación existentes. La aplicación fue desarrollada sobre la plataforma .NET con lenguaje de programación C#, utilizando el Framework 4.5. Para el modelado del sistema se empleó la herramienta Enterprise Architect 7.5.

Palabras clave:

Autenticación, Autenticación unificada, OAuth, OpenID Connect, PACS, RIS.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN

El surgimiento de la informática provocó de forma sistemática el desarrollo de la tecnología, la renovación de los equipos electrónicos y la evolución digital de varias esferas sociales. La medicina en particular ha avanzado con el uso de la tecnología digital, desarrollando así la informática médica. El estudio de los métodos y medios tecnológicos es el objetivo que persigue la informática médica (1). Sus bases son la diversidad de procesos informatizados llevados a cabo mediante distintos protocolos a utilizar, que permiten la comunicación entre los sistemas médicos.

La tecnología y la informática médica están estrechamente relacionadas con la digitalización de los datos y el seguimiento médico a un paciente mediante un sistema de salud; permitiendo, entre otras cosas, brindar calidad de información que asegure una adecuada planificación y gestión de la salud de todos los habitantes. La información que se maneja en dichos sistemas es vulnerable y de carácter privado; por lo que el personal con acceso a los registros sanitarios individuales e institucionales de las unidades de salud debe respetar un código de ética asociado a la seguridad informática.

Dentro del código de ética para la informática médica se encuentran varios principios; entre ellos accesibilidad, privacidad y disposición de la información, transparencia, seguridad, garantía, alternativa menos invasora y responsabilidad (2). Debido a que toda información registrada en los sistemas de salud sobre cualquier paciente es sensible y confidencial, debe ser protegida empleando medidas de seguridad que eviten su pérdida, destrucción, acceso, uso, manipulación, modificación o difusión no autorizada.

Para evitar un acceso no autorizado a la información almacenada en los sistemas de salud es necesario utilizar la autenticación, porque con su empleo se protege la confidencialidad de la información y se garantiza que no sea modificada por cualquier persona. La autenticación detecta y comprueba las credenciales del usuario y valida esas credenciales contra determinada autoridad, permitiendo que la información confidencial utilizada pueda ser tanto obtenida como consultada por las personas asignadas para ello.

La información recogida de forma digital en cada sector de la salud es el factor clave para el trabajo diario del médico o especialista. En la actualidad con el empleo de nuevas tecnologías, esa información se gestiona mediante sistemas informáticos. Dichos sistemas tienen como objetivo

proporcionar, facilitar y ejecutar automáticamente términos que constantemente se realizan de forma manual. Entre los sistemas más empleados dentro del departamento de radiología se encuentran el Sistema de Almacenamiento y Comunicación de Imágenes (PACS por sus siglas en inglés) y el Sistema de Información Radiológica (RIS por sus siglas en inglés).

Para llevar el registro de la actividad del paciente junto al estudio imagenológico que ha sido realizado en el departamento de radiología se utiliza el RIS (3). En diversos centros clínicos y hospitales, estos sistemas se encargan de archivar la documentación de forma digital. Los mismos se utilizan en la elaboración de informes durante el proceso asistencial de un paciente.

El PACS posibilitan el almacenamiento y visualización de imágenes médicas (4). Este sistema a nivel mundial ha demostrado una serie de ventajas funcionales. Está integrado por cuatro componentes principales: estación de visualización, equipos médicos, servidor de imágenes e infraestructura de red; siendo el primero de estos componentes el más importante de un PACS.

La integración de los sistemas PACS y RIS permite a la institución distribuir un conjunto unificado de imágenes, datos y diagnósticos, posibilitando que cualquier usuario autorizado acceda simultáneamente a dicha información cuando lo requiera. En dichos sistemas siempre es vital y necesaria la idea de contar con un enfoque seguro implementando protocolos para la Autenticación, Autorización y Auditoría (AAA por sus siglas en inglés), en aras de garantizar la autorización del usuario a los recursos que tiene acceso y lograr que las acciones que estos realicen en el sistema sean auditables.

Actualmente en Cuba, el Ministerio de Salud Pública (MINSAP) lleva a cabo un proceso abarcador de informatización, que tiene como propósito agilizar los procesos y mejorar tanto la atención al paciente como las condiciones de trabajo de los especialistas. Con este objetivo, en la Universidad de Ciencias Informáticas (UCI) específicamente en el Centro de Informática Médica (CESIM), desarrolla componentes de una solución PACS-RIS.

La solución consta de un RIS, sistema que gestiona los procesos administrativos del departamento de radiología. Entre sus funcionalidades se encuentra la gestión de citas, admisión de pacientes y seguimiento de estudios e información imagenológica de los pacientes. Dicho sistema permite el control de una historia clínica imagenológica, así como las salidas de las estadísticas médicas y las hojas de cargo. Posee un servidor de listas de trabajo que se comunica con los equipos para evitar

incongruencias durante el proceso de realización de los estudios médicos y posibilita realizar búsquedas por pacientes y estudiar sus casos.

Además cuenta con un PACS, sistema que permite conectarse desde diferentes computadoras a un servidor central que almacena imágenes médicas; como son las imágenes de Resonancia Magnética (MR por sus siglas en inglés), Tomografía Computarizada (CT por sus siglas en inglés), Ultrasonidos (US por sus siglas en inglés) y Mamografía (MG por sus siglas en inglés). De igual forma, el PACS posee componentes como reportador (nombrado también como herramienta de edición de informes imagenológicos), estación de diagnóstico general, visor web, servidor de imágenes y servidor de listas de trabajo.

Estos componentes facilitan que queden guardadas dichas imágenes para ser consultadas en cualquier momento. Además de garantizarle a los especialistas el intercambio vía electrónica de sus criterios de diagnóstico, sin necesidad de que el paciente vaya de un especialista a otro.

La solución PACS-RIS presenta diversas maneras de autenticación, tiene un mecanismo de autenticación para el visor web, el servidor de imágenes y otro de forma compartida con el RIS y el reportador. Presenta otros componentes que no permiten la autenticación como son el Gateway, servidor de listas de trabajo y la estación de diagnóstico general. Esto dificulta la evolución tecnológica con respecto a la nube, ya que la existencia de varias autenticaciones no permite controlar el lugar y la manera de autenticación.

La forma de autenticación del RIS junto al reportador se encuentra contaminada por elementos de negocio, imposibilitando su extensión al resto de la solución PACS-RIS. Esta solución está desarrollada en una comunicación basada en el Protocolo Simple de Acceso a Objetos (SOAP por sus siglas en inglés) (5), un protocolo de comunicación de la web que cada vez se encuentra más en desuso, sobre todo para las aplicaciones de Internet, entre ellas varias de la solución PACS-RIS. Dicho protocolo ha sido remplazado por la Notación de Objetos JavaScript (JSON por sus siglas en inglés) (6) por ser un formato más ligero.

Al autenticarse en el reportador el usuario y la contraseña viajan en texto plano. Ello puede ocasionar problemas graves principalmente de seguridad, ya que las credenciales del usuario serían descubiertas fácilmente por cualquier atacante. Para extender el uso de este mecanismo de autenticación, existe un identificador del cliente que tiene un valor fijo para todos los componentes

que lo utilicen. Esto imposibilita restringir el acceso de un usuario a una determinada aplicación, permitiendo que un atacante pueda registrarse en el sistema y hacer uso del mismo.

La diversidad que existe en los procesos de autenticación de la solución PACS-RIS provoca que el administrador de estos sistemas, el especialista y el paciente se enfrenten a una situación engorrosa con el empleo de dicha solución. Para el administrador del sistema los distintos tipos de autenticación representan un empleo desmedido de su tiempo de trabajo, porque tiene que realizar sus acciones o actualizaciones de forma independiente en cada componente de la solución PACS-RIS.

Con respecto al especialista, es una molestia porque tiene que recordar varias contraseñas para acceder a cada uno de los componentes. Al intentar ingresar incorrectamente varias veces la contraseña puede bloquear su cuenta de inicio, dependiendo posteriormente del administrador para que éste resuelva el problema ocurrido. Esta situación podría ocasionar descontento en el proceso de atención del paciente, debido a una demora innecesaria por olvido de una de las contraseñas. La situación planteada provoca errores de autenticación afectando el flujo de trabajo, elemento que puede ser determinante en la institución donde se encuentra implantado el sistema, sobre todo cuando dicha institución tiene como premisa fundamental brindar servicios sanitarios con la asistencia adecuada.

Por lo antes planteado se identifica como **problema a resolver**: ¿Cómo garantizar el acceso único a los diferentes sistemas que componen la solución PACS-RIS?

Este problema define el **objeto de estudio**: Los mecanismos de autenticación. Enmarcado en el **campo de acción**: Los mecanismos de autenticación unificada.

Para la solución del problema se propone como **objetivo general**: Desarrollar un componente para la autenticación unificada de la solución PACS-RIS.

Para dar cumplimiento al objetivo planteado se trazan las siguientes **tareas investigativas**:

1. Análisis crítico y valorativo de las tendencias actuales en la autenticación en los sistemas de salud para definir requisitos del componente.
2. Caracterización del protocolo a utilizar para la autenticación de la solución PACS-RIS.

3. Asimilación de las herramientas, tecnologías y metodología empleadas para el desarrollo de la solución PACS-RIS.
4. Generación de los artefactos propuestos por la metodología de desarrollo Proceso Unificado de Desarrollo de Software (RUP).
5. Implementación del componente de software aplicando las pautas de diseño y siguiendo lo establecido en la Especificación de Requisitos de Software.

Para desarrollar la presente investigación se emplearon los métodos del nivel teórico. A continuación se enuncian los utilizados (7):

Histórico Lógico: para investigar la evolución y utilización de los protocolos de autenticación. Admitirá realizar el estudio del estado del arte de la problemática planteada mediante el análisis de diversas soluciones existentes.

Inductivo Deductivo: para evaluar la forma de unificar el proceso de autenticación para acceder a la solución PACS-RIS y definir los elementos característicos, que permitan proponer un componente que unifique la forma de autenticación.

Analítico y Síntesis: para descomponer el problema de la investigación en elementos concretos de la solución. Analizando los casos de forma particular y determinando los elementos importantes a tener en cuenta durante la selección de los protocolos a utilizar.

Modelación: con el fin de crear un componente de software que permita manejar globalmente y de forma segura la autenticación de los sistemas que componen la solución PACS-RIS basado en un protocolo de autenticación. Para confeccionar los modelos y diagramas asociados al desarrollo del componente.

Los resultados esperados con el desarrollo del componente para la autenticación unificada de la solución PACS-RIS son:

- Eliminará la necesidad de tener un sistema de autenticación en cada una de los componentes de la solución PACS-RIS.
- Descartará los problemas típicos relacionados con las restricciones de acceso por olvido de contraseña.
- Anulará la multiplicidad de cuentas en los componentes.

- Permitirá la administración de las aplicaciones desde una cuenta única.
- Proporcionará seguridad y confiabilidad en los mecanismos de autenticación y autorización con respecto a los sistemas actuales.

Estructuración por capítulos:

Capítulo 1: Fundamentación Teórica del Componente para la autenticación unificada de la solución PACS-RIS: Aborda conceptos de autenticación relacionados con los sistemas de información médica. Se describen los protocolos de autenticación en cuanto a su utilización y evolución, definiendo cuáles de ellos pueden ser relevantes para obtener resultados satisfactorios en la solución del problema planteado. Se presenta el estado del arte del tema a tratar tanto a nivel internacional como nacional. Se precisa la metodología, lenguajes y tecnologías utilizadas, así como las herramientas usadas para la solución propuesta.

Capítulo 2: Características del Componente para la autenticación unificada de la solución PACS-RIS: Como parte de la propuesta de solución, se describen las características del componente a desarrollar, la misma recoge la unión de dos protocolos de autenticación a utilizar. Se muestra el modelo de dominio, las características y funcionalidades que tendrá el sistema a partir de los requerimientos funcionales y no funcionales del componente de software a desarrollar. Se presenta el diagrama de casos de uso del sistema con las correspondientes especificaciones de los casos de uso asociados al componente.

Capítulo 3: Análisis y Diseño del Componente para la autenticación unificada de la solución PACS-RIS: Se presentan los elementos de diseño para el desarrollo de un componente para la autenticación unificada de la solución PACS-RIS. Se conforman los diagramas de clases del análisis y del diseño de los casos de uso definidos en el capítulo anterior. Se realiza una descripción de la arquitectura del sistema y de los patrones tanto arquitectónicos como de diseño.

Capítulo 4: Implementación del Componente para la autenticación unificada de la solución PACS-RIS: Describe todo lo relacionado con el flujo de trabajo de implementación. Aborda la fase de implementación, el diagrama de componentes y de despliegue referente al componente. Se muestran los estándares de codificación utilizados y se explica el tratamiento de errores.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DEL COMPONENTE PARA LA AUTENTICACIÓN UNIFICADA DE LA SOLUCIÓN PACS-RIS

En el presente capítulo se presenta una breve descripción de la autenticación de la solución PACS-RIS. Se dan a conocer conceptos de autenticación relacionados con los sistemas de información médica. Además se exponen las principales características de las tecnologías y lenguajes utilizados para la implementación de la solución propuesta. Se argumenta la selección de las herramientas y metodologías a utilizar.

1.1. Sistemas de salud

Un sistema de salud o sistema de servicios de salud es el que específicamente acepta y afronta la responsabilidad de mantener o mejorar la situación de salud de la población, aunque no sea siempre el único (8). Un sistema de salud abarca todas las organizaciones, las instituciones y los recursos de los que emanan iniciativas cuya principal finalidad es mejorar la salud. Ejemplo: Sistema Nacional de Salud Cubano.

La salud se considera un derecho de todo ciudadano, que cada estado puede y debe garantizar. Sin embargo la forma de atención en la salud varía de un país a otro, dependiendo de las particularidades de cada país. De acuerdo con el tipo de intervención del estado en la atención médica, existen varias clasificaciones; los principales tipos de programas públicos (seguro social), la asistencia pública y el servicio universal (8).

Los sistemas de salud presentan gran relación con la informática médica, ambos se enfocan en el proceso de atención a la población. Mientras que los sistemas informáticos permiten atenciones médicas a larga distancia, ya sea en una misma ciudad o entre países. Los sistemas informáticos de salud incluyen no solo los ordenadores, sino también guías de práctica clínica, terminología médica formal, y de sistemas de información y comunicación (8). Ejemplo de sistemas informáticos o de información de salud son: los PACS y los RIS.

1.1.1. Sistema de Información Radiológica

Un RIS tiene la responsabilidad de gestionar la actividad clínica y administrativa de los departamentos de diagnóstico por imágenes, manejar la información demográfica de los pacientes, programar las citas y la entrega de reportes de diagnóstico. Este sistema informatiza toda la actividad radiológica de un paciente, desde la petición del estudio, al informe del mismo, pasando por la

recogida de las incidencias y consumos de materiales que conlleva la realización de dicha exploración (9). Permite realizar búsquedas por pacientes, estudios y diagnósticos médicos, proporcionando al personal que lo utiliza una mejor organización de la información que se maneja.

1.1.2. Sistema de Almacenamiento y Visualización de Imágenes Médicas

Un PACS está compuesto por cuatro componentes fundamentales, los equipos de adquisición de imágenes médicas, las estaciones de visualización, los servidores de almacenamiento y la infraestructura de red que es capaz de interconectar los elementos anteriormente mencionados. En las estaciones de visualización se brindan un grupo de técnicas digitales para mejorar la calidad de la imagen. Estas funcionalidades incluyen aplicar brillo y contraste a la imagen, ampliar y resaltar regiones de interés, así como realizar un conjunto de transformaciones espaciales, posibilitando de esta manera un aumento de la capacidad de diagnóstico. Utilizar el sistema PACS posibilita la reducción del costo por la utilización de las placas radiológicas o por el daño o la pérdida de información y mejora la accesibilidad de las imágenes debido a que las agrupa en una misma base de datos (10).

1.1.3. Solución PACS-RIS

La integración de los sistemas PACS y RIS, ofrecen a corto plazo, de forma muy rápida y sencilla importantes mejoras (4, 9). Dicha integración es una solución tecnológica aplicada a la salud. El RIS proporciona al PACS toda la información sobre las citas existentes, esto implica que cualquier estudio almacenado en el PACS ha de tener una cita previa en el RIS. A su vez el PACS notifica al RIS que el estudio ha sido realizado y completado, luego el RIS envía una copia al PACS y la notificación de que el informe ha sido realizado (11).

1.2. Control de acceso

El control de acceso es el proceso de conceder permisos a usuarios o grupos para acceder a objetos tales como ficheros o documentos. El control de acceso se basa en tres conceptos fundamentales: identificación, autorización y autenticación (12). Con el uso de estos tres principios un administrador del sistema puede controlar qué recursos están disponibles para los usuarios de un sistema.

1.2.1. Identificación

La identificación es el proceso a través del cual un usuario muestra su identidad, presenta un conjunto de datos o cualidades que lo caracterizan (13). Una de las posibles técnicas para

implementar la identificación es la utilización de un servidor de autenticaciones sobre el cual los usuarios se identifican, y que se encarga luego de autenticarlo sobre los restantes equipos a los que éste pueda acceder. La identificación normalmente es conocida como nombre de usuario, ella es quien dice al sistema quién es el usuario.

1.2.2. Autorización

La autorización es el hecho de que las entidades autorizadas a tener acceso a los recursos de cómputo, tengan acceso únicamente a las áreas de trabajo sobre las cuales ellas deben tener dominio (14). Generalmente tiene que ver con la asignación máxima de recursos del sistema a un usuario dado. Cuando varios usuarios acceden a una computadora a la vez, el administrador debe tener en cuenta la capacidad del equipo y establecer las listas de acceso de tal manera que el servicio que provee dicho equipo no disminuya la calidad por causa de la saturación (15).

En el caso de los datos, la autorización debe asegurar la confidencialidad e integridad de los mismos, ya sea concediendo o denegando el acceso a su lectura, modificación, creación o eliminación. Por otra parte, solo se autoriza a acceder a un recurso a aquellos usuarios que lo necesiten para hacer su trabajo, de lo contrario será negado.

1.2.3. Autenticación

La autenticación verifica la identidad del usuario, generalmente cuando entra en el sistema, la red, o accede a una base de datos (15). La información obtenida durante la autenticación puede ser utilizada directamente por el código.

Existen muchas formas de autenticación entre el usuario y un sistema. Es posible autenticarse mediante una contraseña, una tarjeta magnética o las huellas digitales. La utilización de más de un método a la vez aumenta las probabilidades de que la autenticación sea correcta. La decisión de adoptar más de un modo de autenticación debe estar en relación al valor de la información a proteger (15). El uso de contraseñas es la forma más utilizada, éste método será mejor o peor dependiendo de las características de la contraseña. En la medida que la contraseña sea más grande y compleja para ser adivinada, más difícil será burlar esta técnica.

El Protocolo de Autenticación Extensible (EAP por sus siglas en inglés) es un marco que permite métodos de autenticación para la conexión a Internet, las redes y los servidores de autenticación. Con la autenticación EAP, tanto el cliente de acceso a redes como el autenticador deben admitir el

mismo tipo de EAP para que la autenticación se lleve a cabo correctamente (16). Dicho marco es utilizado con las redes inalámbricas y ayuda a restringir a los intrusos informáticos el acceso a determinadas redes y conexiones.

El Mecanismo de Autenticación Desafío-Respuesta (CRAM por sus siglas en inglés) desafía al usuario a teclear ya sea un nombre de usuario y contraseña, o simplemente una contraseña. Este desafío consiste en la presentación de una ventana emergente que solicita la información necesaria cuando se accede al sitio web. Puede ser limitado a sólo una página dentro de un sitio web o un directorio completo (16).

La autenticación de máquina se utiliza para demostrar que un equipo determinado está autorizado a acceder a una red. Esto se hace generalmente usando Capa de Conexión Segura (SSL por sus siglas en inglés). La autenticación puede hacer mantenimiento de la red y las actualizaciones de servicio más fácil, así como la mejora de la seguridad en general. Cuando una máquina intenta conectarse a una red o acceder a un servidor, el sistema de autenticación de máquina comprueba el equipo para los scripts instalados previamente, la seguridad u otros códigos. Los nombres de usuario y las contraseñas pueden ser conocidos con bastante facilidad, pero los códigos de autorización de la máquina son más complicados y difíciles de evitar.(18)

Por otra parte existen distintos tipos de autenticaciones web. Sin embargo, seleccionar uno de ellos depende de la confidencialidad de la información en el sitio web, teniendo en cuenta el control que se desea ejercer sobre los miembros que pueden acceder a ella.

La autenticación básica del Protocolo de Transferencia de Hipertexto (HTTP por sus siglas en inglés), es el tipo más simple de autenticación web (19). Esta forma de autenticación solicita al usuario que inicie sesión con su nombre de usuario y contraseña. Sin embargo, la información enviada no es segura y podría ser interceptada por un tercero.

La diferencia primaria entre la autenticación básica HTTP y Digest HTTP es que la conexión de esta última se realiza de una manera segura (19). Esto es debido a que las contraseñas son almacenadas en la base de datos de usuario en un formulario cifrado. Nadie puede abrir la base de datos para conocer la contraseña al mirar la secuencia cifrada. De esta manera, la integridad de la contraseña es mucho más segura, ya que sólo puede ser leída por el servidor web.

Otra manera de autenticar sistemas web es la autenticación cliente Protocolo Seguro de Transferencia de Hipertexto (HTTPS por sus siglas en inglés). Un HTTPS se logra cuando un HTTP es combinado con la Capa de Conexión Segura (SSL por sus siglas en ingles). Todo lo contenido dentro del SSL opera en un circuito cerrado, sin ninguna interferencia exterior. Esto permite que el navegador web verifique la legitimidad de cada página que se encuentra en el sitio web mediante la lectura del Certificado de Clave Pública (PKC por sus siglas en inglés) o Criptografía asimétrica del SSL, y comparándolo con el archivo guardado del certificado de seguridad del sitio (19).

HTTPS es ampliamente utilizado en cualquier lugar en el que se accede a información confidencial. Esta forma de autenticación proporciona un alto protocolo de seguridad, porque cada operación de intercambio de información entre el servidor y el navegador está cifrada y se envía a través de un canal seguro (19).

La autenticación basada en formularios por otra parte se utiliza para recoger información de los visitantes que no tienen una necesidad elevada de seguridad. Las encuestas, los formularios de contacto y las páginas de registro se hacen a menudo mediante dicho método de autenticación (19).

La autenticación puede ser en dos factores, esta evita la necesidad de bloques en las instalaciones de infraestructura en cuanto a problemas de terceros. Se utiliza para proteger los datos y los recursos de la empresa en la red y disminuye el costo de operación además de proporcionar fiabilidad y escalabilidad (20).

La autenticación de dos factores se basa en algo que conoce (una contraseña o PIN) y algo que tiene (un autenticador) (21), que proporciona un nivel de autenticación de usuario mucho más fiable que las contraseñas reutilizables. Los métodos de autenticación tradicionales son simples y tienen una serie de puntos débiles, sin embargo la autenticación de dos factores es más complejo (22).

Compañías de servicios en la red, entre ellas Google, Facebook, Twitter, bancos y proveedores de servicios en la nube utilizan la autenticación de dos factores porque es una medida de seguridad extra que frecuentemente requiere de un código obtenido a partir de una aplicación o un mensaje, además de una contraseña para acceder al servicio (23). La autenticación de dos factores presenta la necesidad de emplear procedimientos adicionales para acceder a las áreas protegidas (24).

1.3. Protocolos de autenticación

Los protocolos de autenticación son trabajados para llevar a cabo una forma de autenticación, los mismos son definidos como reglas que precisan la forma y el momento de realizar una acción de

autenticación (19). En el presente epígrafe se analizarán posibles protocolos a utilizar en el desarrollo del componente.

1.3.1. SAML 2.0

SAML es un protocolo basado en XML para servicios Web que permite el intercambio de información de autorización y autenticación entre diferentes sitios Web. Dicho protocolo ha evolucionado en cuanto a su versión, hoy día presenta la versión definitiva SAML 2.0.

SAML 2.0 describe dos roles de federación: el de proveedor del servicio, que es la entidad que da acceso al usuario a un recurso o aplicación; y el de proveedor de identidad, responsable de la autenticación del usuario. Ambos intercambian mensajes para permitir una única forma de acceso. SAML 2.0 permite que un sitio Web otorgue acceso a un usuario que ya ha sido autenticado en otro dominio. Si el usuario que intente acceder a un sitio web no ha sido autenticado, el sitio vuelve a direccionar su navegador a un servidor de federación local (25, 26).

1.3.2. ID-WSF 2.0

Liberty Identity Web Services Framework (ID-WSF por sus siglas en inglés) es un protocolo para crear y gestionar servicios web. La primera versión de ID-WSF se centró en el protocolo SAML 2.0, lo que provocó que su evolución tenga soporte para SAML 2.0. Su última versión es ID-WSF 2.0, el objetivo es ofrecer a las organizaciones un valor real en las soluciones de gestión de identidad interoperables (27).

ID-WSF 2.0 se basa en utilizar afirmaciones para comunicar información entre los servicios web definidos por la identidad. Permite diversos servicios para los usuarios finales, además de que los consumidores de servicios web se suscriban en avisos automáticos de cambios de proveedor (27). Dicho protocolo proporciona a los ejecutores mayor profundidad de funcionalidad, incluyendo la capacidad de aprovechar los servicios web personalizados.

1.3.3. OAuth 2.0

OAuth es un protocolo abierto, anteriormente a él existían diversas soluciones propietarias para que las aplicaciones externas pudieran acceder a sus datos y/o servicios, por ejemplo, Google ya ofrecía la autorización del cliente. Dichas soluciones han tenido el fin de proveer el acceso de aplicaciones de terceros a recursos restringidos, el propietario de dichos recursos comparte sus credenciales con los mismos. (28)

Sin embargo esto trajo consigo serios problemas y limitaciones, tales como; requerir una contraseña de texto y utilizar servidores para admitir la autenticación de contraseña (creando debilidades de seguridad inherentes a las contraseñas). Las aplicaciones de terceros generaban demasiado acceso a los recursos protegidos de los propietarios, además los propietarios de recursos no tenían posibilidad de restringir la duración o acceso a un subconjunto. Situación que obligaba a revocar el acceso individual de un tercero con el cambio de la contraseña de los mismos (29).

OAuth aborda estos problemas mediante la introducción de una capa de autorización y la separación de los roles del cliente y del propietario del recurso. En lugar de utilizar las credenciales del propietario de los recursos para acceder a los recursos protegidos, el cliente obtiene un token de acceso (29). Los tokens de acceso se expiden a los clientes de terceros por un servidor de autorización, con la aprobación del propietario del recurso. El cliente utiliza el token de acceso para acceder a los recursos protegidos alojados en el servidor de recursos (30).

En la Figura 1.1 se muestra la actualización y expiración del token de acceso:

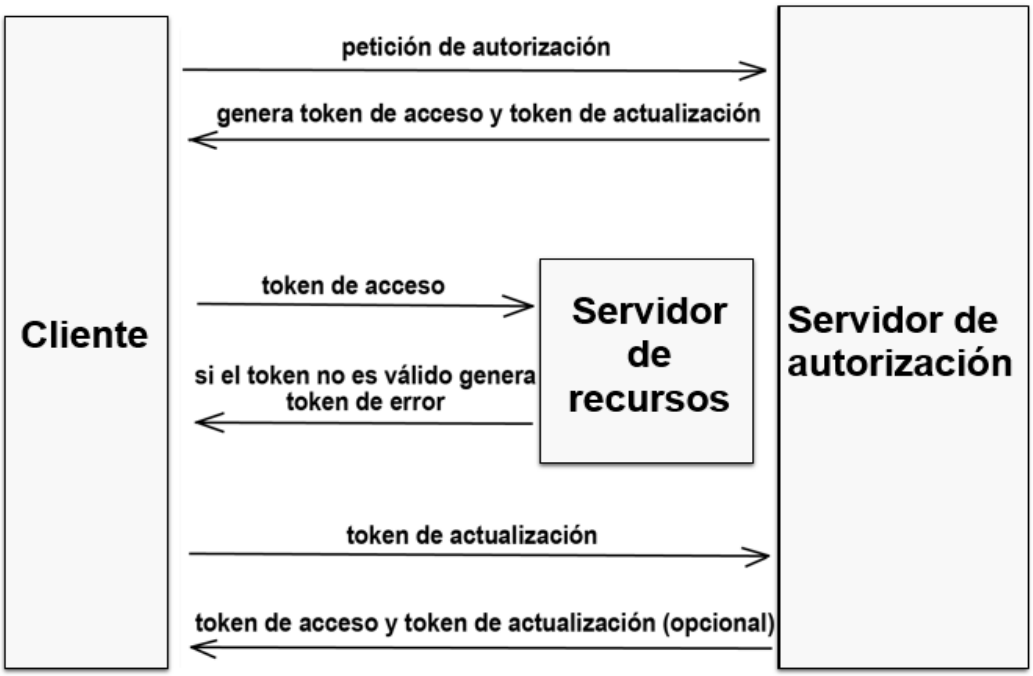


Figura 1.1 Actualización y expiración del token de acceso.
(Fuente: OAuth 2.0 Authorization Framework)

Token de acceso: cadena que denota un ámbito específico, tiempo de vida del acceso y otros atributos de acceso

Para limitar la autorización concedida al cliente por el servidor de recursos se utiliza el scope, este brinda el acceso adecuado a funcionalidades específicas siempre que el token de acceso sea válido. Una vez que el servidor de recursos confirme que el token es válido, OAuth permite realizar la autenticación basada en claims. El claim proporciona atributos de identidad, paquetes, que hacen que el protocolo de autenticación sea más valioso para las aplicaciones desarrolladas. (31)

Para generar la firma necesaria y obtener los recursos protegidos de los usuarios sin necesidad de enviar sus credenciales en cada petición es fundamental conseguir una clave OAuth (conocida como token de acceso) y un secreto de consumidor de OAuth (conocida como token de acceso secreto). Dicho modelo de autenticación presenta importantes inconvenientes, siendo los principales la imposibilidad de autenticar a la aplicación que accede al recurso, la necesidad de que el dueño del recurso comparta con ella sus credenciales (lo cual compromete su seguridad), la imposibilidad de limitar los derechos de acceso de la aplicación con respecto a los del dueño del recurso y la imposibilidad de revocar selectivamente estos derechos a una aplicación en particular (31).

La nueva evolución del protocolo OAuth denominado OAuth 2.0 aborda este problema de manera diferente, es decir, la aplicación cliente utiliza unas credenciales distintas a las del dueño del recurso para acceder al mismo. Estas credenciales consisten en un token de acceso que un servidor de autorización de confianza del servidor de recursos proporciona a la aplicación con la aprobación del dueño del recurso. (32)

Con el uso del protocolo OAuth 2.0 los usuarios no tienen que compartir sus contraseñas directamente en una aplicación, porque éste hace de llave cuando las aplicaciones acceden a los datos de un usuario y puede actuar en su nombre (33). Por ejemplo, los administradores de Google Apps pueden usarlo de doble vía para la delegación de autoridad en todo el dominio. Una aplicación que tiene la clave y el secreto de consumidor de OAuth (algo similar al nombre de usuario y la contraseña de una cuenta de función) puede actuar como cualquier usuario del dominio al acceder a la Interfaz de Programa de Aplicación (API por sus siglas en inglés) de datos de Google (34).

OAuth 2.0 permite la autorización segura de una API (35) de manera simple para aplicaciones de escritorio, móviles y web. Es utilizado de diversas formas por los desarrolladores; para los consumidores es un método de interactuar con datos protegidos y publicarlos mientras que a los

Token de acceso: cadena que denota un ámbito específico, tiempo de vida del acceso y otros atributos de acceso

proveedores de servicio les proporciona a los usuarios un acceso a sus datos al mismo tiempo que protege las credenciales de su cuenta. Es decir OAuth 2.0 permite a un usuario del sitio A compartir su información en el sitio A (proveedor de servicio) con el sitio B (llamado consumidor) sin compartir toda su identidad (33).

OAuth 2.0 utiliza varios tipos de tokens, uno de ellos es JSON Web Token (JWT por sus siglas en inglés), es una codificación de token de seguridad basado en JSON que está diseñado para escenarios de baja conectividad y permite que la información que identifique al usuario se comparta entre los dominios de seguridad (31). El token de referencia también es empleado por OAuth 2.0 y es similar al token de acceso. Tiene como objetivo enviar una ficha de referencia a una API y esta luego recupera la información necesaria del servidor. Dicho token a diferencia del JWT no tiene información de identificación del usuario (30).

OAuth 2.0 define cuatro roles (34), en la Figura 2.1 se muestran dichos roles de manera específica:

Figura 1.2 Roles que define OAuth 2.0.

(Fuente: elaboración propia)

El objetivo de este protocolo de autenticación es poder acceder a recursos protegidos sin la necesidad de enviar el usuario y la contraseña para solicitar los mismos. Este protocolo ofrece

grandes posibilidades de integración entre aplicaciones, sin tener que compartir contraseñas, permitiendo además que el usuario pueda revocar privilegios a las aplicaciones cuando desee (32).

1.3.4. OpenID

El protocolo abierto OpenID cuya primera versión fue definida en 2005, es un protocolo de autenticación federada. Consiste básicamente en que el usuario selecciona un servidor externo (el proveedor de OpenID) que va a ser el que va a validar su identidad en un sistema determinado (el consumidor de OpenID). Dicho servidor utiliza siempre un mismo proveedor de OpenID, el usuario sólo necesita recordar su identificador OpenID y una única contraseña para autenticarse, con una misma identidad, en todos los servicios que acepten OpenID como sistema de autenticación (36).

Sin embargo a OpenID le han sido criticados problemas como complejidad pues no se implementa fácilmente, seguridad ya que es un protocolo muy vulnerable al phishing (consiste en la suplantación maliciosa de una página de autenticación con el objetivo de conseguir el identificador y la contraseña de un usuario), confianza (ofrece un mismo sistema de autenticación para diferentes servidores, pero no ofrece ninguna garantía de la identidad real del usuario) y principalmente la privacidad porque los proveedores de OpenID tendrán mucha información del usuario. En cuanto a la usabilidad, adopción, disponibilidad y patentes OpenID presenta inconvenientes también (36).

1.3.5. OpenID Connect

OpenID Connect es un protocolo de autenticación interoperable basado en la familia de especificaciones de OAuth 2.0. El flujo de información de este protocolo ocurre bajo HTTP utilizando mensajes JSON y está concebido para aprovechar los beneficios de técnicas modernas de los servicios web. OpenID Connect permite a los desarrolladores autenticar sus usuarios a través de sitios web y aplicaciones sin tener que poseer y gestionar archivos de contraseñas. Dicho protocolo aprueba a los clientes de todo tipo, incluyendo JavaScript basado en navegador y aplicaciones móviles nativas, para iniciar sesión en los flujos (37).

El protocolo OpenID Connect es la última propuesta para reactivar OpenID. Su propósito es redefinir y simplificar el protocolo construyéndolo sobre el protocolo OAuth; de ese modo se aprovecha el trabajo desarrollado para OAuth, dotándolo de una funcionalidad base de autenticación. Esa

propuesta ha permitido además, simplificar la implementación de OpenID quien ha recibido críticas por su excesiva complejidad (38).

OpenID Connect utiliza JWT, integrado con OAuth 2.0 (39), JWT es un compacto de medios de seguridad de un Identificador de Recursos Uniformes (URL por sus siglas en inglés). También utiliza JSON Web Signature (JWS por sus siglas en inglés), JSON Web Encryption (JWE por sus siglas en inglés), JSON Web Key (JWK por sus siglas en inglés), JSON Web API (JWA por sus siglas en inglés) y WebFinger (37).

OpenID Connect se basa en la idea original descentralizadora de OpenID, permitiendo que el usuario pueda elegir entre diferentes proveedores de identidad, aprovechando el alto grado de adopción que está consiguiendo OAuth y que permite implementar OpenID sobre él. La funcionalidad básica de OpenID Connect es la autenticación construida encima de OAuth 2.0 y el uso de las reivindicaciones de comunicar información sobre el usuario final (40).

La especificación de OpenID Connect consta de seis documentos; el núcleo, el descubrimiento, el registro dinámico, los tipos de respuestas múltiples de OAuth 2.0, el formulario de respuestas de OAuth 2.0 y la gestión de sesiones. El núcleo limita la funcionalidad básica de OpenID Connect que se basa en la autenticación construida encima de OAuth 2.0. El descubrimiento sintetiza la forma de que los clientes descubran dinámicamente información sobre los proveedores de OpenID. El registro dinámico trabaja de forma opcional solamente enfocado en los clientes y es semejante al descubrimiento, es decir, el documento permite que los clientes encuentren información del proveedor.

Los tipos de respuestas múltiples de OAuth 2.0 y el formulario de respuestas de OAuth 2.0 o el modelo posterior de respuesta presentan relación. Éste último define la forma de devolver los parámetros de respuestas de autenticación y el anterior los nuevos tipos específicos de respuesta OAuth 2.0. La gestión de sesiones es el documento que gestiona las sesiones de OpenID Connect, incluyendo la funcionalidad de cierre de sesión basado en mensajes posteriores.

En la Figura 1.3 se concretar las especificaciones y bases de OpenID Connect:

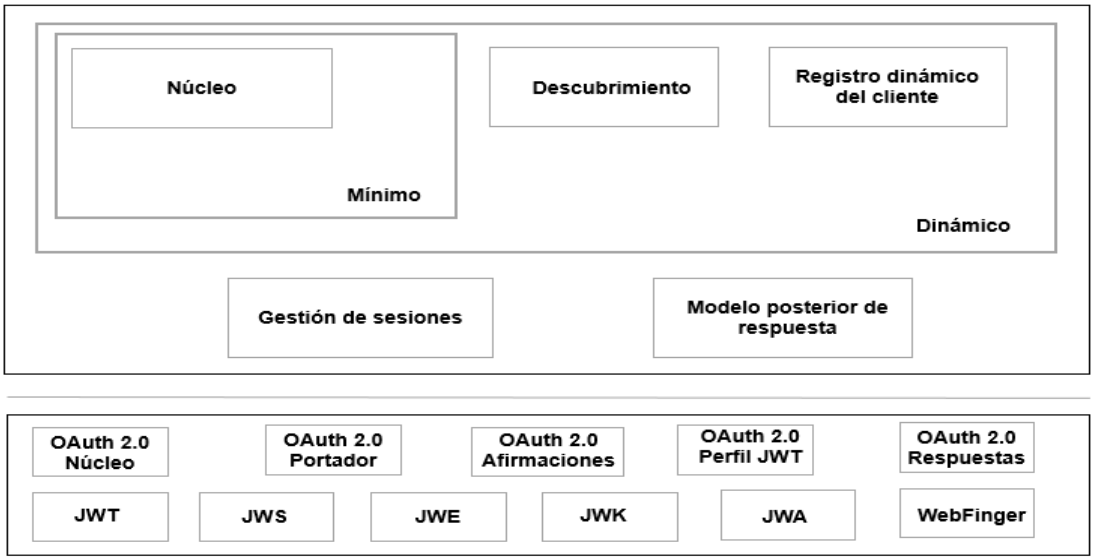


Figura 1.3 Especificación y bases de OpenID Connect.
(Fuente: OpenID Connect Specification Organization)

Para la implementación del componente se decide desarrollar un mecanismo híbrido integrando los protocolos OAuth 2.0 y OpenID Connect, un protocolo de autorización y autenticación respectivamente. OpenID Connect es el protocolo más avanzado con respecto a la autenticación, su integración con OAuth 2.0 posibilita unificar los escenarios de autenticación y autorización en un único protocolo; permitiendo desarrollar una solución que cuente con una única fuente de datos para el manejo de usuarios.

OpenID Connect junto a OAuth 2.0, con el empleo de tokens, elimina el registro de credenciales mediante usuario y contraseña, posibilitando con ello una mayor seguridad para el usuario. OpenID Connect por su parte está concebido para ser utilizado en ambientes de baja conectivad, lo cual es ideal para emplearlo en las instituciones cubanas. Al estar diseñado para aplicaciones, servicios web modernos y tecnologías móviles hace posible la evolución de la solución PACS-RIS evitando la obsolescencia de un componente con el empleo de estos protocolos.

1.4. Autenticación unificada

La autenticación unificada es la comprobación de las credenciales del usuario, esta emplea una sola pantalla de autenticación evitando múltiples cuentas y excesivas contraseñas (23). Diversas compañías punteras en el sector de la informática como Facebook y Google han implantado

mecanismos de autenticación unificada que pueden, a su vez, ser utilizadas por aplicaciones de terceros. (41, 42)

Facebook utiliza una variante del protocolo OAuth 2.0 para confirmar la autenticación y autorización. Esta variante permite controlar los datos que se comparten con otras aplicaciones, además presenta un conjunto de APIs para su interacción a través de la web o de diversas plataformas móviles. Al estar un usuario autenticado en Facebook, puede autenticarse en aplicaciones que interactúen con estas APIs sin necesidad de ingresar nuevamente sus credenciales. (41)

Por otra parte Google utiliza los protocolos OAuth 2.0 y OpenID Connect para realizar la autenticación y autorización de los usuarios de su plataforma (42). Presenta un conjunto de APIs y librerías para aplicaciones móviles y tecnologías de lado del servidor que permiten su integración con terceros. El empleo del componente Sign-in permite, una vez autenticado el usuario con las credenciales de Google, su ingreso a la aplicación utilizando la identidad del usuario en Google sin ingresar nuevamente su usuario y contraseña. Tanto para Google como para Facebook, es necesario registrar un nuevo cliente y que la aplicación se encuentre disponible en Internet. (41, 42)

Sin embargo, la utilización de los mecanismos de autenticación creados por Facebook y Google, traen consigo problemáticas. Primeramente, la solución de autenticación de Facebook no implementa el protocolo OpenID Connect, el cual los autores de la presente investigación consideran necesario para desarrollar el mecanismo en la solución PACS-RIS.

Otro inconveniente, sobre todo en Cuba, reside en que, para utilizar estos mecanismos, las aplicaciones que los implementen tienen que estar todo el tiempo disponibles en Internet. Importante destacar que la integración de estos componentes conlleva la posibilidad de compartir datos sensibles tanto con Google como con Facebook.

1.5. Formas de autenticación en los sistemas de salud

Cualquier transacción realizada por Internet en la que esté involucrada información relacionada con la salud en general, ha de realizarse garantizando los servicios de confidencialidad e integridad de la información, así como la autenticación de todas las partes que intervienen en la transacción. Las tecnologías de seguridad y firma electrónica son la base del desarrollo de los sistemas de salud en Internet (8).

Un sistema de seguridad completo para la salud, ya sea interno en las empresas o como proyecto de ámbito extendido en el que intervengan todos los agentes implicados, realiza inicialmente

consultorías y análisis de forma segura considerando aspectos globales de seguridad. Tiene el objeto de no hacer islas y que el alcance se realice para todos los sistemas de información. Para implantar un control de accesos a la información, el modelo de usuario/contraseña está prácticamente obsoleto y lo mejor es utilizar certificados digitales y sistemas que gestionen autorizaciones a los recursos basadas en perfiles (12).

Múltiples proyectos de entidades privadas y de la administración utilizan para sistemas de salud un completo sistema de firma electrónica para dar servicio a todas las aplicaciones internas y externas que la requieran. Cuentan con Sistemas de Redes Privadas Virtuales (VPN por sus siglas en inglés) y la Infraestructura de Llaves Públicas (PKI por sus siglas en inglés) en diferentes hospitales para cifrado de las comunicaciones. Además poseen diferentes sistemas PKI para automatización de procesos internos e informes, con el objetivo de disminuir el tráfico de papel y optimizar tiempos (12).

1.6. Sistemas de salud que presentan autenticación a nivel nacional e internacional

A continuación se estudian los tipos de autenticación existentes en los sistemas de salud en el ámbito nacional e internacional. El objetivo principal de esta caracterización es conocer los mecanismos de autenticación empleados actualmente.

1.6.1. Ámbito internacional

En el ámbito internacional diversos sistemas de salud utilizan la autenticación unificada. Algunos de ellos son el Sistema de Acceso a los Servicios Farmacéuticos (SIAPS por sus siglas en inglés) que se utiliza en Francia (43), el Sistema de Acceso a los Servicios de Investigaciones en la Salud (HINARI por sus siglas en inglés) establecido en Malasia (44) y el Sistema Farmacéutico de Madrid denominado Digitalfarmadrid de la ciudad de Madrid (45).Todos ellos implantados a partir de 1997, 1998 y 1996 respectivamente.

Cada sistema de salud es diferente y tiene distintas características. Se encuentra como sistema integrado de salud que cubra la mayor parte de las actividades sanitarias el Sistema Integrado de Gestión e Información para la Atención Sanitaria nombrado Diraya, del Servicio Andaluz de Salud (SAS por sus siglas en inglés) y Software Alliance System (VistA por sus siglas en inglés) de la asociación de veteranos de los Estados Unidos. Estos fueron instaurados en España desde 1998 y Estados Unidos desde 1995 respectivamente (46).

Como sistema integrador de información de las diferentes fuentes de datos del sistema sanitario ya sean aplicaciones de atención primaria, especializada, laboratorios o radiología se tiene el sistema Sundhed.dk de Dinamarca y The Spine del sistema sanitario inglés de Dinamarca. Dicho sistema es uno de los más actuales debido a que se instauró a inicios del año 2010 (46).

Varios sistemas médicos tienen el objetivo de integrar toda la información en un solo sistema digital. Esto ocurre en el Expediente Digital Único en Salud denominado EDUS, que a través de la web integra y automatiza toda la información del expediente de salud del paciente en Costa Rica (47). INDRA es un sistema asistencial de sanidad que al igual que EDUS unifica la autenticación, el mismo ha sido implantado en varios países de Latinoamérica principalmente Chile, Argentina y Perú (48). EDUS ha evolucionado notablemente en cuanto a su despliegue, pues fue creado desde 1997 siendo utilizado solamente en Chile (47, 48).

El Sistema Único de Salud (SUS por sus siglas en inglés) de Brasil es uno de los mayores sistemas públicos de salud del mundo, siendo el único que garantiza asistencia integral y totalmente gratuita para la totalidad de la población, inclusive a los pacientes portadores del VIH, sintomáticos o no, a los pacientes renales crónicos y a los pacientes con cáncer. Dicho sistema cuenta con una autenticación unificada para su acceso y se ha venido desarrollando desde hace 10 años atrás (49).

El sistema PACS sin embargo utiliza protocolos propietarios para su autenticación, pero no trabaja con una autenticación unificada. La autenticación que emplea es simple y puede usarse de forma limitada en el sitio local donde se encuentre tanto el usuario como el sistema (50). La integración del RIS y el PACS mantiene la autenticación del PACS accediendo de forma independiente a los informes radiológicos. La solución PACS-RIS lleva acabo el empleo de llaves públicas y privadas para su acceso (51). En Venezuela, Valencia y Laos se encuentra actualmente implantada dicha solución.

Todas las formas de autenticación mencionadas anteriormente han empleado software propietario para su creación, siendo además soluciones a la medida para aplicaciones específicas. Utilizan servidores para admitir la autenticación de contraseña provocando debilidades de seguridad, aplicaciones que no son de código abierto y tecnología propietaria. Debido a estas razones el cliente tiene que pagar licencia (al propietario del software o a la compañía propietaria) para instalar y utilizar un sistema informático.

1.6.2. Ámbito nacional

Los sistemas informáticos investigados, en el ámbito nacional, no presentan autenticación unificada, principalmente por la insuficiente propagación de red que tiene Cuba. La posibilidad de implantar un sistema informático de salud para sus ciudadanos con acceso a citas, historias clínicas o consultas (información necesaria en el proceso de atención al paciente) es difícil.

A continuación se mencionan sistemas informáticos investigados que tienen mecanismos de autenticación lo más cercano posible a la autenticación unificada. El Sistema de Información Hospitalaria (HIS por sus siglas en inglés) del CESIM y el Sistema Informático de Genética Médica nombrado MEDIGEN.

El HIS obtiene los datos más precisos en el proceso de acreditación y les facilita el acceso a los proveedores de servicios de salud, teniendo como objetivo mejorar los procesos de gestión clínica asociados a la explotación de la información. El mismo gestiona información desde diferentes instituciones y a diferentes módulos de la aplicación con un solo usuario. Sin embargo la forma de autenticación que presenta no es unificada si no está construida con sus técnicas y características impuestas (52).

Desde el año 2011 en Cuba se estableció MEDIGEN, también sin autenticación unificada (53).La orientación a servicios que presenta MEDIGEN contribuye a que la información que se gestione en los diferentes estudios de genética médica sea única y confiable. Esta orientación a servicios agiliza el desarrollo de muchas investigaciones científicas en el campo de la genética médica y garantiza la integridad de la información (53).

Actualmente se trabaja en varios sitios web relacionados con la medicina y la salud en Cuba con el objetivo de representar de forma informatizada aspectos esenciales, instructivos y de carácter científico tanto para los médicos como para la población, tal es el caso de Infomed. Este sitio es de libre acceso pues no presenta ningún sistema de autenticación (54).

1.7. Metodología, lenguajes y tecnologías

A continuación se caracteriza la metodología, tecnologías y lenguajes que se utilizarán en el desarrollo del componente. La presente investigación se regirá por las mismas, su selección está basada por las pautas del proyecto PACS-RIS para el desarrollo de su solución.

1.7.1. Proceso Racional Unificado

El Proceso Racional Unificado (RUP por sus siglas en inglés), es un modelo que permite el desarrollo de software a gran escala, mediante un proceso continuo de pruebas y retroalimentación, garantizando el cumplimiento de ciertos estándares de calidad. Presenta el inconveniente de generar mayor complejidad en los controles de administración del mismo, sin embargo, los beneficios obtenidos recompensan el esfuerzo invertido en este aspecto (55).

El proceso de desarrollo constituye un marco metodológico que define en términos de metas estratégicas, objetivos, actividades y artefactos (documentación) requerido en cada fase de desarrollo. Esto permite enfocar esfuerzo de los recursos humanos en términos de habilidades, competencias y capacidades a asumir roles específicos con responsabilidades bien definidas (55).

Dentro de las características principales de RUP se encuentran: dirigido por casos de uso, iterativo e incremental y centrado en la arquitectura. Los casos de uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Los casos de uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor (55), permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Como metodología se decidió emplear RUP, esta permite que el equilibrio entre casos de uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo (55). Que sea centrado en la arquitectura permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo. RUP tiene como estrategia un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos.

1.7.2. Lenguaje Unificado de Modelado 2.1

Se utilizará el Lenguaje Unificado de Modelado (UML por sus siglas en inglés) en su versión 2.1 como lenguaje gráfico para especificar, visualizar y documentar los artefactos que son generados a través de todo el ciclo de desarrollo del software. Está construido sobre los principales conceptos de la programación orientada a objetos. Puede ser utilizado por cualquier metodología de análisis y diseño orientada a objeto para expresar los diseños. Se utiliza para construir modelos y no para

guiar al desarrollador en la forma de realizar el análisis y diseño orientados a objetos ni le indica cuál proceso de desarrollo adoptar (56).

1.7.3. .Net Framework 4.5

El .NET Framework es un componente integral de Windows que admite la compilación y la ejecución de aplicaciones y servicios web XML. Proporciona un entorno coherente de programación orientada a objetos, de ejecución de código que reduce lo máximo posible la implementación de software y los conflictos de versiones. Basa toda la comunicación en estándares del sector para asegurar que el código de la plataforma se pueda integrar con otros tipos de código (57).

Esta plataforma puede hospedarse en componentes no gestionados que cargan los Entorno Común de Ejecución para Lenguajes (CLR por sus siglas en inglés) en sus procesos e inician la ejecución de código administrado. En .NET Framework no sólo se ofrecen varios hosts de motor en tiempo de ejecución, sino que también se admite el desarrollo de estos hosts por parte de terceros. Por ejemplo, ASP.NET hospeda el motor en tiempo de ejecución para proporcionar un entorno de servidor escalable para el código administrado.

Dicha plataforma habilita aplicaciones de ASP.NET y servicios web XML (57). Por las características anteriormente mencionadas se utilizará el .Net Framework.

1.7.4. OWIN

Open Web Interface for .NET (OWIN por sus siglas en inglés) define una interfaz estándar entre los servidores web .NET y aplicaciones web. El objetivo de la interfaz OWIN es fomentar el desarrollo de módulos simples para el desarrollo .NET web, y, al ser un estándar abierto, estimula el entorno de código abierto de .NET (59).

OWIN define tres tipos de componentes básicos, host (configura e inicializa el resto de componentes), servidor (recibe las peticiones y genera las respuestas) y middleware (es el resto de componentes OWIN, son componentes modulares que se encadenan uno tras otro). OWIN incluye componentes de middleware para la autenticación, incluyendo soporte para inicios de sesión que utilizan proveedores externos de identidad (como cuentas de Microsoft, Facebook, Google, Twitter), este se puede utilizar para implementar servidores OAuth 2.0 (60).

1.7.5. ASP.NET

ASP.NET es un modelo de desarrollo web unificado que incluye los servicios necesarios para crear aplicaciones web empresariales con el código mínimo (61). ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el CLR.

Los aspectos transversales de ASP.NET (autenticación, cache, sesión, roles) son los mismos en ASP.NET MVC. Básicamente es una implementación del patrón Modelo Vista Controlador (MVC por sus siglas en inglés). ASP.NET MVC es un framework de Microsoft para desarrollar aplicaciones web, usando tecnología .NET. Dicho framework fue creciendo con el paso de los años surgiendo las versiones 2, 3, 4 y recientemente la 5. Se utilizará esta última versión, pues actualiza la gestión de la identificación y la autentificación para que hagan uso de ASP.NET (62).

ASP.NET Web API es un marco que facilita la creación de servicios HTTP disponibles para una amplia variedad de clientes, entre los que se incluyen exploradores y dispositivos móviles. Es muy similar a ASP.NET MVC ya que contiene las características MVC como el enrutamiento, los controladores, los resultados de la acción y modelos; pero no es una parte del marco MVC. Se trata de una parte de la plataforma ASP.NET núcleo y se puede utilizar con MVC y otros tipos de aplicaciones web. También se puede utilizar como una aplicación de servicios web independiente (63).

OWIN y ASP.NET han dividido al máximo las diferentes capas para poder elegir en cada momento el componente que más interese en cada una de estas. ASP.NET Identity es un framework totalmente compatible con OWIN y permite ser empleada en cualquier aplicación alojada OWIN. (64)

1.7.6. Identity Server 3

Identity Server 3 es un framework que permite implementar el control de inicio de sesión único y el acceso para las aplicaciones web modernas y las API que utilizan protocolos como OpenID Connect y OAuth2. Es compatible con una amplia gama de clientes como móviles, aplicaciones web y de escritorio para admitir la integración de las nuevas y existentes arquitecturas.

En diversas literaturas utilizan diferentes términos para explicar su funcionamiento. De manera general se basa en servicio de token de seguridad, proveedor de identidad y servidor de

autorización, o sea, Identity Server 3 es una librería de software que emite tokens de seguridad a los clientes. Identity Server 3 incluye: autenticar a los usuarios que utilizan un almacén de cuentas local a través de un proveedor de identidad externa, proporcionar la gestión de sesiones y de sesión única, gestionar y autenticar los clientes, emitir tokens de identidad y acceso a clientes y validar los tokens (65).

1.7.7. PostgreSQL 9.2.4

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, su código fuente se encuentra disponible libremente. El desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y apoyada por organizaciones comerciales (66).

Esta versión agrega nuevas características que mejoran la flexibilidad, escalabilidad y rendimiento para diferentes tipos de usuarios de bases de datos. La replicación sincrónica que permite alta disponibilidad con consistencia sobre múltiples servidores y la regionalización por columna que soporta correctamente el ordenamiento por lenguaje en las base de datos, tablas o columnas son ventajas que permite el uso de PostgreSQL (66).

Este gestor incluye mejoras al soporte para JSON y rendimiento de los índices (66). Por las características presentadas anteriormente se utilizará PostgreSQL 9.2.4 como sistema gestor de base de datos.

1.7.8. C# 5.0

C# es un lenguaje orientado a objetos, permite la compilación de diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework. C# es utilizado para crear aplicaciones cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor y aplicaciones de base de datos principalmente. Proporciona un editor de código avanzado, cómodos diseñadores de interfaz de usuario, depurador integrado y numerosas herramientas que facilitan el desarrollo de aplicaciones basadas el lenguaje C# y .NET Framework (67).

Como lenguaje de programación se utilizará C# en su última versión estable 5.0. Entre sus características se puede mencionar la portabilidad del código, no admite funciones ni variables

globales sino que todo el código y los datos han de especificarse dentro de definiciones de tipos de datos, consta de un editor de código completo, plantillas de proyecto, asistentes para código, así como un depurador muy eficiente y fácil de usar.

1.8. Herramientas a utilizar para la solución propuesta

A continuación se caracterizan las herramientas a utilizar para el desarrollo del componente. Dichas herramientas son las definidas por el proyecto PACS-RIS para el desarrollo de sus soluciones.

1.8.1. Microsoft Visual Studio 2013

Como Entorno de Desarrollo Integrado (IDE por sus siglas en inglés) se ha seleccionado Microsoft Visual Studio. Es un conjunto de herramientas de desarrollo para construir aplicaciones web ASP.NET, servicios web XML y aplicaciones de escritorio (68).

Incluye herramientas que simplifican todo el proceso de desarrollo de aplicaciones, de principio a fin. Permite realizar una administración del ciclo de vida de las aplicaciones e incorpora otras de pruebas que ayudan a garantizar la calidad del código en todo momento. Brinda funcionalidades para la implementación y administración de cambios que garantizan que la base de datos y la aplicación estén siempre sincronizadas.

Es compatible con plataformas como Windows, Windows Server, Web, Cloud, Office y SharePoint, entre otras, todo en un único IDE (68). Existen distintos métodos para familiarizarse con Visual Studio, tanto si se han utilizado versiones anteriores, como si es la primera vez que se trabaja con el producto. En cada versión, ciertas herramientas y tecnologías se sustituyen por métodos mejores para conseguir los objetivos de programación. La consecuencia es que las aplicaciones creadas con versiones anteriores de Visual Studio pueden requerir una actualización para cargarse y generarse correctamente con la versión actual de Visual Studio.

1.8.2. Enterprise Architect 7.5

Como herramienta de diseño guiado por computadora se utilizará Enterprise Architect en su versión 7.5. Esta aplicación está basada en estándares abiertos como Lenguaje de Modelado Unificado y Notación de Procesos del Modelo de Negocio. Enterprise Architect soporta los frameworks de arquitectura, gestiona los requisitos, modelos estratégicos, modelos de análisis y procesos (69).

En dominios como la salud, servicios públicos, la astronomía o la defensa, negocios globales, bancario y muchos otros, Enterprise Architect captura y controla la complejidad y desempeña un papel fundamental de confianza en la entrega de alta calidad de soluciones (69).

1.8.3. TortoiseSVN 1.8.2

Como cliente de subversión se utilizará TortoiseSVN en su versión 1.8.2. Es un proyecto de código abierto. Fue diseñado para la revisión y control de versiones de código fuente para Windows (70).

TortoiseSVN controla ficheros y directorios que se almacenan en un repositorio central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho. Esto permite que se pueda recuperar versiones antiguas de los ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién realizó el cambio (70).

Una vez realizado un análisis de los diferentes protocolos de autenticación se decidió integrar OAuth 2.0 y OpenID Connect porque se obtendrá un mejor enfoque para asegurar las aplicaciones modernas para el futuro previsible. La forma de autenticación empleada en Google y Facebook no son las adecuadas pues la información médica de los pacientes no debe ser compartida en Internet, porque no se mantendría la confidencialidad de la misma.

Después de efectuar un estudio y caracterizar varios sistemas de salud con sus respectivas formas de autenticación tanto a nivel nacional como internacional, se determinó que no constituían una solución viable para resolver el problema planteado. Se consideran, luego del estudio de las herramientas, tecnologías y lenguajes, que son las adecuadas para el desarrollo del componente en cuestión, siendo empleadas para el modelado, diseño e implementación de la aplicación. Además son utilizadas en el desarrollo de la solución PACS-RIS.

CAPÍTULO 2. CARACTERÍSTICAS DEL COMPONENTE PARA LA AUTENTICACIÓN UNIFICADA DE LA SOLUCIÓN PACS-RIS

En el presente capítulo se tiene como objetivo explicar cómo se lleva a cabo el proceso de autenticación de la solución PACS-RIS. Se propone una solución al problema planteado para la investigación. Se realiza el modelo de dominio correspondiente al flujo descrito, para obtener posteriormente las características con las que contará el componente a desarrollar. Se detallan los requisitos funcionales y no funcionales, así como el diagrama de casos de uso y la descripción de cada uno de ellos.

2.1. Sistema propuesto

A partir del análisis realizado sobre los principales protocolos de autenticación, se propone desarrollar un componente para la autenticación unificada de la solución PACS-RIS utilizando la integración de OAuth 2.0 con OpenID Connect. El componente constará con el servidor de autorización Celsus Cloud, capaz de generar, responder y procesar peticiones sobre un servicio web confidencial y seguro.

Con el objetivo de garantizar un manejo rápido, seguro y personalizable de la información dentro del servidor de autorización, el componente cuenta con la aplicación Celsus Cloud Web Admin desarrollada sobre el framework MVC de la plataforma ASP.NET. Permite registrar usuarios, roles, clientes (aplicaciones que actúan como usuarios), claims y scopes sin la necesidad de acceder o interrumpir el servicio.

Al registrar un nuevo cliente se genera un ticket cifrado con las credenciales que necesita para hacer una petición de acceso a recursos. El ticket será gestionado por la librería Celsus Client Connector. Esta cuenta con un certificado digital único, donde cifra o no la información que utiliza el cliente para conectarse con el servidor de autorización mediante OAuth 2.0 y OpenID Connect.

El sistema propuesto permite eliminar, cuando se desee, los accesos de forma independiente en el componente. Impide afectaciones en el flujo de trabajo de la institución donde se encuentre implantado el sistema y posibilita la evolución tecnológica con respecto a la nube porque existiría una sola forma de autenticación, lo que permite conocer el lugar y la manera de autenticación del usuario o del cliente.

Al autenticarse un usuario, en lugar de utilizar credenciales del propietario de los recursos para acceder a los recursos protegidos, el cliente obtiene un token de acceso, utilizándolo para acceder a los recursos protegidos alojados en el servidor de recursos. Es decir, el usuario puede acceder a recursos protegidos sin la necesidad de enviar su nombre de usuario y la contraseña para solicitar los mismos, además puede revocar privilegios a las aplicaciones cuando desee.

La información que envían al servidor de autorización los clientes y los usuarios, se realiza mediante cadenas de texto cifradas, incrementando los niveles de seguridad en el envío y procesamiento de datos. La información enviada representa un eslabón fundamental en el flujo de negocio para las aplicaciones de la solución PACS-RIS. Estos flujos responden a una adecuada aplicación del mecanismo híbrido, conformado por los protocolos anteriormente mencionados.

El sistema posee la capacidad de gestionar no solo la autenticación de los usuarios, sino la autorización de las aplicaciones que no requieran utilizar credenciales como usuario y contraseña para acceder a recursos protegidos. Esto último se realiza mediante un control del flujo que cada una de las aplicaciones de la solución PACS-RIS tiene incorporado. Basta con identificar el tipo de flujo que maneja la aplicación que se desea registrar en el sistema y especificar los datos que requiera.

Los flujos identificados por el sistema propuesto son:

- Código de autorización, devuelve el código de autorización requiriendo el token de acceso y el token de acceso secreto del usuario para recuperar los tokens empleados.
- Implícito, permite solicitar fichas sin la autenticación del cliente utilizando una redirección del identificador de recursos para verificar la identidad del cliente.
- Híbrido (combinación de los dos flujos anteriores).
- Propietario de recurso, interactúan constantemente con el implícito.
- Credenciales de cliente, interactúan constantemente con el código de autorización.
- Personalizado, interactúan constantemente con el código de autorización.

Es necesario utilizar la librería IdentityServer 3 para la creación de dicho componente, pues incluye autenticar a los usuarios que utilizan un almacén de cuentas locales a través de un proveedor de identidad externa. Proporciona la gestión de sesiones y de sesión única, además de gestionar y autenticar los clientes. Permite emitir tokens de identidad y acceso a clientes y validar los tokens.

En la Figura 2.1 se muestra claramente en qué consiste el componente a desarrollar:

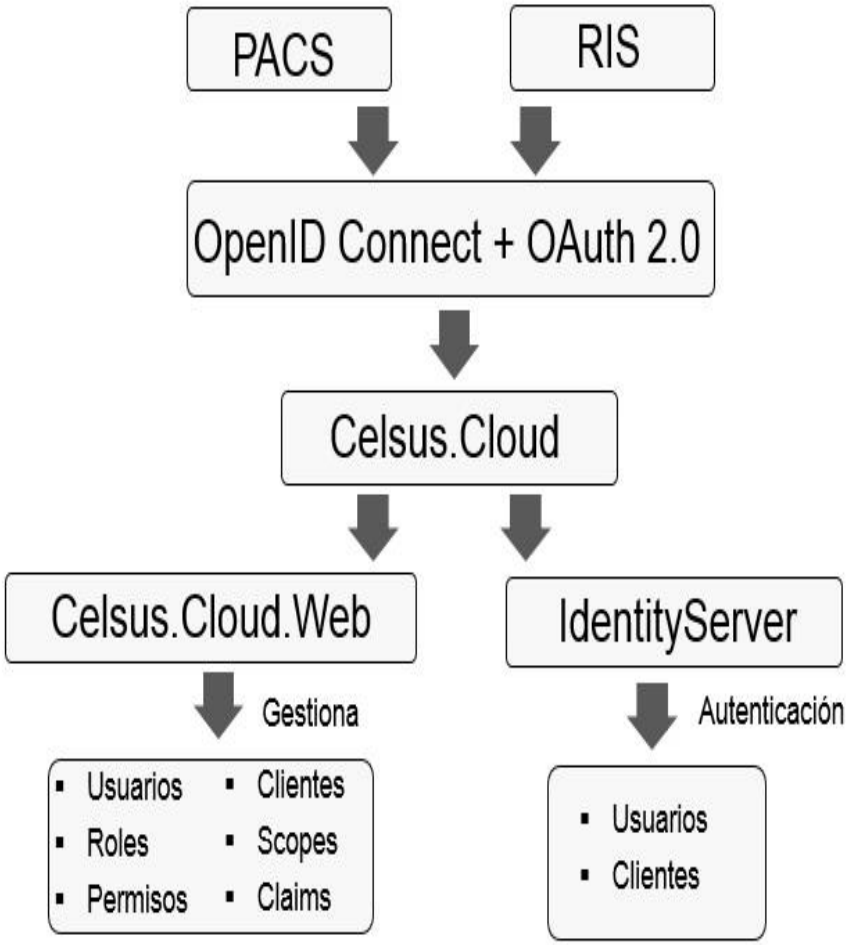


Figura 2.1 Componente para la autenticación unificada de la solución PACS-RIS.

(Fuente: elaboración propia)

2.2. Modelo de Dominio

Un modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés (71). Se puede usar para documentar relaciones entre ellos y responsabilidades de clases conceptuales.

Con el objetivo de lograr una mejor comprensión del modelo de dominio a continuación se describen los principales conceptos asociados al problema en cuestión.

Tabla 1 Entidades y Conceptos del Modelo del Dominio.

Entidades y Conceptos	Descripción
Administrador	Es el usuario encargado de gestionar la configuración de las aplicaciones que conforman la solución PACS-RIS.
Especialista	Es el usuario (especialista en radiología) que trabaja directamente con el paciente.
Transcriptor	Es el usuario (transcriptor) que trabaja con los informes médicos radiológicos.
Usuario	Usuario que accede a la solución PACS-RIS, ya sea el especialista o el administrador.
PACS	Contiene componentes o aplicaciones con diversos tipos de autenticación. Es el sistema que se encarga del almacenamiento, la visualización y la transmisión de las imágenes médicas que se obtienen en los equipos de adquisición producto de los estudios realizados a los pacientes.
RIS	Sistema encargado de la gestión de la información que se maneja en los departamentos de imagenología.
Servidor_Imágenes	Es el servidor que almacena las imágenes médicas. Representa un componente del PACS con autenticación.

Visor_Web	Es la herramienta que permite la visualización de las imágenes en cualquier computadora del hospital que disponga de un navegador. Representa un componente del PACS sin autenticación.
Servidor_Listas_Trabajo	Sistema con un servicio básico que permite a los equipos consultar su lista de trabajo. Representa un componente del PACS sin autenticación.
Estación_Diagnóstico_General	Representa la computadora en la cual se tiene el visor para realizar la función de diagnóstico general.
Herramienta_Edición_Informes_Imagenológicos	Representa un componente del PACS, tiene un mecanismo de autenticación que comparte con el RIS, el mismo es conocido también como reportador.
Mecanismo_de_autenticación	Servicio basado en SOAP que se le pasa usuario y contraseña.

(Fuente: elaboración propia)

Dicho modelo de dominio permitirá mostrar de manera visual los objetos del dominio (clases conceptuales) y las asociaciones entre las clases conceptuales.

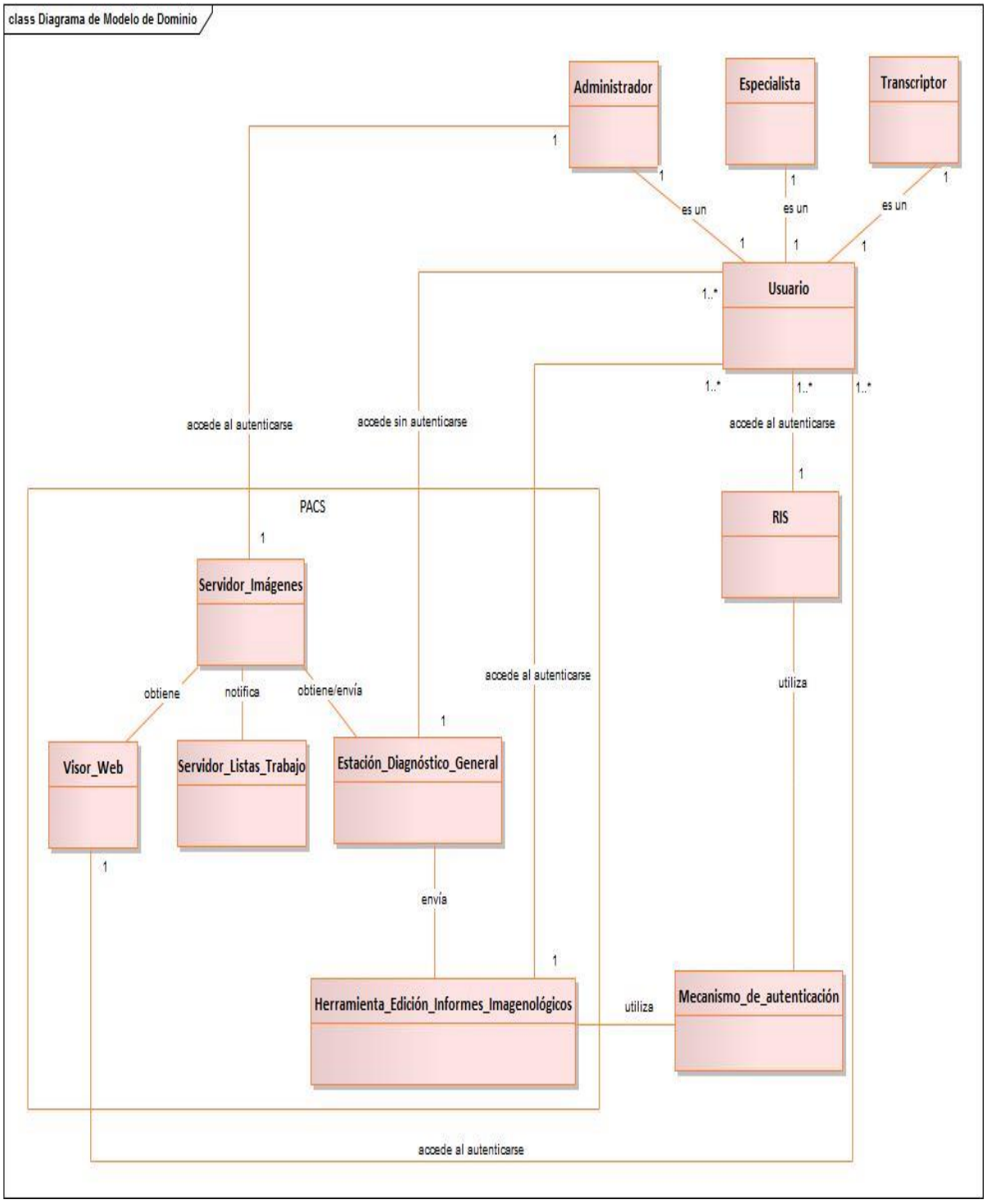


Figura 2.2 Modelo de dominio del componente para la autenticación unificada de la solución PACS-RIS. (Fuente: elaboración propia)

2.3. Especificación de los requerimientos del software

Los requerimientos o requisitos de un sistema son la descripción de los servicios proporcionados por este y sus restricciones. Dichos requerimientos reflejan las necesidades de un cliente que hará uso del sistema en cuestión. (72)

2.3.1. Requerimientos funcionales

Los requerimientos funcionales son servicios que el sistema debe proporcionar. Determinan cómo el sistema debería reaccionar a una determinada entrada y cómo debe comportarse en situaciones particulares. Estos requerimientos dependen del tipo de software que se desarrolle, de los posibles usuarios y del enfoque general tomado por la organización al redactarlos. En algunos casos, los requerimientos funcionales de los sistemas también pueden declarar explícitamente lo que el sistema no debe hacer (72).

La Tabla 1 muestra los requisitos funcionales del componente para la autenticación unificada de la solución PACS-RIS.

Tabla 2 Requisitos funcionales del componente para la autenticación unificada de la solución PACS-RIS.

Requerimiento	Descripción
RF 1.1 Autenticar usuario.	Permite al usuario autenticarse en el sistema introduciendo su usuario y contraseña.
RF 2.1 Registrar App cliente.	Permite registrar un App cliente en el sistema.
RF 2.2 Modificar App cliente.	Esta funcionalidad permite modificar los datos de un App cliente ya existente.
RF 2.3 Eliminar App cliente.	Permite eliminar un App cliente del sistema.
RF 2.4 Buscar App cliente.	Permite buscar App cliente por determinados criterios de búsqueda.
RF 3.1 Registrar scope.	Permite registrar un scope en el sistema.

RF 3.2 Modificar scope.	Esta funcionalidad permite modificar los datos de un scope ya existente.
RF 3.3 Eliminar scope.	Permite eliminar un scope del sistema.
RF 3.4 Buscar scope.	Permite buscar scope por determinados criterios de búsqueda.
RF 4.1 Registrar rol.	Permite registrar un nuevo rol en el sistema.
RF 4.2 Asignar rol.	Permite asignar un rol determinado al usuario.
RF 4.3 Modificar rol.	Permite modificar los datos de un rol en el sistema.
RF 4.4 Eliminar rol.	Permite eliminar un rol del sistema.
RF 4.5 Buscar rol.	Permite buscar rol por determinados criterios de búsqueda.
RF 5.1 Registrar usuario.	Esta funcionalidad permite registrar los datos del usuario.
RF 5.2 Cambiar contraseña.	Esta funcionalidad permite modificar la contraseña de un usuario existente.
RF 5.3 Modificar usuario.	Esta funcionalidad permite modificar los datos del usuario.
RF 5.4 Eliminar usuario.	Permite descartar los datos de un usuario que estaba registrado en el sistema.
RF 5.5 Activar usuario.	Esta funcionalidad permite que el usuario obtenga nuevamente la posibilidad de realizar cualquier acción en el sistema.

RF 5.6 Desactivar usuario.	Esta funcionalidad permite que el usuario quede desactivado, perdiendo la posibilidad de realizar cualquier acción en el sistema y sus datos siguen estando en el sistema.
RF 5.7 Bloquear usuario.	Permite bloquear al usuario, impidiendo que realice cualquier acción en el sistema cuando tiene más de 3 intentos de autenticación incorrectos o cuando no cumple con las políticas de seguridad del sistema.
RF 5.8 Desbloquear usuario.	Permite que un usuario anteriormente bloqueado pueda volver a realizar aquellas acciones a las que está autorizado.
RF 5.9 Buscar usuario.	Permite buscar usuarios por determinados criterios de búsqueda.
RF 6.1 Registrar claim.	Esta funcionalidad permite asignarle claims a un rol.
RF 6.2 Asignar claim.	Esta funcionalidad permite añadir un nuevo claim.
RF 6.3 Modificar claim.	Permite modificar los claims existentes o renovarlos por otros nuevos.

(Fuente: elaboración propia)

En la Figura 2.3 se muestra el diagrama de requisitos funcionales del componente para la autenticación unificada de la solución PACS-RIS.

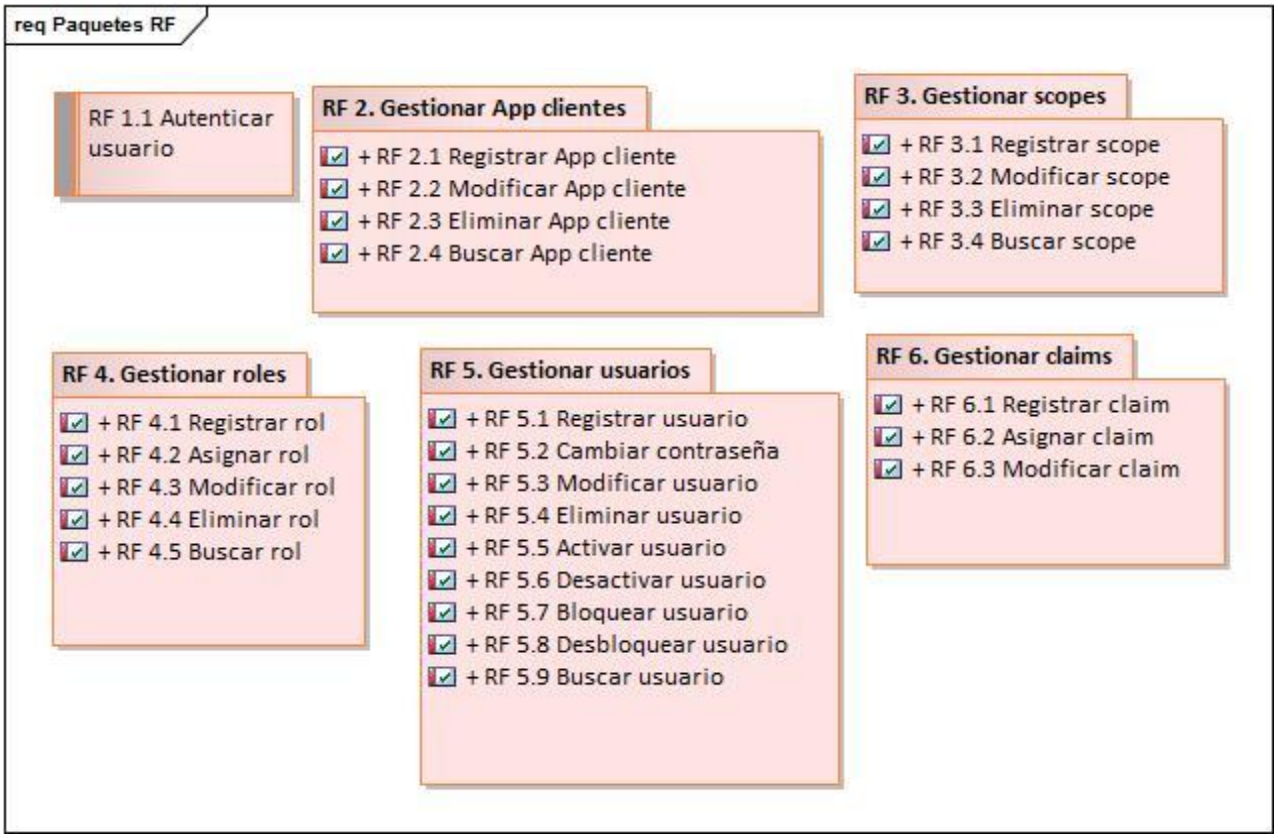


Figura 2.3 Diagrama de requisitos funcionales del componente para la autenticación unificada de la solución PACS-RIS.

(Fuente: elaboración propia)

2.3.2. Requerimientos no funcionales

Los requerimientos no funcionales definen propiedades y restricciones del sistema. Como su nombre sugiere, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. Dichos requerimientos son de gran importancia para que el producto tenga éxito; los mismos pueden ser más críticos que los requisitos funcionales (72).

Los requerimientos no funcionales fueron nombrados asignándoles un prefijo, de acuerdo a las Pautas para la Fase de Requerimientos del proyecto PACS-RIS.

Tabla 3 Requisitos no funcionales del componente para la autenticación unificada de la solución PACS-RIS.

Requerimiento	Descripción
RNU 1. Resaltar visualmente los datos incorrectos introducidos por el usuario.	El sistema debe resaltar el campo que contiene datos incorrectos en rojo y mostrar un mensaje de error.
RNDI 1. Uso de Framework .Net 4.5.	Se especifica el uso de Microsoft Framework .Net 4.5 que ofrece mejoras en cuanto a administración y rendimiento. El lenguaje de programación C# depende de este Framework.
RNDI 2. C# como lenguaje de programación.	Se deberá utilizar C# 5.0 como lenguaje de programación pues está diseñado y optimizado para la plataforma .NET.
RNDI 3. Entorno Integrado de Desarrollo Microsoft Visual Studio 2013.	Se utilizará Microsoft Visual Studio 2013 como entorno integrado de desarrollo.
RNDI 4. TortoiseSVN como herramienta para el control de versiones.	Se utilizará la herramienta TortoiseSVN 1.8.2 para el control de versiones, la cual puede integrarse al entorno de desarrollo seleccionado.
RNDI 5. Librerías a utilizar.	Se utiliza la librería NHibernate para la conexión con PostgreSQL y la librería Identity Server 3 para implementar el protocolo OAuth 2.0.
RNDI 6. Estándar de codificación del proyecto PACS-RIS.	El sistema debe regirse por las pautas del proyecto PACS-RIS.
RNDI 7. Uso de Enterprise Architect como herramienta CASE.	Se utilizará como herramienta CASE Enterprise Architect en su versión 7.5.

RNDI 8. Uso de PostgreSQL como herramienta.	Se utilizará como herramienta para el trabajo con las base de datos PostgreSQL en su versión 9.2.4.
RNDI 9. Patrones a utilizar para el diseño del sistema.	Los patrones a utilizar para el diseño del sistema son los siguientes: MVC, Singleton.
RNL 1. El componente y la documentación generada por el mismo, son propiedad del proyecto PACS-RIS.	El componente y la documentación generada por el mismo, son propiedad del proyecto PACS-RIS.
RNF 1. Disponibilidad del sistema siempre.	El sistema debe estar disponible siempre.
RNS 1. Confiabilidad de la información.	El sistema debe garantizar que la información sea editada únicamente por quien está autorizado y posea permisos para ello.
RNS 2. Garantizar el acceso para realizar cualquier acción.	Los usuarios deben estar autenticados para realizar cualquier acción.
RNE 1. Responder cualquier petición de forma rápida.	El sistema debe dar respuesta para cualquier petición del usuario entre uno (1) y tres (3) segundos.
RNFO 1. Memoria RAM de 2 GB como mínimo.	Para el correcto funcionamiento del sistema, se necesita un CPU Dual CORE o superior y 2GB de memoria RAM como mínimo y se recomienda una tarjeta de red Gigabit Ethernet NIC.
RNFO 2. CPU Dual CORE o superior.	
RNFO 3. Gigabit Ethernet NIC.	
RNFO 4. Sistema operativo Windows 7 o superior.	El software debe instalarse sobre el sistema operativo Windows 7 o superior.

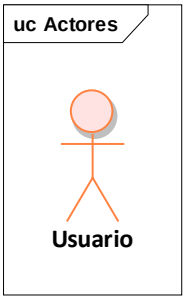
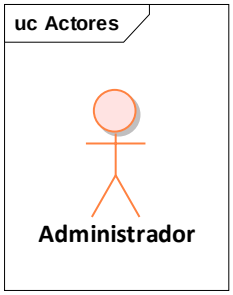
RNFO 5. Capacidad de disco duro 20 GB como mínimo.	La capacidad del disco duro debe ser como mínimo de 20 GB.
RNSO 1. La aplicación debe mostrarse correctamente en cualquier navegador.	La aplicación debe mostrarse correctamente en cualquier navegador; siendo como mínimo compatible con: Internet Explorer 9.0 o superior; Mozilla Firefox 26.x o superior, Google Chrome 24.x.
RNIU 1. Interfaz sugerente y amigable.	La interfaz de usuario debe ser sugerente, amigable, con funcionamiento intuitivo para el usuario. Debe también mostrar claridad y buena organización de la información, que permita la interpretación completa de la misma.
RNI 1. Disponibilidad de la conexión siempre.	Debe haber conexión siempre.

(Fuente: elaboración propia)

2.4. Definición de los actores del sistema

Luego de describir las clases conceptuales del modelo de dominio y los requerimientos funcionales y no funcionales del componente para la autenticación unificada de la solución PACS-RIS, se definen los actores que interactúan con los casos de uso del sistema, los mismos se muestra en la siguiente tabla:

Tabla 4 Actores del sistema.

Actor	Justificación
	Se autentica en el componente para la autenticación unificada de la solución PACS-RIS.
	Gestiona los roles, claims, usuarios, App clientes y scopes.

(Fuente: elaboración propia)

2.5. Diagrama de casos de uso del sistema

Los diagramas de casos de uso documentan el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

En la Figura 2.4 se muestra el diagrama de casos de uso del componente para la autenticación unificada de la solución PACS-RIS.

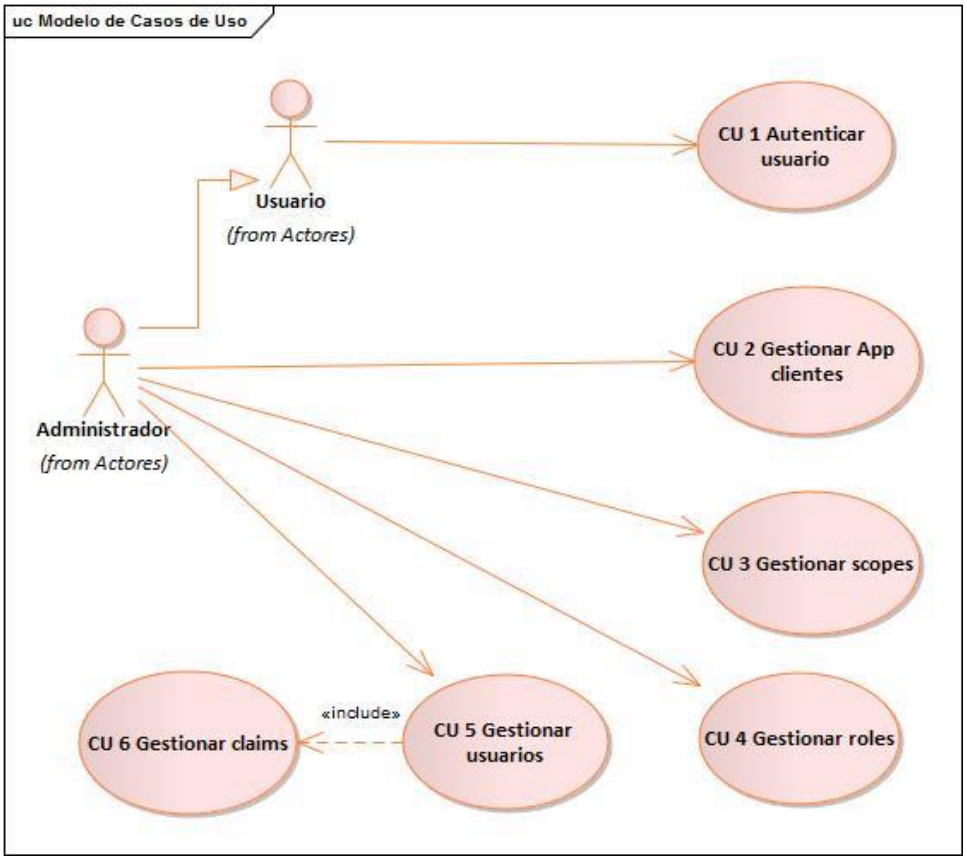


Figura 2.4 Diagrama de casos de uso del componente para la autenticación unificada de la solución PACS-RIS.

(Fuente: elaboración propia)

2.6. Descripción de los casos de uso del sistema

La siguiente tabla muestra el resumen de la descripción correspondiente al caso de uso Gestionar roles. El resto de las descripciones, por su dimensión, se encuentran en el Expediente de Proyecto Celsus.Cloud en el artefacto CESIM_PACS-RIS_010114a_Especificacion_de_casos_de_uso.

- Caso de uso – Gestionar roles.

Tabla 5 Descripción expandida del caso de uso Gestionar roles.

Objetivo	El objetivo de este CU es administrar los roles que interactúan en la aplicación.
Actores	Administrador (inicia)
Resumen	El caso de uso inicia cuando el administrador procede a registrar, modificar, eliminar o buscar un rol. Se muestran los datos correspondientes en dependencia de la acción a realizar. La aplicación permite ver una lista de roles y asignarle los mismos a los usuarios.
Complejidad	Media
Prioridad	Alta
Referencias	RF 4.1, RF 4.2, RF 4.3, RF 4.4, RF 4.5

(Fuente: elaboración propia)

Terminada la propuesta de solución para el desarrollo del componente para la autenticación unificada de la solución PACS-RIS, se realizó un modelo de dominio que permitió una mejor comprensión del problema y una rápida identificación de las funcionalidades a desarrollar. A partir del mismo se identificaron las características que debe cumplir el componente para la autenticación unificada y se describieron en forma de requisitos funcionales y no funcionales. Como resultado de la definición y descripción de los casos de uso del sistema, se obtuvieron las funcionalidades de dicho componente, estas proporcionaron el inicio a la fase de diseño.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL COMPONENTE PARA LA AUTENTICACIÓN UNIFICADA DE LA SOLUCIÓN PACS-RIS

En este capítulo se presenta el diseño del componente para la autenticación unificada de la solución PACS-RIS, mediante los diagramas de clases de diseño y de secuencia que participan en la realización de los casos de uso. Además, se realiza una breve descripción de las clases involucradas en la realización de los casos de uso, así como el modelo arquitectónico y los patrones utilizados en el componente.

3.1. Diseño

En la etapa de diseño se transforman los requerimientos en un modelo de diseño del sistema a implementar, se define la arquitectura robusta para el sistema y se adapta el mismo a un entorno de implementación. Esto permite que el sistema cumpla con parámetros como extensibilidad, reusabilidad, compatibilidad, portabilidad y robustez, lo que permite que todo el software alcance un alto nivel de calidad.

La línea base de la arquitectura sostendrá posteriormente los requerimientos del sistema. En esta etapa los requerimientos son transformados en un conjunto de clases que, relacionadas entre sí, lograrán llevarlos a cabo. Además, se generan artefactos que expresan gráficamente cómo se realiza ese proceso.

Los patrones de diseño son soluciones simples y refinadas a problemas específicos y comunes del diseño orientado a objetos (73). Estos expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas de software. Al entender el funcionamiento de estos patrones los diseños serán mucho más flexibles, modulares y reutilizables.

El diseño debe ser muy bien delineado para que el sistema pueda ser implementado sin imprecisiones, y su principal objetivo es la elaboración de los diagramas de clases de diseño, que muestra las clases participantes en la realización de un caso de uso con todos sus atributos (74). Los diagramas de las clases del diseño correspondientes a los casos de uso arquitectónicamente significativos, por su dimensión, se encuentran en el Expediente de Proyecto Celsus.Cloud en el artefacto CESIM_PACS-RIS_010215_Modelo_de_disenno.

Los diagramas de secuencia muestran gráficamente las interacciones del actor y de las operaciones a que dan origen. Con el objetivo de alcanzar un entendimiento de las actividades que se llevan a

cabo en los casos de uso arquitectónicamente significativos (75), Los diagramas de secuencia correspondientes a los casos de uso arquitectónicamente significativos, por su dimensión, se encuentran en el Expediente de Proyecto Celsus.Cloud en el artefacto CESIM_PACS-RIS_010215_Modelo_de_disenno.

3.2. Descripción de las clases

La clase es el componente más importante en el paradigma de la programación orientada a objetos. Esta se presenta a sí misma, como una descripción en materia de propiedades y acciones pertenecientes a un objeto de la vida real. El paradigma de programación mencionado, se basa en la identificación de esas clases que se evidencian en el dominio en que se enmarca el sistema, mediante sus operaciones y relaciones, logrando llevar a cabo la funcionalidad esperada del mismo (74).

La descripción de las clases permite obtener detalles de las mismas, principalmente las que están involucradas en la realización de los casos de uso arquitectónicamente significativos. Las descripciones de clases, por su dimensión, se encuentran en el Expediente de Proyecto Celsus.Cloud en el artefacto CESIM_PACS-RIS_010215_Modelo_de_disenno.

3.3. Modelo Arquitectónico

Las técnicas metodológicas desarrolladas con el fin de facilitar la programación, se engloban dentro de la llamada Arquitectura de Software o Arquitectura Lógica. La arquitectura es un nivel de diseño que consiste en un grupo de abstracciones y patrones que brindan un esquema de referencia útil para guiar el desarrollo de software dentro de un sistema informático.(76)

El patrón es un esquema de solución que se aplica a un tipo de problema, esta aplicación del patrón no es mecánica, sino que requiere de adaptación y matices (76). Por ello, plantea Alexander Christopher que los numerosos usos de un patrón no se repiten dos veces de la misma forma. Mediante dichos patrones se pueden tomar buenas decisiones y emanan conceptos que permitan la comunicación entre los equipos que participen en un proyecto.

La arquitectura orientada a servicios se basa en un servidor de autorización basado en la tecnología web API, dicho servidor utiliza los datos en forma de nube y es el encargado de proporcionar servicios a los clientes. Los clientes pueden ser; aplicación web, aplicación de teléfono, aplicación de escritorio y los servicios web.

La arquitectura N-capas por su parte se basa en la aplicación de administración. La programación por capas se refiere a un estilo de programación que tiene como objetivo separar la lógica de diseño de la lógica de negocios (62). Una de las ventajas que tiene este estilo es que el desarrollo del software se puede llevar a cabo en varios tipos de niveles, de esta forma un cambio en algún nivel no afectará los demás o las afectaciones de estos serán mínimas.

De dicha arquitectura se utiliza la capa de presentación, de acceso a datos, de dominio y de configuración. El estilo arquitectónico MVC se encuentra presente en la capa de presentación, ver Figura 3.3. MVC es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (62).

3.3.1. Patrón arquitectónico

Para el desarrollo del componente y teniendo en cuenta las herramientas, tecnologías y metodologías propuestas, se define como parte de la línea base de la arquitectura, la implementación del patrón arquitectónico MVC. Este permite la separación de los datos de una aplicación, la interfaz de usuario y la lógica de control, en tres componentes distintos: el modelo, donde se encuentran los datos; la vista, que muestra la información del modelo al usuario; y el controlador, que gestiona las entradas del usuario.

Se utiliza MVC en el desarrollo de la aplicación, con el objetivo de crear un componente que asegure la calidad atendiendo a diversos parámetros que son deseables para todo desarrollo, como la estructuración de la aplicación o reutilización del código, lo que debe influir positivamente en la facilidad de desarrollo y el mantenimiento.

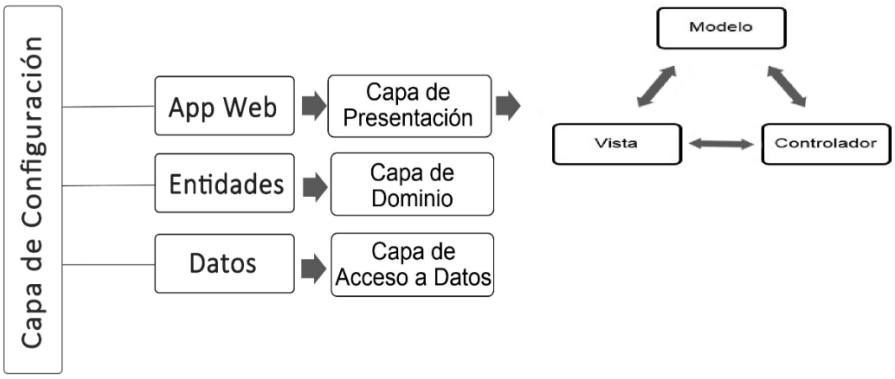


Figura 3.1 Estilo arquitectónico MVC y su relación con la arquitectura N-capas. (Fuente: elaboración propia)

El modelo representa las reglas de negocio de la aplicación (y el acceso a datos subyacente), las vistas, la presentación de la aplicación y el controlador actúa de intermediario entre el usuario, el modelo y las vistas. En el contexto de ASP.NET MVC toda la lógica de negocio y el acceso a datos es el modelo, las vistas contienen, básicamente, el código que se envía al navegador y los controladores reciben las peticiones del navegador y en base a esas, deciden las vistas y los datos que se deben enviar de vuelta al navegador (62).

3.3.2. Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. Están orientados al cambio y soportan la reutilización de arquitecturas de software (77).

- **Patrón GoF**

En la implementación del componente se usa Singleton, uno de los patrones GoF (Gang of Four) clasificado como patrón de creación, el cual tiene el objetivo de ocultar los detalles de cómo los objetos son creados o inicializados. Este patrón se evidencia en la clase DBFactory, dicha clase se utiliza para la conexión con la base de datos de la aplicación.

Todo el proceso que lleva a cabo MVC se integra con el patrón de diseño Singleton, pues garantiza que una clase sólo tenga una instancia y proporciona un punto de acceso global a ésta instancia. La instancia es una operación de clase, Singleton hace que la clase sea responsable de su única instancia (77), es decir, oculta el constructor de la clase Singleton, para que los clientes no puedan crear instancias.

En .NET, Singleton hace función de un objeto que se puede configurar como objeto remoto, de esta forma dicho patrón sirve a múltiples clientes y comparte datos, almacenando información de estado entre las distintas invocaciones. Es útil en escenarios donde los datos deben ser compartidos en forma explícita entre clientes (77).

- **Patrones GRASP**

Los patrones GRASP (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos (77). Uno de ellos es la alta cohesión, el cual indica que la información almacenada en las clases debe ser coherente y relacionada a lo que se maneja en dicha clase, el mismo se evidencia en la clase IdentityRole. Se

utiliza además el patrón bajo acoplamiento, reflejado en la poca relación existente entre las clases que conforman el componente, este se encuentra presente en la clase RegisterViewModel.

El patrón experto por su parte manifiesta que cada clase conozca su información y sea la encargada de implementar las funcionalidades que le corresponde como por ejemplo RollerController.

Con la realización de este capítulo se definió desarrollar el componente utilizando una arquitectura híbrida basada en la arquitectura N-capas y la arquitectura orientada a servicios. Se determinó emplear el patrón Singleton con el objetivo de ocultar los detalles de cómo los objetos son creados o inicializados. Se obtuvo una mejor abstracción para la realización del componente, además de facilitar la comprensión de su funcionamiento al confeccionar el diagrama de clases y de secuencia. Aspectos esenciales para comenzar la implementación.

CAPÍTULO 4. IMPLEMENTACIÓN DEL COMPONENTE PARA LA AUTENTICACIÓN UNIFICADA DE LA SOLUCIÓN PACS-RIS

En este capítulo se describen los temas asociados a la implementación del sistema. Se presenta el modelo de componentes y la descripción de los mismos. Se muestra el modelo de despliegue dando una visión de cómo quedará desarrollada y distribuida la aplicación, además de describir los nodos de dicho modelo. Se realiza una descripción de los estándares de codificación utilizados en la aplicación para una mejor comprensión y se refleja el tratamiento de errores.

4.1. Diagrama de componentes

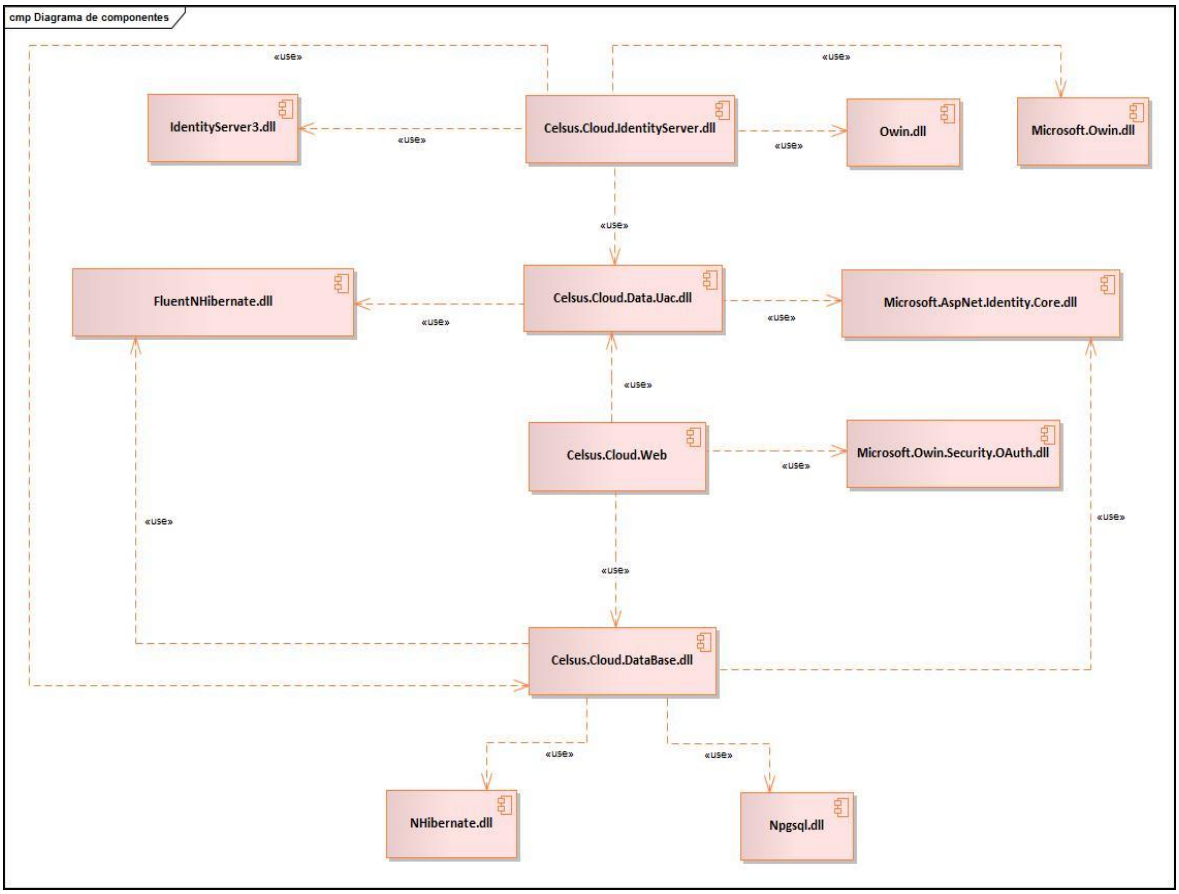


Figura 4.1 Diagrama de componentes del componente para la autenticación de la solución PACS-RIS. (Fuente: elaboración propia)

4.1.1. Descripción de los componentes

Tabla 6 Descripción del diagrama de componentes.

Componente	Descripción
Celsus.Cloud.IdentityServer.dll	Brinda un servicio web de servidor de identidad, capaz de autenticar y autorizar usuarios de otras aplicaciones (clientes).
Celsus.Cloud.DataBase.dll	Realiza la configuración necesaria para la creación, conexión y consulta con la base de datos.
Celsus.Cloud. Web	Aplicación web que gestiona las entidades persistentes en base de datos del sistema.
Celsus.Cloud.Data.Uac.dll	Contiene la configuración de las entidades persistentes en base de datos que interactúan en el sistema.
NHibernate.dll	Librería que permite la conexión y configuración de la base de datos.
Npgsql.dll	Driver de conexión con el proveedor de base de datos PostgreSQL.
FluentNHibernate.dll	Permite crear el mapeo de las entidades persistentes en base de datos.
Microsoft.AspNet.Identity.Core.dll	Librería que maneja las credenciales de usuarios.
IdentityServer3.dll	Librería que permite la configuración de los servicios web.
Microsoft.Owin.Security.OAuth.dll	Permite configurar una aplicación para consumir servicios de un proveedor de identidad.
Owin.dll	Define un interfaz estándar de entre aplicaciones web y servidores web de la plataforma .Net.
Microsoft.Owin.dll	Permite simplificar algunas configuraciones de la librería OWIN.

(Fuente: elaboración propia)

4.2. Modelo de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. Un nodo es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional, que generalmente tiene algo de memoria y, a menudo, capacidad de procesamiento.

La Figura 4.2 muestra el modelo de despliegue del componente de software de la solución PACS-RIS, el cual responde al despliegue de la solución en su conjunto.

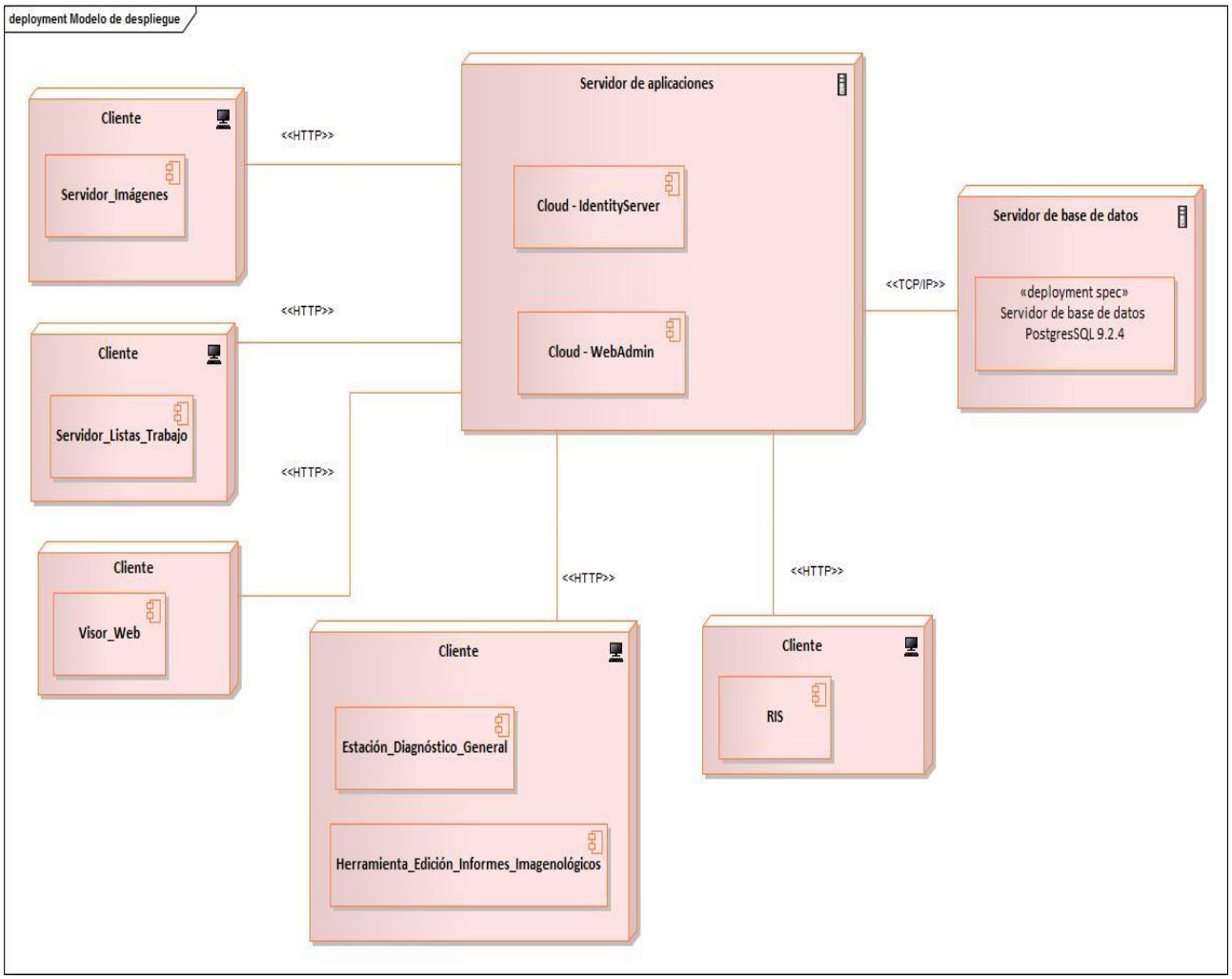


Figura 4.2 Modelo de despliegue del componente para la autenticación de la solución PACS-RIS. (Fuente: elaboración propia)

4.2.1. Descripción de los nodos

Tabla 7 Descripción del modelo de despliegue.

Nodos	Descripción
Cliente	Consumen los servicios brindados por el servidor de imágenes, servidor de listas de trabajo, visor web, estación de diagnóstico general, herramienta de informes imagenológicos (reportador) y el RIS.
Servidor de aplicaciones	Proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas.
Servidor de base de datos	Es donde se almacenan los datos del sistema alas RIS y del servicio web propuesto.

(Fuente: elaboración propia)

4.3. Estándares de codificación

Los estándares de codificación son reglas específicas de cada lenguaje de programación, cuyo cumplimiento reduce de forma significativa, la posibilidad de cometer errores no detectados por los compiladores. Al implementar y verificar el cumplimiento de estos estándares de codificación, se evitan los errores en la introducción del código, reduciendo el tiempo y coste de las actividades de depuración y pruebas necesarias para la detección y corrección de los mismo.

Con vistas a mejorar el entendimiento del código perteneciente al componente para la autenticación unificada de la solución PACS-RIS y alcanzar una uniformidad en el mismo, se empleó el estándar de codificación que se muestra a continuación:

- Usar nombres descriptivos en inglés para las clases, propiedades y métodos.

```
public AccountController(ApplicationUserManager userManager)
{
    UserManager = userManager;
}
```

- Declarar los nombres de los atributos de las clases con underscore (_) y letra inicial minúscula, o iniciarlos con letra inicial mayúscula. En caso de ser un nombre compuesto se utiliza mayúscula también en la nueva palabra.

```
private ApplicationRoleManager _roleManager;
```

```
public virtual int Id { get; set; }
```

- Los nombres de los métodos iniciarán con mayúscula y si son palabras compuestas, se empleará notación CamelCase.

```
public ActionResult CreateRole()
```

- Comentariar las clases y métodos de difícil comprensión.

```
/// <summary>
/// Método para obtener un usuario por su identificador
/// </summary>
/// <param name="userId">
/// El identificador del usuario que se desea buscar
/// </param>
/// <returns>
/// Retorna una entidad de tipo IdentityUser
/// </returns>
0 references
public Task<IdentityUser> FindByIdAsync(int userId)
{
    try
    {
        this.ThrowIfDisposed();
        var user = Task.FromResult(this.Context.Get<IdentityUser>(userId));
        return user;
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

4.4. Tratamiento de errores

Una aplicación C# debe tener capacidad para enfrentarse a lo inesperado. Por muchas comprobaciones de error que se añadan al código, siempre habrá algo que pueda ir mal. Para el manejo de este tipo de situaciones C# ofrece un sistema denominado manejo de excepciones.

Las excepciones no son siempre una señal de que algo catastrófico ha ocurrido en el programa. A menudo es una forma más conveniente de dejar una sección de código que ya no es pertinente o es una señal de que un método no se ha realizado con éxito. Muchos de los métodos de clase de .NET Framework crean excepciones para advertir de una determinada condición (78).

C# proporciona varias palabras clave, try, catch y finally, que permiten a los programas detectar las excepciones, resolverlas y seguir trabajando. Las palabras clave try y catch se utilizan conjuntamente; try para delimitar el bloque de código que podría generar una excepción y catch para contener el código que se ejecutará si la excepción se genera. El código contenido en un bloque finally siempre se ejecuta, se produzca o no una excepción (78).

```
public async Task<ActionResult> Create(RegisterClaimViewModel model)
{
    try
    {
        var claim=new IdentityClaim{ClaimType=model.ClaimType, ClaimValue=model.ClaimValue};
        var result=await _manager.CreateClaimAsync(claim);
        if (result.Succeeded)
        {
            return RedirectToAction("Index", "Claim");
        }
        else
        {
            AddErrors(result);
            return View(model);
        }
    }
    catch (Exception ex)
    {
        ModelState.AddModelError("", ex);
        return View(model);
    }
}
```

Al concluir el presente capítulo se obtuvo como resultado la culminación de la implementación del componente para la autenticación unificada de la solución PACS-RIS y la claridad del código empleado debido a los estándares de codificación utilizados. Con la elaboración del diagrama de componentes y la descripción de sus elementos se centró de manera general la forma de trabajo de cada librería empleada y a través del diagrama de despliegue se mostró la disposición física de los nodos que componen el sistema.

CONCLUSIONES

Una vez finalizado el proceso de investigación se puede arribar a las siguientes conclusiones:

- Se determinó, después de estudiar las tendencias actuales, que los mecanismos de autenticación unificada empleados en los sistemas de salud, no constituyen un recurso viable para la solución PACS-RIS, porque a nivel internacional están concebidos como componentes a la medida para las aplicaciones; y la no existencia de estos en el ámbito nacional.
- Se decidió, luego de analizar varios protocolos de autenticación, aplicar a la solución un mecanismo híbrido entre los protocolos OAuth 2.0 y OpenID Connect porque a partir de su combinación se obtuvo un sistema robusto capaz de unificar el proceso de autenticación de las aplicaciones de la solución PACS-RIS.
- Se desarrolló el componente Celsus.Cloud que permitió unificar el mecanismo de autenticación de la solución PACS-RIS, el cual posibilitó el manejo unificado de usuarios y roles, el uso de tokens para el intercambio seguro de las credenciales del usuario y admitió extender el mecanismo de autenticación a nuevas aplicaciones sin comprometer la seguridad de las ya existentes.

RECOMENDACIONES

Después de realizada la investigación se proponen las siguientes recomendaciones:

- Integrar la solución a los componentes de la solución PACS-RIS mediante la librería Celsus.Cloud.Connector.
- Adicionar al sistema Celsus.Cloud componentes para el manejo de licencias y trazas de la solución PACS-RIS, haciendo al sistema más adaptable a su despliegue en la nube.

REFERENCIAS BIBLIOGRÁFICAS

1. HERNÁNDEZ CÁCERES, José Luis. Identificación de las principales áreas de investigación en informática en salud según los artículos presentados en el 13 Congreso Mundial de Informática Médica y de Salud. [En línea]. 2006. [Accedido 10 marzo 2015]. Disponible en: http://www.rcim.sld.cu/revista_23/articulo_pdf/areasinvestigacion.pdf

2. JIMÉNEZ MORALES, Emilio L. Informática Médica. Antecedentes históricos. Su desarrollo en Cuba hasta 1988. — I Congreso Virtual de Gestión de la Información en Salud. [En línea]. 27 marzo 2009. [Accedido 10 marzo 2015]. Disponible en: http://gis2009.sld.cu/Members/emilio_morales/informatica-medica-antecedentes-historicos-su-desarrollo-en-cuba-hasta-1988/

3. VASILJEVSKI, Nikola y RIS, Frederic. RIS - Sistema de información Radiológica. [En línea]. 2011. [Accedido 10 marzo 2015]. Disponible en: http://openhealth.com.co/es/ris-sistema-de-informacion-radiologica

4. ROVIRA, Francisco Bordils i y DÍAZ, Miguel Chavarría. Almacenamiento y transmisión de imágenes. PACS. [En línea]. 2010. [Accedido 10 marzo 2015]. Disponible en: http://www.conganat.org/SEIS/is/is45/IS45_54.pdf

5. HANSEN, Mark, SIEGEL, Michael y MADNICK, Stuart. Data Integration Using Web Services. [En línea]. 2015. [Accedido 10 marzo 2015]. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.161.4783&rep=rep1&type=pdf

6. JEAN-PHILIPPE y ADAM, Dunkels. JSON (JavaScript Object Notation). [En línea]. 2010. [Accedido 10 marzo 2015]. Disponible en: https://www.json.com/

7. HERNÁNDEZ SAMPIERI, C. Roberto, COLLADO, Carlos Fernández y LUCIO, Pila Baptista. *Metodología de la Investigación*. México: Editorial Mc Graw Hill, 2010. ISBN 968-422-931-3.

8. CHÁVEZ, Caridad Fresno. Sistemas de atención de salud (I). [En línea]. 1996. [Accedido 10 marzo 2015]. Disponible en: http://www.bvs.sld.cu/revistas/spu/vol22_1_96/spu05196.htm

9. RIS, Frederic. Sistema de Información de Radiología (RIS) | Cualquier Cosa de Tecnología. [En línea]. 4 diciembre 2014. [Accedido 10 marzo 2015]. Disponible en: https://cualquiercosadetecnologia.wordpress.com/2014/04/12/sistema-de-informacion-de-radiologia-ris/

10. DÍAZ, Carlos Guzmán y AGUILAR, Denys Bárbaro Vega. Sistema para el Almacenamiento y Transmisión de Imágenes Médicas Versión 3-10. *Ediciones pensando en el futuro* [En línea]. 2013. [Accedido 10 marzo 2015]. Disponible en: http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/viewFile/178/10

11. AGUILAR, Ana Bertha Pimente y ORTIZ-POSADAS, M. R. Sistema PACS/RIS FutuRIS. [En línea]. 2000. [Accedido 25 marzo 2015]. Disponible en: http://es.slideshare.net/joaquinglz/futuris-system-es

12. CHANG y TSAI. Artículos - seguridad - Seguridad informática. Capítulo 4 Control de acceso. [En línea]. 2011. [Accedido 10 marzo 2015]. Disponible en: http://mygnet.net/codigos/vb/varios/control_de_acceso.438

13. BORGHELLO, Cristian Fabián y ARNES, André. Seguridad Informática / Seguridad Lógica - Identificación y Autentificación. [En línea]. 2000. [Accedido 10 marzo 2015]. Disponible en: http://www.segu-info.com.ar/logica/identificacion.htm

14. BORGHELLO, Cristian Fabián y ARNES, André. Seguridad Informática / Vulnerar para Proteger. [En línea]. 2000. [Accedido 10 marzo 2015]. Disponible en: http://www.segu-info.com.ar/protección/vulnerar.htm

15. FABRÍZIO, Rosa. Conceptos de Autenticación y Autorización. [En línea]. 2006. [Accedido 10 marzo 2015]. Disponible en: http://www.bdat.net/documentos/cursos/ar01s96.html

16. MICROSOFT. Introducción al acceso inalámbrico autenticado mediante 802.1X. [En línea]. 2015. [Accedido 10 marzo 2015]. Disponible en: https://technet.microsoft.com/es-es/library/hh994700.aspx

17. JOSÉ, Diogo. ¿Qué es la autenticación CRAM? [En línea]. 2014. [Accedido 10 marzo 2015]. Disponible en: http://ordenador.wingwit.com/Redes/network-security/76108.html

18. TORRES, Gisela. RedIRIS - Autenticación de usuarios. [En línea]. 2015. [Accedido 10 marzo 2015]. Disponible en: http://www.rediris.es/cert/doc/unixsec/node14.html

19. GARNER, Jerry y FARAH, Helúe. Tipos de autenticación web | eHow en Español. [En línea]. 2013. [Accedido 10 marzo 2015]. Disponible en: http://www.ehowenespanol.com/tipos-autenticacion-web-lista_314874/

20. MANUEL, José. Doble autenticación unificada - Conocimientos informáticos - kilbot.net. [En línea]. 2011. [Accedido 10 marzo 2015]. Disponible en: http://www.kilbot.net/stag/doble-autenticacion-unificada/

21. MACHADO, Ivan Barbosa. IBM Knowledge Center. [En línea]. 2000. [Accedido 26 marzo 2015]. Disponible en: http://www-01.ibm.com/support/knowledgecenter/SS2GNX_5.1.1/com.ibm.tivoli.tpm.sec.doc/users/csec_twofactor.html?lang=es

22. MACHADO, Ivan Barbosa. 2FA-IBM Knowledge Center. [En línea]. 2000. [Accedido 26 marzo 2015]. Disponible en: http://www-01.ibm.com/support/knowledgecenter/SS2GNX_5.1.1/com.ibm.tivoli.tpm.sec.doc/users/csec_twofactorexample.html?lang=es

23. MIERES, Jorge. Autenticación de 2FA. [En línea]. 2013. [Accedido 26 marzo 2015]. Disponible en: http://www.pcactual.com/etiqueta/7536/autenticacion.html

24. TUÑÓN, Sandor Ernesto. Doble autenticación unificada - Conocimientos informáticos - kilbot.net. [En línea]. 2010. [Accedido 26 marzo 2015]. Disponible en: http://www.kilbot.net/stag/doble-autenticacion-unificada/

25. SÁNCHEZ GUERRERO, Rosa. SAML 2.0 simplifica la federación de identidades | | NetworkWorld. [En línea]. 2013. [Accedido 14 marzo 2015]. Disponible en: http://www.networkworld.es/archive/saml-20-simplifica-la-federacion-de-identidades

26. HERNÁNDEZ, Ardieta, CARVAJAL-VIÓN, José Fernando y HEPPE, John. Autenticación de inicio de sesión único mediante SAML 2.0. [En línea]. 2014. [Accedido 14 marzo 2015]. Disponible en: http://help.exacttarget.com/es/documentation/exacttarget/administrador/autenticacion_de_inicio_de_sesion_unico_mediante_saml_20/Jorge López

27. BLANCO, A. Cover Pages: Liberty Identity Web Services Framework (ID-WSF) Supports SAML 2.0. [En línea]. 11 febrero 2005. [Accedido 26 marzo 2015]. Disponible en: http://xml.coverpages.org/ni2005-02-11-b.html

28. HAMMER-LAHAV, Eran. Introduction — OAuthn. [En línea]. 5 septiembre 2007. [Accedido 10 marzo 2015]. Disponible en: http://oauth.net/about/

29. HAMMER, Eran y ATWOOD, Mark. OAuth Core 1.0a. [En línea]. 24 junio 2009. [Accedido 10 marzo 2015]. Disponible en: http://oauth.net/core/1.0a/

30. MANGAL, Agraj. OAuth 2.0 Authorization Code Requests and Responses | tutorials.jenkov.com. [En línea]. 12 julio 2013. [Accedido 10 marzo 2015]. Disponible en: http://tutorials.jenkov.com/oauth2/authorization-code-request-response.html

31. HAMMER, Eran. OAuth: Cómo administrar la clave y el secreto OAuth - Ayuda de Google Apps. [En línea]. 2010. [Accedido 10 marzo 2015]. Disponible en: https://support.google.com/a/answer/162105?hl=es&ref_topic=2759255

32. HAMMER, Eran. Draft-Ietf-OAuth-v2-31 - The OAuth 2.0 Authorization Framework. [En línea]. Julio 2012. [Accedido 10 marzo 2015]. Disponible en: https://tools.ietf.org/html/draft-ietf-oauth-v2-31

33. MANGAL, Agraj. OAuth 2.0 Tutorial | tutorials.jenkov.com. [En línea]. 12 julio 2013. [Accedido 10 marzo 2015]. Disponible en: http://tutorials.jenkov.com/oauth2/index.html

34. MANGAL, Agraj. OAuth 2.0 Roles | tutorials.jenkov.com. [En línea]. 12 julio 2013. [Accedido 10 marzo 2015]. Disponible en: http://tutorials.jenkov.com/oauth2/roles.html

35. MOSQUERA, Diego. ¿Qué es un API? [En línea]. 2010. [Accedido 10 marzo 2015]. Disponible en: http://www.rastersoft.com/OS2/CURSO/APIEXPL.HTM

36. JONES, Michael B. Review of proposed final OpenID 2.0 to OpenID Connect Migration specification | OpenID. [En línea]. 2014. [Accedido 11 marzo 2015]. Disponible en: http://openid.net/2015/02/01/review-of-proposed-final-openid-2-0-to-openid-connect-migration-specification/

37. RUBÉN, Cristian. OpenID Connect | OpenID. [En línea]. 2012. [Accedido 11 marzo 2015]. Disponible en: http://openid.net/tag/openid-connect/

38. DUJO, Tapiador Del. OpenID Connect FAQ and Q&As | OpenID. [En línea]. 2008. [Accedido 11 marzo 2015]. Disponible en: http://openid.net/connect/faq/

39. THOMPSON, James Walter. JSON Web Tokens - jwt.io. [En línea]. 2000. [Accedido 25 marzo 2015]. Disponible en: http://jwt.io/

40. SÁNCHEZ, Jordi. *Sistemas de autenticación y autorización en internet*. España: Editorial Computación Distribuida, 2011. ISBN 84-9788-329-2.

41. FACEBOOK. Facebook Developers. [En línea]. 2015. [Accedido 12 junio 2015]. Disponible en: https://developers.facebook.com/docs/facebook-login/login-flow-for-web/v2.3

42. GOOGLE. Google Developers. [En línea]. 2015. [Accedido 13 junio 2015]. Disponible en: https://developers.google.com/identity/protocols/OpenIDConnect

43. MEDINA, Nestor, Brito. SIAPS Program. [En línea]. 2012. [Accedido 26 marzo 2015]. Disponible en: http://siapsprogram.org/

44. GUERRERO, Mirta Prendes, MANTECÓN, Mercedes López y SABURIT, Arelys Borrell. OMS | Acerca de HINARI. [En línea]. 2008. [Accedido 26 marzo 2015]. Disponible en:

http://www.who.int/hinari/about/es/

45. TRAMULLAS, Jesús. Digital Farmadrid / 39 / Bélgica ensaya un nuevo sistema de remuneración para las farmacias. [En línea]. Junio 2010. [Accedido 27 marzo 2015]. Disponible en: http://farmadrid.cofm.es/es/index.asp?MP=28&MS=126&MN=2

46. SALVADOR, Javier Cabo. Alternativas de la historia de salud regional/nacional | Gestión Sanitaria - Grupos Relacionados por el Diagnóstico (GRD). [En línea]. Febrero 2015. [Accedido 27 marzo 2015]. Disponible en: http://www.gestion-sanitaria.com/2-alternativas-historia-salud-regional-nacional.html

47. ETRETOS, Javier. Expediente Digital Único en Salud (EDUS). [En línea]. 2014. [Accedido 27 marzo 2015]. Disponible en: http://programafrida.net/projects/projects/view/428

48. ROJAS, David y CARNICERO, Javier. Sanidad | Indra. [En línea]. 2000. [Accedido 27 marzo 2015]. Disponible en: http://www.indracompany.com/sector/sanidad

49. SOUZA, Renilson Rehem de. El sistema público de salud en Brasil.pdf. [En línea]. Agosto 2002. [Accedido 27 marzo 2015]. Disponible en: http://www.redadultosmayores.com.ar/Material%202014/ArchivosSEGURIDADSOCIAL/9%20El%20sistema%20publico%20de%20salud%20en%20Brasil.pdf

50. MACGREGOR, William, MEHTA, Ketan, COOPER, David y SCARFONE, Karen. *NIST SP 800-116, A Recommendation for the Use of PIV Credentials in Physical Access Control Systems (PACS) - SP800-116.pdf* [En línea]. 2012. NIST Special Publication 800-116. [Accedido 26 marzo 2015]. Disponible en: http://csrc.nist.gov/publications/nistpubs/800-116/SP800-116.pdf

51. ELIESER, Adrián y FUENTES, Alcolea. Alliance Activities : Publications : Authentication Mechanisms for PACS » Smart Card Alliance. [En línea]. 2009. [Accedido 26 marzo 2015]. Disponible en: http://www.smartcardalliance.org/publications-authentication-mechanisms-for-pacs/

52. MARTÍNEZ, Yoandy González, GARCÍA, Marileisy Castillo y QUEVEDO, Yunier Silva. Revista Cubana de Informática Médica - Componente para la lectura de datos por alas-HIS desde máquinas de anestesia. [En línea]. Junio 2014. Vol. vol.6 no.1. [Accedido 26 marzo 2015]. Disponible en: http://scielo.sld.cu/scielo.php?pid=S1684-18592014000100011&script=sci_arttext

53. SMARTH, Dayana Joseph, GONZÁLEZ, Yudiel La Rosa, ARMAS, Elvismary Molina De y PERODÍN, Yusdenis Sánchez. Revista Cubana de Informática Médica - AlasMEDIGEN v1.1: Sistema informático de Genética Médica. [En línea]. Julio 2014. Vol. vol.6. [Accedido 26 marzo 2015].

Disponible en: http://scielo.sld.cu/scielo.php?pid=S1684-18592014000200008&script=sci_arttext

54. VÁZQUES, Miguel R. Centro Nacional de Información de Ciencias Médicas. [En línea]. 1999. Disponible en: http://www.sld.cu/

55. BOOCH, Rumbaugh Grady y JAMES, Ivar Jacobson. *El Proceso Unificado de Desarrollo de Software.* Madrid: Editorial Pearson Educación, 2000. ISBN 84-7829-036-2.

56. CRAIG, LARMAN. Unified Modeling Language (UML). [En línea]. 2014. [Accedido 11 marzo 2015]. Disponible en: http://www.uml.org/

57. MICROSOFT. Microsoft .NET Framework Download - Softpedia. [En línea]. Febrero 2015. [Accedido 11 marzo 2015]. Disponible en: http://www.softpedia.com/get/Others/Signatures-Updates/NET-Framework.shtml

58. Andrés Vallés Botella. *ASP.NET MVC 4* [En línea]. Campus de Sant Vicent del Raspeig | 03690 | España. [Accedido 13 octubre 2014]. Disponible en: http://si.ua.es/es/

59. GABRIEL, Cándido. OWIN — Open Web Interface for .NET. [En línea]. 2010. [Accedido 11 marzo 2015]. Disponible en: http://owin.org/

60. MICROSOFT. OWIN and Katana | The ASP.NET Site. [En línea]. 2015. [Accedido 11 marzo 2015]. Disponible en: http://www.asp.net/aspnet/overview/owin-and-katana

61. SERRANO, Jorge. Información general sobre ASP.NET. [En línea]. 20014. [Accedido 11 marzo 2015]. Disponible en: http://ASP.NET.htm

62. BERZAL, Fernando. Introducción a ASP.NET MVC. [En línea]. 2009. [Accedido 11 marzo 2015]. Disponible en: http://www.desarrolloweb.com/articulos/introduccion-asp-net-mvc-dotnet.html

63. CHAUHAN, Shailendra. What is Web API and why to use it? [En línea]. 25 julio 2014. [Accedido 11 marzo 2015]. Disponible en: http://www.dotnet-tricks.com/Tutorial/webapi/VG9K040413-What-is-Web-API-and-why-to-use-it-?.html

64. CÁNDIDO, Gabriel García, SEBASTIÀ, Natividad, BLASCO, Esther y SORIANO DEL CASTILLO, José Miguel. ASP.NET - Introducción al proyecto Katana. [En línea]. 2014. [Accedido 12 marzo 2015]. Disponible en: https://msdn.microsoft.com/es-es/magazine/dn451439.aspx

65. TAPIADOR, Antonio. IdentityServer/IdentityServer3 · GitHub. [En línea]. 2000. [Accedido 14 marzo 2015]. Disponible en: https://github.com/IdentityServer/IdentityServer3

66. CUBERO, Juan Carlos, CORTIJO, Francisco J. y BERZAL, Fernando. PostgreSQL Overview

| EnterpriseDB. [En línea]. 2005. [Accedido 11 marzo 2015]. Disponible en: http://www.enterprisedb.com/products-services-training/products/postgresql-overview

67. JAVIER, Francisco. Introducción al lenguaje C# y .NET Framework. [En línea]. 2002. [Accedido 11 marzo 2015]. Disponible en: https://msdn.microsoft.com/es-es/library/z1zx9t92.aspx

68. W, BRIAN Kernighan, RITCHIE, DENNIS y GREGORY, KATE. Visual Studio Ultimate 2013 en Español Full Final + Serial de Activación + Crack + Key | PROGRAMAS WEB FULL. [En línea]. 2013. [Accedido 11 marzo 2015]. Disponible en: http://www.programaswebfull.com/2013/12/visual-studio-ultimate-2013-espanol-full-final-serial-activacion-crack-key.html

69. ARCINIEGAS, José Luis y GALLÓN, Álvaro Rendón. Enterprise Architect v7.1 - Sparx Systems. [En línea]. 2010. [Accedido 11 marzo 2015]. Disponible en: http://www.sparxsystems.com.au/press/articles/ea71.html

70. GENEVISH, Scott. TortoiseSVN - Home. [En línea]. 2013. [Accedido 11 marzo 2015]. Disponible en: http://tortoisesvn.net/

71. LARMAN, Craig, RUMBAUGH, James y BOOCH, Grady. *Microsoft Word - Modelo del Dominio.docx - Modelo_Dominio.pdf* [En línea]. 2003. UML y Patrones. 2ª Edición. [Accedido 12 marzo 2015]. Disponible en: http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf

72. QUIROGA, Juan Pablo. Requerimientos Funcionales y No Funcionales. [En línea]. 2011. [Accedido 12 marzo 2015]. Disponible en: https://sistemas.uniandes.edu.co/~csof5101/dokuwiki/lib/exe/fetch.php?media=principal:csof5101-requerimientos.pdf

73. HASSAN, Yusef, LAZZA, Ghzala y FERNÁNDEZ, Francisco J. Martín. Diseño de Software. [En línea]. 2004. [Accedido 12 marzo 2015]. Disponible en: http://es.slideshare.net/montoya118/diseo-de-software-10781906

74. BERZAL, Fernando. Microsoft Word - 2 OOP.doc - 3C-Relaciones.pdf. [En línea]. 2010. [Accedido 25 marzo 2015]. Disponible en: http://elvex.ugr.es/decsai/java/pdf/3C-Relaciones.pdf

75. M. ABBOTT., Michael. Diagrama de secuencia. [En línea]. 2010. [Accedido 25 marzo 2015]. Disponible en: http://es.slideshare.net/still01/diagrama-de-clases-16815247?related=1

76. VÁZQUEZ, Arturo Cuauhtémoc Salinas. Arquitectura - Software libre o propietario - Documentos. [En línea]. Marzo 2014. [Accedido 27 marzo 2015]. Disponible en: http://www.buenastareas.com/ensayos/Software-Libre-o-Propietario/1497209.html

77. MICROSOFT. El Patrón Singleton. [En línea]. 2000. [Accedido 27 marzo 2015]. Disponible en: https://msdn.microsoft.com/es-es/library/bb972272.aspx

78. MICROSOFT. Control de errores y excepciones (Visual C# Express). [En línea]. 2008. Disponible en: https://msdn.microsoft.com/es-es/library/384webt3%28v=vs.90%29.aspx

BIBLIOGRAFÍA

AGUILAR, Ana Bertha Pimente y ORTIZ-POSADAS, M. R. Sistema PACS/RIS FutuRIS. [En línea]. 2000. [Accedido 25 marzo 2015]. Disponible en: http://es.slideshare.net/joaquinglz/futuris-system-es

Andrés Vallés Botella. *ASP.NET MVC 4* [En línea]. Campus de Sant Vicent del Raspeig | 03690 | España. [Accedido 13 octubre 2014]. Disponible en: http://si.ua.es/es/

ARCINIEGAS, José Luis y GALLÓN, Álvaro Rendón. Enterprise Architect v7.1 - Sparx Systems. [En línea]. 2010. [Accedido 11 marzo 2015]. Disponible en: http://www.sparxsystems.com.au/press/articles/ea71.html

BERZAL, Fernando. Introducción a ASP.NET MVC. [En línea]. 2009. [Accedido 11 marzo 2015]. Disponible en: http://www.desarrolloweb.com/articulos/introduccion-asp-net-mvc-dotnet.html

BERZAL, Fernando. Microsoft Word - 2 OOP.doc - 3C-Relaciones.pdf. [En línea]. 2010. [Accedido 25 marzo 2015]. Disponible en: http://elvex.ugr.es/decsai/java/pdf/3C-Relaciones.pdf

BLANCO, A. Cover Pages: Liberty Identity Web Services Framework (ID-WSF) Supports SAML 2.0. [En línea]. 11 Febrero 2005. [Accedido 26 marzo 2015]. Disponible en: http://xml.coverpages.org/ni2005-02-11-b.html

BOOCH, Rumbaugh Grady y JAMES, Ivar Jacobson. El Proceso Unificado de Desarrollo de Software. Madrid: Editorial Pearson Educación, 2000. ISBN 84-7829-036-2.

BORGHELLO, Cristian Fabián y ARNES, André. Seguridad Informática / Seguridad Lógica - Identificación y Autentificación. [En línea]. 2000. [Accedido 10 marzo 2015]. Disponible en: http://www.segu-info.com.ar/logica/identificacion.htm

BORGHELLO, Cristian Fabián y ARNES, André. Seguridad Informática / Vulnerar para Proteger. [En línea]. 2000. [Accedido 10 marzo 2015]. Disponible en: http://www.segu-info.com.ar/protección/vulnerar.htm

CÁNDIDO, Gabriel García, SEBASTIÀ, Natividad, BLASCO, Esther y SORIANO DEL CASTILLO, José Miguel. ASP.NET - Introducción al proyecto Katana. [En línea]. 2014. [Accedido 12 marzo 2015]. Disponible en: https://msdn.microsoft.com/es-

es/magazine/dn451439.aspx

CHANG y TSAI. Artículos - seguridad - Seguridad informática. Capítulo 4 Control de acceso. [En línea]. 2011. [Accedido 10 marzo 2015]. Disponible en: http://mygnet.net/codigos/vb/varios/control_de_acceso.438

CHAUHAN, Shailendra. What is Web API and why to use it ? [En línea]. 25 julio 2014. [Accedido 11 marzo 2015]. Disponible en: http://www.dotnet-tricks.com/Tutorial/webapi/VG9K040413-What-is-Web-API-and-why-to-use-it-?.html

CHÁVEZ, Caridad Fresno. Sistemas de atención de salud (I). [En línea]. 1996. [Accedido 10 marzo 2015]. Disponible en: http://www.bvs.sld.cu/revistas/spu/vol22_1_96/spu05196.htm

CRAIG, LARMAN. Unified Modeling Language (UML). [En línea]. 2014. [Accedido 11 marzo 2015]. Disponible en: http://www.uml.org/

CUBERO, Juan Carlos, CORTIJO, Francisco J. y BERZAL, Fernando. PostgreSQL Overview | EnterpriseDB. [En línea]. 2005. [Accedido 11 marzo 2015]. Disponible en: http://www.enterprisedb.com/products-services-training/products/postgresql-overview

DÍAZ, Carlos Guzmán y AGUILAR, Denys Bárbaro Vega. Sistema para el Almacenamiento y Transmisión de Imágenes Médicas Versión 3-10. Ediciones pensando en el futuro [En línea]. 2013. [Accedido 10 marzo 2015]. Disponible en: http://www.informatica2013.sld.cu/index.php/informaticasalud/2013/paper/viewFile/178/10

DUJO, Tapiador Del. OpenID Connect FAQ and Q&As | OpenID. [En línea]. 2008. [Accedido 11 marzo 2015]. Disponible en: http://openid.net/connect/faq/

ELIESER, Adrián y FUENTES, Alcolea. Alliance Activities : Publications : Authentication Mechanisms for PACS » Smart Card Alliance. [En línea]. 2009. [Accedido 26 marzo 2015]. Disponible en: http://www.smartcardalliance.org/publications-authentication-mechanisms-for-pacs/

ETRETOS, Javier. Expediente Digital Único en Salud (EDUS). [En línea]. 2014. [Accedido 27 marzo 2015]. Disponible en: http://programafrida.net/projects/projects/view/428

FABRÍZIO, Rosa. Conceptos de Autenticación y Autorización. [En línea]. 2006.

[Accedido 10 marzo 2015]. Disponible en: http://www.bdat.net/documentos/cursos/ar01s96.html

FACEBOOK. Facebook Developers. [En línea]. 2015. [Accedido 12 junio 2015]. Disponible en: https://developers.facebook.com/docs/facebook-login/login-flow-for-web/v2.3

GABRIEL, Cándido. OWIN — Open Web Interface for .NET. [En línea]. 2010. [Accedido 11 marzo 2015]. Disponible en: http://owin.org/

GARNER, Jerry y FARAH, Helúe. Tipos de autenticación web | eHow en Español. [En línea]. 2013. [Accedido 10 marzo 2015]. Disponible en: http://www.ehowenespanol.com/tipos-autenticacion-web-lista_314874/

GENEVISH, Scott. TortoiseSVN - Home. [En línea]. 2013. [Accedido 11 marzo 2015]. Disponible en: http://tortoisesvn.net/

GOOGLE. Google Developers. [En línea]. 2015. [Accedido 13 junio 2015]. Disponible en: https://developers.google.com/identity/protocols/OpenIDConnect

GUERRERO, Mirta Prendes, MANTECÓN, Mercedes López y SABURIT, Arelys Borrell. OMS | Acerca de HINARI. [En línea]. 2008. [Accedido 26 marzo 2015]. Disponible en: http://www.who.int/hinari/about/es/

HAMMER, Eran y ATWOOD, Mark. OAuth Core 1.0a. [En línea]. 24 junio 2009. [Accedido 10 marzo 2015]. Disponible en: http://oauth.net/core/1.0a/

HAMMER, Eran. Draft-Ietf-OAuth-v2-31 - The OAuth 2.0 Authorization Framework. [En línea]. Julio 2012. [Accedido 10 marzo 2015]. Disponible en: https://tools.ietf.org/html/draft-ietf-oauth-v2-31

HAMMER, Eran. OAuth: Cómo administrar la clave y el secreto OAuth - Ayuda de Google Apps. [En línea]. 2010. [Accedido 10 marzo 2015]. Disponible en: https://support.google.com/a/answer/162105?hl=es&ref_topic=2759255

HAMMER-LAHAV, Eran. Introduction — OAuthn. [En línea]. 5 septiembre 2007. [Accedido 10 marzo 2015]. Disponible en: http://oauth.net/about/

HANSEN, Mark, SIEGEL, Michael y MADNICK, Stuart. Data Integration Using Web Services. [En línea]. 2015. [Accedido 10 marzo 2015]. Disponible en: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.161.4783&rep=rep1&type=pdf

HASSAN, Yusef, LAZZA, Ghzala y FERNÁNDEZ, Francisco J. Martín. Diseño de Software. [En línea]. 2004. [Accedido 12 marzo 2015]. Disponible en: http://es.slideshare.net/montoya118/diseo-de-software-10781906

HERNÁNDEZ CÁCERES, José Luis. Identificación de las principales áreas de investigación en informática en salud según los artículos presentados en el 13 Congreso Mundial de Informática Médica y de Salud. [En línea]. 2006. [Accedido 10 marzo 2015]. Disponible en: http://www.rcim.sld.cu/revista_23/articulo_pdf/areasinvestigacion.pdf

HERNÁNDEZ, Ardieta, CARVAJAL-VIÓN, José Fernando y HEPPE, John. Autenticación de inicio de sesión único mediante SAML 2.0. [En línea]. 2014. [Accedido 14 marzo 2015]. Disponible en: http://help.exacttarget.com/es/documentation/exacttarget/administrador/autenticacion_de_inicio_de_sesion_unico_mediante_saml_20/Jorge López

HERNÁNDEZ SAMPIERI, C. Roberto, COLLADO, Carlos Fernández y LUCIO, Pila Baptista. Metodología de la Investigación. México: Editorial Mc Graw Hill, 2010. ISBN 968-422-931-3.

JAVIER, Francisco. Introducción al lenguaje C# y .NET Framework. [En línea]. 2002. [Accedido 11 marzo 2015]. Disponible en: https://msdn.microsoft.com/es-es/library/z1zx9t92.aspx

JEAN-PHILIPPE y ADAM, Dunkels. JSON (JavaScript Object Notation). [En línea]. 2010. [Accedido 10 marzo 2015]. Disponible en: https://www.json.com/

JIMÉNEZ MORALES, Emilio L. Informática Médica. Antecedentes históricos. Su desarrollo en Cuba hasta 1988. — I Congreso Virtual de Gestión de la Información en Salud. [En línea]. 27 marzo 2009. [Accedido 10 marzo 2015]. Disponible en: http://gis2009.sld.cu/Members/emilio_morales/informatica-medica-antecedentes-historicos-su-desarrollo-en-cuba-hasta-1988/

JONES, Michael B. Review of proposed final OpenID 2.0 to OpenID Connect Migration specification | OpenID. [En línea]. 2014. [Accedido 11 marzo 2015]. Disponible en: http://openid.net/2015/02/01/review-of-proposed-final-openid-2-0-to-openid-connect-migration-specification/

JOSÉ, Diogo. ¿Qué es la autenticación CRAM? [En línea]. 2014. [Accedido 10 marzo 2015]. Disponible en: http://ordenador.wingwit.com/Redes/network-security/76108.html

LARMAN, Craig, RUMBAUGH, James y BOOCH, Grady. Microsoft Word - Modelo del Dominio.docx - Modelo_Dominio.pdf [En línea]. 2003. UML y Patrones. 2a Edición. [Accedido 12 marzo 2015]. Disponible en: http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf

M. ABBOTT., Michael. Diagrama de secuencia. [En línea]. 2010. [Accedido 25 marzo 2015]. Disponible en: http://es.slideshare.net/still01/diagrama-de-clases-16815247?related=1

MACGREGOR, William, MEHTA, Ketan, COOPER, David y SCARFONE, Karen. NIST SP 800-116, A Recommendation for the Use of PIV Credentials in Physical Access Control Systems (PACS) - SP800-116.pdf [En línea]. 2012. NIST Special Publication 800-116. [Accedido 26 marzo 2015]. Disponible en: http://csrc.nist.gov/publications/nistpubs/800-116/SP800-116.pdf

MACHADO, Ivan Barbosa. 2FA-IBM Knowledge Center. [En línea]. 2000. [Accedido 26 marzo 2015]. Disponible en: http://www-01.ibm.com/support/knowledgecenter/SS2GNX_5.1.1/com.ibm.tivoli.tpm.sec.doc/users/csec_twofactorexample.html?lang=es

MACHADO, Ivan Barbosa. IBM Knowledge Center. [En línea]. 2000. [Accedido 26 marzo 2015]. Disponible en: http://www-01.ibm.com/support/knowledgecenter/SS2GNX_5.1.1/com.ibm.tivoli.tpm.sec.doc/users/csec_twofactor.html?lang=es

MANGAL, Agraj. OAuth 2.0 Authorization Code Requests and Responses | tutorials.jenkov.com. [En línea]. 12 julio 2013. [Accedido 10 marzo 2015]. Disponible en: http://tutorials.jenkov.com/oauth2/authorization-code-request-response.html

MANGAL, Agraj. OAuth 2.0 Roles | tutorials.jenkov.com. [En línea]. 12 julio 2013. [Accedido 10 marzo 2015]. Disponible en: http://tutorials.jenkov.com/oauth2/roles.html

MANGAL, Agraj. OAuth 2.0 Tutorial | tutorials.jenkov.com. [En línea]. 12 julio 2013. [Accedido 10 marzo 2015]. Disponible en: http://tutorials.jenkov.com/oauth2/index.html

MANUEL, José. Doble autenticación unificada - Conocimientos informáticos - kilbot.net. [En línea]. 2011. [Accedido 10 marzo 2015]. Disponible en: http://www.kilbot.net/stag/doble-autenticacion-unificada/

MARTÍNEZ, Yoandy González, GARCÍA, Marileisy Castillo y QUEVEDO, Yunier Silva. Revista

Cubana de Informática Médica - Componente para la lectura de datos por alas-HIS desde máquinas de anestesia. [En línea]. Junio 2014. Vol. vol.6 no.1. [Accedido 26 marzo 2015]. Disponible en: http://scielo.sld.cu/scielo.php?pid=S1684-18592014000100011&script=sci_arttext

MEDINA, Nestor, Brito. SIAPS Program. [En línea]. 2012. [Accedido 26 marzo 2015]. Disponible en: http://siapsprogram.org/

MICROSOFT. Control de errores y excepciones (Visual C# Express). [En línea]. 2008. Disponible en: https://msdn.microsoft.com/es-es/library/384webt3%28v=vs.90%29.aspx

MICROSOFT. El Patrón Singleton. [En línea]. 2000. [Accedido 27 marzo 2015]. Disponible en: https://msdn.microsoft.com/es-es/library/bb972272.aspx

MICROSOFT. Introducción al acceso inalámbrico autenticado mediante 802.1X. [En línea]. 2015. [Accedido 10 marzo 2015]. Disponible en: https://technet.microsoft.com/es-es/library/hh994700.aspx

MICROSOFT. Microsoft .NET Framework Download - Softpedia. [En línea]. Febrero 2015. [Accedido 11 marzo 2015]. Disponible en: http://www.softpedia.com/get/Others/Signatures-Updates/NET-Framework.shtml

MICROSOFT. OWIN and Katana | The ASP.NET Site. [En línea]. 2015. [Accedido 11 marzo 2015]. Disponible en: http://www.asp.net/aspnet/overview/owin-and-katana

MIERES, Jorge. Autenticación de 2FA. [En línea]. 2013. [Accedido 26 marzo 2015]. Disponible en: http://www.pcactual.com/etiqueta/7536/autenticacion.html

MOSQUERA, Diego. ¿Qué es un API? [En línea]. 2010. [Accedido 10 marzo 2015]. Disponible en: http://www.rastersoft.com/OS2/CURSO/APIEXPL.HTM

QUIROGA, Juan Pablo. Requerimientos Funcionales y No Funcionales. [En línea]. 2011. [Accedido 12 marzo 2015]. Disponible en: https://sistemas.uniandes.edu.co/~csof5101/dokuwiki/lib/exe/fetch.php?media=principal:csof5101-requerimientos.pdf

RIS, Frederic. Sistema de Información de Radiología (RIS) | Cualquier Cosa de Tecnología. [En línea]. 4 diciembre 2014. [Accedido 10 marzo 2015]. Disponible en:

https://cualquiercosadetecnologia.wordpress.com/2014/04/12/sistema-de-informacion-de-radiologia-ris/

ROJAS, David y CARNICERO, Javier. Sanidad | Indra. [En línea]. 2000. [Accedido 27 marzo 2015]. Disponible en: http://www.indracompany.com/sector/sanidad

ROVIRA, Francisco Bordils i y DÍAZ, Miguel Chavarría. Almacenamiento y transmisión de imágenes. PACS. [En línea]. 2010. [Accedido 10 marzo 2015]. Disponible en: http://www.conganat.org/SEIS/is/is45/IS45_54.pdf

RUBÉN, Cristian. OpenID Connect | OpenID. [En línea]. 2012. [Accedido 11 marzo 2015]. Disponible en: http://openid.net/tag/openid-connect/

SALVADOR, Javier Cabo. Alternativas de la historia de salud regional/nacional | Gestión Sanitaria - Grupos Relacionados por el Diagnóstico (GRD). [En línea]. Febrero 2015. [Accedido 27 marzo 2015]. Disponible en: http://www.gestion-sanitaria.com/2-alternativas-historia-salud-regional-nacional.html

SÁNCHEZ GUERRERO, Rosa. SAML 2.0 simplifica la federación de identidades | | NetworkWorld. [En línea]. 2013. [Accedido 14 marzo 2015]. Disponible en: http://www.networkworld.es/archive/saml-20-simplifica-la-federacion-de-identidades

SÁNCHEZ, Jordi. Sistemas de autenticación y autorización en internet. España: Editorial Computación Distribuida, 2011. ISBN 84-9788-329-2.

SÁNCHEZ, Jordi. Sistemas de autenticación y autorización en internet. España: Editorial Computación Distribuida, 2011. ISBN 84-9788-329-2.

SCHENKER, Gabriel Nicolás y CURE, Aaron. NHibernate 3. BIRMINGHAM - MUMBAI: Editorial Copyright © 2011 Packt Publishing, 2011. ISBN 00596.

SERRANO, Jorge. Información general sobre ASP.NET. [En línea]. 20014. [Accedido 11 marzo 2015]. Disponible en: http://ASP.NET.htm

SMARTH, Dayana Joseph, GONZÁLEZ, Yudiel La Rosa, ARMAS, Elvismary Molina De y PERODÍN, Yusdenis Sánchez. Revista Cubana de Informática Médica - AlasMEDIGEN v1.1: Sistema informático de Genética Médica. [En línea]. Julio 2014. Vol. vol.6. [Accedido 26 marzo 2015].

Disponible en: http://scielo.sld.cu/scielo.php?pid=S1684-18592014000200008&script=sci_arttext

SOUZA, Renilson Rehem de. El sistema público de salud en Brasil.pdf. [En línea]. Agosto 2002. [Accedido 27 marzo 2015]. Disponible en: http://www.redadultosmayores.com.ar/Material%202014/ArchivosSEGURIDADSOCIAL/9%20El%20sistema%20publico%20de%20salud%20en%20Brasil.pdf

SPANCER, Travis. API Security. [En línea]. 2015. [Accedido 10 mayo 2015]. Disponible en: http://nordicapis.com/api-security-oauth-openid-connect-depth/

SPANCER, Travis. Twobo Technologies. [En línea]. 2013. [Accedido 13 mayo 2015]. Disponible en: http://www.twobotechnologies.com/blog/2012/08/cloud-security-standards.html

TAPIADOR, Antonio. IdentityServer/IdentityServer3 · GitHub. [En línea]. 2000. [Accedido 14 marzo 2015]. Disponible en: https://github.com/IdentityServer/IdentityServer3

THOMPSON, James Walter. JSON Web Tokens - jwt.io. [En línea]. 2000. [Accedido 25 marzo 2015]. Disponible en: http://jwt.io/

TORRES, Gisela. RedIRIS - Autenticación de usuarios. [En línea]. 2015. [Accedido 10 marzo 2015]. Disponible en: http://www.rediris.es/cert/doc/unixsec/node14.html

TRAMULLAS, Jesús. Digital Farmadrid / 39 / Bélgica ensaya un nuevo sistema de remuneración para las farmacias. [En línea]. Junio 2010. [Accedido 27 marzo 2015]. Disponible en: http://farmadrid.cofm.es/es/index.asp?MP=28&MS=126&MN=2

TUÑÓN, Sandor Ernesto. Doble autenticación unificada - Conocimientos informáticos - kilbot.net. [En línea]. 2010. [Accedido 26 marzo 2015]. Disponible en: http://www.kilbot.net/stag/doble-autenticacion-unificada/

VASILJEVSKI, Nikola y RIS, Frederic. RIS - Sistema de información Radiológica. [En línea]. 2011. [Accedido 10 marzo 2015]. Disponible en: http://openhealth.com.co/es/ris-sistema-de-informacion-radiologica

VÁZQUES, Miguel R. Centro Nacional de Información de Ciencias Médicas. [En línea]. 1999. Disponible en: http://www.sld.cu/

GLOSARIO DE TÉRMINOS

App: Abreviatura de aplicación.

App cliente: Aplicaciones que componen la solución PACS-RIS.

ASP.Net: Es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

Caso de Uso: secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

Componente: forma parte de la composición de un todo. Se trata de elementos que, a través de algún tipo de asociación o contigüidad, dan lugar a un conjunto uniforme.

CU: Abreviatura de caso de uso.

Estándar: Conjunto de protocolos que sirve como tipo, modelo, norma, patrón o referencia.

Campo: Cuadro de texto en el cual el usuario puede introducir datos, en la programación web se denomina input.

PACS: Sistemas de Almacenamiento y Transmisión de Imágenes Digitales.

PACS-RIS: Integración de los sistemas PACS y RIS, solución desarrollada en la UCI por el CESIM.

Protocolo: Conjunto de reglas que gobiernan la comunicación entre diferentes entidades.

Requisito: Una condición o capacidad necesitada por el usuario para resolver un problema o lograr un objetivo.

RIS: Sistema de información Radiológica.

Rol: término que agrupa un conjunto de permisos; son asociados a los usuarios.

Servidor: Computadora que forma parte de una red, brindando servicios a otras computadoras que reciben el nombre de clientes.

Token de acceso: Cadena que denota un ámbito específico, tiempo de vida del acceso y otros atributos de acceso.

Usuario: persona o sistema PACS-RIS que interactúa con la aplicación.