

Universidad de las Ciencias Informáticas
Facultad 7



**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

**Diseño de una herramienta para la gestión
del otorgamiento de licencias para
el sistema alas PACSViewer**

Autoras: Danairy Averoff Cabrera
María Luisa Enriquez Pérez

Tutores: Ing. Nuvia Angélica Estévez Rojas
Ing. Jorge Carlos Yero García

Ciudad de la Habana, Junio del 2010

“Año 52 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los - días del mes de junio del año 2010.

Autoras:

Danairy Averoff Cabrera

Firma de la Autora

María Luisa Enriquez Pérez

Firma de la Autora

Tutores:

Ing. Nuvia Angélica Estevez Rojas

Firma de la Tutora

Ing. Jorge Carlos Yero García

Firma del Tutor

DATOS DE CONTACTO

TUTORA: Ing. Nuvia Angélica Estevez Rojas

Ingeniera en Ciencias Informáticas. Graduada en la Universidad de las Ciencias Informáticas en el año 2007. Posee la categoría docente de Profesor Instructor. Actualmente es profesora de Física I, ha impartido también Física II en la facultad 7. Se desempeña como Analista en el proyecto de Planificación Quirúrgica del Área Temática, Software Médico Imagenológico. Correo electrónico: nestevez@uci.cu.

TUTOR: Ing. Jorge Carlos Yero García

Instructor recién graduado en el año 2007 de Ingeniero Industrial en la Universidad de Holguín "Oscar Lucero Moya". Profesor de la Facultad # 7. Ha impartido las asignaturas de Comercio Electrónico, Matemática I y II. Se ha desempeñado como Planificador General del área temática de GPI, jefe de proyecto de Despliegue, Soporte y Capacitación y Comercial en el Centro de Informática Médica (CESIM). Correo electrónico: jyero@uci.cu.

RESUMEN

Los sistemas de gestión del otorgamiento de licencias de software, han alcanzado un auge significativo gracias a la tecnología y a su vez a la necesidad de proteger el software que se desarrolla y se utiliza. Aunque en muchos países los sistemas de gestión del otorgamiento de licencias de software son muy comunes, Cuba debido a su difícil situación económica no puede acceder a estos pues son en su mayoría sistemas propietarios.

Debido a esta situación en la Universidad de las Ciencias Informáticas se desarrolló un sistema para la gestión del otorgamiento de licencias para el sistema alas PACSViewer para llevar el control de este producto por parte del Departamento de Software Médico Imagenológico. En la actualidad, se ha comprobado que el proceso de gestión del otorgamiento de licencias para el sistema alas PACSViewer no se lleva a cabo de la forma más eficiente.

La presente investigación, tiene como objetivo diseñar dos módulos encargados de mejorar la gestión del otorgamiento de licencias para el sistema alas PACSViewer. Con este fin, se propone la utilización del lenguaje C# y el marco de trabajo Framework .Net de Microsoft.

Como resultado de la investigación se obtuvo el diseño de un sistema con vistas a su futura implementación, el cual permitirá mejorar la forma en que se lleva a cabo actualmente el proceso de gestión del otorgamiento de licencias para el sistema alas PACSViewer, garantizando así que el proceso se realice de forma segura y menos trabajosa.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Tendencias actuales en el licenciamiento del software.....	4
1.1.1 Soluciones basadas en hardware.....	4
1.1.2 Soluciones basadas en software.	5
1.2 Estado del arte.....	5
1.3 Herramientas y Tecnologías.....	7
1.3.1 Lenguajes de Programación.....	7
1.3.2 Herramientas CASE.....	8
1.3.3 IDEs de Desarrollo.....	8
1.3.4 Sistemas de Gestión de Base de Datos (SGBD).....	9
1.4 CMMI (Capability Maturity Model Integration).....	10
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	13
2.1 Descripción del sistema propuesto.....	13
2.2 Información que se maneja.....	13
2.3 Modelo de Procesos del Negocio.....	13
2.3.1 Procesos del negocio actual.....	14
2.3.2 Proceso del Negocio Mejorado.....	14
2.4 Especificación de los requerimientos del software.....	17
2.4.1 Requerimientos Funcionales del Sistema.....	17
2.4.2 Requerimientos No Funcionales del Sistema.....	20
2.5 Definición de los actores del sistema.....	22
2.6 Diagrama de casos de uso del sistema.....	22
2.7 Descripción de los casos de uso del sistema.....	23
2.7.1 Listado de los casos de uso del Cliente.....	23
2.7.2 Listado de los casos de uso del Gestor.....	25
2.8 Casos de uso arquitectónicamente significativos.....	26
2.8.1 Diagrama de Caso de Uso Significativos del Cliente.....	26
2.8.2 Diagrama de Caso de Uso Significativos del Gestor.....	27
CAPÍTULO 3: ARQUITECTURA Y DISEÑO	28
3.1 Arquitectura.....	28
3.2 Modelo de Análisis.....	29
3.2.1 Diagramas del Cliente.....	29
3.2.2 Diagramas del Gestor.....	31

3.3 Diseño.....	33
3.3.1 Diagramas del Cliente.....	33
3.3.2 Diagramas del Gestor.....	39
3.3.3 Descripción de las Clases.....	44
3.3.4 Diagrama de Clases del Diseño.....	52
3.4 Modelo de datos.....	53
3.4.1 Descripción de las Tablas.....	54
3.5 Prototipo no Funcional.....	55
CONCLUSIONES.....	57
RECOMENDACIONES.....	58
REFERENCIAS BIBLIOGRÁFICAS.....	59
BIBLIOGRAFÍA.....	61
ANEXOS.....	63
ANEXO I. Descripción del proceso Gestión del otorgamiento de licencias del software.....	63
ANEXO II. Descripción ampliada de los Casos de Uso del Sistema.....	66
II.1 Casos de Uso Expandidos del Cliente.....	67
II.2 Casos de Uso Expandidos del Gestor.....	71

INTRODUCCIÓN

En las últimas décadas la industria del software ha sido uno de los sectores que más auge ha alcanzado debido a su dinamismo y rápido crecimiento. Estas características han traído consigo que las últimas tendencias en la industria indiquen que el licenciamiento del software que se desarrolla tenga gran importancia, ya que el uso no autorizado del mismo puede provocar grandes pérdidas en la industria.

El desarrollo de software supone un esfuerzo de equipo que combina las ideas creativas y los talentos de programadores, documentadores y diseñadores gráficos. Éste al igual que la mayoría de las obras creativas, tales como libros, composiciones musicales y películas, el software informático está protegido por las leyes de propiedad intelectual. Cuando se compra un software, se está comprando el derecho de utilizar el software conforme a ciertas restricciones impuestas por el titular del derecho de autor, normalmente el editor del software. Las normas exactas se describen en la documentación que acompaña al software: la licencia. (1)

La licencia de software es una especie de contrato, donde se especifican todas las normas y cláusulas que rigen el uso de un determinado programa, principalmente se estipulan los alcances de uso, instalación, modificación, reproducción y copia de estos productos. (2)

Las licencias son el principio legal del software una vez que éste se desarrolla y se utiliza. Mediante ellas se protege el software de los usos atrevidos u oportunistas y son además una forma eficaz de proteger los gastos que se realizan.

Dentro de los elementos objetivos de una licencia de software se encuentran, el precio de la licencia, que no es más que el valor que debe pagar el licenciataria al licenciante por el concepto de la cesión de derechos establecidos en la licencia, y el plazo de la licencia, que es el tiempo de duración en que se mantiene vigente las condiciones y los términos declarados en la licencia.

Las empresas llevan el control a la hora de asignar las licencias de sus productos y la activación de los mismos debido a la gran importancia que requiere este proceso, pues es una forma de garantizar que los usuarios finales se comprometan a no proporcionarle el software a terceros y a la no reinstalación del programa en equipos distintos al que se instaló originalmente.

Además, de esta forma evitan el peligro que representa la piratería, que se ha convertido en uno de los principales riesgos en la industria. Esto posibilita una mejora en la seguridad de los productos y así mantener la confianza del cliente y del mercado.

El sistema PACS tiene incluido tres componentes: el servidor PACSServer como el servidor conectado a los equipos generadores de imágenes, el servidor PACSDicoMail para la transmisión inter e intrahospitalaria y el sistema PACSViewer como visor de imágenes.

En estos momentos el CESIM, tiene establecido un procedimiento basado en llaves de software que utiliza datos del hardware para la protección y licenciamiento del sistema PACSViewer, para protegerse contra el uso no autorizado del mismo. Cada instalación de este producto debe ser activada usando un mecanismo que emplea datos del hardware. Éste consiste en extraer un grupo de datos que son únicos para la PC en la que se va instalar el sistema, debido a que están asociados a los componentes de hardware de dicha PC. A partir de estos datos se aplica un procedimiento para obtener una clave única que permite usar el sistema en una PC exclusiva.

Para implementar este proceso, se creó una herramienta, la cual es llevada por cada desarrollador encargado del despliegue en las instituciones hospitalarias de forma manual, usando dispositivos externos. Estos deben ir por cada PC activando cada una de las instalaciones del sistema.

La información que manejan es muy valiosa y confidencial. Lo anterior, unido a que dicha actividad no la puede llevar a cabo una sola persona, porque sería un trabajo muy engorroso y agotador, hace que haya una gran probabilidad de que ocurra alguna filtración o robo de información de los medios utilizados durante el despliegue del sistema para su licenciamiento o el software en general, lo que derivaría en pérdidas para la Universidad y el país. Como se puede apreciar el proceso es ineficiente y poco fiable.

La situación existente lleva a plantearse el siguiente **Problema Científico**. ¿Cómo facilitar el proceso de gestión para el otorgamiento de licencias del sistema PACSViewer? Para resolver este problema se toma como **Objeto de Estudio** el proceso de gestión del otorgamiento de licencias del software en el Departamento de Software Médico Imagenológico, delimitando como **Campo de Acción** el proceso de otorgamiento de licencias para el sistema PACSViewer.

Para dar solución al problema planteado se propone como **Objetivo General** diseñar dos módulos que permitan la gestión del otorgamiento de licencias para el sistema PACSViewer.

Respondiendo al objetivo planteado se plantean las siguientes tareas de la investigación:

1. Analizar el estado del arte general sobre el tema de protección y otorgamiento de licencias de software.

2. Analizar las herramientas que existen en el mundo para gestionar el otorgamiento de licencias de software.
3. Realizar la especificación de los procesos del negocio asociados al otorgamiento de licencias para el software médico alas PACSViewer según el procedimiento establecido por el Departamento de Software Médico Imagenológico del CESIM.
4. Realizar la especificación de los requerimientos de software.
5. Realizar el análisis de los módulos Cliente y Gestor para el otorgamiento de licencias.
6. Diseñar los módulos Cliente y Gestor para el otorgamiento de licencias.
7. Diseñar la Base de Datos correspondiente al Gestor.

El documento consta de tres capítulos:

Capítulo 1: Fundamentación Teórica: Describe características sobre las tendencias actuales en el licenciamiento por software y llaves de hardware, así como una referencia al estado del arte en que se encuentran los sistemas en el mundo encargados de la gestión del otorgamiento de licencias del software. Se realiza un análisis de las herramientas que se van a utilizar para construir la solución propuesta, se analiza el modelo CMMI (Capability Maturity Model Integration) para la mejora y evaluación de los procesos del sistema.

Capítulo 2: Características del Sistema: Para guiar el desarrollo del sistema se describe el sistema propuesto, se enumeran los requerimientos funcionales y no funcionales, agrupando los primeros en Casos de Uso, así como las descripciones textuales de estos.

Capítulo 3: Arquitectura y Diseño: Muestra los resultados obtenidos en el desarrollo de los procesos de análisis y diseño del sistema, así como los diagramas que fueron necesarios para obtener una mayor claridad a la hora de elaborar la solución que se propone.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se describen características sobre el licenciamiento y protección del software a través de llaves de hardware y del software. Además se hace una referencia al estado del arte en que se encuentran las herramientas en el mundo encargadas de la gestión del otorgamiento de licencias del software. Luego se hace un análisis sobre las principales herramientas y tecnologías a utilizar para construir la solución propuesta. Se analiza el modelo CMMI (Capability Maturity Model Integration) para la mejora y evaluación de los procesos del sistema.

1.1 Tendencias actuales en el licenciamiento del software.

Además de proteger el software y la propiedad intelectual, se necesita proteger los ingresos de las ventas de los productos. Para el aseguramiento de que el software sólo está disponible para los usuarios adecuados, de acuerdo con los términos que se definan, este proceso se controla a través de las licencias. Las licencias proporcionan la flexibilidad necesaria para implementar las estrategias del negocio para la distribución del software. (3)

Para obtener el máximo beneficio de la estrategia de licencias en una empresa, es necesario un sistema de licencias de software que le proporcione la flexibilidad para adaptar los términos de licencias, de modo que coincidan con la estrategia de negocio y para adaptarse rápidamente a los cambios del mercado y a las necesidades de la empresa. (4)

HASP SRM, simplifica la capacidad de protección y licencias del software mediante la utilización de protección flexible y herramientas de licencias, junto con una llave de protección HASP SRM que protege el software. Estas llaves pueden estar basadas en Hardware o en Software. (5)

1.1.1 Soluciones basadas en hardware

Las soluciones basadas en hardware, proporcionan un dispositivo hardware externo junto con el software. El funcionamiento de su software dependerá de la conexión del dispositivo al ordenador del usuario final. En el tiempo de ejecución, el software se comunica con el dispositivo hardware y sólo funciona correctamente si recibe una respuesta verosímil de éste. HASP SRM proporciona varios dispositivos hardware en forma de llaves HASP HL. (6)

Las llaves HASP HL se distribuyen con el software a los usuarios finales, y se conectan a los equipos de estos. Las llaves HASP HL proporcionan el más alto nivel de seguridad. Para que un usuario acceda a su software, y para que funcione correctamente, se debe acceder a la llave desde la aplicación. Esto significa que la llave puede llevarse de un ordenador a otro. De este modo, el software se puede instalar en múltiples equipos pero sólo se ejecutará si el software relaciona y autentifica la llave. (7)

Capítulo 1: Fundamentación Teórica

1.1.2 Soluciones basadas en software.

Con las soluciones basadas en software, basta con seguir la instalación del software del equipo del usuario final para que la protección y licencias se asocien a ese equipo en concreto. El software sólo funcionará cuando el usuario haya introducido la llave del producto. En el tiempo de ejecución, el servidor comprueba que el software está en el equipo que tiene licencia para ejecutarlo y que se está utilizando de acuerdo con los términos de licencia del usuario. HASP SRM proporciona una solución sólida basada en software mediante el uso de llaves HASP SL. (8)

Después de instalar el software en el equipo, el usuario final normalmente introduce una llave de producto que se envía, ya sea por Internet o por transferencia de archivo, al servidor HASP SRM (Business Studio Server), el cual valida que la llave del producto no se haya utilizado para activar el software en más equipos de los permitidos, y a cambio, el usuario recibe una llave HASP SL con términos de licencia que debe instalar en su ordenador.

1.2 Estado del arte.

En el mundo existen gran cantidad de herramientas encargadas de la gestión del otorgamiento de licencias de software. Algunas de estas herramientas son:

NetSupport DNA

NetSupport DNA es una completa solución que permite realizar inventarios de hardware y de software así como la gestión de licencias. Permite asignar información de licencias a los datos de activos de una PC. Tiene una interfaz muy amigable que permite que los operadores puedan asignar rápidamente licencias a cuantos PCs deseen.

Permite monitorear el uso de licencias dentro de una organización. Al introducir el número de licencias para cada aplicación es posible identificar el uso de ellas cada vez que se ejecute el inventario de software. Además de mostrarse en las listas de Inventario de software, la información de licencia también se puede mostrar en la sección de Medición de aplicaciones de DNA. NetSupport DNA soporta Windows 98 y posteriores. (9)

MSIA (Microsoft Software InventoryAnalyzer)

Microsoft Software Inventory Analyzer (MSIA) explora las estaciones de trabajo individuales o redes completas de PCs para los productos de software de Microsoft. MSIA genera los informes que indican los títulos de software, tipo y número de licencias para cada título, el número de instalaciones para ese título y los vínculos a una lista de PCs exploradas. (10)

Capítulo 1: Fundamentación Teórica

El MSIA podrá funcionar con redes que cuentan con hasta 250 o menos ordenadores. La última versión de MSIA 5.1 tiene características que le permite:

- Actualización de la información de licencias en formato HTML cuando Internet Explorer 8 está instalado y Soporte para Windows 7 y Office 2010.

ElecKey (Scientsoft Research) Versión 2.0

ElecKey es la solución completa para la protección de copia de software, licencias de software y distribución de software de forma segura.. ElecKey proporciona la capacidad de concesión de licencias de software, permitiendo así crear una gran variedad de modelos de licencia, por ejemplo, por un plazo limitado versión de evaluación, licencia flotante, la activación del software, entre otros. (11)

Una vez que se distribuye el software, ElecKey ofrece una amplia variedad de opciones que permiten gestionar la licencia y la concesión de licencias desde un servidor de activación completamente automatizado.

Por otra parte, ElecKey 2.0 también proporciona la solución para hacer frente a las difíciles cuestiones de soporte al usuario final de concesión de licencias. Puede crear las utilidades de usuario final, junto con la aplicación protegida que permitirá al usuario activar, desactivar, transferir y retirar la licencia de software fácil y cómoda. Mediante estos servicios se puede verificar si el usuario ha destruido la licencia antes de la emisión de una restitución. (12)

Protector de Licencias (License Protector) 2.5

Protector de Licencias, administra y controla licencias de software, genera versiones de prueba, proporciona un programa de protección contra copia y soporta pruebas de usuario concurrentes. Posibilita generar licencias automáticamente para tiendas en línea. (13)

Ofrece archivos de licencia cifrados con claves hechas por el cliente para cada proyecto de informática y Claves de Activación Segura que pueden ser usadas sólo una vez. Sin pagos por derechos de uso. Pague una vez - distribúyalo tanto como guste. (14)

Debido a que algunos de estos sistemas son propietarios y por ende sus costos por uso son muy elevados, financiar los gastos de la implantación de un sistema de los existentes en el mundo, para la gestión del otorgamiento de licencias de software sería muy difícil para países subdesarrollados como Cuba.

Todo esto ligado a que otros sistemas no cubren en su totalidad con lo que se quiere diseñar, puesto que algunas monitorean el uso de licencias pero no logran asignarlas, se limitan además a

Capítulo 1: Fundamentación Teórica

chequear la cantidad de instalaciones de cada sistema informático, con el objetivo de verificar el cumplimiento de las licencias, sin embargo no implementan mecanismos para asegurar el acatamiento de éstas.

Algunos de estos sistemas funcionan sólo para productos de empresas específicas, por lo que el Departamento de Software Médico Imagenológico no tendría acceso a las mismas. Es por ello que se ha decidido desarrollar soluciones propias para facilitar el proceso de gestión del otorgamiento de licencias para el sistema alas PACSViewer.

Actualmente en Cuba a pesar de que se han hecho numerosos esfuerzos en el campo de la Industria de Software, esta es relativamente nueva y su desarrollo se ha visto frenado por las consecuencias de un bloqueo económico. Pero a pesar de los contratiempos externos, muchas cosas faltan por hacer aún y una de ellas es llegar a alcanzar experiencia en la protección de software mediante licenciamiento. En la actualidad este proceso se ha visto retrasado, tanto así que no existe un sistema nacional mediante el cual se le otorguen las licencias al software a través de su activación y que lleve el control sobre las mismas, para de esta forma tener una mayor seguridad del uso del software que se distribuye.

1.3 Herramientas y Tecnologías.

1.3.1 Lenguajes de Programación.

Java: Java no es distribuido, sino que proporciona las librerías y herramientas para que los programas puedan ser distribuidos, es decir, que se corran en varias máquinas, interactuando. Potente lenguaje orientado a objetos y multiplataforma. (15) Java tiene varias limitaciones como es la velocidad, los programas hechos en Java no tienden a ser muy rápidos. Los programas de Java son interpretados, nunca alcanzan la velocidad de un verdadero ejecutable.

Visual C++ .NET: Proporciona a los programadores un lenguaje orientado a objetos de probada eficacia para generar aplicaciones eficaces de alto rendimiento. Gracias a plantillas avanzadas, acceso a plataformas de bajo nivel y un compilador que optimiza las compilaciones, Visual C++.NET ofrece funcionalidad superior para generar aplicaciones y componentes sólidos. Visual C++ .NET ofrece a los programadores una herramienta excelente con la que generar software. (16) Tiene un uso de DLLs (librerías dinámicas) muy complejo, no es totalmente abierto, es difícil programar bases de datos.

Microsoft Visual C# .NET 2.0: Lenguaje estrella de la plataforma de desarrollo .NET. Cuenta con una extensa librería base para el desarrollo de todo tipo de aplicaciones. C# es un lenguaje de programación simple pero eficaz, este toma las mejores características de lenguajes

Capítulo 1: Fundamentación Teórica

preexistentes como Visual Basic, Java o C++ y las combina en un solo lenguaje. (17) C# presenta entre otras características:

- Sencillez, Modernidad, Orientación a Objetos, Orientación a componentes, Gestión automática de memoria, Seguridad de tipos, Instrucciones seguras, Sistema de tipos unificado, Extensibilidad de tipos básicos, Extensibilidad de operadores, Extensibilidad de modificadores, Posibilidad de crear versiones, Eficiencia y Compatibilidad.

1.3.2 Herramientas CASE.

CASE (Computer Aided Software Engineering): Ingeniería de software asistida por computadoras.

Rational Suite: Es la única solución integrada y completa para todo el ciclo de vida del desarrollo de Software. Reúne las mejores prácticas, herramientas y servicios. Provee todas las herramientas que precisa un equipo en una única caja. (18) Es bastante cara y poco intuitiva para trabajar.

Enterprise Architect 7.5: Enterprise Architect (EA) es una herramienta para modelar y gestionar información compleja, diseñar y visualizar software, o construir y desplegar diversos sistemas, es flexible, completa y potente de modelado en UML bajo plataforma Windows. Provee lo más nuevo en desarrollo de sistemas, administración de proyectos y análisis de negocio. (19) EA abarca el ciclo de vida completo del desarrollo de software. (20)

La edición corporativa está pensada para grandes equipos de desarrollo. Soporta todas las funcionalidades de las versiones Desktop y Professional, así como también la habilidad de conectarse a SQL Server, MySQL, Oracle 9i and 10g, PostgreSQL, MSDE, Adaptive Server Anywhere y MS Access backends como repositorio compartido. (21)

Visual Paradigm: Se caracteriza por ser una Herramienta Case que soporta las últimas versiones de UML (Lenguaje de Modelado Unificado) y la Notación y Modelado de Procesos de Negocios. Desde un Grupo Administrador de Objetos. (22) Tiene problemas de integración con otras herramientas de desarrollo.

1.3.3 IDEs de Desarrollo.

Microsoft Visual Studio 2008: Es el nuevo Entorno Integrado de Desarrollo (también conocido como IDE por sus siglas en inglés, Integrated Development Environment), que Microsoft ha desarrollado para la creación de Software enfocado a su Sistema Operativo Microsoft Windows Vista y realizar aplicaciones para trabajar con su paquetería Microsoft Office 2007. Ha sido desarrollado específicamente para la plataforma .NET y en especial para el lenguaje C#. (23)

Capítulo 1: Fundamentación Teórica

Mejoras y funcionalidades de Microsoft Visual Studio 2008:

- Utilizar el Framework .NET 3.5 y poder programar para las versiones anteriores (2.0, 3.0).
- Conjunción con XAML (eXtensible Application Markup Language).
- Un diseñador para Windows Presentation Foundation y Workflow Foundation que son parte del Framework .NET 3.0.
- IntelliSense para JavaScript.
- El nuevo Lenguaje LINQ (Language Integrated Query) siendo éste un agregado a los lenguajes Visual Basic y Visual C# para la realización de consultas SQL.

SharpDevelop: Es un entorno de desarrollo integrado libre para desarrollar aplicaciones en distintos lenguajes de programación, soporta completamente el lenguaje C# 2.0, no está concebido para aplicaciones de gran envergadura y carece de integración con muchas tecnologías asociadas a la plataforma .NET. (24)

1.3.4 Sistemas de Gestión de Base de Datos (SGBD).

Microsoft SQL Server: Diseñada desde su inicio para trabajar en entornos Internet e Intranet, Microsoft SQL Server integra los nuevos desarrollos para estos entornos específicos con los desarrollos heredados de aplicaciones "tradicionales", tiene una plataforma de desarrollo fácil y abierta, integrada con las mejores tecnologías de Internet, mínimo coste de Propiedad, y la potencia de sus herramientas de gestión y el menor coste de toda la industria para entornos Internet, hacen de Microsoft SQL Server la mejor opción con el menor coste. (25)

Tiene como desventajas el costo de las licencias comparadas con otros competidores, los costos varían desde Gratis (Express y Compact Edition), hasta \$ 25 000 USD (Enterprise Edition). Estos costos dependen de los servidores a utilizar y el número de clientes (usuarios).

MySQL: Tiene un diseño multihilo que le permite soportar una gran carga de forma muy eficiente. Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL. Este gestor de bases de datos es el más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. (26)

Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (27) Un gran porcentaje de las utilidades de MySQL no están documentadas.

Capítulo 1: Fundamentación Teórica

PostgreSQL8.3: Es la primera base de datos de código abierto en implementar Recorrido Sincronizado, que reduce el uso de E/S en aplicaciones de minería de datos. El grupo de Windows ha implementado un sistema de compilación con Visual C++, lo cual mejora la estabilidad y rendimiento en Windows, así como la accesibilidad para otros contribuyentes Windows. Nuevas opciones de registro (logging) han sido agregadas y el sobrecosto del recolector de estadísticas ha sido disminuido para hacer más fácil el monitoreo de sus servidores. Entrega mayor consistencia en el rendimiento que versiones anteriores. (28)

Las mejoras de rendimiento más importantes incluyen:

- Heap Organized Tuples (HOT), que eliminan hasta un 75% de la sobrecarga de mantenimiento en tablas frecuentemente actualizadas.
- Checkpoints extendidos y autoafinamiento del escritor en segundo plano, que reducen el impacto de los checkpoints en los tiempos de respuesta.
- Opción de confirmación (commit) asíncrono de transacciones, que permite tiempos de respuesta más breves para algunas transacciones.

Estos cambios, aceleran significativamente la tasa de procesamiento de transacciones.

1.4 CMMI (Capability Maturity Model Integration).

CMMI (Capability Maturity Model Integration) fue creado a partir de una solicitud del Departamento de defensa de EUA, es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Este modelo representa un camino de perfeccionamiento, y permite determinar la madurez y evaluar las capacidades de las organizaciones que desarrollan software. Es una colección estructurada de elementos, que describe características de procesos que han demostrado, por experiencia, ser exitosos.

Cada enfoque del modelo CMMI soporta una representación:

- El enfoque de capacidad del proceso soporta la representación continua.
- El enfoque de madurez de la organización soporta la representación escalonada.

La representación continua de una organización mostrará la representación de nivel de capacidad de cada una de las áreas de proceso del modelo. La representación escalonada definirá a la organización, dándole en su conjunto un nivel de madurez del 1 al 5.

- Nivel1 Inicial: Proceso impredecible, control reactivo, Nivel2 Administrado: Administración básica del proyecto, Nivel 3 Definido: Proceso caracterizado por la organización y proactivo, Nivel 4 Cuantitativamente Administrado: Control cuantitativo del proceso, Nivel 5 Optimizado: Mejora continua de proceso.

Capítulo 1: Fundamentación Teórica

El modelo CMMI contiene 22 áreas de proceso comunes a las dos representaciones (continua y escalonada). Las áreas correspondientes al nivel 2 de CMMI son:

- Administración de Requisitos (REQM), Planeación del Proyecto (PP), Monitoreo y Control del Proyecto (PMC), Administración de Acuerdo con Proveedores (SAM), Medición y Análisis (MA), Aseguramiento de la Calidad de Procesos y Productos (PPQA) y Administración de la Configuración(CM)

En la actualidad la está acometiendo un proyecto de mejora de sus procesos basado en el modelo CMMI. El proceso de mejora está encaminado a que la Universidad alcance en el 2010 una certificación internacional del nivel 2 del modelo CMMI. (29)

El desarrollo de la herramienta para la gestión del otorgamiento de licencias de software estará enmarcado fundamentalmente en el área de proceso Administración de Requisitos, la cual tiene el propósito de administrar los requisitos del proyecto e identificar inconsistencias entre dichos requisitos. Esta área de proceso tiene 5 prácticas específicas:

- Obtener el entendimiento de los requisitos: El análisis de los requisitos se hace junto a los proveedores para comprobar que son compatibles y que se comparte el mismo entendimiento.
- Obtener el compromiso a los requisitos: El compromiso debe obtenerse tanto a nivel externo, con el cliente, como a nivel interno, con el equipo de trabajo.
- Gestionar los cambios a los requisitos: Los requisitos cambian por diversas razones, para un análisis efectivo del impacto de los cambios a los requisitos es necesario que sea conocido el origen de cada requisito y este documentado el motivo del cambio.
- Mantener la trazabilidad bidireccional de los requisitos: Establecer la trazabilidad desde la fuente de los requisitos hasta el nivel más bajo de descomposición del requisito y viceversa, permite establecer una evaluación de impacto por los cambios a los requisitos.
- Identificar inconsistencias entre el trabajo del proyecto y los requisitos: Identificar las inconsistencias de los requisitos e iniciar las acciones correctivas para solucionarlos.

Para lograr el cumplimiento de estas prácticas fue creado en la UCI el IPP-3510: Libro de Proceso para la Administración de Requisitos, que tiene como objetivo definir el proceso para el área de proceso REQM. Precisa el ciclo de vida del desarrollo de un proyecto con 9 fases:

- Estudio preliminar, Modelación del Negocio, Requisitos, Análisis y diseño, Implementación, Pruebas Internas, Pruebas de liberación, Despliegue y Soporte.

Este libro define además las actividades, roles y artefactos (productos típicos de trabajo) generados durante la vida del proyecto. Fue elaborado siguiendo como guía la metodología RUP.

RUP (Proceso Unificado de Desarrollo) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el

Capítulo 1: Fundamentación Teórica

análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos y roles. Tiene una amplia usabilidad y potencial de desarrollo.

RUP se apoya además en **UML (Unified Modeling Language)**, que es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema de software orientado a objetos. Se ha convertido en el estándar de facto de la industria.

Se define **BPMN (Business Process Modeling Notation)** para representar los procesos de negocio en un flujo de trabajo. Es una notación gráfica estándar que ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre ellos.

Después de la investigación realizada sobre las herramientas y tecnologías existentes en el mundo para la creación del software, se ha decidido utilizar Microsoft Visual C# .NET 2.0 como lenguaje de desarrollo ya que el mismo provee un entorno de desarrollo flexible, y la posibilidad de desarrollar soluciones para una gran gama de plataformas y dispositivos. Se seleccionó como IDE de desarrollo Microsoft Visual Studio 2008 por sus funcionalidades y por el soporte del lenguaje C# 2.0 como tecnología.

Se escogió como herramienta CASE el Enterprise Architect 7.5 por ser la herramienta más potente y flexible para la plataforma de Windows y por su integración con el Visual Studio 2008 y compatibilidad con C# 2.0. Como Sistema de Gestión de Base de Datos se seleccionó PostgreSQL 8.3 por ser libre y orientado a objetos. Además todas fueron seleccionadas por formar parte del documento de arquitectura del SWMI de la UCI.

Se escogió CMMI como modelo para la mejora y evaluación de los procesos del desarrollo, mantenimiento y operación del sistema, por tener una vista integradora de mejora de procesos a través de múltiples disciplinas.

En este capítulo se analizaron las principales herramientas sobre el otorgamiento de licencias de software a nivel mundial y se apreció que tienen características similares como: permiten monitorear y controlar el uso de licencias, además le permiten a los usuarios finales la activación del sistema a través de una clave que le es otorgada, posibilitando así verificar a través de un servidor de activación una vez introducida bien la clave, que la PC tenga permiso para ejecutar el software. Por otra parte se evidencia que algunas de estas herramientas tienen un alto valor, por lo que se hace difícil su adquisición.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En el presente capítulo se describen los procesos actuales en el flujo de negocio relacionado con el otorgamiento de licencias, se menciona la información adicional del sistema propuesto, se hace una breve descripción de la solución propuesta. Así como de sus requerimientos funcionales y no funcionales, agrupando los primeros, en Casos de Uso, con el fin de configurar los Diagrama de Casos de Uso del Sistema todo esto se hace en función de guiar el desarrollo del sistema y de esta forma comprender su contexto e identificar las condiciones o capacidades que debe cumplir.

2.1 Descripción del sistema propuesto.

Debido a lo engorroso e inseguro de la forma en que se lleva el proceso del otorgamiento de licencias, durante el despliegue de los sistemas informáticos, por parte del departamento de Software Médico Imagenológico (SWMI) y dado que no existe un sistema que se ajuste a las necesidades de este departamento, se plantea la creación de dos módulos para la gestión del otorgamiento de licencias del sistema alasPACS Viewer: Cliente y Gestor.

El módulo Cliente es el encargado de recoger los datos del hardware únicos para cada PC, esto consiste en capturar los datos de cada una de ellas donde se quiere instalar el sistema y generar un fichero de solicitudes de licencias para cada una. Además debe ser capaz de importar el paquete de licencias asociado a un grupo de PCs.

El módulo Gestor se encarga de generar licencias, que no es más que importar los ficheros de solicitudes de licencias y generar todas las licencias asociadas a cada PC, creando así un único paquete con todas las licencias incluidas. La creación de estos Módulos le permitirá al Departamento de Software Médico Imagenológico la activación del sistema alas PACSViewer de forma más eficiente y segura.

2.2 Información que se maneja.

La información que se maneja son los datos de hardware de las PCs, únicos para cada una de ellas. De estas se recogen además información adicional, como son: el nombre y dirección de la Institución a la que pertenece, número de serie y la ubicación.

2.3 Modelo de Procesos del Negocio.

El modelado de procesos de negocio es el punto de partida para el diseño y creación del software, es la base para comprender mejor la operación de una organización, documentar y publicar los procesos buscando una estandarización en la organización. Es la fase de entendimiento y

Capítulo 2: Características del Sistema

modelado de los requisitos del sistema. Además es donde se incide en analizar y obtener un conocimiento pleno de cada uno de los procesos de negocio, consiguiendo un modelo del sistema que se adapte mejor a sus necesidades.

2.3.1 Procesos del negocio actual.

El Especialista en Despliegue va a la institución con un dispositivo externo (memoria), el cual tiene almacenado un Gestor para generar licencias y el sistema a instalar. Cuando el Especialista en Despliegue llega, instala el sistema en la PC y seguido lo ejecuta, cuando realiza esta acción se levanta una ventana con el nombre Serial, que tiene un botón Generar y otro Importar. Luego oprime el botón Generar y le sale una ventana para guardar la solicitud de la licencia, el Especialista en Despliegue crea la carpeta donde la solicitud va a ser guardada.

Una vez que de guardar, automáticamente se abre otra ventana llamada Registro para llenar los datos de la solicitud, la cual cuando se levanta extrae los datos de hardware de dicha PC de una forma no visible. El Especialista en Despliegue llena los datos y una vez que realice esta acción queda guardada la solicitud en la carpeta creada anteriormente.

Acto seguido el Especialista en Despliegue ejecuta el Gestor y se levanta una ventana con el nombre de Licencias de Clientes, luego le da buscar e importa la solicitud de licencias ya creada y automáticamente se llenan todos los campos de esta ventana. Después registra y guarda la licencia generada para esa solicitud, luego el Especialista oprime el botón Importar que se encuentra en la ventana Serial y se importa esa licencia generada, una vez ocurrido esto el usuario puede utilizar el software debidamente. El Especialista en Despliegue concluye la tarea retirando el dispositivo y ejecuta este mismo proceso en todas las PCs en las cuales se tiene previsto instalar el sistema.

2.3.2 Proceso del Negocio Mejorado.

En la gestión del otorgamiento de licencias para el sistema alas PACSViewer se define el proceso: gestión del otorgamiento de licencias del software, el cual cuenta con un flujo de actividades diferentes bien definidas. A continuación se describe dicho proceso.

El proceso se inicia cuando el Especialista en Despliegue llega a la Institución Hospitalaria y procede a introducir el dispositivo externo (memoria) en la PC para instalar el sistema alas PACSViewer. Luego llena los datos: nombre y dirección de la Institución a la que pertenece la PC, número de serie de la PC, y ubicación de la PC y obtiene los datos de hardware que son únicos para cada PC de forma automática, después de realizar esta acción retira su dispositivo. Este proceso lo hace en todas las PC que le hayan sido otorgadas para instalar el sistema. Una vez que termina se retira de la institución y se dirige al Grupo de Desarrollo localizado en la Universidad de

Capítulo 2: Características del Sistema

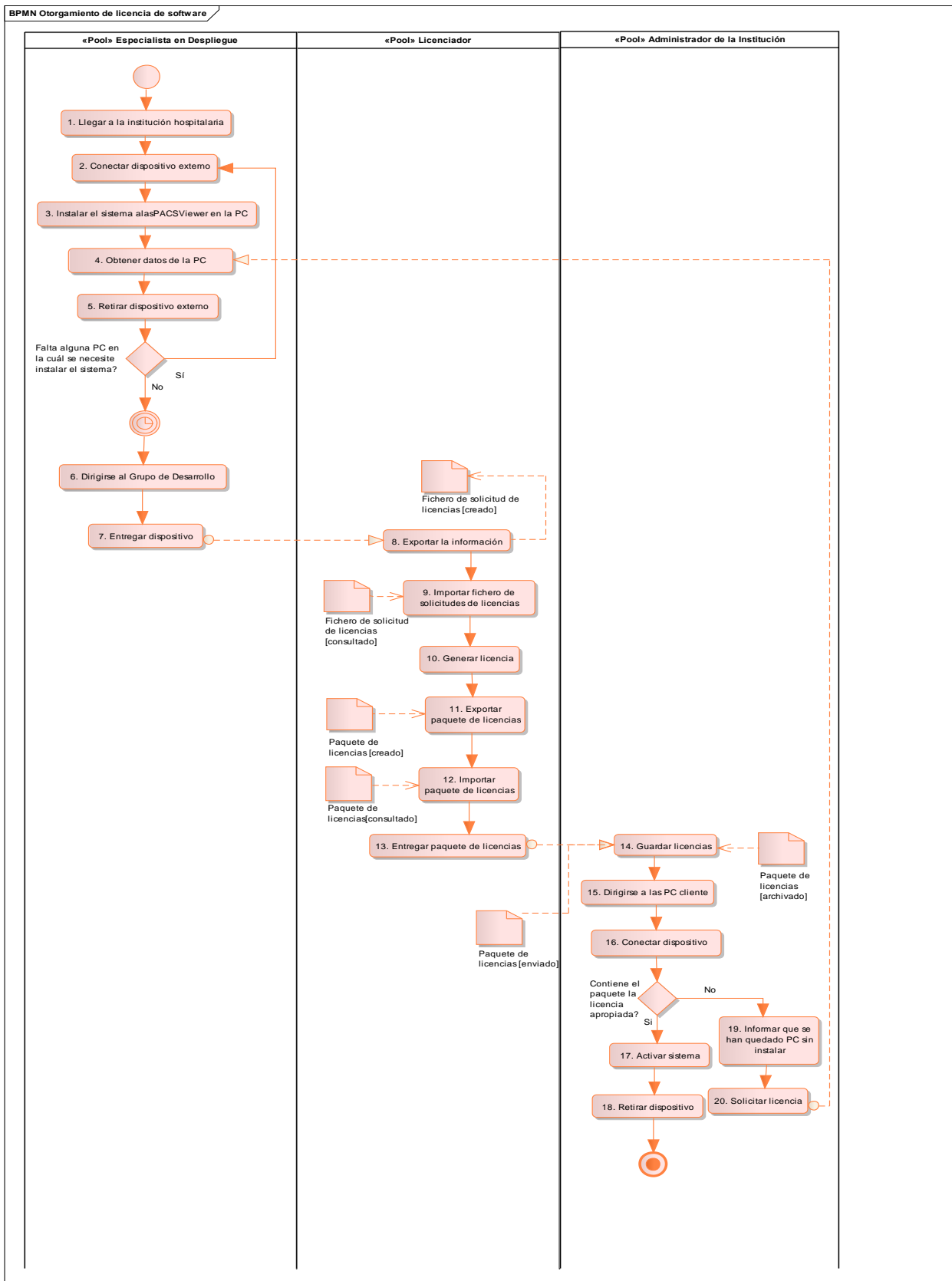
las Ciencias Informáticas. En caso de que se haya olvidado recoger los datos de alguna PC, el Especialista volvería a ir otro día a la Institución y realizaría el mismo procedimiento.

Una vez que el Especialista en Despliegue llega al Departamento de Software Médico Imagenológico le entrega el dispositivo al Licenciador con los datos recogidos de cada PC. Este último se encarga de gestionar las solicitudes de licencias y de exportar la información contenida a un fichero de solicitudes de licencias. Luego importa dicho fichero y genera las licencias asociadas a cada PC, creándose así un paquete con todas las licencias incluidas. Después el Licenciador se encarga de exportar e importar dicho paquete.

El Licenciador luego de haber obtenido el paquete de licencias generadas se lo hace llegar al Administrador de la Institución donde fueron recogidas las solicitudes, este una vez que recibe el paquete de licencias se encarga de activar el sistema alas PACSViewer en cada PC, comprobando con los datos del hardware de dichas PCs que el paquete contenga una licencia apropiada para cada una de ellas. En caso de no contener dicha licencia tendría el Administrador de la Institución que informar al Departamento de Software Médico Imagenológico que se han quedado PC sin instalar.

El Administrador luego de haber activado el sistema en cada una de las PCs a licenciar, queda asignado como responsable de conservar estas licencias, las cuales pueden ser reutilizables en caso de existir algún problema que no esté asociado con los datos de hardware de la PC. En caso contrario, es decir en caso de que ocurra algún percance como: rotura de una PC o algún cambio de los componente de hardware de alguna PC, el Administrador tendría que informar el suceso para que se vuelva a iniciar el proceso de gestión del otorgamiento de licencias del software sobre dicha PC. En el **¡Error! No se encuentra el origen de la referencia.** se encuentra la descripción del proceso y la **¡Error! No se encuentra el origen de la referencia.**muestra el flujo de actividades de dicho proceso.

Capítulo 2: Características del Sistema



Capítulo 2: Características del Sistema

Fig.1 Diagrama del Proceso Otorgamiento de licencia de software

2.4 Especificación de los requerimientos del software.

Los requisitos constituyen el hilo conductor de todo desarrollo de software. Pues es una forma de evitar discrepancias y variedades de soluciones entre el equipo de desarrollo y el usuario final. Además una obtención de requisitos con calidad daría pie a que el desarrollo del software se culmine con éxito, pues el software poseerá lo que el cliente desea, y permitirá gestionar las necesidades del proyecto en forma estructurada, además evita el rechazo de usuarios finales debido a que obliga a los usuarios a considerar sus requerimientos cuidadosamente.

2.4.1 Requerimientos Funcionales del Sistema.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

Cliente: Aplicación encargada de captar los datos de hardware de una PC, así como la posterior activación del sistema alas PACSViewer.

Gestor: Aplicación encargada de generar las licencias para los datos de hardware recogidos por el Cliente.

Requerimientos funcionales del Cliente.

alas PACSViewer Cliente		
Requerimientos Funcionales	Descripción	Etapa
RF 1 Obtener Datos de la PC.		Instalación
➤ RF 1.1 Obtener datos de ubicación de la PC.	Se llenan los datos asociados a las PCs que se que se van a licenciar, como son: nombre de la Institución a la que pertenece la PC, Número de serie de la PC, y Ubicación de la PC.	
➤ RF 1.2 Obtener datos de hardware de la P	Se recogen los datos de configuración de hardware de cada PC en la que se va a instalar el sistema de forma automática.	
RF 2 Almacenar Datos Obtenidos.	Los datos recogidos se mantienen en una representación interna que solo puede ser accedida con el Cliente.	Instalación

Capítulo 2: Características del Sistema

RF 3 Generar Solicitud.	A partir de los datos recogidos se crea una solicitud de licencia.	Instalación
RF 4 Gestionar Solicitud.		Gestión
➤ RF 4.1 Visualizar Solicitud.	Incluye la solicitud de licencia en una lista que contiene cada una de las solicitudes de las PCs cuyos datos han sido recogidos y muestra dicha lista.	
➤ RF 4.2 Buscar Solicitud.	Permite encontrar una solicitud dada usando como criterio de búsqueda: Número de serie de la PC.	
➤ RF 4.3 Eliminar solicitud.	Permite eliminar una solicitud correspondiente a una PC que no deba estar licenciada y se le hayan recogido los datos por equivocación.	
RF 5 Exportar Fichero de Solicitud de Licencias.	Se exportan los datos almacenados en el Cliente hacia un fichero con un formato conocido tanto por el Cliente como por el Gestor.	Gestión
RF 6 Importar Paquete de Licencias.	Incorpora y almacena los datos de las licencias generadas correspondientes a las capturas realizadas.	Registro
RF 7 Identificar una licencia que corresponda a una estación dada.	Busca que haya en el paquete de licencias una correspondiente a la PC a activar.	Registro
RF 8 Verificar que la licencia se corresponda con la estación a activar.	Verifica con los datos de hardware de la PC que la licencia identificada sea la correcta.	Registro
RF 9 Activar Licencia.	Activa el sistema, permitiendo la utilización del mismo.	Instalación
RF 10 Renovar Licencia.	En caso de que una PC se haya roto y se le haya sustituido algún componente de hardware, la licencia que disponía ya no es válida entonces es necesario renovarla.	Soporte

Capítulo 2: Características del Sistema

RF 11 Gestionar Licencias.		Gestión
➤ RF 11.1 Visualizar Lista de Licencia.	Incluye la licencia a una lista que contiene todas las licencias generadas correspondientes a las capturas realizadas y muestra dicha lista.	
➤ RF 11.2 Buscar Licencia.	Permite encontrar una licencia dada, usando como criterio de búsqueda: Número de serie de la PC.	
RF 12 Vaciar almacén de solicitudes de licencias y paquete de licencias.	Se encarga de borrar todas las solicitudes de licencia y las licencias asociadas a cada una de ellas una vez que se haya terminado de licenciar la Institución Hospitalaria.	Soporte

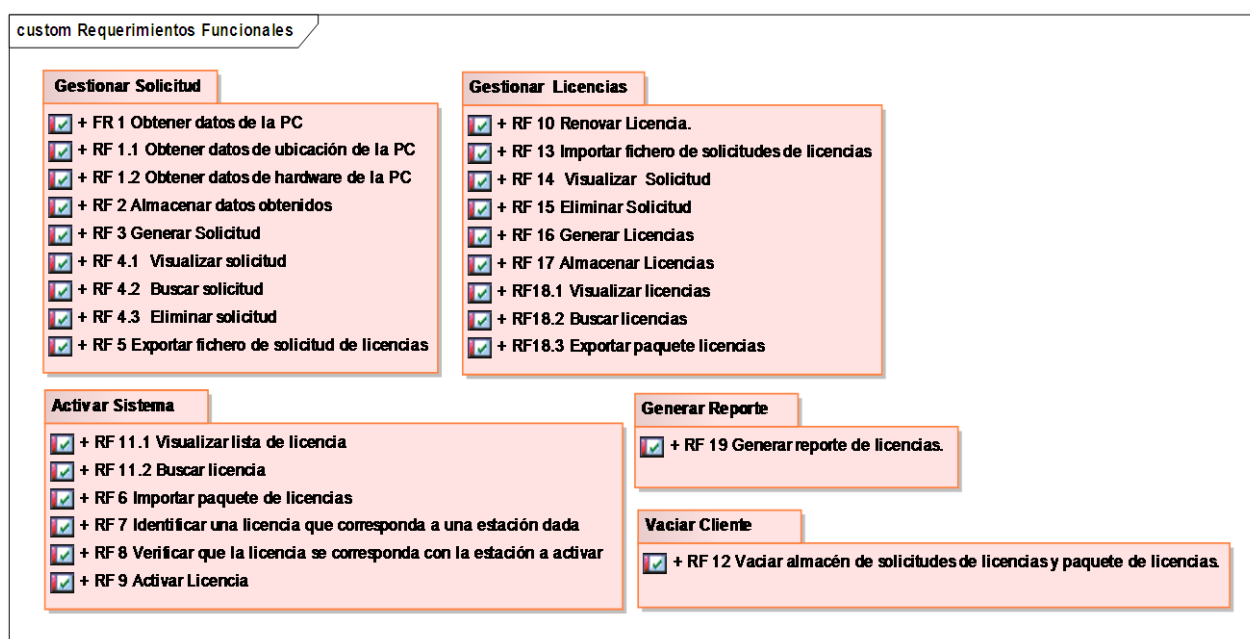
Requerimientos funcionales del Gestor.

al as PACSViewer Gestor		
Requerimientos Funcionales	Descripción	Etapa
RF 13 Importar Fichero de Solicitudes de Licencias.	Incorpora y almacena el fichero donde están contenidas las capturas de información de Hardware de cada PC.	Gestión
RF 14 Visualizar Solicitud.	Muestra una lista con cada solicitud de licencia incluida en el fichero importado.	Gestión
RF 15 Eliminar Solicitud.	Permite eliminar una solicitud de licencia que por alguna razón no deba ser atendida.	Gestión
RF 16 Generar Licencias.	Genera una licencia para cada una de las PCs que se muestran en la lista de solicitudes de licencias.	Soporte
RF 17 Almacenar Licencias.	Guarda de manera persistente las licencias generadas en una representación interna del mismo.	Soporte

Capítulo 2: Características del Sistema

RF 18 Gestionar Paquete Licencias.		
➤ RF18.1 Visualizar Licencias.	Muestra una lista con todas las licencias generadas.	Soporte
➤ RF18.2 Buscar Licencias.	Permite encontrar una licencia dada usando como criterio de búsqueda: Número de serie de la PC.	
RF 19 Exportar Paquete Licencias.	Exporta el paquete de licencias.	
RF 20 Generar reporte de licencias.	Muestra un informe con todas las solicitudes de licencias de una determinada Institución Hospitalaria, las licencias generadas para cada una de ellas, así como todas aquellas que han sido renovadas.	Soporte

En la **¡Error! No se encuentra el origen de la referencia.** se muestra el diagrama de requerimientos funcionales agrupados por paquetes lógicos con el objetivo de ganar organización en el desarrollo de la investigación. Además de ello, se garantiza tener los requerimientos funcionales agrupados según su relación.



Capítulo 2: Características del Sistema

Fig.2 Diagrama de los requerimientos funcionales

2.4.2 Requerimientos No Funcionales del Sistema.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Son características que hacen al producto atractivo, usable, rápido o confiable.

A continuación se presentarán los requerimientos no funcionales definidos para la herramienta. Estos están divididos por categorías y cada una tiene asociado un prefijo con la que será identificada: Usabilidad: RNU, Diseño e Implementación: RNDI, Seguridad: RNS, Funcionamiento: RNFO.

Categoría	Prefijo	Descripción
Usabilidad	RNU	RNU 1. El sistema debe mostrar mensajes al usuario que le ayuden a llevar a cabo la tarea que realiza. RNU 2. Se debe hacer uso de botones con textos claros y entendibles que indiquen de modo intuitivo la función que realizan.
Diseño e implementación	RNDI	RNDI 1. Se deberá utilizar C# como lenguaje de programación pues está diseñado y optimizado para la plataforma .NET.
Seguridad	RNS	RNS 1. Cada licencia generada es específica para una única PC.
Funcionamiento	RNFO	RNFO 1. Se recomienda Windows XP SP2 (Service Pack 2) o superior. RNFO 2. Se recomienda .Net Framework 2.0 o superiores. RNFO 3. Se debe reservar 512 de memoria RAM o superiores para que la aplicación corra con mayor rapidez. RNFO4. Se debe reservar 80 GB de Hard Disc Driver para la instalación de .Net Framework y la aplicación.

Capítulo 2: Características del Sistema

En la **¡Error! No se encuentra el origen de la referencia.** se muestra el diagrama de requerimientos no funcionales, con el objetivo fundamental de agruparlos según su categoría y poder ganar en organización y estructuración de la documentación.

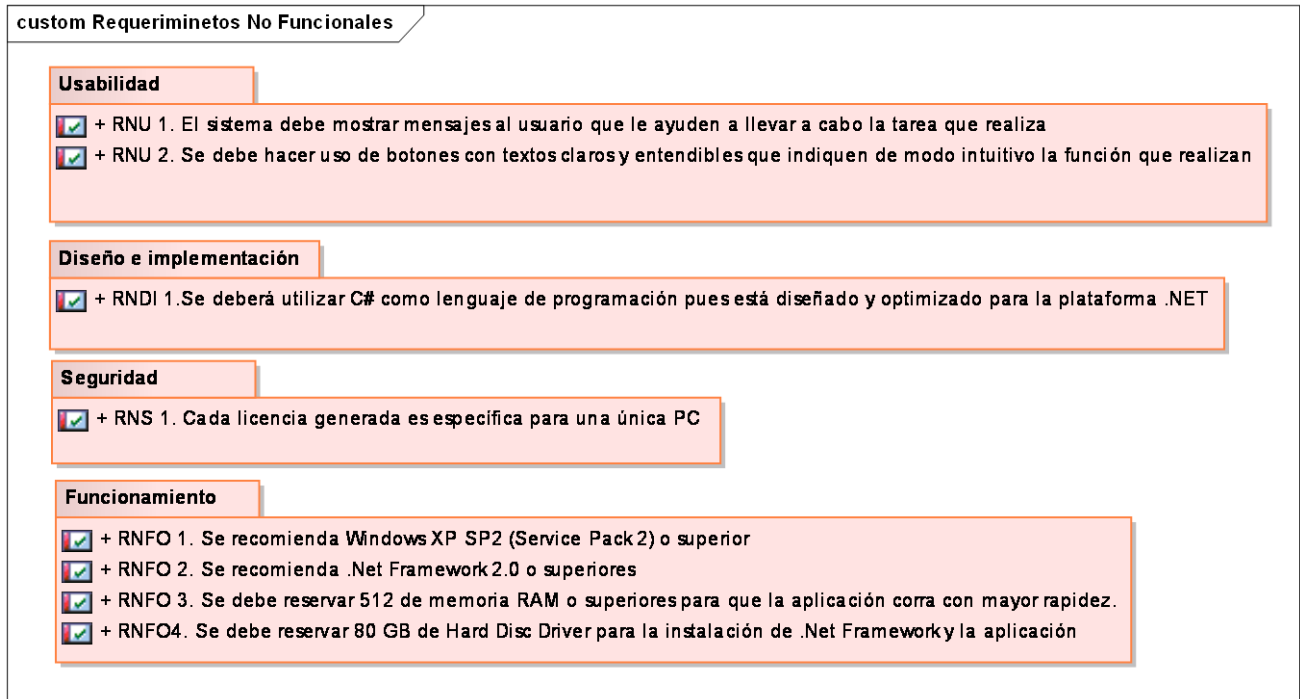


Fig.3 Diagrama de los requerimientos no funcionales

2.5 Definición de los actores del sistema.

Un actor del sistema representa un conjunto coherente de roles que juegan los usuarios. Es una entidad externa al sistema que se modela y que puede interactuar con él. Los actores pueden ser humanos, hardware externo, u otros sujetos que interactúen con el sistema que se esté especificando.

Se definieron dos actores, los cuales realizarán un grupo de acciones en el sistema, los mismos se describen a continuación:

Actor	Descripción
Operador del Cliente	Rol que representa un usuario que interactúa con el Cliente.
Licenciador	Rol que representa un usuario encargado de generar las licencias para las PC, cuyos datos fueron previamente recolectados.

2.6 Diagrama de casos de uso del sistema.

Los diagramas de casos de uso se utilizan para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un

Capítulo 2: Características del Sistema

diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

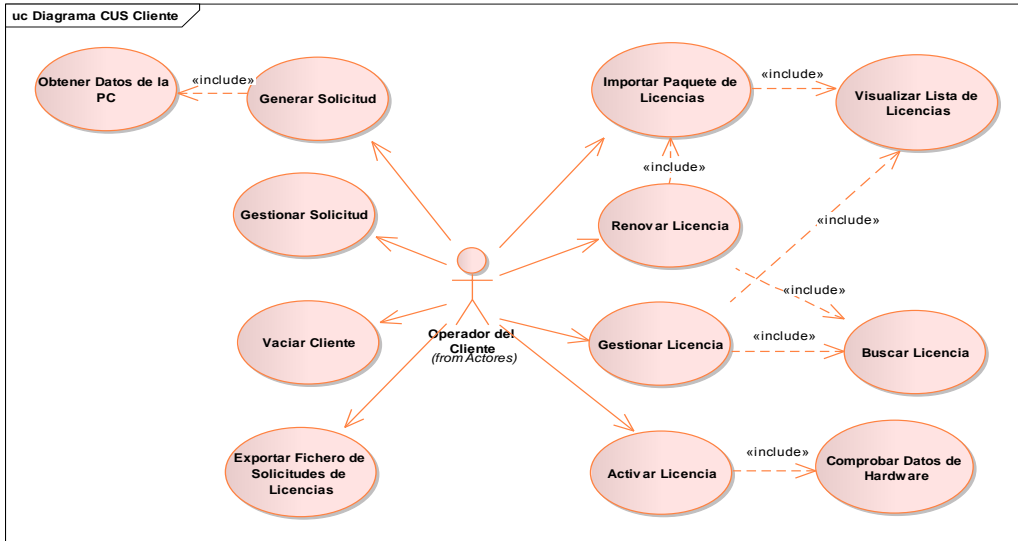


Fig.4 Diagrama de Casos de Uso del sistema del módulo Cliente

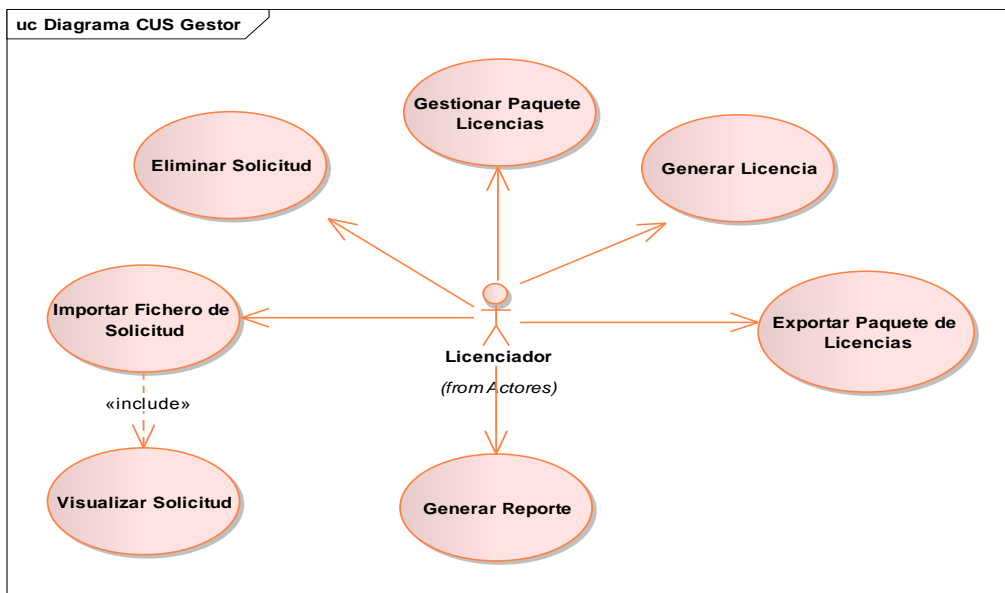


Fig.5 Diagrama de Casos de Uso del módulo Gestor.

2.7 Descripción de los casos de uso del sistema.

Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Normalmente, en los casos de usos se evita el empleo de jergas técnicas, prefiriendo en su lugar un lenguaje más cercano al usuario final. En otras palabras, un caso de uso es una

Capítulo 2: Características del Sistema

secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

2.7.1 Listado de los casos de uso del Cliente.

<u>CU # 1</u>	Obtener Datos de la PC.
Actor	Operador del Cliente.
Descripción	Obtiene los datos de configuración de hardware de una PC así como otros datos adicionales como son: nombre de la Institución a la que pertenece la PC, número de serie de la PC, y ubicación de la PC y los almacena internamente en el Cliente.
Referencia	RF 1, RF 2.
<u>CU # 2</u>	Generar Solicitud.
Actor	Operador del Cliente.
Descripción	Permite crear una solicitud de licencia a partir de los datos recogidos.
Referencia	RF 3.
<u>CU # 3</u>	Gestionar Solicitud.
Actor	Operador del Cliente.
Descripción	Permite Visualizar Solicitud, Buscar Solicitud y Eliminar Solicitud.
Referencia	RF 4.
<u>CU # 4</u>	Exportar Fichero de Solicitudes de Licencias.
Actor	Operador del Cliente.
Descripción	Permite exportar los datos previamente obtenidos, hacia un fichero de solicitudes de licencias cuyo formato es conocido tanto por el Cliente como por el Gestor.
Referencia	RF 5.
<u>CU # 5</u>	Importar Paquete de Licencias.
Actor	Operador del Cliente.
Descripción	Permite incorporar y almacenar los datos de las licencias generadas correspondientes a las capturas realizadas.
Referencia	RF 6.
<u>CU # 6</u>	Comprobar Datos de Hardware.
Actor	Operador del Cliente.
Descripción	Permite buscar en el paquete de licencias una correspondiente a la PC a activar, verificando con los datos de hardware de la PC que la licencia identificada sea la correcta.

Capítulo 2: Características del Sistema

Referencia	RF 7, RF 8.
<u>CU # 7</u>	Activar Licencia.
Actor	Operador del Cliente.
Descripción	Permite activar el sistema, permitiendo la utilización del mismo.
Referencia	RF 9.
<u>CU # 8</u>	Renovar Licencia.
Actor	Operador del Cliente.
Descripción	Permite sustituir una licencia que haya dejado de ser válida por una nueva.
Referencia	RF 10.
<u>CU # 9</u>	Gestionar Licencia.
Actor	Operador del Cliente.
Descripción	Permite Visualizar Lista Licencia, Buscar Licencia.
Referencia	RF 11.
<u>CU # 10</u>	Vaciar Cliente.
Actor	Operador del Cliente.
Descripción	Elimina toda la información contenida una vez que se halla terminado de licenciar una determinada Institución Hospitalaria.
Referencia	RF 12.

2.7.2 Listado de los casos de uso del Gestor.

<u>CU # 11</u>	Importar Fichero de Solicitud.
Actor	Licenciador.
Descripción	Permite importar los datos contenidos en el fichero que tiene las solicitudes de licencias.
Referencia	RF 13.
<u>CU # 12</u>	Visualizar Solicitud.
Actor	Licenciador.
Descripción	Permite mostrar una lista con cada solicitud de licencia incluida en el fichero importado.
Referencia	RF 14.
<u>CU # 13</u>	Eliminar Solicitud.
Actor	Licenciador.

Capítulo 2: Características del Sistema

Descripción	Permite eliminar una solicitud de licencia que por alguna razón no deba ser atendida.
Referencia	RF 15.
<u>CU # 14</u>	Generar Licencia.
Actor	Licenciador.
Descripción	Permite generar una licencia para cada una de las PCs cuyos datos se encuentran en el fichero de solicitudes de licencias y las almacena internamente.
Referencia	RF 16, RF 17.
<u>CU # 15</u>	Gestionar Paquete Licencias.
Actor	Licenciador.
Descripción	Permite Visualizar Licencias, Buscar Licencias.
Referencia	RF 18.
<u>CU # 16</u>	Exportar Paquete de Licencias.
Actor	Licenciador.
Descripción	Exporta el paquete de licencias.
Referencia	RF 19.
<u>CU # 17</u>	Generar Reporte.
Actor	Licenciador.
Descripción	Permite mostrar un informe con todas las solicitudes de licencias de una determinada Institución Hospitalaria, las licencias generadas para cada una de ellas, así como todas aquellas que han sido renovadas.
Referencia	RF 20.

2.8 Casos de uso arquitectónicamente significativos.

Los casos de uso arquitectónicamente significativos son aquellos que representan las partes más críticas de la arquitectura del sistema y demuestran la funcionalidad del mismo. Son los más importantes para el usuario y ayudan a controlar los riesgos.

A continuación se muestra los casos de uso definidos como arquitectónicamente significativos teniendo en cuenta que ellos son indispensables para garantizar las utilidades principales de la aplicación, representando las bases del funcionamiento de la misma. Específicamente los casos de

Capítulo 2: Características del Sistema

uso representados en esta vista son los responsables de la obtención de los datos de las PCs en las que se va a activar el sistema a las PACSViewer y de la generación de las licencias asociadas.

2.8.1 Diagrama de Caso de Uso Significativos del Cliente.

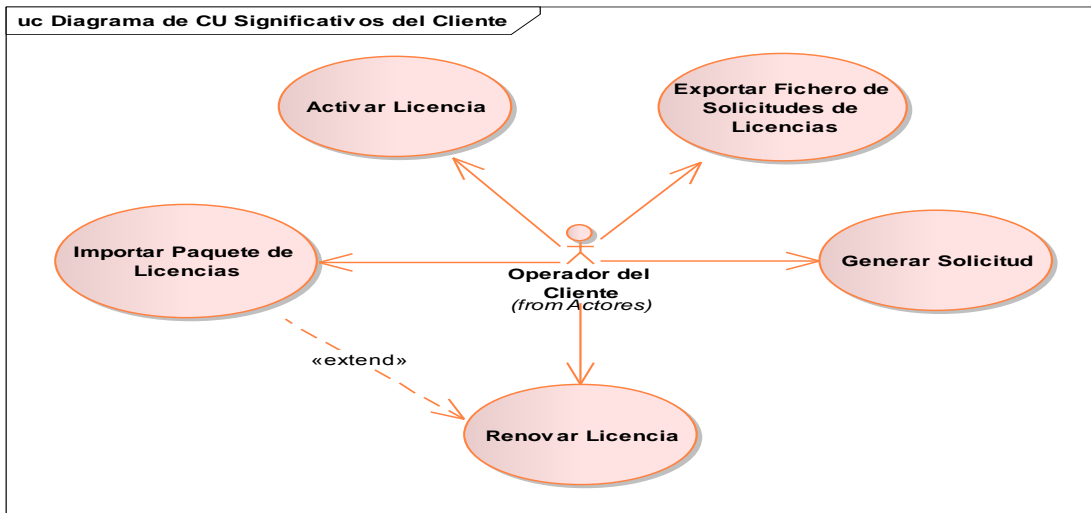


Fig.6 Diagrama de Casos de Uso Significativos del Cliente

2.8.2 Diagrama de Caso de Uso Significativos del Gestor.

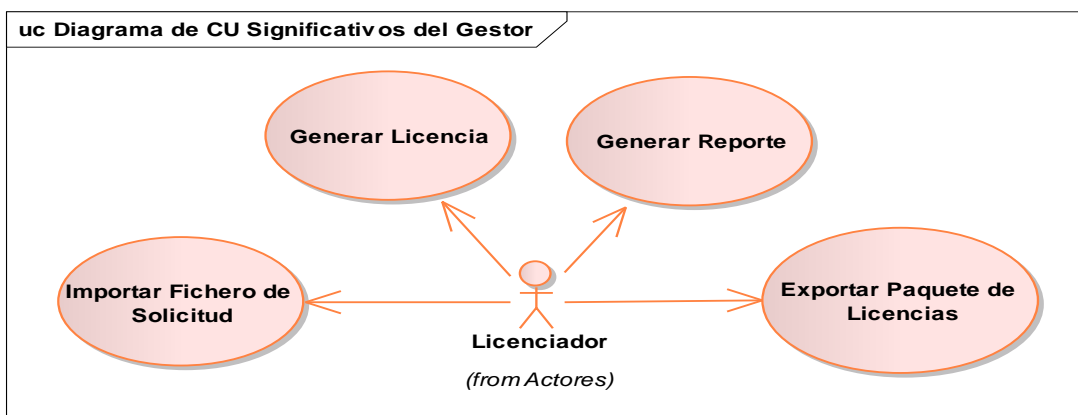


Fig.7 Diagrama de Casos de Uso Significativos del Cliente

En el 0 de la presente investigación se encuentra la descripción textual detallada de los casos de usos arquitectónicamente significativos del Cliente y del Gestor.

En este capítulo se realizó un estudio sobre los principales conceptos que se manejan en el proceso de gestión del otorgamiento de licencias para el sistema a las PACSViewer, se propone realizar el diseño de un sistema que permita realizar este proceso. Con este objetivo se obtuvo el modelo de procesos del negocio así como los requerimientos funcionales y no funcionales del sistema.

CAPÍTULO 3: ARQUITECTURA Y DISEÑO

Este capítulo tiene como objetivo principal, convertir los requisitos del software en el diseño de la herramienta para la gestión del otorgamiento de licencias de software para el sistema alas PACSViewer. El diseño se apoya en el resultado del análisis, es por esto que se desarrollan las actividades del análisis, para facilitar el diseño.

3.1 Arquitectura.

La solución propuesta consta de dos módulos: El Cliente que es el encargado de obtener los datos de las PCs en la que se va a instalar el sistema alas PACSViewer y el Gestor que es el encargado de generar las licencias asociadas a cada solicitud de licencia creada a partir de los datos obtenidos de cada PC.

El Cliente es una aplicación que se aloja en un dispositivo externo (memoria flash), el cual es portado por los Especialistas en Despliegue encargados del despliegue en las Instituciones Hospitalarias y es usado para obtener datos únicos de las PC, almacenándolos en una representación interna que luego son exportados a un fichero de solicitudes de licencias. Además carga el paquete de licencias generadas por el Gestor, permitiendo activar cada instalación del sistema en cada una de las PCs, cuyos datos fueron recogidos.

El Gestor es una aplicación que se usa de forma centralizada por el Licenciador que es la persona encargada de generar las licencias, asociadas a cada solicitud de licencia exportada por el Cliente, y exporta un paquete de licencias. Usa una Base de Datos en la que almacena toda la información relacionada con las solicitudes de licencias, las licencias generadas para cada solicitud, y la renovación de las licencias. Genera un informe asociado a toda esta información.

Ambas aplicaciones están diseñadas según el patrón Modelo Vista Controlador, este separa los datos de una aplicación, la lógica de control e interfaz de usuario en tres componentes distintos:

Modelo: Es la representación específica de la información con la cual el sistema opera. Encapsula los datos y las funcionalidades del mismo, siendo independiente de cualquier representación de entrada y/o salida.

Vista: Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

Controlador: Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Ante estos eventos usualmente invoca peticiones al modelo y probablemente a la vista.

3.2 Modelo de Análisis.

El análisis consiste en obtener una visión del sistema que se preocupa de lo que hace el mismo. El modelo de análisis es la entrada fundamental para el comienzo del diseño.

3.2.1 Diagramas del Cliente.

Generar Solicitud.

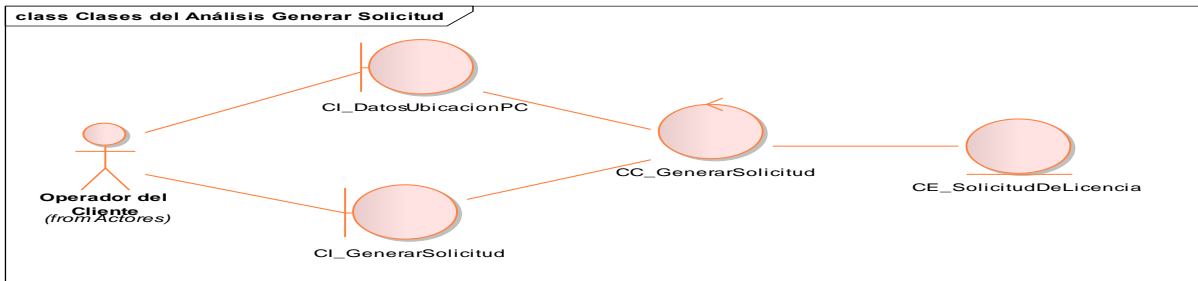


Fig.8 Clases del Análisis CUS Generar Solicitud

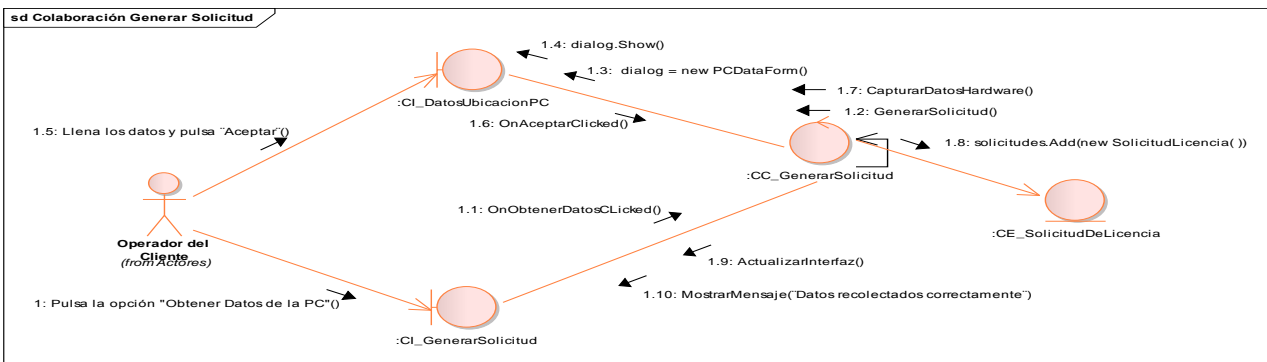


Fig.9 Colaboración del Análisis CUS Generar Solicitud

Exportar Fichero de Solicitudes de Licencias.

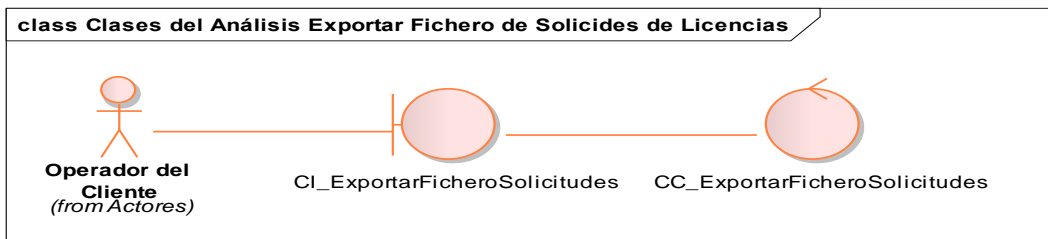


Fig.10 Clases del Análisis CUS Exportar Fichero de Solicitudes de Licencias

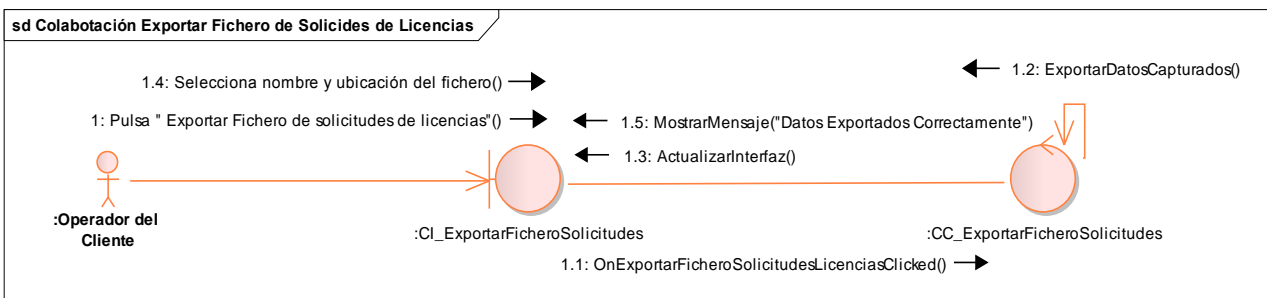


Fig.11 Colaboración del Análisis CUS Exportar Fichero de Solicitudes de Licencias

Importar Paquete de Licencias.

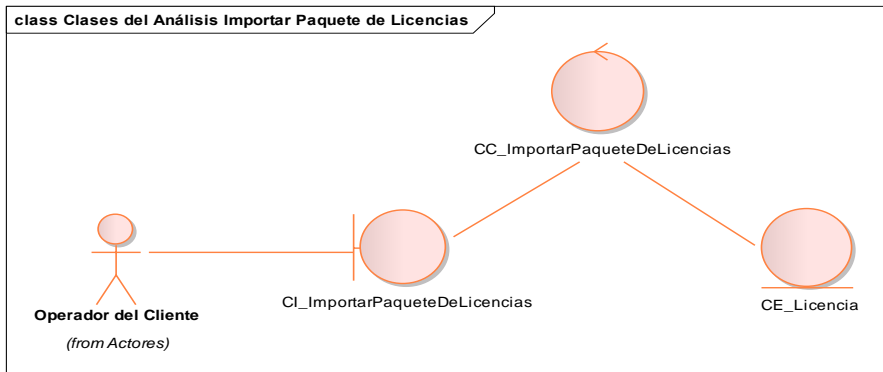


Fig.12 Clases del Análisis CUS Importar Paquete de Licencias

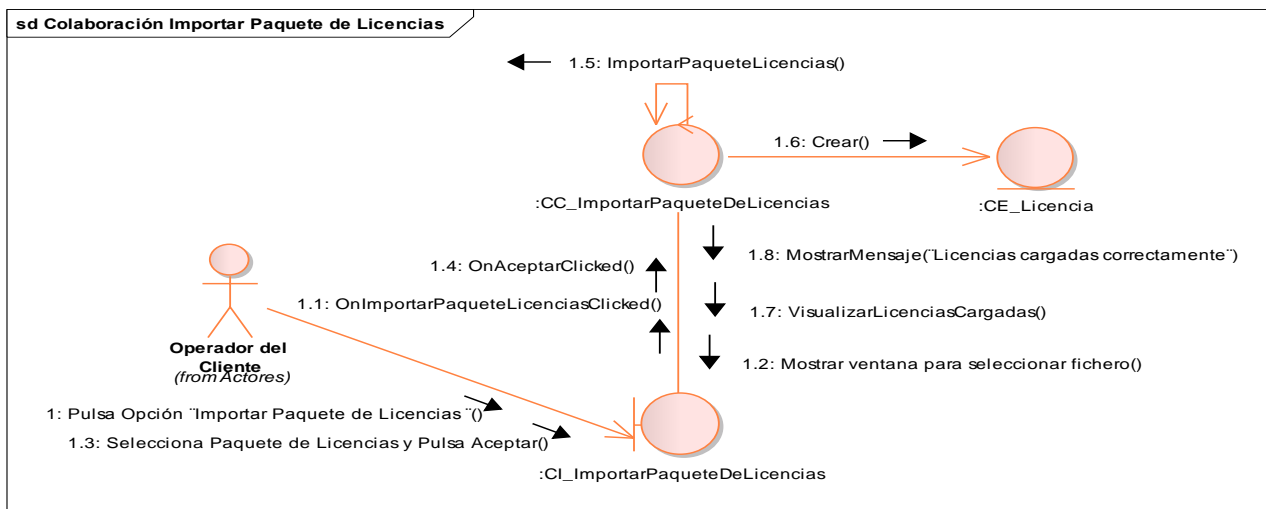


Fig.13 Colaboración del Análisis CUS Importar Paquete de Licencias

Activar Licencia.

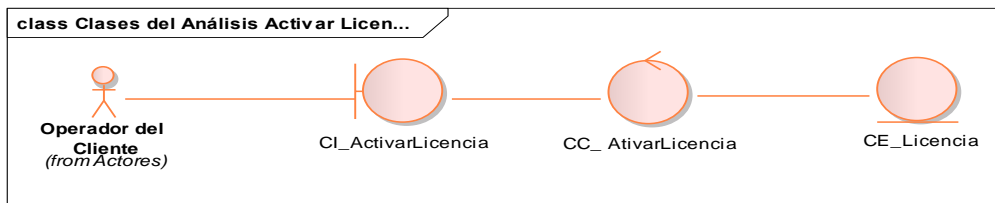


Fig.14 Clases del Análisis CUS Activar Licencia

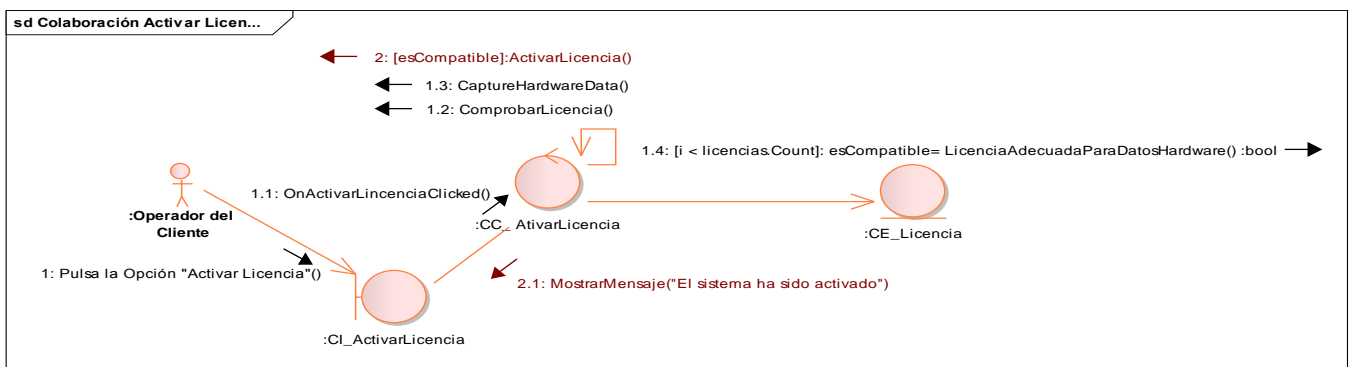


Fig.15 Colaboración del Análisis CUS Activar Licencia

Renovar Licencia.

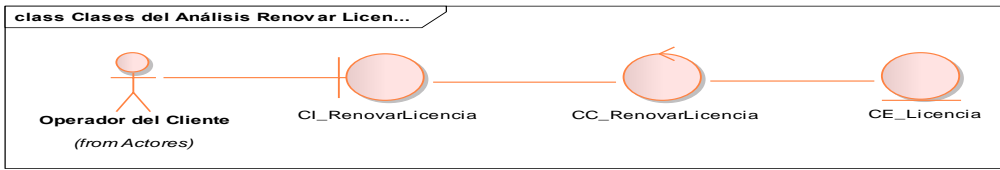


Fig.16 Clases del Análisis CUS Renovar Licencia

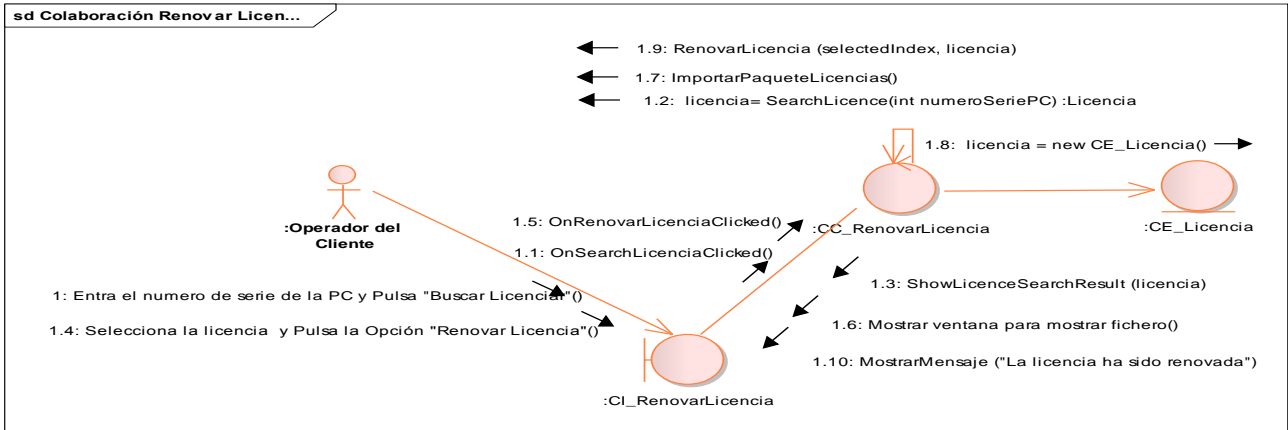


Fig.17 Colaboración del Análisis CUS Renovar Licencia

3.2.2 Diagramas del Gestor.

Importar Fichero de Solicitud.

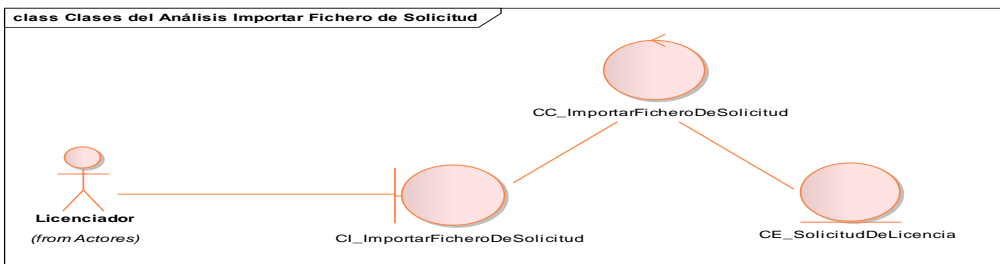


Fig.18 Clases del Análisis CUS Importar Fichero de Solicitud

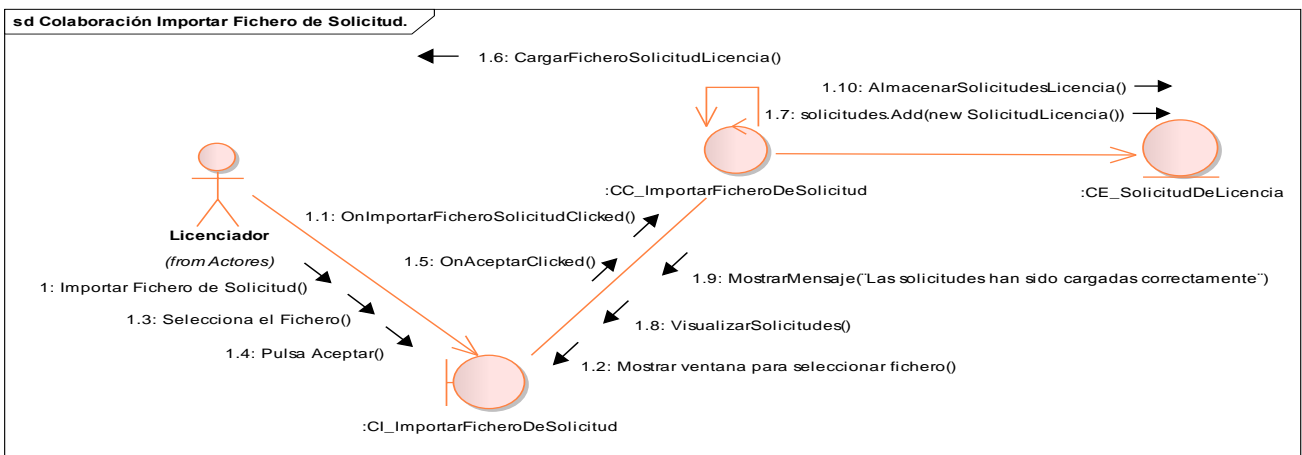


Fig.19 Colaboración del Análisis CUS Importar Fichero de Solicitud

Generar Licencia.

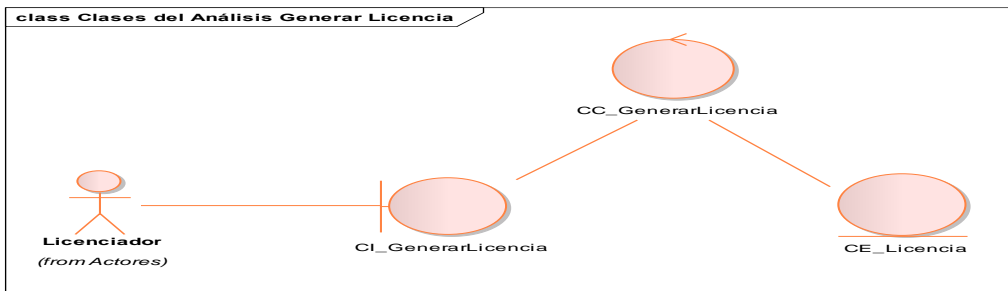


Fig.20 Clases del Análisis CUS Generar Licencia

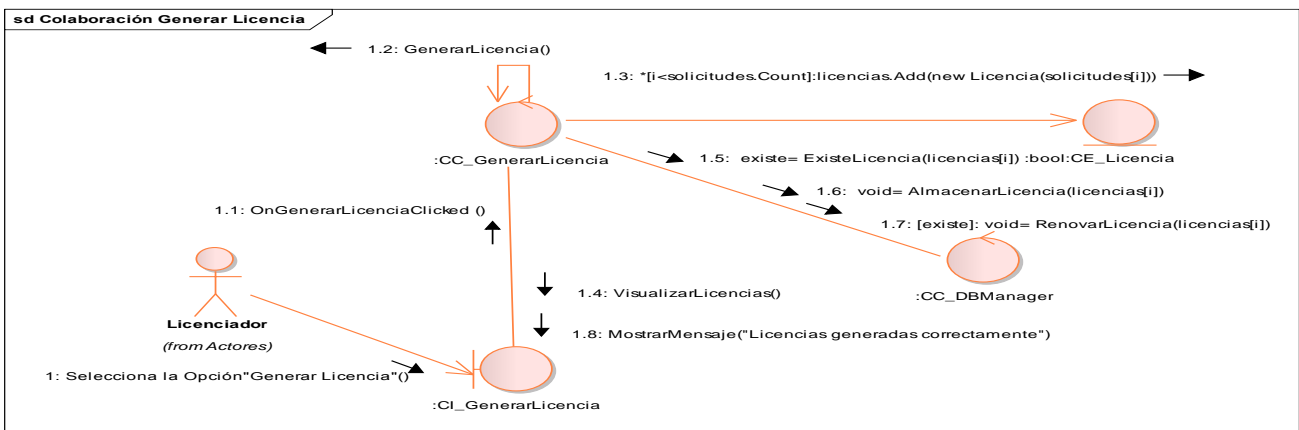


Fig.21 Colaboración del Análisis CUS Generar Licencia

Exportar Paquete de Licencias.

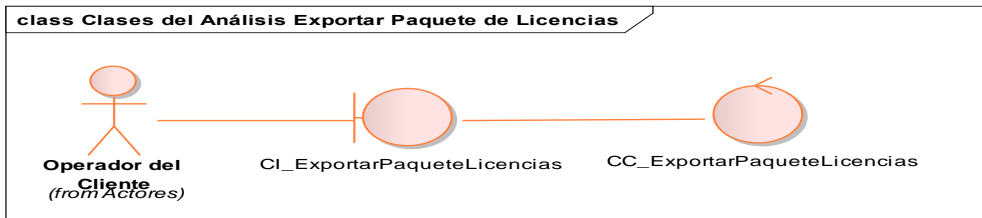


Fig.22 Clases del Análisis CUS Exportar Paquete de Licencias

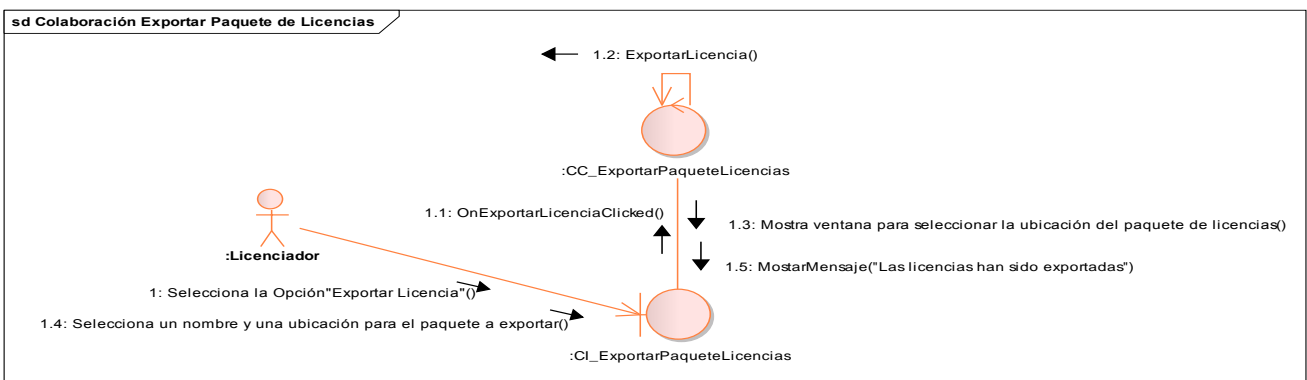


Fig.23 Colaboración del Análisis CUS Exportar Paquete de Licencias

Generar Reporte.

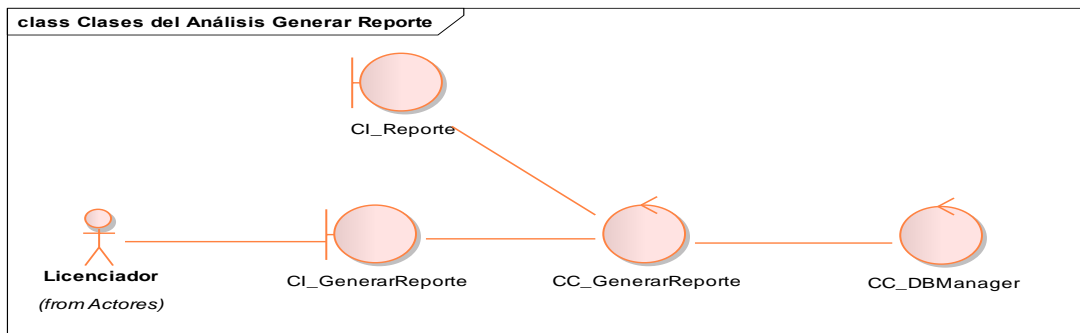


Fig.24 Clases del Análisis CUS Generar Reporte

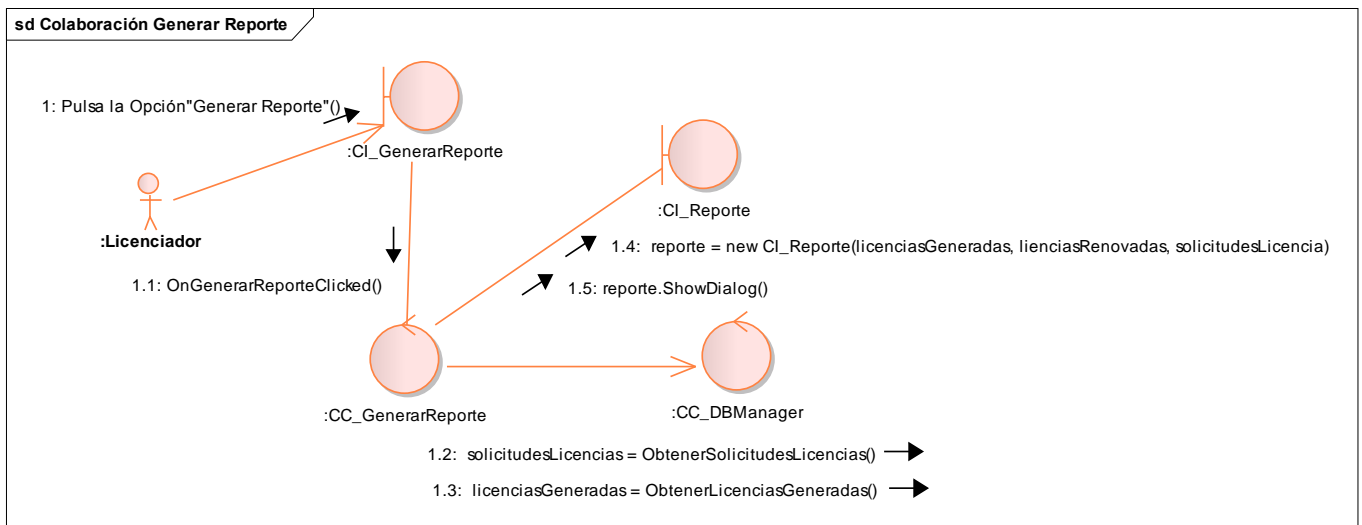


Fig.25 Colaboración del Análisis CUS Generar Reporte

3.3 Diseño.

Un Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema. Éste artefacto constituye la entrada fundamental utilizada para el correcto desarrollo de la implementación.

A continuación se presentan los diagramas de clases para cada caso de uso y un diagrama de secuencia para cada uno de ellos, así como la descripción de las clases.

3.3.1 Diagramas del Cliente.

Generar Solicitud.

➤ Diagrama de Clases:

Capítulo 3: Arquitectura y Diseño

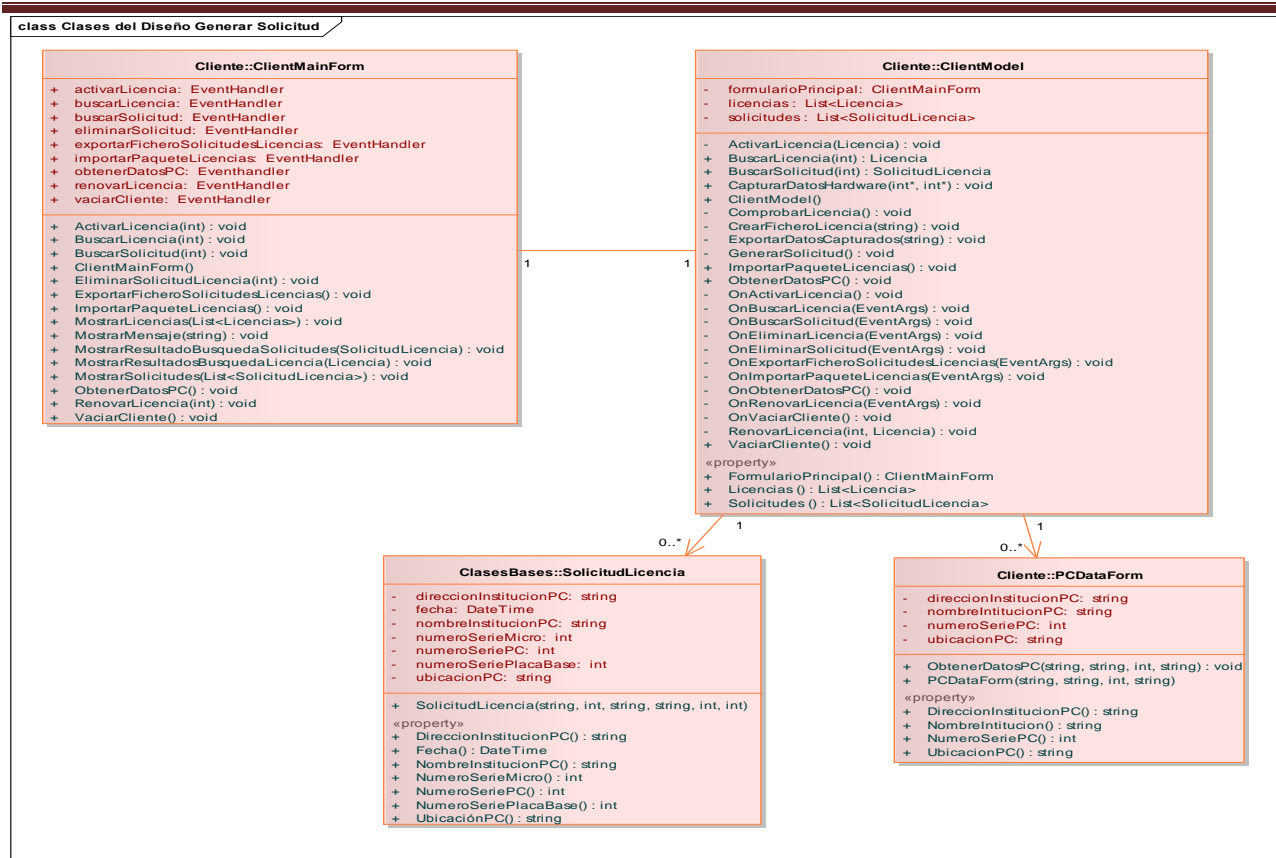


Fig.26 Diagrama de Clases CU Generar Solicitud

➤ Diagrama de Secuencia:

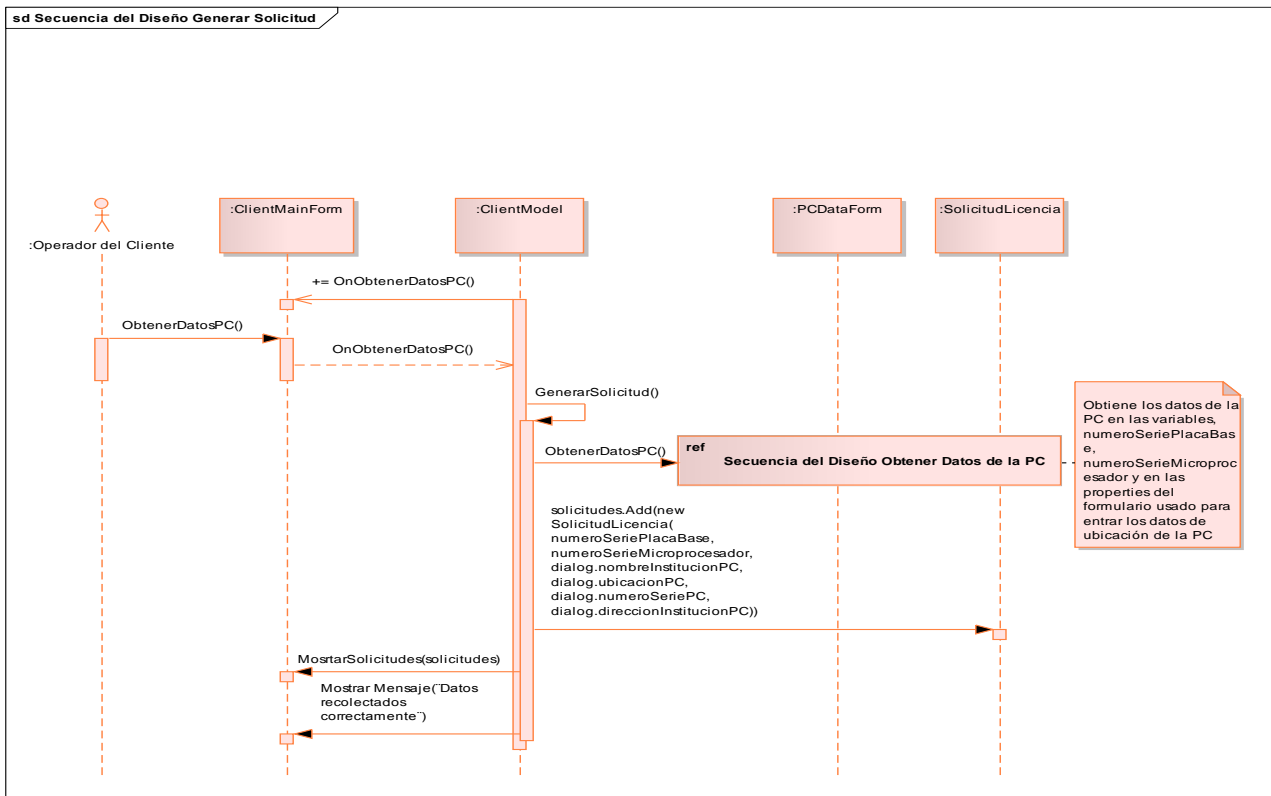


Fig.27 Diagrama de Secuencia CU Generar Solicitud

Exportar Fichero de Solicitudes de Licencias.

➤ Diagrama de Clases:

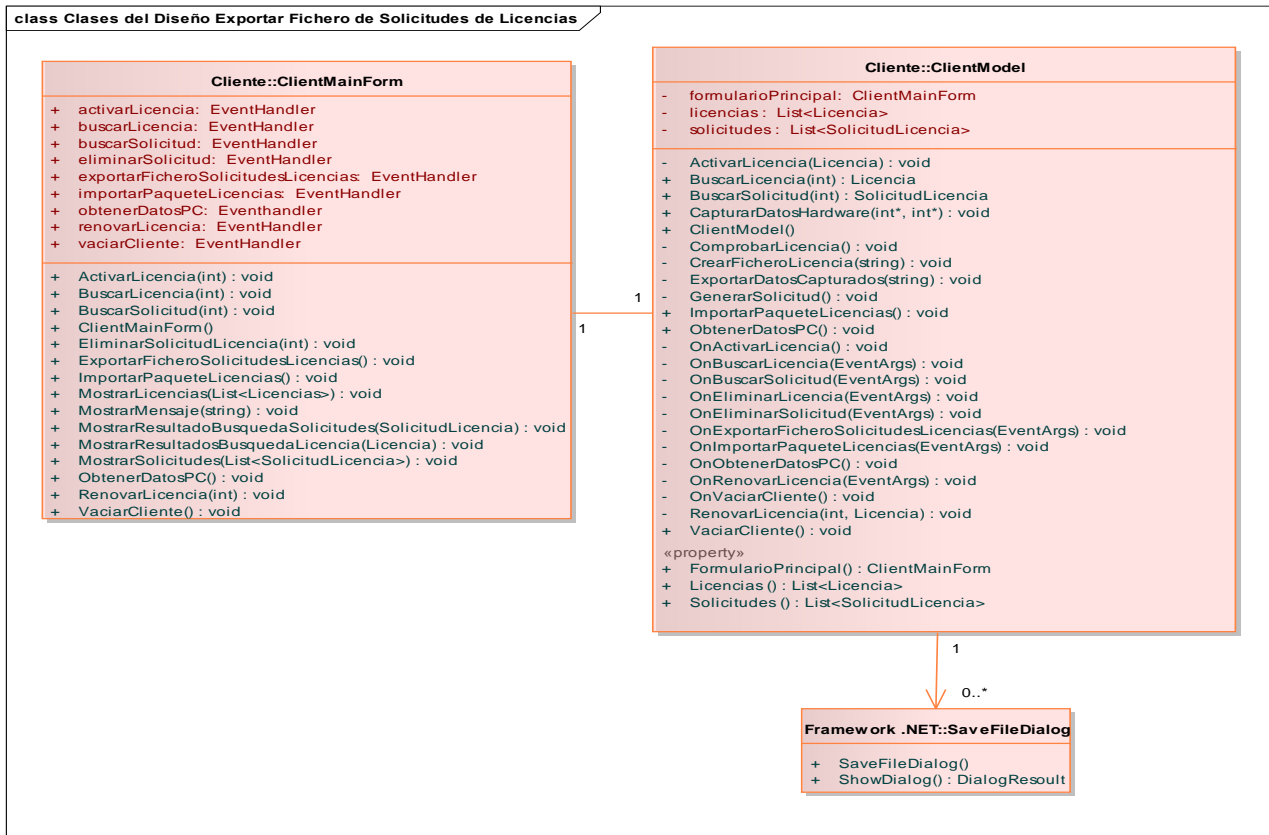


Fig.28 Diagrama de Clases CU Exportar Fichero de Solicitudes de Licencias

➤ Diagrama de Secuencia:

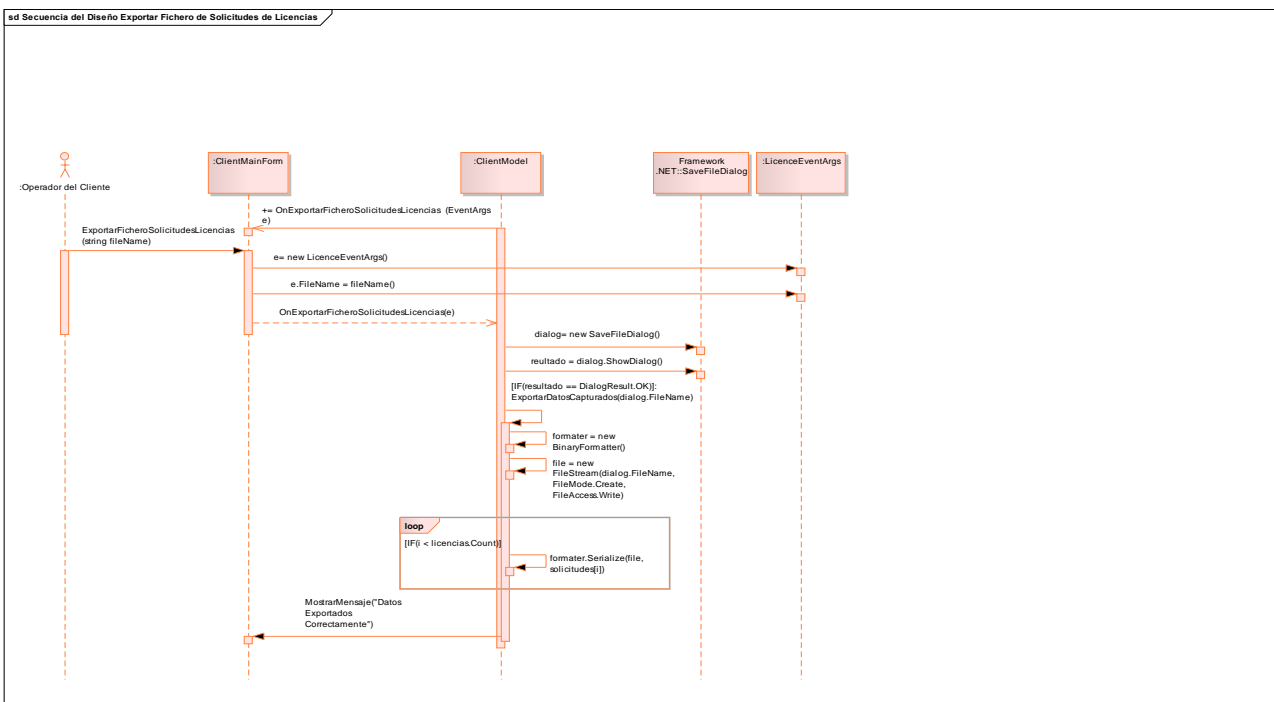


Fig.29 Diagrama de Secuencia CU Exportar Fichero de Solicitudes de Licencias

Importar Paquete de Licencias.

➤ Diagrama de Clases:

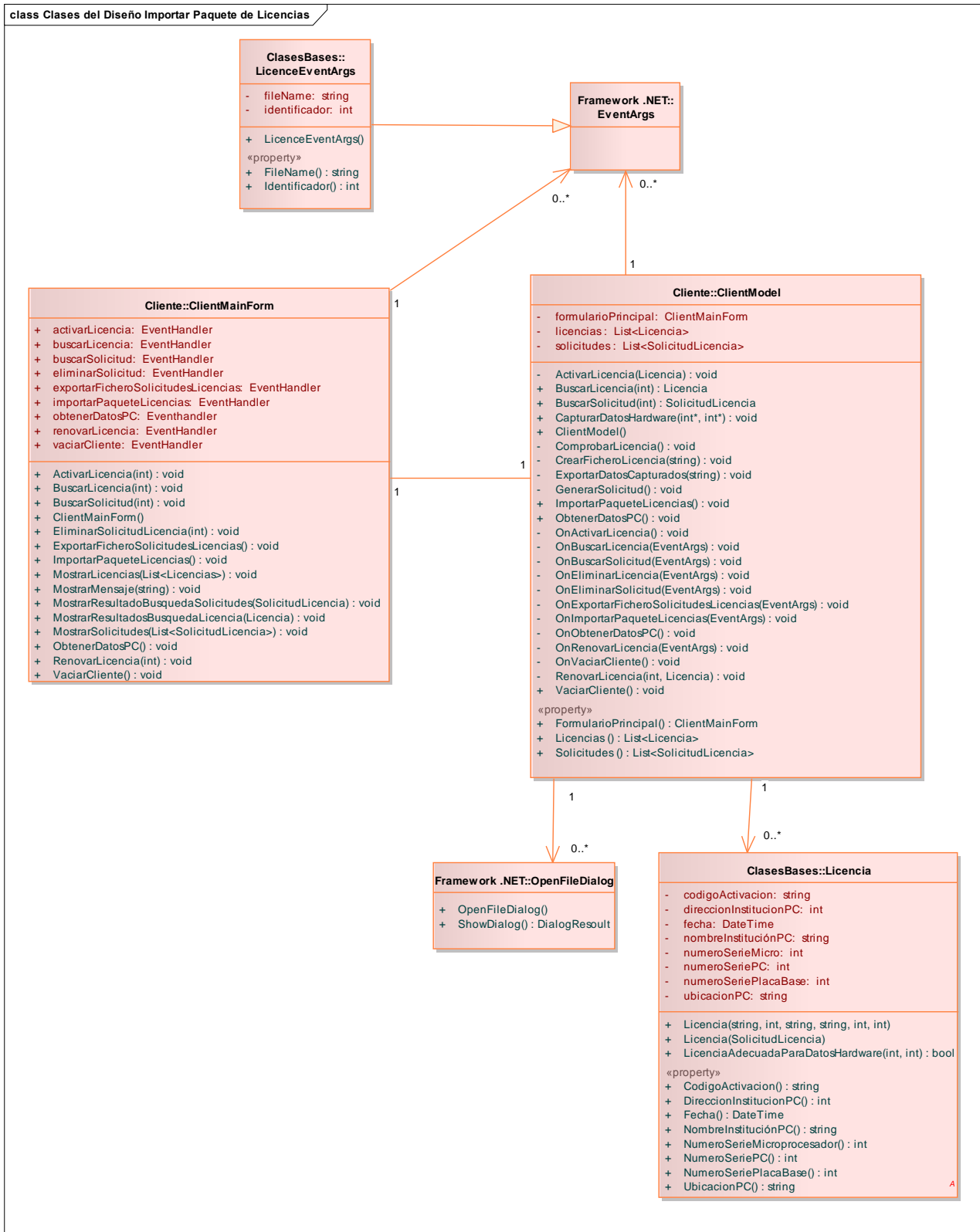


Fig.30 Diagrama de Clases CU Importar Paquete de Licencias

➤ Diagrama de Secuencia:

Capítulo 3: Arquitectura y Diseño

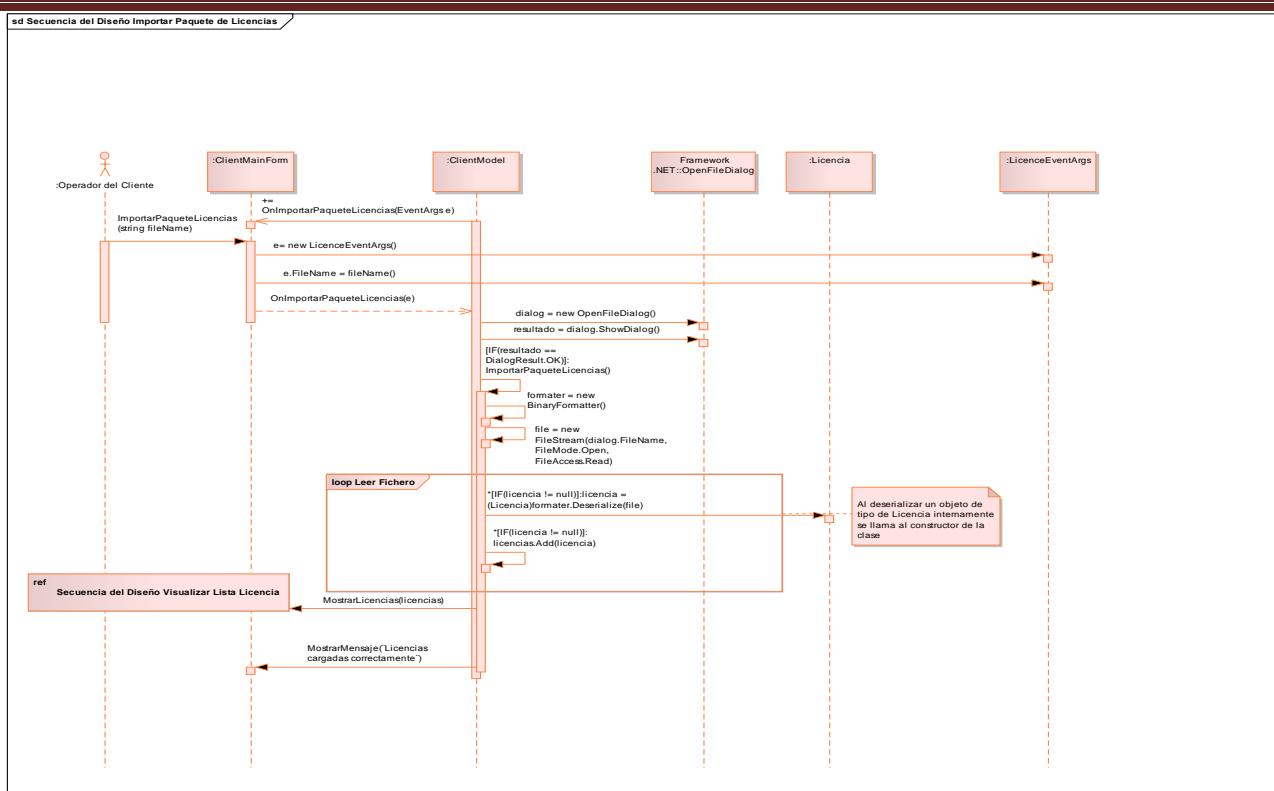


Fig.31 Diagrama de Secuencia CU Importar Paquete de Licencias Activar Licencia.

➤ Diagrama de Clases:

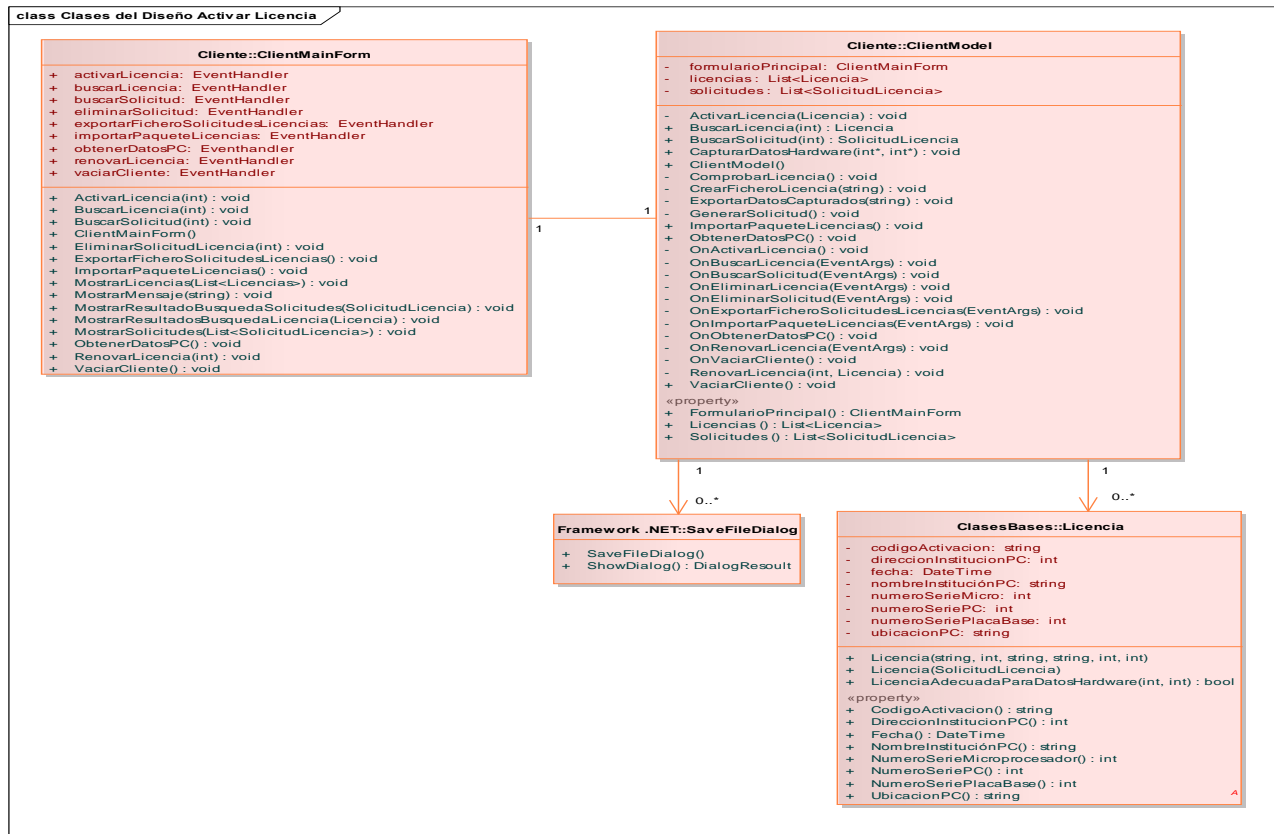


Fig.32 Diagrama de Clases CU Activar Licencia

➤ Diagrama de Secuencia:

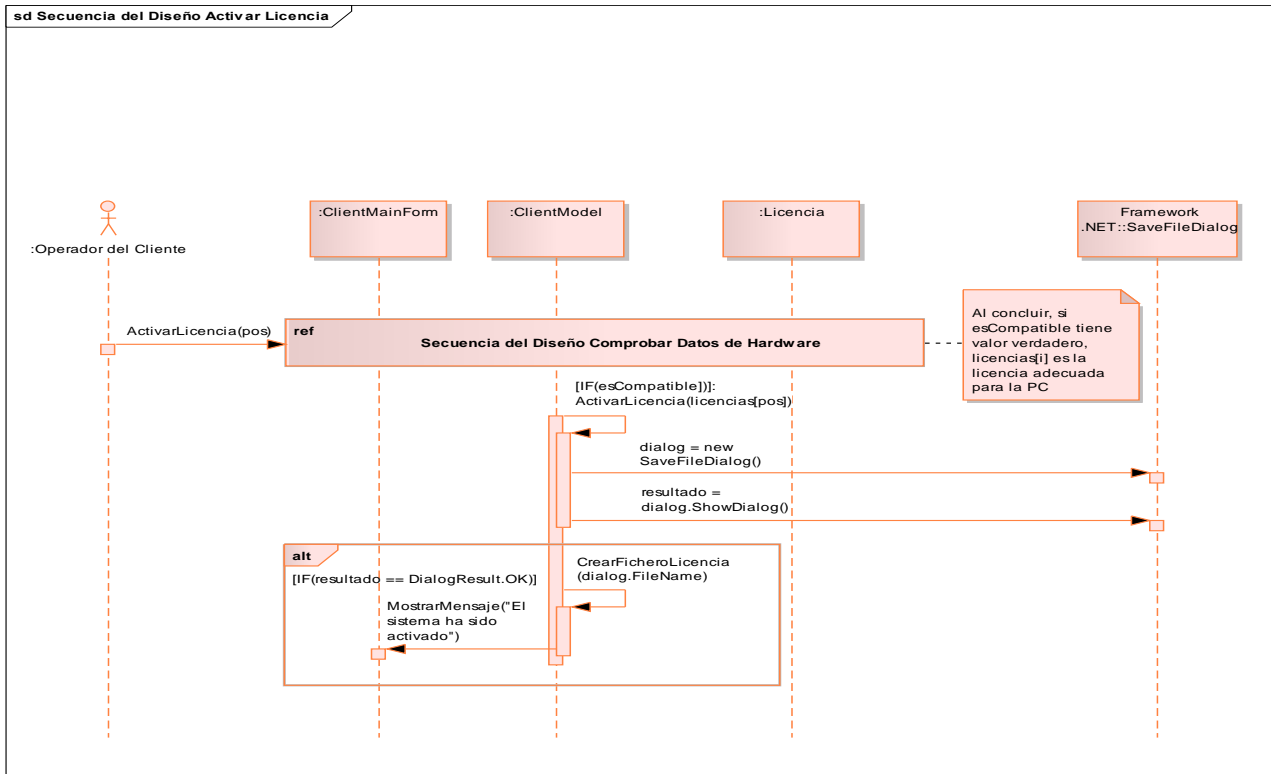


Fig.33 Diagrama de Secuencia CU Activar Licencia
Renovar Licencia.

➤ Diagrama de Clases:

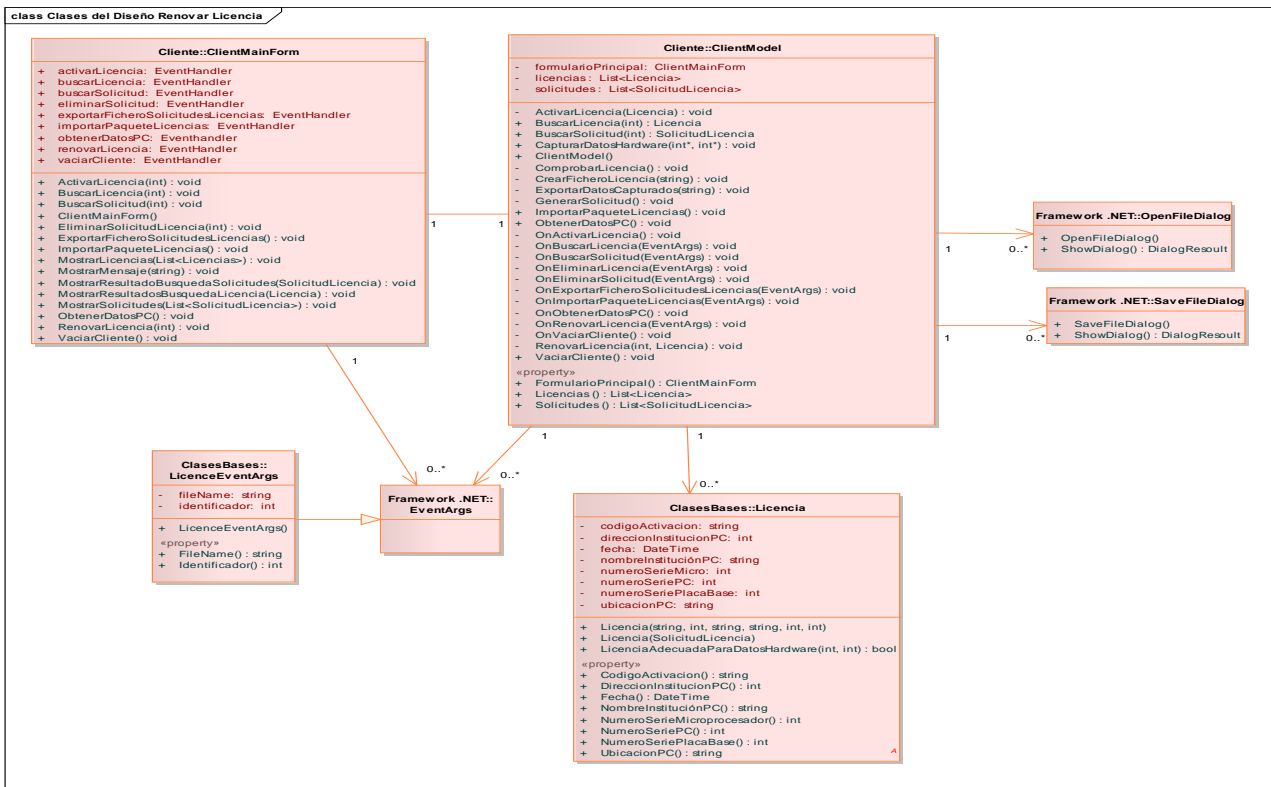


Fig.34 Diagrama de Clases CU Renovar Licencia

➤ Diagrama de Secuencia:

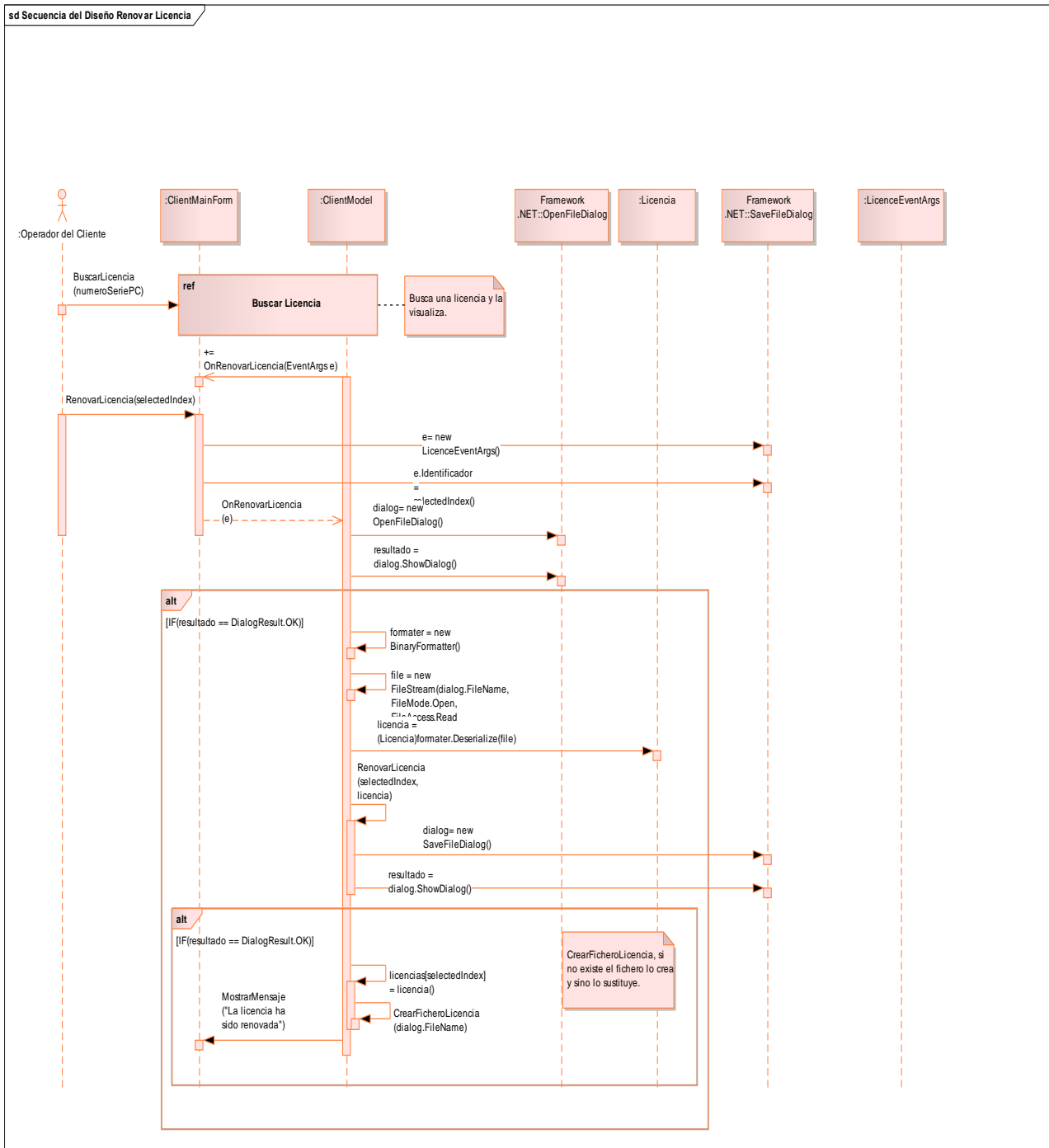


Fig.35 Diagrama de Secuencia CU Renovar Licencia

3.3.2 Diagramas del Gestor.

Importar Fichero de Solicitud.

➤ Diagrama de Clases:

Capítulo 3: Arquitectura y Diseño

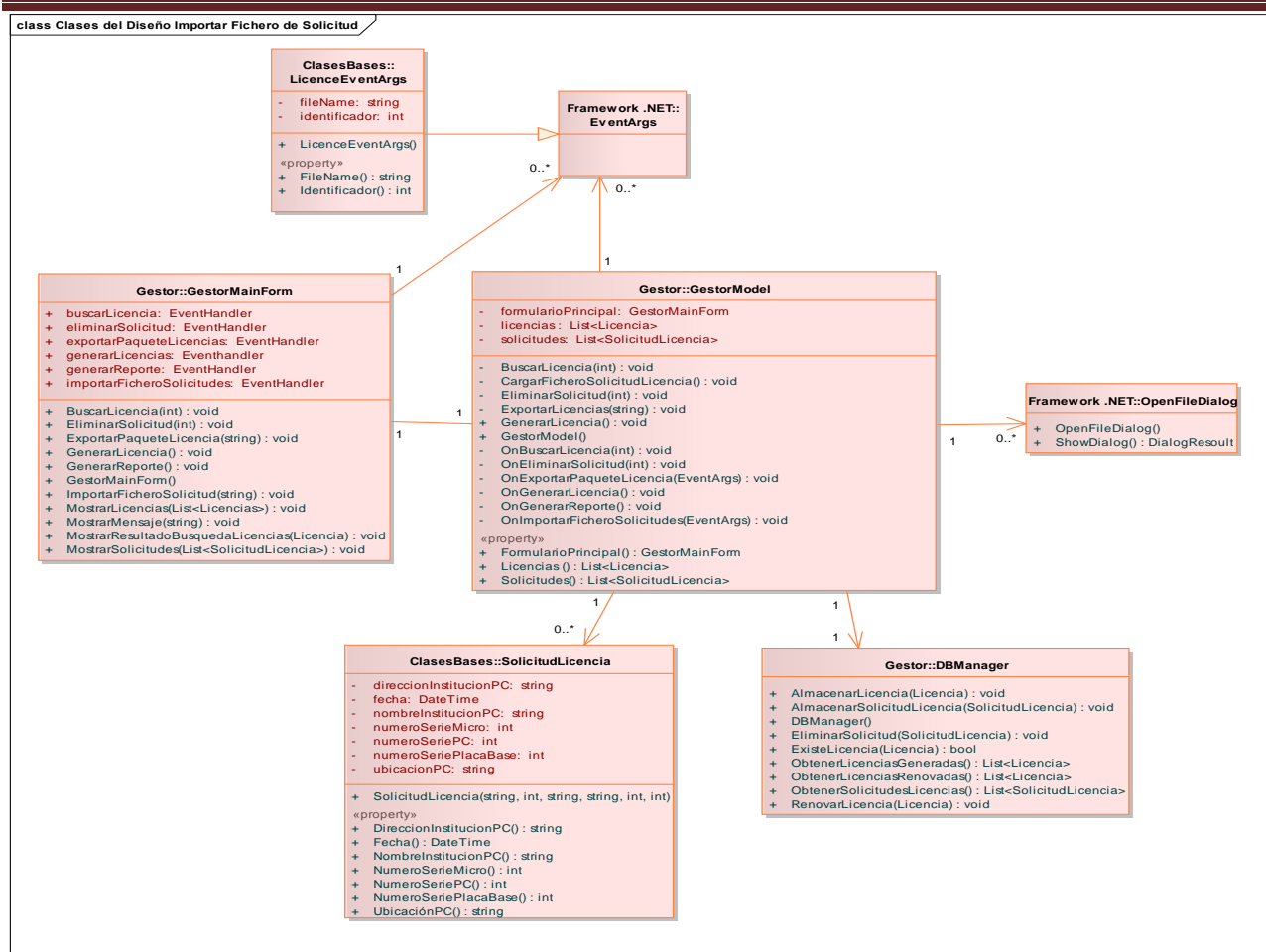


Fig.36 Diagrama de Clases CU Importar Fichero de Solicitud

➤ Diagrama de Secuencia:

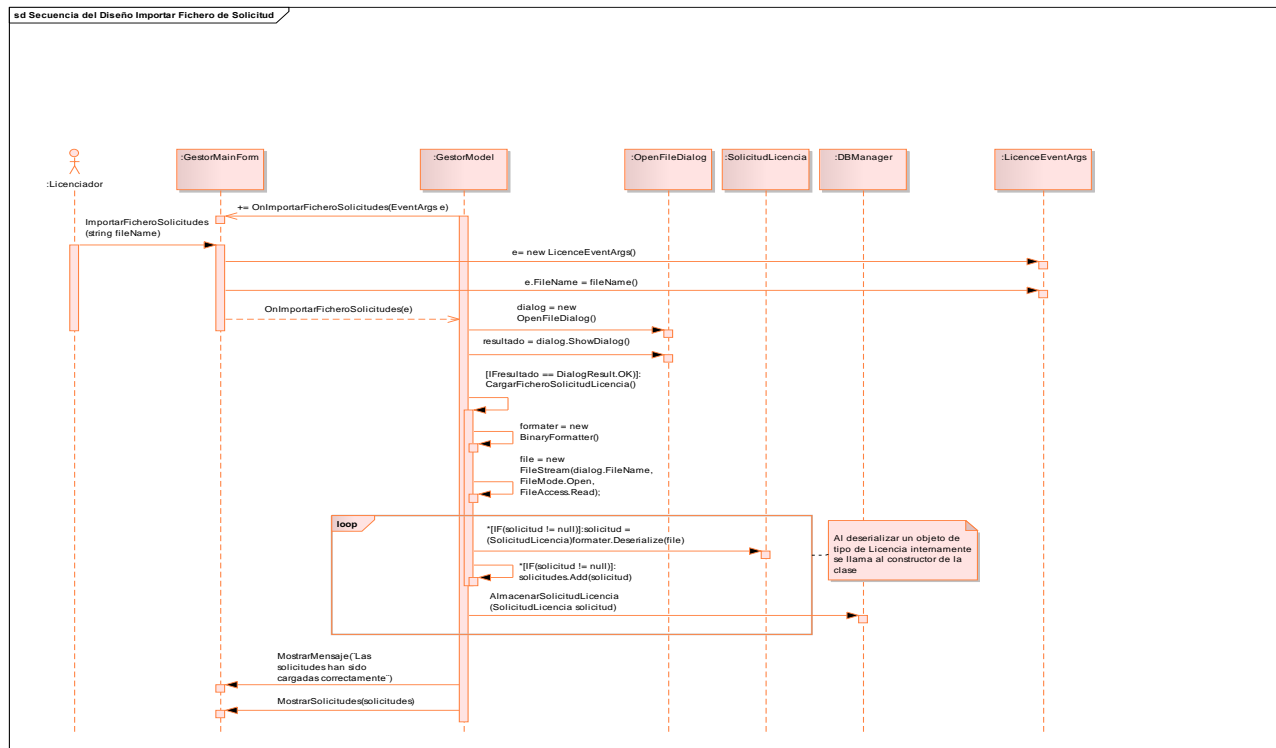


Fig.37 Diagrama de Secuencia CU Importar Fichero de Solicitud

Generar Licencia.

➤ Diagrama de Clases:

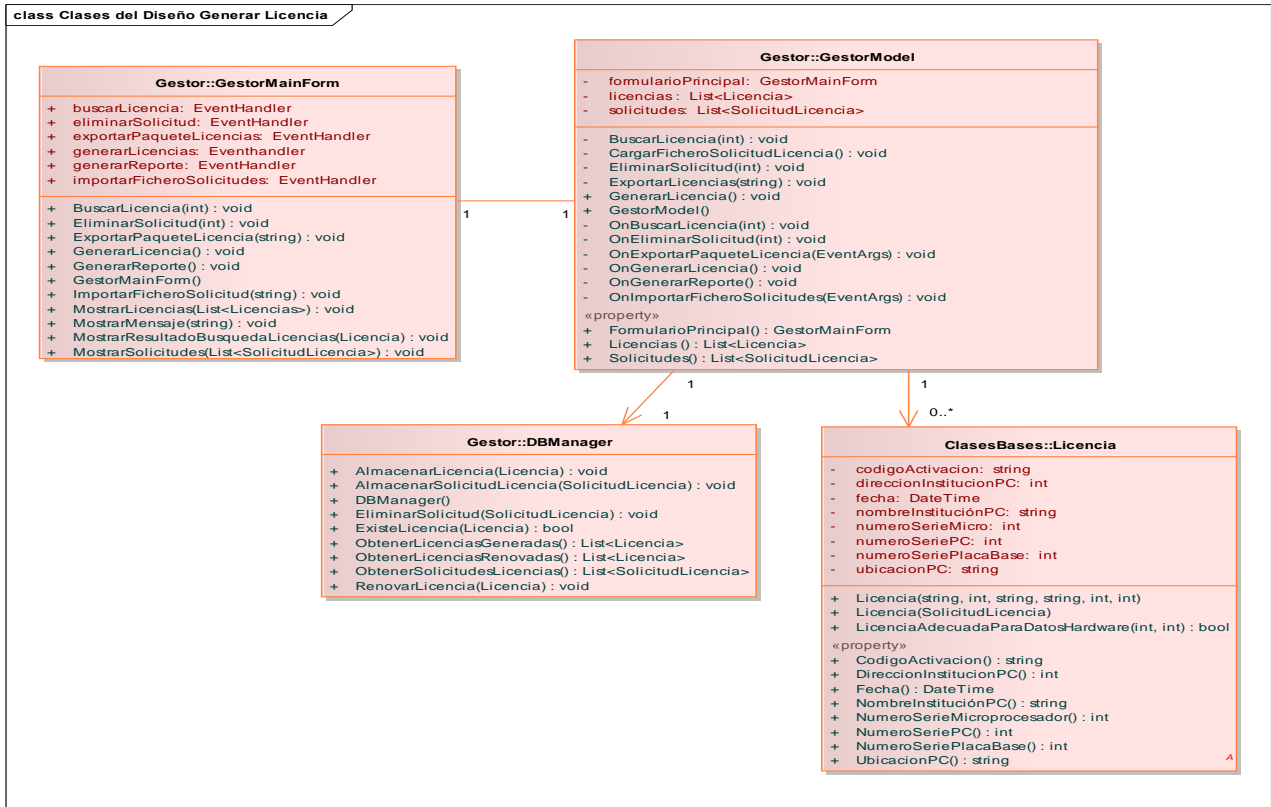


Fig.38 Diagrama de Clases CU Generar Licencia

➤ Diagrama de Secuencia:

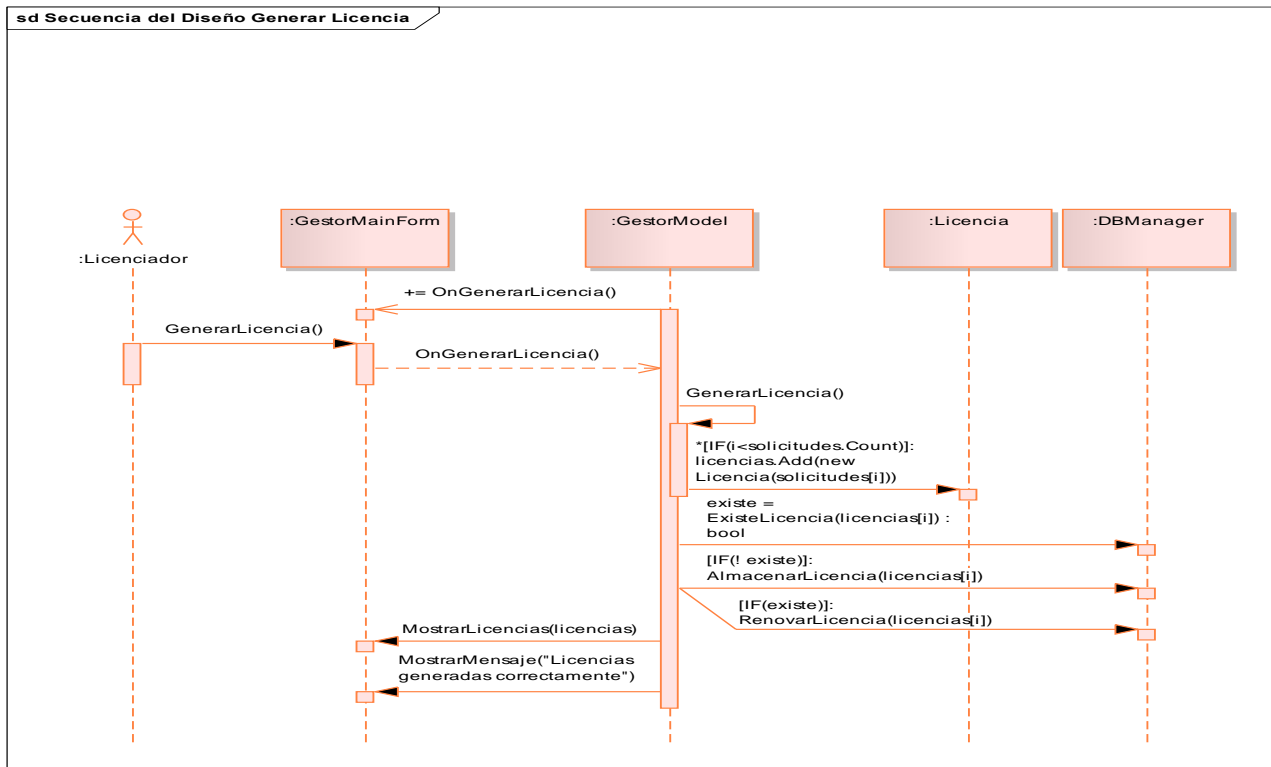


Fig.39 Diagrama de Secuencia CU Generar Licencia

Exportar Paquete de Licencias.

➤ Diagrama de Clases:

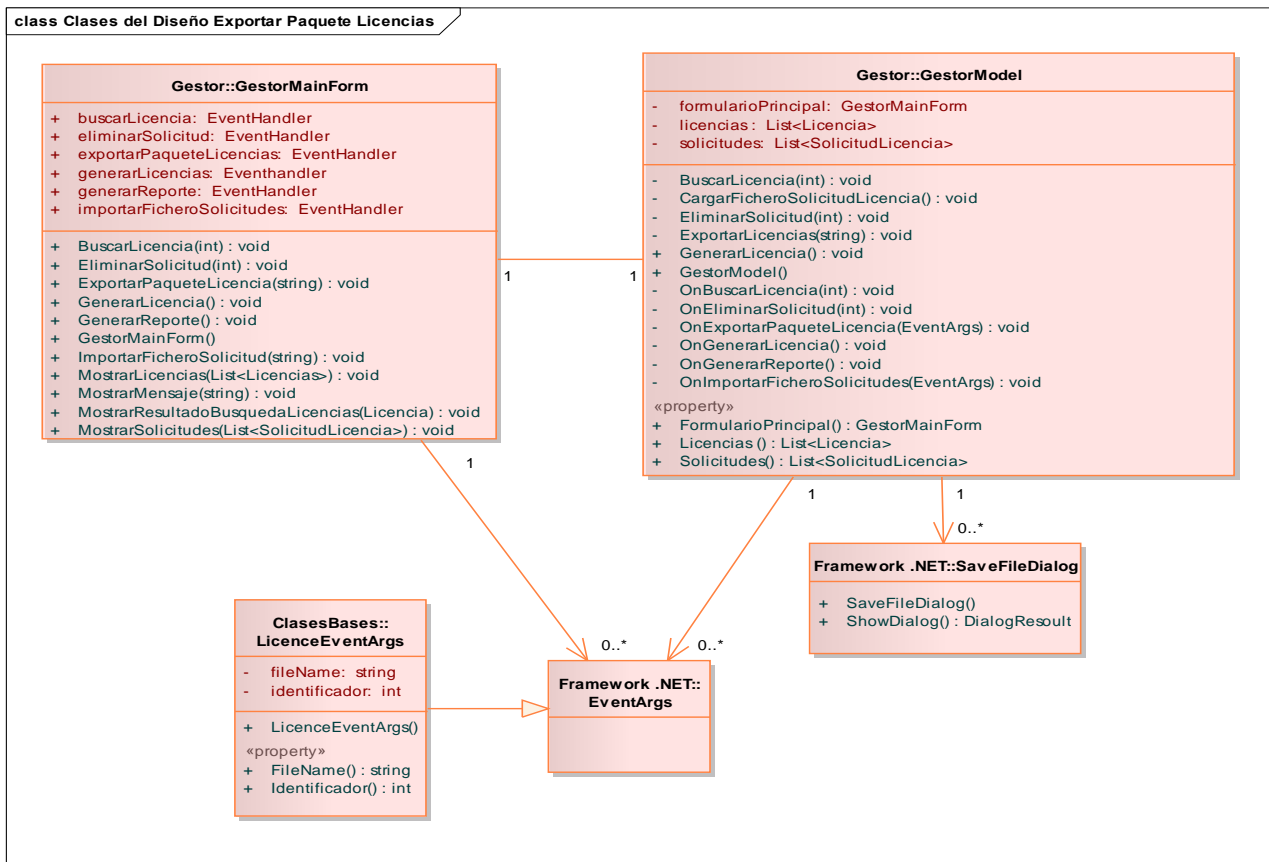


Fig.40 Diagrama de Clases CU Exportar Paquete de Licencias

➤ Diagrama de Secuencia:

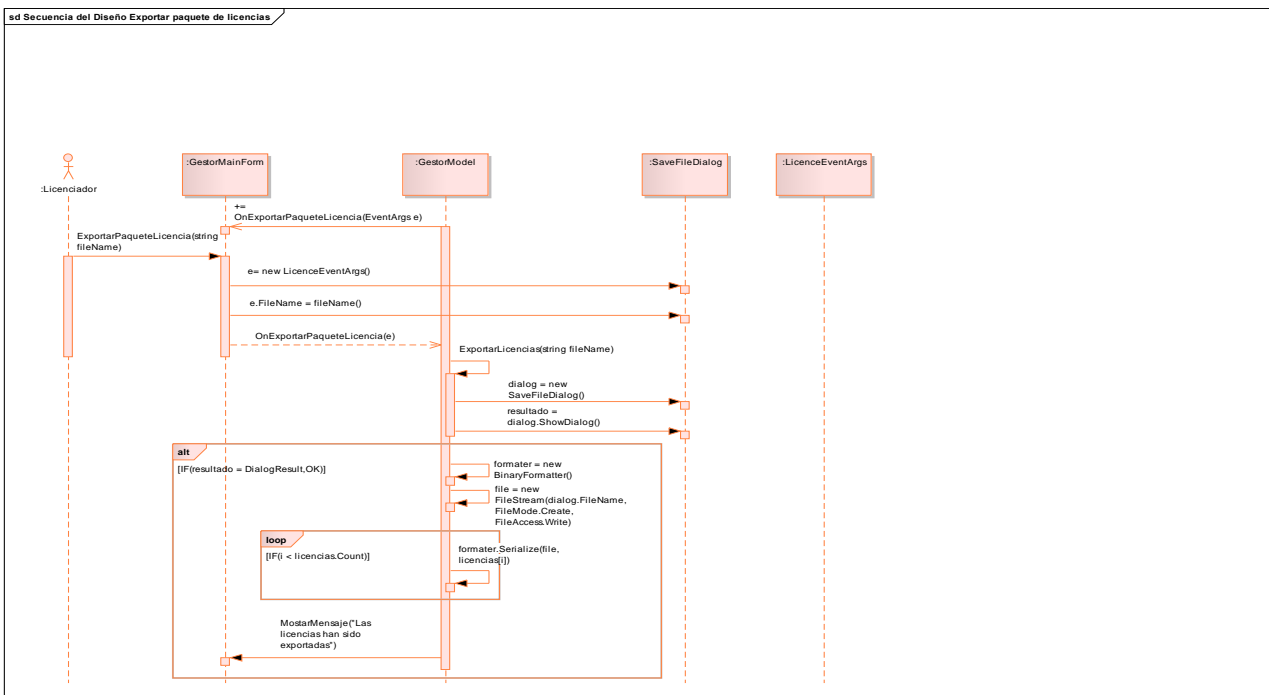


Fig.41 Diagrama de Secuencia CU Exportar Paquete de Licencias

Generar Reporte.

➤ Diagrama de Clases:

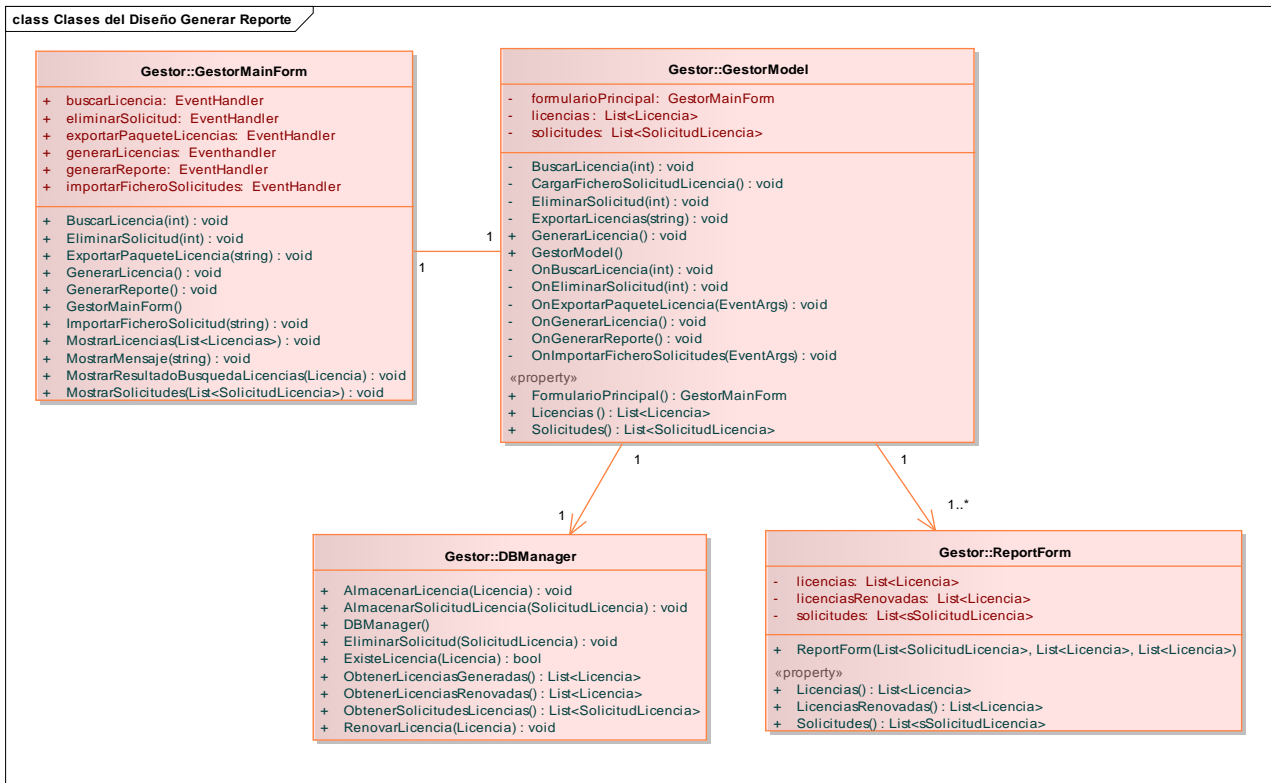


Fig.42 Diagrama de Clases CU Generar Reporte

➤ Diagrama de Secuencia:

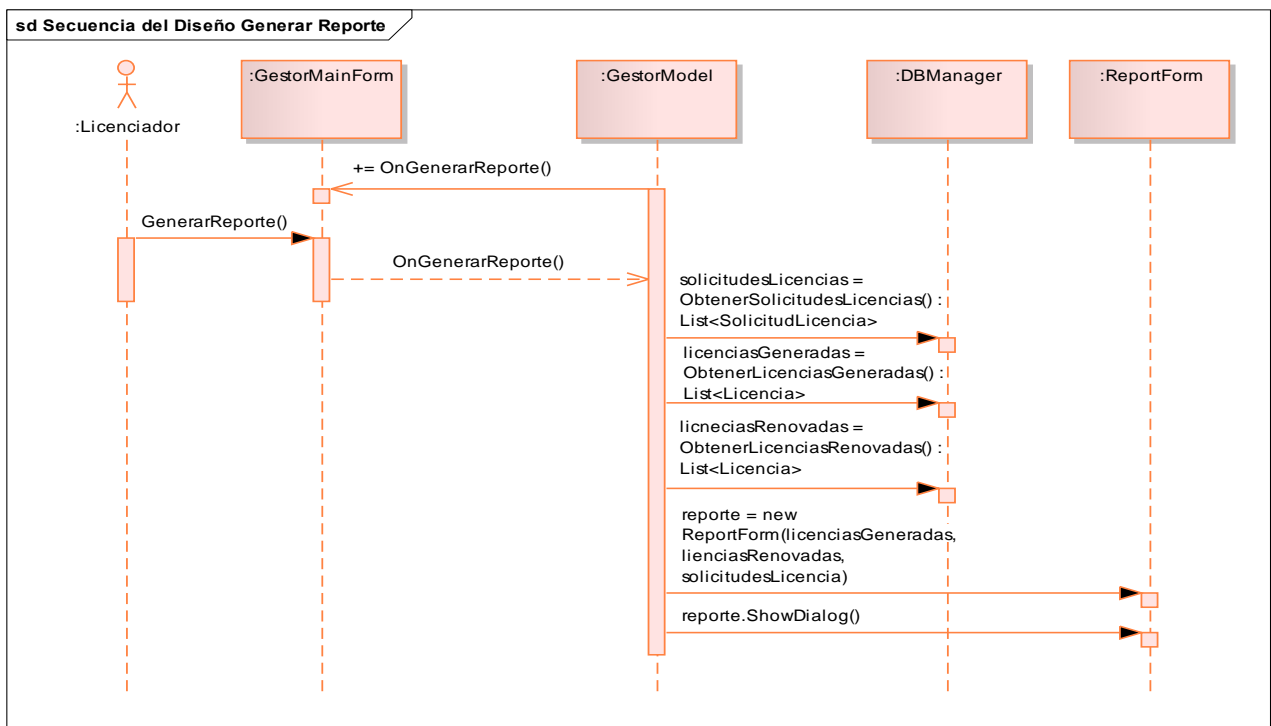


Fig.43 Diagrama de Secuencia CU Generar Reporte

Capítulo 3: Arquitectura y Diseño

3.3.3 Descripción de las Clases.

<u>Nombre: ClientMainForm</u>	
Tipo de clase: Interfaz.	
Atributo	Tipo
activarLicencia	EventHandler
buscarLicencia	EventHandler
buscarSolicitud	EventHandler
eliminarSolicitud	EventHandler
exportarFicheroSolicitudesLicencias	EventHandler
importarPaqueteLicencias	EventHandler
obtenerDatosPC	EventHandler
renovarLicencia	EventHandler
vaciarCliente	EventHandler
Para cada responsabilidad:	
Nombre:	ClientMainForm()
Descripción:	Es el constructor de la clase, permite construir objetos de la clase ClientMainForm
Nombre:	void ActivarLicencia(int pos)
Descripción:	Verifica si existe la licencia adecuada para la máquina, si esta existe la usa para activar el sistema alas PACSViewer.
Nombre:	void BuscarLicencia(int numeroSeriePC)
Descripción:	Permite buscar una licencia dado el número serie de la PC a la que corresponde.
Nombre:	void BuscarSolicitud(int numeroSeriePC)
Descripción:	Permite buscar una solicitud dado el número serie de la PC a la que corresponde.
Nombre:	void EliminarSolicitudLicencia (int index)
Descripción:	Ocurre cuando el usuario pulsa la opción "Eliminar SolicitudLicencia".
Nombre:	void ImportarPaqueteLicencias()
Descripción:	Carga las licencias contenidas en un paquete de licencias.
Nombre:	void RenovarLicencia(int index)
Descripción:	Sustituye una licencia por una nueva, que esta nueva es cargada desde un paquete de licencia.
Nombre:	void VaciarCliente()
Descripción:	Elimina todas las licencias y solicitudes de licencias en el Cliente.
Nombre:	void ExportarFicheroSolicitudesLicencias()

Capítulo 3: Arquitectura y Diseño

Descripción:	Exporta las solicitudes de licencias creadas hacia un Fichero de Solitudes de Licencias.
Nombre:	void MostrarLicencias (List<Licencias> licencias)
Descripción:	Muestra la lista de licencias cargadas del paquete de licencias.
Nombre:	void MostrarResultadosBusquedaLicencia (Licencia licencia)
Descripción:	Muestra los resultados de la búsqueda de una licencia.
Nombre:	void MostrarMensaje(string mensaje)
Descripción:	Muestra un mensaje al usuario.
Nombre:	void MostrarResultadoBusquedaSolicitudes (SolicitudLicencia solicitud)
Descripción:	Muestra los resultados de la búsqueda de una solicitud de licencia.
Nombre:	Void MostrarResultadoBusquedaLicencias(List<SolicitudLicencia> solicitudes)
Descripción:	Muestra una lista con todas las solicitudes de licencias generadas para cada PC.
Nombre:	void MostrarSolicitudes (List<SolicitudLicencia> solicitudes)
Descripción:	Muestra la lista de solicitudes.
Nombre:	void ObtenerDatosPC()
Descripción:	Obtiene todos los datos de una PC que son necesario para generar una solicitud de licencia.
<u>Nombre: ClientModel</u>	
Tipo de clase: Controladora.	
Atributo	Tipo
formularioPrincipal	ClientMainForm
licencias	List<Licencia>
solicitudes	List<SolicitudLicencia>
Para cada responsabilidad:	
Nombre:	void ActivarLicencia(Licencia licencia)
Descripción:	Verifica si existe la licencia adecuada para la máquina, si esta existe la usa para activar el sistema alas PACSViewer.
Nombre:	Void CapturarDatosHardware(int out numeroSeriePlacaBase, int out numeroSerieMicroprocesador)
Descripción:	Captura datos de hardware de la PC que son únicos para cada una de ellas.
Nombre:	Licencia BuscarLicencia(int numeroSeriePC)
Descripción:	Permite buscar una licencia dado el número serie de la PC a la que

Capítulo 3: Arquitectura y Diseño

	corresponde.
Nombre:	SolicitudLicencia BuscarSolicitud(int numeroSeriePC)
Descripción:	Permite buscar una solicitud dado el número serie de la PC a la que corresponde.
Nombre:	ClientModel()
Descripción:	Es el constructor de la clase, permite construir objetos de la clase ClientModel.
Nombre:	void ComprobarLicencia()
Descripción:	Comprueba si una licencia es compatible con los datos de hardware de la PC.
Nombre:	void ExportarDatosCapturados (string fileName)
Descripción:	Exporta las solicitudes de licencias creadas hacia un Fichero de Solitudes de Licencias.
Nombre:	Void CrearFicheroLicencia(string path)
Descripción:	Crea un fichero con una única licencia, el cuál será utilizado por el sistema alas PACSViewer para activarse.
Nombre:	void GenerarSolicitud()
Descripción:	Permite crear una solicitud a partir de los datos de hardware y ubicación recolectados para la PC.
Nombre:	void ImportarPaqueteLicencias()
Descripción:	Carga las licencias contenidas en un paquete de licencias.
Nombre:	void ObtenerDatosPC()
Descripción:	Obtiene todos los datos de una PC que son necesario para generar una solicitud de licencia.
Nombre:	void OnExportarFicheroSolicitudesLicencias()
Descripción:	Ocurre cuando el usuario pulsa la opción "Exportar Fichero de Solicitudes de Solicitudes de Licencias".
Nombre:	void OnActivarLincencia()
Descripción:	Ocurre cuando el usuario pulsa la opción "Activar Licencia".
Nombre:	void ImportarPaqueteLicencias()
Descripción:	Carga las licencias contenidas en un paquete de licencia.
Nombre:	void OnBuscarLicencia (EventArgs e)
Descripción:	Ocurre cuando el usuario pulsa la opción "Buscar Licencia".
Nombre:	Void OnBuscarSolicitud(EventArgs e)
Descripción:	Ocurre cuando el usuario pulsa la opción "Buscar Solicitud".
Nombre:	void OnEliminarLicencia(EventArgs e)
Descripción:	Ocurre cuando el usuario pulsa la opción "Eliminar Licencia".
Nombre:	void OnEliminarSolicitud(EventArgs e)

Capítulo 3: Arquitectura y Diseño

Descripción:	Ocurre cuando el usuario pulsa la opción "Eliminar Solicitud".
Nombre:	void OnImportarPaqueteLicencias()
Descripción:	Ocurre cuando el usuario pulsa la opción "Importar Paquete de Licencias".
Nombre:	void OnObtenerDatosPC()
Descripción:	Ocurre cuando el usuario pulsa la opción "Obtener Datos".
Nombre:	void OnRenovarLicencia(EventArgs e)
Descripción:	Ocurre cuando el usuario pulsa la opción "Renovar Licencia".
Nombre:	void OnVaciarCliente()
Descripción:	Ocurre cuando el usuario pulsa la opción "Vaciar Cliente".
Nombre:	void RenovarLicencia(int index, Licencia nueva)
Descripción:	Sustituye una licencia por una nueva, que esta nueva es cargada desde un paquete de licencia.
Nombre:	void VaciarCliente()
Descripción:	Elimina todas las licencias y solicitudes de licencias en el Cliente.
<u>Nombre: PCDataForm</u>	
Tipo de clase: Interfaz.	
Atributo	Tipo
nombreIntitucionPC	string
numeroSeriePC	int
ubicacionPC	string
direccionInstitucionPC	string
Para cada responsabilidad:	
Nombre:	void PCDataForm(string direccionInstitucionPC, string nombreIntitucionPC, int numeroSeriePC, string ubicacionPC)
Descripción:	Es el constructor de la clase, permite construir objetos de la clase PCDataForm.
Nombre:	void ObtenerDatosPC (string nombreInstitucionPC, string ubicacionPC, int numeroSeriePC, string direccionInstitucionPC)
Descripción:	Permite recolectar los datos entrados por el usuario.
<u>Nombre: GestorMainForm</u>	
Tipo de clase: Interfaz.	
Atributo	Tipo
buscarLicencia	EventHandler
eliminarSolicitud	EventHandler
exportarPaqueteLicencias	EventHandler
generarLicencias	EventHandler

Capítulo 3: Arquitectura y Diseño

generarReporte	EventHandler
importarFicheroSolicitudes	EventHandler
Para cada responsabilidad:	
Nombre:	GestorMainForm ()
Descripción:	Es el constructor de la clase, permite construir objetos de la clase GestorMainForm.
Nombre:	void BuscarLicencia (int numeroSeriePC)
Descripción:	Permite buscar una licencia dado el número de serie de la PC a la que corresponde.
Nombre:	void EliminarSolicitud(index: int)
Descripción:	Permite eliminar una solicitud dado la posición que ocupa en la lista de solicitudes.
Nombre:	void ExportarPaqueteLicencia(string fileName)
Descripción:	Permite exportar las licencias generadas hacia un paquete de licencias.
Nombre:	void GenerarReporte()
Descripción:	Permite generar un reporte que contiene todas las solicitudes de licencias, las licencias generadas y las licencias renovadas.
Nombre:	void GenerarLicencia ()
Descripción:	Permite generar una licencia asociada a cada solicitud de licencia.
Nombre:	void ImportarFicheroSolicitud (string fileName)
Descripción:	Permite cargar las solicitudes contenidas en un fichero de solicitudes de licencias.
Nombre:	void MostrarLicencias (List<Licencias> licencias)
Descripción:	Muestra la lista de licencias.
Nombre:	void MostrarMensaje (string mensaje)
Descripción:	Muestra un mensaje al usuario.
Nombre:	void MostrarResultadoBusquedaLicencias (Licencia licencia)
Descripción:	Muestra la licencia que se obtuvieron en la búsqueda.
Nombre:	void MostrarSolicitudes (List<SolicitudLicencia> solicitudes)
Descripción:	Muestra la lista de solicitudes.
<u>Nombre: GestorModel</u>	
Tipo de clase: Controladora.	
Atributo	Tipo
formularioPrincipal	GestorMainForm
licencias	List<Licencia>

Capítulo 3: Arquitectura y Diseño

solicitudes	List<SolicitudLicencia>
Para cada responsabilidad:	
Nombre:	void BuscarLicencia (int numeroSeriePC)
Descripción:	Permite buscar una licencia dado el número serie de la PC a la que corresponde.
Nombre:	void CargarFicheroSolicitudLicencia ()
Descripción:	Carga las solicitudes de licencias contenidas en un fichero de solicitudes de licencias.
Nombre:	Void EliminarSolicitud (int index)
Descripción:	Elimina una solicitud de licencia del fichero de solicitudes de licencias.
Nombre:	GestorModel ()
Descripción:	Es el constructor de la clase, permite construir objetos de la clase GestorModel.
Nombre:	void ExportarLicencias(string fileName)
Descripción:	Crea un paquete de licencia que contiene las licencias generadas.
Nombre:	void GenerarLicencia()
Descripción:	Permite crear una licencia para cada solicitud de licencia.
Nombre:	void On BuscarLicencia (int numeroSeriePC)
Descripción:	Ocurre cuando el usuario pulsa la opción "Buscar Solicitud".
Nombre:	void OnEliminarSolicitud(int index)
Descripción:	Ocurre cuando el usuario pulsa la opción "Eliminar Solicitud".
Nombre:	void OnExportarPaqueteLicencia (EventArgs e)
Descripción:	Ocurre cuando el usuario pulsa la opción "Exportar Paquete de Licencia".
Nombre:	void OnGenerarReporte()
Descripción:	Ocurre cuando el usuario pulsa la opción "Generar Reporte".
Nombre:	void OnGenerarLicencia()
Descripción:	Ocurre cuando el usuario pulsa la opción "Generar Licencia".
Nombre:	void OnImportarFicheroSolicitudes(EventArgs e)
Descripción:	Ocurre cuando el usuario pulsa la opción "Importar Fichero de Solicitud".
<u>Nombre: ReportForm</u>	
Tipo de clase: Interfaz.	
Atributo	Tipo
licencias	List<Licencia>
licenciasRenovadas	List<Licencia>
solicitudes	List<SolicitudLicencia>
Para cada responsabilidad:	

Capítulo 3: Arquitectura y Diseño

Nombre:	void ReportForm (List<Licencias licenciasGeneradas>, List<Licencias licenciasRenovadas>, List<SolicitudesLicencias>solicitudesLicencias)
Descripción:	Es el constructor de la clase, permite construir objetos de la clase PCDataForm.
<u>Nombre: DBManager</u>	
Para cada responsabilidad:	
Nombre:	void AlmacenarLicencia (Licencia licencia)
Descripción:	Guarda una licencia en la base de datos.
Nombre:	Void AlmacenarSolicitudLicencia(solicitud SolicitudLicencia)
Descripción:	Guarda una solicitud en la base de datos.
Nombre:	DBManager ()
Descripción:	Es el constructor de la clase, permite construir objetos de la clase DBManager.
Nombre:	void EliminarSolicitud (SolicitudLicencia solicitud)
Descripción:	Elimina una solicitud de la base de datos.
Nombre:	bool ExisteLicencia (Licencia licencia)
Descripción:	Busca en la base de datos si existe o no una licencia determinada.
Nombre:	List<Licencia> ObtenerLicenciasGeneradas ()
Descripción:	Obtiene una lista con las licencias que el Gestor ha generado.
Nombre:	List<Licencia> ObtenerLicenciasRenovadas ()
Descripción:	Obtiene una lista con las licencias que han sido renovadas.
Nombre:	List<SolicitudLicencia> ObtenerSolicitudesLicencias ()
Descripción:	Obtiene una lista con las solicitudes de licencias que están almacenadas en la base de datos del Gestor.
Nombre:	void RenovarLicencia (Licencia licencia)
Descripción:	Guarda una licencia en la base de datos como renovada.
<u>Nombre: Licencia</u>	
Tipo de clase: Entidad.	
Atributo	Tipo
nombreInstitucionPC	string
numeroSeriePC	int
ubicacionPC	string
numeroSerieMicro	int
numeroSeriePlacaBase	int
codigoActivacion	string
fecha	DateTime

Capítulo 3: Arquitectura y Diseño

direccionInstitucionPC	string
Para cada responsabilidad:	
Nombre:	Licencia (int numeroSeriePlacaBase, int numeroSerieMicroprocesador, string nombreInstitucionPC, int ubicacionPC, string numeroSeriePC, DateTime fecha, string direccionInstitucionPC)
Descripción:	Es el constructor de la clase, permite construir objetos de la clase Licencia a partir de los datos de la PC para la cual se genera la licencia.
Nombre:	Licencia(solicitud SolicitudLicencia)
Descripción:	Es el constructor de la clase, permite construir objetos de la clase Licencia a partir de una solicitud de licencia previamente creada.
Nombre:	bool LicenciaAdecuadaParaDatosHardware (int numeroSeriePlacaBase, int numeroSerieMicroprocesador)
Descripción:	Verifica si la licencia es la adecuada para la PC en la cual se quiere activar el sistema alas PACSViewer.
<u>Nombre: SolicitudLicencia</u>	
Tipo de clase: Entidad.	
Atributo	Tipo
nombreInstitucionPC	string
numeroSeriePC	int
ubicacionPC	string
numeroSerieMicro	int
numeroSeriePlacaBase	int
fecha	DateTime
direccionInstitucionPC	string
Para cada responsabilidad:	
Nombre:	SolicitudLicencia (int numeroSeriePlacaBase, int numeroSerieMicroprocesador, string nombreInstitucionPC, int ubicacionPC, string numeroSeriePC, string direccionInstitucionPC, DateTime fecha)
Descripción:	Es el constructor de la clase, permite construir objetos de la clase Licencia.
Nombre: LicenceEventArgs	
Tipo de clase: Entidad.	
Atributo	Tipo
FileName	string
Identificador	int
Para cada responsabilidad:	

Capítulo 3: Arquitectura y Diseño

Nombre:	LicenceEventArgs()
Descripción:	Es el constructor de la clase, permite construir objetos de la clase LicenceEventArgs.

3.3.4 Diagrama de Clases del Diseño

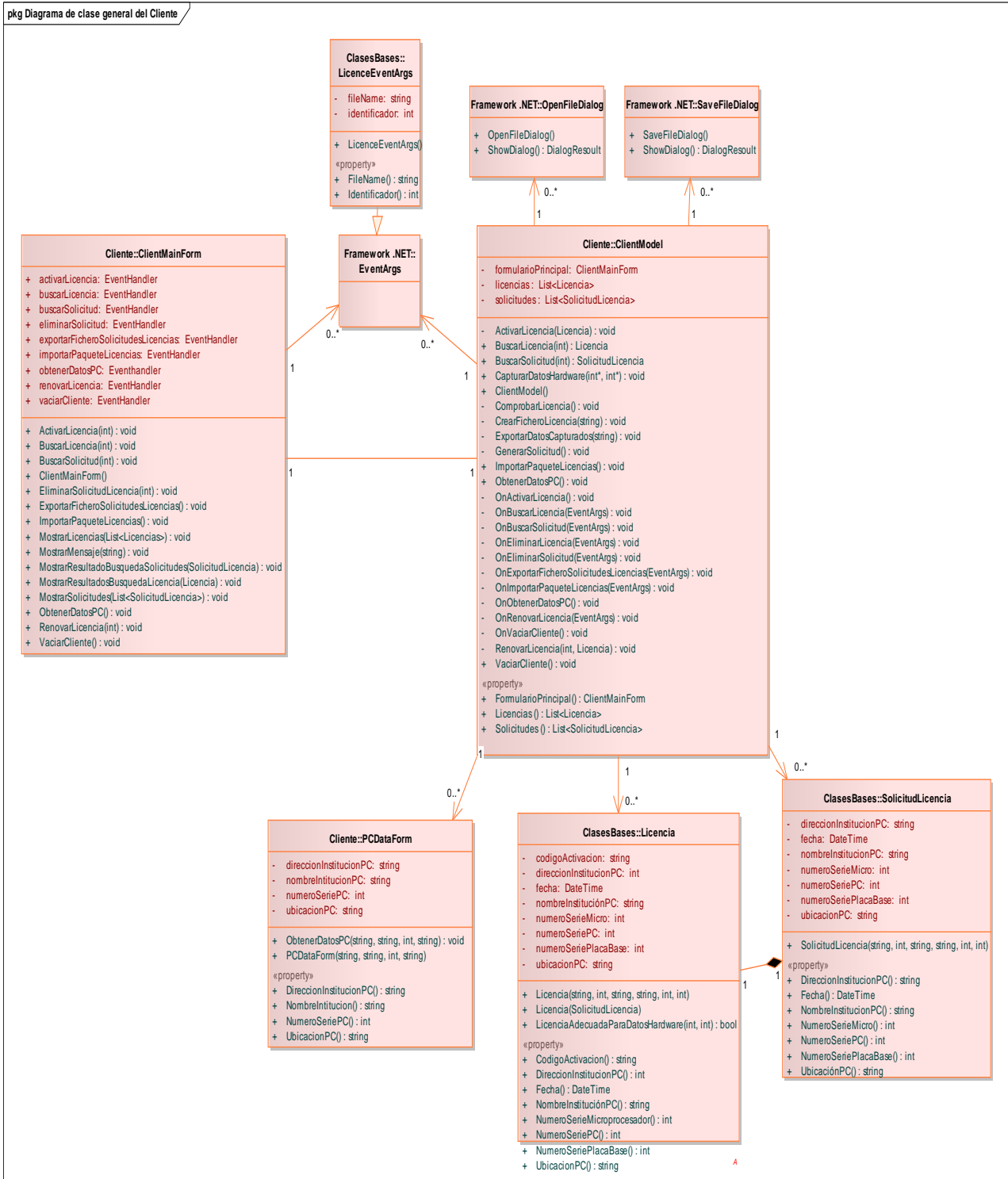


Fig.44 Diagrama de Clases del Diseño del Cliente

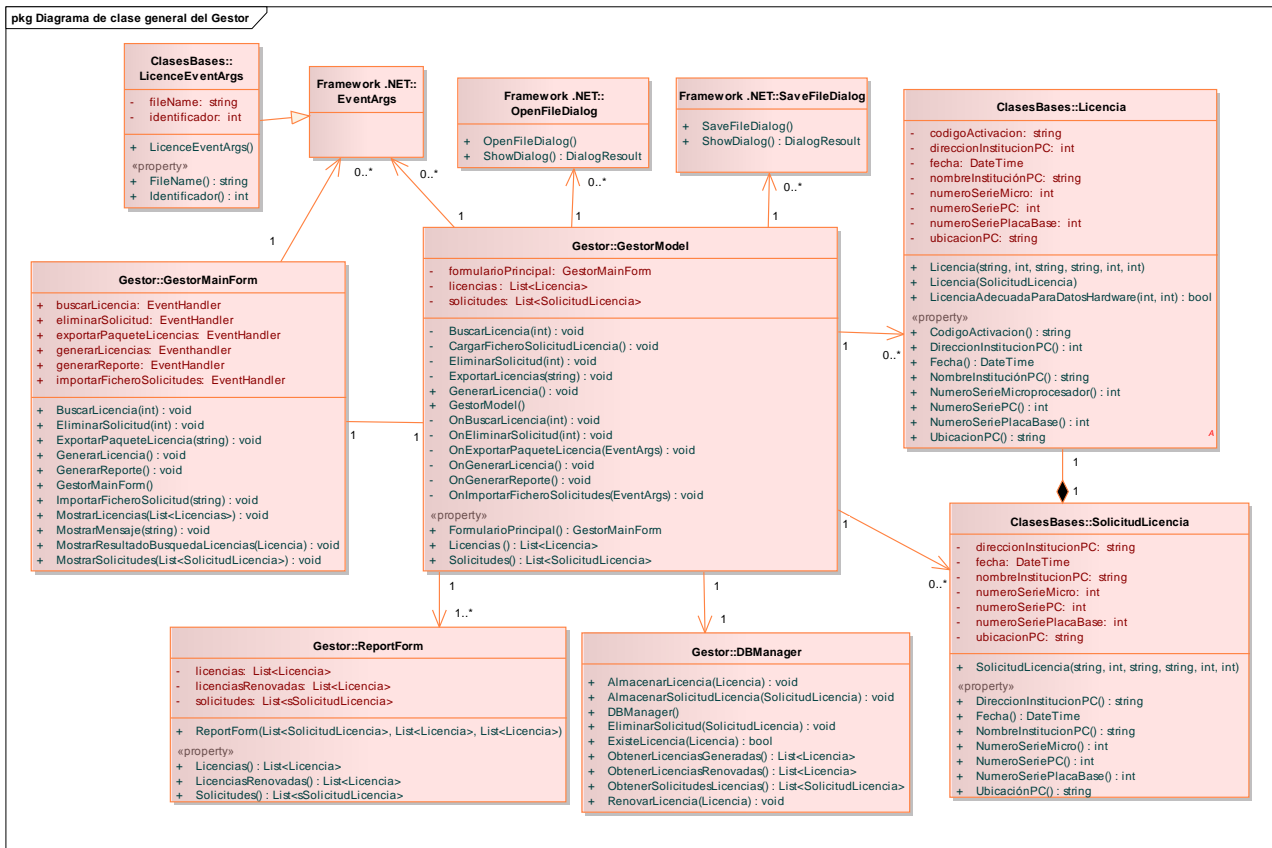


Fig.45 Diagrama de Clases del Diseño del Gestor

3.4 Modelo de datos.

Un modelo de datos es un conjunto de conceptos, reglas y convenciones que nos permiten describir y en ocasiones manipular los datos de un cierto mundo real que deseamos almacenar en la base de datos. Al producto del modelo de datos se le llama esquema (descripción de la estructura de la base de datos) y a los datos en concreto almacenados en la base de datos en ese momento, ocurrencia del esquema.

En la la **¡Error! No se encuentra el origen de la referencia.** se muestra el modelo de datos diseñado (Modelo Relacional), para almacenar los datos que se manejan. En el mismo se establece las relaciones entre los datos que estarán guardados en las tablas, y a través de dichas interconexiones estarán relacionados los datos de ambas tablas. Se diseñaron 5 tablas (Institución, Ubicación, PC, SolicitudLicencia y Licencia) cada una con sus datos correspondientes. Se definió la tabla Ubicación como hija de la tabla Institución, la tabla PC como hija de la tabla Ubicación, la tabla SolicitudLicencia como hija de la tabla PC y la tabla Licencia como hija de la tabla Solicitud Licencia. Las relaciones entre las tablas padres y su hijo se llevaron a cabo por medio de las claves primarias y foráneas. Las claves primarias son la clave

Capítulo 3: Arquitectura y Diseño

principal del registro dentro de cada una de las tablas. Las claves foráneas se colocaron en la tabla hija, pues contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

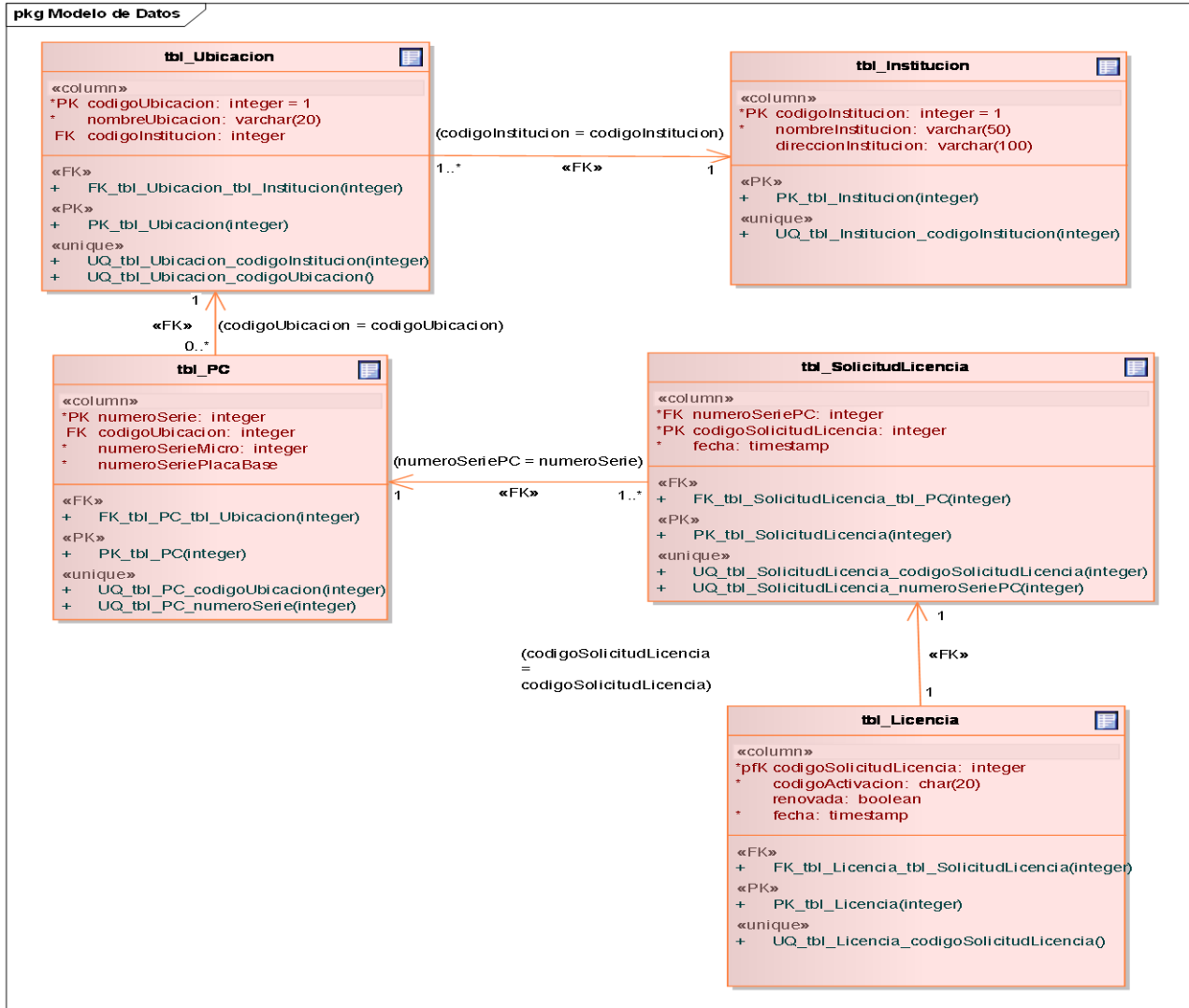


Fig.46 Diagrama del Diseño de la Base de Datos

3.4.1 Descripción de las Tablas.

<u>Nombre: tbi_Institucion</u>		
Descripción: Representa una institución en la que se despliega el alasPACS Viewer.		
Atributo	Tipo	Descripción
codigoInstitucion	integer	Identificador de la Institución.
nombreInstitucion	Varchar(50)	Nombre de la Institución.
direccionInstitucion	Varchar(100)	Dirección de la Institución
<u>Nombre: Ubicacion</u>		

Capítulo 3: Arquitectura y Diseño

Descripción: Representa un local de una institución en el que esta ubicado una PC en la cual se instala el alasPACS Viewer.		
Atributo	Tipo	Descripción
codigoUbicacion	integer	Identificador de la Ubicación.
nombreUbicacion	Varchar(20)	Nombre de la Ubicación.
codigoInstitucion	integer	Identificador de la Institución a la que pertenece.
<i>Nombre: tbl_PC</i>		
Descripción: Representa una PC en la que se va a instalar el alasPACS Viewer.		
Atributo	Tipo	Descripción
numeroSerie	integer	Número de Serie de la PC y es el identificador de la misma.
codigoUbicacion	integer	Identificador de la Ubicación en la cual esta localizada la PC.
numeroSerieMicro	integer	Numero de Serie del Microprocesador de la PC.
numeroSeriePlacaBase	integer	Numero de Serie del Placa Base de la PC.
<i>Nombre: tbl_SolicitudLicencia</i>		
Descripción: Representa una Solicitud de Licencia.		
Atributo	Tipo	Descripción
numeroSeriePC	integer	Identificador de la PC a la que corresponde la Solicitud de Licencia.
codigoSolicitudLicencia	integer	Identificador de la Solicitud de Licencia.
fecha	timestamp	Es la fecha en la que se creó la solicitud de licencia.
<i>Nombre: tbl_Licencia</i>		
Descripción: Representa una Licencia otorgada a una PC determinada.		
Atributo	Tipo	Descripción
codigoSolicitudLicencia	integer	Código de la Solicitud de Licencia a la que pertenece.
codigoActivacion	char	Código usado durante la activación del sistema alasPACS Viewer.
renovada	boolean	Indica si la Licencia es renovada.
fecha	timestamp	Es la fecha en la que se generó la licencia, si es una licencia renovada, entonces representa la fecha en la que se renovó la licencia.

3.5 Prototipo no Funcional.

El prototipo no funcional se utiliza para probar algunos aspectos del diseño del software y hacer las validaciones con el cliente sobre las posibles interfaces de usuarios.

Capítulo 3: Arquitectura y Diseño

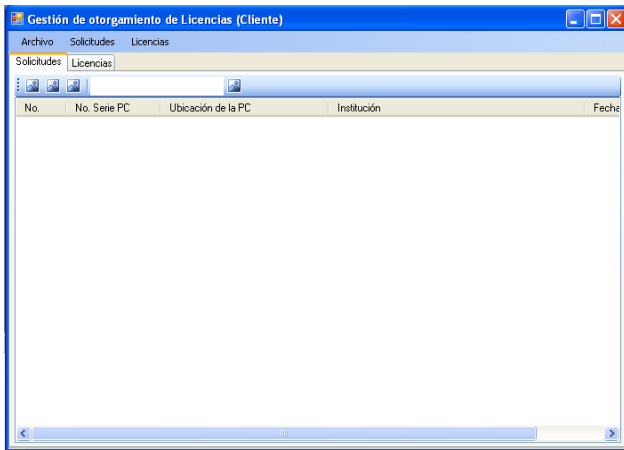


Fig.47 Pantalla de Gestión del otorgamiento de Licencias (Cliente).

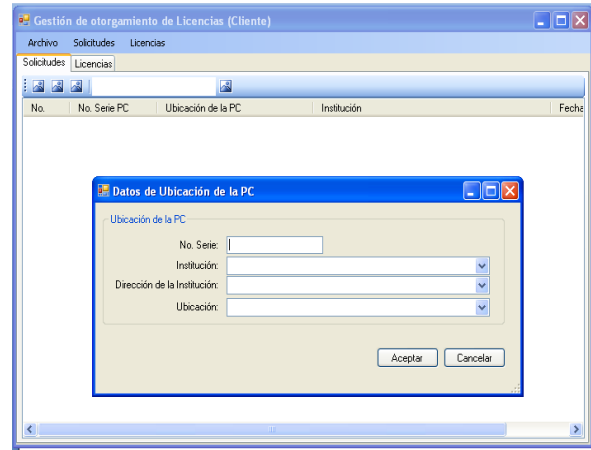


Fig.48 Pantalla para generar la nueva solicitud.

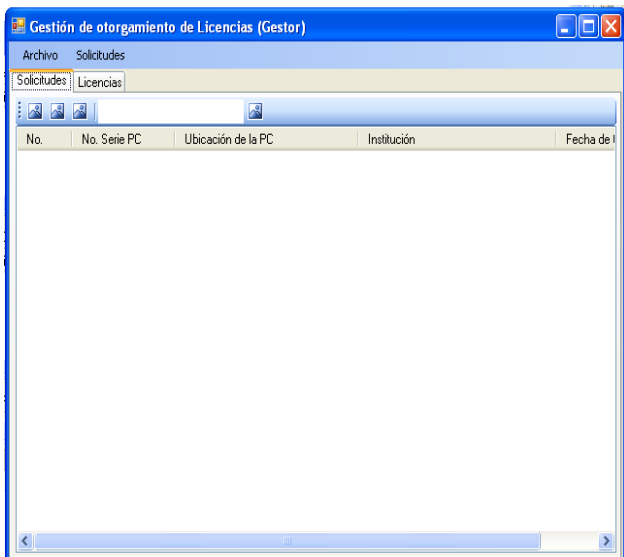


Fig.49 Pantalla de Gestión del otorgamiento de Licencias (Gestor).

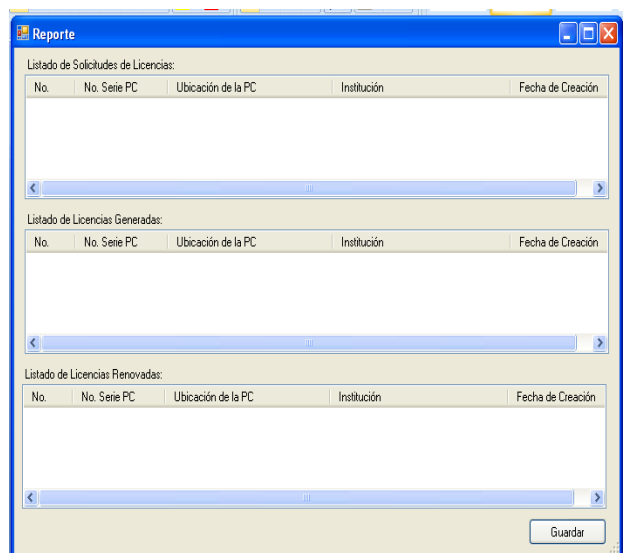


Fig.50 Pantalla de Generación de Reportes.

En este capítulo se describió la arquitectura del sistema, además se realizaron los diagramas de clases del análisis y colaboración del análisis, que facilitaron una primera aproximación al modelo del diseño. Pudiendo definir las clases más significativas del mismo, con sus atributos y métodos, permitiendo así un mejor entendimiento del diseño del sistema para su futura implementación.

CONCLUSIONES

Durante el desarrollo de la presente investigación, mediante el cumplimiento de las tareas y objetivos propuestos se llegó a las siguientes conclusiones:

- El sistema diseñado permite gestionar el otorgamiento de licencias para el sistema alas PACSViewer mejorando la forma en que se realiza actualmente.
- El sistema diseñado permite manejar de manera organizada las solicitudes de licencias separando el proceso de obtención de estas del proceso de generación de las licencias asociadas.
- Mediante el sistema diseñado el proceso de generación de licencias se realiza de manera centralizada lo que minimiza el peligro de que las herramientas usadas para este propósito sean robadas con las consiguientes pérdidas económicas que esto traería para el país.

RECOMENDACIONES

Con el objetivo de ofrecer una mejor protección del sistema alas PACSViewer durante su despliegue y en pos de lograr funcionalidades cada vez más completas de las herramientas usadas para el control de las licencias que se otorgan para el uso de este sistema, los autores hacen las siguientes recomendaciones:

- Implementar el sistema diseñado y usarlo en el otorgamiento de licencias para el sistema alas PACSViewer.
- Diseñar un sistema que haga uso de la infraestructura de redes existente para mejorar el proceso del otorgamiento de licencias para el sistema alas PACSViewer, utilizando llave pública para la protección y confiabilidad de la información que se maneja.

REFERENCIAS BIBLIOGRÁFICAS

1. Promoviendo un mundo digital seguro y legal. Información antipiratería. *Business Software Alliance*. [Online] 2000-2009. <http://w3.bsa.org/paraguay/antipiracy/Why-a-License-Matters.cfm>.
2. **Pedagógico de Ceibal.** Plan Ceibal. *Plan Ceibal*. [Online] http://www.ceibal.edu.uy/index.php?option=com_content&view=article&id=682:ique-es-la-licencia-de-software&catid=80:que-es-una-computadora&Itemid=224.
3. **Ltd, Aladdin Knowledge Systems** . Software Rights Management. Guia de proteccion y licencia del software v 2.10 [Online] ftp://ftp.aladdin.com/pub/hasp/srm/Documentation/HASP_SRM_Guide_ES.pdf.
4. idem al 3. [Online]
5. idem al 3. [Online]
6. idem al 3. [Online]
7. idem al 3. [Online]
8. idem al 3. [Online]
9. **Ltda, Softart.** NetSupport DNA Inventario. *Softart Ltda*. [Online] Diciembre 13, 2008. <http://www.ns-la.com/index.php/netsupport-dna-inventario>.
10. Guía del usuario de Microsoft Software Inventory Analyzer 4.0. *Guía del usuario de Microsoft Software Inventory Analyzer 4.0*. [Online] http://download.microsoft.com/download/6/c/9/6c9912cf-9bc9-43ed-b4ca-bf68a383b6d8/Guia_MSIA.pdf.
11. Sciensoft. *Products. Eleckey 2.0*. [Online] 1997- 2010. <http://www.sciensoft.com/products/eleckey/>.
12. idem al 11. [Online]
13. Free Download Manager. *Protector de Licencias(Licence Protector)2.5*. [Online] http://www.freedownloadmanager.org/es/downloads/Protector_de_Licencia_3512_p/.
14. idem al 13. [Online]
15. **Community, Interactive Programmers.** LWP.Comunidad de Programadores. [Online] 2000. <http://www.lawebdelprogramador.com/cursos/mostrar.php?id=44&texto=Java>.
16. **emagister.com.** wikilearning. *wikilearning*. [Online] 2007. http://www.wikilearning.com/apuntes/c++_net-c_net/17570-1.
17. **Semanat Aldana, Edmis Deivis and Verdecia Four, Leonor.** *Sistema de Video Vigilancia*. Ciudad de la Habana : UCI, 2009.
18. **GRUPO SOLUCIONES** GSinnova. *Rational Suite*. [Online] <http://www.rational.com.ar/herramientas/rationalsuite.html>.

Referencias Bibliográficas

19. **WIROOS, Hosting.** TARINGA! Inteligencia Colectiva. *TARINGA! Inteligencia Colectiva*. [Online] http://www.taringa.net/posts/downloads/3277137/Enterprise-Architect-7_5-%28Edicion-Corporativa%29.html.
20. **Ltd, Sparx Systems Pty.** SPARX Systems. *SPARX Systems*. [Online] 2000-2010. <http://www.sparxsystems.com.ar/>.
21. idem al 20. [Online]
22. **SlideShare Inc.** slideshare. *Visual Paradigm For Uml*. [Online] 2010 . <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
23. **grupotressinternacional.** Microsoft Visual Estudio 2008. [Online] febrero 2008. <http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.05.01.Articulo.Lo%20nuevo%20en%20Visual%20Studio%202008.pdf>.
24. idem al 18. [Online]
25. Ibercom. [Online] 1996-2009. https://www.ibercom.com/soporte/index.php?_m...a...996 .
26. PostGreSQL vs. MySQL. [Online] http://danielpecos.com/docs/mysql_postgres/x57.html.
27. idem al 29. [Online]
28. Astra Nuevas Tecnologías de la Información. [Online] <http://www.astra.es/noticias/postgresql-8-3-ya-disponible> .
29. Calisoft. [Online] http://calisoft.uci.cu/index.php?option=com_content&view=article&id=34&Itemid=27.

BIBLIOGRAFÍA

1. Astra Nuevas Tecnologías de la Información. [Online] <http://www.astra.es/noticias/postgresql-8-3-ya-disponible>.
2. Calisoft. [Online] http://calisoft.uci.cu/index.php?option=com_content&view=article&id=34&Itemid=27.
3. **Community, Interactive Programmers.** LWP.Comunidad de Programadores. [Online] 2000. <http://www.lawebdelprogramador.com/cursos/mostrars.php?id=44&texto=Java>.
4. **Emagister.com.** wikilearning. *wikilearning.* [Online] 2007. http://www.wikilearning.com/apuntes/c++_net-c_net/17570-1.
5. Free Download Manager. *Protector de Licencias(Licence Protector)2.5.* [Online] http://www.freedownloadmanager.org/es/downloads/Protector_de_Licencia_3512_p/.
6. GlobbTV Professional Television. *PC WORLD digital.* [Online] <http://www.idg.es/pcworld/Llaves-de-proteccion-Blue-Beige-T-Lock-Hardlock-/art33338.htm>.
7. GRUPO SOLUCIONES GSinnova. *Rational Suite.* [Online] <http://www.rational.com.ar/herramientas/rationalsuite.html>.
8. **Grupotressinternacional.** Microsoft Visual Estudio 2008. [Online] febrero 2008. <http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.05.01.Articulo.Lo%20nuevo%20en%20Visual%20Studio%202008.pdf>.
9. Guía del usuario de Microsoft Software Inventory Analyzer 4.0. *Guía del usuario de Microsoft Software Inventory Analyzer 4.0.* [Online] http://download.microsoft.com/download/6/c/9/6c9912cf-9bc9-43ed-b4ca-bf68a383b6d8/Guia_MSIA.pdf.
10. Ibercom. [Online] 1996-2009. <https://www.ibercom.com/soporte/index.php?m...a...996>.
11. **Ltda, Softart.** NetSupport DNA Inventario. *Softart Ltda.* [Online] Diciembre 13, 2008. <http://www.ns-la.com/index.php/netsupport-dna-inventario>.
12. **Ltd, Aladdin Knowledge Systems .** Software Rights Management. Guía de protección y licencia del software v 2.10. [Online] ftp://ftp.aladdin.com/pub/hasp/srm/Documentation/HASP_SRM_Guide_ES.pdf.
13. **Ltd, Sparx Systems Pty.** SPARX Systems. *SPARX Systems.* [Online] 2000-2010. <http://www.sparxsystems.com.ar/>.
14. Oracle. [Online] <http://www.uaem.mx/posgrado/mcruz/cursos/miic/oracle3.ppt>.
15. **Pedagógico de Ceibal .** Plan Ceibal. *Plan Ceibal.* [Online] http://www.ceibal.edu.uy/index.php?option=com_content&view=article&id=682:ique-es-la-licencia-de-software&catid=80:que-es-una-computadora&Itemid=224.
16. PostGreSQL vs. MySQL. [Online] http://danielpecos.com/docs/mysql_postgres/x57.html.

17. Promoviendo un mundo digital seguro y legal. Información antipiratería. *Business Software Alliance*. [Online] 2000-2009. <http://w3.bsa.org/paraguay/antipiracy/Why-a-License-Matters.cfm>.
18. Sciensoft. *Products. Eleckey 2.0*. [Online] 1997- 2010. <http://www.sciensoft.com/products/eleckey/>.
19. **Semanat Aldana, Edmis Deivis and Verdecia Four, Leonor**. *Sistema de Video Vigilancia*.
20. **SlideShare Inc.** slideshare. *Visual Paradigm For Uml*. [Online] 2010 . <http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml>.
21. Taiwan excellence 2009. *canal visual basic .net*. [Online] <http://www.canalvisualbasic.net/>.
22. **Trovamala, Cynthia; Bahena,Nancy and Romero,Alejandro**. Microsoft SQL Server. [Online] <http://www.uaem.mx/posgrado/mcruz/cursos/miic/sql5.pdf>.
23. **WIROOS, Hosting**. TARINGA! Inteligencia Colectiva. *TARINGA! Inteligencia Colectiva*. [Online] http://www.taringa.net/posts/downloads/3277137/Enterprise-Architect-7_5-%28Edicion-Corporativa%29.html.

ANEXOS

ANEXO I. Descripción del proceso Gestión del otorgamiento de licencias del software.

Nombre:	Gestión del otorgamiento de licencias del software.
Objetivos:	Otorgar las licencias correspondientes al sistema alas PACSViewer.
Evento(s) que lo generan:	No aplicable.
Precondiciones:	
Poscondiciones:	El sistema alas PACSViewer queda activado en las PCs a licenciar.
Reglas de Negocio:	
Responsables:	Especialista en Despliegue, Licenciador, Administrador de la Institución.
Clientes internos:	
Clientes externos:	
Entradas:	Fichero de solicitudes de licencias.
Salidas:	Paquete de licencias.
Actividades:	<ol style="list-style-type: none"> 1. Llegar a la institución hospitalaria. 2. Conectar dispositivo externo. 3. Instalar el sistema alas PACSViewer en la PC. 4. Obtener datos de la PC. 5. Retirar dispositivo externo. 6. Dirigirse al Grupo de Desarrollo. 7. Entregar dispositivo. 8. Exportar la información. 9. Importar fichero de solicitudes de licencias. 10. Generar licencia. 11. Exportar e importar paquete de licencias. 12. Importar paquete de licencias. 13. Entregar paquete de licencias. 14. Guardar licencias. 15. Dirigirse a las PC cliente. 16. Conectar dispositivo.

	<p>17. Activar sistema.</p> <p>18. Retirar dispositivo.</p> <p>19. Informar que se han quedado PC sin instalar.</p> <p>20. Solicitar licencia.</p>
--	--

Descripción del flujo básico

1. Llegar a la institución hospitalaria: El Especialista en Despliegue se dirige a la Institución Hospitalaria donde va a ser instalado el sistema alas PACSViewer.

Precedente: No aplicable

Responsable: Especialista en Despliegue.

Entradas: No aplicable.

Salidas: No aplicable.

2. Conectar dispositivo externo: El Especialista en Despliegue Introduce el dispositivo Externo en la PC para Instalar el sistema.

Precedente: 1

Responsable: Especialista en Despliegue.

Entradas: Dispositivo externo.

Salidas: No aplicable.

3. Instalar el sistema alasPACS Viewer en la PC: Se procede a Instalar el sistema alas PACSViewer en todas las PCs que le hayan sido otorgadas para su licenciamiento.

Precedente: 2

Responsable: Especialista en Despliegue.

Entradas: No aplicable.

Salidas: No aplicable.

4. Obtener datos de la PC: Se llenan los datos como son: nombre de la Institución a la que pertenece la PC, número de serie de la PC, y ubicación de la PC y obtiene los datos de hardware que son únicos para cada PC de forma automática.

Precedente: 3

Responsable: Especialista en Despliegue.

Entradas: No aplicable.

Salidas: No aplicable.

5. Retirar dispositivo externo: El Especialista en Despliegue procede a retirar el dispositivo externo una vez que se termina de recoger los datos asociados de la PC.

Precedente: 4

Responsable: Especialista en Despliegue.

Entradas: No aplicable.

Salidas: Dispositivo externo.

6. Dirigirse al Grupo de Desarrollo: Luego de haber terminado de instalar el sistema en todas las PCs que le fueron otorgadas se dirige al Grupo de Desarrollo localizado en la UCI.

Precedente: 5

Responsable: Especialista en Despliegue.

Entradas: No aplicable.

Salidas: No aplicable.

7. Entregar dispositivo: Se le entrega el dispositivo al Licenciador con la información obtenida.

Precedente: 6

Responsable: Especialista en Despliegue.

Entradas: No aplicable.

Salidas: No aplicable.

8. Exportar la información: El Licenciador se encarga de exportar la información contenida a un fichero de solicitudes de licencias.

Precedente: 7

Responsable: Licenciador.

Entradas: Dispositivo externo.

Salidas: Fichero de solicitudes de licencias.

9. Importar fichero de solicitudes de licencias: Procede a importar el fichero de solicitudes de licencias obtenido.

Precedente: 8

Responsable: Licenciador.

Entradas: Fichero de solicitudes de licencias.

Salidas: No aplicable.

10. Generar licencia: Se genera las licencias asociadas a cada PC, creándose así un paquete con todas las licencias incluidas.

Precedente: 9

Responsable: Licenciador.

Entradas: No aplicable.

Salidas: Paquete de licencias.

11. Exportar paquete de licencias: Se exporta el paquete con todas las licencias generadas y luego se importa dicho paquete.

Precedente: 10

Responsable: Licenciador.

Entradas: No aplicable.

Salidas: Paquete de licencias.

12. Importar paquete de licencias: Se exporta el paquete con todas las licencias generadas y luego se importa dicho paquete.

Precedente: 11

Responsable: Licenciador.

Entradas: Paquete de licencias.

Salidas: No aplicable.

13. Entregar paquete de licencias: El Licenciador luego de haber obtenido el paquete de licencias generadas se lo hace llegar al Administrador de la Institución donde fueron recogidas las solicitudes.

Precedente: 12

Responsable: Licenciador.

Entradas: Paquete de licencias.

Salidas: No aplicable.

14. Guardar licencias: El Administrador queda asignado como responsable de conservar estas licencias, las cuales pueden ser reutilizables en caso de existir algún problema que no esté asociado con los datos de hardware de la PC.

Precedente: 13

Responsable: Administrador de la Institución.

Entradas: Paquete de licencias.

Salidas: No aplicable.

15. Dirigirse a las PC cliente: El Administrador se dirige a las PCs donde fue instalado el sistema alas PACSViewer.

Precedente: 14

Responsable: Administrador de la Institución.

Entradas: No aplicable.

Salidas: No aplicable.

16. Conectar dispositivo: Se conecta el dispositivo en la PC a activar.

Precedente: 15

Responsable: Administrador de la Institución.

Entradas: Dispositivo externo.

Salidas: No aplicable.

17. Activar sistema: Se comprueba con los datos del hardware de dichas PCs que el paquete contenga una licencia apropiada para cada una de ellas permitiendo así utilizar el sistema alas PACSViewer.

Precedente: 16

Responsable: Administrador de la Institución.

Entradas: No aplicable.

Salidas: No aplicable.

18. Retirar dispositivo: Una vez que se termina de activar el sistema alas PACSViewer en la PC, retira el dispositivo.

Precedente: 17

Responsable: Administrador de la Institución.

Entradas: No aplicable.

Salidas: No aplicable.

19. Informar que se han quedado PC sin instalar: En caso de que paquete de licencia no contenga una licencia apropiada para la PC que se quiere activar, tendría el Administrador de la Institución que informar al Departamento de Software Médico Imagenológico que se han quedado PC sin instalar.

Precedente: 18

Responsable: Administrador de la Institución.

Entradas: No aplicable.

Salidas: No aplicable.

20. Solicitar licencia: Luego de haber informado que se han quedado PC sin instalar se procede a solicitar una licencia para dicha PC, iniciándose así el proceso de gestión del otorgamiento de licencias del software sobre la PC.

Precedente: 19

Responsable: Administrador de la Institución.

Entradas: No aplicable.

Salidas: No aplicable.

ANEXO II. Descripción ampliada de los Casos de Uso del Sistema.

II.1 Casos de Uso Expandidos del Cliente.

<u>CU # 2. Generar Solicitud.</u>		
Objetivo	Permite genera una solicitud de licencia a partir de los datos de configuración de hardware de una PC así como los datos de ubicación de la PC como son: nombre de la Institución a la que pertenece la PC, número de serie de la PC, y ubicación de la PC.	
Actores	Operador (Inicia).	
Resumen	El caso de uso se inicia cuando el Operador pulsa la opción “Obtener Datos de la PC”, el sistema recolecta los datos de hardware de la PC, el usuario entra los datos de ubicación de la PC y el Sistema crea una solicitud de licencia con todos estos datos y de esta forma termina el caso de uso.	
Complejidad	Media.	
Prioridad	Crítico.	
Referencias	RF 3.	
Precondiciones		
Poscondiciones	Se genera una solicitud de licencia.	
Flujo de eventos		
Flujo básico < Generar Solicitud >		
<ol style="list-style-type: none"> 1. El Operador pulsa la opción “Obtener Datos de la PC”. 2. El Sistema muestra una ventana para entrar los datos de ubicación de la PC. 3. El Operador entra los datos de ubicación de la PC y pulsa “Aceptar”. 4. El Sistema captura los datos de hardware de la PC. 5. El Sistema crea una solicitud de licencia a partir de los datos de Hardware obtenidos. 6. El Sistema visualiza la solicitud creada. 7. EL Sistema muestra un mensaje “Datos recolectados correctamente”. 		
Relaciones	CU Incluidos	Obtener Datos de la PC.
	CU Extendidos	No Aplicable.
Requisitos no funcionales	RNS 1. Cada licencia generada es específica para una única PC.	
Asuntos pendientes	No aplicable.	
<u>CU # 4. Exportar Fichero de Solicitudes de Licencias.</u>		
Objetivo	Permite exportar las solicitudes de licencias correspondientes a las PCs, cuyos datos fueron recolectados hacia un fichero de solicitud de licencia.	
Actores	Operador (Inicia).	
Resumen	El caso de uso se inicia cuando el Operador pulsa la opción “Exportar Fichero de	

	solicitudes de licencias”. El usuario selecciona la ubicación y el nombre del fichero a crear, el sistema crea un fichero con todas las solicitudes de licencias y de esta forma termina el caso de uso.	
Complejidad	Media.	
Prioridad	Crítico.	
Referencias	RF 5.	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo básico < Exportar Fichero de Solicitudes de Licencias >		
<ol style="list-style-type: none"> 1. El Operador pulsa la opción “Exportar Fichero de Solicitudes de Licencias”. 2. El Sistema muestra una ventana para que el usuario seleccione la ubicación y el nombre del fichero. 3. El Operador selecciona la ubicación, escribe un nombre para el fichero y pulsa la opción “Aceptar”. 4. El Sistema crea el fichero con todas las solicitudes de licencias. 5. El Sistema muestra un mensaje indicando que las solicitudes de licencias han sido correctamente exportadas. 		
Flujos alternos		
<i>2a. No existen solicitudes de licencias.</i>		
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no existen solicitudes de licencias para exportar hacia el fichero. 		
Relaciones	CU Incluidos	No Aplicable.
	CU Extendidos	No Aplicable.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes		No aplicable.
<u>CU # 5. Importar Paquete de Licencias.</u>		
Objetivo	Carga los datos de las licencias contenidas en un paquete de licencias.	
Actores	Operador (Inicia).	
Resumen	El caso de uso se inicia cuando el Operador pulsa la opción “Importar Paquete de Licencias”. El usuario selecciona un paquete de licencias y el sistema carga las licencias contenidas en este.	
Complejidad	Media.	
Prioridad	Crítico.	

Referencias	RF 6.	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo básico < Importar Paquete de Licencias >		
<ol style="list-style-type: none"> 1. El Operador pulsa la opción "Importar Paquete de Licencias". 2. El Sistema muestra una ventana para que el usuario seleccione el fichero. 3. El Operador selecciona el paquete de licencias y pulsa la opción "Aceptar". 4. El Sistema carga las licencias desde el fichero y los almacena en una representación interna. 5. El Sistema visualiza las licencias cargadas. 6. El Sistema muestra un mensaje indicando que las licencias han sido correctamente cargadas. 		
Flujos alternos		
<i>4a. El fichero no existe.</i>		
<ol style="list-style-type: none"> 7. El sistema muestra un mensaje indicando de que el fichero no existe. 		
<i>4b. No se puede abrir el fichero.</i>		
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando de que el fichero no se pudo abrir. 		
Relaciones	CU Incluidos	Visualizar Lista de Licencias.
	CU Extendidos	No aplicable.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes		No aplicable.
<u>CU # 7. Activar Licencia.</u>		
Objetivo	Busca dentro del paquete de licencias una adecuada para la PC y activa el sistema alas PACSViewer.	
Actores	Operador (Inicia).	
Resumen	El caso de uso se inicia cuando el Operador pulsa la opción "Activar Licencia". El sistema busca en el paquete una licencia adecuada y activa el sistema alas PACSViewer terminando así el caso de uso.	
Complejidad	Media.	
Prioridad	Crítico.	
Referencias	RF 9.	
Precondiciones	Tiene que haberse cargado el paquete de licencia.	
Poscondiciones		
Flujo de eventos		
Flujo básico < Activar Licencia>		

1. El **Operador** pulsa la opción “Activar Licencia”.
2. El Sistema busca dentro del paquete de licencias una que se corresponda con los datos de Hardware de la PC.
3. El Sistema activa el alas PACSViewer usando la licencia encontrada.
4. El Sistema muestra un mensaje indicando que las licencias han sido activas correctamente.

Flujos alternos

3a. *No se encontró ninguna licencia.*

1. El sistema muestra un mensaje indicando que no se encontró una licencia adecuada para la PC.

Relaciones	CU Incluidos	Comprobar Datos de Hardware.
	CU Extendidos	No Aplicable.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes		No aplicable.

CU # 8. Renovar Licencia.

Objetivo	Permite sustituir una licencia que haya dejado de ser válida por una nueva.
Actores	Operador (Inicia).
Resumen	El caso de uso se inicia cuando el Operador busca una licencia y pulsa la opción “Renovar Licencia”, se selecciona un paquete de licencias y desde este se carga la nueva licencia, terminando así el caso de uso.
Complejidad	Media.
Prioridad	Crítico.
Referencias	RF 10.
Precondiciones	
Poscondiciones	La licencia es renovada.

Flujo de eventos

Flujo básico < Renovar Licencia >

1. El **Operador** introduce el número de serie de la PC y pulsa la opción “Buscar Licencia.
2. El Sistema busca la licencia correspondiente y la muestra.
3. El **Operador** selecciona la licencia y pulsa la opción “Renovar licencia”
4. El Sistema muestra una ventana para seleccionar el paquete que contiene la nueva licencia.
5. El **Operador** selecciona el paquete de licencia y pulsa “Aceptar”.
6. El Sistema carga la nueva licencia y sustituye la seleccionada.

Flujos alternos

2a. *No se encontró ninguna licencia.*

1. Termina el caso de uso.		
Relaciones	CU Incluidos	Buscar Licencia.
	CU Extendidos	Importar Paquete de Licencias.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes		No aplicable.

II.2 Casos de Uso Expandidos del Gestor.

<u>CU # 11. Importar Fichero de Solicitud.</u>	
Objetivo	Permite cargar desde un fichero de solicitudes de licencias los datos de las PCs para las que se van a generar las licencias correspondientes.
Actores	Licenciador (Inicia).
Resumen	El caso de uso se inicia cuando el Licenciador pulsa la opción "Importar Fichero de Solicitud", terminando así el caso de uso.
Complejidad	Media.
Prioridad	Crítico.
Referencias	RF 13.
Precondiciones	No Aplicable.
Poscondiciones	El sistema guarda internamente los datos de las solicitudes de licencias.
Flujo de eventos	
Flujo básico < Importar Fichero de Solicitud >	
<ol style="list-style-type: none"> 1. El Licenciador pulsa la opción "Importar Fichero de Solicitud". 2. El Sistema muestra una ventana para que el usuario seleccione el fichero. 3. El Licenciador selecciona el fichero de solicitudes de licencias y pulsa la opción "Aceptar". 4. El Sistema carga las solicitudes de licencias desde el fichero y los almacena internamente. 5. El Sistema visualiza las solicitudes de licencias cargadas. 6. El Sistema muestra un mensaje indicando que las peticiones de licencias han sido correctamente cargadas. 	
Flujos alternos	
<i>4a. El fichero no existe.</i>	
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando de que el fichero no existe. 	
<i>4b. No se puede abrir el fichero.</i>	
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando de que el fichero no se pudo abrir. 	

Relaciones	CU Incluidos	Visualizar Solicitud.
	CU Extendidos	No Aplicable.
Requisitos funcionales no	No aplicable.	
Asuntos pendientes	No aplicable.	
<u>CU # 14. Generar Licencia.</u>		
Objetivo	Genera una licencia para cada una de las PCs cuyos datos se encuentran en el fichero de solicitudes de licencias.	
Actores	Licenciador (Inicia).	
Resumen	El caso de uso se inicia cuando el Licenciador pulsa la opción "Generar Licencia", el sistema genera una licencia para cada solicitud de licencia y las almacena en la base de datos, terminando así el caso de uso.	
Complejidad	Media.	
Prioridad	Crítico.	
Referencias	RF 16, RF 17.	
Precondiciones	Las solicitudes de licencias han sido cargadas.	
Poscondiciones	Las licencias son almacenadas en la base de datos.	
Flujo de eventos		
Flujo básico < Generar Licencia >		
<ol style="list-style-type: none"> 1. El Licenciador pulsa la opción "Generar Licencia". 2. El Sistema genera una licencia para cada solicitud de licencia. 3. El Sistema almacena las licencias generadas en la base de datos. 4. El Sistema muestra la lista con las licencias generadas. 		
Flujos alternos		
<i>2a. No hay solicitudes de licencias cargadas.</i>		
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no existen solicitudes de licencias cargadas. 		
<i>3a. La Licencia ya existe en la base de datos.</i>		
<ol style="list-style-type: none"> 1. El sistema renueva la licencia en la base de datos. 		
Relaciones	CU Incluidos	No Aplicable.
	CU Extendidos	No Aplicable.
Requisitos funcionales no	No aplicable.	
Asuntos pendientes	No aplicable.	
<u>CU # 16. Exportar Paquete de Licencias.</u>		

Objetivo	Crea un paquete de licencia que contiene las licencias generadas.	
Actores	Licenciador (Inicia).	
Resumen	El caso de uso se inicia cuando el Licenciador pulsa la opción “Exportar Licencia”, el sistema crea un paquete que contiene todas las licencias generadas, terminando así el caso de uso.	
Complejidad	Media.	
Prioridad	Crítico.	
Referencias	RF 19.	
Precondiciones	Las licencias han sido generadas.	
Poscondiciones	Se crea el paquete de licencias.	
Flujo de eventos		
Flujo básico < Exportar Paquete de Licencias >		
<ol style="list-style-type: none"> 1. El Licenciador pulsa la opción “Exportar Licencia”. 2. El Sistema muestra una ventana para seleccionar la ubicación y el nombre del paquete a exportar. 3. El Licenciador selecciona la ubicación y el nombre del paquete a exportar y pulsa “Aceptar”. 4. El Sistema crea el paquete de licencias. 5. El Sistema muestra un mensaje indicando que las licencias han sido exportadas correctamente. 		
Flujos alternos		
<i>1a. No hay licencias generadas.</i>		
<ol style="list-style-type: none"> 1. El sistema muestra un mensaje indicando que no hay licencias para exportar. 		
Relaciones	CU Incluidos	No Aplicable.
	CU Extendidos	No Aplicable.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes		No aplicable.
<u>CU # 17. Generar Reporte.</u>		
Objetivo	Muestra un informe que incluye información relacionada con las solicitudes de licencias, las licencias generadas y las licencias renovadas.	
Actores	Licenciador (Inicia).	
Resumen	El caso de uso se inicia cuando el Licenciador pulsa la opción “Generar Reporte”, el sistema muestra un informe que contiene las solicitudes de licencias, las licencias generadas y las licencias renovadas, terminando así el caso de uso.	
Complejidad	Media.	

Prioridad	Crítico.	
Referencias	RF 20.	
Precondiciones		
Poscondiciones		
Flujo de eventos		
Flujo básico < Generar Reporte >		
<ol style="list-style-type: none"> 1. El Licenciador pulsa la opción “Generar Reporte”. 2. El Sistema obtiene de la base de dato las solicitudes de licencias. 3. El Sistema obtiene de la base de dato las licencias generadas. 4. El Sistema obtiene de la base de dato las licencias renovadas. 5. El Sistema muestra un reporte que contiene la información de las solicitudes de licencias, las licencias generadas y las licencias renovadas. 		
Relaciones	CU Incluidos	No Aplicable.
	CU Extendidos	No Aplicable.
Requisitos funcionales	no	No aplicable.
Asuntos pendientes	No aplicable.	