Universidad de las Ciencias Informáticas

Facultad 7

Trabajo de Diploma para optar por el Título de

Ingeniero en Ciencias Informáticas

Aplicación web para el control de las licencias de software de los productos de la solución alas PACS-RIS

Autores: Arami Pelaez Cervera

Raidel Licor Valcarcel

Tutores: Ing. Yoel Rivera Suárez

Ing. Jesmar E. Fajardo Martín

La Habana, junio de 2013

"Año 55 de la Revolución"

DATOS DE CONTACTO:

TUTORES:

Ing. Yoel Rivera Suárez: Ingeniero en Ciencias Informáticas. Graduado de la UCI en el año 2008. Posee la categoría docente de Profesor Asistente. Ha impartido las asignaturas de Programación 5, Física I, II en la Facultad 7. Se desempeña como Arquitecto en el proyecto en el Departamento de Producción de Software Médico Imagenológico, CESIM. Correo electrónico: yrsuarez@uci.cu.

Ing. Jesmar E. Fajardo Martín: Graduado de Ingeniero en Ciencias Informáticas, egresado de la UCI en el año 2008. Ha impartido las asignaturas de Programación II, Programación III, Inteligencia Artificial y Práctica Profesional. Es profesor Instructor de la Facultad 7 y se desempeña actualmente como programador en el Departamento de Producción de Software Médico Imagenológico, CESIM. Correo electrónico: jefajardo@uci.cu.

DEDICATORIA

De Raidel

A mis padres por ser los mejores del mundo.

A mi madre por ser ejemplo, guía y mi apoyo en todo momento, por ser única y especial.

A mi padre porque es la razón por la que cada día me supero, es mi inspiración.

A mi hermano por cuidar de mí desde pequeño.

A mis abuelos José Antonio y Zenaida porque en ellos encontré el mayor afecto del mundo y los llevo siempre en mi corazón.

A mis abuelos Agustín y Juana por ser ejemplos de respeto y dedicación.

A mis tíos Orlando, Puchi y Onel por ser únicos e incomparables, por confiar siempre en mí.

De Arami:

A mi abuelo Carlos Rafael, por estar ahora y siempre en mi corazón.

A mi madre, por su amor, dedicación y empeño en ayudar a cumplir mis sueños.

A mi padre, por ser mi apoyo en estos largos años.

A mi hermano, para que luche por construir su propio camino.

A mi abuela Juana, por ser única e incondicional.

A mi prima Leiny, por ser mi guía en los momentos difíciles.

A mi familia, por ser la mejor familia del mundo.

A mis amigos, por brindar su amistad sin pedir nada a cambio.

AGRADECIMIENTOS

De Arami y Raidel

A la Revolución y a Fidel por crear esta universidad de excelencia.

A los profesores de la Facultad 7 por habernos formado y guiado durante el camino.

A Yamilet y Eilen por su amistad y apoyo incondicional, por estar siempre presentes cuando las necesitamos, toda la vida estaremos en deuda con ustedes, mil gracias este éxito también es suyo.

A nuestros tutores Yoel y Jesmar por su apoyo en el desarrollo de la tesis.

A nuestros amigos por haber compartido con nosotros estos 5 años.

De Raidel

***Doy** gracias a la vida por permitirme alcanzar esta meta y cumplir mis sueños.*

***A** mis padres porque siempre han sido y serán los mejores del mundo, mi padre ejemplo de honestidad y es por el que cada día me levanto con ganas de alcanzar nuevas metas y superarme, pues ha sido el más exigente en cuanto a mi vida profesional y siempre confiando en que puedo dar más. Mi madre una mujer llena de alegría y una luchadora incansable por los sueños de sus hijos y los de ella propio, mi mano derecha. Los dos son mi universo, conforman una estrella que es la que me guía cada día.*

***A** mi hermano porque es mi amigo y es la persona en quien confío plenamente. Pues sé que hoy está más orgulloso que nadie de tenerme como hermano y porque en estos años pues ha asumido el papel de guardián.*

***A** mi amiga Yudith porque es alguien especial en mi vida.*

***A** mi amiga Doramis porque ha sido mi apoyo y mi hermana en estos años difíciles, con la que he compartido buenos y malos momentos y la que siempre estuvo ahí para lo que necesitara.*

***A** Dunieska porque tenemos una excelente amistad.*

***A** Fernando por ser un buen amigo y un excelente profesor que siempre me ayudó cada vez que lo necesité.*

***A** las personas hermosas que conocí en esta universidad y que marcaron mi vida para siempre, a las que nunca olvidaré pues me aportaron muchísima experiencia.*

***A** mi compañero de tesis pues me enseñó a trabajar en equipo y con el que compartí todo un año de sacrificio y mucho trabajo, mil gracias amigo pues formamos juntos un gran equipo.*

AGRADECIMIENTOS

A *mis dos hermanos de esta universidad, Lorenzo y Tony, amigos que nunca olvidaré a pesar de la distancia que nos pueda separar.*

A *mis amigos Yuney y Alex pues nuestra amistad va más allá, hemos sido más que amigos, hermanos.*

A *Mary una amiga que nunca olvidaré.*

Al *piquete de la FEU, Yasmani, Alexei, Eilen, Yamilet, Alturo,Yanela, Arletis con los cuales aprendí mucho.*

A *mis amistades, Laura, Jorge, Manuel, Roldos, Diego, Randy, Yoandy, Elizandra, Yani, Reinier, Ariel, Pedro, Adolfo, Danilo,Yinay*

A *los profesores que nos ayudaron en esta tarea, Yania, Luis Eduardo, Bonet, Leodan, Liudmila y al tribunal por sus críticas constructivas.*

A *Beatriz que también me ayudó mucho en estos últimos años y por la gran amistad que tenemos.*

De Arami:

A *mi abuelo Carlos Rafael, por estar ahora y siempre en mi corazón, este triunfo es dedicado a ti.*

A *mi mamá que sin su amor, cariño y comprensión jamás hubiera sido capaz de llegar tan lejos para cumplir mis sueños, quisiera que algún día estuvieras tan orgullosa de mí como yo lo estoy que seas mi mamá.*

A *mi papá que se ha esforzado mucho para que sus hijos estudien sin preocupaciones y por mostrarnos la fuerza para enfrentarse a la vida.*

A *mi hermano lo reto a que persiga y alcance sus sueños al igual que yo he alcanzado los míos, sin duda puedes llegar tan lejos como te lo propongas.*

A *mi abuela Juana, por preocuparse por sus nietos más que por ella misma y ser ejemplo de abnegación y sacrificio.*

A *mi primita Leiny y a Ernesto al que considero ya mi primo, gracias por su ejemplo y consejos todos estos años sin duda ustedes son unos de los pilares que hicieron posible que llegara tan lejos.*

A mis tíos y primos, por el apoyo y la confianza que tuvieron en mí.

A *mi otro hermano Dany, tu perseverancia para alcanzar tus metas me dio fuerzas para lograr las mías.*

A *mis amigos, Rembe, Celio, Ivet, Ada, las Katys, Carlos, Henry, gracias por compartir su amistad estos años de sacrificio sin pedir nada a cambio.*

AGRADECIMIENTOS

A *todo el grupo 7502, en especial al piquete de los letales del apartamento Ernesto, Pedro, Adrián, Ariel, Jorge, Dory, Reynier y Damián, gracias por lo pequeños y grandes momentos que vivimos juntos, muchos éxitos a todos en su vida profesional y personal.*

A *mi eterna princesita, gracias por los momentos maravillosos que vivimos juntos, los llevare por siempre en mi corazón.*

RESUMEN

El desarrollo y comercialización de software se ha ido incrementando con el avance progresivo de las tecnologías. En la actualidad existe una tendencia a utilizar, distribuir y copiar programas sin la debida autorización de sus creadores, contribuyendo a la piratería informática. Muchas empresas enfocan su trabajo al desarrollo de aplicaciones para proteger sus propios productos y los de las demás empresas, garantizando mediante el uso de licencias de software que no se violen los derechos de propiedad intelectual.

La Universidad de las Ciencias Informáticas (UCI) cuenta con el Sistema de Almacenamiento, Transmisión y Visualización de Imágenes Médicas (alas PACS) y el Sistema de Información Radiológica (alas RIS), los cuales constituyen la solución alas PACS-RIS, esta representa uno de los productos más comercializados de dicha institución y se encuentra desplegada en varios hospitales cubanos y venezolanos. Su protección contra la piratería se ha convertido en un pilar fundamental, pues como constituye una fuente importante de ingresos al país se corre el riesgo de que pueda ser objeto de un acto de este tipo.

La presente investigación propone una aplicación web para el control de las licencias de software de los productos de la solución alas PACS-RIS. El desarrollo de esta aplicación fue guiada por las fases definidas en la UCI para el ciclo de vida de los proyectos del programa de mejoras. Utilizando el Lenguaje de Modelado Unificado (UML por sus siglas en inglés) en su versión 2.1 como lenguaje de modelado, como Entorno Integrado de Desarrollo (IDE por sus siglas en inglés) Visual Studio Express 2012 con la tecnología de desarrollo ASP.NET MVC 4 y C# como lenguaje de programación. La herramienta de modelado usada fue Enterprise Architect 7.5 y como Sistema de Gestión de Base de Datos se utiliza PostgreSQL en su versión 9.1.1.

La aplicación desarrollada permite generar una o varias licencias de software a los usuarios autorizados y además almacena las licencias de software emitidas para llevar el control de las mismas.

Palabras clave:

Licencia, software propietario, piratería, derecho de autor, propiedad intelectual.

ÍNDICE

ÍNDICE

ÍNDICE

INTRODUCCIÓN

El desarrollo de software está fuertemente sujeto al avance económico de cada nación, como parte del proceso de revolución de las Tecnologías de la Información y las Comunicaciones existente en todo el mundo. Es por ello que en la actualidad, como ocurre con otros productos de la creatividad humana, el software es derecho exclusivo de la compañía fabricante o autor y está protegido por los derechos de propiedad intelectual (PI). La PI está relacionada con las invenciones, las obras literarias y artísticas, los símbolos, los nombres, las imágenes, los dibujos y modelos utilizados en el comercio. Se divide en dos categorías: la propiedad industrial que incluye las invenciones, patentes, marcas, dibujos y modelos industriales e indicaciones geográficas de origen; y el derecho de autor que abarca las obras literarias y artísticas. (1) Los derechos de PI en el software se adquieren en la forma de patentes, derechos de autor, marcas registradas, secretos industriales o licencias. (2)

La licencia de software es el contrato entre el desarrollador de un producto sometido a PI y el usuario final, en el cual se define con precisión los derechos y deberes de ambas partes. La forma en que se distribuye el software es elegida por el desarrollador o aquel a quien este haya cedido los derechos de explotación. También se especifican todas las normas y cláusulas que rigen su uso, principalmente se estipulan, los alcances de uso, instalación, reproducción y copia del producto. (3)

La tendencia de la mayoría de las empresas que se dedican a la producción y comercialización de software en el mundo, es utilizar licencias comerciales, que sólo permiten ejecutar el programa a las personas que han comprado la licencia. Prohíben la modificación o adaptación del mismo y para evitarlo suelen distribuirlo sin el código fuente. (4)

En Cuba existen varias empresas y grupos que se dedican al desarrollo de aplicaciones informáticas con fines comerciales, dentro de las cuales se destaca la Universidad de las Ciencias Informáticas. La solución alas PACS-RIS constituye uno de los productos más comercializados del Centro de Informática Médica (CESIM) de dicha institución, la cual posibilita establecer un flujo de trabajo más organizado en los departamentos de diagnósticos por imágenes, así como el mantenimiento de las imágenes y de la información clínica de los pacientes.

El sistema alas PACS está diseñado para ofrecer al personal médico que labora en los departamentos de diagnóstico por imágenes, una gama de herramientas de propósito general, para la visualización,

procesamiento y almacenamiento de imágenes médicas. Facilita el acceso a las imágenes desde cualquier punto de la institución de salud, el intercambio de imágenes entre unidades médicas y la creación de las listas de trabajo para los equipos de adquisición de imágenes médicas compatibles. Está formado por cinco sistemas altamente integrados: la estación de diagnóstico general (alas PACSViewer), el servidor de imágenes médicas (alas PACSServer), el sistema para la transmisión de imágenes médicas (alas PACSDMail), el servidor de listas de trabajo (alas PACSWorklist) y la herramienta de edición de informes imagenológicos (alas PACSReport). (5)

El sistema alas RIS permite el registro de pacientes y sus citas para estudios o consultas de imagenología, el registro de los datos de los especialistas y los equipos médicos. Proporciona la personalización mediante perfiles de usuario, es altamente configurable y es adaptable a las condiciones particulares de las instituciones hospitalarias. 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 que estos actualicen sus listas de trabajo o las de los especialistas. Posibilita realizar búsquedas por pacientes, estudios y diagnósticos médicos, facilitando la realización de estudios de morbilidad. (5)

En el marco de los proyectos de informatización de los hospitales del Ministerio del Poder Popular para la Salud (MPPS) y Petróleos de Venezuela S.A. (PDVSA) de Venezuela y de conjunto con el Ministerio de Salud Pública de Cuba (MINSAP), se incrementan las instituciones que solicitan la instalación de la solución alas PACS-RIS. También existen otros países que tienen intenciones de negocios con ALBET S.A. empresa encargada de comercializar los productos de la UCI, entre los que se pueden mencionar, México, Costa Rica y Angola.

Para la negociación con diferentes países que solicitan la instalación de la solución se tienen en cuenta varios estilos comerciales, uno de ellos es la venta de la solución donde se incluyen servicios profesionales, y otro es la venta de licencias de software. Para proteger esta solución en el momento de su comercialización se le incorporó al visor de imágenes médicas, al servidor de imágenes y al reportador, un mecanismo de validación de licencias de software, mientras el sistema alas RIS es protegido localmente en los servidores de la institución donde es instalada la solución. La protección a dicho producto es porque constituye una fuente importante de ingresos de divisas a la economía nacional y hasta la fecha se han firmado contratos por un monto de 32 millones de dólares.

INTRODUCCIÓN

Dentro del proceso de validación de licencias de software se utiliza la herramienta SerialShield Manager, desarrollada por la empresa francesa Ionworx Technology. Este software se especializa en la protección de copia y gestión de licencias para los desarrolladores de software profesional y fue adquirido por la UCI para el licenciamiento de algunas de sus soluciones comerciales. (6)

En el proceso de licenciamiento del visor de imágenes médicas se deben recoger un conjunto de datos de hardware de las estaciones de trabajo y servidores donde es instalada esta aplicación. El almacenamiento de estos datos se realiza de forma manual y en archivos Word y Excel, provocando que durante la fase soporte, sea engorroso el control de las licencias de software emitidas, así como la generación de nuevas licencias. Actualmente no se cuenta con información actualizada de la cantidad, características y detalles de las estaciones de trabajo que han sido registradas en cada institución hospitalaria u otras organizaciones.

Para generar las licencias se introducen manualmente en la herramienta de generación de licencias SerialShield Manager dos datos fundamentales, el Machine ID y el nombre de la Entidad de la Aplicación. El Machine ID constituye un id único para cada computadora y es obtenido a partir de la combinación del id del CPU, información del BIOS, la placa base y la serie del fabricante del disco duro. El nombre de la Entidad de la Aplicación se genera a partir de una nomenclatura definida por el proyecto que desarrolla cada producto. Estos datos proporcionan una licencia única para cada estación de trabajo, siendo necesario generar una licencia para cada computadora donde fue instalado el sistema, manteniendo la probabilidad de cometer errores durante la generación de la licencia. Luego se registra el código de la licencia en un documento que finalmente se envía al hospital y se almacena. Tampoco se controla si la institución ha instalado el producto en más computadoras de las que se estipuló en el contrato.

Cuando se reinstala una computadora que utilice la solución, el cliente solicita el reenvío de licencia, para esto es necesario adquirir una vez más la información de la estación, compararla con la que se encuentra registrada en los documentos, si aún se conservan, posteriormente enviar la licencia registrada o generarla nuevamente si ha cambiado algún elemento de hardware de esta estación. Por otra parte, los responsables de la generación de las licencias de software, varían en dependencia del proyecto de despliegue que se ejecute, y al menos en Venezuela, más de cuatro personas han tenido acceso físico a la aplicación que genera las licencias, por tanto, se corre el riesgo de que la aplicación pueda ser malversada o duplicada.

INTRODUCCIÓN

Por lo anteriormente expresado se identifica como **problema a resolver**: ¿Cómo facilitar la gestión de licencias de software para los productos de la solución alas PACS-RIS?

Este problema se enmarca en el **objeto de estudio:** El mecanismo de licenciamiento de software de las aplicaciones de salud, siendo el **campo de acción:** El proceso de generación de licencias de software de los productos de la solución alas PACS-RIS.

Para la solución del problema se plantea como **objetivo general**: Desarrollar una aplicación web que permita la generación masiva y almacenamiento de las licencias de software de los productos que forman parte de la solución alas PACS-RIS.

Para dar cumplimiento al objetivo planteado se proponen las siguientes **tareas de la investigación**:

1. Realizar un análisis crítico y valorativo de los sistemas informáticos de control de licencias de software existentes a nivel nacional e internacional, estableciendo similitudes con la investigación en curso.
2. Describir las características de los sistemas alas PACS, alas RIS y SerialShield Manager con el objetivo de explicar el procedimiento de licenciamiento de software.
3. Analizar el proceso de negocio asociado a la generación y control de licencias de software, logrando un modelo único como guía para la implementación de la aplicación.
4. Generar los artefactos correspondientes a las fases definidas en la UCI para el ciclo de vida de los proyectos del programa de mejoras.
5. Realizar la implementación de la aplicación aplicando las pautas de diseño y lo establecido en la especificación de requisitos de software.
6. Realizar las pruebas a la aplicación implementada.

El documento presentado se encuentra estructurado en cuatro capítulos:

Capítulo 1: Contiene un estudio de los principales conceptos relacionados con la investigación, así como del estado del arte de los sistemas para el control y generación de licencias de software existentes en Cuba y en el mundo. Se describen las herramientas, tecnologías y lenguajes de programación que se usarán en la solución.

Capítulo 2: Contiene la propuesta de la aplicación a desarrollar, la selección y descripción de los requisitos funcionales y no funcionales del sistema. La descripción del modelo de procesos de negocio asociado a la investigación y la descripción de casos de usos del sistema.

Capítulo 3: Contiene la descripción de la arquitectura y su fundamentación. Como parte de la solución se modelan los diagramas de clases del diseño, así como los diagramas de secuencia correspondientes. Se muestra el modelo de datos del sistema y se describen los patrones de diseño.

Capítulo 4: Se muestran los conceptos de los métodos existentes para realizar la evaluación del sistema en la etapa de pruebas, describiéndose el empleado y se muestran los casos de prueba. Se presenta el diagrama de componentes de la aplicación así como el diagrama de despliegue de la misma y se describe el estándar de codificación empleado en la implementación de la solución.

CAPÍTULO 1. **FUNDAMENTACIÓN TEÓRICA**

En este capítulo se explican los conceptos relacionados con la investigación para un mejor entendimiento de la misma. Además se describen los sistemas para la generación de licencias de software existentes en el mundo. Se incluye un estudio sobre las metodologías, tecnologías, herramientas, lenguajes y notaciones que se emplearán en el desarrollo de la aplicación.

1.1. Derechos de autor

El derecho de autor es una forma de protección que concede al creador de una obra original el derecho exclusivo para copiarla, distribuirla, venderla y modificarla, excepto en circunstancias especiales descritas por las leyes de derechos de autor.

Casi todo el software exhibe un aviso de derechos de autor. Este aviso no es requerido por la ley, es decir, incluso los programas sin el aviso están protegidos por la ley de derechos de autor. Las personas que eluden la ley de derechos de autor y copian, distribuyen o modifican software de manera ilegal son conocidas como piratas, y sus copias ilegales se llaman software pirata. (7)

1.2. Licencias de software

El software está protegido por los términos de una licencia además de la protección de los derechos de autor. Una licencia de software, o un acuerdo de licencia, es un contrato legal que define el modo en el cual se puede utilizar un programa de computadora. Las licencias de software imponen restricciones adicionales en el uso del software u ofrecen derechos adicionales a los consumidores.

Las licencias de software suelen ser extensas y contener mucho lenguaje legal, pero su derecho para utilizar el software se mantiene siempre y cuando se respete los términos de la licencia. Por lo tanto, debe comprender la licencia para cualquier software que utilice. (8)

1.3. Categorías de software

Desde una perspectiva legal, existen dos categorías de software: de dominio público y patentado. El software de dominio público no está protegido por una reserva de derechos de autor porque han expirado o el autor ha puesto el programa en el dominio público para que todos lo usen sin restricciones. También se

puede copiar, distribuir e incluso revender sin problemas. La restricción principal con el software de dominio público es que usted no puede reservar los derechos de autor sobre él.

El software patentado tiene restricciones en su uso señaladas por la reserva de los derechos de autor, las patentes o los acuerdos de licencia. Cierto software patentado se distribuye de manera comercial, mientras que otro es gratuito. Con base en los derechos de las licencias, el software patentado se distribuye como software comercial, demoware, shareware, freeware y de código abierto. (9)

1.3.1. Software comercial

El software comercial se vende en las tiendas de computadoras o en los sitios web, también puede adquirirse mediante un contrato con la empresa propietaria del mismo. Si se compra este tipo de software, en realidad solo se adquiere el derecho a utilizarlo bajo los términos de la licencia. Una licencia para software comercial suele apegarse a las limitaciones mencionadas por las leyes de derechos de autor, aunque puede darle permiso para instalar el software en el trabajo y en el hogar siempre y cuando utilice una a la vez.

La solución alas PACS-RIS es considerada un software comercial, pues se distribuye mediante el uso de licencias de software. Estas licencias solo permiten la ejecución de la solución en las computadoras de las instituciones para las cuales fueron generadas, lo que impide que el producto sea comercializado a terceros y garantiza que esté protegido contra la piratería.

1.3.2. Software demoware

El software comercial está disponible en versiones de prueba, que suelen denominarse demoware. El demoware se distribuye en forma gratuita y viene preinstalado en las computadoras nuevas, pero está limitado de algún modo hasta que se adquiere. Funciona una cantidad precisa de días antes de expirar y solicitar su pago. Puede funcionar un tiempo limitado y se puede configurar para que funcione solo una cantidad limitada de veces. También pueden inhabilitarles funciones importantes, como imprimir, lo cual sirve para dar al demoware el poco halagador nombre de mutilado-ware.

El sistema alas PACSViewer constituye un software demoware, pues cuenta con un período de 30 días donde pueden utilizarse todas sus funcionalidades. Luego de este período de prueba se debe registrar el producto mediante una licencia de software proporcionada por los desarrolladores del software.

1.3.3. Software shareware

Las características del shareware se asemejan a las del demoware. El shareware es un software con una reserva de derecho de autor que se comercializa bajo una política de pruébelo antes de usar. Suele incluir una licencia que permite emplear el software durante un período de prueba. Para utilizarlo después del período de prueba, se debe pagar una tarifa de registro. La idea detrás del shareware es que el pago sería bajo un sistema de honor. A diferencia del demoware el shareware es un software con todas las funciones.

En la actualidad, muchos autores de software shareware emplean técnicas de demoware para limitar sus programas hasta recibir un pago. El término shareware se refiere a los programas distribuidos por programadores independientes, mientras que demoware se emplea para referirse a las versiones de prueba del software de empresas grandes, como Microsoft, Adobe Systems y Oracle. (9)

1.3.4. Software freeware

El freeware es un software con una reserva de derechos de autor que se ofrece de manera gratuita. Tiene funciones completas y no requiere un pago para utilizarlo. Debido a que este software está protegido por derechos de autor, solo se puede hacer lo que autoriza de manera explícita la ley de derechos de autor o el autor. Una licencia de freeware le permite usar el software, copiarlo y distribuirlo, pero no autoriza a alterarlo o venderlo. Muchas utilerías, controladores de dispositivos y algunos juegos se distribuyen como freeware. (10)

1.3.5. Software con código fuente abierto

El software con código fuente abierto proporciona a los programadores las instrucciones no compiladas de un programa, para que las modifiquen y mejoren. Este software puede venderse o distribuirse gratuitamente en forma compilada pero, en todos los casos, también debe incluir el código fuente. Linux es un ejemplo del software con código abierto, igual que FreeBSD: una versión de UNIX diseñada para computadoras personales. A pesar de la falta de restricciones en la distribución y el uso, el software con código fuente abierto tiene derechos de autor y no es de dominio público.

Muchas características del código fuente abierto se aplican al software gratuito, el cual no debe confundirse con el freeware, que no se puede vender ni modificar. Estos tipos de software pueden copiarse una cantidad ilimitada de veces, distribuirse de manera gratuita, venderse y modificarse. (10)

1.4. Sistemas para el control y generación de licencias

En los últimos años el desarrollo de la industria del software ha tenido un incremento considerable, los programas que muchas empresas desarrollan y luego comercializan se ven cada día más afectados por la piratería, la copia y la distribución ilegal. Es por esto que varias compañías han enfocado su atención en este tema y desarrollan aplicaciones para dar protección al software que otros crean, mediante la generación y control de licencias de software.

En la bibliografía consultada no se encontró referencia de sistemas nacionales que generen licencias de software y mantenga un control sobre las licencias emitidas, mientras que a nivel internacional se encuentran varios productos de este tipo. La UCI adquirió la herramienta SerialShield Manager, esta constituye una aplicación propietaria al igual que las que se describen a continuación, pero a diferencia de las demás contiene un módulo CGI con el cual se pueden generar licencias de software y puede ser utilizado con dicho propósito en un servidor interno de la universidad.

SerialShield Manager

SerialShield SDK es una solución desarrollada por la empresa francesa Ionworx Technology para la protección de copia y gestión de licencias para los fabricantes de software profesional que deseen obtener el control total de las licencias de software para sus aplicaciones.

La protección, todo en uno, protege los proyectos .Net y Win32. La solución permite crear una clave de licencia para los proyectos y de esta forma ahorrar dinero. Se utiliza para agregar capacidades de evaluación para las aplicaciones y gestiona todos los aspectos para garantizar que una aplicación está dentro del período de evaluación y registrado correctamente en una máquina específica.

Se puede utilizar en todos los lenguajes .NET (C #, VB.NET, Delphi.NET) y herramientas para Win32 como Microsoft Visual Basic, VBA Access, Word VBA, VBA Excel, Microsoft Visual C + +, Borland Delphi y C + + Builder, así como otros lenguajes de programación.

Con SerialShield se puede bloquear cada licencia para una máquina específica y evitar cualquier duplicación o la instalación no autorizada. (6)

SerialShield inspecciona la información de hardware para generar un identificador único de cada computadora para esto utiliza:

- ID CPU.
- Placa base y la información del BIOS.
- Serie de fabricante del disco duro.

Esta herramienta cuenta con el módulo de interfaz de entrada común (CGI por sus siglas en inglés) SerialShield Internet Server (SSIS por sus siglas en inglés), el cual se ejecuta en un servidor de aplicaciones y mediante peticiones a través del protocolo de transferencia de hipertexto (HTTP) genera licencias de software de las aplicaciones definidas por la institución que brinda este servicio.

NetSupport DNA

NetSupport DNA es un software desarrollado por la empresa británica NetSupport Limited el cual permite realizar el seguimiento y gestión de las licencias con la función de medición de inventario y aplicaciones. Identifica software ilegal y no utilizado para saber el número óptimo de licencias que una compañía debe adquirir. NetSupport DNA proporciona una puerta de enlace de comunicaciones integrada que permite interactuar con los activos con toda seguridad, por Internet, en cualquier lugar, todo ello sin necesidad de una red privada virtual (VPN por sus siglas en inglés) ni cambios en una red existente o en la configuración de cortafuegos. Este producto requiere de una licencia para cada estación de trabajo de la red donde se instale. (11)

Sentinel RMS

Sentinel RMS es la solución de protección para software desarrollada por la compañía norteamericana Sentinel SafeNet, que permite incorporar a las aplicaciones algoritmos propietarios de alta seguridad, brindando la comodidad de distribuir las aplicaciones por todo el mundo. Utilizando un algoritmo propietario de alta seguridad, Sentinel RMS protege el software frente al uso no autorizado de forma que puede colocarse en la Web o en millones de CDs para su distribución masiva. La potente función de Sentinel RMS para crear huellas dactilares de sistema asegura que sólo los usuarios registrados pueden utilizar una aplicación al anclarla a las características únicas de la computadora en la que se encuentra instalada. (12)

Protector de Licencias (License Protector)

Licence Protector desarrollado por la empresa alemana Mirage Computer Systems, administra licencias y módulos, genera versiones de demostración y de tiempo limitado, proporciona un programa de protección

contra copia y soporta pruebas de usuario concurrentes. Ofrece archivos de licencia cifrados con claves hechas por el cliente para cada proyecto de informática y claves de activación seguras que pueden ser usadas sólo una vez. Licence Protector está disponible en 3 ediciones para satisfacer diferentes requerimientos: Principiante, Básico y Edición Profesional. (13)

ElecKey 2.0

ElecKey 2.0 es una solución desarrollada por la empresa norteamericana Sciensoft Research para la protección de copia de software, licencias de software y distribución de software. Es un software de seguridad que puede proteger las aplicaciones de software contra la piratería, manipulación y la ingeniería inversa. La capacidad de concesión de licencias puede permitirle convertir fácilmente un software en una amplia variedad de distribuciones, por ejemplo, una versión de prueba, evaluación, licencia nodo bloqueado, licencia de red flotante.

Ofrece un conjunto de características y soluciones. Muchas de las herramientas que proporciona ofrecen opciones para proteger y administrar software de licencia, y para satisfacer todas las necesidades y expectativas. (14)

Sentinel HASP

Sentinel HASP, llamada antiguamente Aladdin HASP SRM, es una solución desarrollada por la compañía norteamericana Sentinel SafeNet que permite el uso de claves de protección basadas ya sea en hardware o en software, para la protección del software y las licencias. Puede aumentar las ganancias de una empresa protegiéndola contra las pérdidas provocadas por la piratería de software y el robo de la propiedad intelectual. Permitiendo que innovadores modelos comerciales incrementen el valor y diferencien a sus productos. Proporciona una fuerte y robusta solución de seguridad de software, un respaldo líder en la industria para el licenciamiento en entornos virtuales. (15)

Licence Master

Licence Master es una aplicación desarrollada por la empresa Dotcom Software Solutions para la gestión de licencias para desarrolladores y vendedores de software que deseen proteger aplicaciones web del uso no autorizado. Con Licence Master, el desarrollador de un software podrá asegurarse de que solo las instalaciones autorizadas estén operativas. También automatiza el proceso de gestión de múltiples

aplicaciones web, con un ahorro significativo de esfuerzos administrativos, gestiona de manera flexible la política de licencias y permite al administrador intervenir manualmente cuando sea preciso. (16)

Machine ID

Machine ID SDK es una solución desarrollada por la empresa francesa Ionworx Technology basada en librerías de enlace dinámico (32 bits) que brinda al software el control de uso e impide su distribución no autorizada. Permite generar un ID para cada computadora y utiliza esta identificación para la protección de las licencias en cada computadora, restringiendo la instalación no autorizada de productos de software. La solución puede trabajar en entornos de usuarios restringidos al no necesitar tener derechos de administrador para ejecutarse. El ID obtenido es único para cada PC y es adquirido directamente desde el hardware y no desde el registro. (17)

1.5. Proceso de licenciamiento del sistema alas PACSViewer

Después de instalado el sistema alas PACS en las estaciones de visualización de los especialistas médicos, se podrá acceder a sus funcionalidades por un periodo de 30 días. Una vez terminado este plazo será necesario licenciar el producto.

El licenciamiento del sistema puede realizarse de dos maneras si se tiene en cuenta los actores que intervienen en el proceso. Una de las opciones es la presencia directa de un equipo de despliegue y soporte proveniente del CESIM. La otra variante comprende a los especialistas informáticos que trabajan en la institución sanitaria donde se despliega el software, los cuales realizarían una solicitud de licencia o clave serial que permita registrar el sistema, brindando en este último caso lo que se conoce como un soporte de primer nivel.

Una vez instalado el sistema alas PACS se procede al licenciamiento del software con el objetivo de habilitar sus funcionalidades permanentemente. En la siguiente secuencia de pasos se procede a realizar la explicación del procedimiento de licenciamiento del sistema alas PACS: (18)

1. Primeramente se procede a ejecutar la herramienta (alas PACSSerial.exe) para licenciar el sistema alas PACS.

2. Una vez ejecutada esta acción aparecerá una ventana donde se muestra un Serial ID que identifica a la estación de trabajo que se desea licenciar. Debe copiarse este valor para posteriormente realizar la solicitud de clave serial y licenciar el sistema.
3. El serial ID obtenido anteriormente es registrado en el documento (Registro de licencias de software). Es importante para el tema en cuestión que también quede registrado el AE del sistema (nombre que identifica a la estación en la red imagenológica) asociado al Serial ID. Una vez terminado este informe debe enviarse vía correo a los especialistas que dan soporte al sistema alas PACS para realizar la solicitud de clave serial.
4. Una vez recibido el documento (Registro de licencias de software), el equipo de soporte del sistema alas PACS procede a generar las claves serial de cada estación registrada en el documento dejando plasmado este valor en la sección correspondiente. Una vez terminado este trabajo, se le envía por correo al informático de la institución sanitaria que realizó la solicitud.
5. Cuando es recibida la información del equipo de soporte del sistema alas PACS, el informático responsable de licenciar la aplicación debe ejecutar la herramienta alas PACSSerial.exe nuevamente. En este caso se seleccionara en Tipo de Licencia: Modo Completo y en el apartado usuario colocar el AE de esa estación de trabajo, acto seguido en Clave Serial introducir el valor recibido. Después debe presionarse el botón Chequear Serial, entonces aparecerá un mensaje confirmando la finalización del proceso.
6. Una vez finalizada esta acción, se puede ejecutar el sistema alas PACS sin problemas, la aplicación ya está registrada. Si volvemos a ejecutar la herramienta para licenciar el sistema nos dice que la versión del sistema está registrada.

Mediante la utilización de la herramienta SerialShield Manager y haciendo uso del procedimiento descrito anteriormente para el licenciamiento del visor de imágenes médicas, no se logra resolver la problemática planteada. Es por esto que se propone el desarrollo de una aplicación web que facilite el proceso de generación de licencias para los productos de la solución alas PASC-RIS y que lleve un control de las licencias emitidas.

1.6. Descripción de tecnologías, patrones, lenguajes de programación y herramientas

En la actualidad el uso de tecnologías de última generación es un requisito casi indispensable para garantizar la calidad de un sistema. Se impone la utilización de tecnologías que brinden rapidez y eficacia,

que ofrezcan a los desarrolladores herramientas poderosas que permita realizar un conjunto de procedimientos ya sean básicos o altamente avanzados.

1.6.1. Plataforma de desarrollo .NET Framework 4.5

NET Framework 4.5 es un componente integral de Windows que admite la compilación y la ejecución de la siguiente generación de aplicaciones y servicios web. Los componentes clave de .NET Framework son Common Language Runtime (CLR) y la biblioteca de clases .NET Framework, incluye ADO.NET, ASP.NET, formularios Windows Forms, Windows Presentation Foundation (WPF) y Windows Workflow Foundation (WF). Proporciona un entorno de ejecución administrado y una integración con una gran variedad de lenguajes de programación.

La plataforma .NET Framework 4.5 proporciona varias características y mejoras para la informática en paralelo. Incluyen rendimiento mejorado, mayor control, mejor compatibilidad con la programación asincrónica, una nueva biblioteca de flujo de datos, y mejor compatibilidad para la depuración y el análisis de rendimiento en paralelo. Además incluye nuevas características para el desarrollo de aplicaciones ASP.NET entre las que se destaca mencionar:

- Compatibilidad con los nuevos tipos de formulario HTML5.
- Compatibilidad con JavaScript discreto en scripts de validación en el cliente.
- Control mejorado del script de cliente mediante agrupación y minificación para un rendimiento de la página mejorado.
- Compatibilidad con el protocolo de WebSockets.
- Compatibilidad con la lectura y escritura de solicitudes y respuestas HTTP de forma asincrónica.
- Compatibilidad con módulos y controladores asincrónicos. (19)

Por las características y ventajas antes descritas se utilizará esta plataforma para el desarrollo de la aplicación.

1.6.2. Lenguaje de programación C# 4.0

C# es un lenguaje orientado a objetos y con seguridad de tipos que permite a los desarrolladores compilar diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Se puede utilizar C# para

crear aplicaciones cliente de Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y mucho más.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. (20) Por todas estas características y potencialidades se decide utilizar este lenguaje para el desarrollo de la aplicación.

1.6.1. Lenguaje de programación HTML 5

HTML5 es la versión más nueva del Lenguaje de Marcado de Hipertexto. Esta nueva versión cambia los paradigmas de desarrollo y diseño web existentes, al introducir herramientas notables que permiten la publicación de archivos de audio y video con soportes de distintos codecs; cambios en los llenados de formularios y una web semántica mejor aprovechada. Agrupa las especificaciones relacionadas al desarrollo de páginas web: HTML 4, XHTML[1] 1, DOM[2] nivel 2 e integra algunos elementos de CSS[3] nivel 2. (21) Por sus potencialidades para la representación de la interfaz de usuario en el cliente se decide utilizar este lenguaje.

1.6.2. Lenguaje de programación JavaScript

JavaScript es un lenguaje interpretado que permite incluir macros en páginas Web. Estas macros se ejecutan en el ordenador del visitante de las páginas web y no en el servidor. Este lenguaje soporta cuatro tipos de datos, no es necesario declarar el tipo de las variables, argumentos de funciones ni valores de retorno de las funciones. El tipo de las variables cambia implícitamente cuando es necesario, lo que dificulta el desarrollo de programas complejos, pero ayuda a programar con rapidez macros sencillas. (22)

---

[1] Extensible Hypertext Markup Language

[2] Document Object Model

[3] Cascading Style Sheets

1.6.3. Visual Studio Express 2012

Visual Studio 2012 es una de las versiones más importantes hasta el momento de este IDE. Viene especialmente diseñado para prosperar en un entorno en el que las ideas escasean y la velocidad es fundamental. Toda la interfaz ha sido rediseñada para simplificar el flujo de trabajo y brindar fácil acceso a las herramientas más usadas. Las barras de herramientas están simplificadas, se redujo el desorden de pestañas y ahora tiene formas nuevas y rápidas de encontrar códigos. Es de carácter gratuito y es proporcionado por la compañía Microsoft Corporation orientándose a principiantes, estudiantes y aficionados de la programación web y de aplicaciones.

Visual Studio 2012 ofrece plantillas, diseñadores y herramientas de evaluación y depuración. Blend para Visual Studio proporciona un conjunto de herramientas visuales para aprovechar al máximo la nueva y hermosa interfaz de Windows 8. Cuando se trata de desarrollo web, este IDE ofrece nuevas plantillas, mejores herramientas de publicación y soporte integral para estándares emergentes, como HTML5 y CSS3, así como para los últimos avances en ASP.NET. Facilita la depuración con Page Inspector, mediante la interacción con la página que está codificando y sin salir del IDE. (23) Por sus características, potencialidades y las novedosas herramientas en el desarrollo de aplicaciones web se utilizará este IDE para la implementación de la aplicación.

1.6.4. Enterprise Architect 7.5

Enterprise Architect 7.5 cuenta con profundo soporte para ingeniería de software y de negocio y para desarrollo de sistemas, totalmente integrada en un singular ambiente de desarrollo. Presenta trazabilidad completa y una visión global verdadera unificando estrategias, procesos de negocios, interfaces, software, reglas, datos y sistemas muy separados. Herramientas de alta potencia, tecnologías especificas del dominio, marcos de trabajo, plataformas de integración y una interfaz consistente, escalable y robusta todas trabajan en conjunto para ayudarlo a cumplir la promesa del desarrollo dirigido por modelo. (24) Por las particularidades antes expuestas se propone utilizar Enterprise Architect 7.5 como herramienta CASE[4] de modelado.

[4] CASE: Computer Aided Software Engineering.

1.6.5. PostgreSQL 9.1.1

Entre las características más sobresalientes de este sistema gestor se encuentran su potencia y fiabilidad, la verificación de la integridad referencial, el soporte nativo para SQL y para interfaces de programación nativas de ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby, la capacidad para hacer copias de respaldo "en caliente" o la posibilidad de emplear índices, reglas y vistas. (25) La disponibilidad de replicación de datos sincrónica en 9.1.1 provee a los clientes nuevas e innovadoras formas de proteger sus datos de misión crítica, y valida a PostgreSQL como uno de los almacenes de datos de más rápido crecimiento disponibles. (26) Por su potencia,fiabilidad y las peculiaridades antes descritas se decide utilizar este gestor de base de datos para el almacenamiento de la información de la aplicación.

1.6.6. ASP.NET MVC 4

ASP.NET MVC 4 es un marco de trabajo para crear aplicaciones web escalables y basadas en estándares mediante patrones de diseño bien establecidos y la potencia de ASP.NET y .NET Framework. Es un framework de aplicaciones web que implementa el patrón modelo-vista-controlador que separa los componentes de una aplicación web, esta separación ofrece más control sobre las partes individuales de la aplicación, lo que facilita su desarrollo, modificación y prueba.

ASP.NET MVC forma parte del marco de trabajo ASP.NET. Desarrollar una aplicación ASP.NET MVC es una alternativa al desarrollo de páginas de formularios Web Forms de ASP.NET; no reemplaza el modelo de formularios Web Forms. (27) Por sus avances en el desarrollo de aplicaciones web se hará uso de esta tecnología para el desarrollo de la aplicación.

1.6.7. Interfaz de entrada común (CGI)

Un CGI es un programa que se ejecuta en el servidor por petición del navegador de un usuario. El CGI produce un resultado, el cual se envía al navegador que provocó la ejecución del programa. Los CGI otorgan dinamismo a la web. Las páginas web puras (archivos HTML) son archivos de texto y por tanto estáticos. Sin embargo, si en lugar de pedir una página web el navegador ejecuta un programa, éste puede generar la página al vuelo y decidir en el momento cómo va a ser la página.

En principio los CGI se pueden realizar con cualquier lenguaje de programación, ya que pueden ser ejecutables (archivos .exe). Sin embargo, lo más recomendable es utilizar un lenguaje de script con facilidades para realizar CGI.

Las aplicaciones CGI son una de las primeras formas de desarrollo de aplicaciones dinámicas para la web, lo que añadido a su estabilidad hace que se trate de un sistema de desarrollo bastante extendido. Sin embargo, debe tener presente también sus dificultades, entre las que destacan su lentitud frente a otros sistemas de desarrollo de aplicaciones dinámicas (como PHP o ASP), sobre todo su mayor consumo de recursos en el servidor (CPU, memoria) al ejecutar un proceso independiente por cada petición. (28) Por la necesidad de utilizar el SerialShield Internet Server para la generación de licencias de software se hará uso de esta tecnología.

1.6.8. Entity Framework 5.0

Entity Framework es un marco de trabajo para la plataforma .NET. Este permite superponer varias capas de abstracción sobre el almacén relacional, con el fin de hacer posible una programación más conceptual y reducir a una mínima expresión el desajuste de impedancias, causado por las diferencias entre los modelos de programación relacional y orientado a objetos. Incluye un nuevo proveedor de ADO.NET, llamado Entity Client, que habilita el acceso a los modelos conceptuales. (29)Por sus características y potencialidades se utilizará este marco de trabajo como herramienta de mapeo objeto-relacional para el desarrollo de la capa de acceso de datos.

1.6.9. jQuery

jQuery es un Framework de Javascript, un conjunto de funciones que ya fueron desarrolladas y probadas. Estas funciones están listas para utilizarlas de una manera muy simplificada y se puede lograr los mismos resultados en menos tiempo, sin necesidad de programar una funcionalidad completamente.

Permite agregar efectos y funcionalidades complejas a las aplicaciones web, como por ejemplo: galerías de fotos dinámicas y elegantes, validación de formularios, calendarios, entre otras. Otra ventaja es la posibilidad que brinda de trabajar con AJAX, sin preocuparse de los detalles complejos de la programación. Además cuenta con la posibilidad de agregar plugins, facilitando más el trabajo. (30) Se decide emplear este marco de trabajo para la validación de formularios y de esta forma lograr la interactividad de la aplicación.

1.7. Modelo de calidad, lenguaje y notación de modelado

1.7.1. Modelo de Madurez de Capacidad Integrado (CMMI)

CMMI pertenece a la familia de modelos desarrollados por el Instituto de Ingeniería Software (SEI por sus siglas en inglés) para evaluar las capacidades de las organizaciones de ingeniería de sistemas, ingeniería de software, además del desarrollo integrado del producto y del proceso. Es un modelo descriptivo que detalla los atributos esenciales que deberían caracterizar a una organización en un determinado nivel de maduración.

Es un modelo normativo donde las prácticas detalladas caracterizan los tipos normales de comportamiento esperables en una organización que ejecuta proyectos a gran escala. La mejora continua de los procesos se basa en muchos pasos pequeños y evolutivos en vez de innovaciones revolucionarias. CMMI proporciona un marco para organizar estos pasos evolutivos dentro de cinco niveles de maduración que sientan fundamentos sucesivos para la mejora continua del proceso. (31)

1.7.2. Proceso de mejoras

La representación continua de CMMI está estructurado por categorías de procesos: Administración de Procesos, Administración de Proyectos, Ingeniería de Software y Soporte.

La representación escalonada de CMMI alcanza cinco niveles de madurez: Inicial, Gestionado, Definido, Gestionado cuantitativamente y Optimizado. En la UCI se obtuvo en algunos centros de desarrollo de software la certificación del nivel 2, que comprende los procesos de gerencia de proyectos básicos, entre estos centros se encuentra CESIM.

En nivel 2 se dispone de unas prácticas institucionalizadas de gestión de proyectos, existen unas métricas básicas y un razonable seguimiento de la calidad. La relación con subcontratistas y clientes está gestionada sistemáticamente. Los procesos que hay que implantar para alcanzar este nivel son: Gestión de requisitos, Planificación de proyectos, Seguimiento y control de proyectos, Gestión de proveedores, Aseguramiento de la calidad y Gestión de la configuración (32)

1.7.3. Lenguaje Unificado de Modelado

El Lenguaje de Modelado Unificado es la sucesión de una serie de métodos de análisis y diseño orientados a objetos. Fusiona los conceptos de la orientación a objetos aportados por el método Booch, la Ingeniería de Software Orientada a Objetos (OOSE por sus siglas en inglés) y la Técnica de Modelado de Objetos (OMT por sus siglas en inglés). UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos.

El lenguaje de modelado es la notación gráfica que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño. La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo. (33)

1.7.4. Notación para el Modelado de Procesos de Negocio (BPMN)

El modelado de procesos, así como su nombre lo indica, tiene 2 aspectos que lo definen: el modelado y los procesos. Frecuentemente, los sistemas, conjuntos de procesos y subprocesos integrados en una organización, son difíciles de comprender, amplios, complejos y confusos; con múltiples puntos de contacto entre sí y con un buen número de áreas funcionales, departamentos y puestos implicados. Un modelo puede dar la oportunidad de organizar y documentar la información sobre un sistema.

La Notación para el Modelado de Procesos de Negocio (BPMN por sus siglas en inglés), es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo. BPMN fue inicialmente desarrollada por la organización Business Process Management Initiative (BPMI por sus siglas en inglés), y es actualmente mantenida por el Object Management Group (OMG por sus siglas en inglés), luego de la fusión de las dos organizaciones en el año 2005. Su versión actual es la 1.2 y hay una versión futura propuesta, la 2.0.

El principal objetivo de BPMN es proveer una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio (stakeholders). Entre estos interesados están los analistas de negocio, los desarrolladores técnicos y los gerentes y administradores del negocio. En síntesis

BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. (34)

Conclusiones del capítulo

En este capítulo se describieron términos importantes para guiar al lector a través de la investigación científica desarrollada. Además fueron presentados los sistemas que generan y controlan licencias de software en la actualidad como resultado de un profundo estudio del estado del arte nacional e internacional, ratificando el uso de la herramienta SerialShield Manager para la generación de licencias de software de los productos de la solución alas PACS-RIS. Finalmente fueron descritas las tecnologías, lenguajes de programación, herramientas, el lenguaje y la notación de modelado a utilizar en desarrollo de la aplicación web propuesta en la investigación.

CAPÍTULO 2. **CARACTERÍSTICAS DEL SISTEMA**

En el presente capítulo se describe la propuesta de la aplicación web a desarrollar. Se explica el proceso de licenciamiento del visor de imágenes médicas, se muestra su modelo de proceso y se incluye la descripción del flujo básico del mismo. Se listan los requisitos funcionales y no funcionales de la aplicación y se muestra una breve descripción de los casos de uso de la misma.

2.1. Propuesta de la aplicación web a desarrollar

Se propone una aplicación web que permita generar licencias de software de forma masiva para los productos de la solución alas PACS-RIS, haciendo uso del módulo CGI del software SerialShield Manager. Esto permitirá incrementar la eficiencia en la generación de las licencias y un mayor control de las licencias de software emitidas. De esta forma los responsables de la generación de licencias no tienen acceso físico a la aplicación que genera las licencias y no se corre el riesgo de que esta pueda ser duplicada y usada con otros fines.

2.2. Modelo de procesos de negocio

Un proceso de negocio es una colección de actividades diseñadas para producir una salida específica para un cliente o mercado particular. Así, un proceso es un ordenamiento específico de actividades de trabajo a través del tiempo y del espacio, con un comienzo, un fin, entradas y salidas claramente identificados: una estructura para la acción. (35)

2.2.1. Otorgar licencias de software

El proceso comienza cuando una institución solicita la instalación del sistema alas PACSViewer. Luego de instalado el sistema, se recogen datos del hardware de las estaciones de trabajo que fueron instaladas para generar las licencias de software. A partir de los datos recogidos el jefe de proyecto genera las licencias de software, posteriormente estas son almacenadas por el administrador de la institución. Finalmente se realiza la activación del sistema para su adecuado uso. En el **Anexo 1** se encuentra la descripción del proceso y la **Figura 1** muestra el flujo de actividades del mismo.

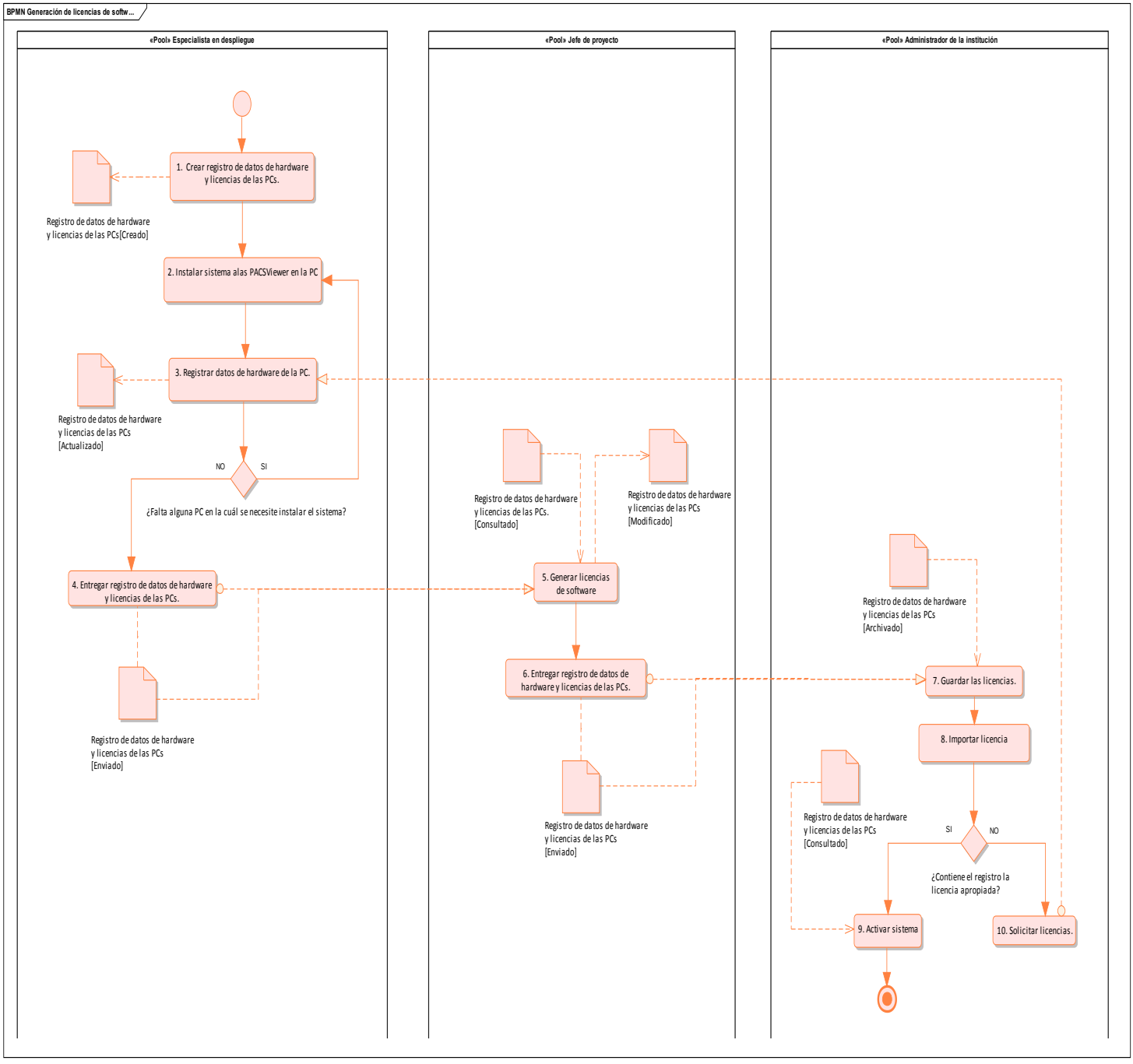


Figura 1 : Otorgar licencias de software.

2.3. Especificación de los requisitos de software

El análisis de requisitos es una de las etapas más importantes en el ciclo de desarrollo de software. La captura, análisis y especificación de requisitos es crucial, pues de esta etapa depende en gran medida el éxito de los objetivos planteados.

2.3.1. Requisitos funcionales

Los requisitos funcionales constituyen las capacidades o condiciones que el sistema debe cumplir. La siguiente tabla muestra los requisitos funcionales del sistema propuesto.

Tabla 1: Requisitos funcionales del sistema.

Nº	Funcionalidad	Descripción
RF 1	Autenticar usuario	Permite al usuario identificarse para acceder a la aplicación de acuerdo al rol que se le otorgó.
RF 2	Generar una licencia	Permite al usuario generador construir una licencia a partir de los datos suministrados.
RF 3	Generar varias licencias	Permite al usuario generador construir un conjunto de licencias a partir de una plantilla Excel, que contiene los datos necesarios para la generación de las licencias de software.
RF 4	Gestionar usuarios	Permite adicionar, eliminar, listar y buscar usuarios.
RF 4.1	Adicionar usuario local	Permite al Administrador adicionar un usuario con privilegios administrativos o de generador de licencias.
RF 4.2	Adicionar usuario global	Permite al Administrador adicionar un usuario existente en el directorio LPAD de la UCI con privilegios administrativos o de generador de licencias.
RF 4.3	Eliminar usuarios	Permite al Administrador eliminar un usuario local o global.
RF 4.4	Listar usuarios	Permite listar todos los usuarios que tienen permisos de acceso a la aplicación.
RF 4.5	Buscar usuarios	Permite buscar los usuarios registrados en el sistema, dado un usuario

		o un rol.
RF 5	Gestionar perfiles de aplicación	Permite al Administrador adicionar, eliminar, listar y buscar perfiles de aplicación.
RF 5.1	Adicionar perfil de aplicación	Permite al Administrador un perfil de aplicación.
RF 5.2	Eliminar perfil de aplicación	Permite al Administrador eliminar un perfil de aplicación.
RF 5.3	Listar perfiles de aplicación	Permite al Administrador listar todos los perfiles de aplicación existentes en la aplicación.
RF 5.4	Buscar perfiles de aplicación	Permite al Administrador buscar perfiles de aplicación dado el nombre o la llave de aplicación.
RF 6	Gestionar asignaciones	Permite al Administrador asignar a un usuario, uno o varios perfiles de aplicación, así como eliminar, listar y buscar asignaciones.
RF 6.1	Asignar usuarios a perfiles	Permite al Administrador asignar a un usuario, uno o varios perfiles de aplicación.
RF 6.2	Eliminar asignación	Permite al Administrador eliminar una asignación existente en la aplicación.
RF 6.3	Listar asignaciones	Permite al Administrador listar las asignaciones existentes en la aplicación.
RF 6.4	Buscar asignaciones	Permite al Administrador buscar una o varias asignaciones, dado un nombre de asignación, nombre de aplicación o usuario.
RF 7	Exportar licencias de software generadas	Permite al usuario generador exportar en un archivo Excel los datos y licencias de software generadas o buscadas.
RF 8	Buscar Licencias	Busca licencias de software generadas anteriormente.
RF 9	Importar información de las estaciones clientes y servidores	Importa la información necesaria de las estaciones de trabajo y servidores para generar las licencias de software.
RF 10	Configurar datos del servidor donde se aloja el módulo	Permite al Administrador realizar la configuración del servidor donde se

	SerialShield Internet Server.	aloja el módulo SerialShield Internet Server.

En la **Figura 2** se muestra el diagrama de requisitos funcionales del sistema agrupados en paquetes lógicos, lo que garantiza tenerlos unidos según las relaciones que existen entre ellos, para posteriormente conformar los casos de uso (CU) del sistema.

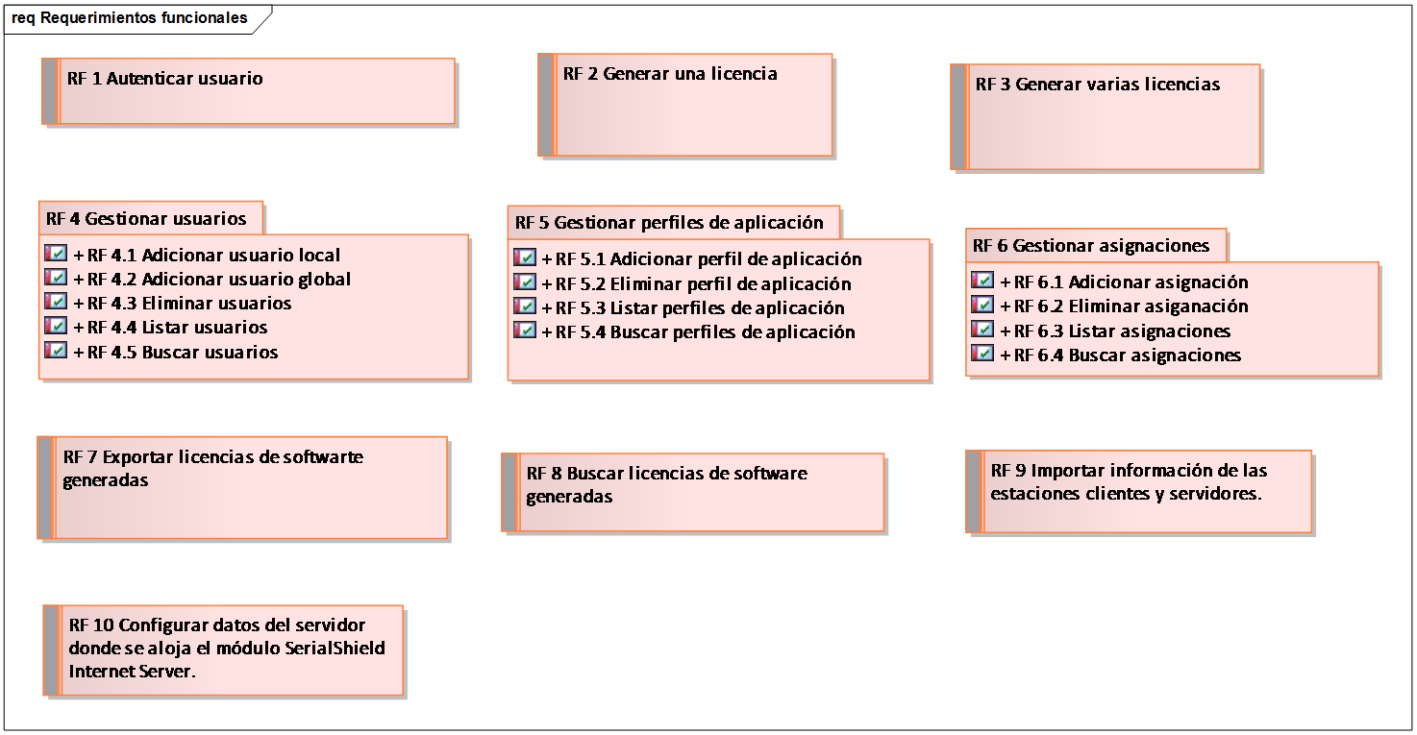


Figura 2 : Diagrama de paquetes de requisitos funcionales del sistema.

2.3.2. Requisitos no funcionales

Los requisitos no funcionales definen propiedades emergentes del sistema, tales como el tiempo de respuesta, las necesidades de almacenamiento, la fiabilidad. Pueden especificar también la utilización de un lenguaje de programación o un método del desarrollo. Las siguientes tablas muestran los requisitos no funcionales del sistema en sus diferentes clasificaciones.

Tabla 2: Usabilidad.

Requisito No Funcional	Descripción
RNU 1. Fácil empleo para usuarios del de la aplicación.	La aplicación presenta una interfaz amigable para que los usuarios puedan aprovechar sus conocimientos y familiarizarse rápidamente.
RNU 2. Menú horizontal.	La aplicación presenta un menú horizontal haciendo relación a los menús tradicionales de la web. En este menú se puede acceder a las funcionalidades de forma sencilla.

Tabla 3: Seguridad.

Requisito No Funcional	Descripción
RNS 1. Autenticación mediante un Directorio LDAP.	La aplicación permite la autenticación mediante el directorio LDAP que proporciona la UCI.
RNS 2. Protección de la Base de Datos.	Se debe proteger la Base de Datos de la aplicación para evitar la pérdida total de la información en caso de algún accidente o ataque.
RNS 3. Permitir el acceso a la aplicación basado en roles.	La aplicación brinda la autorización necesaria para que cada usuario tenga los permisos que le han sido asignados según su rol.

Tabla 4: Diseño e implementación.

Requisito No Funcional	Descripción
RNDI 1. Uso de Enterprise Architect 7.5.	Utilización del Enterprise Architect 7.5 como herramienta CASE para el modelado de los artefactos.
RNDI 2. Uso de Visual Studio Express 2012.	Se utilizará como IDE de desarrollo Visual Studio Express 2012.
RNDI 3. Uso de Microsoft Framework .Net 4.5.	La aplicación debe desplegarse sobre la plataforma de desarrollo Microsoft Framework .Net 4.5 ya que la tecnología ASP.NET MVC 4 y el lenguaje C# 4.0 dependen de este Framework.
RNDI 4. Utilizar el gestor de base de datos PostgreSQL 9.1.1	La aplicación se debe desempeñar sobre el gestor de bases de datos PostgreSQL 9.1.1 pues es un gestor potente, seguro y estable.
RNDI 5. Utilizar C# 4.0, HTML 5 y Java	La aplicación debe desarrollarse con los lenguajes de programación

Script como lenguajes de programación.	C# 4.0, HTML 5 y JavaScript. Se utilizará el Framework jQuery 1.9.1, que proporciona buena potencia para el trabajo en la programación del lado del cliente.
RNDI 6. Utilizar Razor como motor de vistas.	La aplicación utilizará Razor como motor de vistas para aplicaciones web en ASP.Net MVC 4.
RNDI 7. Utilizar el Servidor Apache.	La aplicación debe funcionar sobre un Servidor Apache pues es un servidor multiplataforma, de código abierto y con gran soporte.
RNDI 8. Utilizar las librerías NpglSQL y Mono.Security.	Se utilizará las librerías NpglSQL y Mono.Security para el trabajo con PostgreSQL.
RNDI 9. Emplear la librería NPOI.	Se utilizará la librería NPOI para el trabajo con archivos Excel.
RNDI 10. Emplear el módulo CGI SerialShield Internet Server.	El módulo CGI SerialShield Internet Server es el encargado de recibir las peticiones de generación de licencias de software y transformarla devolviendo como resultado una licencia de software.
RNDI 11. Compatibilidad con los diferentes navegadores.	La aplicación debe ser capaz de poderse ejecutar sin problemas en navegadores como Internet Explorer, Mozilla Firefox y Google Chrome.

Tabla 5: Soporte.

Requisito No Funcional	Descripción
RNSO 1. Ayuda y documentación.	La aplicación brindará la documentación y las plantillas necesarias para la generación de las licencias de software.

Tabla 6: Legales.

Requisito No Funcional	Descripción
RNL 1. Licencia para el uso de SerialShield Manager.	Es necesario contar con el soporte legal adecuado para obtener la herramienta de generación de licencias de software SerialShield Manager pues es el software que se utilizará para la generación de licencias de software.
RNL 2. Aplicación para uso exclusivo de la Universidad de Ciencias Informáticas.	La aplicación está concebida para su uso solamente de la Universidad de Ciencias Informáticas.

Tabla 7: Funcionamiento.

Requisito No Funcional	Descripción
RNFO 1. Sistema operativo Windows Server 2008 o superior.	Debe desplegarse en un Sistema operativo Windows Server 2008 o superior.
RNFO 2. Memoria RAM 512 MB o superior.	La aplicación necesita al menos 512 MB de memoria para funcionar de forma óptima.
RNFO 3. Procesador Pentium IV 2.0 GHz o superior.	La aplicación necesita al menos un procesador Pentium IV 2.0 GHz o superior.

Tabla 8: Interfaz de usuario.

Requisito No Funcional	Descripción
RNIU 1. Predominación del color azul.	En la aplicación predomina el color azul, este es refrescante y agradable a la vista.
RNIU 2. Interfaz de la aplicación tipo Windows.	La interfaz de la aplicación debe ser tipo Windows porque es la más común en la Web y la mayoría de los usuarios poseen una mayor experiencia con este tipo de interfaz.

Tabla 9: Fiabilidad.

Requisito No Funcional	Descripción
RNF 1. Disponibilidad de la aplicación en todo momento.	La aplicación debe estar siempre disponible para que los responsables de generar las licencias puedan usarlo en el momento que sea necesario.
RNF 2. Exactitud en las salidas de la aplicación.	La aplicación debe brindar salidas precisas.

En la **Figura 3** se muestra el diagrama de requisitos no funcionales del sistema agrupados en paquetes según su clasificación.

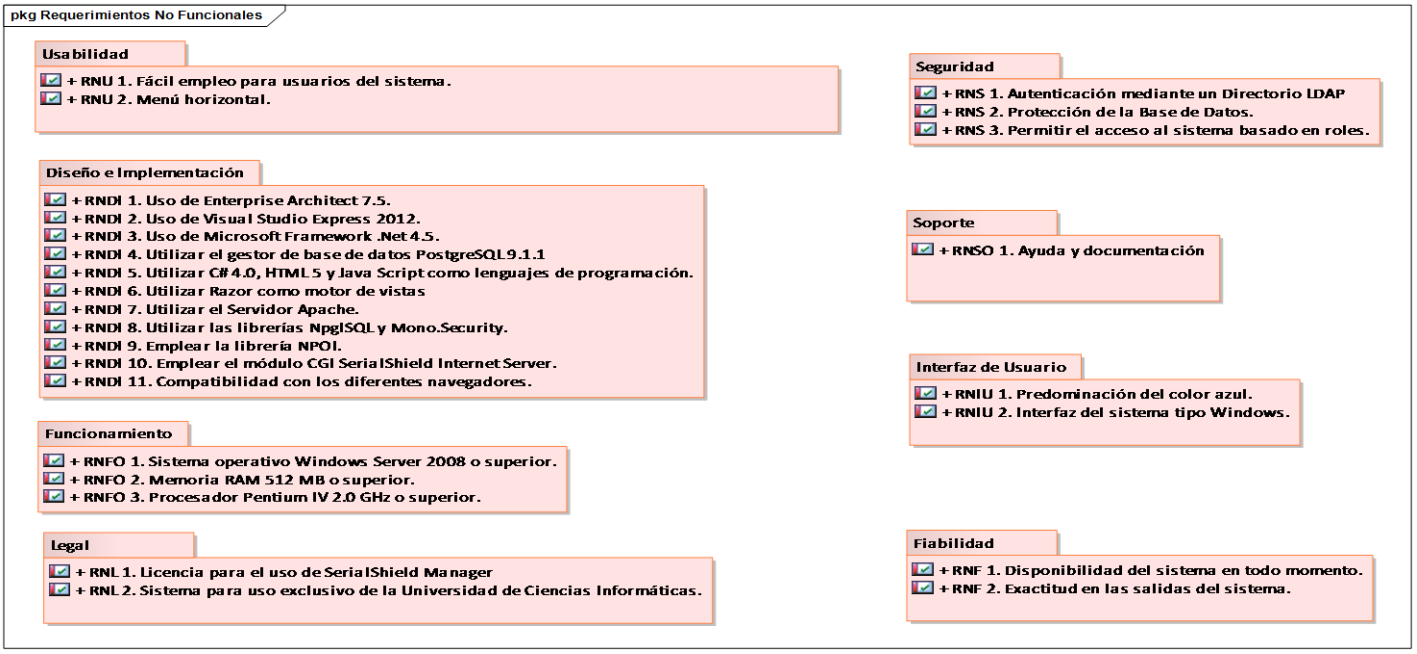


Figura 3: Diagrama de paquetes de requisitos no funciónales del sistema.

2.4. Diagrama de casos de uso del sistema

La **Figura 4** muestra el diagrama de casos de uso del sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

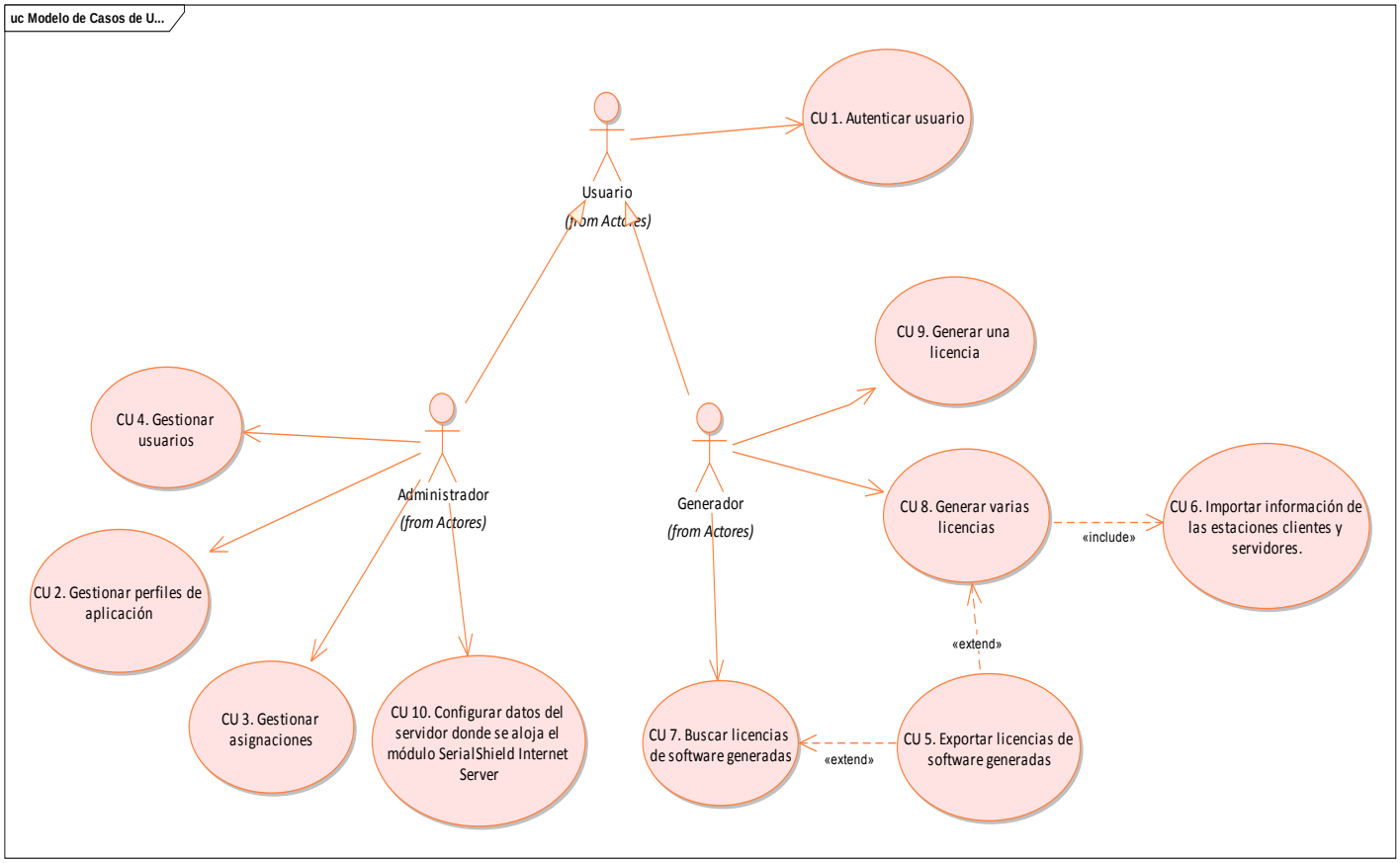


Figura 4: Diagrama de casos de uso del sistema.

2.4.1. Descripción de los casos de uso del sistema

Las tablas que se muestran a continuación contienen un resumen de las descripciones textuales de casos de uso arquitectónicamente significativos. En el **Anexo 2** se muestra la descripción textual completa del caso de uso Gestionar perfiles de aplicación.

Tabla 10: CU Autenticar usuario.

Objetivo	El objetivo que persiguen los actores al realizar este caso de uso es identificarse como un usuario válido del sistema para acceder a sus funcionalidades.
Actores	Usuario (Inicia)
Resumen	Los usuarios acceden a la página de inicio de la aplicación para autenticarse.
Complejidad	Media
Prioridad	Crítico
Referencias	RF 1

Tabla 11: CU Gestionar perfiles de aplicación.

Objetivo	Insertar un nuevo perfil de aplicación, listar los perfiles existentes en el sistema, buscar los perfiles existentes y eliminar perfiles del sistema.
Actores	Administrador del Sistema (Inicia)
Resumen	El Administrador del Sistema introduce en el sistema los perfiles de aplicación necesarios para los cuales se generaran las licencias de software.
Complejidad	Media
Prioridad	Crítico
Referencias	RF 5, RF 5.1, RF 5.2, RF 5.3, RF 5.4

Tabla 12: **CU Gestionar asignaciones.**

Objetivo	Insertar una nueva asignación, listar las asignaciones existentes en el sistema, buscar las asignaciones existentes y eliminar las asignaciones del sistema.
Actores	Administrador del Sistema (Inicia)
Resumen	El caso de uso se inicia cuando el Administrador del Sistema accede a la opción Administración del menú de opciones de la página principal, en el submenú Asignación el sistema brinda la posibilidad de Asignar usuarios a perfiles o Listar asignaciones, opción que permite Buscar y Eliminar una asignación. El caso de uso termina una vez que se haya realizado una o varias de las operaciones anteriores.
Complejidad	Media
Prioridad	Crítico.
Referencias	RF 6, RF 6.1, RF 6.2, RF 6.3, RF 6.4

Tabla 13: CU Gestionar usuario.

Objetivo	Insertar un nuevo usuario del sistema, eliminar usuario del sistema, listar los usuarios existentes en el sistema y buscar los datos de los usuarios del sistema.
Actores	Administrador del Sistema (Inicia)
Resumen	El caso de uso se inicia cuando el Administrador del Sistema accede al menú de Administración y luego al submenú Usuarios, en cuyo caso el sistema brinda la posibilidad de Insertar un usuario local, un usuario global proveniente del directorio LDAP o eliminar un usuario registrado en la base de datos ya sea un usuario local o global. En la opción de Listar Usuario se listan todos los usuarios existentes en el sistema y se permite buscar un usuario por sus datos.
Complejidad	Media
Prioridad	Crítico
Referencias	RF 4, RF 4.1, RF 4.2, RF 4.3, RF 4.4, RF 4.5

Tabla 14: CU Importar información de las estaciones clientes y servidores.

Objetivo	Importar un archivo Excel que contiene la información necesaria para generar licencias de software de las estaciones clientes y servidores.
Actores	Generador (Inicia)
Resumen	El caso de uso se inicia cuando el generador accede a la opción del menú Licencias y al submenú Generar varias licencias y presiona el botón “Examinar” para buscar el directorio del archivo Excel que se desea importar. Este archivo contiene la información de las estaciones clientes y servidores necesarios para generar licencias de software.
Complejidad	Media
Prioridad	Crítico
Referencias	RF 9

Tabla 15: CU Generar varias licencias**.**

Objetivo	Generar varias licencias de software según la aplicación deseada.
Actores	Generador (Inicia)
Resumen	El caso de uso se inicia cuando el generador accede a la opción del menú “Licencias” y luego al submenú “Generar varias licencias”, donde tendrá la opción de cargar un Excel con la información de las PCs y seleccionar el nombre de la aplicación para generar las licencias. El caso de uso termina cuando se han generado las licencias.
Complejidad	Media
Prioridad	Crítico
Referencias	RF 3

Tabla 16: CU Generar una licencia.

Objetivo	Generar una única licencia de software dado los datos necesarios.
Actores	Generador (Inicia)
Resumen	El caso de uso se inicia cuando el Generador accede a la opción del menú “Licencias” y luego al submenú “Generar una licencia” en cuyo caso muestra una pantalla pidiendo los datos necesarios para generar una licencia de software. Luego el sistema muestra la licencia generada.
Complejidad	Media
Prioridad	Crítico
Referencias	RF 2

Tabla 17: CU Configurar datos del servidor donde se aloja el módulo SerialShield Internet Server.

Objetivo	Realizar la configuración de la dirección donde se ubica el servidor que aloja el módulo CGI.
Actores	Administrador (Inicia)
Resumen	El caso de uso se inicia cuando el Administrador accede en el menú principal a la opción “Configuración”, en cuyo caso muestra una pantalla de la configuración actual del servidor y puerto donde se encuentra alojado el módulo CGI SerialShield Internet Server. También permite la modificación del servidor y puerto en caso de que sea necesario cambiarlos.
Complejidad	Baja
Prioridad	Crítico
Referencias	RF 10

Conclusiones del capítulo

En el capítulo se definió la propuesta de la aplicación web. Se describió el procedimiento de licenciamiento de software a partir de las características de los sistemas alas PACS, alas RIS y SerialShield Manager, analizando el proceso de negocio asociado a la generación y control de licencias de software, logrando un modelo único como guía para la implementación del sistema. Se especificaron los requisitos funcionales y no funcionales del sistema.

CAPÍTULO 3. **ARQUITECTURA Y DISEÑO**

En el capítulo se presenta la arquitectura del sistema así como los diagramas de clases de diseño y de secuencia realizados de los casos de uso. Se brinda la descripción de las clases involucradas en la realización de los casos de uso arquitectónicamente significativos.

3.1. Diseño

La actividad del diseño se refiere al establecimiento de las estructuras de datos, la arquitectura general del software, representaciones de interfaz y algoritmos. El proceso de diseño traduce requisitos en una representación de software. (36)

El diseño es técnicamente la parte central de la ingeniería del software. Durante el diseño se desarrollan, revisan y se documentan los refinamientos progresivos de las estructuras de datos, de la estructura del programa y de los detalles procedimentales. El diseño da como resultado representaciones cuya calidad puede ser evaluada. (37)

Los diagramas de las clases de diseño muestran una estructura estática del sistema. En este se modelan clases, atributos y las relaciones entre ellos. Para un pleno entendimiento de los requisitos y las actividades a implementar en el desarrollo de esta investigación, se muestra en la siguiente figura el diagrama de clases de diseño correspondiente al caso de uso arquitectónicamente significativo Gestionar perfiles de aplicación.

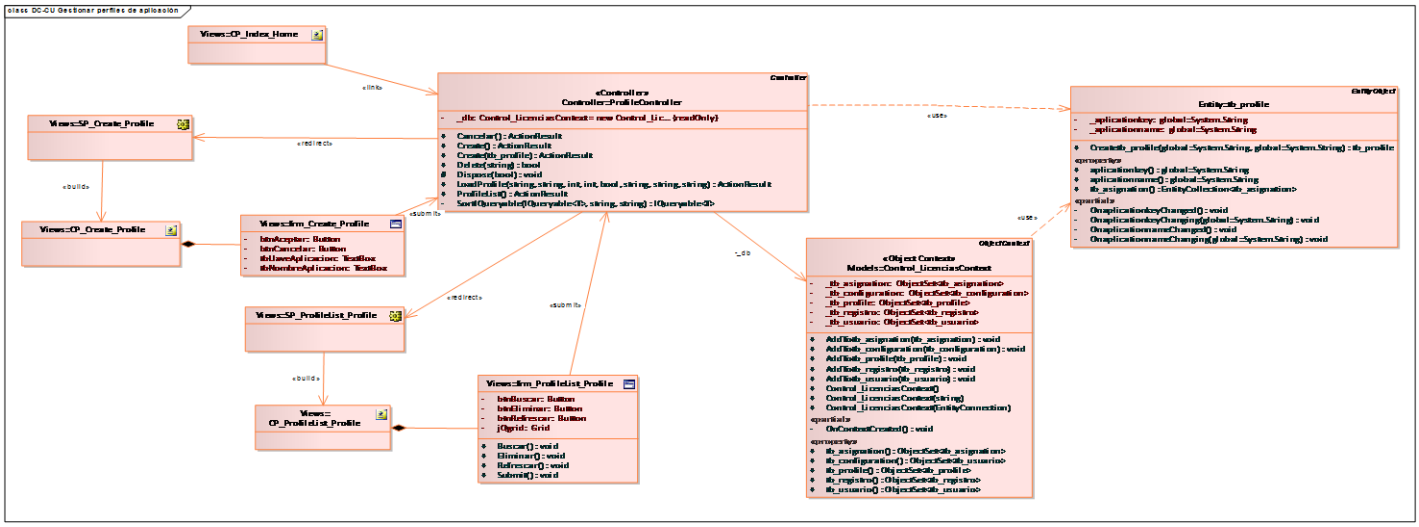


Figura 5: DC-CU Gestionar perfiles de aplicación.

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, se muestran en las siguientes figuras los diagramas de secuencia del caso de uso Gestionar perfiles de aplicación.

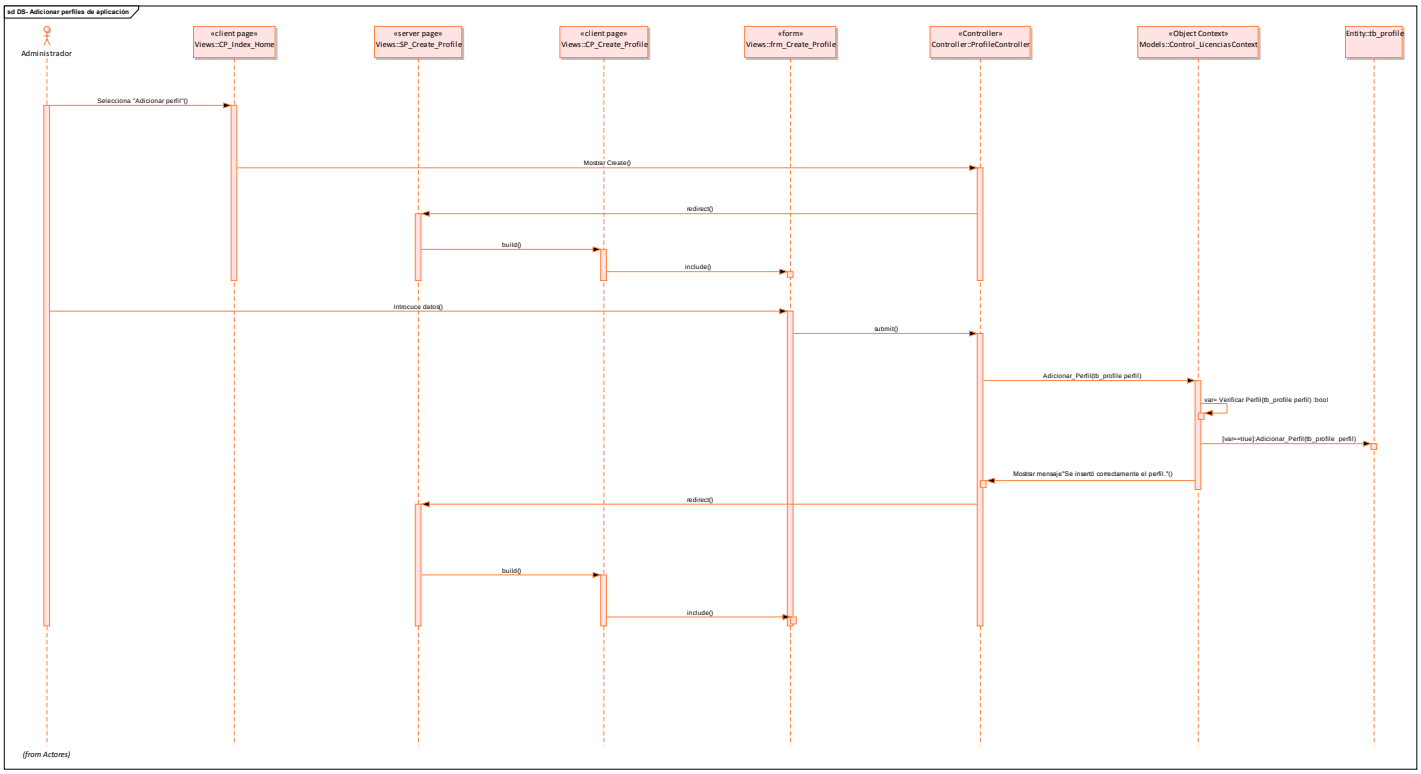


Figura 6: DS-CU Adicionar perfiles de aplicación.

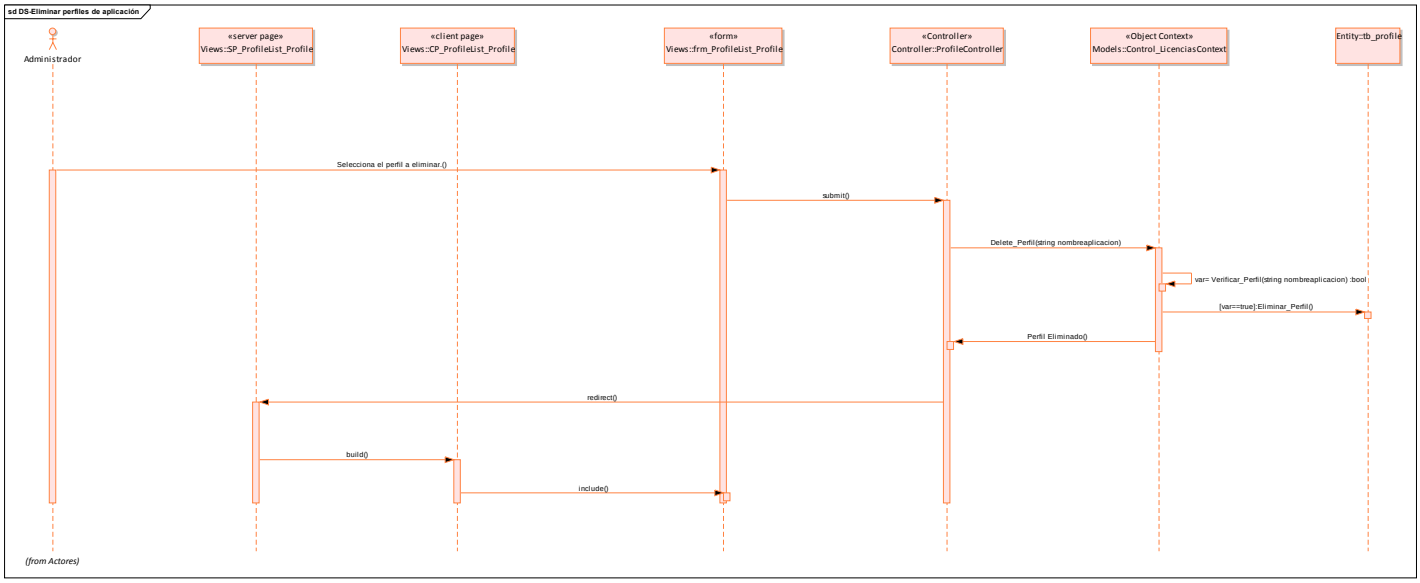


Figura 7: DS-CU Eliminar perfiles de aplicación.

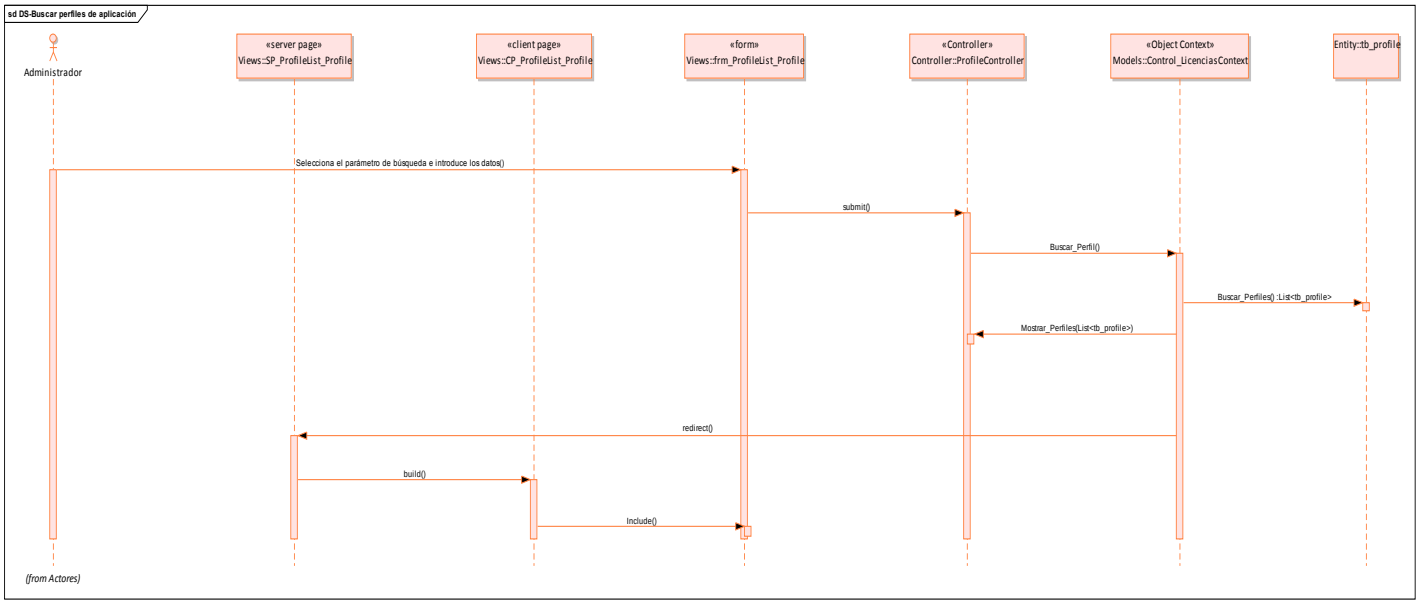


Figura 8: DS-CU Buscar perfiles de aplicación.

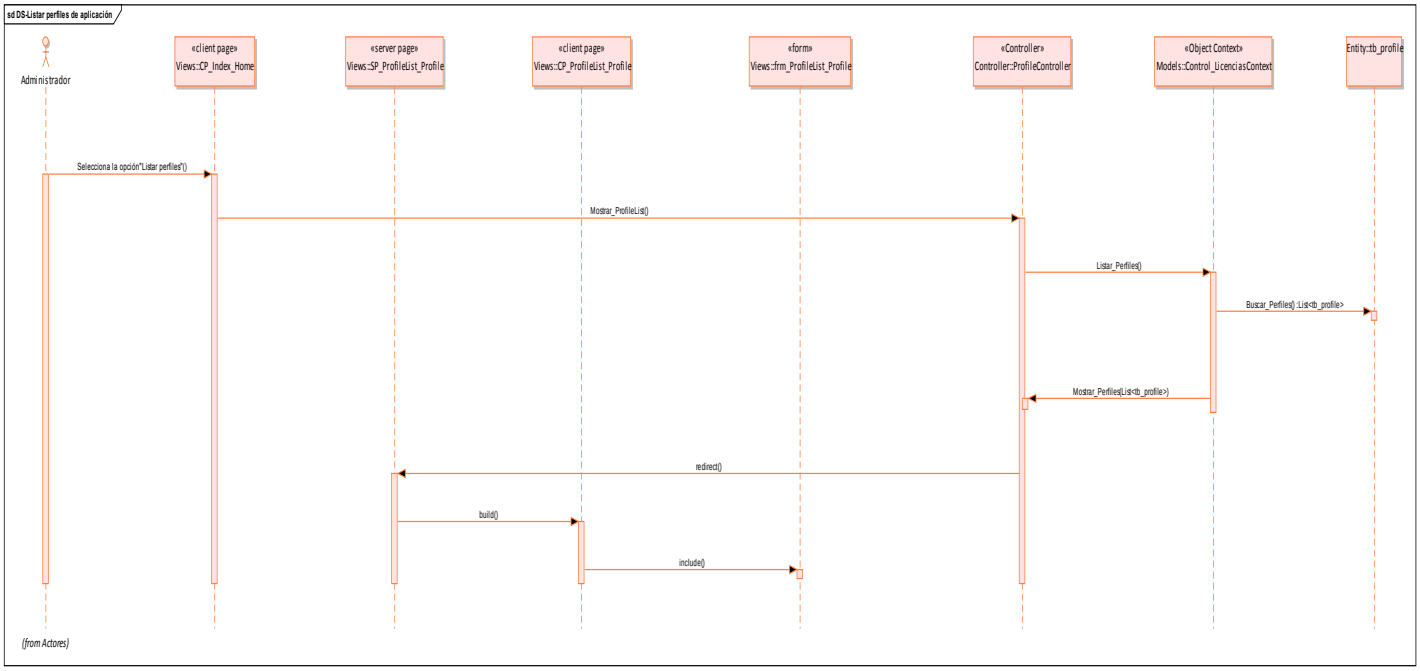


Figura 9: DS-CU Listar perfiles de aplicación.

3.1. Patrones de diseño

En la aplicación se utilizan varios patrones de diseño orientados a objetos, que constituyen patrones generales de software para asignación de responsabilidades, y a la vez son considerados buenas prácticas recomendables en el diseño de software. Dentro de los más utilizados se encuentran:

Patrón Controlador

Con la utilización de este patrón se consigue separar la lógica de negocio de la capa de presentación, logrando la reutilización de código y un mayor control. Además fueron creadas varias clases controladoras que permiten recibir los datos del usuario y enviarlos a las distintas clases según el método invocado, alcanzando aumentar la cohesión y disminuir el acoplamiento. En la siguiente figura se muestra el uso del patrón Controlador.

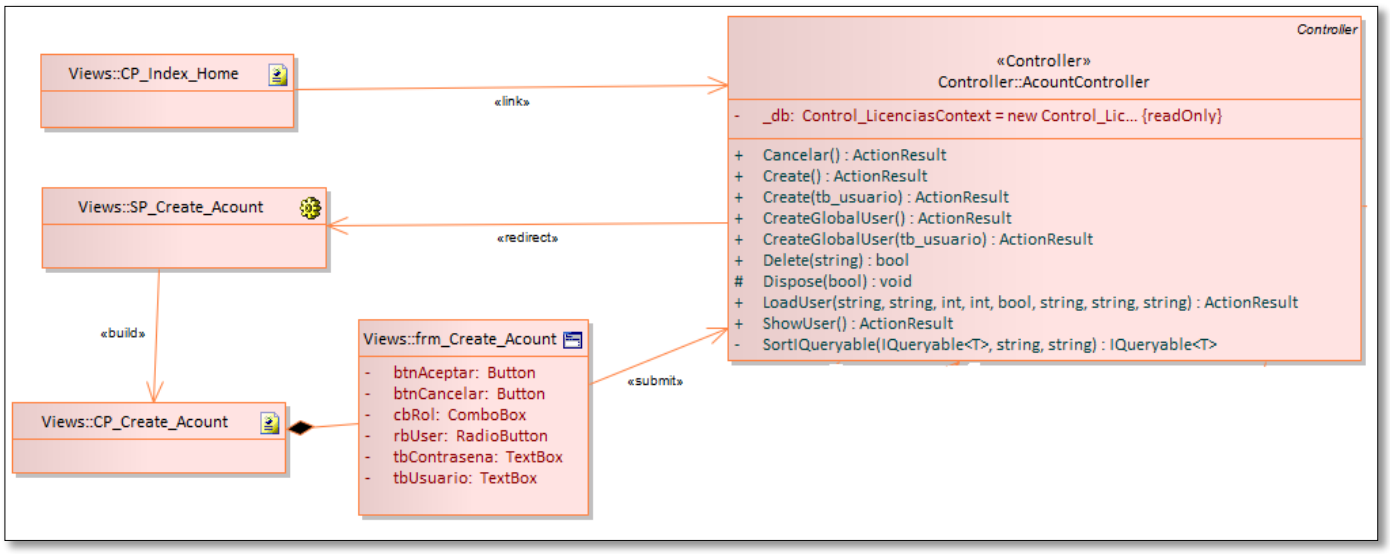


Figura 10: Patrón Controlador.

Patrón Experto

Se utiliza el patrón experto para otorgar las responsabilidades a las clases que contienen la información necesaria para ejecutar un determinado método. En la aplicación se hace uso de este en el momento de exportar las licencias generadas a un archivo Excel. En este caso la clase controladora GenerateController le proporciona los datos necesarios para exportar las licencias a la clase ExcelFileResult, experta en el trabajo y la manipulación de archivos Excel. En la siguiente figura se muestra el uso del patrón Experto.

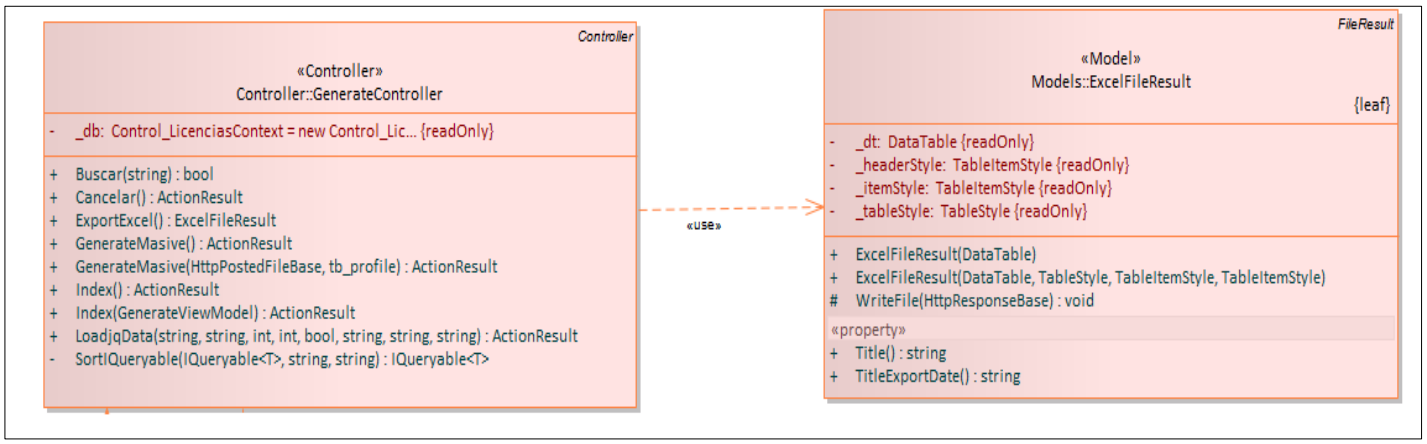


Figura 11: Patrón Experto.

3.2. Modelo de datos

Las bases de datos y sus tecnologías tienen un impacto decisivo en el progresivo uso de las computadoras, pues desempeñan un papel crucial en casi todas las áreas de aplicaciones de la informática. Bajo la estructura de la base de datos se encuentra el modelo de datos: colección de herramientas conceptuales para describir datos, las relaciones, la semántica y las restricciones de consistencia.

En la investigación en curso se hace uso del conjunto de tecnologías de ADO.NET Entity Framework, que permite el desarrollo de aplicaciones de software orientadas a datos. Entity Framework permite a los desarrolladores trabajar con datos en forma de objetos y en un nivel mayor de abstracción cuando tratan con datos, pueden crear y mantener aplicaciones orientadas a datos con menos código que en las aplicaciones tradicionales. La **Figura 12** muestra el modelo de datos de la aplicación.

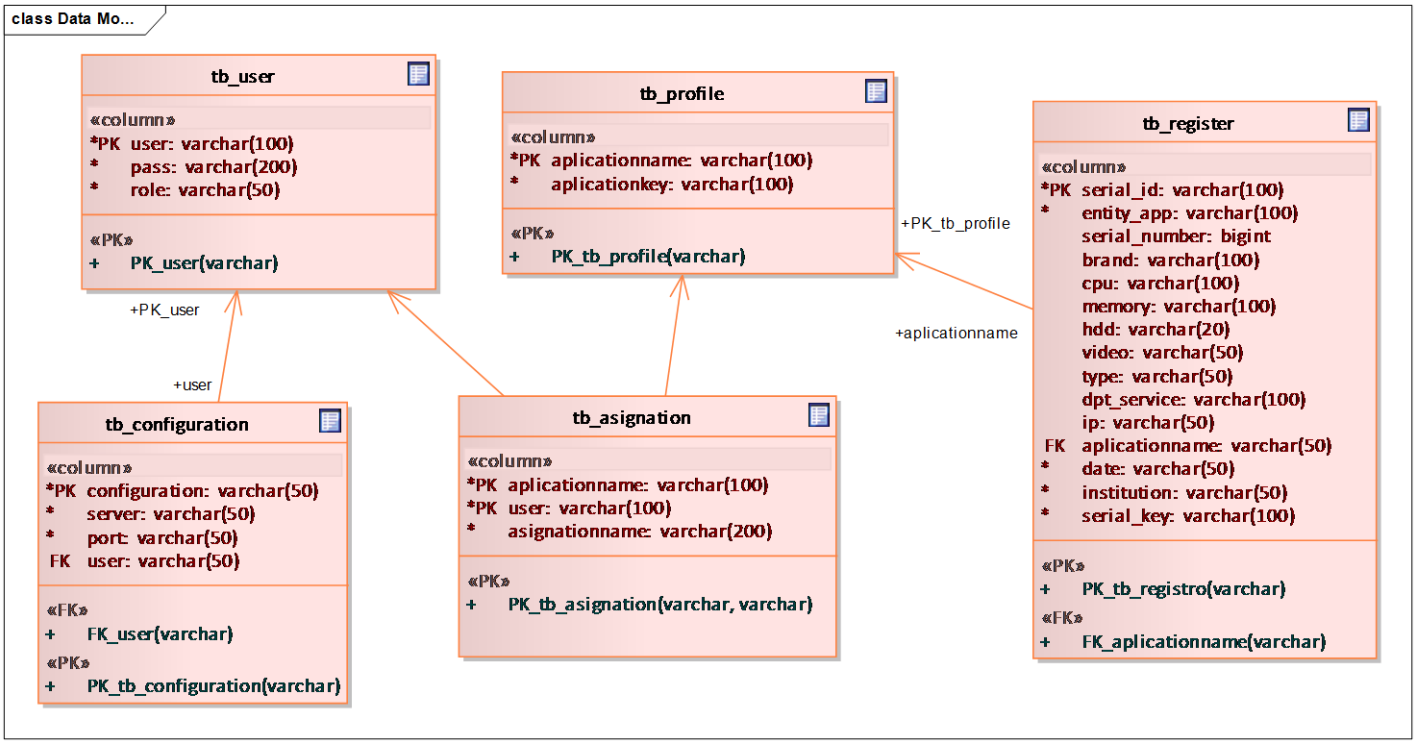


Figura 12: Modelo de datos.

3.3. Modelo arquitectónico

El objetivo de la arquitectura es identificar los requisitos que producen un impacto en la estructura del sistema y reducir los riesgos asociados con la construcción del mismo. La arquitectura debe soportar los cambios futuros del software, del hardware y de las funcionalidades demandadas por los clientes.

La aplicación para la generación y control de licencias de software de los productos de la solución alas PACS-RIS se concibe como una aplicación web basada en el patrón arquitectónico Modelo Vista Controlador (MVC) y el estilo arquitectónico Cliente-Servidor.

Una aplicación web basada en el patrón arquitectónico MVC separa su código en tres partes diferenciadas:

- El controlador: es el punto de entrada de la aplicación, se mantiene a la escucha de todas las peticiones, ejecuta la lógica de la aplicación, y muestra la vista apropiada para cada caso.
- El modelo: contiene todo el código relacionado con el acceso a datos. Es importante que sea un código lo más genérico posible y se pueda reutilizar en otras situaciones y proyectos. Nunca se incluye lógica en el modelo, solamente consultas a la base de datos y validaciones de entrada de datos.
- La vista: contiene el código que representará lo que el usuario verá en pantalla.

La **Figura 13** representa el flujo del patrón arquitectónico MVC:

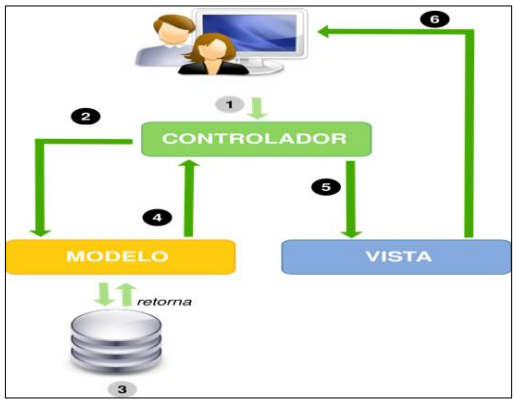


Figura 13: Flujo del patrón arquitectónico MVC.

MVC desacopla vista y modelo, estableciendo un protocolo de subscripción / notificación entre ellos. Este mecanismo asegura que todas las vistas reflejan el estado actual del modelo, mediante notificaciones de

cambio del modelo a todas las vistas que dependen de él. Permite crear nuevas vistas sin tener que modificar el modelo y tener tantas vistas como se desee. El controlador encapsula todos los mecanismos de respuesta a una entrada producida en el sistema. (38)

El estilo Cliente-Servidor es el que define una relación entre dos aplicaciones en las cuales una de ellas (cliente) envía peticiones a otra (servidor fuente de datos), para su procesamiento. Es un estilo para sistemas distribuidos que divide el sistema en una aplicación cliente, una aplicación servidora y una red que las conecte, en la cual la primera realiza peticiones y la segunda emite respuesta usando un amplio rango de protocolos y formatos de datos para comunicar información. En la **Figura 14** se muestra el flujo del estilo arquitectónico Cliente-Servidor.

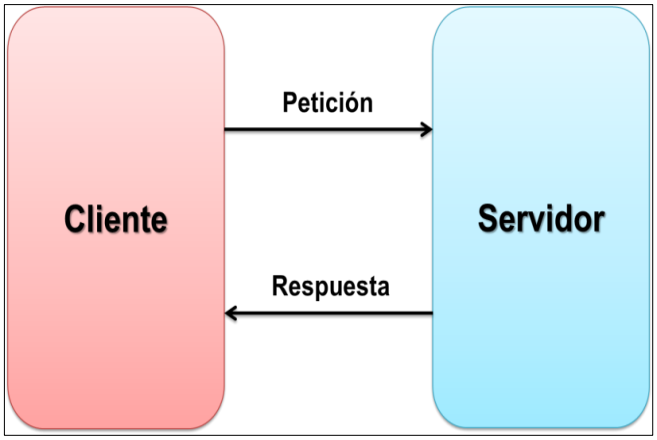


Figura 14: Flujo del estilo arquitectónico Cliente-Servidor.

Esta arquitectura consiste básicamente en el intercambio constante de información entre el cliente o los clientes, que realizan solicitudes o peticiones de forma activa, y el o los servidores, que envían respuestas de forma pasiva. (39) En la aplicación web para la generación y control de licencias de software se evidencia como una de las características principales, el modelo Cliente-Servidor.

La comunicación entre la aplicación cliente y la servidora es a través de la Notación de Objetos de JavaScript (JSON) encargada del intercambio de datos entre ambos lados de la solución. JSON es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son

ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. (40)

En la aplicación cliente se utilizan tecnologías como HTML5, JQuery y el motor de vistas Razor para la creación de las interfaces de usuario. Por otra parte la aplicación servidora combina el lenguaje C# y las librerías de clases del framework .Net en su versión 4.5. La aplicación utiliza Entity Framework que permite el desarrollo de aplicaciones de software orientadas a datos y permite trabajar con datos en forma de objetos y propiedades específicos del dominio, como clientes y direcciones de cliente, sin tener que preocuparse por las tablas y columnas de la base de datos subyacente donde se almacenan estos datos. (41)

Conclusiones del capítulo

En el desarrollo del capítulo se presentaron los diagramas de clases del diseño, los diagramas de secuencia de los casos de uso arquitectónicamente significativos. Se definió como arquitectura de la aplicación web para la generación y control de las licencias de software de los productos de la solución alas PACS-RIS el patrón arquitectónico MVC y el estilo arquitectónico cliente servidor.

CAPÍTULO 4. **IMPLEMENTACIÓN Y PRUEBAS**

En este capítulo se describen los temas relacionados con la implementación de la aplicación, se muestran los componentes fundamentales, así como la relación que existe entre los mismos mediante el diagrama de componentes y se específica a través del diagrama de despliegue como se realizará la distribución física de la aplicación.

4.1. Diagrama de componentes

En los diagramas de componentes se muestran los elementos de diseño de un sistema de software. Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. Los diagramas de componentes pueden utilizarse para describir un diseño que se implemente en cualquier lenguaje o estilo. Solo es necesario identificar los elementos del diseño que interactúan con otros elementos del diseño a través de un conjunto restringido de entradas y salidas. Los componentes pueden tener cualquier escala y pueden estar interconectados de cualquier manera. (42)La **Figura 15** muestra el diagrama de componentes de la aplicación desarrollada.

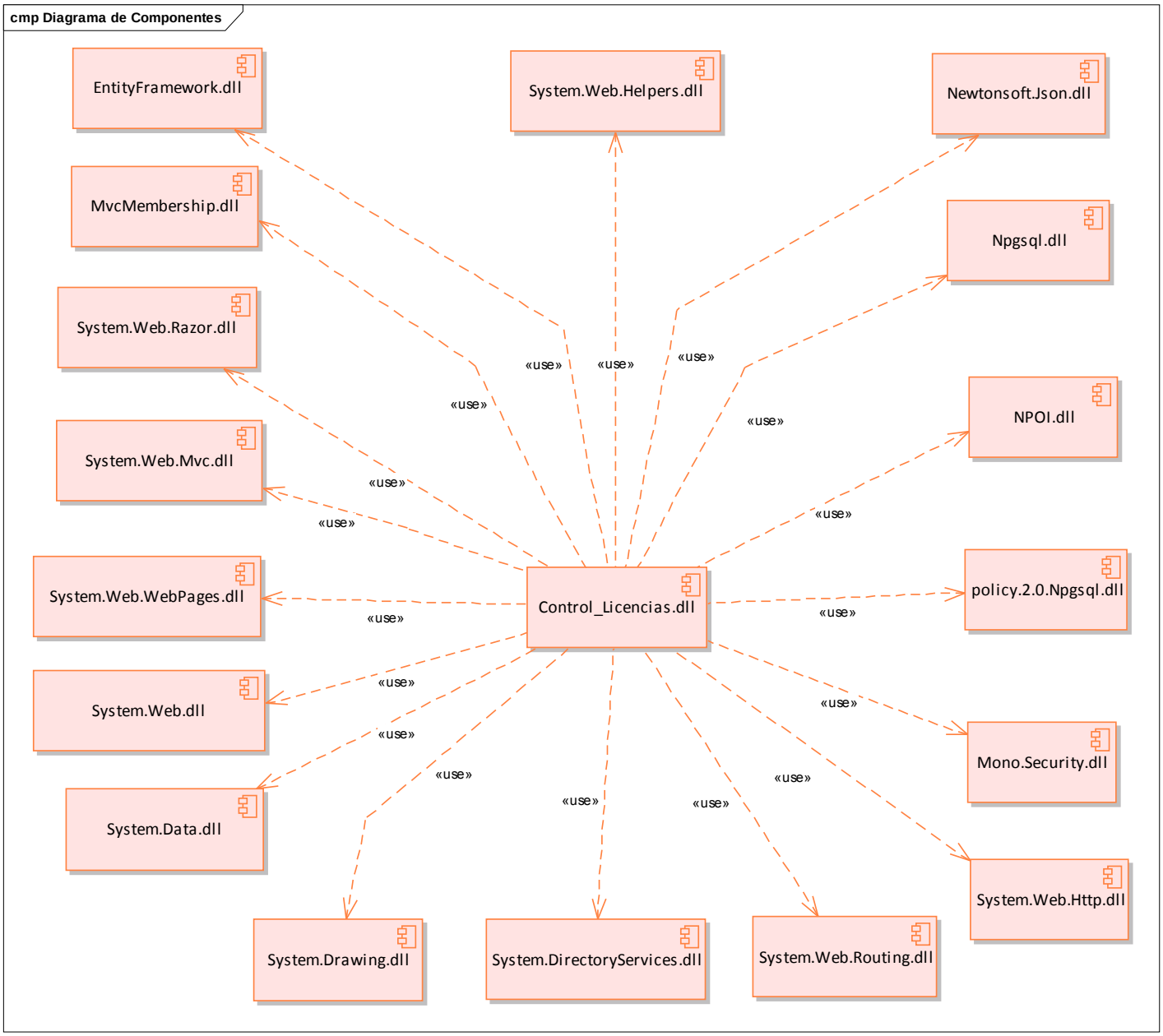


Figura 15: Diagrama de componentes.

4.2. Diagrama de despliegue

El diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Indica la situación física de los componentes lógicos desarrollados. Es decir se sitúa el software en el hardware que lo contiene, donde cada elemento de hardware se representa como un nodo .La siguiente figura muestra el diagrama de despliegue de la aplicación.

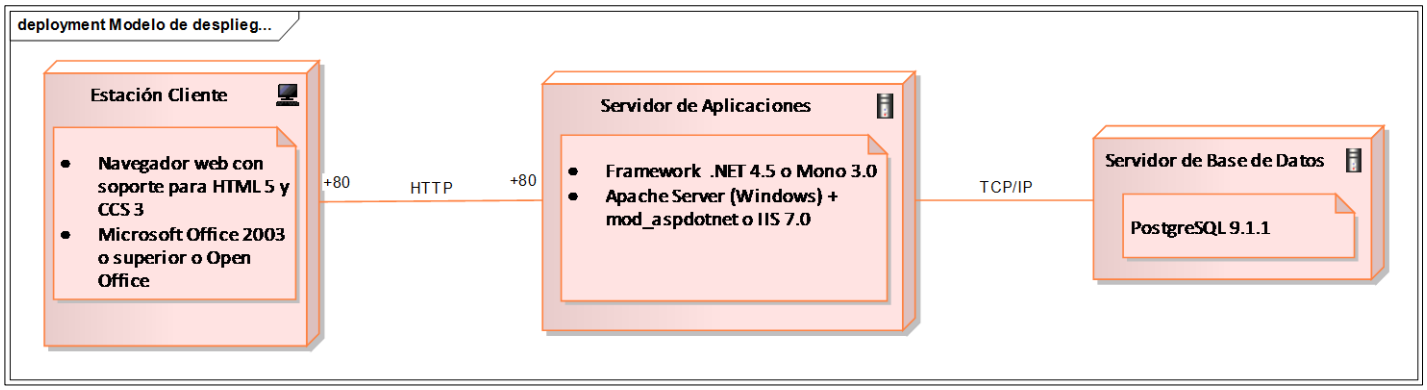


Figura 16: Diagrama de despliegue.

4.3. Estándar de codificación

Para lograr una mayor comprensión del código de la aplicación por otros desarrolladores y la uniformidad del mismo, se utilizó un estándar de codificación el cual se muestra a continuación:

- ✓ Usar nombres descriptivos para los nombres de clases, propiedades y métodos. Asignar los nombres en idioma Inglés, siempre buscando que estén escritos correctamente.
- ✓ Declarar los nombres de los atributos de las clases con underscore (_) y letra inicial minúscula, si es un nombre compuesto utilizar minúscula y mayúscula. Ejemplos: _wadoRequest, _name.
- ✓ Declarar los nombres de las propiedades en mayúscula y si son palabras compuestas, notación camel case, o sea Mayúscula y Mayúscula. Ejemplos: Description, DateAdded.
- ✓ Declarar los nombres de los métodos en mayúscula y si son palabras compuestas, notación camel case, o sea Mayúscula y Mayúscula. Ejemplos: Salary, ComputePrice.
- ✓ Hacer uso de clases estáticas para colocar métodos que se utilicen en varias partes del programa, y funciones que puedan desacoplarse para hacer una reutilización posterior del código.

- ✓ Respetar y promover el uso correcto de los modificadores de visibilidad, internal, protected y private.
- ✓ Poner comentario en las clases y en los métodos. Además agregar comentario a las instrucciones complejas, que sean de mediano a alto nivel de comprensión. Escribir los comentarios en español para facilitar su comprensión.

A continuación se muestra un fragmento de código de la clase AcountController donde se observa el uso del estándar de codificación anterior:

```
namespace Control_Licencias.Controllers
{
    [Authorize(Roles = "Administrador")]
    //Clase controladora para la gestión de todos los usuarios de la aplicación.
    public class AcountController : Controller
    {
        private readonly Control_LicenciasContext _db = new Control_LicenciasContext();

        public ActionResult Create()
        {
            return View();
        }

        /* Adicionar un usuario local, en este se almacena el usuario, la contraseña
        y el rol del usuario creado por el administrador en la BD local.*/

        [HttpPost]
        public ActionResult Create(tb_user tbUser)
        {
            if (ModelState.IsValid)
            {
                try
                {
                    if (tbUser.user != null && tbUser.pass != null && tbUser.role != null)
                    {
                        var user = new tb_user
                                    {
                                        user = tbUser.user,
                                        pass = Encriptation.GetMd5(tbUser.pass),
                                        role = tbUser.role
                                    };

                        _db.tb_user.AddObject(user);
                        _db.SaveChanges();
                        ViewBag.Message = "Se insertó correctamente el usuario.";
                        return View();
                    }
```

```
                else
                {
                    ViewBag.Message = "Existen campos vacíos.";
                }
            }
            catch (Exception)
            {
                ViewBag.Message = "Este usuario ya existe en el sistema.";
            }
        }
        return View(tbUser);
    }
  }
}
```

4.4. Pruebas realizadas sobre la aplicación web para el control de licencias de software de los productos de la solución alas PACS-RIS

Las pruebas del software son un elemento crítico para la garantía de la calidad del software y presentan una revisión final de las especificaciones, diseño y de la codificación. No es raro que el coste de las pruebas del software suponga el 40% del coste total de todo el desarrollo. (43)

La fase de pruebas es una de las más importantes del desarrollo de un proyecto web, la misma se centra en 7 elementos fundamentales: contenido, interfaz, navegación, componente, configuración, desempeño y prueba de seguridad. Estas pruebas son una actividad a través de la que un sistema se ejecuta sobre unas condiciones o requerimientos específicos. Los resultados obtenidos a partir de estos procesos son observados, registrados y evaluados por los desarrolladores del software.

Una vez concluida la implementación de la aplicación es necesario realizar pruebas para determinar errores antes de la interacción de los usuarios finales con la aplicación. Las pruebas realizadas a la aplicación web para el control de las licencias de software de los productos de la solución alas PACS-RIS fueron las siguientes:

Nivel de Prueba: Pruebas de Sistema (seguridad, carga / estrés)

Después de implementar los requisitos funcionales del sistema se procede a la revisión del mismo para verificar su correcto funcionamiento. Estas pruebas verifican que se alcanza las funcionalidades y el rendimiento del sistema total. Las pruebas de sistema tienen como propósito ejercitar profundamente el sistema, verificar el cumplimiento de los requisitos funcionales establecidos, el funcionamiento y rendimiento

de las interfaces hardware, software y de usuario, la adecuación de la documentación de usuario y el rendimiento y respuesta en condiciones límite y de sobrecarga. En este tipo de prueba se recomienda realizar pruebas de funcionalidad, seguridad, de volumen/estrés.

Descripción de las pruebas:
Para la realización de las pruebas de sistema se diseñaron los Casos de Prueba basados en casos de uso y el rol especificado fue el desarrollador. En el desempeño de las mismas se verificó la correspondencia de la aplicación con lo descrito en los casos de prueba.

Técnicas de Prueba: Pruebas de Funcionalidad.

Para probar las funcionalidades de la aplicación se diseñaron y aplicaron Casos de Pruebas basado en casos de uso. Se comprobó que la aplicación se corresponda con lo definido para el desarrollo de la misma, entiéndase Requisitos Funcionales, Descripción de Casos de Uso, Estándares de Diseño, entre otros.

Las pruebas de funcionalidad se utilizaron para la verificación del trabajo con los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. Estas se centraron en verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio comprobando la apropiada aceptación de datos. Se determinó que las pruebas a realizar se regirán por el método de caja negra.

Métodos de Prueba: Caja Negra.
Para la realización de las pruebas se seleccionó el método de prueba de caja negra. El probador ejecutó todos los escenarios de prueba descrito en los casos de prueba. El objetivo de probar cada uno de los escenarios fue la verificación de que los flujos funcionen como se describen en la Especificación de requisitos, se introdujeron datos válidos e inválidos incluyendo los campos en blanco para verificar las correctas respuestas de la aplicación y los mensajes de precaución en caso de error.

Dentro del método de Caja Negra se utilizó la técnica de la Partición de Equivalencia. Esta es una de las más efectivas, permite examinar los valores válidos e inválidos entrados en la aplicación y descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos de prueba antes de detectar el error genérico.

Diseño de Casos de Prueba basado en casos de uso:

Para diseñar los casos de prueba se tienen en cuenta cada uno de los flujos descritos en las especificaciones de casos de uso. Los casos de prueba contienen la especificación de un caso de uso, dividido en secciones y escenarios. Estos escenarios describen las funcionalidades a modo de actividades. Se describe cada variable que recoge el caso de uso en cuestión. Para detallar el caso de uso se utiliza una tabla, desglosándose cada funcionalidad en secciones y estas a su vez en escenarios, haciendo así más ventajosa la realización de las pruebas. En las siguientes tablas se presentan los escenarios probados de los casos de uso Autenticar usuario y Generar una licencia:

Tabla 18: Escenarios de prueba para el caso de uso Autenticar usuario.

Nombre de sección	**Descripción de la funcionalidad.**
EC 1.1 Autenticar usuario satisfactoriamente	El usuario introduce su usuario y contraseña correctamente en la aplicación.
EC 1.2 Autenticar usuario con datos incorrectos.	El usuario introduce un usuario y una contraseña incorrectos.
EC 1.3 Autenticar usuario con datos incompletos	El usuario deja algún campo necesario sin introducir los datos.

Las variables asociadas a los escenarios descritos se muestran en la siguiente tabla:

Tabla 19: Descripción de las variables asociadas a los escenarios de caso de uso Autenticar usuario.

Número	**Nombre del campo**	**Clasificación**	**Valor nulo**	**Descripción**
Variable 1	Usuario	Campo de Texto	No	El campo debe tener como entrada un string. Este string debe ser un usuario introducido con anterioridad en la base de datos de la aplicación.
Variable 2	Contraseña	Campo de Texto	No	El campo debe tener como entrada un string. Este string debe ser una contraseña introducida con anterioridad en la base de datos de la aplicación o

				en el directorio LDAP de la UCI.
Variable 3	Entrar	Botón	No	Se debe dar clic en el botón

Tabla 20: Escenarios de prueba para el caso de uso Generar una licencia.

Nombre de la Sección	Descripción de la funcionalidad
EC. 1.1 Generar una licencia de software satisfactoriamente.	Permite al usuario generador construir una licencia a partir de los datos suministrados.
EC. 1.2 Generar una licencia de software con datos incompletos.	El usuario Generador deja uno o los dos campos de texto vacíos o no selecciona el nombre de aplicación.

Las variables asociadas a los escenarios descritos se muestran en la siguiente tabla:

Tabla 21: Descripción de las variables asociadas a los escenarios de caso de uso Generar una licencia.

No	Nombre del campo	Clasificación	Valor Nulo	Descripción
Variable 1	Entidad de aplicación	Campo de Texto	No	El campo debe tener como entrada un string. Este string debe ser un nombre de entidad de aplicación.
Variable 2	Serial ID	Campo de Texto	No	El campo debe tener como entrada un string. Este string debe ser el serial id de la pc en cuestión.
Variable 3	Nombre de aplicación	Selección	No	Debe seleccionarse un elemento de la lista.
Variable 4	Institución	Campo de Texto	No	El campo debe tener como entrada un string. Este string debe ser el nombre de la institución donde se realiza la instalación de la solución alas PACS-RIS.
Variable 5	Aceptar	Botón	No	Se debe dar clic en el botón.
Variable 6	Cancelar	Botón	No	Se debe dar clic en el botón.

Con la descripción de las variables asociadas a los escenarios de pruebas se evaluó y probó la validez de cada uno de los datos introducidos en la aplicación para los casos de uso en cuestión. Las siguientes tablas muestran los resultados de las pruebas utilizando la técnica de partición de equivalencia.

Tabla 22: **Matriz de datos del caso de uso Autenticar usuario.**

Caso de uso	**Variable**	**Respuesta del sistema**	**Resultado de la prueba**	**Flujo central**
Autenticar usuario.	rlicor raidel Clic	El sistema entra a la página principal.	Satisfactorio	1. El Usuario accede mediante el navegador web a la página de inicio de la aplicación. 2. El Sistema muestra la página de inicio solicitando usuario y contraseña para la autenticación. 3. El Usuario introduce su usuario y contraseña y presiona el botón “Entrar”. 4. El Sistema valida los datos. 5. El Sistema comprueba que el usuario sea válido en el sistema. 6. El Sistema comprueba que la contraseña esté correcta, mostrando la página principal. 7. Termina el caso de uso.
	rlicor24 Clic	El sistema muestra el mensaje “El nombre de usuario o la contraseña son incorrectos”	Satisfactorio	1. El Usuario accede mediante el navegador web a la página de inicio de la aplicación. 2. El Sistema muestra la página de inicio solicitando usuario y contraseña para la autenticación. 3. El Usuario introduce un usuario y contraseña que no está registrado en la base de datos. 4. El Sistema muestra el mensaje “El nombre de usuario o la contraseña son incorrectos” 5. Termina el caso de uso.

Tabla 23: **Matriz de datos del caso de uso Generar una licencia.**

Caso de uso	Variable	Respuesta del sistema	Resultado de la prueba	Flujo central
Generar una licencia.	PHAS_IMG_WS1 4878-0047-F729-5BC9 Frank País Alas RIS Clic Clic	El sistema muestra la licencia generada y el mensaje "Licencias generadas satisfactoria mente".	Satisfactorio	1. El Generador selecciona la opción “Generar una licencia” ubicado dentro del menú “Licencias”. 2. El Sistema muestra una pantalla permitiendo introducir los datos necesarios para generar una licencia. •Entidad de aplicación. •Serial ID. •Nombre de aplicación. y dando la posibilidad de: •“Aceptar”, donde se procedería a generar la licencia de software. •“Cancelar”. 3. El Generador introduce los datos para generar la licencia de software. •Entidad de aplicación. •Serial ID. •Institución. •Nombre de aplicación. 4. El Sistema valida los datos introducidos. 5. El Sistema muestra una licencia de software. 6. Termina el caso de uso.

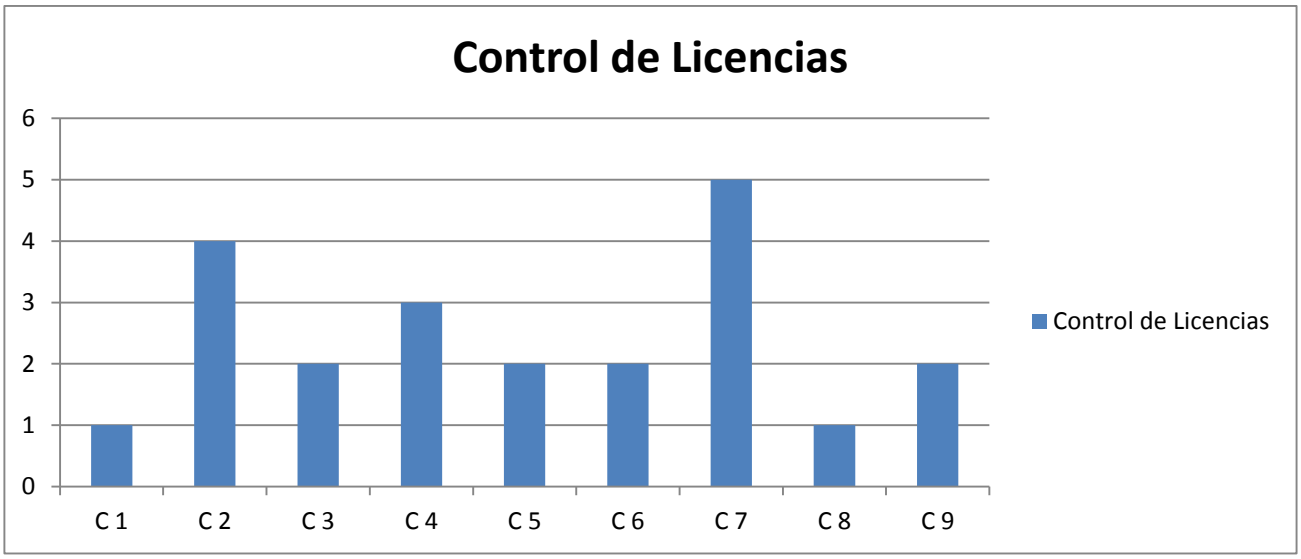
	En blanco En blanco En blanco Alas RIS Clic Clic	El sistema muestra el mensaje "Existen campos vacíos".	Satisfactorio	1. El Generador selecciona la opción “Generar una licencia” ubicado dentro del menú “Licencias”. 2. El Sistema muestra una pantalla permitiendo introducir los datos necesarios para generar una licencia. •Entidad de aplicación. •Serial ID. •Institución. •Nombre de aplicación. y dando la posibilidad de: •“Aceptar”, donde se procedería a generar la licencia de software. •“Cancelar”. 3. El Generador introduce los datos para generar la licencia de software. •Entidad de aplicación. •Serial ID. •Nombre de aplicación. 4. El Sistema valida los datos introducidos. 5. El Sistema muestra el mensaje "Existen campos vacíos". 6. Termina el caso de uso.

En la realización de las pruebas se detectaron errores de diferentes tipos, siguiendo los siguientes criterios:

1. Error Técnico
2. Excepciones
3. Funcionalidad
4. Error de Interfaz
5. Formato
6. Ortografía

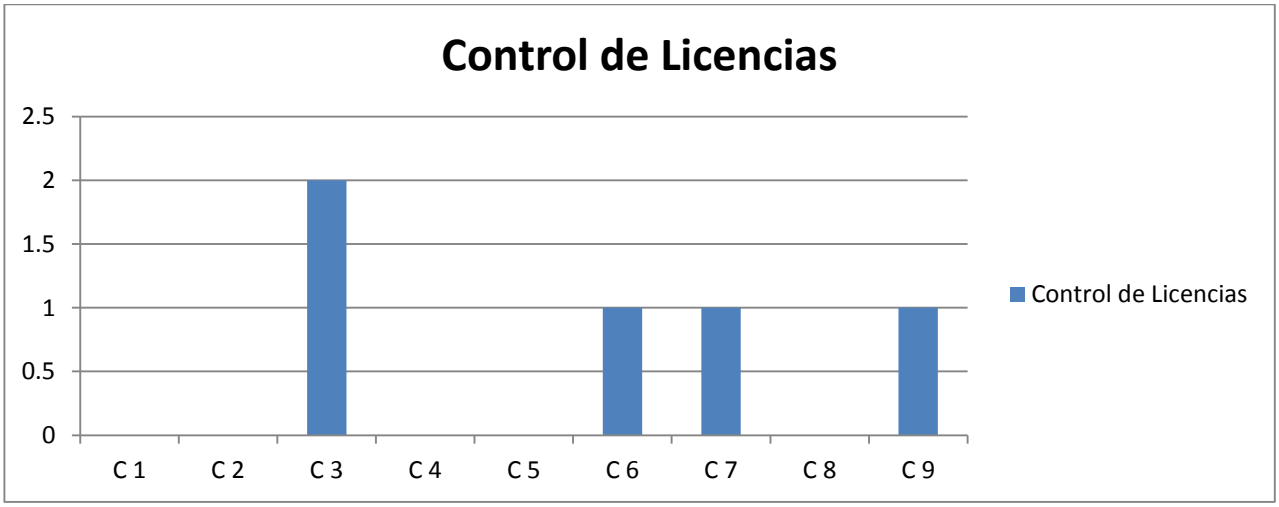
7. Validación 8. Redacción 9. Diseño CP

La siguiente gráfica muestra los resultados de las pruebas en la 1ra iteración:



Gráfica 1: Resultados de la primera iteración de las pruebas.

Luego de la 1ra iteración de las pruebas y la corrección de las no conformidades se realizó la segunda iteración de pruebas arrojando los siguientes resultados:



Gráfica 2: Resultados de la segunda iteración de las pruebas.

Al terminar la segunda iteración de las pruebas a la aplicación se detectaron 5 no conformidades, de las cuales se lograron resolver 3, pues las 2 no conformidades detectadas en el criterio de funcionalidad corresponden a que el módulo SerialShield Internet Server está expirado. Las siguientes figuras muestran la respuesta de la aplicación en las funcionalidades Generar una licencia y Generar varias licencias:

Figura 17: Funcionalidad Generar una licencia.

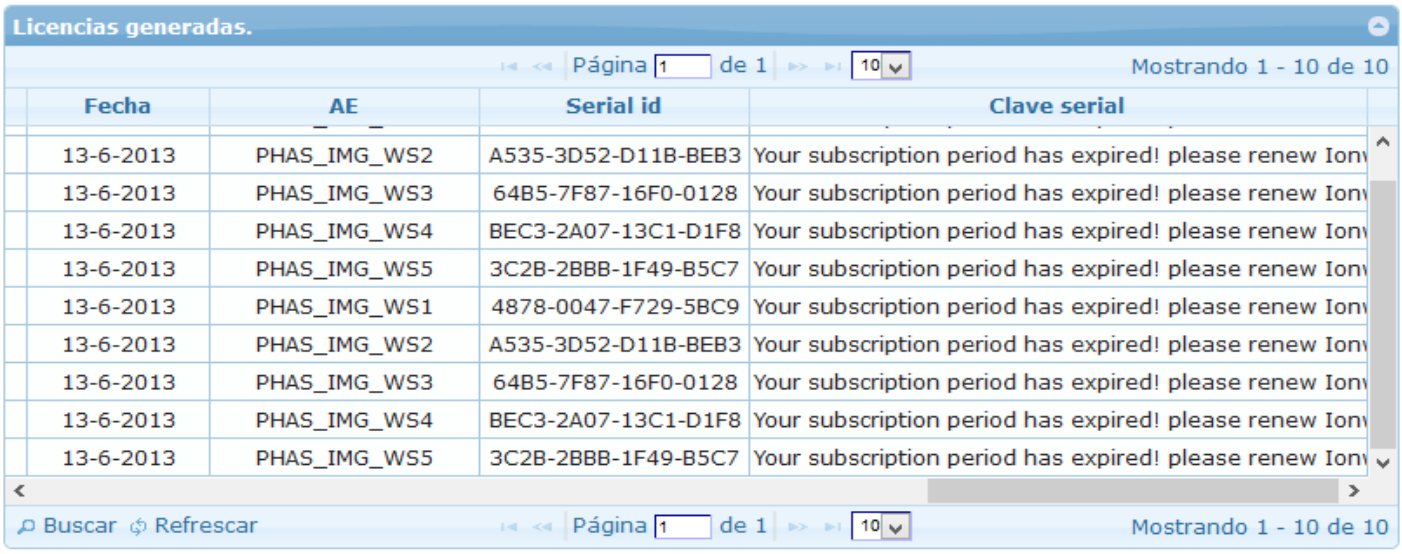


Figura 18: Funcionalidad Generar varias licencias.

Conclusiones del capítulo

En el capítulo se mostraron los diagramas de despliegue y de componente de la aplicación. Se describió el estándar de codificación utilizado en la implementación de la aplicación y se realizaron las pruebas de caja negra para evaluar el correcto funcionamiento de la misma, arrojando como resultado que el módulo CGI SerialShield Internet Server que se utiliza para generar las licencias de software está expirado.

CONCLUSIONES

Luego de dar cumplimiento al objetivo planteado en la presente investigación se arriban a las siguientes conclusiones:

Se realizó un análisis crítico y valorativo de los sistemas informáticos de control de licencias de software existentes a nivel nacional e internacional, ratificando el uso de la herramienta SerialShield Manager para la generación de licencias de software de los productos de la solución alas PACS-RIS.

Se describió el procedimiento de licenciamiento de software a partir de las características de los sistemas alas PACS, alas RIS y SerialShield Manager, analizando el proceso de negocio asociado a la generación y control de licencias de software, logrando un modelo único como guía para la implementación de la aplicación.

Se generaron los productos correspondientes a las fases definidas en la UCI para el ciclo de vida de los proyectos del programa de mejoras, permitiendo documentar todos los procesos de la aplicación en cuestión.

Con el debido uso de las herramientas propuestas, los patrones definidos y las pautas y estándares establecidos se desarrolló la aplicación dando cumplimiento a las especificaciones de Requisitos de software, convirtiendo la aplicación para el control de licencias de software de los productos de la solución alas PACS-RIS en una aplicación robusta e integral que servirá como herramienta de soporte a dicha solución.

Se comprobó el correcto funcionamiento de la aplicación a través de las pruebas de software definidas, validando la calidad, seguridad y cumplimiento de las pautas y estándares propuestos para la misma.

RECOMENDACIONES

A partir de la investigación efectuada y los conocimientos y experiencias acumuladas durante el trabajo realizado se proponen las siguientes recomendaciones:

- ✓ Adquirir el módulo CGI SerialShield Internet Server para lograr la generación de las licencias de software a través de la aplicación web desarrollada.
- ✓ Incorporar el mecanismo de validación de licencias de software haciendo uso de la librería SerialShield.dll a las aplicaciones desarrolladas por la UCI en los lenguajes .NET (C #, VB.NET, Delphi.NET) y herramientas para Win32 como Microsoft Visual Basic, Microsoft Visual C + +, Borland Delphi y C + + Builder.
- ✓ Hacer uso de la aplicación web desarrollada para la generación y control de las licencias de software en la Universidad de las Ciencias Informáticas.

REFERENCIAS BIBLIOGRÁFICAS

1. **Intelectual, Organización Mundial de la Propiedad.** OMPI. *OMPI.* [Online] [Cited: enero 10, 2013.] http://www.wipo.int/about-ip/es/index.html.

2. **Microsoft .** Microsoft . *Microsoft .* [Online] abril 7, 2008. [Cited: enero 10, 2013.] http://www.microsoft.com/spain/sharedsource/licensingbasics/licensingmodels.mspx.

3. **Ramón, Gómez Labrador.** *TIPOS DE LICENCIAS DE SOFTWARE.* 2005.

4. **García Moreno, Rafael .** Vector. *Vector.* [Online] marzo 28, 2007. [Cited: enero 10, 2013.] http://www.vectorsf.com/noticias-y-eventos/articulos-opinion/el-software-libre-el-futuro-del-desarrollo-de-aplicaciones.

5. **Intercontinental, SCM Soporte.** SCM Soporte Intercontinental. *SCM Soporte Intercontinental.* [Online] [Cited: enero 11, 2013.] http://www.scmsi.es/scmsi/images/pdf/alas-pacs.pdf.

6. **Ionworx.** Ionworx.com. [Online] [Cited: Enero 11, 2013.] http://www.ionworx.com/serialshield.html.

7. **Jamrich Parsons, June and Oja, Dan .** *Conceptos de Computación, nuevas perspectivas.* México : s.n., 2008. p. 142. 13:978-970-686-834-3.

8. **Jamrich Parsons, June and Oja, Dan.** *Conceptos de Computacion, nuevas perspectivas.* México : s.n., 2008. p. 143. 13:978-970-686-834-3.

9. **Jamrich Parsons, June and Oja, Dan .** *Conceptos de Computacion, nuevas perspectivas.* México : s.n., 2008. p. 145. 13:978-970-686-834-3.

10. **Jamrich Parsons, June and Oja, Dan.** *Conceptos de Computacion, nuevas perspectivas.* México : s.n., 2008. p. 146. 13:978-970-686-834-3.

11. NetSupport DNA. *NetSupport DNA.* [Online] [Cited: enero 9, 2013.] http://www.netsupportdna.com/es/index.asp.

12. SafeNet. *SafeNet.* [Online] [Cited: enero 10, 2013.] http://www.safenet-inc.com/software-monetization/sentinel-rms/.

13. **Mirage Computer Systems GmbH .** Mirage Computer Systems. *Mirage Computer Systems.* [Online] [Cited: enero 10, 2013.] http://www.mirage-systems.de/products/licence-protector/.

14. Sciensoft. *Sciensoft.* [Online] [Cited: enero 10, 2013.] http://www.sciensoft.com/products/eleckey/eleckey-integrator/.

15. SafeNet. *SafeNet.* [Online] [Cited: enero 10, 2013.] http://www.safenet-inc.es/Products/Detail.aspx?id=2147483967&terms=hasp+srm&LangType=1034.

16. Dotcom Software Solutions Ltd. *Dotcom Software Solutions Ltd.* [Online] 2010. [Cited: enero 10, 2013.] http://www.dotcomsoftwaresolutions.com/es/Productos/Licence-Master.aspx.

17. **Ionworx .** Ionworx.com. [Online] [Cited: Enero 10, 2013.] http://www.ionworx.com/machineid.html.

18. **Fernández Orozco, Adrián .** *Guía para el licenciamiento del sistema alas PACSViewer.* CESIM, UCI. La Habana : CESIM-UCI, 2012. pp. 4-8.

19. **Microsoft.** msdn.microsoft.com. [Online] 2012. [Cited: Enero 10, 2013.] http://msdn.microsoft.com/es-es/library/ms171868.aspx.

20. —. msdn. *msdn.* [Online] 2012. [Cited: enero 11, 2013.] http://msdn.microsoft.com/es-es/library/z1zx9t92.aspx.

21. **Erlandsen, Matthias.** Guioteca. *Guioteca.* [Online] abril 17, 2012. [Cited: mayo 14, 2013.] http://www.guioteca.com/internet/%C2%BFque-es-html5-y-que-cambios-introduce/.

22. **Universidad de Malaga.** Lenguajes y Ciencias de la Computación. *Lenguajes y Ciencias de la Computación.* [Online] [Cited: mayo 14, 2013.] http://www.lcc.uma.es/~eat/services/html-js/manual14.html.

23. **Microsoft.com.** Microsoft.com. [Online] [Cited: Enero 10, 2013.] http://www.microsoft.com/visualstudio/esn/products/visual-studio-express-for-web#product-express-web.

24. **Sparxsystem.** Sparxsystem. [Online] [Cited: Junio 3, 2012.] http://www.sparxsystems.com.ar/products/ea/ultimate.html.

25. **Postgresql.org.** Postgresql.org. [Online] [Cited: Enero 10, 2013.] http://www.postgresql.org/docs/9.1/static/release-9-1-7.html.

26. **Rafael Martinez.** Postgresql.org. [Online] Septiembre 12, 2012. [Cited: Enero 10, 2013.] http://www.postgresql.org.es/node/655.

27. **Microsoft.** ASP.NET. *ASP.NET.* [Online] [Cited: mayo 15, 2013.] http://www.asp.net/mvc/mvc4.

28. maestros del web. *maestros del web.* [Online] marzo 23, 2001. [Cited: febrero 6, 2013.] http://www.maestrosdelweb.com/editorial/cgiintro/.

29. **Zorrilla Castro, Unai, Hernández, Octavio and Quintás, Eduardo.** ADO.NET Entity Framework - Aplicaciones y servicios centrados en datos. *ADO.NET Entity Framework - Aplicaciones y servicios centrados en datos.* [Online] [Cited: mayo 14, 2013.] http://www.campusmvp.com/libros/indices/indice_libro_entity_framework_krasis_press.pdf.

30. **Murphey, Rebecca.** Fundamentos de jQuery. *Fundamentos de jQuery.* [Online] Abril 2013. [Cited: Junio 9, 2013.] http://librojquery.com/.

31. Vates. *Vates.* [Online] [Cited: febrero 6, 2013.] http://www.vates.com/cmmi/que-es-cmmi.html.

32. **Gracia, Joaquin.** Ingenieros de Software. [Online] Agosto 14, 2005. [Cited: Junio 4, 2012.] http://www.ingenierosoftware.com/calidad/cmm-cmmi.php.

33. DocIRS. *DocIRS.* [Online] [Cited: febrero 6, 2013.] http://www.docirs.cl/uml.htm.

34. CEUR Workshop Proceddings. [Online] [Cited: Junio 4, 2012.] ceur-ws.org/Vol-558/Art_12.pdf.

35. **Sparx Systems.** Sparx Systems. *Sparx Systems.* [Online] [Cited: marzo 29, 2013.] http://www.sparxsystems.com.ar/resources/tutorial/business_process_model.html.

36. **Kraffczyk, Joaquín Federico Fuentes.** Colección de Tesis Digitales Universidad de las Américas Puebla. *Colección de Tesis Digitales Universidad de las Américas Puebla.* [Online] [Cited: abril 15, 3013.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo2.pdf.

37. **Gil, Manuel Torres.** Universidad de Almería. *Universidad de Almería.* [Online] [Cited: abril 15, 2013.] http://indalog.ual.es/mtorres/LP/FundamentosDiseno.pdf.

38. **Sebastián, Juan.** ComuSoft.com Seguridad de la Información, Software y Tecnlogía. *ComuSoft.com Seguridad de la Información, Software y Tecnlogía.* [Online] noviembre 13, 2010. [Cited: abril 15, 2013.] http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas.

39. **Koval, Santiago.** Micropixel . *Micropixel .* [Online] agosto 23, 2009. [Cited: abril 15, 2013.] http://www.micropixel.com.ar/secciones/blog/web/arquitectura-cliente-servidor/33.

40. **Tarjuccino.** Tarjuccino. *Tarjuccino.* [Online] [Cited: abril 15, 2013.] http://tarjuccino.com/tutoriales/programacion-web/introduccion-a-json/.

41. **Microsoft.** msdn. *msdn.* [Online] [Cited: abril 15, 2013.] http://msdn.microsoft.com/es-es/library/bb399567.aspx.

42. —. msdn. *msdn.* [Online] [Cited: abril 18, 2013.] http://msdn.microsoft.com/es-es/library/dd409390.aspx.

43. **Alonso, Fernando, Martínez, Loic and Segovia, Francisco Javier.** *Introducción a la ingeniería del software. Modelos de desarrollo de programas.* Madrid : s.n., 2005. 84-96477-00-2.

BIBLIOGRAFÍA

Alonso, Fernando, Martínez, Loic y Segovia, Francisco Javier. 2005. *Introducción a la ingeniería del software. Modelos de desarrollo de programas.* Madrid : s.n., 2005. 84-96477-00-2.

Arias, José R. Álvarez y Manuel. 2002. Universidad Nacional de Educacion a Distancia. [En línea] 2002. [Citado el: 4 de Junio de 2012.] http://www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node60.html.

CEUR Workshop Proceddings. [En línea] [Citado el: 4 de Junio de 2012.] ceur-ws.org/Vol-558/Art_12.pdf.

Cornejo, José Enrique González. DocIRS. [En línea] [Citado el: 4 de Junio de 2012.] http://www.docirs.cl/uml.htm.

DocIRS. *DocIRS.* [En línea] [Citado el: 6 de febrero de 2013.] http://www.docirs.cl/uml.htm.

2010. Dotcom Software Solutions Ltd. *Dotcom Software Solutions Ltd.* [En línea] 2010. [Citado el: 10 de enero de 2013.] http://www.dotcomsoftwaresolutions.com/es/Productos/Licence-Master.aspx.

elicenser.net. elicenser.net. [En línea] [Citado el: 10 de Enero de 2013.] http://www.elicenser.net/en/latest_downloads.html.

Erlandsen, Matthias. 2012. Guioteca. *Guioteca.* [En línea] 17 de abril de 2012. [Citado el: 14 de mayo de 2013.] http://www.guioteca.com/internet/%C2%BFque-es-html5-y-que-cambios-introduce/.

Fernandez, Carlos Alberto. 2000. Universidad Tecnogica de la Mixteca. [En línea] 26 de Octubre de 2000. [Citado el: 4 de febrero de 2013.] http://nuyoo.utm.mx/~caff/doc/El%20Proceso%20Unificado%20Rational.pdf.

Fernández Orozco, Adrián . 2012. *Guía para el licenciamiento del sistema alas PACSViewer.* CESIM, UCI. La Habana : CESIM-UCI, 2012. págs. 4-8.

Filecluster.es. Filecluster.es. [En línea] [Citado el: 10 de Enero de 2013.] http://www.filecluster.es/programas/EMS-SQL-Manager-for-PostgreSQL-11901.html.

García Moreno, Rafael . 2007. Vector. *Vector.* [En línea] 28 de marzo de 2007. [Citado el: 10 de enero de 2013.] http://www.vectorsf.com/noticias-y-eventos/articulos-opinion/el-software-libre-el-futuro-del-desarrollo-de-aplicaciones.

Gil, Manuel Torres. Universidad de Almería. *Universidad de Almería.* [En línea] [Citado el: 15 de abril de 2013.] http://indalog.ual.es/mtorres/LP/FundamentosDiseno.pdf.

Gracia, Joaquin. 2005. Ingenieros de Software. [En línea] 14 de Agosto de 2005. [Citado el: 4 de Junio de 2012.] http://www.ingenierosoftware.com/calidad/cmm-cmmi.php.

Intelectual, Organización Mundial de la Propiedad. OMPI. *OMPI.* [En línea] [Citado el: 10 de enero de 2013.] http://www.wipo.int/about-ip/es/index.html.

Intercontinental, SCM Soporte. SCM Soporte Intercontinental. *SCM Soporte Intercontinental.* [En línea] [Citado el: 11 de enero de 2013.] http://www.scmsi.es/scmsi/images/pdf/alas-pacs.pdf.

Ionworx . Ionworx.com. [En línea] [Citado el: 10 de Enero de 2013.] http://www.ionworx.com/machineid.html.

Ionworx. Ionworx.com. [En línea] [Citado el: 11 de Enero de 2013.] http://www.ionworx.com/serialshield.html.

Iteraprocces. Iteraprocces. [En línea] [Citado el: 4 de Junio de 2012.] http://diagnosticos.iteraprocess.com/documentos/cmmi.pdf.

Jamrich Parsons, June y Oja, Dan. 2008. *Conceptos de Computacion, nuevas perspectivas.* México : s.n., 2008. pág. 146. 13:978-970-686-834-3.

Jamrich Parsons, June y Oja, Dan . 2008. *Conceptos de Computacion, nuevas perspectivas.* México : s.n., 2008. pág. 145. 13:978-970-686-834-3.

Jamrich Parsons, June y Oja, Dan. 2008. *Conceptos de Computacion, nuevas perspectivas.* México : s.n., 2008. pág. 143. 13:978-970-686-834-3.

Jamrich Parsons, June y Oja, Dan . 2008. *Conceptos de Computación, nuevas perspectivas.* México : s.n., 2008. pág. 142. 13:978-970-686-834-3.

Koval, Santiago. 2009. Micropixel . *Micropixel .* [En línea] 23 de agosto de 2009. [Citado el: 15 de abril de 2013.] http://www.micropixel.com.ar/secciones/blog/web/arquitectura-cliente-servidor/33.

Kraffczyk, Joaquín Federico Fuentes. Colección de Tesis Digitales Universidad de las Américas Puebla. *Colección de Tesis Digitales Universidad de las Américas Puebla.* [En línea] [Citado el: 15 de abril de 3013.] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/fuentes_k_jf/capitulo2.pdf.

2001. maestros del web. *maestros del web.* [En línea] 23 de marzo de 2001. [Citado el: 6 de febrero de 2013.] http://www.maestrosdelweb.com/editorial/cgiintro/.

Microsoft. ASP.NET. *ASP.NET.* [En línea] [Citado el: 15 de mayo de 2013.] http://www.asp.net/mvc/mvc4.

Microsoft . 2008. Microsoft . *Microsoft .* [En línea] 7 de abril de 2008. [Citado el: 10 de enero de 2013.] http://www.microsoft.com/spain/sharedsource/licensingbasics/licensingmodels.mspx.

—. 2012. http://msdn.microsoft.com. [En línea] 2012. [Citado el: 10 de Enero de 2013.] http://msdn.microsoft.com/es-es/library/vstudio/w0x726c2.aspx.

—. msdn. *msdn.* [En línea] [Citado el: 7 de febrero de 2013.] http://msdn.microsoft.com/es-es/library/gg416514%28v=vs.108%29.aspx.

—. msdn. *msdn.* [En línea] [Citado el: 15 de abril de 2013.] http://msdn.microsoft.com/es-es/library/bb399567.aspx.

—. msdn. *msdn.* [En línea] [Citado el: 18 de abril de 2013.] http://msdn.microsoft.com/es-es/library/dd409390.aspx.

—. 2012. msdn. *msdn.* [En línea] 2012. [Citado el: 11 de enero de 2013.] http://msdn.microsoft.com/es-es/library/z1zx9t92.aspx.

—. 2012. msdn.microsoft.com. [En línea] 2012. [Citado el: 10 de Enero de 2013.] http://msdn.microsoft.com/es-es/library/ms171868.aspx.

Microsoft.com. Microsoft.com. [En línea] [Citado el: 10 de Enero de 2013.] http://www.microsoft.com/visualstudio/esn/products/visual-studio-express-for-web#product-express-web.

Mirage Computer Systems GmbH . Mirage Computer Systems. *Mirage Computer Systems.* [En línea] [Citado el: 10 de enero de 2013.] http://www.mirage-systems.de/products/licence-protector/.

Mora, Francisco. Universidad de Alicante. [En línea] [Citado el: 4 de Junio de 2012.] http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF.

Murphey, Rebecca. 2013. Fundamentos de jQuery. *Fundamentos de jQuery.* [En línea] Abril de 2013. [Citado el: 9 de Junio de 2013.] http://librojquery.com/.

NetSupport DNA. *NetSupport DNA.* [En línea] [Citado el: 9 de enero de 2013.] http://www.netsupportdna.com/es/index.asp.

Piattini , Mario G. 1996. *Análisis y diseño detallado de aplicaciones informáticas de gestión.* Madrid : RA - MA Editorial, 1996.

Postgresql.org. Postgresql.org. [En línea] [Citado el: 10 de Enero de 2013.] http://www.postgresql.org/docs/9.1/static/release-9-1-7.html.

PROPUESTA DE APLICACIÓN PARA EL REGISTRO DE ESTUDIOS IMAGENOLÓGICOS DE MODALIDADES NO DICOM. **Rivero Castro, Arelys y Hernández Noguera, Alejandro. 2012.** 2, Ciudad de La Habana : Revista Cubana de Informática Médica, 2012.

Rafael Martinez. 2012. Postgresql.org. [En línea] 12 de Septiembre de 2012. [Citado el: 10 de Enero de 2013.] http://www.postgresql.org.es/node/655.

Ramón, Gómez Labrador. 2005. *TIPOS DE LICENCIAS DE SOFTWARE.* 2005.

SafeNet. *SafeNet.* [En línea] [Citado el: 10 de enero de 2013.] http://www.safenet-inc.com/software-monetization/sentinel-rms/.

SafeNet. *SafeNet.* [En línea] [Citado el: 10 de enero de 2013.] http://www.safenet-inc.es/Products/Detail.aspx?id=2147483967&terms=hasp+srm&LangType=1034.

Sciensoft. *Sciensoft.* [En línea] [Citado el: 10 de enero de 2013.] http://www.sciensoft.com/products/eleckey/eleckey-integrator/.

Sebastián, Juan. 2010. ComuSoft.com Seguridad de la Información, Software y Tecnlogía. *ComuSoft.com Seguridad de la Información, Software y Tecnlogía.* [En línea] 13 de noviembre de 2010. [Citado el: 15 de abril de 2013.] http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas.

Sparx Systems. Sparx Systems. *Sparx Systems.* [En línea] [Citado el: 29 de marzo de 2013.] http://www.sparxsystems.com.ar/resources/tutorial/business_process_model.html.

Sparxsystem. Sparxsystem. [En línea] [Citado el: 3 de Junio de 2012.] http://www.sparxsystems.com.ar/products/ea/ultimate.html.

SQL Manager.net. SQL Manager.net. [En línea] [Citado el: 10 de Enero de 2013.] http://www.sqlmanager.net/en/products/postgresql/manager.

Tarjuccino. Tarjuccino. *Tarjuccino.* [En línea] [Citado el: 15 de abril de 2013.] http://tarjuccino.com/tutoriales/programacion-web/introduccion-a-json/.

Tortoisesvn. Tortoisesvn. [En línea] [Citado el: 3 de Junio de 2012.] http://tortoisesvn.net/docs/release/TortoiseSVN_es/tsvn-introduction.html.

Universidad de Malaga. Lenguajes y Ciencias de la Computación. *Lenguajes y Ciencias de la Computación.* [En línea] [Citado el: 14 de mayo de 2013.] http://www.lcc.uma.es/~eat/services/html-js/manual14.html.

Vates. *Vates.* [En línea] [Citado el: 6 de febrero de 2013.] http://www.vates.com/cmmi/que-es-cmmi.html.

Zorrilla Castro, Unai, Hernández, Octavio y Quintás, Eduardo. ADO.NET Entity Framework - Aplicaciones y servicios centrados en datos. *ADO.NET Entity Framework - Aplicaciones y servicios centrados en datos.* [En línea] [Citado el: 14 de mayo de 2013.] http://www.campusmvp.com/libros/indices/indice_libro_entity_framework_krasis_press.pdf.

ANEXOS

Anexo 1. Descripción del proceso Otorgar licencias de software

1. <u>Crear registro de datos de hardware y licencias de las PCs:</u> el Especialista de Despliegue crea el registro de datos de hardware y licencias para la institución donde se realizará el despliegue del sistema alas PACSViewer.

Precedente: No aplicable

Responsable: Especialista en despliegue.

Entradas: No aplicable

Salidas: Registro de datos de hardware y licencias de las PCs.

2. <u>Instalar el sistema alas PACSViewer en la PC:</u> el Especialista de Despliegue instala el sistema alas PACSViewer en las PCs que hayan sido designadas para ubicar el software.

Precedente: 1

Responsable: Especialista en despliegue.

Entradas: No aplicable

Salidas: No aplicable

3. <u>Registrar datos de hardware de la PC:</u> el Especialista de Despliegue registra en el documento Registro de datos de hardware y licencias de las PCs los datos de la PC.

Precedente: 2

Responsable: Especialista en despliegue.

Entradas: No aplicable

Salidas: Registro de datos de hardware y licencias de las PCs.

4. <u>Entregar registro de datos de hardware y licencias de las PCs:</u> el Especialista en Despliegue al llegar hasta el Jefe de Proyecto le entrega el Registro de datos de hardware y licencias de las PCs al mismo.

Precedente: 3

Responsable: Especialista en despliegue.

Entradas: No aplicable.

Salidas: Registro de datos de hardware y licencias de las PCs.

5. Generar licencias de software: el Jefe de Proyecto genera las licencias asociadas a cada PC y se incluyen en el Registro de datos de hardware y licencias de las PCs según corresponda.

Precedente: 4

Responsable: Jefe de proyecto.

Entradas: No aplicable.

Salidas: Registro de datos de hardware y licencias de las PCs.

6. Entregar registro de datos de hardware y licencias de las PCs: el Jefe de Proyecto por la vía que sea más efectiva en ese momento, ya sea por correo o presentándose él mismo en la institución hospitalaria donde fueron recogidas las solicitudes, entrega el registro de datos de hardware y licencias de las PCs al Administrador de la institución.

Precedente: 5

Responsable: Jefe de proyecto.

Entradas: No aplicable.

Salidas: Registro de datos de hardware y licencias de las PCs.

7. Guardar licencias: el Administrador de la institución tras recibir el registro de datos de hardware y licencias de las PCs conserva estas licencias, las cuales pueden ser reutilizables en caso de existir algún problema con algunas de las PC.

Precedente: 6

Responsable: Administrador de la institución.

Entradas: No aplicable.

Salidas: Registro de datos de hardware y licencias de las PCs.

8. Activar sistema: si el registro de datos de hardware y licencias de las PCs contiene la licencia correspondiente a la PC que se desea instalar el Administrador de la Institución activa el sistema en la misma.

Precedente: 7

Responsable: Administrador de la institución.

Entradas: No aplicable.

Salidas: No aplicable.

9. Solicitar licencias: si existe alguna Pc sin activar el Administrador de la institución hace una nueva solicitud de licencia comenzando el proceso desde la recogida correcta de los datos de hardware de la PC a instalar.

Precedente: 8

Responsable: Administrador de la institución.

Entradas: No aplicable.

Salidas: No aplicable.

Anexo 2 Descripción textual del caso de uso arquitectónicamente significativo Gestionar perfiles de aplicación

Tabla 24: CU Gestionar perfiles de aplicación.

Objetivo	Insertar un nuevo perfil de aplicación, listar los perfiles existentes en el sistema, buscar los perfiles existentes y eliminar perfiles del sistema.
Actores	Administrador del Sistema (Inicia)
Resumen	El caso de uso se inicia cuando el Administrador del Sistema accede a la opción Administración del menú de opciones de la página principal, en el submenú Perfiles el sistema brinda la posibilidad de Adicionar perfil o Listar perfiles, opción que permite Buscar y Eliminar un perfil. El caso de uso termina una vez que se haya realizado una o varias de las operaciones anteriores.
Complejidad	Media
Prioridad	Crítico
Referencias	RF 5, RF 5.1, RF 5.2, RF 5.3, RF 5.4
Precondiciones	El Administrador del Sistema ha sido autenticado en el sistema.
Postcondiciones	Se realizó la inserción, listado, búsqueda y/o eliminación de un perfil.
Flujo de eventos	
Flujo básico “Gestionar perfiles de aplicación”	

1. El **Administrador del Sistema** accede al submenú “Perfiles” del menú “Administración” ubicado en la página principal.
2. El **Administrador del Sistema** selecciona la opción que desee realizar:
   - Si selecciona “Adicionar perfil” ir a la Sección 1 “Adicionar perfil”.
   - Si selecciona “Listar perfiles” ir a la Sección 2 “Listar perfiles”.
3. Termina el caso de uso.

Sección 1 “Adicionar perfil”

1. El **Administrador de Sistema** selecciona la opción “Adicionar perfil”
2. El **Sistema** muestra una pantalla permitiendo introducir los datos del perfil:
   - Nombre de Aplicación
   - Llave de Aplicación

   y dando la posibilidad de:
   - “Aceptar”, donde se procedería a insertar el perfil en el sistema.
   - “Cancelar”.
3. El **Administrador del Sistema** introduce los datos del perfil :
   - Nombre de Aplicación
   - Llave de Aplicación

   y presiona el botón “Aceptar”.
4. El **Sistema** valida los datos introducidos
5. El **Sistema** verifica que no exista otro perfil con el mismo nombre de aplicación.
6. El **Sistema** registra el perfil, mostrando el mensaje de información *“Se insertó correctamente el perfil.”*
7. Termina el caso de uso.

Flujos alternos

2a. Cancelar operación de adicionar.

1. El **Administrador del Sistema** presiona el botón “Cancelar”.
2. El **Sistema** regresa a la página principal
3. Termina el caso de uso

4a.Existen campos vacíos.

1. El sistema muestra el mensaje de error *“Existen campos vacíos”.*

5a. Existe otro perfil con el mismo nombre de aplicación

1. El sistema muestra el mensaje de error *"Este perfil ya existe en el sistema."*

Sección 2 "Listar Perfiles"

1. El **Administrador de Sistema** selecciona la opción "Listar perfil"
2. El **Sistema** muestra una pantalla con un listado de los perfiles existentes con los datos:
    - Nombre de Aplicación
    - Llave de Aplicación

   Ofreciendo la posibilidad de listar el perfil que se desee mediante un paginado de 10,20 o 30 elementos por página, permitiendo de esta forma navegar por el listado de perfiles obtenido.
3. El Sistema brinda la posibilidad de buscar y/o eliminar un perfil específico, donde si desea:
    - "Eliminar" ir a la Sección 3 "Eliminar perfil".
    - "Buscar" ir a la Sección 4 "Buscar perfil".

Sección 3 "Eliminar perfil"

1. El **Administrador del Sistema** selecciona el perfil que desea eliminar del listado de perfiles y presiona el botón "Eliminar".
2. El **Sistema** muestra el mensaje de advertencia *"¿Desea eliminar los registros seleccionados? "*, permitiendo:
    - "Eliminar", donde se procederá a la eliminación del perfil.
    - "Cancelar".
3. El **Sistema** elimina el perfil seleccionado.
4. Termina el caso de uso.

Flujos alternos

2a. El Administrador del Sistema selecciona la opción "Cancelar".

1. El **Administrador del Sistema** presiona el botón "Cancelar".
2. El **Sistema** cancela la operación y regresa a la pantalla anterior.
3. Termina el caso de uso.

Sección 3 "Buscar perfil"

1. El **Administrador del Sistema** presiona el botón Buscar y selecciona el parámetro por el cuál desea buscar el perfil, introduce la información y presiona el botón "Buscar".

2. El **Sistema** muestra una pantalla con los datos del perfil encontrado: • Nombre de Aplicación • Llave de Aplicación 3. Termina el caso de uso.		
Flujos alternos		
2a. El perfil no existe		
1. El **Sistema** muestra la pantalla en blanco. 2. Termina el caso de uso.		
Relaciones	**CU Incluidos**	No aplicable.
	CU Extendidos	No aplicable.
Requisitos no funcionales	No aplicable.	
Asuntos pendientes	No aplicable.	