

# Universidad de las Ciencias Informáticas

## Facultad 7



### **Título: Cassandra Clinic. Sistema de gestión de información para clínicas imagenológicas.**

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

**Autor(es):** Jublar Ramón García Ramos.  
Karel Eddy Tamayo Peña.

**Tutor(es):** Lic. Rolando Bonal Cáceres.

**Co-tutor:** Msc. Martha Ambruster Crespo.

**Asesor:** Lic. Héctor Raúl González Díez.

Ciudad de la Habana, Mayo 2007.

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Grupo de Procesamiento de Imágenes y Señales (GPI) de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de mayo del año 2007.

Jublar Ramón García Ramos

Karel Eddy Tamayo Peña

Lic. Rolando Bonal Cáceres

---

---

## DATOS DE CONTACTO

TUTOR: Lic. Rolando Bonal Cáceres ([bonal@uci.cu](mailto:bonal@uci.cu)).

Profesor graduado de Licenciatura en Física. Ha impartido la asignatura Física I y Física II. Posee categoría docente de Asistente. Es especialista de la Dirección de Software para la Salud. Actualmente se desempeña como Jefe del Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI).

CO-TUTOR: Msc. Martha Yurina Ambruster Crespo ([mambruster@uci.cu](mailto:mambruster@uci.cu))

Profesora graduada de Ingeniería Informática en el Instituto Superior Politécnico José Antonio Echeverría. Posee categoría docente de Instructor. Ha impartido las asignaturas de Programación 2, Programación 3 y Gráfico por Computadoras. Es profesora de la facultad 7 y en la actualidad pertenece al Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI), en el cual se desempeña como Jefe de Proyecto.

ASESOR: Lic. Héctor Raúl González Díez ([hglez@uci.cu](mailto:hglez@uci.cu))

Profesor graduado de Licenciatura en Física Nuclear. Ha impartido las asignaturas de Física I, Física II, Matemática 3 y Matemática 4. Es profesor de la facultad 7 y se desempeña actualmente como Jefe de Proyecto dentro del Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas (UCI).

## DEDICATORIA

### **De Jublar:**

A todos los que de una forma u otra han hecho posible mi formación como profesional, a mis profesores, compañeros y amigos. A mi familia y muy en especial a mi mamá y papá que confiaron siempre en mí, cuando hace 5 años me despidieron para hoy recibirme hecho ingeniero.

### **De Karel:**

A Mamaloli, con todo el amor, por lo mucho que le hubiese enorgullecido verme ingeniero. A lolín, que siempre ha sido y será, más que mi madre, mi luz y mi fuerza. A Keyla y papi, por el apoyo, la fe y la confianza. A toda mi familia.

## RESUMEN

En la actualidad en nuestro país existen numerosos centros especializados en el diagnóstico médico y la realización de exámenes basados en técnicas avanzadas de imagenología. En estos centros, se ha invertido gran cantidad de recursos y capital para proveerlos de tecnología de punta, que permita la realización de estos exámenes de forma eficiente y con una alta calidad.

**Cassandra Clinic** es una solución de software orientada a la gestión de la información generada en clínicas imagenológicas, desarrollada en Microsoft Visual Studio 2005, empleando la tecnología ASP.NET 2.0, configurable y adaptable; que tuvo su génesis en el proceso de informatización del Hospital Cardiológico Infantil Latinoamericano Dr. Gilberto Rodríguez Ochoa, de la ciudad de Caracas, República Bolivariana de Venezuela, a mediados de 2006.

**Cassandra Clinic** contempla además de la gestión de la **información imagenológica**, la gestión de la **información demográfica** de los pacientes, los datos socios económicos del núcleo familiar, la gestión de citas, la gestión de la ubicación durante el ingreso y el registro de egreso de pacientes, entre otras características. Está orientado a **roles**, estableciendo diferentes niveles de acceso a la información para asegurar su protección y adecuado tratamiento.

## PALABRAS CLAVES

Sistemas de Gestión de Información clínica, Imagenología, Cuba, Cassandra Clinic.

## TABLA DE CONTENIDO

DEDICATORIA .....	I
RESUMEN.....	II
INTRODUCCIÓN .....	1
<i>Objeto de estudio y campo de acción</i> .....	4
<i>Objetivo general</i> .....	5
<i>Idea a defender</i> .....	5
<i>Tareas</i> .....	5
<i>Métodos teóricos</i> .....	6
Analítico - sintético.....	6
Análisis histórico –lógico.....	6
Modelación.....	6
<i>Métodos Empíricos</i> .....	6
Observación.....	6
Entrevista .....	6
<i>Estructuración del contenido</i> .....	7
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	8
1.1 <i>¿Qué es DICOM?</i> .....	8
1.2 <i>¿Qué es HL7?</i> .....	9
1.3 <i>Sistemas para la Transmisión, Almacenamiento y Visualización de Imágenes Médicas (PACS)</i> .....	9
1.3.1 Componentes de PACS.....	10
1.3.2 Conectividad con otros sistemas de información intra - hospitalaria. ....	11
1.4 <i>Estado del arte de los sistemas de gestión de información clínica</i> .....	12
1.4.1 En el mundo.....	12
1.4.2 En Cuba.....	13
1.4.3 En la Universidad de las Ciencias Informáticas .....	13
1.5 <i>Tendencias y tecnologías actuales</i> .....	14
1.5.1 Aplicaciones Web .....	15
1.5.2 Lenguajes de Programación Web .....	16
1.5.3 Servidores Web .....	19
1.5.4 AJAX .....	19

1.5.5 Sistemas Gestores de Bases de Datos (SGBD).....	19
1.5.6 Fundamentación de la metodología a utilizar.....	24
1.5.7 Paquetes de software y herramientas usados para la solución del problema propuesto.....	25
1.5.8 Tecnologías auxiliares.....	26
1.5.9 Herramientas auxiliares.....	26
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	28
2.1 Problema científico y situación problemática.....	28
2.2 Objeto de automatización.....	30
2.3 Propuesta de sistema.....	31
2.4 Modelo de negocio.....	33
2.5 Diagrama de casos de uso del negocio.....	35
2.6 Diagramas de actividades de los Casos de Uso del negocio.....	37
2.7 Diagrama de Clases del Modelo de Objetos.....	44
2.8 Especificación de los requisitos de software.....	44
2.9 Requerimientos Funcionales.....	45
2.10 Requerimientos no funcionales del sistema.....	46
2.11 Definición de los casos de uso.....	48
2.12 Diagrama de casos de uso del sistema.....	54
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.....	62
Análisis.....	62
Diseño.....	62
3.1 Descripción de las clases.....	62
3.2 Diseño de la BD.....	63
3.3 Descripción de las tablas.....	63
CAPÍTULO 4: IMPLEMENTACIÓN.....	64
4.1 Diagrama de despliegue.....	64
4.2 Diagrama de componentes.....	65
4.3 Estudio de compatibilidad con la plataforma libre.....	66
CONCLUSIONES.....	68
RECOMENDACIONES.....	69
REFERENCIA BIBLIOGRÁFICA.....	70

BIBLIOGRAFÍA.....	71
ANEXOS .....	73
<i>Anexo 1 Casos de uso expandidos.....</i>	73
<i>Anexo 2 Diagramas de clases del análisis.....</i>	83
3.1.1 Diagrama de clases – Buscar pacientes .....	83
3.1.2 Diagrama de clases – Admitir paciente .....	84
3.1.3 Diagrama de clases – Ingresar paciente .....	85
3.1.4 Diagrama de clases – Egresar paciente.....	86
3.1.5 Diagrama de clases – Reservar cita con Especialista.....	87
3.1.6 Diagrama de clases – Reservar cita para Equipo Médico.....	88
3.1.7 Diagrama de clases – Registrar Orden de Estudio Imagenológico.....	89
3.1.8 Diagrama de clases – Mostrar Imágenes Médicas .....	90
3.1.9 Diagrama de clases – Mostrar Reportes Médicos.....	90
3.1.10 Diagrama de clases – Asignar Cama a Paciente .....	91
<i>Anexo 3 Diagramas de clases del diseño .....</i>	91
3.2.1 Diagrama de clases del diseño – Buscar pacientes.....	92
3.2.2 Diagrama de clases del diseño - Admitir paciente .....	93
3.2.3 Diagrama de clases del diseño – Ingresar paciente.....	94
3.2.4 Diagrama de clases del diseño – Egresar paciente .....	95
3.2.5 Diagrama de clases del diseño – Reservar cita con Especialista .....	96
3.2.6 Diagrama de clases del diseño – Reservar cita para Equipo Médico .....	97
3.2.7 Diagrama de clases del diseño – Registrar Orden de Estudio Imagenológico .....	98
3.2.8 Diagrama de clases del diseño – Mostrar Imágenes Médicas .....	99
3.2.9 Diagrama de clases del diseño – Mostrar Reportes Médicos .....	100
3.2.10 Diagrama de clases del diseño – Asignar Cama a Paciente.....	101

<i>Anexo 4 Diagramas de secuencia</i> .....	102
3.3.1 Diagrama de secuencia – Buscar pacientes .....	102
3.3.2 Diagrama de secuencia - Admitir paciente .....	103
3.3.3 Diagrama de secuencia – Ingresar paciente .....	104
3.3.4 Diagrama de secuencia – Egresar paciente.....	105
3.3.5 Diagrama de secuencia – Reservar cita con Especialista.....	106
3.3.6 Diagrama de secuencia – Reservar cita para Equipo Médico.....	107
3.3.7 Diagrama de secuencia – Registrar Orden de Estudio Imagenológico.....	108
3.3.8 Diagrama de secuencia – Mostrar Imágenes Médicas .....	109
3.3.9 Diagrama de secuencia – Mostrar Reportes Médicos.....	110
3.3.10 Diagrama de secuencia – Asignar Cama a Paciente .....	111
<i>Anexo 5 Descripción de las clases</i> .....	112
3.4.1 Clases entidades .....	112
3.4.2 Clases controladoras .....	132
<i>Anexo 6 Diagramas Entidad – Relación</i> .....	140
3.5.1 Diagrama Entidad Relación - Paquete Gestión de Citas .....	140
3.5.2 Diagrama Entidad Relación - Paquete Gestión de HC Imagenológica .....	141
3.5.3 Diagrama Entidad Relación - Paquete Gestión de Ubicación.....	142
<i>Anexo 7 Descripción de las tablas de la Base de Datos</i> .....	142
3.6 Descripción de las tablas.....	142
GLOSARIO .....	151

## INTRODUCCIÓN

El surgimiento de la informática, las computadoras personales, y las redes de comunicaciones durante la segunda mitad del siglo XX constituyó, sin duda, un fenómeno importante que ha dotado a la humanidad de medios potentísimos para la expansión del conocimiento, la cultura y la sabiduría.

A principio de la década de los setenta surgió el término radiología digital. Este concepto evolucionó la tecnología de los equipos de adquisición de imágenes médicas. En los ochenta surgieron modalidades de gran impacto, como la Resonancia Magnética de Imágenes, que se convertiría rápidamente en una de las más fiables y flexibles. Con el avance y la amplia aceptación de estas y otras técnicas se fueron desarrollando equipos de alta tecnología capaces de comunicarse entre sí, acoplarse y posibilitar la automatización de tareas que antaño se desarrollaban manualmente o con técnicas muy incipientes que introducían un gran número de errores e inconvenientes.

En nuestro país hoy en día existen centros hospitalarios que poseen entre su equipamiento médico tecnología de punta para la realización de exámenes médicos mediante el uso de la Imagenología, dígase equipos de Resonancia Magnética, Tomógrafos, entre otros; sin embargo, no existe la informatización necesaria para gestionar el gran número de información que se genera en este proceso y su concatenación con la información clínica del paciente. Lo que comúnmente se observa es la separación entre los dos tipos de información: por una parte, la información imagenológica, relativa a estudios, exámenes y resultados de estos y, por otra, la información demográfica y/o socio – económica, con un nivel muy bajo de integración y relación.

En muchos de estos centros se gestiona la información generada, manualmente, presuponiendo esto una gran ineficiencia y aumentando la posibilidad de extravíos de información valiosa y sensible de los pacientes atendidos. Existen grandes archivos físicos en los que se guardan historias clínicas y se anexan a éstas los resultados de los estudios de los pacientes, los resúmenes de evolución redactados por los especialistas y otros resultados. Esta historia clínica no es única, y no pocas veces se encuentra deteriorada y no disponible en el momento preciso para el personal calificado que demanda dicha información para emitir un diagnóstico correcto, confiable y basado en un verdadero historial clínico de un paciente.

El reto y la expectativa del gobierno cubano es la informatización total de su Sistema de Salud Pública y para lograr este objetivo ha invertido recursos y capital para la adquisición de equipamiento, tecnología y ha capacitado gran cantidad de personal en materia de informática. Como medida estratégica se creó la Universidad de las Ciencias Informáticas (UCI) en mayo de 2002, para potenciar el estudio de esta ciencia y convertirla en una de las principales ramas de la economía del país. Esta universidad debe integrarse al trabajo de las demás instituciones de la esfera, que se han creado en nuestro país, y convertirse en el centro del polo científico de la informática, de donde surjan los principales sistemas de software para la informatización de la sociedad cubana en general.

Otra de las medidas tomadas es la impartición de clases de computación desde edades tempranas a los niños, la potenciación de los Joven Clubs de Computación y Electrónica de todo el país, lo que ha asegurado la instrucción de la población y facilitará su interacción con la tecnología permitiendo la globalización del conocimiento de las ciencias informáticas.

Por las razones expuestas se hace pertinente y necesaria la creación de un sistema de software, que cumpla con los requerimientos que en materia de gestión de información imagenológica se demandan por las instituciones de salud de nuestro país y otros países, donde Cuba tiene presencia mediante sus médicos y otro personal de la rama de la salud, para lograr una gestión eficiente de la información imagenológica acorde con el nivel que ostenta nuestro país en esta rama. Para lograr esto, se hace necesario a su vez, cumplir con ciertos estándares presentes en el mundo y que son implementados por los productos líderes del mercado. Existen dos grandes estándares definidos que cualquier producto que quiera ser exitoso debe implementar, el estándar **DICOM** y el estándar **HL7**.

Tanto **HL7** como **DICOM** son regulados por organismos creados en el mundo para tal fin y estos aseguran la actualización y futuras versiones de las especificaciones de estos estándares, así como su adecuación para asegurar la compatibilidad con los nacientes sistemas de redes, protocolos, formatos de codificación de información y otras tecnologías que por su relevancia lo ameriten.

**DICOM** es un estándar para el almacenamiento, la transmisión y la comunicación de imágenes médicas y su organismo regulador es **NEMA** (National Electrical Manufacturers Association), quien además tiene los derechos reservados sobre este estándar, aunque el surgimiento del mismo fue un esfuerzo conjunto entre NEMA y ACR (American College of Radiology), cuando conformaron en 1983 el comité **DICOM Standards Committee**, que en 1985 lanza la primera versión.

**ACR** se funda en 1923, es una organización médica sin fines de lucro compuesta por radiólogos especialistas en diagnósticos, oncólogos especializados en radiación, radiólogos cirujanos, doctores en medicina nuclear y biofísicos. Tiene sedes en Reston, Virginia y oficinas en Filadelfia, Pensilvania y Washington D.C en Estados Unidos. Este colegio tiene una publicación oficial llamada: “*The Journal of the American College of Radiology (JACR)*”.

**NEMA**, por otra parte, se fundó el 1ro de septiembre de 1926 con la fusión de la Asociación de Fabricantes de Suministros Eléctricos y el Club de Energía Eléctrica, dos empresas americanas de aquel entonces. Tiene su sede en Rosslyn, Virginia, Estados Unidos; dicta numerosos estándares comunes usados en productos eléctricos y cuenta con aproximadamente 400 miembros.

El estándar **Health Level Seven (HL7)** es una especificación para un estándar de intercambio de datos electrónicos en el área de la salud, con especial énfasis en las comunicaciones intra – hospitalarias en el área de la información clínica y administrativa. Es el resultado del trabajo de un Comité de proveedores de usuarios, vendedores y consultores de sistemas de aplicación al área de salud, y la empresa HL7 Inc., es quien tiene los derechos de propiedad intelectual sobre este estándar. Esta empresa, por su parte, es una organización completamente voluntaria y sin fines de lucro, implicada en el desarrollo de estándares para los servicios de salud. Fue fundada en 1987, para producir un estándar para los sistemas de información hospitalaria y acreditada, en 1997, por el Instituto Nacional Americano de Normalización (ANSI, por sus siglas en inglés). Tiene su sede en Ann Arbor, en el estado de Michigan, Estados Unidos.

En nuestro país desde hace varios años se ha comenzado a desarrollar sistemas de gestión de información en el área de la salud, y algunos de los productos se encuentran hoy en día en funcionamiento en las instituciones nacionales e incluso en algunos centros en países extranjeros.

Como exponentes de lo anterior podemos citar GalenLab, sistema de gestión de medios de diagnóstico, elaborado por la empresa de software Softel, ampliamente desplegado en nuestro país y en la República Bolivariana de Venezuela, en los Centros de Diagnóstico Integral de la misión Barrio Adentro; Imagis®, sistema de visualización, diagnóstico y transmisión de imágenes médicas, implementado por el centro de Biofísica Médica de Santiago de Cuba, sistema que se encuentra instalado en 21 instituciones nacionales, así como en La República Dominicana y Ecuador; Alianza Vital, sistema de información para el control y el análisis de los pacientes atendidos en los servicios de cirugía pediátrica; SIH, sistema de información hospitalaria, también de la empresa Softel y que aún hoy sigue desarrollándose; Cassandra PACS,

sistema para el almacenamiento, transmisión y visualización de imágenes médicas desarrollado por estudiantes y profesores del Grupo de Procesamiento de Imágenes (GPI) de la Universidad de las Ciencias Informáticas. Este sistema se encuentra instalado en el Hospital Hermanos Ameijeiras de la capital cubana, y en Centros Médicos de Alta Tecnología en la República Bolivariana de Venezuela.

En el mundo el tema imagenológico está bien representado por compañías como SIEMENS, Phillips, entre otras consagradas y bien posicionadas en el mercado de equipos médicos y en los sistemas PACS y en algunos sistemas de gestión de información clínica; que invierten tiempo, dinero y recursos en lograr que sus productos cumplan con los estándares requeridos y con las normas de calidad; con un mercado principalmente expandido en Europa y América del Norte, aunque con presencia igualmente en Latinoamérica.

El desarrollo de un sistema de gestión de información para clínicas esencialmente imagenológicas, como la propuesta de este trabajo, puede aportar resultados como la correcta manipulación de la información, personal, demográfica y socioeconómica de los pacientes atendidos por estos centros, podría significar una mejora notable en el proceso de gestión de citas, organizando el flujo de pacientes y liberando al personal especializado de los trabajos de gestión de calendario, que hoy en día se realizan de forma manual. Permitiría la visualización de reportes médicos por parte de los especialistas, y de información estadística por parte de directivos, lo que aumentaría en alto grado el nivel de la eficiencia y la calidad total de los servicios de salud brindados. Permitiría dar un paso más, y muy importante, en la consecución de la Historia Clínica única y digital del paciente, que se ha trazado nuestro país en el plan de informatización del sistema de salud cubano. Su integración con los sistemas PACS presentes en las instituciones hospitalarias potenciaría en gran medida el uso de la tecnología existente en las mismas, además de brindar otras ventajas propias de la informatización, como por ejemplo un servicio de intranet hospitalaria, publicación de noticias, anuncios, repositorios de información, materiales de multimedia, etc.

### **Objeto de estudio y campo de acción**

El objeto de estudio del presente trabajo de diploma es la gestión de la información clínica, y el campo de acción se define como la gestión de información en clínicas imagenológicas.

## **Objetivo general**

Como objetivo general, los autores proponen, desarrollar un sistema que permita llevar a cabo de forma eficiente la gestión de la información generada en las clínicas imagenológicas y vincularla con la información del paciente.

## **Idea a defender**

El desarrollo de un sistema de gestión de la información en las clínicas imagenológicas para lograr la eficiente explotación de las tecnologías con las que cuentan las mismas y aumentar la eficiencia de nuestro Sistema de Salud Pública.

## **Tareas**

Para el cumplimiento del objetivo propuesto se han trazado las siguientes tareas:

- Estudiar los estándares DICOM 3.0 y HL7 2.x, para lograr un entendimiento claro de los estándares que rigen el desarrollo de sistemas en el campo de la salud.
- Investigar la compatibilidad de la tecnología existente, que permita hacer un despliegue del sistema en la plataforma libre.
- Investigar acerca de las formas de optimización de las aplicaciones y servicios de gestión de imágenes médicas.
- Desarrollar un sistema compatible con los estándares internacionales de salud.
- Desarrollar un sistema que garantice una eficiente gestión de la información generada en las clínicas imagenológicas.
- Diseñar una Base de Datos que soporte las funcionalidades necesarias para la correcta gestión de la información imagenológica.

Durante el desarrollo de la presente investigación se han utilizado los siguientes métodos científicos:

## **Métodos teóricos**

### **Analítico - sintético**

Este método ha sido útil en el análisis y comprensión de la teoría y documentación relacionada con el tema de investigación, permitiendo así extraer los elementos más relacionados e importantes con el objeto de estudio.

### **Análisis histórico –lógico**

La aplicación de método del análisis histórico – lógico, ha ayudado a los autores de esta investigación a entender el surgimiento y la evolución del tema tratado, así como otras temáticas estrechamente relacionadas a lo largo de la historia de la Informática.

### **Modelación**

Mediante este método se ha podido modelar el resultado, en menor escala, de lo que en una etapa avanzada de desarrollo, debe ser la presente investigación, los principales elementos que lo componen y su funcionamiento. Debido a que se ha escogido el Proceso Unificado de Rational como metodología de software, los autores se han auxiliado del lenguaje unificado de modelado (UML, por sus siglas en inglés), para generar los diferentes artefactos de software y modelos que han ayudado a representar el sistema, de desde distintas vistas, útiles a los miembros del equipo de desarrollo del mismo.

## **Métodos Empíricos**

### **Observación.**

El método de la observación ha provisto a los autores de una vía mas para la verificación del cumplimiento de las características que debe poseer el sistema y garantizar el cumplimiento de los objetivos trazados.

### **Entrevista**

Se han entrevistado y encuestado a personas que pueden aportar elementos significativos a la presente investigación sobre su resultado final y su uso.

## Estructuración del contenido

En el capítulo 1 se abordará en detalle todo lo relacionado con la fundamentación teórica que sustenta la presente investigación, se hará un estudio del estado del arte del tema y se expondrán las principales tendencias, técnicas y tecnologías usadas para la implementación de la solución de software propuesta por el presente trabajo.

En el Capítulo 2 se explicarán las características del sistema que se propone y se expondrán en total detalle, entre otros, el objeto de estudio, el problema científico, la situación problémica, el objeto de automatización y la información que se maneja.

El Capítulo 3 abordará el tema del análisis y diseño de software usando la metodología RUP se mostrarán y describirán los diagramas de clases del análisis y del diseño, así como los diagramas Entidad – Relación de la base de datos usada por el sistema.

En el Capítulo 4 se tratará todo lo relacionado con el flujo de trabajo de implementación y se expondrán los diagramas de implementación y despliegue, artefactos generados durante la puesta a punto del sistema.

En los Anexos disponibles en el presente material se encuentran los modelos auxiliares, tablas descriptivas de elementos del sistema, descripciones textuales de casos de uso, entre otros artefactos generados durante los flujos de trabajo de la metodología empleada.

El Glosario de términos que se ha incorporado al final del documento, define los términos de difícil comprensión, los cuales hemos resaltado en **negritas** para su mejor identificación.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se hace una breve explicación de los estándares DICOM y HL7. Se expone la estructura y componentes de un Sistema de Almacenamiento, Transmisión y Visualización de Imágenes Médicas (**PACS**) para ubicar el contexto en el que puede enmarcarse el sistema propuesto por la presente investigación. Luego se hace un análisis del estado del arte en nuestra universidad, a nivel nacional e internacional de los sistemas informáticos vinculados a la salud y enmarcados en la temática de la gestión de información clínica. Se presentan las principales tendencias en el mundo en cuanto al uso de tecnologías para dar solución a problemas similares al nuestro. Se exponen las principales características de las tecnologías usadas en la implementación del sistema propuesto. Se analiza la metodología de software empleada para la modelación del sistema, así como los paquetes de software empleados en la implementación de la solución.

### 1.1 ¿Qué es DICOM?

La versión 3 del estándar **DICOM** (Digital Imaging and Communications in Medicine) vio la luz en el año 1993, a partir de un rediseño completo de la publicación normalizada No 33-1988 de ACR-NEMA, y pertenece al campo de la informática médica, por lo que, en principio esta norma se solapa con otras de este campo. El estándar DICOM v.3.0 (DICOM en adelante) permite la transmisión, tratamiento e impresión de archivos DICOM, que son imágenes biomédicas que portan además informes del estudio, o estudios, realizados a un paciente.

Mediante el uso de DICOM se facilita la comunicación de imágenes en medicina, mejorando el manejo de información médica entre hospitales y centros de investigación y posibilitando el intercambio de información clínica regido por normas que permiten interconectar sistemas informáticos de diferentes fabricantes y hace posible la comunicación entre ellos. No es extraño que en un centro hospitalario exista una diversidad notable en las marcas y modelos de los equipos médicos, debido a la alta especialización y la gran variedad de productores de estos equipos en el mercado mundial; DICOM entra entonces a regir las normas y protocolos que deben cumplir estos dispositivos y hace posible que los archivos médicos puedan viajar de forma segura entre hospitales, centros de investigación y departamentos; luego esa

información puede ser vista remotamente para que los médicos puedan diagnosticar desde cualquier punto de la geografía y buscar segundas opiniones de otros expertos de una forma rápida y sencilla.

Las imágenes médicas son muy importantes para los diagnósticos de pacientes, tratamientos terapéuticos y evaluación de resultados. Gracias a las nuevas técnicas de imágenes digitales, tales como la tomografía computarizada, la angiografía digital por sustracción, la resonancia magnética, entre otras se ha visto reducida las dosis de radiación a los pacientes y los cortes anatómicos y se ha potenciado la investigación y tratamiento de dolencias y patologías que antes no era posible diagnosticar.

## **1.2 ¿Qué es HL7?**

El estándar **Health Level Seven (HL7)** es una especificación para un estándar de intercambio de datos electrónicos en el área de la salud, con especial énfasis en las comunicaciones intra – hospitalarias<sup>1</sup> en el área de la información clínica y administrativa. Es el resultado del trabajo de un comité de proveedores, de usuarios, vendedores y consultores de sistemas de aplicación en esta área.

En la actualidad los hospitales del primer mundo poseen sistemas independientes que se ocupan de la hospitalización de pacientes (**HIS**), de información de laboratorio (**LIS**), de informes de radiología (**RIS**), entre otros. La mayoría de las veces estos sistemas no son desarrollados por el mismo grupo de trabajo; de esta forma cada aplicación almacena la información con un formato diferente. La necesidad de la comunicación y de compartir los datos entre los sistemas hospitalarios es cada vez mayor, HL7 entra en juego, rigiendo y estipulando las normas necesarias para la comunicación entre estos sistemas de salud de naturaleza heterogénea. Este estándar permite con el uso de encabezados bien definidos la transmisión de la información con un formato específico y entendible a estos sistemas por igual.

## **1.3 Sistemas para la Transmisión, Almacenamiento y Visualización de Imágenes Médicas (PACS)**

A principio de la década de los setenta surgió el término radiología digital. Este concepto evolucionó la tecnología de los equipos de adquisición de imágenes médicas. Modalidades como la Tomografía Computarizada (CT) comenzaron a desarrollarse con muy buenos resultados. En los ochenta surgieron otras de gran impacto como la Resonancia Magnética de Imágenes (MRI). Por el carácter no invasivo de las radiaciones que emite y las ventajas que desde el punto de vista clínico ofrece, se convirtió rápidamente en una de las más fiables y flexibles.

A partir de estas y otras tecnologías como el ultrasonido (US), la medicina nuclear (NM), las redes de comunicación, las computadoras personales y los medios de almacenamiento, se hizo factible la transmisión de imágenes a distancia, el archivo de estudios imagenológicos en formato electrónico, las técnicas y algoritmos para el procesamiento de imágenes digitales y comienza la carrera de los Sistemas de Transmisión y Almacenamiento de Imágenes (PACS). Estos sistemas están formados por los equipos o modalidades de adquisición, la infraestructura de comunicaciones, servidores y las estaciones de visualización y procesamiento de la información. Todos esos elementos se integran de forma coherente a través de aplicaciones de software que son la interfaz entre los usuarios y la infraestructura tecnológica.

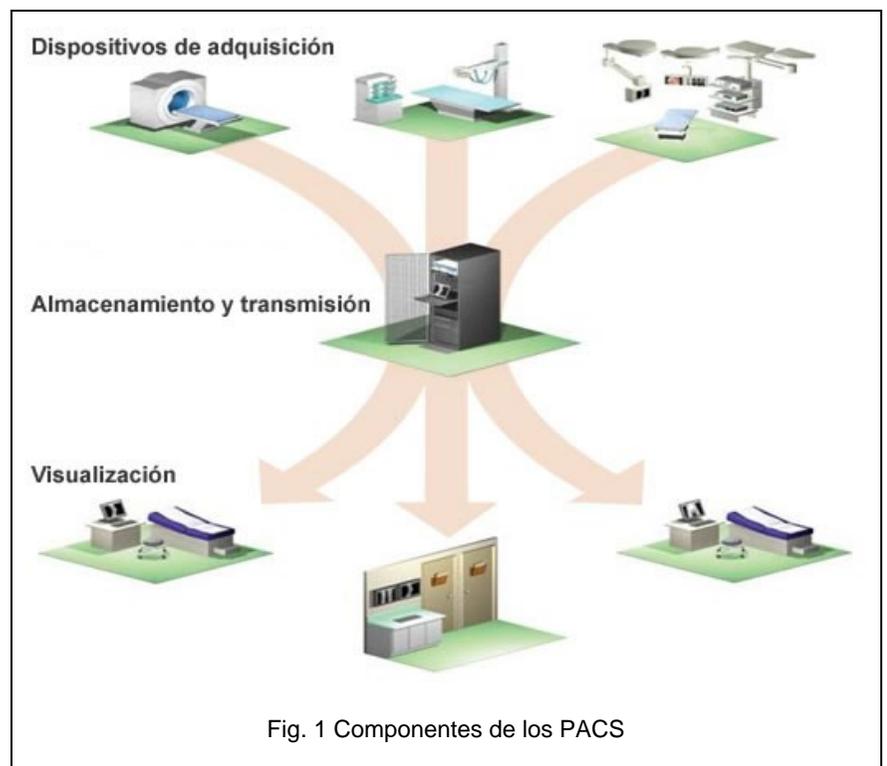
### 1.3.1 Componentes de PACS

Los PACS están formados por componentes individuales los cuales actúan como pequeños bloques que, integrados, constituyen la solución completa, como se muestra en la Figura 1. Entre los componentes fundamentales se encuentran:

- **Dispositivos de adquisición**

Son los dispositivos o equipos médicos encargados de generar las imágenes conforme al estándar DICOM 3.0. Entre estos equipos los más conocidos son los Tomógrafos, Angiógrafos, Ultrasonidos Digitales, entre otros. Los dispositivos de

adquisición tienen incorporadas determinadas funcionalidades que les permiten comunicarse con servidores de imágenes y de listas de trabajo para asegurar el flujo de información hacia los demás



componentes del PACS. Estas operaciones son implementadas siguiendo las especificaciones de los estándares DICOM y HL7.

- **Software Servidor**

El servidor central controla el flujo de información entre los componentes del PACS y otros sistemas integrados de salud. Incluye mecanismos de seguridad, automatización de tareas, administración de sistemas, creación de copias de respaldo de información, así como medidas de protección de la misma frente a crisis o situaciones excepcionales: compresión de la información ante carencia de espacio, copias hacia repositorios habilitados, entre otras.

- **Estaciones de visualización**

Permite la visualización de imágenes médicas generadas en los dispositivos de adquisición. Usualmente están provistas de paquetes de software que incorporan mecanismos de manipulación de imágenes, herramientas de medición, filtros, paletas etiquetadas de colores, y algunos más avanzados poseen funcionalidades como la reconstrucción tridimensional de estructuras corporales, entre otras características.

Por lo general son estaciones de trabajo, computadoras personales, con características especiales: prestaciones optimizadas para el alto procesamiento de video, periféricos de visualización de alta resolución, como monitores de radiología, que aseguran a los especialistas una visualización de alto detalle para la correcta emisión del diagnóstico imagenológico, entre otros dispositivos especializados.

### **1.3.2 Conectividad con otros sistemas de información intra - hospitalaria.**

Los PACS deben tener como una de sus características la comunicación con sistemas existentes en el hospital. Esta comunicación debe ser basada en el estándar HL7. Entre sus principales funcionalidades está la transmisión de datos, estos pueden ser datos demográficos de los pacientes, datos de su historia clínica y datos de estudios médicos realizados y sus resultados. El objetivo principal de esta comunicación es eliminar o reducir al máximo la necesidad de registro manual y en archivos físicos de los resultados de exámenes u órdenes de estudio, mejorando notablemente la gestión de la información clínica.

## **1.4 Estado del arte de los sistemas de gestión de información clínica.**

### **1.4.1 En el mundo**

En Estados Unidos existen compañías consagradas en este campo, como Siemens, reconocidas por sus innovadores productos, servicios y soluciones, que abarcan desde equipamiento de diagnóstico y tratamiento, hasta soluciones en tecnología informática que optimizan el flujo de trabajo e incrementan la eficiencia en hospitales, clínicas y consultorios. Siemens tiene su principal mercado en el continente europeo y en América del Norte, aunque también posee una poderosa sucursal en Chile. Esta división latinoamericana, Siemens Chile, ofrece una variedad de soluciones de gestión de datos para clientes y es actualmente el mayor Proveedor mundial de Servicios de Acceso en el sector. Es además productora de dispositivos de adquisición y equipos médicos de alta calidad.

DPI, empresa líder en Latinoamérica, ha desarrollado un software llamado DPI Software, pensado sobre la base de los requerimientos de la práctica diaria. DPI es un sistema de computación de última generación destinado a llevar un control total de Centros de Diagnóstico por Imágenes (ecografías, tomografías, radiografías, etc.) tanto de la parte médica como de la contable y comercial. Una copia de este sistema tiene un precio de aproximadamente 1000 dólares para 10 estaciones simultáneas y solo incluye la gestión de información.

TIMSA, empresa mexicana con más de 20 años de experiencia en ofrecer soluciones integrales para la administración de distintas áreas del Sector Salud. Desarrolla e implementa soluciones, facilitando el control administrativo y médico, enfocadas a cubrir las necesidades de ese sector. Su principal producto, Softmed, es una solución informática integral para hospitales que posee los siguientes componentes: TIMLAB, para la gestión de laboratorios clínicos; TIMCONSUL, para la informatización de consultorios médicos; TIMHEMO, para la informatización de los bancos de sangre; TIMCLINIC, para la gestión de información clínica; TIMPATO, para laboratorios de patologías y TIMIMAGEN, para la gestión de información en gabinetes de imagenología.

### **1.4.2 En Cuba**

La mayor concentración de esfuerzos para el desarrollo de software que existe en Cuba actualmente vinculado a la salud se encuentra en la UCI, aunque se desarrollan productos en centros de alto prestigio actualmente.

Alianza Vital es un sistema de información para el control y el análisis de los pacientes atendidos en los servicios de cirugía pediátrica. Su aplicación garantiza la captación de información de interés para el control de los pacientes ingresados, operados, graves y crónicos, asegurando el seguimiento de estos pacientes por consulta externa. Posibilita además seleccionar afecciones específicas para analizar variables diversas tales como la morbilidad, la mortalidad, las complicaciones, la procedencia, las operaciones realizadas y otras. Por todo esto, el sistema constituye una herramienta para el análisis administrativo, asistencial y la investigación. Se ha desarrollado íntegramente para trabajar sobre plataforma web, con servidor de base de datos SQL Server y una red de bases de datos distribuidas para la consulta a nivel nacional de información con carácter estadístico y criterios de expertos.

La empresa Softel comenzó a desarrollar en el año 1997 el Sistema de Información Hospitalaria (SIH) con el objetivo de la informatización del sistema de salud cubano. Dicho sistema aun se encuentra en fase de desarrollo.

Otro de los sistemas en uso en nuestro país y de origen nacional es Imagis®, que aunque no se orienta en el área de gestión de información, sí lo hace en el área imagenológica. Imagis® es un sistema para la recepción, visualización, procesamiento y transmisión de imágenes digitales. Puede ejecutar operaciones sobre imágenes de Tomografía Axial Computarizada, Resonancia Magnética, Ultrasonido, Angiografía, entre otras. Se ha instalado en 21 instituciones de 9 provincias del país y en 2 países extranjeros: República Dominicana y Ecuador.

### **1.4.3 En la Universidad de las Ciencias Informáticas**

En la Universidad de las Ciencias Informáticas (UCI) existen dos facultades que se dedican al desarrollo de software vinculado a la salud. Existe además una dirección de producción propia para dirigir el trabajo relacionado con esta esfera y recientemente está en planes de desarrollo la creación del polo productivo dedicado a esta fundamental temática.

Durante los últimos 5 años se han realizado productos vinculados estrechamente con la salud en sus diversas ramas; entre las principales aplicaciones desarrolladas están las vinculadas con la Bioinformática, la Atención Primaria de la Salud, la atención en los hospitales, informatización de bancos de sangre, Nefrología y Fisioterapia, entre otros. En la facultad 7 de esta universidad se está desarrollando un sistema de gestión hospitalaria (GeHos). Este sistema está basado en el sistema SIH de la empresa Softel aunque con una propuesta arquitectónicamente diferente y empleando nuevas tecnologías para su desarrollo como Microsoft.Net y como gestor de Base de datos PostgreSQL.

Recientemente, también en esta facultad, se viene desarrollando una solución PACS para los principales centros hospitalarios que poseen alta tecnología en materia de imagenología. Este sistema, Cassandra PACS, contempla entre sus componentes, un Visor de Imágenes Médicas, Cassandra Viewer; un Servidor DICOM, Cassandra Server; un cliente de correo electrónico DICOM compatible, Cassandra DMail; una aplicación para la gestión de reportes, Cassandra XWeb. El sistema propuesto por el presente trabajo, Cassandra Clinic, constituye el componente de gestión necesario para completar la funcionalidad de dicho PACS, integrándose a éste sistema y asegurando la correcta gestión de la información imagenológica manipulada y organizando otros procesos de negocio, propios de instituciones hospitalarias esencialmente imagenológicas.

### **1.5 Tendencias y tecnologías actuales**

En sus inicios, los sistemas de gestión eran aplicaciones de escritorio, producto del gran auge sobre todo de las aplicaciones **Win32** y la poca explotación de la Web. Hoy en día podemos encontrar sistemas de esta naturaleza en completo funcionamiento y con un alto rendimiento y aceptables beneficios a las instituciones en las que se emplean; sin embargo la tendencia en estos días, luego del potenciamiento y la aceptación universal de la web como uno de los recursos, por excelencia, de centralización de contenido y de distribución de acceso, es el desarrollo de sistemas web, con arquitectura cliente servidor, lo que permite el acceso desde cualquier parte de la geografía mundial, con solo una estación de trabajo con prestaciones mínimas y un navegador web, a toda la información que antes requería la instalación de controladores, y grandes requisitos de memoria y otros recursos.

Las herramientas se han desarrollado para hacer de la web un espacio ya no solo de comunicación, sino también de negocios y un estilo de trabajo. Se han creado empresas que existen puramente en el ciberespacio y no físicamente. Empresas de soporte, consultoras especializadas, y muchas más, son

ejemplo del uso del “**e-business**” como alternativa de trabajo. El área de la salud y específicamente la de gestión de información imagenológica no se ha quedado al margen y hoy en día no es inusual encontrar varias decenas de compañías exitosas que brindan servicios ya sea de venta y soporte mediante la red de redes, exponentes de lo anterior lo constituyen las analizadas en la [sección 1.4](#) del presente capítulo.

### 1.5.1 Aplicaciones Web

Con la introducción de Internet y de la Web, se han abierto múltiples posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Este avance representa un desafío a los desarrolladores de aplicaciones, ya que la continua evolución de las tecnologías demanda cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar la Web<sup>2</sup>. Esta plataforma está basada en una arquitectura Cliente/Servidor y su funcionamiento básico en las solicitudes y respuestas que se envían a través de

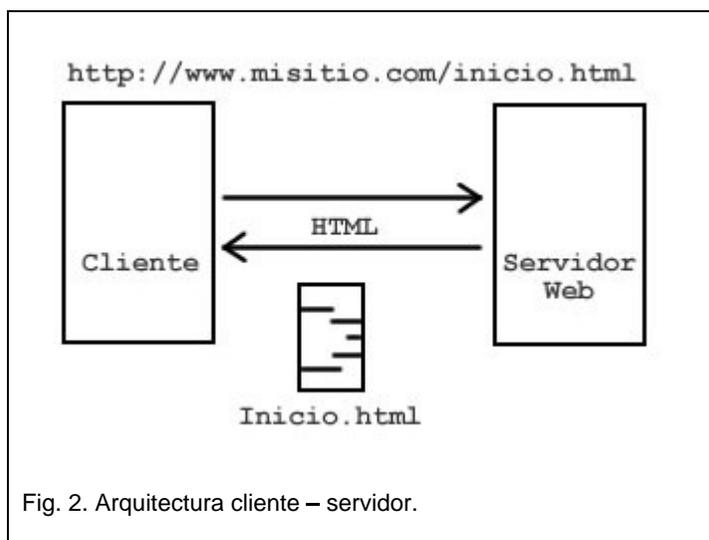


Fig. 2. Arquitectura cliente – servidor.

la red. Cuando un usuario que usa un navegador, hace una petición de una página al servidor Web, éste recibe dicha petición, la procesa y envía la respuesta al navegador en forma de **HTML** (Figura 2).

Las Aplicaciones Web por su parte constituyen un caso más complejo de esta plataforma. Estas aplicaciones se ejecutan en el servidor Web, encargándose de controlar el estado de un negocio o una problemática dada mediante el uso de uno o varios lenguajes de programación. De esta forma se generan páginas dinámicas cuyo contenido la mayor parte de las veces está vinculado con los datos almacenados en Bases de Datos Relacionales. Actualmente las Aplicaciones Web cuentan con gran popularidad debido al auge que tuvo Internet en la década pasada, y dado que brindan grandes posibilidades de despliegue en oposición a las Aplicaciones de Escritorio por el simple hecho que los usuarios solo necesitan un navegador para hacer uso de ellas. Además, estas aplicaciones permiten el aprovechamiento de todas las características de Internet, estando disponibles desde cualquier parte del mundo donde se tenga acceso a la Red de Redes. Son fáciles de usar, ya que no requieren conocimientos avanzados por parte de los usuarios. Por todo lo planteado, la solución propuesta consistirá en una Aplicación Web.

### 1.5.2 Lenguajes de Programación Web

Existen distintos tipos de lenguajes de programación que están vinculados con la Web, los mismos se pueden dividir en dos grandes grupos debido a la característica de esta plataforma:

- **Lenguajes del lado del servidor**

Los lenguajes del lado del servidor son aquellos, que se ejecutan en el servidor Web, antes de que se envíe la página a través de la red al navegador. Los códigos puestos en el servidor se encargan de construir las páginas que serán enviadas de respuesta al cliente, estos lenguajes pueden acceder a Bases de Datos, ficheros en el servidor, recursos en la red, etc. Estos códigos no son visibles por los clientes, ya que cuando se solicitan a través del Servidor Web lo que se genera es código HTML que es entendible y representable por un navegador Web.

Entre los principales lenguajes del lado del servidor podemos encontrar: PERL, PHP; las tecnologías Active Server Pages (ASP y ASP.Net), Java Server Pages (JSP), Common Gateway Interface (CGI), entre otros. Todos estos se caracterizan por la creación de páginas Web dinámicas, desarrollando la lógica del negocio en el Servidor Web.

**Perl** es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo web, programación en red, desarrollo de interfaces gráficas de usuario y más<sup>3</sup>.

Es muy utilizado para construir aplicaciones CGI para la Web. Es software libre, es un lenguaje de programación interpretado y extensible a partir de otros lenguajes.

**PHP** es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es un acrónimo recursivo para "**PHP Hypertext Pre-processor**" (inicialmente PHP Tools, o, Personal Home Page Tools), y se trata de un lenguaje interpretado, usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la biblioteca GTK+<sup>4</sup>. En PHP los scripts del lado del servidor se insertan dentro del código HTML. Es gratuito y multiplataforma y está ampliamente difundido en el mundo entre la comunidad de programadores.

**Java Server Pages (JSP)**, es una tecnología para crear aplicaciones web. Es un desarrollo de la compañía Sun Microsystems, y su funcionamiento se basa en scripts, que utilizan una variante del

lenguaje java. Esta tecnología permite a los programadores generar contenido dinámico para web, en forma de documentos HTML y/o **XML** e incrustar en el contenido estático de las páginas Web código Java y comando predefinidos. Con JSP se pueden crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas.

**Common Gateway Interface (CGI)** es una importante tecnología que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa. Las aplicaciones CGI fueron una de las primeras maneras prácticas de crear contenido dinámico para las páginas web. En una aplicación CGI, el servidor web pasa las solicitudes del cliente a un programa externo. La salida de dicho programa es enviada al cliente en lugar del archivo estático tradicional<sup>5</sup>. Se convirtió en un estándar y tomó un gran auge debido a la posible implementación de nuevas funcionalidades en las páginas Web.

**Active Server Pages (ASP)** es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS)<sup>6</sup>, el servidor de la gigante compañía de software Microsoft. ASP ha pasado por cuatro iteraciones mayores, ASP 1.0 (distribuido con IIS 3.0), ASP 2.0 (distribuido con IIS 4.0), ASP 3.0 (distribuido con IIS 5.0) y ASP.NET (parte de la plataforma .NET de Microsoft). Las versiones pre.NET se denominan desde 2002 ASP *clásico*. Las páginas ASP al igual que las PHP tienen incrustado el código del lado del servidor dentro del HTML, en este caso, se puede usar Visual Basic Script o JScript. Entre las ventajas que tiene ASP se puede mencionar que es muy fácil de aprender para el que es familiar con cualquier lenguaje de programación, por ejemplo, gracias a que usa principalmente VBScript, cualquiera que conoce como programar en Visual Basic, le resulta relativamente fácil tratar con él. El soporte de ASP también se encuentra habilitado por defecto en el servidor IIS, facilitando su instalación y ejecución.

**ASP.NET** es un conjunto de tecnologías de desarrollo de aplicaciones web comercializado por Microsoft. Es usado por programadores para construir sitios web domésticos, aplicaciones web y servicios web XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP)<sup>7</sup>. Comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. Usando ASP, se puede programar del cliente en C#, Visual Basic.NET, o con cualquier lenguaje tradicional del lado del cliente. ASP.NET, constituye una de las formas más rápidas y escalables de implementar y ejecutar aplicaciones Web, esta es una de las razones por las cuales la productividad de

los programadores ha aumentado. Además, se ha convertido en una de las primeras opciones que eligen estos programadores, para desarrollar aplicaciones Web. ASP.NET ofrece varias ventajas importantes sobre los modelos de programación Web anteriores:

- Mejor rendimiento.
  - Compatibilidad con herramientas de primer nivel.
  - Eficacia y flexibilidad.
  - Simplicidad.
  - Facilidad de uso.
  - Escalabilidad y disponibilidad.
  - Posibilidad de personalización y extensibilidad.
  - Seguridad.
- **Lenguajes del lado del cliente**

Los lenguajes del lado del cliente, por otra parte, son los que se ejecutan en el cliente o navegador. Estos son los encargados de darle dinamismo a la página sin necesidad de realizar viajes al servidor para realizar operaciones sencillas que no siempre lo requieren. Estos lenguajes son interpretados, pueden acceder a la información HTML que se muestra en un navegador pudiendo modificarla y actualizarla según las necesidades de los programadores. Entre los lenguajes del lado del cliente se encuentran principalmente JavaScript y Visual Basic Script.

**JavaScript** es el lenguaje de programación del lado del cliente más utilizado por ser compatible con la mayoría de los navegadores modernos. Es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, se puede acceder a los elementos que componen la página Web, permitiéndole al programador modificar el contenido de la página dinámicamente.

**Visual Basic Script**, por su parte, es un lenguaje que, a diferencia de JavaScript, es solamente compatible con Internet Explorer, aunque posee toda la funcionalidad que brinda JavaScript.

### 1.5.3 Servidores Web

**Internet Information Server (IIS)**, es el principal servidor de aplicaciones web de Microsoft. Sus principales funcionalidades son la publicación de sitios y aplicaciones Web, sitios **FTP**, **SMTP** (Simple Mail Transport Protocol) y Servicios de noticias. Dispone de soporte necesario para crear páginas en ASP. Su principal problema radicaba en la pobre seguridad de sus primeras versiones, aspecto que fue resuelto en la versión 6.0, en la que Microsoft ha cambiado el comportamiento de los controles **ISAPI** pre - instalados. Es un sistema esencial destinado a la utilización de los servicios de Internet basado en la plataforma Windows.

**Apache** es el principal servidor Web del mundo del software libre. Después de la segunda mitad de la década del 90 ha tomado un gran auge en las aplicaciones web, fundamentalmente en el soporte de aplicaciones programadas en PHP. Entre sus principales características están la flexibilidad para configurar los mensajes de error, contiene sus propias bases de datos de autenticación y negociado de contenido. Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

### 1.5.4 AJAX

AJAX es un acrónimo para Asynchronous JavaScript and XML HttpRequest, es más que una tecnología de desarrollo Web, el resultado de la unión de varias tecnologías existentes con el fin de crear aplicaciones interactivas sin la necesidad de cargar la página completamente entre una solicitud y otra. Las peticiones se realizan asíncronamente al servidor web, provocando un aumento considerable de la experiencia del usuario, acercándolo cada vez más a la experiencia de trabajo con una aplicación de escritorio. Además, debido a que se reduce el tamaño de los envíos de paquetes entre el servidor y el cliente, se gana en rapidez e interactividad a medida que se manipula la interfaz del navegador con el lenguaje JavaScript.

### 1.5.5 Sistemas Gestores de Bases de Datos (SGBD).

Los **Sistemas de gestión de base de datos** son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta<sup>8</sup>. La bibliografía especializada a menudo se refiere a estos sistemas como SGBD o DBMS, siendo ambos equivalentes y

acrónimos de Sistema Gestor de Bases de Datos y *DataBase Management System*, respectivamente. En la actualidad existe una gran variedad de SGBD, tanto de tipo comercial como libre. Entre los más usados dentro del grupo de los comerciales se encuentran:

- **Microsoft SQL Server**

Es un sistema de administración de bases de datos relacionales producido por Microsoft. Su principal lenguaje de consulta es Transact – SQL, una implementación del Lenguaje de Consulta Estructurado ANSI/ISO usado tanto por Microsoft como por Sybase. SQL Server es comúnmente usado por empresas que necesitan de pequeñas a medianas bases de datos, aunque en los últimos años ha sido ampliamente adoptado por empresas que poseen grandes bases de datos empresariales. Microsoft SQL Server tiene su base (antes de la versión 7) en Sybase SQL Server y constituyó la entrada de Microsoft hacia el mercado de bases de datos de nivel empresarial, compitiendo con Oracle, IBM, y luego, el propio Sybase. Su última versión SQL Server 2005, es un potente gestor de bases de datos, que incorpora numerosas características completamente novedosas, lo que lo hace un producto extremadamente costoso.

- **PostgreSQL**

Es un Sistema de Administración de Bases de datos de objetos relacionales. Pertenece al movimiento de software libre y fue lanzado bajo la licencia BSD (Berkeley Software Distribution). Ofrece una alternativa a los demás sistemas de administración de bases de datos. Al igual que otros proyectos de software libre como Apache, GNU Linux y Media Wiki, PostgreSQL no está controlado por una sola compañía, sino que cuenta con comunidad global de desarrolladores y compañías para su evolución. Se conoce además como Postgres, que fue su nombre de original y su origen se remonta a que inicialmente fue concebido como post – Ingress database (base de datos de post ingreso). Sus autores desarrollaron además el sistema de base de datos Ingress. Entre sus principales características podemos citar:

- **Funciones.**

Las funciones permiten que bloques de código sean ejecutados por el servidor. Estas funciones pueden ser escritas en los siguientes lenguajes:

- Un lenguaje incorporado llamado PL/pgSQL que re ensambla el lenguaje procedural de Oracle PL/SQL.
- Lenguajes de script como PL/Perl, pIPHP, PL/Phyton, entre otros.

- Lenguajes compilados como C, C++ o Java a través de PL/Java.
- El lenguaje estadístico R a través de PL/R

- Índices

Se pueden crear índices por el usuario o pueden usarse los índices incorporados, B-tree, hash table y GiST.

- Triggers

Los triggers están completamente soportados y pueden ser adjuntados a tablas pero no a vistas. Las vistas pueden tener reglas. Varios triggers son lanzados en orden alfabético y los triggers pueden invocar funciones escritas en otros lenguajes además de PL/pgSQL.

- MVCC

PostgreSQL administra la concurrencia a través de un sistema conocido como MVCC (Multi Version Concurrency Control) que le da al usuario una especie de foto fija de la base de datos que le permite hacer cambios sin ser visibles a otros usuarios hasta que la transacción no sea completada. Esta técnica elimina la necesidad hacer bloqueos de lectura y asegura los principios ACID (**Atomicidad**, **Consistencia**, **Aislamiento**, y **Durabilidad**) de la base de datos de una manera eficiente.

- Reglas

Las reglas permiten que el árbol de consulta de una consulta entrante sea reescrito. Un uso común es la creación de vistas actualizables.

- Tipos de Datos.

Se soportan varios tipos de datos nativos como:

- Tipos numéricos de precisión arbitraria.
- Texto de longitud ilimitada.
- Primitivas geométricas.
- Direcciones IP e IPV6.
- Tipos de datos de bloques CIDR (**Classless Inter-Domain Routing**) y direcciones MAC (Media Access Control).

- Arreglos.
- Objetos definidos por el usuario.
- **Oracle**

Larry Ellison fundó Software Development Laboratories en 1977. En 1979 SDL cambió el nombre de su compañía a Relational Software, Inc. (RSI) e introdujo su producto Oracle V2 como un sistema de administración de bases de datos incipiente disponible comercialmente. Esa versión no soportaba transacciones pero implementaba la funcionalidad básica de SQL, consultas y joins (uniones). RSI nunca lanzó una versión 1 de su producto. Lo que hizo fue llamar a su primera versión “versión 2” como un truco ingenioso de mercado.

Desde 1985 el Sistema de Administración de Bases de Datos Oracle DBMS comenzó a soportar el modelo cliente – servidor, cuando las redes se hicieron sentir en la mitad de la década de los 80, Oracle comenzó a soportar consultas distribuidas. En 1992 Oracle versión 7h (h significaba “datawareHouse”) apareció con soporte para la integridad referencial, procedimientos almacenados y triggers. En 1999 Oracle 8i apareció y tenía en la mira operar más de acuerdo con las necesidades de Internet. En 2001 Oracle 9i apareció en el Mercado con 400 nuevas características, incluyendo la posibilidad de leer y escribir documentos XML. 9i también brindaba una opción para Oracle RAC (Real Application Clusters), una base de datos de clústeres de computadoras, como reemplazo a Oracle Parallel Server (OPS) que había lanzado antes.

- **MySQL**

Es un sistema de administración de base de datos, multi - hilo, multiusuario, con más de seis millones de instalaciones por todo el planeta. MySQL es propietaria y patrocinada por una sola firma, la compañía sueca MySQLAB que mantiene el derecho de la mayor parte del Código base. La compañía desarrolla y mantiene el sistema, vende soporte y contratos de servicio, así como copias de licenciadas a propietarios de MySQL y emplea a personas de todo el mundo quienes colaboran vía Internet. MySQL AB fue fundada por David Axmark, Allan Larson y Michael “Monty” Wadenius, y también vende otro producto llamado MaxDB, que está formado por otra Base de Código no relacionada con la de MySQL.

Existen muchas API (Application Programming Interface – Interfaz de Programación de Aplicaciones) que permiten que aplicaciones escritas en varios lenguajes de programación accedan a bases de datos. MySQL es muy popular en aplicaciones Web y actúa como un componente de bases de datos para las

plataformas LAMP, MAMP y WAMP (Linux/MAC/Windows-Apache-MySQL-PHP/Perl/Phyton) y para herramientas de búsqueda e identificación de errores como Bugzilla. Para administrar bases de datos en MySQL pueden usarse herramientas de línea de comandos como mysql y mysqladmin. Descargables desde el sitio de MySQL están las herramientas basadas en Interfaz grafica de usuario (GUI), MySQL Administrador y MySQL Query Browser, estas dos incluidas en un paquete llamado MySQL GUI Tools.

MySQL trabaja en numerosas plataformas como AIX, HP-UX, GNU/Linux, Mac OS X, Novell NetWare, OpenBSD, OS/2, Solaris, SunOS, y todas las versiones de Windows. Su mayor desempeño se logra cuando se combina con el lenguaje de programación PHP.

### **Justificación de la selección.**

Después de haber analizado las características fundamentales de las tecnologías, lenguajes de programación y sistemas gestores de bases de datos candidatos para la implementación de la propuesta de este trabajo, seleccionamos la tecnología ASP.NET. Nuestra selección se basó fundamentalmente en su alto rendimiento y escalabilidad, la seguridad mejorada sobre otras tecnologías web existentes y la posibilidad de hacer despliegue de sistemas desarrollados con esta tecnología en ambiente tanto Windows como Linux, objetivo específico de nuestra investigación.

Se usará C# como lenguaje del lado del servidor, por sus grandes potencialidades, gran volumen de información y la experiencia creciente por parte de nuestro equipo de desarrollo con este lenguaje. Además está diseñado y optimizado para la plataforma .NET. C# es un lenguaje orientado a objetos, simple, elegante y con seguridad en el tratamiento de tipos<sup>9</sup>. Además, permite programar fácilmente a bajo nivel, acceder a las características avanzadas de la plataforma sobre la que trabajamos, crear código muy eficiente en aquellos puntos de la aplicación que son críticos y acceder a las interfaces de programación de aplicaciones (API) existentes.

Como lenguaje del lado del cliente se propone el uso de JavaScript, entre otras cosas por su universal aceptación y la limitante de portabilidad de Visual Basic Script al ser compatible solamente con Internet Explorer.

Como sistema de gestión de base de datos a utilizar se seleccionó el PostgreSQL en su versión 8.1. PostgreSQL es un potente gestor de Bases de Datos. Es software libre, corre en un gran número de sistemas operativos, siendo multiplataforma. Tiene todas las características de los SGBD modernos, como

son las llaves foráneas, vistas, procedimientos almacenados, varios tipos modernos de datos, etc.; posee una característica importante, que es la realización de múltiples “backups” o salvadas de la información.

PostgreSQL cuenta además con una característica llamada Write - Ahead Logging (WAL) que permite asegurar la atomicidad y durabilidad de la información. Dos de las características ACID de las bases de datos. Usando WAL las modificaciones son escritas primero a un log o bitácora antes de ser aplicadas a la base de datos. Esto garantiza que las operaciones de deshacer o rehacer se puedan ejecutar sin problemas y garantizar la integridad de los datos. Además de garantizar las otras dos características ACID, consistencia e integridad. Teniendo en cuenta que se trabajará en el área del software para la salud, y que la información que se manipulará será sensible y relevante, por ser información relativa a las personas; así como a los estudios y diagnósticos asociados a éstas, conviene entonces, escoger un DBMS que cumpla con las funciones de respaldo, y PostgreSQL deviene candidato ideal, además de que no se ha querido encarecer el sistema final usando un gestor como Oracle o SqlServer.

#### **1.5.6 Fundamentación de la metodología a utilizar.**

Para modelar la propuesta de este sistema, decidimos usar como metodología de desarrollo el Proceso Unificado de Rational (RUP), por sus ventajas en la organización de la información. RUP goza de un grupo de características y facilidades, que hacen más dinámico el desarrollo del trabajo. En nuestra universidad es muy usado, además que fue estudiado por nosotros durante la carrera. RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituyen la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado que constituye una propuesta de proceso para el desarrollo de software orientado a objetos, basándose en componentes. Sus principales características son:

- Iterativo e incremental: A medida que avanza el proceso de desarrollo se producen versiones incrementales, las cuales se acercan cada vez más al producto terminado.
- Guiado por los casos de uso: Los casos de uso son los que indican como debe actuar el sistema con el usuario final o con otro sistema para conseguir su objetivo.
- Centrado en la arquitectura: Los modelos son proyecciones del análisis y el diseño lo cual constituye la arquitectura del producto a desarrollar.

- Centrado en los Modelos: Los diagramas ayudan a expresar y entender mejor los elementos que componen un sistema que el lenguaje natural. Su objetivo es minimizar al máximo el uso de descripciones textuales.
- Forma disciplinada para la asignación de las tareas y responsabilidades (quién hace qué, cuándo y cómo).

RUP está compuesto por cuatro fases que son: Inicio, Elaboración, Construcción y Transición, cada una de ellas compuesta de una o varias iteraciones. A su vez está compuesto por nueve Flujos de Trabajo de Ingeniería: Modelamiento del negocio, Requerimientos, Análisis y diseño, Implementación, Pruebas y Despliegue, Administración de cambio y configuración, Administración de proyecto y Entorno. En cada una de sus fases se emplean todos los flujos de trabajo pero con diferente énfasis.

## **UML**

**Lenguaje Unificado de Modelado (UML**, por sus siglas en inglés) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está muy bien respaldado por **OMG** (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables<sup>11</sup>. Cabe destacar que UML es un lenguaje para especificar y no un método. Es utilizado para definir los componentes y artefactos en un sistema.

### **1.5.7 Paquetes de software y herramientas usados para la solución del problema propuesto**

Para la modelación, análisis, diseño e implementación del sistema que propondremos se usarán varios paquetes de software y herramientas de programación y modelación: herramientas de diseño guiado por computadora (CASE, por sus siglas en inglés) compiladores, entre otros.

Se usará una herramienta CASE, Enterprise Architect v.6.5, que permite la modelación de todos los flujos de trabajo de la metodología seleccionada, técnicas de ingeniería inversa contra gestores de bases de datos y múltiples lenguajes de programación, así como una gran cantidad de plantillas incluidas (o derivables) para la generación de la documentación necesaria.

Como Entorno de desarrollo integrado (IDE, por sus siglas en inglés), se usará Microsoft Visual Studio 2005 y como herramienta visual de administración del sistema gestor de bases de datos, se usará EMS PostgreSQL Manager 3. Otras herramientas usadas son depuradores (debuggers), para el traceo de las variables del **DOM** de los navegadores usados.

Uno de los objetivos de nuestra investigación es proveer a las clínicas imagenológicas de un entorno de intranet, donde se puedan publicar las noticias y los principales eventos de importancia del centro hospitalario. Para esto se ha decidido usar un Manejador de Contenidos (**CMS**, por sus siglas en inglés).

Los CMS son usados principalmente para manejar texto, gráficos, video y sonido, además traen incorporadas las funcionalidades administrativas de usuarios y roles, y otras funciones adicionales como foros de discusión, publicación de encuestas, etc. La tendencia de montar portales en CMS que brinden servicios de intranet ha crecido considerablemente en los últimos años llegando a ser una estrategia de rápido desarrollo.

En el caso de la presente solución, se optó por usar el CMS **Joomla!** Este CMS es uno de los más usados actualmente a nivel mundial, debido a que es muy fácil de configurar, muy potente en su gestión de la información y es libre.

### **1.5.8 Tecnologías auxiliares**

Las **hojas de estilo en cascada (CSS)** son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en **XHTML**). El **W3C** (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores<sup>10</sup>. Se usan las hojas de estilo en cascada, ya que nos ayudan mucho en la aplicación de los estilos propios que les daremos a las páginas, dándonos la posibilidad de modificar solamente una vez los colores, formas, e imágenes a aplicar a la interfaz. En resumen, las hojas de estilo permiten separar el formato visual de las páginas de contenido.

### **1.5.9 Herramientas auxiliares**

Uno de los requerimientos no funcionales del sistema, es la posibilidad de ejecución en entornos UNIX. Para la realización de pruebas de compatibilidad se ha empleado una novedosa herramienta desarrollada por los miembros del proyecto MONO, la alternativa para la plataforma UNIX de Microsoft .NET. Esta herramienta, denominada MoMA, es un acrónimo para Analizador de Migración a MONO, que permite el

análisis de los **ensamblados** y chequear el estado de las operaciones invocadas desde el código y que requieren de las funcionalidades del .NET Framework. Los resultados obtenidos de la realización de estas pruebas de compatibilidad se detallan en el [epígrafe 4.3](#) del Capítulo 4.

En este capítulo se ha fundamentado la selección de las tecnologías y herramientas que se utilizarán para la modelación, diseño e implementación del sistema propuesto por la presente investigación. Se ha hecho un estudio del estado del arte de los sistemas de gestión de información en el mundo, en nuestro país y específicamente en nuestra universidad; y se han argumentado las razones por las que se ha decidido utilizar la metodología de desarrollo del proceso unificado de Rational, conjuntamente con herramientas CASE, para modelar el problema planteado usando el lenguaje UML.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

En este capítulo se presentarán formalmente el problema científico, la situación problemática, el objeto de automatización, punto en el cual se describirán los procesos de negocio que serán objeto de automatización, así como una descripción de los documentos que se procesan y la información que se manipula. También se hará una descripción general de la propuesta del sistema que propone este trabajo de diploma y se hará una comparación con algunos de los sistemas existentes. Se expondrá además el modelo de negocio, la especificación de requerimientos funcionales y no funcionales del sistema y finalmente se mostrarán los diagramas de casos de uso.

### **2.1 Problema científico y situación problemática.**

El principal objetivo de las clínicas imagenológicas es brindar servicio médico de diagnóstico de alta confiabilidad basado en técnicas avanzadas de imagenología y alta tecnología.

Entre los principales procesos de negocio que soportan las clínicas imagenológicas se encuentran la realización de estudios imagenológicos a pacientes, la emisión de diagnósticos y, en algunas clínicas más especializadas, poder obtener una segunda opinión para el diagnóstico a través de un sistema de tele consulta médica. Este último proceso, aunque es muy importante, no se encuentra disponible en todas las clínicas imagenológicas. Para el correcto desarrollo de todos estos procesos de negocio se usa el PACS con que cuenta la institución. Otros servicios que se brindan a menor escala, pero no de menor importancia, son los de admisión, ingreso y egreso de pacientes. También se brindan servicios de cuerpo de guardia, emergencia, consulta externa, y oftalmología en algunos casos puntuales.

En los últimos meses se está llevando a cabo el proceso de instalación piloto de un PACS desarrollado en nuestro país, en distintos hospitales cubanos y venezolanos. Dicho PACS está compuesto por varios módulos, uno de ellos es el denominado repositorio de reportes o XWeb. Esta aplicación permite el acceso a los reportes imagenológicos emitidos desde el visor, además que permite la visualización de dichos reportes en distintos tipos de formatos como **PDF**, **RTF** y **PostScript**.

Desde que el paciente llega al hospital, se debe llevar un seguimiento de todos sus datos y la información que se ha generado durante su estadía en la clínica, ya sea por concepto de ingreso o por servicios ambulatorios. Por lo tanto, cuando un paciente se presenta en la clínica para ser ingresado con el objetivo de realizarle estudios imagenológicos, debe ser admitido en el centro, recogiéndose todos los datos del

mismo. Una vez que a un paciente se le orienta un estudio imagenológico, debe presentarse con la orden de estudio en el local del equipo, donde normalmente se forma una cola de pacientes que se organiza según el orden de llegada de los mismos. Cuando el paciente entra en la sala, el técnico le pide sus datos y los introduce en el equipo médico. Los resultados de los estudios se guardan en XWeb, o en otro sistema de gestión de reportes, si el especialista así lo desea. Los médicos a su vez pueden citar a los pacientes para un día determinado en un equipo imagenológico o para una consulta en un local definido.

Lo que sucede, es que las funcionalidades de XWeb son básicas, solamente de visualización de la información. La propuesta de este trabajo es una continuación a XWeb, permitiendo la introducción de toda la información en un repositorio de datos, desde la llegada del paciente a la institución, hasta su seguimiento futuro, facilitando la 2da opinión.

En una clínica que no cuente con un PACS y un sistema de gestión de información, sucede que cuando al paciente se le orienta un estudio, los datos personales de éste se recogen varias veces. Esto trae consigo la demora y redundancia de información en el local de imagenología. Este proceso se realiza en hojas de cargo, y se registra de forma manual. Por lo que puede haber deterioro, pérdidas y mala legibilidad de la información. Esto afecta la correcta y eficiente realización de los procesos de negocio.

El proceso de consulta externa (citas) se ve muy afectado también por el hecho de que la gestión de calendarios y citas es una labor complicada, y en clínicas pequeñas, quienes muchas veces la realiza es el mismo especialista; hecho que afecta grandemente su desempeño y desvía su línea principal de trabajo, que es el diagnóstico de pacientes, y no el de llevar su calendario de citas. En el mejor de los casos, esta labor de gestión la realiza personal designado para ello, (entiéndase secretarías, trabajadores sociales), esto, aunque ya libera el especialista de la responsabilidad de la gestión, no asegura la eficiencia, porque los involucrados en esta tarea deben chequear la existencia o no, de capacidades para turnos y citas; y como hemos dicho anteriormente, esto en pequeñas clínicas, pero con alta afluencia de pacientes, es una labor engorrosa.

Por tanto, con el sistema propuesto, los datos del paciente serán enviados a una base de datos gestionada eficientemente por el sistema de gestión de la información y que puede integrarse con el PACS instalado en la clínica y estos sistemas se encargarán de asegurar la atomicidad, integridad, disponibilidad y confiabilidad de la información imagenológica y hospitalaria mínima de los pacientes.

Consecuentemente desde que el paciente se presenta en la clínica se recogen sus datos, se verifica si no existe en la Base de Datos, o sea que fuera atendido anteriormente por la institución y posteriormente se procede a ingresar, consultar, orientar estudio o citar, según se requiera. Estos procesos se realizarán entonces asistidos por un sistema de gestión, que ordenará el flujo de los procesos y asegurará la correcta realización de estos.

## **2.2 Objeto de automatización**

Los siguientes son procesos del negocio que serán objeto de automatización por nuestro sistema:

- **Admisión de pacientes**

Proceso en el que se recogen los datos de los pacientes. En el momento de recoger los datos no se tiene en cuenta si el paciente ya estuvo admitido en la institución, corriendo el riesgo que los datos se repitan.

- **Ingreso de pacientes**

Este es el proceso mediante el cual se registra un paciente como que está ingresado en la clínica, registrando la fecha de ingreso, su diagnóstico al ingresar, también se registra la cama en la que el paciente estará ubicado.

- **Egreso de pacientes**

Mediante este proceso, como su nombre lo indica, se da de alta de la institución a un paciente previamente ingresado.

- **Orientación de estudios imagenológicos**

Este proceso es el que se lleva a cabo cuando un especialista determina que a un paciente se le debe hacer un estudio imagenológico y le orienta el mismo mediante una orden de estudio.

- **Reservación, reprogramación y cancelación de citas a pacientes.**

El médico puede citar un paciente, lo mismo para una consulta posterior, o para la realización de exámenes imagenológicos en un equipo determinado según la modalidad del examen orientado.

- **Visualización de reportes médicos**

Es el proceso que se lleva a cabo después que los pacientes se realizan los estudios imagenológicos y estos son guardados en el servidor de imágenes. Actualmente se lleva a cabo con el visor de imágenes médicas instalado en la clínica.

Como se ha mencionado, en las clínicas imagenológicas, debe existir un PACS, el cual se encargará de todo lo vinculado con la gestión de las imágenes médicas, es decir, cuando las imágenes se generan en un equipo médico, deben ser almacenadas en un servidor. A su vez deben existir estaciones dedicadas a la visualización de dichas imágenes, las cuales sirven para el diagnóstico médico y la segunda opinión.

Con todo lo mencionado anteriormente se garantiza la gestión de las imágenes, pero no de los datos de los pacientes y sus estudios, compréndase que se debe gestionar la información desde la llegada del paciente a la clínica. También debe ser objeto de informatización el proceso de gestión de citas en los equipos médicos y/o con especialistas y el proceso de consultas.

En las instituciones de salud, dedicadas al área imagenológica, toda la información manipulada es de extrema sensibilidad, debido a que se manejan datos personales muy importantes. Estos datos pueden pertenecer a personalidades o figuras relevantes de la política, o personalidades de diversas esferas y cuyo conocimiento por personal ajeno a la institución, puede ser de gran impacto. Entre los datos más importantes están por supuesto los relacionados con los padecimientos de los pacientes, el historial de su evolución y el tratamiento aplicado.

### **2.3 Propuesta de sistema.**

De acuerdo con los estudios realizados y luego de materializar un profundo análisis del objeto de estudio, hemos concebido implementar un sistema para gestionar la información generada y manejada en las clínicas imagenológicas de acuerdo a las necesidades. El sistema debe entrar en función desde que el paciente llega a la clínica, hasta que se retira de la misma. Registrando todos los servicios que se le presten al paciente durante su estancia en la institución, permitiendo de esta forma el análisis estadístico en un futuro, además el seguimiento médico que se le realizará.

Es imprescindible además que no solamente los pacientes como principales beneficiarios de este sistema, noten una mejoría importante en los procesos del negocio, sino que además para los especialistas y los demás usuarios del software, sea una ventaja su instauración y puesta en práctica. Para lograr esto, los requisitos funcionales se han capturado partiendo de la necesidad real de los procesos de negocio de las instituciones imagenológicas, de manera que a la hora de implementar la solución, evitemos caer en anti -

patrones de software, o intentemos implementar el gran sistema, lo que a la postre traerá consigo el rechazo y la no utilización de la solución informática, evitando el cumplimiento del objetivo fundamental de este trabajo.

Existen varios sistemas dedicados al mismo campo que la solución aquí propuesta. Ejemplos de los existentes se han mencionado en epígrafes anteriores. Entre los más significativos se encuentra CardioWeb.

CardioWeb es un software cubano diseñado por la Empresa de Servicios Informáticos de Holguín (ESI en aquel entonces, año 2003) y el hospital Vladimir Ilich Lenin, dirigido a la informatización de los servicios de cardiología y cirugía cardiovascular, que se ha implantado con éxito en los cardiocentros de Villa Clara, Santiago de Cuba y en el Instituto de Cardiología y Cirugía Cardiovascular en Ciudad de La Habana, el CIMEQ, el Hospital Hermanos Ameijeiras, entre otras instituciones. Según sus principales autores, la ingeniera Yamina Aldana y el cardiólogo Delfín Rodríguez, este sistema forma parte del proyecto de informatización de la sociedad cubana y permite la telemedicina. Es decir, que se puede obtener opinión de expertos sin importar la ubicación geográfica de ellos, tanto a nivel nacional como internacional. El sistema permite además hacer un seguimiento al paciente desde que ingresa a una institución y posteriormente a través de consultas externas.

Otro de los sistemas a los que se puede hacer referencia es el SIH, de la empresa de software Softel, que tiene su génesis en 1997, al trazarse una nueva estrategia dentro del sistema de salud cubano cuyo objetivo principal era la informatización del Sistema Nacional de Salud. Para lograr esto se evidenció la necesidad de desarrollar herramientas y sistemas de software, es entonces que se propone la creación de un sistema de información para los hospitales. Luego de algún tiempo de desarrollo se crea el Sistema de Información Hospitalaria (SIH), que fue desarrollado en diferentes centros y llevado a un posterior mantenimiento dentro de la Universidad de las Ciencias Informáticas (UCI). Este sistema está implementado sobre Microsoft Visual Basic 6.0 y Borland Delphi 6.0 y usa como gestor de bases de datos Microsoft SQL Server 2000.

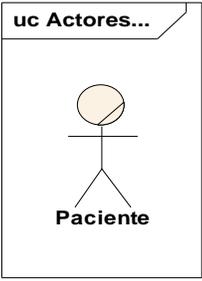
GalenLab, también de la autoría de Softel, es otro de los sistemas a los que se puede hacer alusión. Está orientado a la gestión de medios de diagnóstico, dígase, laboratorio clínico, radiología, ultrasonografía, electrocardiología, endoscopia y SUMA (Sistema Ultra Micro Analítico). Está ampliamente desplegado en Venezuela, en la mayoría de los Centros de Diagnóstico Integral (CDI). GalenLab también permite el seguimiento de pacientes durante su ingreso y la gestión de citas de los diferentes departamentos, pero

no vincula en su radio de acción los estudios imagenológicos, sino solamente la información de captura y entrega de resultados a los pacientes mediante dispositivos de impresión, y la generación de estadísticas. Además, no tiene en cuenta la información generada en los equipos médicos; así como tampoco implementa el estándar DICOM 3.0, ni es HL7 compatible.

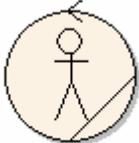
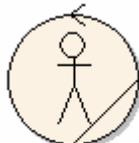
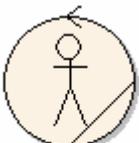
El sistema que aquí se propone, está orientado no solamente a la información cardiológica y personal de los pacientes, ni a la automatización de medios de diagnóstico o al control completo de la información hospitalaria; sino a la gestión de la información imprescindible, de orden imagenológico, y la necesaria para hacer el seguimiento de un paciente durante la estadía en una clínica imagenológica, con servicios generales y posteriormente al egreso, mediante consultas externas. Se concibe además la gestión semi - automatizada de las citas y los calendarios de los especialistas. La integración y comunicación con sistemas de control de Listas de Trabajo (**WorkList Server**), PACS, Repositorios de Reportes y se han trazado objetivos concretos de desarrollo de un producto compatible con los estándares mundiales de salud y comunicación de imágenes médicas HL7 2.x y DICOM 3.0 respectivamente, lo que facilitaría la integración de la solución con productos de software desarrollados y que se encuentran en funcionamiento, y le proporcionarían un alto valor agregado y una alta calidad.

## 2.4 Modelo de negocio

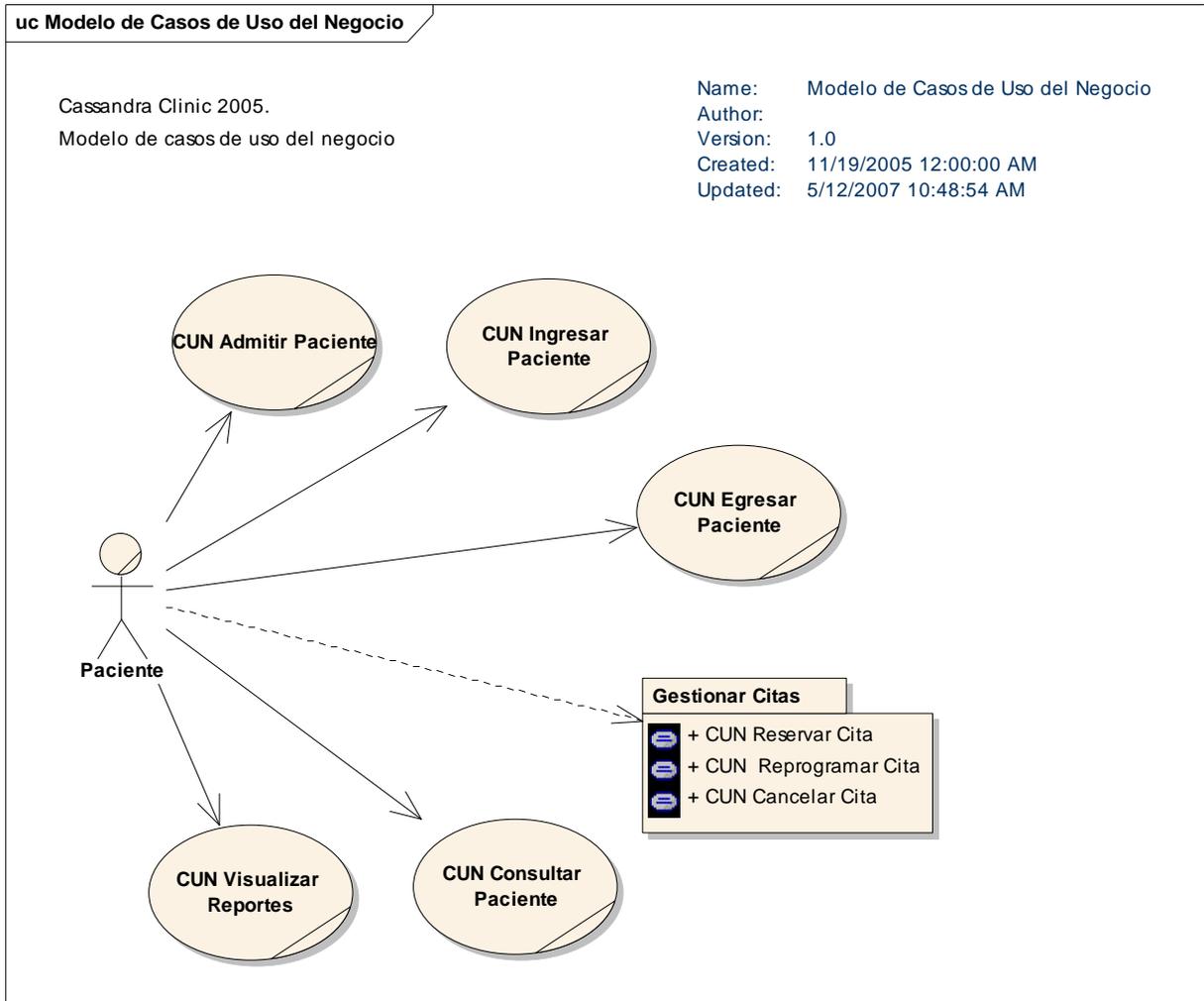
Los actores del negocio son aquellas personas o sistemas que obtienen un resultado de valor de uno o varios procesos del negocio.

Actores del negocio	Justificación
 <p>uc Actores...</p> <p>Paciente</p>	<p>Interviene en el proceso de negocio.</p> <p>Se beneficia con los servicios médicos que brinda la clínica imagenológica.</p> <p>Es el centro de la gestión de la información y el protagonista principal de los procesos de negocio.</p>

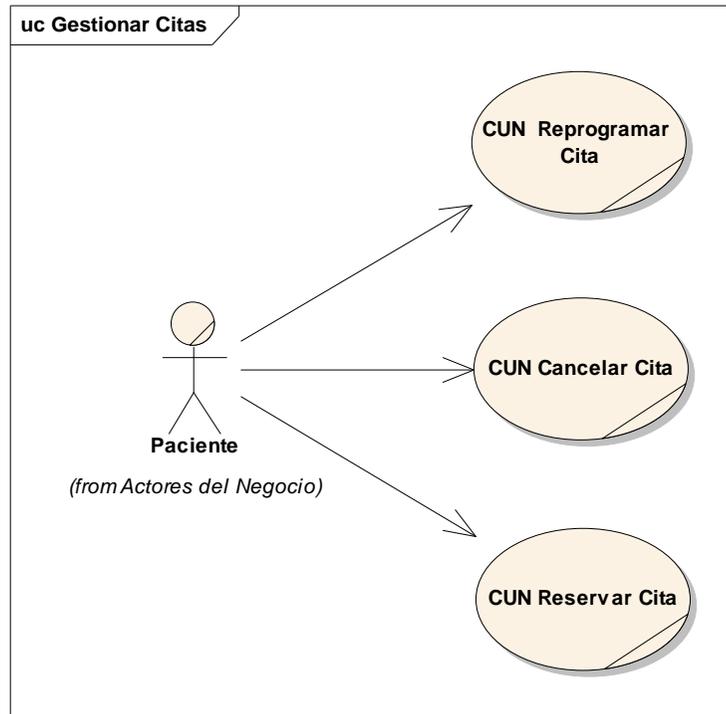
Los trabajadores del negocio, por otra parte, son aquellas personas o sistemas que están involucrados en uno o más procesos del negocio, que participan en ellos, pero no obtienen ningún resultado de valor. A continuación se muestra en la siguiente tabla:

Trabajadores del negocio	Justificación
 <p data-bbox="199 527 370 548">Trabajador Social</p>	<p data-bbox="594 369 1495 537">Interviene en el proceso recogiendo información socio – económica del núcleo familiar de los pacientes. En países como Cuba y Venezuela, esta información puede ser usada en la toma de decisiones de las misiones y los programas sociales.</p>
 <p data-bbox="237 768 337 789">Secretaria</p>	<p data-bbox="594 611 1495 737">Interviene en el proceso de negocio en la admisión de pacientes, reservación, cancelación, reprogramación de citas y gestión de calendario de los especialistas.</p>
 <p data-bbox="224 1010 342 1031">Especialista</p>	<p data-bbox="594 852 1495 978">Interviene en el proceso brindando los servicios de consulta, ingreso, egreso, modificación y actualización de la historia clínica con la información imagenológica.</p>
 <p data-bbox="199 1251 399 1272">Servicio Web - XWeb</p>	<p data-bbox="594 1094 1495 1220">Subsistema externo que interviene en el proceso de negocio, en la visualización y localización de los estudios imagenológicos y en la gestión de los reportes médicos.</p>

## 2.5 Diagrama de casos de uso del negocio



A continuación se muestra el modelo de negocio para el paquete Gestionar Citas de la imagen de arriba.



Modelo de negocio del Paquete Gestionar Citas.

## 2.6 Diagramas de actividades de los Casos de Uso del negocio

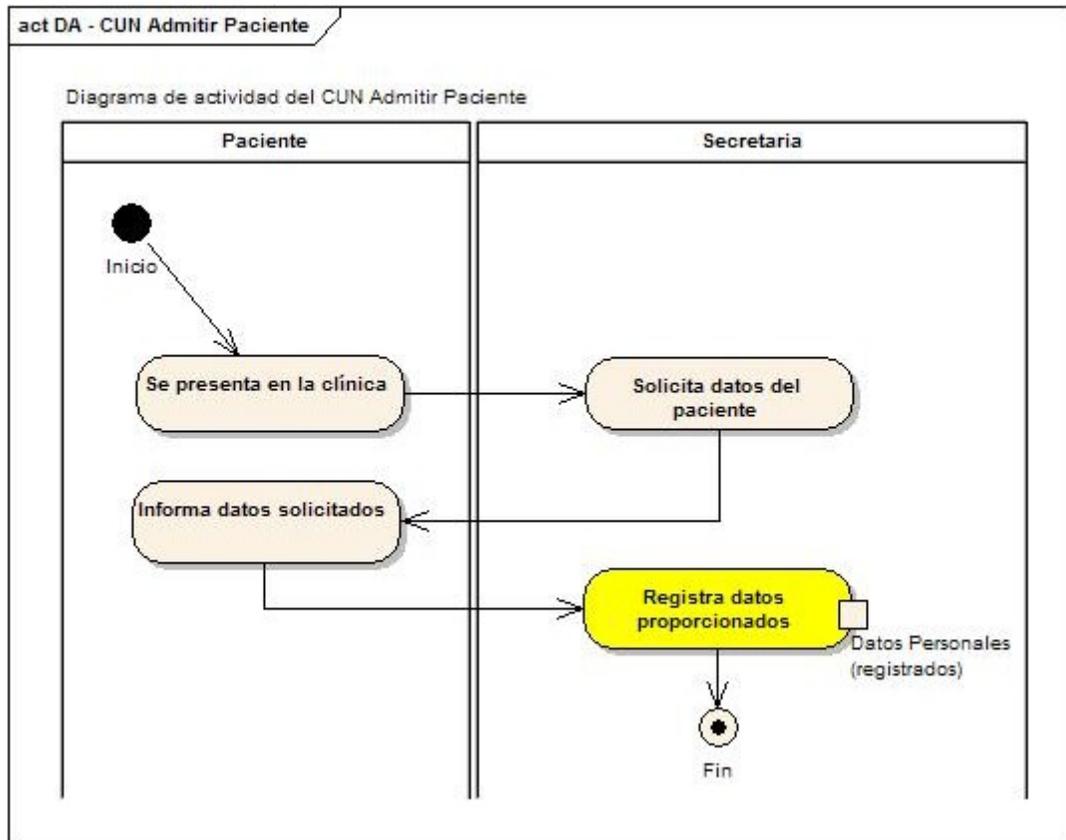


Diagrama de actividades – CUN Admitir paciente.

Diagrama de actividades - CUN Ingresar paciente

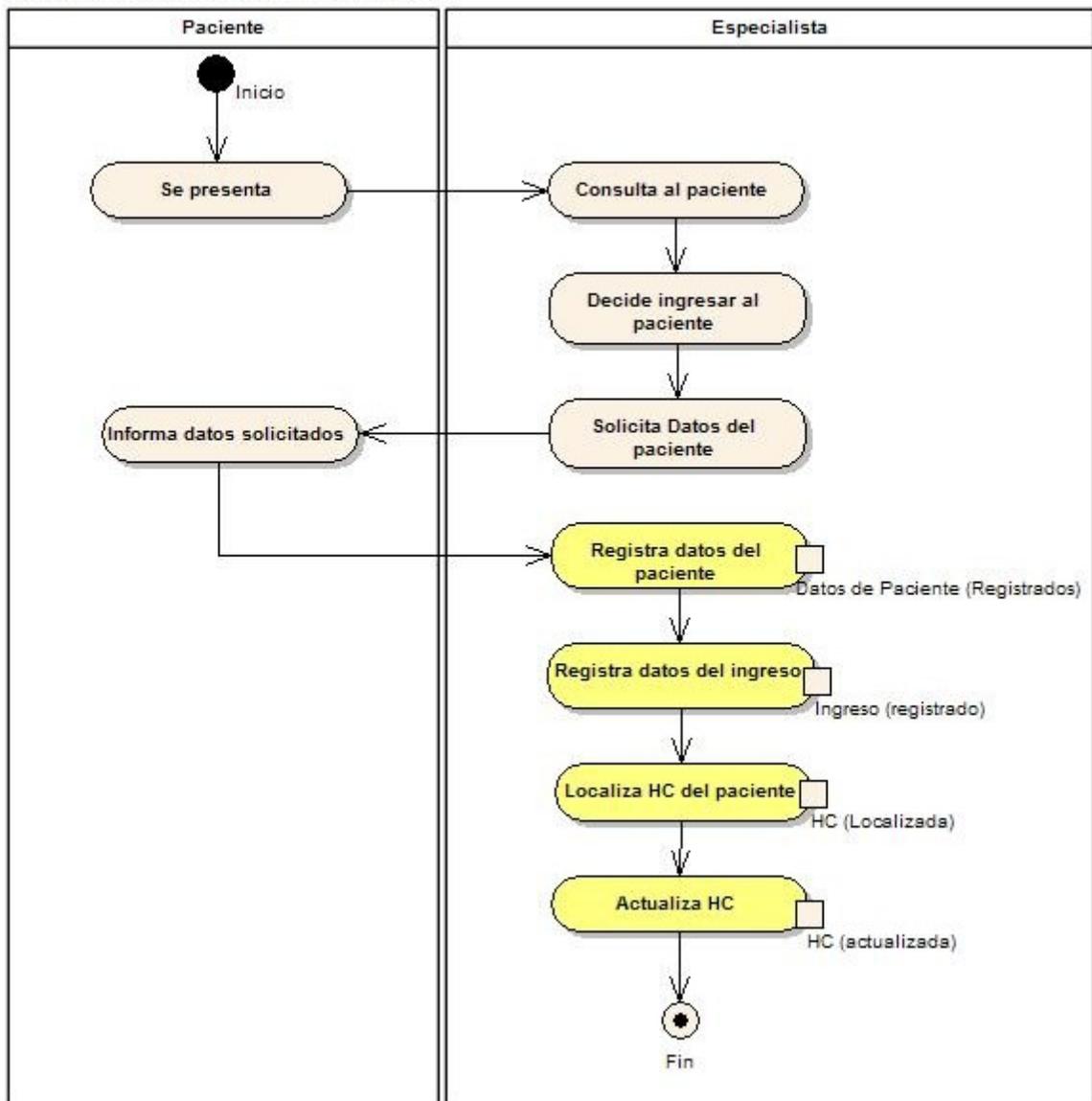


Diagrama de actividades – CUN Ingresar paciente

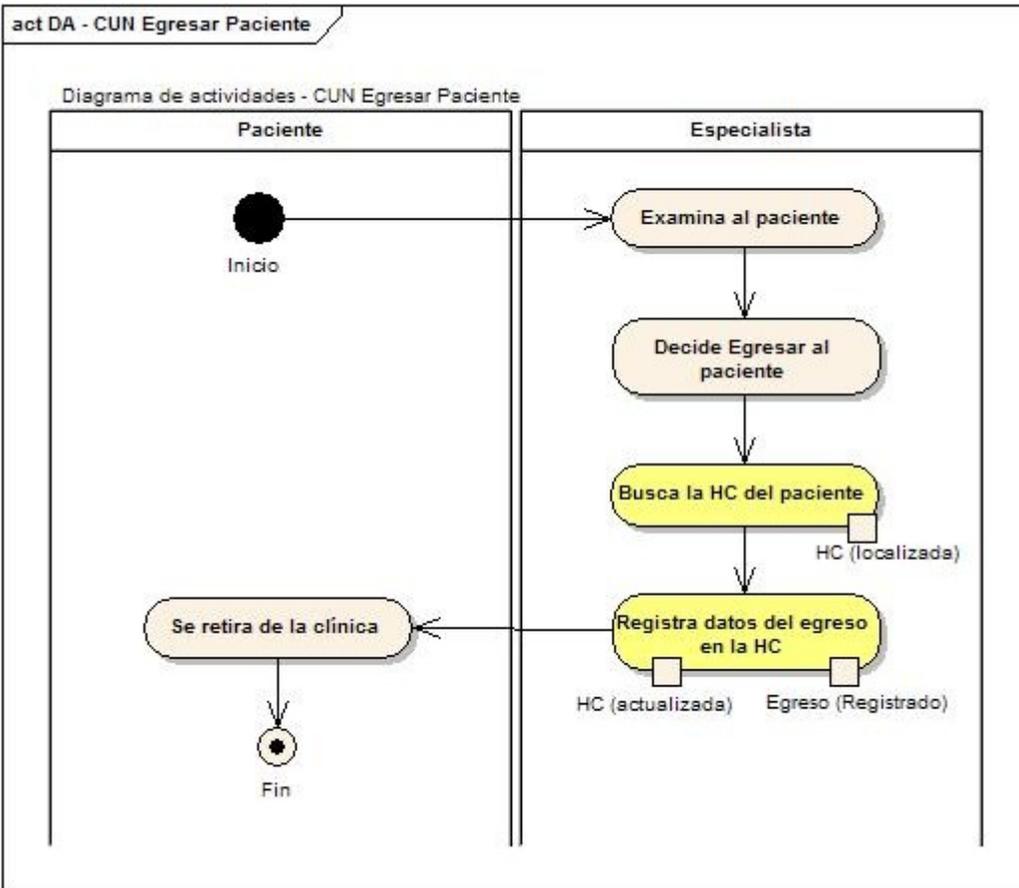


Diagrama de actividades – CUN Egresar paciente

Diagrama de actividades - CUN Consultar paciente

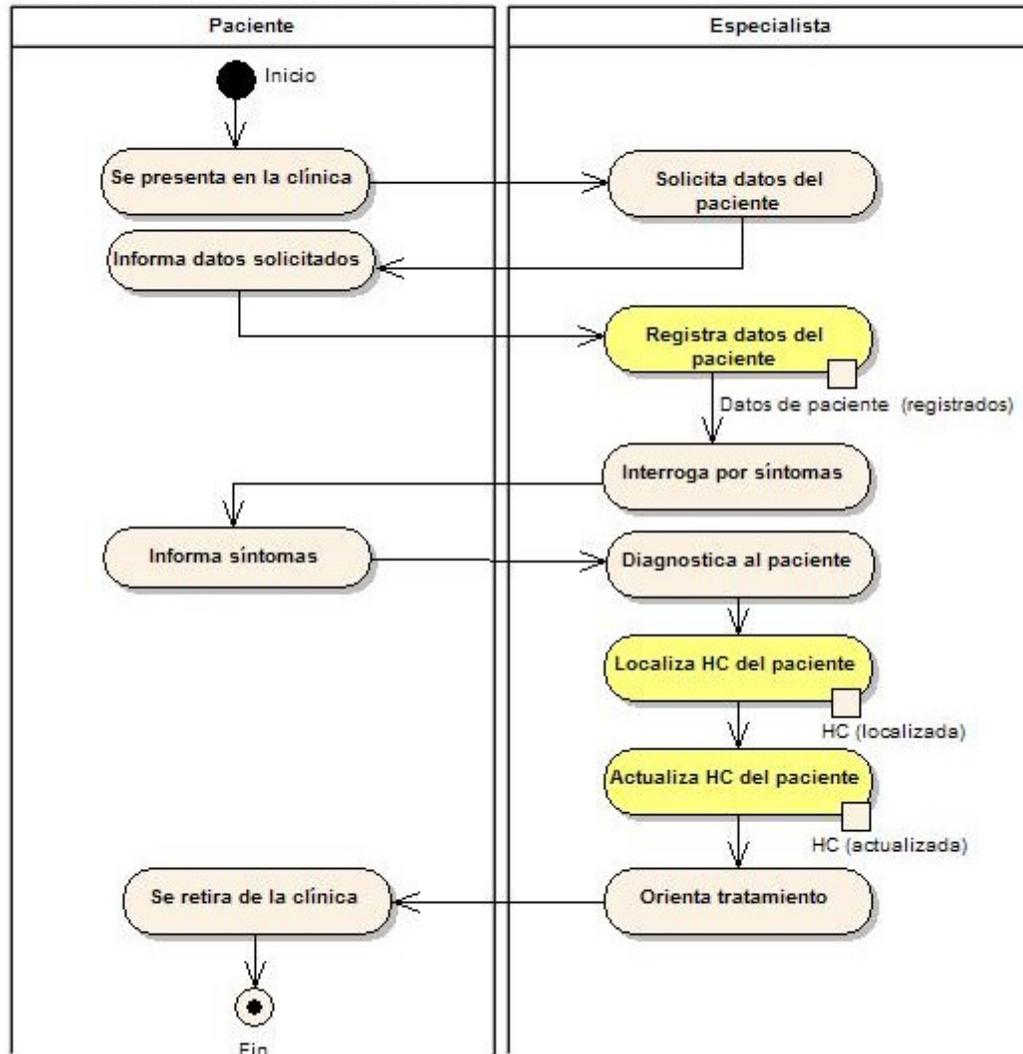


Diagrama de actividades – CUN Consultar paciente

Diagrama de actividades. CUN - Reservar Cita

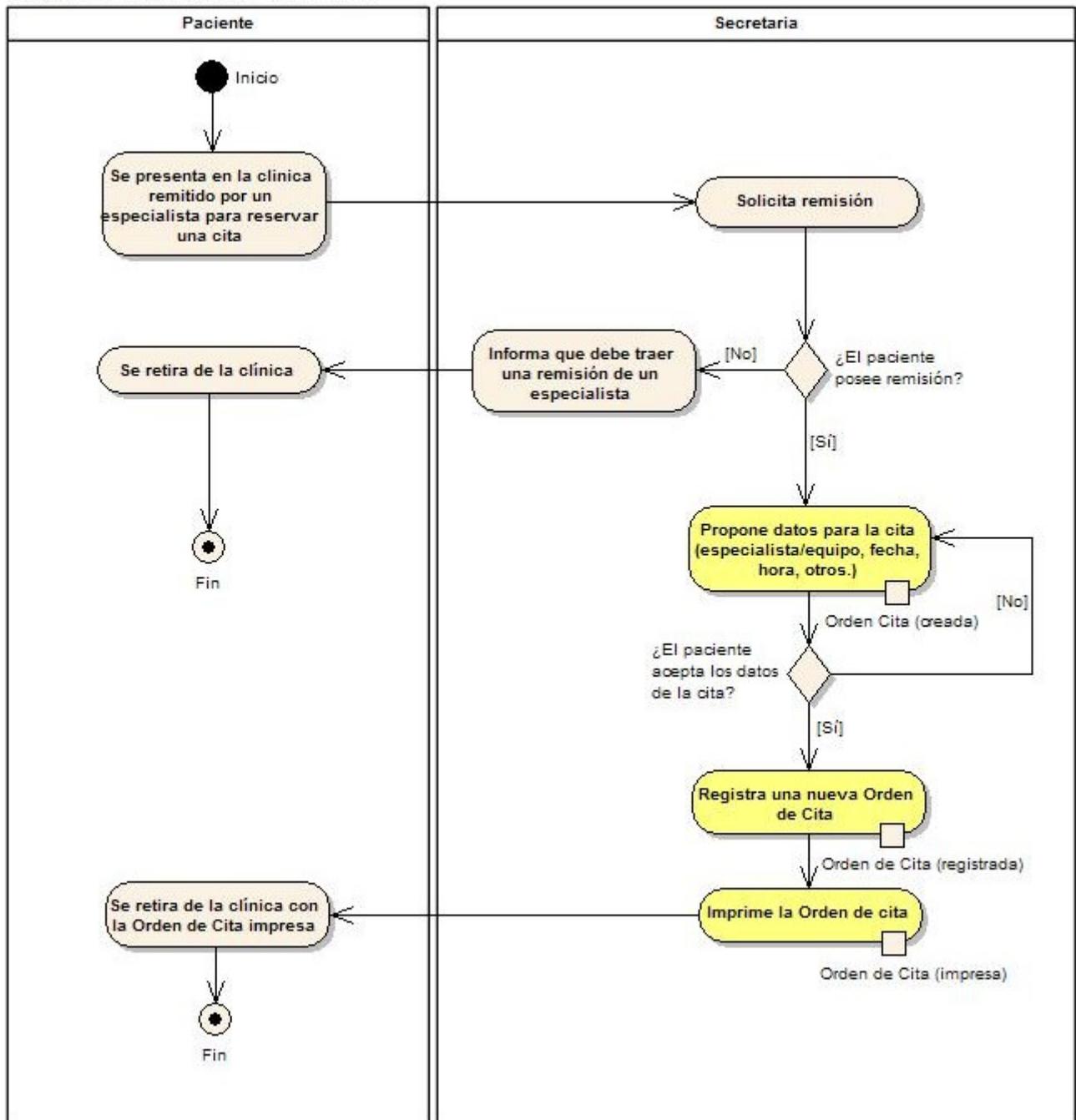


Diagrama de actividades – CUN Reservar cita

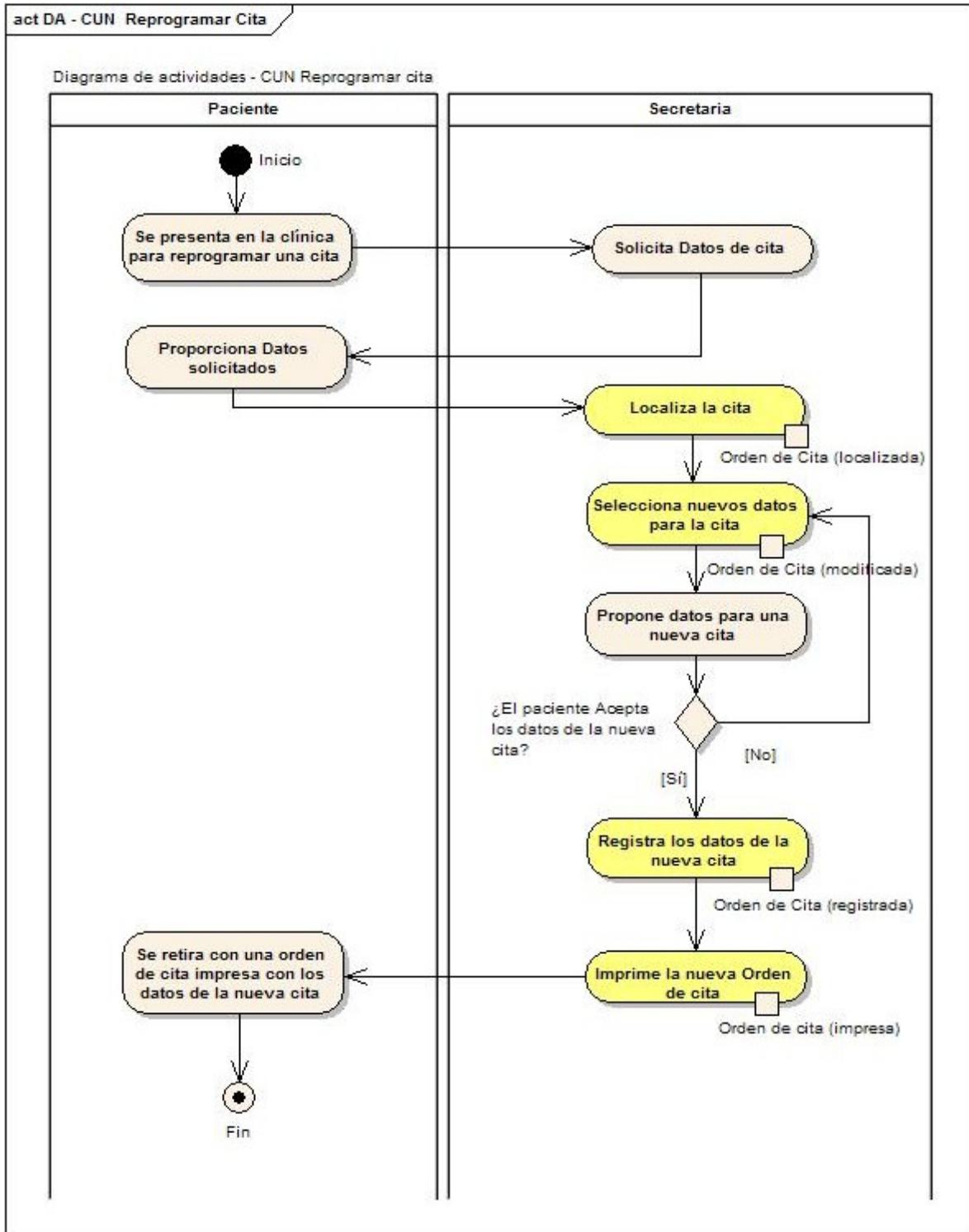


Diagrama de actividades – CUN Reprogramar cita

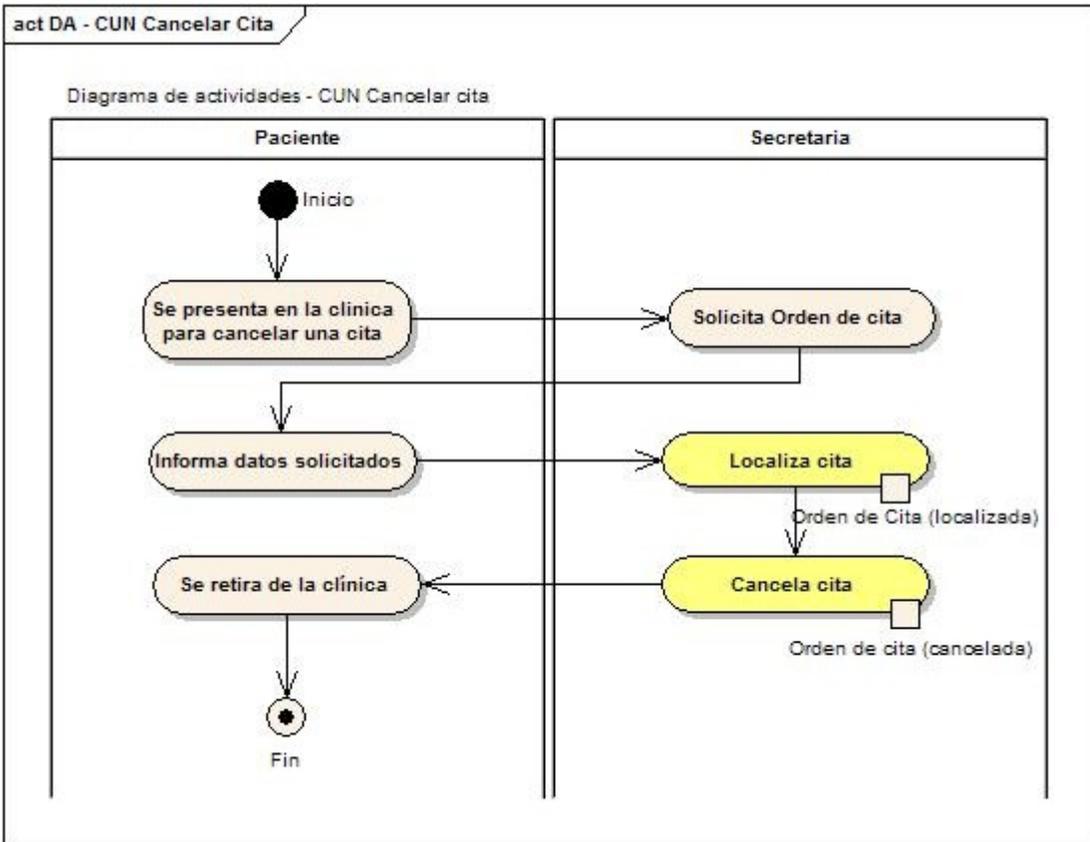
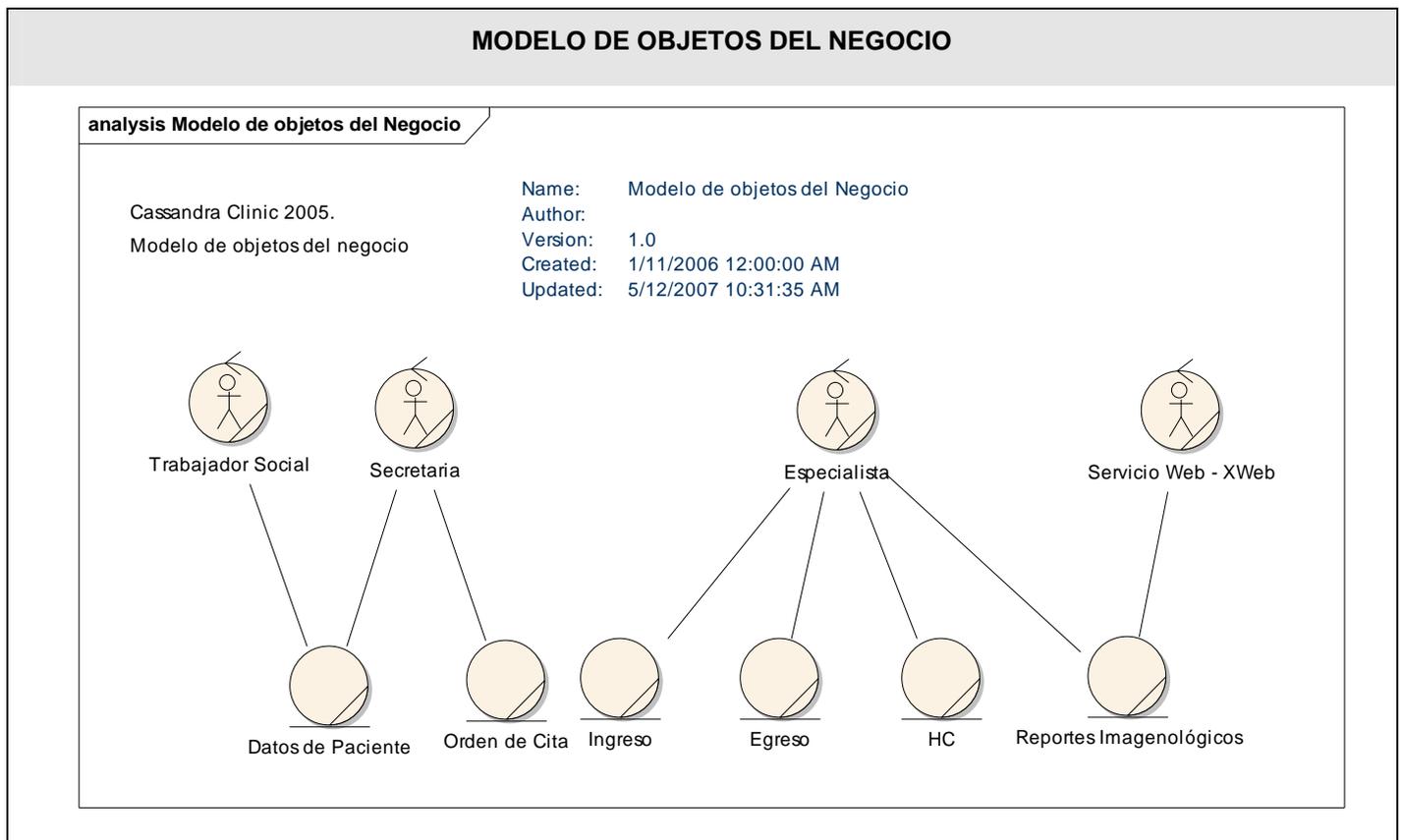


Diagrama de actividades – CUN Cancelar cita

## 2.7 Diagrama de Clases del Modelo de Objetos

El modelo de objetos del negocio expresa las relaciones que existen entre los trabajadores del negocio y las entidades del mismo.



## 2.8 Especificación de los requisitos de software.

Cassandra Clinic es un sistema que, aunque está concebido como un producto independiente, forma parte de Cassandra PACS, es integrable a éste y puede cooperar con los demás elementos del mismo. Por consiguiente tiene una relación con el componente de gestión de reportes XWeb. De este servicio web recibiremos información de reportes médicos, así como imágenes que pertenezcan a determinado paciente, para permitir con esto el seguimiento y actualización de la información imagenológica y la visualización de los reportes emitidos por especialistas en ocasiones anteriores.

Otro de los componentes con los que pudiera interactuar Cassandra Clinic es con el Cassandra WorkList Sever. Esta interacción se producirá al solicitar una cita para la realización de un estudio en un equipo médico, para un paciente. Esta solicitud se realizará a través de la interfaz de Cassandra Clinic y luego a través de una solicitud HL7 se informará al servidor de listas de trabajo que debe reservarse una cita para el equipo solicitado en la fecha indicada.

## **2.9 Requerimientos Funcionales**

Los requerimientos funcionales del sistema expresan las operaciones que éste debe implementar para satisfacer a los clientes. El sistema aquí expuesto, consta de 9 requerimientos, tres de ellos representan paquetes de requerimientos: Gestionar citas, que se divide en las operaciones de reservación, reprogramación, cancelación, impresión, verificación y ejecución de citas; Gestionar Historia Clínica, que contempla las operaciones necesarias de actualización de la historia clínica general e imagenológica asociada con el sistema, y por último, Gestionar Ubicación, que engloba las funcionalidades de asignación y reasignación de camas de ingreso y temporales y la liberación de las mismas.

- 1. Buscar Paciente**
- 2. Admitir Paciente**
- 3. Ingresar Paciente**
- 4. Insertar Datos Socio Económicos de un Paciente**
- 5. Mostrar Ingresos**
- 6. Egresar Paciente**
- 7. Gestionar Citas.**
  - 7.1.Reservar Cita con especialista
  - 7.2.Reservar Cita para Equipo
  - 7.3.Reprogramar Cita
  - 7.4.Cancelar Cita
  - 7.5.Imprimir Cita
  - 7.6.Verificar Cita

7.7.Pasar Paciente Citado a Consulta

7.8.Listar Pacientes Citados

## **8. Gestionar HC.**

8.1.Listar las hojas clínicas de un Paciente

8.2.Registrar Consulta

8.3.Registrar Examen Físico

8.4.Registrar Operación Quirúrgica

8.5.Registrar Orden Tratamiento

8.6.Registrar Orden de Estudio Imagenológico

8.7.Mostrar Imágenes Médicas de XWeb

8.8.Mostrar Reportes Médicos de XWeb

## **9. Gestionar Ubicación.**

9.1.Asignar Cama a Paciente

9.2.Asignar Cama Temporal a Paciente

9.3.Reasignar Cama

9.4.Reasignar Cama Temporal

9.5.Liberar Cama Temporal

## **2.10 Requerimientos no funcionales del sistema**

- Usabilidad.

El sistema debe ser lo más atractivo posible y que facilite el trabajo a los especialistas de las clínicas imagenológicas. Su acceso debe ser fácil y rápido, los usuarios no deberán poseer grandes conocimientos de informática para poder operar con este.

- Rendimiento.

El sistema debe ser capaz de responder varias transacciones por segundo, con el objetivo de dar una respuesta rápida al usuario y evitar demoras innecesarias en el servicio médico brindado por las clínicas imagenológicas.

- Portabilidad.

El sistema debe ser capaz de ejecutarse sobre la plataforma libre.

- Seguridad.

El acceso a la información estará restringido según los roles de los usuarios.

- Trabajador social
- Secretaria
- Especialista
- Administrador

- Políticos - culturales.

El sistema actualmente solamente está disponible en el idioma español, pero en un futuro está pensado implementar las versiones correspondientes para otros idiomas. Los logotipos e imágenes usadas están en correspondencia con el tipo de institución donde será usado.

- Legales.

El sistema y toda la documentación generada con el mismo, pertenecen al Grupo de Procesamiento de Imágenes y Señales de la Universidad de la Ciencias Informáticas.

- Confiabilidad.

El sistema debe estar disponible todo el tiempo de forma tal que se pueda acceder las 24 horas del día. Debe ser preciso en la información que le suministra al usuario para evitar cualquier tipo de error.

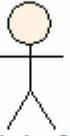
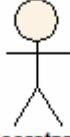
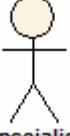
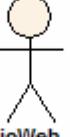
El sistema debe estar bien documentado, de esta forma se garantiza que el tiempo de mantenimiento sea mínimo.

- Interfaz.

La aplicación propuesta poseerá una interfaz sencilla, amigable y cómoda para los usuarios a quien va dirigida. La interfaz deberá ser lo suficientemente intuitiva como para permitir que usuarios con conocimientos básicos, puedan operar el sistema sin mayores contratiempos.

## 2.11 Definición de los casos de uso

- **Definición de los actores**

Actores del sistema	Justificación
<p>Trabajador Social</p>  <p>Trabajador Social</p>	<p>Inicia los casos de uso, Buscar Paciente y Asignar Datos Socio económicos a paciente. Es el rol de menos privilegios en el sistema.</p>
<p>Secretaria</p>  <p>Secretaria</p>	<p>Inicia los casos de uso Admitir paciente, todos los del paquete Gestionar Citas y Gestionar Ubicación. Es una especialización del actor Trabajador Social porque puede en cualquier momento realizar las tareas que este realiza.</p>
<p>Especialista</p>  <p>Especialista</p>	<p>Inicia los casos de uso Ingresar y Egresar pacientes y todos los del Paquete Gestionar Historia Clínica. Es una especialización del actor Secretaria porque puede en cualquier momento realizar las actividades que ésta realiza. Es el rol con mayores privilegios.</p>
<p>ServicioWeb_XWeb</p>  <p>ServicioWeb_XWeb</p>	<p>Es un componente del sistema Cassandra PACS con el cual interactuará nuestro sistema, para obtener los reportes imagenológicos e imágenes médicas de los pacientes.</p>

- **Listado de casos de uso.**

CU-1	Buscar Paciente
<b>Actor</b>	Trabajador Social
<b>Descripción</b>	Buscar pacientes en el sistema mediante la especificación de determinados criterios de búsqueda.
<b>Referencia</b>	RF-1

CU-2	Admitir Paciente
<b>Actor</b>	Secretaria
<b>Descripción</b>	La Secretaria de un hospital puede realizar la admisión de un paciente, recogiendo para ello la información relacionada con los datos personales del paciente y opcionalmente, los de su representante legal y la información socioeconómica del núcleo familiar del paciente. El paciente solo se admite una vez.
<b>Referencia</b>	RF-1, RF-2, RF-4

CU-3	Ingresar Paciente
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> de un hospital puede realizar la admisión de un paciente, recogiendo para ello la información relacionada con los datos personales del paciente y opcionalmente, los de su representante legal y la información socioeconómica del núcleo familiar del paciente.
<b>Referencia</b>	RF-1, RF-3

CU-4	Insertar Datos Socio económicos del paciente
<b>Actor</b>	Trabajador Social
<b>Descripción</b>	El <b>Trabajador Social</b> registra los datos socioeconómicos de un paciente que ha sido admitido en el hospital.
<b>Referencia</b>	RF-1, RF-4

CU-5	Mostrar Ingresos
<b>Actor</b>	Secretaria
<b>Descripción</b>	La Secretaria puede en cualquier momento visualizar los ingresos del hospital. De estos ingresos se muestran detalles como la ubicación de los pacientes y desde esta misma vista se debe permitir gestionar la ubicación de los pacientes.
<b>Referencia</b>	RF-5, RF-9 completo

CU-6	Egresar Paciente
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> busca en el sistema al paciente que desea egresar, registra la información correspondiente al egreso y elige la opción <b>Egresar</b> . El sistema registra la información correspondiente al egreso del paciente y libera la(s) cama(s) asignada(s) a éste para que pueda ser utilizada por otro paciente.
<b>Referencia</b>	RF-1, RF-6

CU-7	Reservar Cita con Especialista
<b>Actor</b>	Secretaria, Especialista
<b>Descripción</b>	El <b>Especialista o la Secretaria</b> , pueden efectuar el proceso de reservación de una cita para un paciente. Este caso de uso tiene dos escenarios y la gestión está basada en la existencia de un calendario centralizado al que puede acceder cualquiera de los dos actores. En el primer escenario <b>la Secretaria</b> reserva una cita para un paciente con un especialista en un turno determinado de una fecha dada. En el segundo escenario, <b>el Especialista</b> puede reservar una cita, en su propio calendario para un paciente en una fecha determinada.
<b>Referencia</b>	RF-1, RF-7.1

CU-8	Reservar Cita para Equipo
<b>Actor</b>	Secretaria, Especialista
<b>Descripción</b>	El <b>Especialista o la Secretaria</b> , pueden efectuar el proceso de reservación de una cita para un paciente en un equipo médico. Este caso de uso tiene dos escenarios y la gestión está basada en la existencia de un calendario centralizado del equipo al que puede acceder cualquiera de los dos actores. En el primer escenario, <b>la Secretaria</b> reserva una cita para un equipo, de determinado servicio, en una fecha dada. En el segundo escenario, <b>el Especialista</b> puede reservar una cita, para un paciente en un equipo médico de un servicio especificado, en una fecha determinada.
<b>Referencia</b>	RF-1, RF-7.2

CU-9	Reprogramar Cita
<b>Actor</b>	Secretaria, Especialista
<b>Descripción</b>	La <b>Secretaria o el Especialista</b> , pueden efectuar el proceso de reprogramación de una cita. Este caso de uso tiene dos escenarios y la gestión está basada en la existencia de un calendario por escenario al que puede acceder cualquiera de los dos actores. En el primer escenario, <b>la Secretaria</b> reprograma una cita para un equipo, de determinado servicio, en una fecha dada. En el segundo escenario, <b>el Especialista</b> reprograma una cita, para un paciente en un equipo médico de un servicio especificado, en una fecha determinada.
<b>Referencia</b>	RF-1, RF-7.1.1, RF-7.2, RF-7.3

CU-10	Cancelar Cita
<b>Actor</b>	Secretaria, Especialista
<b>Descripción</b>	La <b>Secretaria o el Especialista</b> , pueden efectuar el proceso de cancelación de una cita para un paciente en un equipo médico. Este caso de uso tiene dos escenarios y la gestión está basada en la existencia de un calendario centralizado del equipo al que puede acceder cualquiera de los dos actores. En el primer escenario, <b>la Secretaria</b> cancela una cita en el calendario de un equipo o de un especialista. En el segundo escenario, <b>el Especialista</b> cancela una cita, en el calendario de un equipo médico o de un especialista.
<b>Referencia</b>	RF-1, RF-7.1, RF-7.2

CU-11	Imprimir Cita
<b>Actor</b>	Secretaria
<b>Descripción</b>	La <b>Secretaria</b> imprime una cita, previamente concertada para un paciente con un especialista o equipo determinados.
<b>Referencia</b>	RF-1, RF-7.1, RF-7.2, RF-7.5
CU-12	Verificar Cita
<b>Actor</b>	Secretaria
<b>Descripción</b>	La <b>Secretaria</b> verifica una cita, previamente concertada para un paciente con un especialista o equipo determinados.
<b>Referencia</b>	RF-1, RF-7.1, RF-7.2, RF-7.6
CU-13	Pasar paciente citado a consulta
<b>Actor</b>	Secretaria
<b>Descripción</b>	La <b>Secretaria</b> indica al paciente que su cita esta en turno marca la cita como en progreso en el sistema.
<b>Referencia</b>	RF-1, RF-7.1, RF-7.2, RF-7.7
CU-14	Listar pacientes citados
<b>Actor</b>	Secretaria
<b>Descripción</b>	La <b>Secretaria</b> puede en cualquier momento listar los pacientes que tienen cita reservada para un día determinado, pudiendo quizás imprimir las citas para esa fecha.
<b>Referencia</b>	RF-1, RF-7.1, RF-7.2, RF-7.5, RF-7.8
CU-15	Listar hojas clínicas de un paciente
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> puede en cualquier momento listar las hojas clínicas de un paciente, ya sea para visualizar los reportes de estudios, historial de consulta, etc. del paciente en cuestión.
<b>Referencia</b>	RF-1, RF-8.1
CU-16	Registrar consulta
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> puede registrar una consulta realizada a un paciente. Entre los tipos de consulta que pudiera registrar están, consultas externas, de urgencia y de oftalmología.
<b>Referencia</b>	RF-1, RF-8.2
CU-17	Registrar examen físico
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> registra un examen físico realizado a un paciente.
<b>Referencia</b>	RF-1, RF-8.3
CU-18	Registrar operación quirúrgica
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> registra una operación quirúrgica.
<b>Referencia</b>	RF-1, RF-8.4

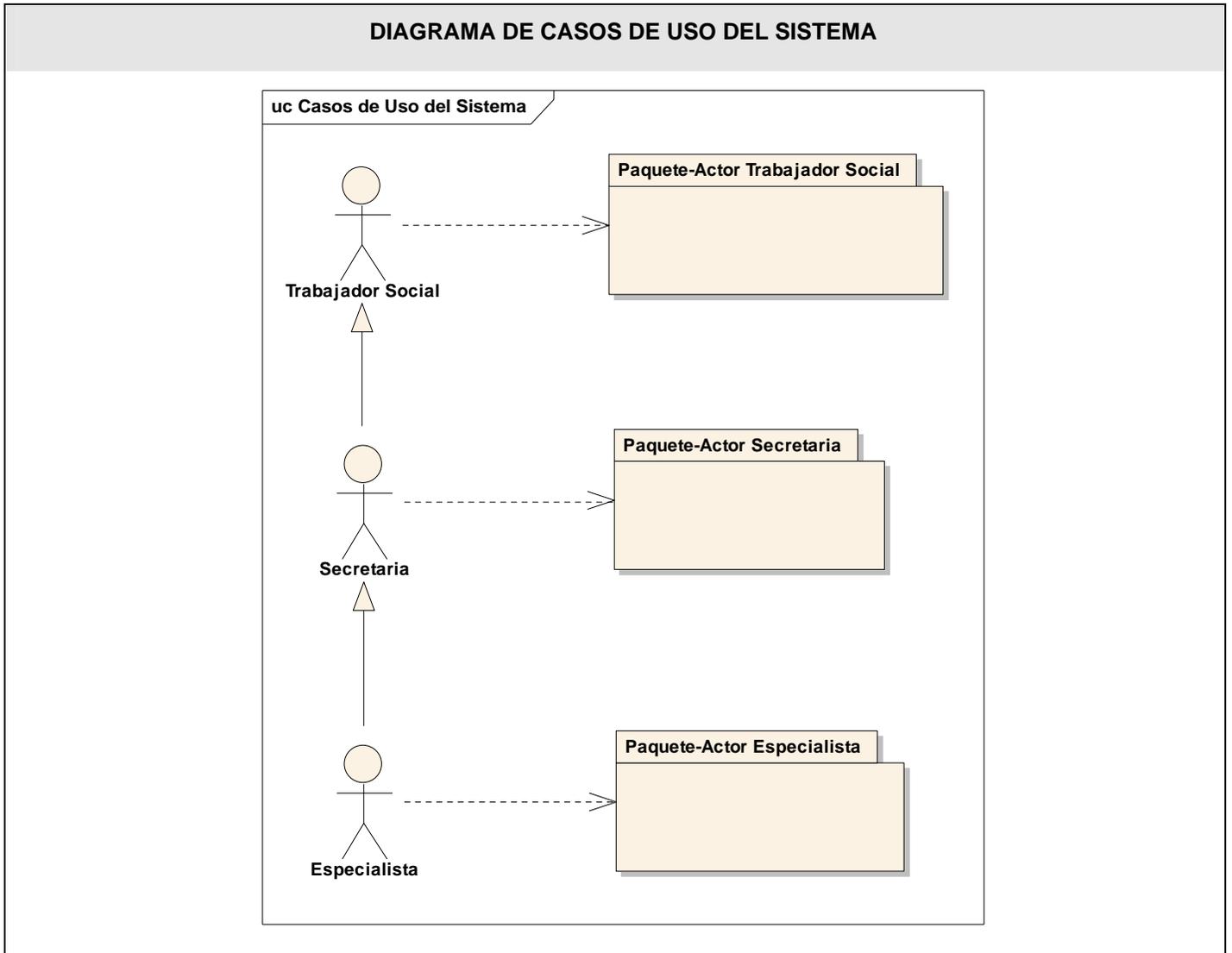
CU-19	Registrar orden de tratamiento
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> registra una orden de tratamiento en el sistema.
<b>Referencia</b>	RF-1, RF-8.5
CU-20	Registrar orden de estudio imagenológico
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> registra una orden de estudio imagenológico en el sistema.
<b>Referencia</b>	RF-1, RF-8.6
CU-21	Mostrar imágenes médicas de XWeb.
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> visualiza imágenes médicas pertenecientes a los estudios de un paciente y luego opcionalmente puede decidir si diagnostica la imagen usando el software de visualización y diagnóstico del PACS instalado en la institución.
<b>Referencia</b>	RF-1, RF-8.7
CU-22	Mostrar reportes médicos de XWeb.
<b>Actor</b>	Especialista
<b>Descripción</b>	El <b>Especialista</b> visualiza reportes médicos pertenecientes a los diagnósticos emitidos por otros especialistas sobre determinados estudios de un paciente.
<b>Referencia</b>	RF-1, RF-8.8
CU-23	Asignar cama a paciente.
<b>Actor</b>	Secretaria
<b>Descripción</b>	Desde la vista de ingresos, la <b>Secretaria</b> selecciona el paciente ingresado al que desea asignarle cama y selecciona la opción asignar de los detalles del ingreso del paciente. Luego selecciona un ala (norte, sur, este, oeste), un nivel del hospital relacionado con la especialidad en la que el paciente será atendido, un local de ingreso y una cama de éste. Por último selecciona la <b>Asignar cama</b> quedando el paciente en esa ubicación.
<b>Referencia</b>	RF-5, RF-9.1
CU-24	Asignar cama temporal a paciente.
<b>Actor</b>	Secretaria
<b>Descripción</b>	Desde la vista de ingresos, la <b>Secretaria</b> selecciona el paciente ingresado al que desea asignarle cama temporal y selecciona la opción <b>asignar cama temporal</b> de los detalles del ingreso del paciente. Luego selecciona un ala (norte, sur, este, oeste), un nivel del hospital relacionado con la especialidad en la que el paciente será atendido, un local de ingreso y una cama de éste. Por último selecciona la <b>Asignar cama temporal</b> quedando el paciente en esa ubicación.
<b>Referencia</b>	RF-5, RF-9.2

CU-25	Reasignar cama.
<b>Actor</b>	Secretaria
<b>Descripción</b>	Desde la vista de ingresos, la <b>Secretaria</b> ubica a los pacientes ingresados que tienen asignada una cama permanente que ocupará durante el tiempo de ingreso. Para esto debe asignar al paciente a un nivel del hospital relacionado con la especialidad en la que el paciente será atendido, dentro de este a una sala que está ubicada en un ala específica del hospital, dentro de la sala en una habitación y cama específicas.
<b>Referencia</b>	RF-5, RF-9.3

CU-26	Reasignar cama temporal.
<b>Actor</b>	Secretaria
<b>Descripción</b>	Desde la vista de ingresos, la <b>Secretaria</b> asigna a los pacientes ingresados, que tienen asignada, una cama temporal, que ocupará durante un plazo corto de tiempo. Para esto debe asignar al paciente a un nivel del hospital relacionado con la especialidad en la que el paciente será atendido, dentro de este a una sala, que está ubicada en un ala específica dentro de la institución.
<b>Referencia</b>	RF-5, RF-9.4

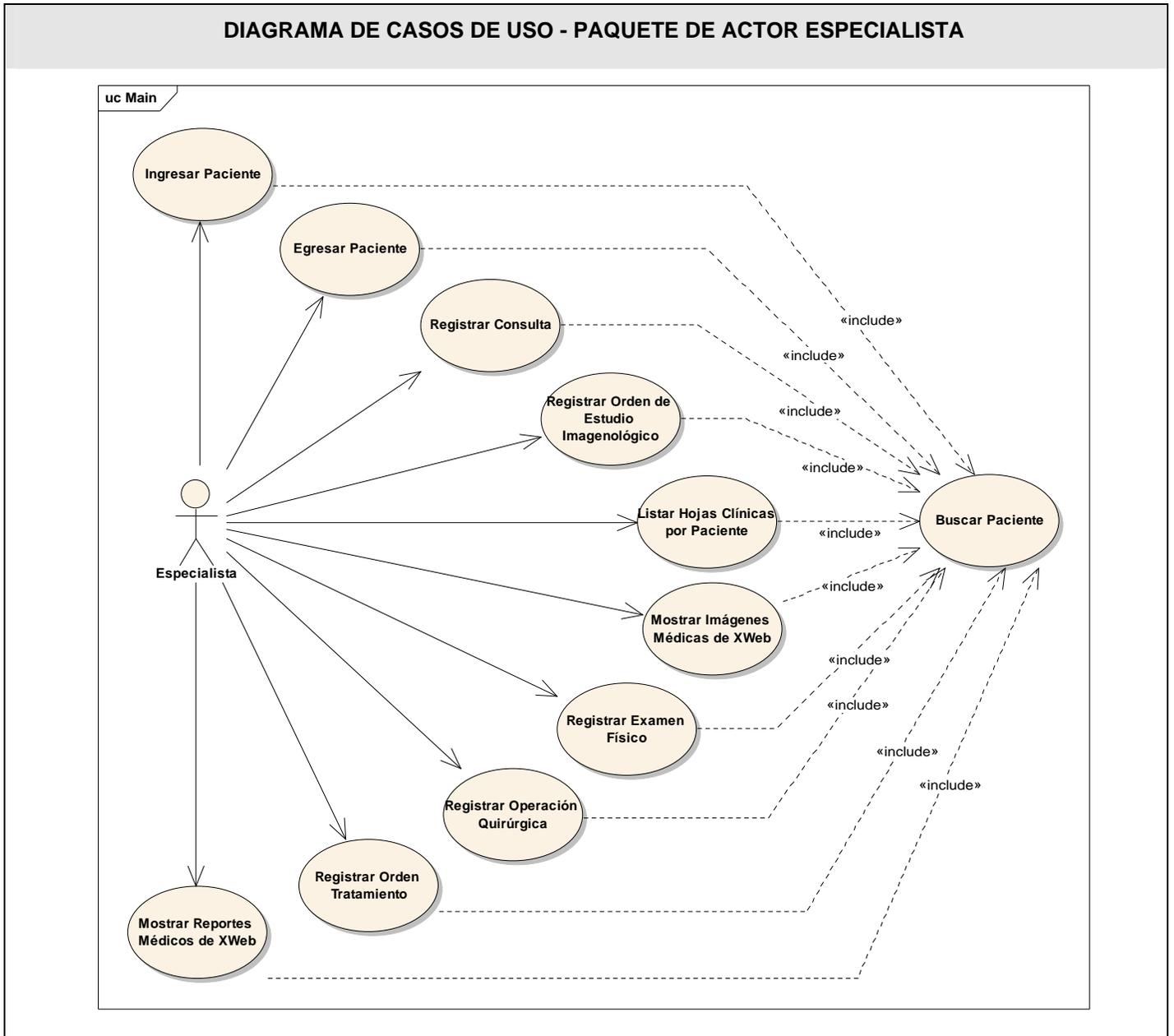
CU-27	Liberar cama temporal.
<b>Actor</b>	Secretaria
<b>Descripción</b>	Desde la vista de ingresos, la <b>Secretaria</b> selecciona el paciente ingresado al que desea liberarle la cama temporal y selecciona la opción <b>liberar cama temporal</b> de los detalles del ingreso del paciente. Luego selecciona <b>liberar cama temporal</b> quedando la cama temporal del paciente libre a partir de ese momento.
<b>Referencia</b>	RF-5, RF-9.5

## 2.12 Diagrama de casos de uso del sistema

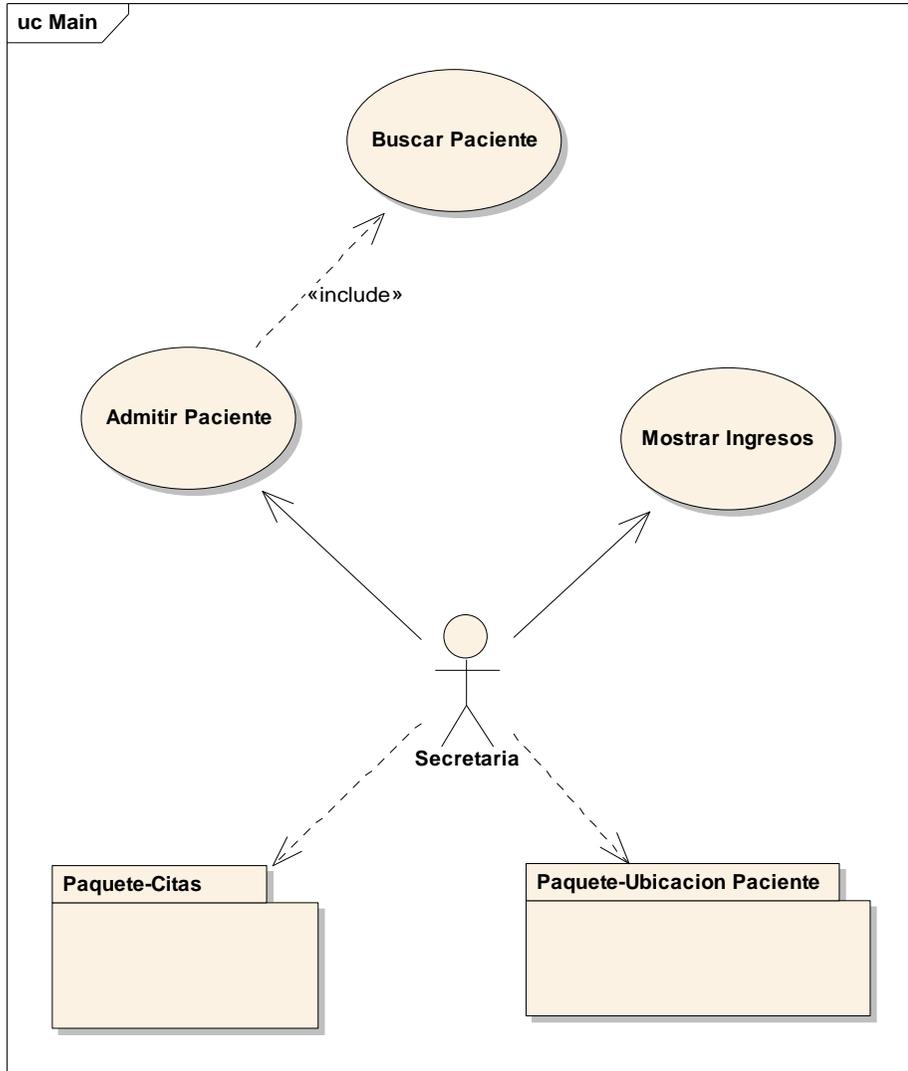


- Diagrama de casos de uso por paquetes.

Se han agrupado los casos de uso en paquetes por actores.



## DIAGRAMA DE CASOS DE USO - PAQUETE DE ACTOR SECRETARIA



Por razones de claridad hemos dividido el paquete del actor especialista en dos sub - paquetes: Paquete – Citas y Paquete – Ubicación Paciente. A continuación mostramos los diagramas de casos de uso de ambos paquetes.

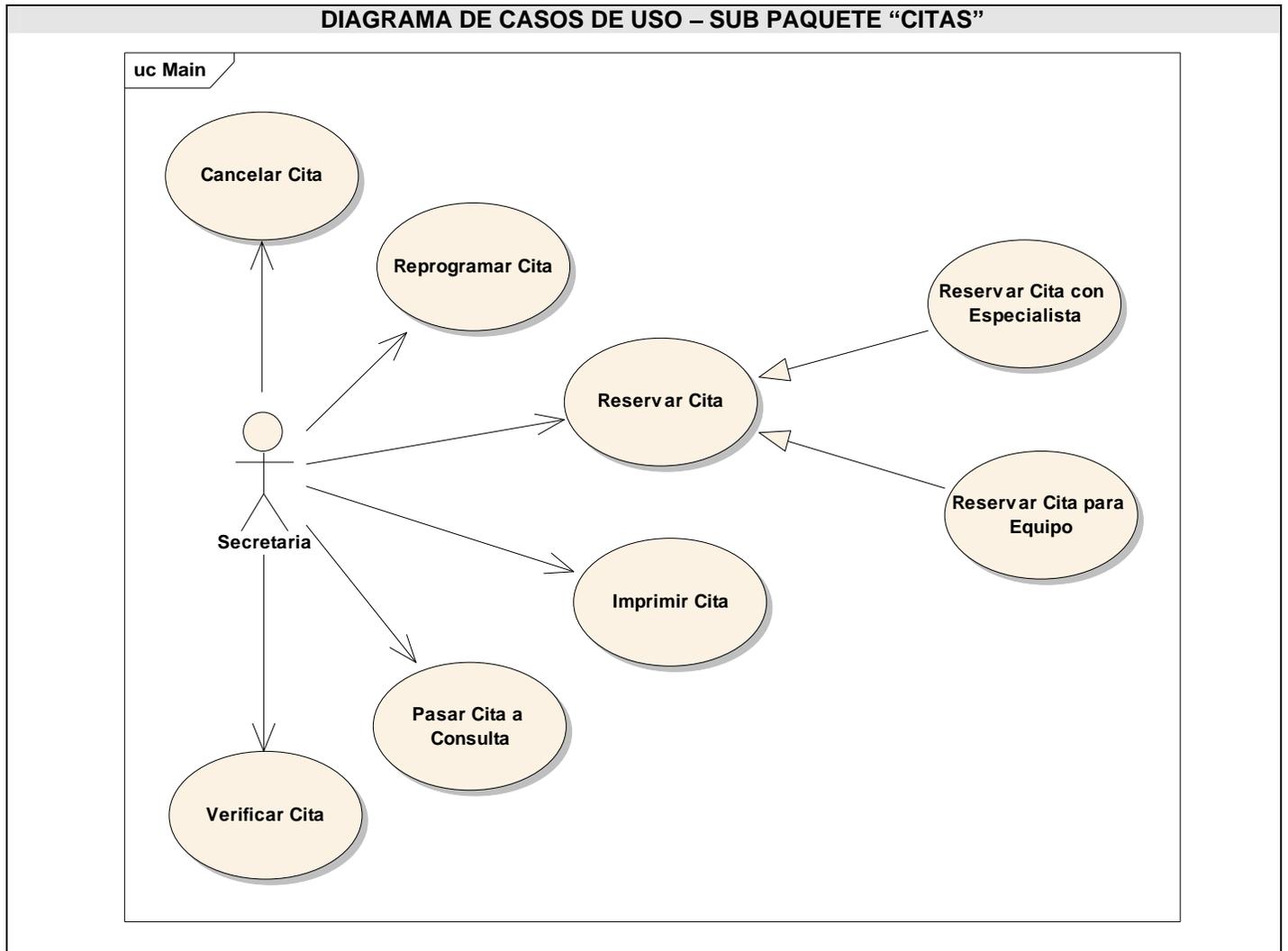


DIAGRAMA DE CASOS DE USO – SUB PAQUETE “UBICACION PACIENTE”

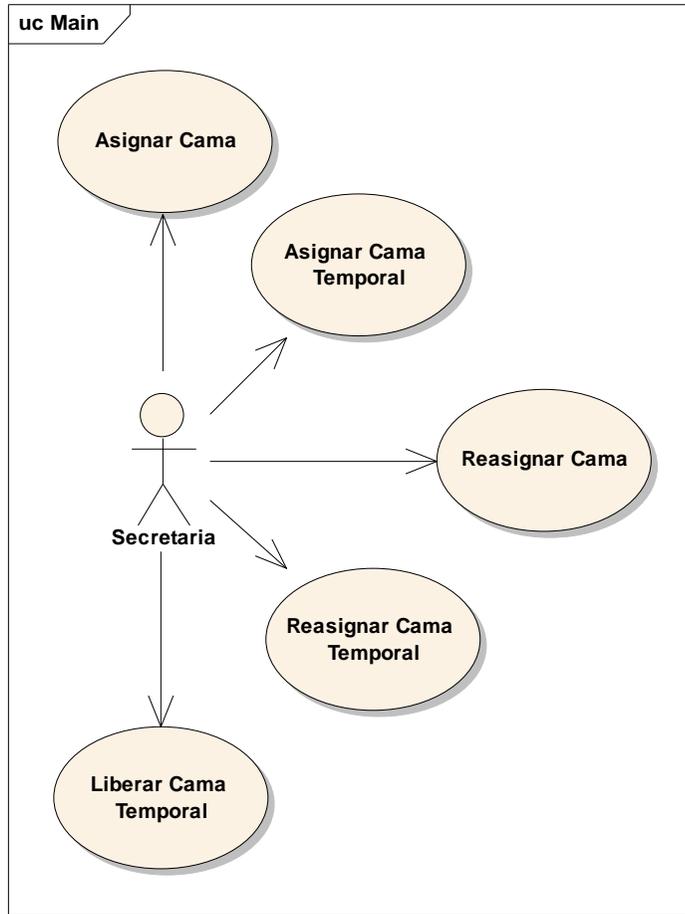
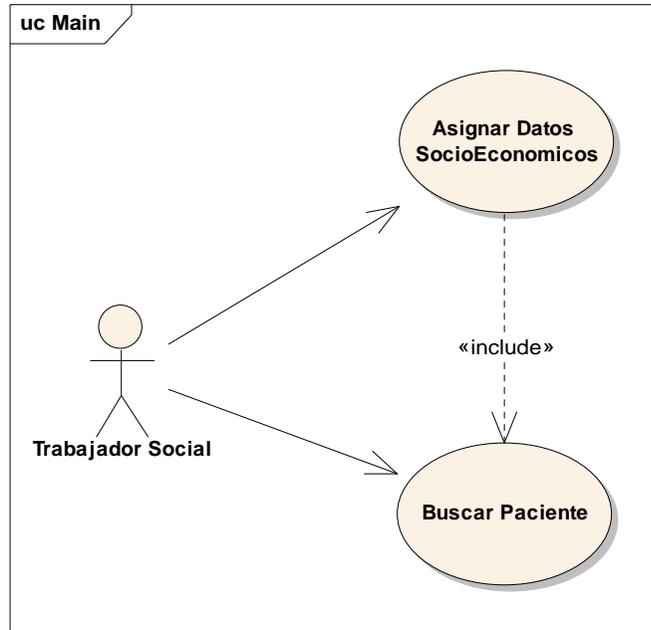


DIAGRAMA DE CASOS DE USO – PAQUETE ACTOR “TRABAJADOR SOCIAL”



- **Casos de uso por ciclos de desarrollo**

Primer ciclo de desarrollo

Cód.	Nombre de caso de uso	Paquete	Justificación
CU-1	Buscar Paciente.	Trabajador Social	Estos son los CU que son primarios, sin ellos el sistema dejaría de brindar un servicio fundamental. Estos CU son la base de la realización de los que se desarrollan en los ciclos posteriores.
CU-2	Admitir Paciente.	Secretaria	
CU-3	Ingresar Paciente.	Especialista	
CU-6	Egresar Paciente.	Especialista	
CU-7	Reservar Cita con Especialista.	Secretaria	
CU-8	Reservar Cita para Equipo.	Secretaria	
CU-20	Registrar Orden de Estudio Imagenológico.	Especialista	
CU-21	Mostrar Imágenes Médicas.	Especialista	
CU-22	Mostrar Reportes Médicos.	Especialista	
CU-23	Asignar Cama.	Secretaria	

Segundo ciclo de desarrollo

Cód.	Nombre de caso de uso	Paquete	Justificación
CU-4	Asignar Datos Socio Económicos	Trabajador Social	Estos CU son de menos importancia que los del 1er ciclo, en estos se ve la continuidad de los primeros. No son tan importantes, pero sin ellos el sistema dejaría de brindar un servicio muy significativo en la gestión de la información en las clínicas imagenológicas.
CU-5	Mostrar Ingresos	Secretaria	
CU-10	Cancelar Cita	Secretaria	
CU-12	Verificar Cita	Secretaria	
CU-13	Pasar paciente citado a Consulta	Secretaria	
CU-14	Listar pacientes citados	Secretaria	
CU-15	Listar Hojas Clínicas por paciente	Especialista	
CU-16	Registrar Consulta	Especialista	
CU-18	Registrar operación quirúrgica	Especialista	
CU-24	Asignar Cama Temporal	Secretaria	
CU-25	Reasignar Cama	Secretaria	

Tercer ciclo de desarrollo

Cód.	Nombre de caso de uso	Paquete	Justificación
CU-9	Reprogramar Cita	Secretaria	Estos CU son los menos importantes, por esta razón se dejó su desarrollo para el último ciclo. Su implementación, le daría al sistema el valor que se espera del mismo, pero aportando un número menos significativo que los CU perteneciente a los ciclos anteriores.
CU-11	Imprimir Cita	Secretaria	
CU-17	Registrar Examen Físico	Especialista	
CU-20	Registrar Orden de Tratamiento	Especialista	
CU-26	Reasignar Cama Temporal	Secretaria	
CU-27	Liberar Cama Temporal	Secretaria	

- **Casos de uso expandidos**

En el [Anexo 1](#) se muestra la descripción textual de los casos de uso que se escogieron fueran desarrollados en el primer ciclo. En esta descripción se especifica textualmente el flujo de los eventos a seguir entre el actor y la respuesta recibida por parte del sistema.

En el presente capítulo se han analizado los procesos de negocio que serán objeto de automatización. Se describe el sistema propuesto, exponiendo el modelo del negocio del mismo, los requerimientos funcionales y no funcionales, mostrándose además los diagramas de casos de uso que se han modelado.

## CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En el presente capítulo se exponen los diagramas de clases de análisis que participan en la realización de los casos de usos del primer ciclo de desarrollo. Además se representa los distintos diagramas de clases referentes al diseño y los diagramas de secuencia de los casos de usos. Se da una breve descripción de las clases Entidades y Controladoras usadas en el flujo de trabajo “Análisis y diseño” y de las tablas de la Base de Datos, de la que se muestra a su vez los diagramas Entidad-Relación agrupados por paquetes de responsabilidad para su mayor entendimiento.

### **Análisis**

Durante esta etapa, se analizan los requisitos funcionales que se describen en el capítulo anterior, refinándolos y estructurándolos con el objetivo de conseguir una comprensión más precisa y una descripción más detallada que sea fácil de mantener y que ayude a estructurar el sistema.

El objetivo principal de este flujo son los diagramas de clases de análisis, los cuales muestran que clases participan en las realizaciones de los distintos casos de usos del primer ciclo de desarrollo planteados en capítulo anterior. Estos diagramas se muestran en el [Anexo 2](#).

### **Diseño**

La esencia de esta etapa es la elaboración de los diagramas de clases de diseño, que muestra las clases participantes en la realización en un caso de uso con todos sus atributos, ver [Anexo 3](#).

Otro objetivo de esta etapa es la elaboración de los diagramas de interacción que muestran gráficamente como lo objetos se comunican entre ellos con el objetivo de cumplir los requerimientos.

Esta interacción se puede expresar en diagramas de colaboración y en diagramas de secuencia. Estos últimos, que son los que los autores han usado en el presente trabajo, detallan las secuencias de interacciones ordenadas en el tiempo y pueden ser encontrados en el [Anexo 4](#).

### **3.1 Descripción de las clases.**

La descripción de las clases da detalles de las clases entidades y controladoras que están involucradas en la realización de los casos de uso del primer ciclo de desarrollo y se encuentran en el [Anexo 5](#).

### **3.2 Diseño de la BD**

En este epígrafe se exponen los distintos diagramas Entidad Relación agrupados por paquetes de funcionalidad. [Anexo 6](#).

### **3.3 Descripción de las tablas.**

En este epígrafe se describen los atributos que tendrá cada tabla de la Base de Datos para el almacenamiento de la información requerida en la realización de los caso de uso, esta descripción se encuentra en el [Anexo 7](#).

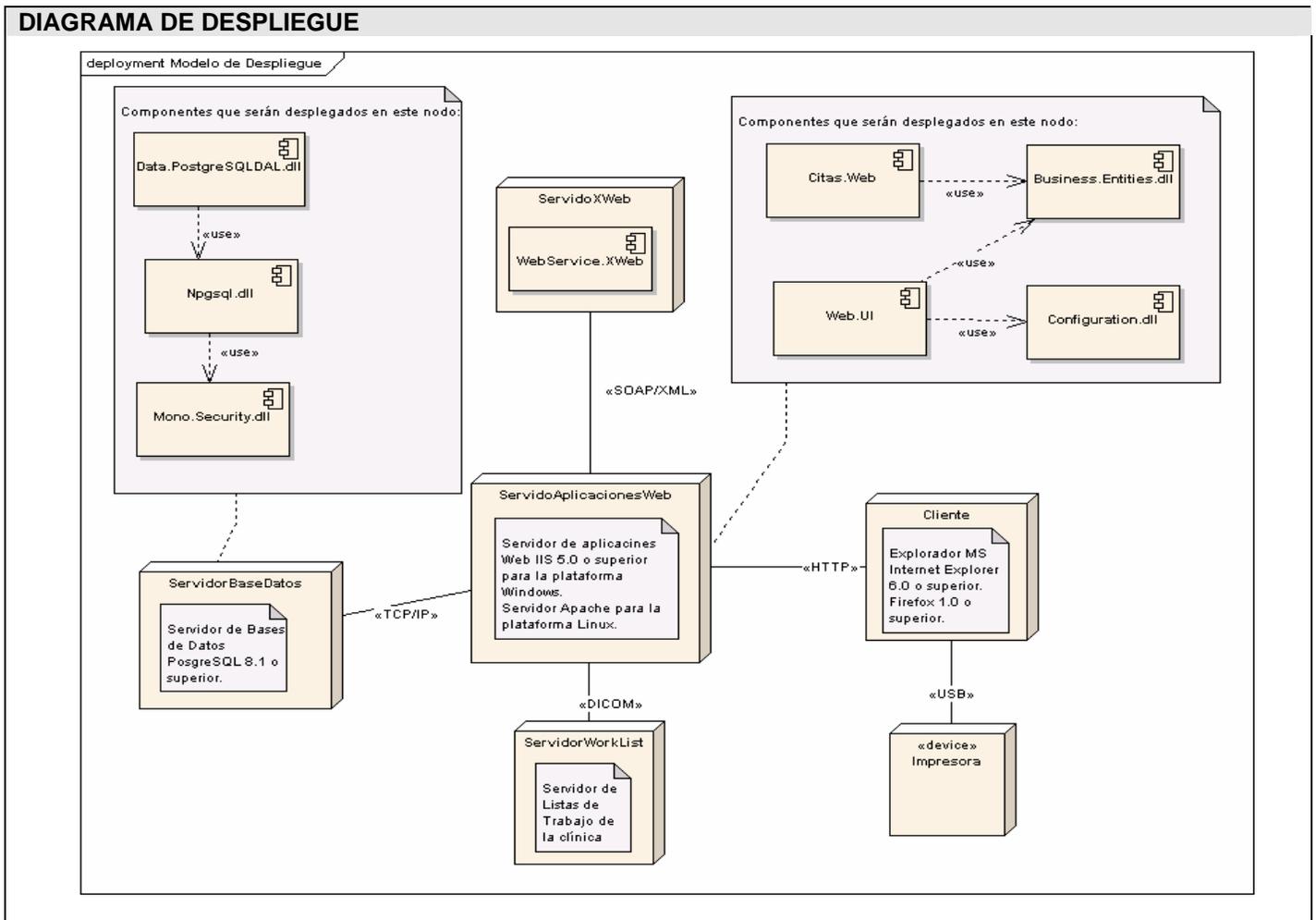
En este capítulo se ha mostrado los diagramas de clases de análisis y diseño, así como los diagramas de secuencia de cada realización de los casos de uso. Se ha descrito las clases entidades y las controladoras, y las tablas de la Base de Datos, Y por último se expusieron los diagramas Entidad – Relación.

## CAPÍTULO 4: IMPLEMENTACIÓN

En este capítulo se expone el diagrama de despliegue de la propuesta, en el cual se expone la ubicación física de cada componente, los cuales se exponen mas adelante en el diagrama de componentes. También se exponen los resultados del estudio de compatibilidad realizado con la herramienta MOMA, con el fin de una migración a la plataforma UNIX.

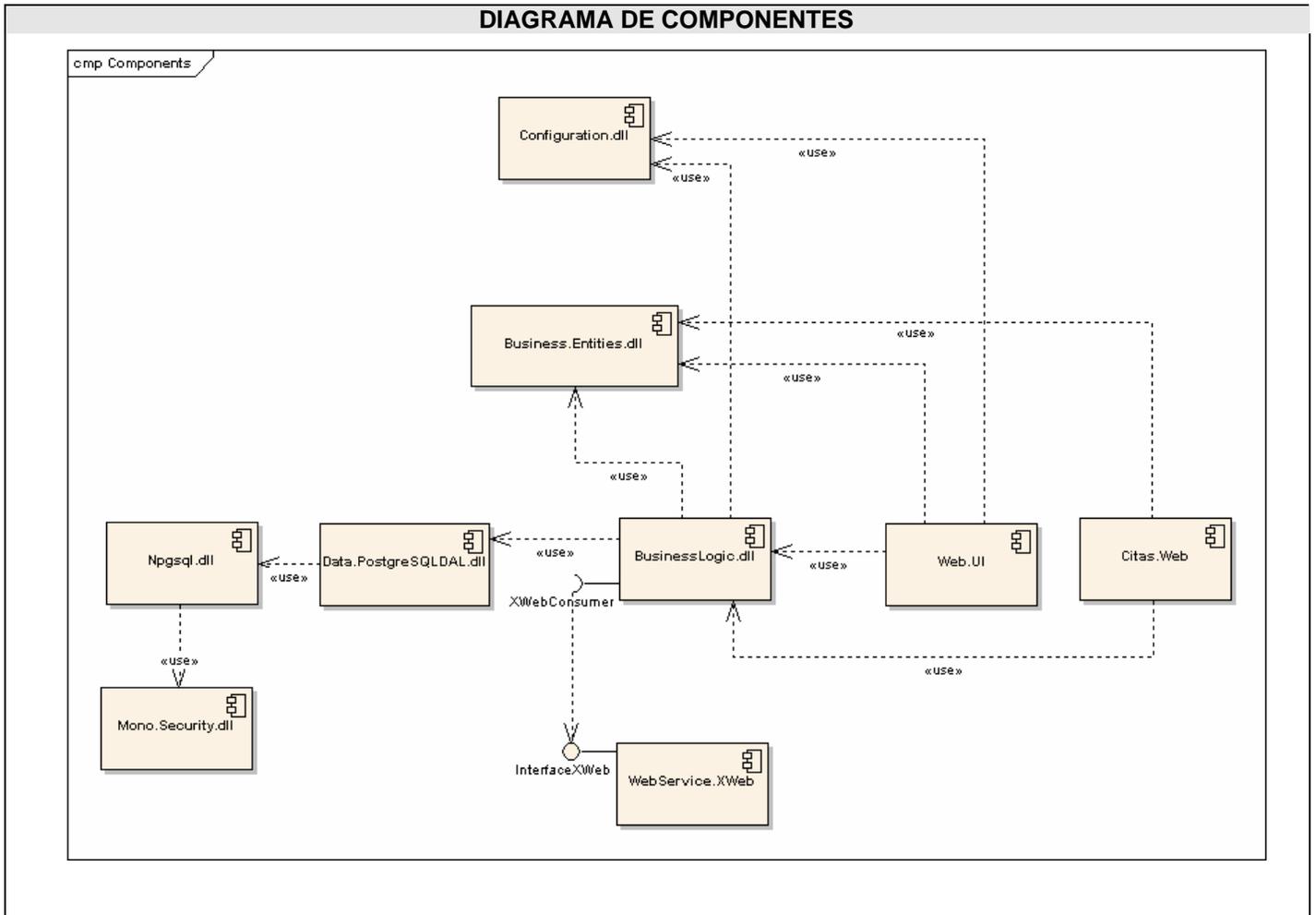
### 4.1 Diagrama de despliegue.

EL Modelo de Despliegue describe cómo y dónde el sistema será puesto en funcionamiento. Las estaciones de trabajo, dispositivos y procesadores son reflejados como nodos y su estructura interna puede ser representada adicionando otros nodos o artefactos.



## 4.2 Diagrama de componentes.

El Diagrama de Componentes define cómo las clases, artefactos y otros elementos de bajo nivel, se unen para formar componentes de alto nivel y las conexiones entre ellos. Los componentes son artefactos de software compilados que trabajan acoplados para brindar el comportamiento requerido dentro de las restricciones definidas en el proceso de captura de requisitos.



### 4.3 Estudio de compatibilidad con la plataforma libre.

Los estudios realizados con la herramienta MoMA (Mono Migration Analyzer), arrojan los siguientes resultados:



## Mono Migration Analyzer

**Analysis Summary:**

-  Methods that are still missing in Mono: 1
-  No P/Invokes are called.
-  Methods called that throw NotImplementedException: 2
-  Methods called marked with [MonoTodo]: 12

Como se puede apreciar, solamente existen 3 métodos en los componentes del sistema, que no se encuentran incluidos aún en la especificación de la versión 1.2.3 de Mono. A continuación se detallan los resultados obtenidos.

Para los siguientes ensamblados no se ha obtenido incompatibilidades para su migración:

Nombre del ensamblado	Descripción
CassandraClinic2005.Business.Entities.dll	En este ensamblado se definen las entidades del negocio.
CassandraClinic2005.BusinessLogic.dll	En este ensamblado se definen las clases controladoras.
CassandraClinic2005.Data.PostgreSQLDAL.dll	En este ensamblado se definen las clases de acceso a datos.

Para el ensamblado encargado de la gestión de la configuración, se encontró un método que está marcado con el atributo [**MonoTodo**] para la versión actual. Este atributo se aplica a las operaciones que no están completamente implementadas o que los programadores quieren dejar marcadas como que necesitan realizar alguna limpieza en ellas. Por tanto, los métodos marcados con este atributo, aunque no evitan que los sistemas compilen hacia UNIX, sí pueden provocar que la funcionalidad esperada no sea correctamente cumplimentada.

Los datos de la operación se muestran en la siguiente tabla:

<b>Ensamblado: CassandraClinic2005.Configuration.dll</b>	
Clase: CassandraClinic2005.Configuration.GestorConfiguracion	
<b>Nombre del método no implementado</b>	<b>Observaciones</b>
ConnectionStringSettingsCollection ConfigurationManager.get_ConnectionStrings()	No implementado

En este caso, el problema es que, hasta la versión actual, no se ha implementado la operación que incorpora .NET Framework en su versión 2.0 en la clase ConfigurationManager, dentro del ensamblado System.Configuration.dll, para leer las cadenas de conexión a bases de datos que residen en el fichero de configuración de la aplicación. Esto significa que para que el sistema propuesto, pueda ejecutarse completamente en la plataforma libre, debe sustituirse ese método por uno existente o de lo contrario esperar hasta que el método sea implementado por los desarrolladores de MONO o algún colaborador.

El problema no es, ni mucho menos, algo intratable. De hecho, se puede lograr la sustitución del método ausente con el uso de la clase que provee el propio .NET Framework en sus primeras versiones para el manejo de los ficheros de configuración. De esta manera se logra la compatibilidad de este ensamblado.

Los restantes problemas de compatibilidad son originados por el uso de la interfaz de programación de aplicación (API, por sus siglas en inglés) que se usa para acceder a PostgreSQL: Npgsql.dll y su módulo complementario de seguridad: Mono.Security.dll. Pero esto, tampoco representa un problema ya que la razón de la incompatibilidad es porque el API que se ha probado es la versión para Microsoft Windows. En caso de realizar una migración del sistema hacia UNIX se usaría la versión para esa plataforma.

## CONCLUSIONES

Históricamente, entre las grandes empresas dedicadas al desarrollo de paquetes de software orientados a la gestión de la información clínica, se ha apreciado una tendencia a separar en dos grandes grupos, según la entidad primaria sobre la que recae el peso fundamental de la gestión, a los sistemas. En unos, el enfoque es hacia la Historia Clínica y en otros, el centro de esta gestión lo constituye el paciente.

Durante el desarrollo de la presente investigación, los autores han arribado a las siguientes conclusiones:

El segundo enfoque, orientado al paciente, reporta mejores resultados, por cuanto, es realmente éste, el beneficiado por la correcta realización de los flujos de negocio de instituciones del área de la salud. Se considera que la centralización en otras entidades, alejan del verdadero objetivo de gestión a los sistemas, haciéndolos mas complejos y menos adaptables a las necesidades crecientes del campo de estudio.

Con el desarrollo y la implementación del presente trabajo se logra optimizar la gestión de los servicios brindados en las instituciones hospitalarias, de acuerdo con los estándares internacionales; así como la vinculación de la información imagenológica, perteneciente a los pacientes, con sus datos clínicos, demográficos y/o socio – económicos.

Mediante el uso de herramientas de compatibilidad, se logra demostrar, la posibilidad de un despliegue del sistema en la plataforma libre. El desarrollo creciente del Proyecto Mono, y la casi total implementación de la alternativa UNIX al Microsoft .NET Framework 2.0, propuesta por el mencionado proyecto, hace factible el hecho de realizar un despliegue exitoso del producto en esta plataforma.

La aplicación de la solución informática desarrollada, brinda a las clínicas o centros en los que se instale, posibilidades inherentes a la informatización, por ejemplo servicios de intranet hospitalaria, publicación de avisos, noticias, foros de discusión de temas de interés a la comunidad médica, entre otros.

## RECOMENDACIONES

Sobre la presente investigación los autores recomiendan:

- Someter el sistema a pruebas de calidad de software para su posterior instalación en instituciones que lo requieran.
- El estudio de las posibilidades de integración con los subsistemas desarrollados por otras empresas o grupos de desarrollo de la UCI, involucrados en el proceso de informatización del sistema cubano de salud pública.
- Desarrollar Hojas de Historia Clínica Especializada, que puedan ser adjuntadas a la solución aquí expuesta, de forma que se gestione no solamente la información imagenológica, sino que se pueda expandir hacia la gestión de la información en otras áreas.

## REFERENCIA BIBLIOGRÁFICA

1. MORA, J. *HL7*, 2005. [Disponible en: <http://es.wikipedia.org/wiki/Usuario:Jlmora>]
2. VEGAS, J. *Desarrollo de aplicaciones Web*, 2006. [Disponible en: <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node17/>]
3. WIKIPEDIA. *Perl*, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Perl#Descripci.C3.B3n>]
4. WIKIPEDIA. *PHP*, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Php>]
5. WIKIPEDIA. *CGI*, 2007. [Disponible en: [http://es.wikipedia.org/wiki/Common\\_Gateway\\_Interface](http://es.wikipedia.org/wiki/Common_Gateway_Interface)]
6. WIKIPEDIA. *Active Server Pages*, 2007. [Disponible en: [http://es.wikipedia.org/wiki/Active\\_Server\\_Pages](http://es.wikipedia.org/wiki/Active_Server_Pages)]
7. WIKIPEDIA. *ASP.NET*, 2007. [Disponible en: <http://es.wikipedia.org/wiki/ASP.NET>]
8. WIKIPEDIA. *SGBD*, 2007. [Disponible en: <http://es.wikipedia.org/wiki/SGBD>]
9. MSDN Library. *Ejemplos de código de C# para desarrolladores de Java*, 2007. [Disponible en: [http://msdn2.microsoft.com/es-es/library/ms228363\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/ms228363(VS.80).aspx)]
10. WIKIPEDIA. *CSS*, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Css>]
11. WIKIPEDIA. *UML*, 2007. [Disponible en: <http://es.wikipedia.org/wiki/Uml>]

## BIBLIOGRAFÍA

Anónimo. *JavaHispano*, 2005. [Disponible en: [www.javaHispano.org](http://www.javaHispano.org)]

ABIÁN, M. Á. *J2EE Y .NET: La rivalidad permanente*, 2002.

ACR-NEMA. *Estándar DICOM*. National Electrical Manufacturers Association, 2004. p.

ARMSTRONG, E.; J. BALL, *et al.* *The J2EE 1.4 Tutorial* 2005.

AUTORES, C. D. *Programación Java Server con J2EE Edición 1.3*. 2002. p.

CERAMI, E. *Web Services Essentials. Distributed Applications with XML-RPC, SOAP, UDDI & WSDL*.  
O'Reilly, 2002. 304 p. 0-596-00224-6

CORPORATION, M. *Cambio de aplicaciones a .NET*, 2002. [Disponible en:

[http://www.microsoft.com/spanish/msdn/articulos/archivo/130902/voices/dotnet\\_MovingJavaApps.a  
sp](http://www.microsoft.com/spanish/msdn/articulos/archivo/130902/voices/dotnet_MovingJavaApps.a<br/>sp)]

---. *La seguridad en los servicios Web XML*, 2002. [Disponible en:

<http://www.microsoft.com/spanish/msdn/articulos/archivo/030502/voices/xmlwssec.asp>]

DPI. *DPI Software*, 2007. [Disponible en: <http://gbsystems.com/dpi/>]

ECHARTE, P. *Introducción a la plataforma .NET y Mono*, 2005

FONSECA, R. A. *Programación de funciones en PL/pgSQL para PostgreSQL*, 2002.

FOWLER, M. *Patterns of Enterprise Application Architecture*. Boston, Addison-Wesley, 2002. 560 p.

GROUP, P. G. D. *Sitio oficial de PostgreSQL*, 2007 [Disponible en: <http://www.postgresql.org/>]

HL7INC. *What is HL7?*, 2007. [Disponible en: <http://www.hl7.org/>]

INFOMED. *Alianza Vital*, 2006. [Disponible en: <http://www.sld.cu/sitios/renacip/temas.php?idv=2026>]

JIMÉNEZ, L. E. *¿Una comparación entre Java y .NET?*, 2003. [Disponible en:

<http://dspace.icesi.edu.co/dspace/bitstream/item/391/1/ljimenez-javaypontonet.pdf>]

LARMAN, C. *UML y PATRONES. Introducción al análisis y diseño orientado a objetos*. México, Prentice  
Hall, 1999. 536 p.

MACDONALD, M. *ASP.NET Manual de Referencia*, 2002.

MCCALLUM, G. *Construir Aplicaciones EJB con JBoss, Lomboz y Eclipse*, 2004. [Disponible en: [http://www.programacion.com/java/tutorial/jap\\_aplic\\_jboss/3/](http://www.programacion.com/java/tutorial/jap_aplic_jboss/3/)]

POSTGRESQL, E. E. D. D. D. *Tutorial de PostgreSQL*, 2002.

RECIO, F. and D. PROVENCIO. *Manual sobre la plataforma .Net*, 2004. [Disponible en: <http://www.desarrolloweb.com/manuales/48/>]

SIEMENS, C. *Acerca de.. Siemens Chile*, 2007. [Disponible en: [http://www.siemens.cl/med/gestion\\_imagenes\\_medicas.htm](http://www.siemens.cl/med/gestion_imagenes_medicas.htm)]

TIMSA. *Perfil de la empresa*, 2007. [Disponible en: <http://www.timsa.com.mx/indeks.htm>]

WALTHER, S. *Nuevas características de seguridad en ASP.NET 2.0*, 2004.

WIKIPEDIA. *American College of Radiology*, 2007. [Disponible en: [http://en.wikipedia.org/wiki/American\\_College\\_of\\_Radiology](http://en.wikipedia.org/wiki/American_College_of_Radiology)]

---. *National Electrical Manufacturers Association*, 2007. [Disponible en: [http://en.wikipedia.org/wiki/National\\_Electrical\\_Manufacturers\\_Association](http://en.wikipedia.org/wiki/National_Electrical_Manufacturers_Association)]

WORSLEY, J. and J. DRAKE. *Practical PostgreSQL*. 2001. p.

ANEXOS

**Anexo 1 Casos de uso expandidos**

Caso de uso – Buscar Paciente.

<b>CU-1</b>	Buscar paciente.	
<b>Actores:</b>	<b>Trabajador Social</b> (inicia)	
<b>Propósito:</b>	Permitir realizar búsquedas de pacientes en el sistema mediante la especificación de determinados criterios.	
<b>Resumen:</b>	Permite a cualquier usuario del sistema buscar un paciente en el mismo. Para ello pueden especificarse ninguno, uno, varios y/o todos los criterios de búsqueda siguientes: Nombres, Apellidos, Género, CI, Nacionalidad y Fecha de nacimiento.	
<b>Referencias:</b>		
<b>Tipo:</b>	Primario y Real	
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1. El <b>Trabajador Social</b> selecciona la opción Buscar paciente.		1.1. Muestra una pantalla de búsqueda de pacientes. 1.2. Carga los nomencladores Sexo y Nacionalidad.
2. El <b>Trabajador social</b> introduce los datos de búsqueda deseados y presiona buscar.		2.1. Realiza la búsqueda de acuerdo a los criterios seleccionado. 2.2. Muestra el paciente o la lista de pacientes encontrados y un conjunto de opciones posibles a ejecutar sobre estos.
<b>Flujos Alternos</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
		2.2a- Si el sistema no encuentra ningún paciente del hospital que cumpla con los criterios especificados en la búsqueda enviará un mensaje de notificación y termina el caso de uso.

Caso de uso – Admitir paciente

<b>CU-2</b>	Admitir Paciente
<b>Actores:</b>	<b>Secretaria</b> (Inicia).
<b>Propósito:</b>	Permitir la admisión de un paciente en el hospital.
<b>Resumen:</b>	La Secretaria de un hospital puede realizar la admisión de un paciente, recogiendo para ello la información relacionada con los datos personales del paciente y opcionalmente, los de su representante legal y la información socioeconómica del núcleo familiar del paciente.
<b>Referencias:</b>	RF-1
<b>Precondiciones:</b>	El paciente que se desea admitir no debe haberse admitido previamente en el sistema. (Garantizado mediante la búsqueda)
<b>Flujo Alternativo</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1. La <b>Secretaria</b> selecciona la opción Admisión.	1.1- Muestra la pantalla de búsqueda de pacientes. 1.2- Carga los nomencladores de búsqueda: Nacionalidad, Sexo.
2. La <b>Secretaria</b> introducen los criterios de búsqueda deseados y ordenan buscar.	2.1- Busca en la base de datos los pacientes que están admitidos y cuyos datos coinciden con los criterios de búsqueda suministrados. (No debe haber coincidencias) 2.2- (Si no se encuentran pacientes) muestra en pantalla un mensaje indicando que se puede continuar el proceso de admisión porque no se han encontrado pacientes con los datos suministrados.
3. La <b>Secretaria</b> selecciona la opción registrar datos personales del paciente.	3.1- Muestra la página para registrar los datos Personales del paciente y rellena los campos que ya habían sido suministrados en la búsqueda con los valores que el usuario brindó. 3.2- Carga los nomencladores: Sexo, Nacionalidad, Lugar de Nacimiento, Entidad Federal, Municipio, Estudios Cursados, Instituciones referidoras, Religiones y Médicos referidores.
4. La <b>Secretaria</b> selecciona la opción <u>Datos de la madre, el padre o del representante legal del paciente.</u>	4.1- Muestra la página para registrar los datos correspondientes al representante legal del paciente. 4.2- Carga los nomencladores: Lugar de nacimiento, Sexo, Nacionalidad, Parentesco, Estudios Cursados, Estado civil y Ocupación.
5. Una vez que la <b>Secretaria</b> ha completado toda la información deben seleccionar la opción <b>Admitir (submit).</b>	5.1- Guarda la información necesaria para la admisión del paciente en el hospital.
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>

<p>2.a Una vez que la <b>Secretaria</b> ha ordenado buscar si existen pacientes cuyos datos coinciden con los criterios de búsqueda, podrán elegir entre un grupo de opciones para ejecutar sobre ellos.</p> <p>5.a Una vez que la <b>Secretaria</b> completan la información correspondiente al representante legal del paciente opcionalmente pueden registrar además la información socioeconómica del mismo, seleccionando la opción Datos Socioeconómicos del núcleo familiar.</p> <p>5.a.2. La <b>Secretaria</b> registra los datos socioeconómicos del paciente.</p>	<p>2.a.1 Muestra la lista de pacientes cuyos datos coinciden con los criterios proporcionados y un conjunto de opciones posibles para cada uno de ellos. (Ej. Ingresar, Citar).</p> <p>5.a.1 Muestra la página para registrar los datos Socioeconómicos del paciente.</p> <p>5.a.3 Guarda los datos socioeconómicos del paciente.</p>
---	---

Caso de uso – Ingresar paciente

<b>CU-3</b>	Ingresar Paciente
<b>Actores:</b>	<b>Especialista</b> (Inicia)
<b>Propósito:</b>	Ingresar un paciente en el Hospital.
<b>Resumen:</b>	El <b>Especialista</b> ingresa un paciente en el hospital cuando luego examinarlo considera prudente realizarlo. Durante la estadía del paciente en la institución éste deberá tener una cama de ingreso asignada y quizás una cama temporal.
<b>Referencias:</b>	RF-1, RF-3
<b>Precondiciones:</b>	El paciente debe haber sido previamente admitido en la institución. El paciente no debe estar actualmente ingresado en la institución.
<b>Acciones del Actor</b>	
<b>Respuesta del Sistema</b>	
1. El <b>Especialista</b> selecciona la opción Ingresar Paciente.	1.1. Muestra la página que permite buscar un paciente.
2. El <b>Especialista</b> introduce los criterios de búsqueda y presionar <b>buscar</b> .	2.1. Busca los pacientes que están admitidos, que no están ingresados y cuyos datos coinciden con los criterios de búsqueda. 2.2. Muestra los pacientes encontrados con la opción <b>Seleccionar</b> para cada uno.
3. El <b>Especialista</b> selecciona el paciente que desea ingresar mediante la opción <b>Seleccionar</b> que aparece al lado de cada paciente encontrado.	3.1. Muestra la página que permite ingresar un paciente. 3.2. Carga los nomencladores Vía de Ingreso, Diagnóstico de admisión, Diagnóstico Asociado, Diagnóstico Final y la lista de Médicos Tratantes.
4. El <b>Especialista</b> introduce la información relativa al ingreso y presiona <b>Ingresar</b>	4.1. Valida los datos. 4.2. Registra el ingreso. 4.3. Notifica a la Secretaria.
<b>Flujo Alterno</b>	
<b>Acciones del Actor</b>	
<b>Respuesta del Sistema</b>	
	2.1.a Si el sistema no encuentra pacientes en la búsqueda, notificará al especialista.

Caso de uso – Egresar paciente

<b>CU-6</b>	Egresar paciente	
<b>Actores:</b>	<b>Especialista</b> (inicia)	
<b>Propósito:</b>	Dar de alta un paciente que ha sido ingresado en el hospital.	
<b>Resumen:</b>	El <b>Especialista</b> busca en el sistema al paciente que desea egresar, registra la información correspondiente al egreso y elige la opción <b>Egresar</b> . El sistema registra la información correspondiente al egreso del paciente libera la(s) cama(s) asignada(s) a éste para que pueda ser utilizada por otro paciente.	
<b>Referencias:</b>	RF-1, RF-6	
<b>Precondiciones:</b>	El paciente que se va a egresar debe estar admitido e ingresado en la institución.	
<b>Poscondiciones:</b>	La cama de ingreso del paciente, así como la cama temporal asignadas al paciente deben ser liberadas para que puedan ser usadas por otro paciente.	
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
1- El <b>Especialista</b> solicita realizar un nuevo egreso seleccionando la opción Egresar Paciente.		1.1. Muestra la página que permite buscar pacientes.
2- El <b>Especialista</b> introduce los criterios de búsqueda del paciente que desea egresar.		2.1. Busca el paciente en la base de datos. 2.2. Muestra la pagina que permite egresar un paciente. 2.3. Carga el nomenclador: Motivo de egreso y la lista de médicos internos. (medico que egresa)
3- El <b>Especialista</b> registra la información correspondiente al egreso.		
4- El <b>Especialista</b> selecciona <b>Egresar</b> .		4.1. Registra el egreso del paciente. 4.2. Libera la cama de ingreso y la cama temporal (si está asignada) que se le había asignado al paciente durante el tiempo que estuvo ingresado.
<b>Flujo Alternativo</b>		
<b>Acciones del Actor</b>		<b>Respuesta del Sistema</b>
		2.1.a- Si el paciente Buscado no ha sido encontrado entre los pacientes ingresados del Hospital se notifica a la Secretaria y termina el caso de uso.

Caso de uso - Reservar Cita con Especialista

<b>CU-7</b>	Reservar cita con especialista.	
<b>Actores:</b>	<b>Secretaria</b> (inicia)	
<b>Propósito:</b>	Reservar una cita para un paciente con un especialista.	
<b>Resumen:</b>	<b>La Secretaria</b> reserva una cita para un paciente con un especialista en un turno determinado de una fecha dada.	
<b>Referencias:</b>	RF-1, RF-7.1	
<b>Precondiciones:</b>	El paciente debe estar admitido.	
<b>Escenario 1</b>		
	<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1- La <b>Secretaria</b> solicita reservar cita para paciente con un especialista.	1.1- Muestra la página para buscar pacientes.
	2- La <b>Secretaria</b> introduce los criterios de búsqueda y presiona <b>buscar</b> .	2.1-Muestra los pacientes encontrados
	3- La <b>Secretaria</b> selecciona el paciente que desea citar.	3.1 Muestra la página para introducir las indicaciones (información significativa relativa al proceso) de la cita.
	4- La <b>Secretaria</b> introduce las indicaciones y selecciona la opción <b>continuar</b> .	4.1- Muestra el calendario de gestión de citas.
	5- La <b>Secretaria</b> selecciona los datos de la cita (servicio, local, especialidad, especialista, Institución referidora)	5.1-Carga el calendario del especialista seleccionado.
	6-En el calendario mostrado presiona seleccionar en el horario deseado.	6.1-Reserva la cita y actualiza el calendario.
<b>Escenario 2</b>		
	<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1- La <b>Secretaria</b> solicita ver el calendario medico de los especialistas.	1.1-Muestra el calendario medico.
	2- La <b>Secretaria</b> selecciona los datos de la cita (servicio, local, especialidad, especialista, Institución referidora)	2.1-Carga el calendario del especialista seleccionado.
	3- En el calendario mostrado presiona seleccionar en el horario deseado.	3.1-Habilita la opción <b>Nueva</b> .
	1- Selecciona la opción <b>Nueva</b>	1.1- Muestra la página para buscar pacientes.
	2- Introduce los criterios de búsqueda y presiona <b>buscar</b> .	2.1-Muestra los pacientes encontrados
	3- La <b>Secretaria</b> selecciona el paciente que desea citar.	3.1 Muestra la página para introducir las indicaciones (información significativa relativa al proceso) de la cita.
	4- La <b>Secretaria</b> introduce las indicaciones y selecciona la opción <b>terminar</b> .	4.1- Registra la cita.

Caso de uso – Reservar cita para equipo

<b>CU-7</b>	Reservar cita para equipo médico.
<b>Actores:</b>	Secretaria (inicia)
<b>Propósito:</b>	Reservar una cita para un paciente en un equipo médico.
<b>Resumen:</b>	La Secretaria reserva una cita para un paciente para un equipo médico en un turno determinado de una fecha dada.
<b>Referencias:</b>	RF-1, RF-7.2
<b>Precondiciones:</b>	El paciente debe estar admitido.
<b>Escenario 1</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- La <b>Secretaria</b> solicita reservar cita para paciente con para un equipo médico.	1.1- Muestra la página para buscar pacientes.
2- La <b>Secretaria</b> introduce los criterios de búsqueda y presiona <b>buscar</b> .	2.1-Muestra los pacientes encontrados
3- La <b>Secretaria</b> selecciona el paciente que desea citar.	3.1 Muestra la página para introducir las indicaciones (información significativa relativa al proceso) de la cita.
4- La <b>Secretaria</b> introduce las indicaciones y selecciona la opción <b>continuar</b> .	4.1- Muestra el calendario de gestión de citas.
5- La <b>Secretaria</b> selecciona los datos de la cita (servicio, local, especialidad, institución referidora, modalidad y equipo médico)	5.1-Carga la lista de trabajo del equipo médico seleccionado.
6- En la lista de trabajo mostrada presiona seleccionar en el horario deseado.	6.2-Reserva la cita y actualiza la lista de trabajo del equipo médico.
<b>Escenario 2</b>	
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
1- La <b>Secretaria</b> solicita ver la lista de trabajos de los equipos médicos.	1.1-Muestra la pagina de la lista de trabajo de los equipos médicos.
2- La <b>Secretaria</b> selecciona los datos de la cita (servicio, local, especialidad, institución referidora, modalidad y equipo médico)	2.1- Carga la lista de trabajo del equipo médico seleccionado.
3- En la lista de trabajo mostrada presiona seleccionar en el horario deseado.	3.1-Habilita la opción <b>Nueva</b> .
1- Selecciona la opción <b>Nueva</b>	1.1- Muestra la página para buscar pacientes.
2- Introduce los criterios de búsqueda y presiona <b>buscar</b> .	2.1-Muestra los pacientes encontrados
3- La <b>Secretaria</b> selecciona el paciente que desea citar.	3.1 Muestra la página para introducir las indicaciones (información significativa relativa al proceso) de la cita.
4- La <b>Secretaria</b> introduce las indicaciones y selecciona la opción <b>terminar</b> .	4.1- Registra la cita.

Caso de uso – Registrar orden de Estudio Imagenológico

<b>CU-20</b>	Registrar Orden Estudio Imagenológico.	
<b>Actores:</b>	<b>Especialista</b> (Inicia)	
<b>Propósito:</b>	Registrar una nueva Orden de Estudio Imagenológico realizada a un paciente.	
<b>Resumen:</b>	El <b>Especialista</b> registra una Orden de Estudio Imagenológico realizada previamente a un paciente. Se registra las partes del cuerpo sometidas al estudio, tipo de estudio solicitado, y donde se debe buscar específicamente.	
<b>Referencias:</b>	RF-1, RF-8.6	
<b>Precondiciones:</b>	El paciente debe estar admitido en el hospital.	
<b>Curso Normal de los eventos</b>		
<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>	
1- El <b>Especialista</b> selecciona la opción de modificar historia clínica.	1.1- Muestra una página para buscar los pacientes admitidos y así modificar su historia clínica.	
2- Inserta los datos del paciente al cual se quiere registrar la Orden Estudio Imagenológico y busca.	2.1- Muestra la lista de pacientes que concuerden con los criterios de búsqueda entrados.	
3- Selecciona un paciente para registrar la Orden Estudio Imagenológico.	3.1 Muestra la página para registrar la Orden Estudio Imagenológico.	
4- Llena los datos para registrar la Orden Estudio Imagenológico y guarda los datos.	4.1- Guarda la información.	

Caso de uso – Mostrar Imágenes Médicas de XWeb

<b>CU-20</b>	Mostrar Imágenes Médicas de XWeb.	
<b>Actores:</b>	<b>Especialista.</b>	
<b>Propósito:</b>	Mostrar las imágenes médicas reportadas en XWeb de un paciente.	
<b>Resumen:</b>	El <b>Especialista</b> selecciona un paciente para ver las imágenes médicas que tiene reportado dicho paciente en XWeb.	
<b>Referencias:</b>	RF-1, RF-8.7	
<b>Precondiciones:</b>	El paciente debe estar admitido en el hospital.	
	<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1- El <b>Especialista</b> selecciona la opción de ver historia clínica.	1.1- Muestra una página para buscar los pacientes admitidos y así visualizar su historia clínica.
	2- Inserta los datos del paciente del cual se quiere ver las imágenes en XWeb.	2.1- Muestra la lista de pacientes que concuerden con los criterios de búsqueda entrados.
	3- Selecciona un paciente para visualizar sus imágenes.	3.1 Muestra la página que contendrá las imágenes del paciente en XWeb.

Caso de uso – Mostrar reportes médicos de XWeb

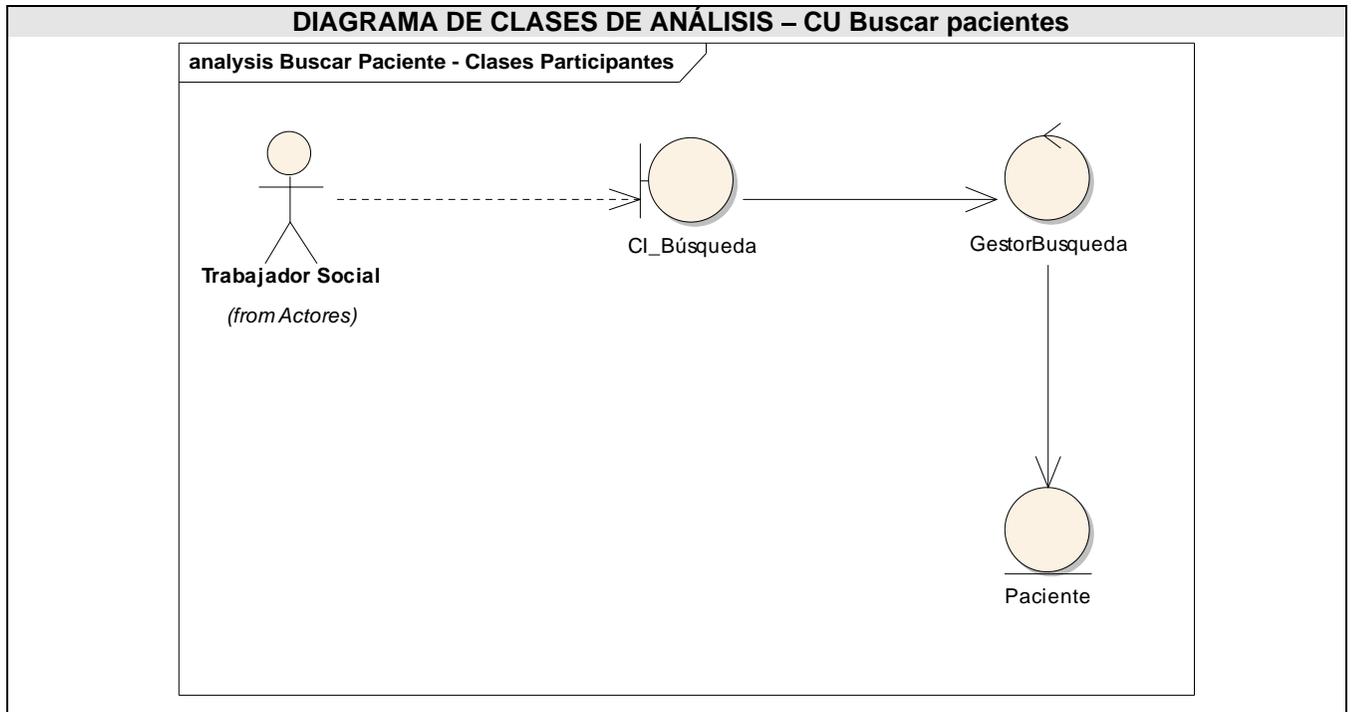
<b>CU-21</b>	Mostrar Reportes Médicos de XWeb.	
<b>Actores:</b>	<b>Especialista.</b>	
<b>Propósito:</b>	Mostrar los reportes médicos existentes en XWeb de un paciente.	
<b>Resumen:</b>	El <b>Especialista</b> selecciona un paciente para ver los reportes médicos que tiene dicho paciente en XWeb.	
<b>Referencias:</b>	RF-1, RF-8.8	
<b>Precondiciones:</b>	El paciente debe estar admitido en el hospital.	
	<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1- El <b>Especialista</b> selecciona la opción de ver historia clínica.	1.1- Muestra una página para buscar los pacientes admitidos y así visualizar su historia clínica.
	2- Inserta los datos del paciente del cual se quiere ver los reportes médicos.	2.1- Muestra la lista de pacientes que concuerden con los criterios de búsqueda entrados.
	3- Selecciona un paciente para visualizar los reportes.	3.1 Muestra la página que contendrá los reportes médicos del paciente en XWeb.

Caso de uso - Asignar cama

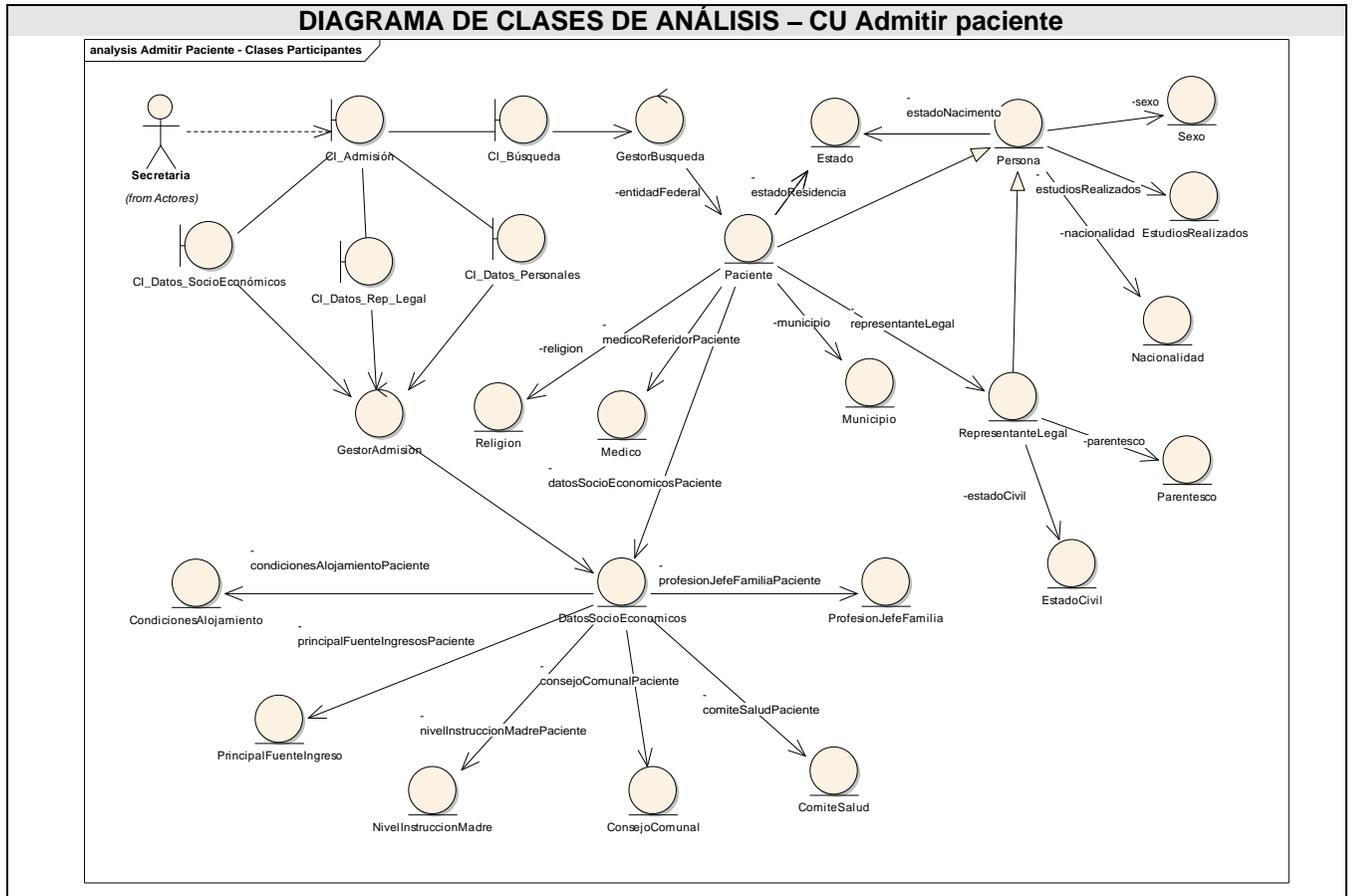
<b>CU-23</b>	Asignar Cama.	
<b>Actores:</b>	<b>Secretaria</b> (inicia)	
<b>Propósito:</b>	Asignar la cama de ingreso de un paciente.	
<b>Resumen:</b>	La <b>Secretaria</b> selecciona el paciente ingresado al que desea asignarle cama y selecciona la opción asignar de los detalles del ingreso del paciente. Luego selecciona un ala (norte, sur, este, oeste), un nivel del hospital relacionado con la especialidad en la que el paciente será atendido, un local de ingreso y una cama de éste. Por último selecciona la <b>Asignar cama</b> quedando el paciente en esa ubicación.	
<b>Referencias:</b>	RF-5, RF-9.1	
<b>Precondiciones:</b>	El paciente debe estar ingresado y no poseer cama asignada.	
	<b>Acciones del Actor</b>	<b>Respuesta del Sistema</b>
	1- La <b>Secretaria</b> solicita ver los ingresos actuales del hospital.	1.1- Muestra una página que tendrá la lista de los pacientes ingresados.
	2- La <b>Secretaria</b> selecciona un paciente para ver los detalles de su ingreso.	2.1- Muestra los detalles del ingreso seleccionado.
	3- En caso que el paciente no tenga asignada una cama permanente La <b>Secretaria</b> selecciona la opción <b>asignar cama</b> .	3.1 Muestra una página con los datos del ingreso actual del paciente seleccionado y muestra la lista de camas disponibles.
	4- La <b>Secretaria</b> selecciona la ubicación del paciente, o sea, en que nivel, ala, habitación y la cama donde se ubicará al paciente durante su ingreso y presiona <b>Asignar</b> .	4.1- Guarda la información correspondiente a la ubicación del paciente en el hospital y notifica a la <b>Secretaria</b> .

## Anexo 2 Diagramas de clases del análisis

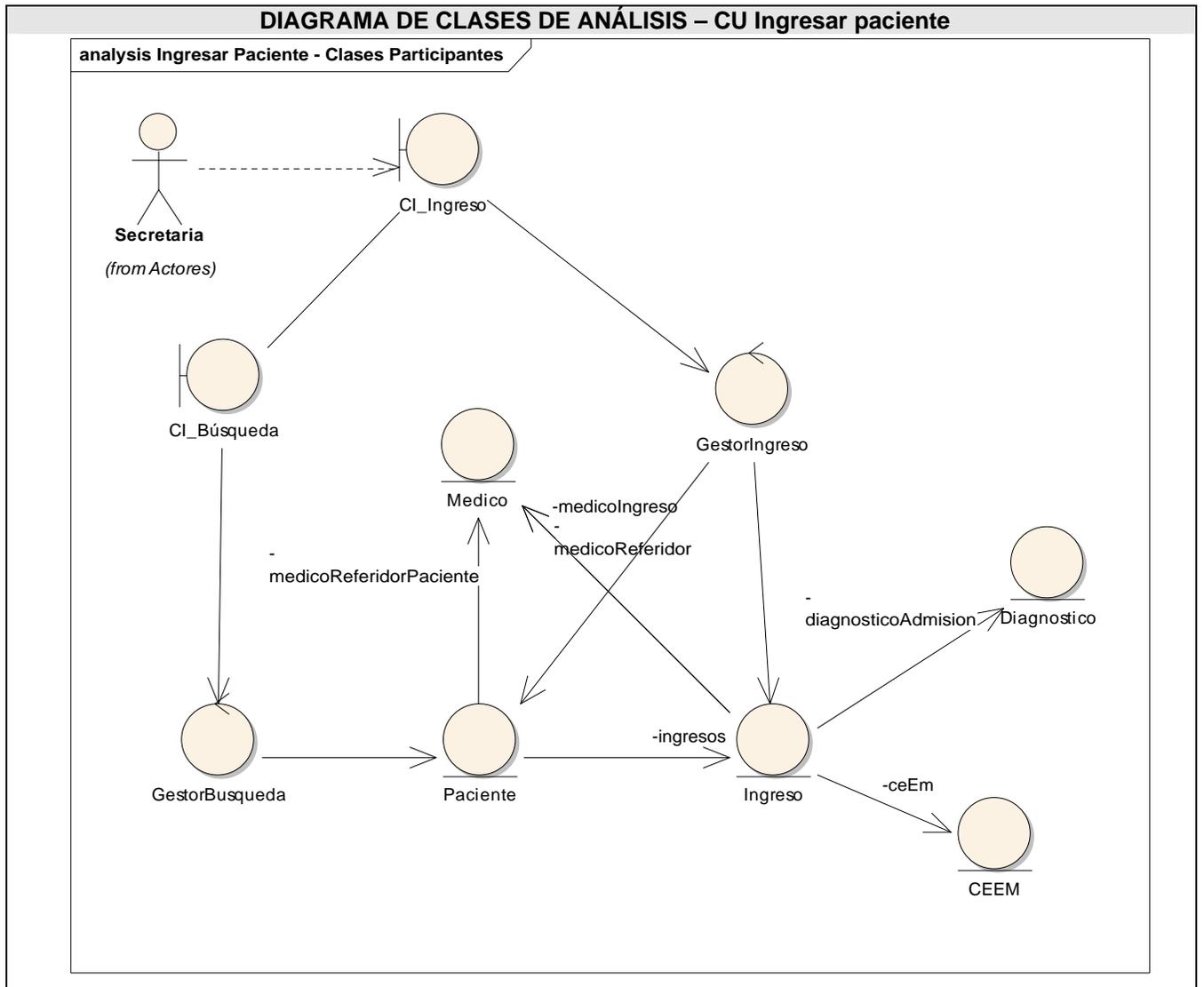
### 3.1.1 Diagrama de clases – Buscar pacientes



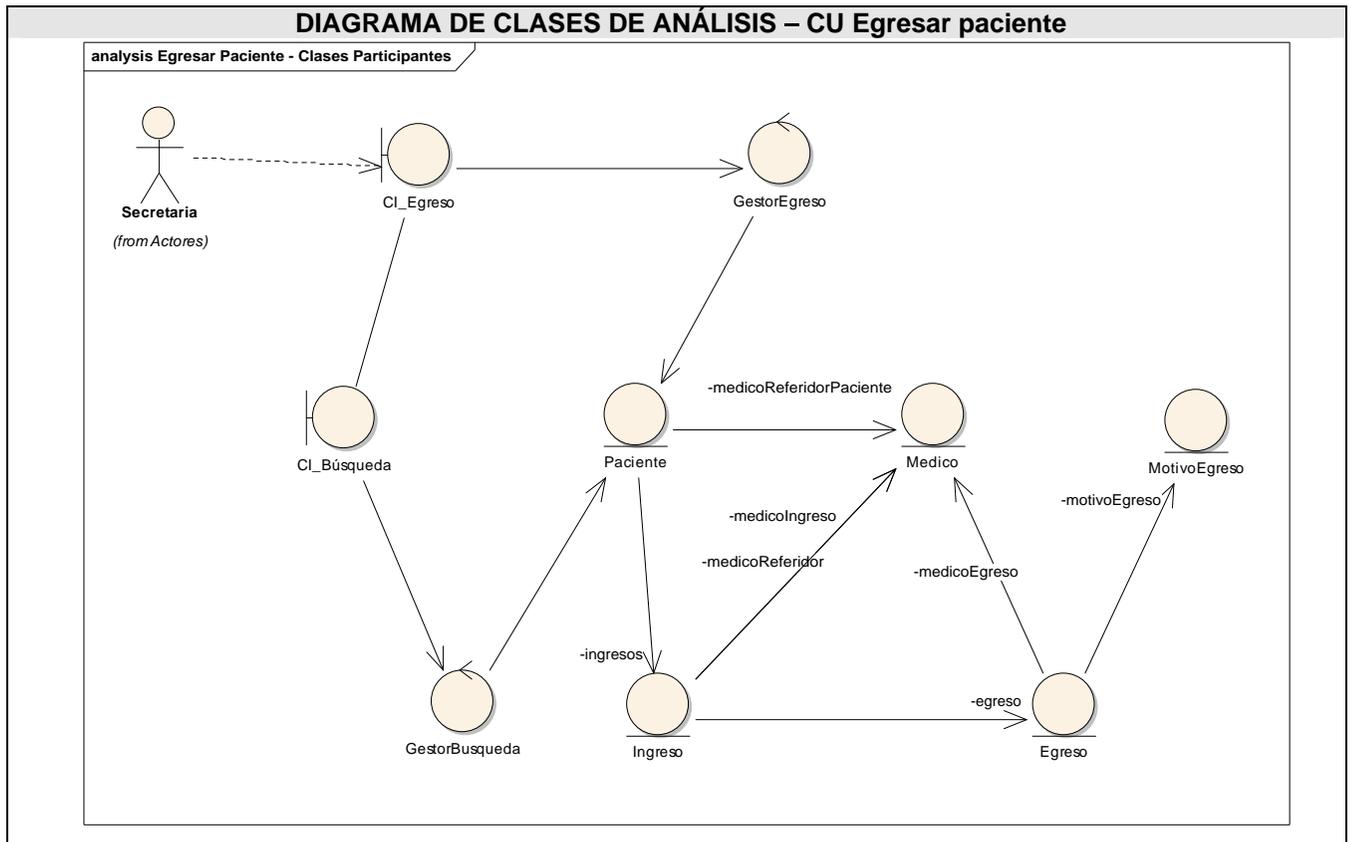
### 3.1.2 Diagrama de clases – Admitir paciente



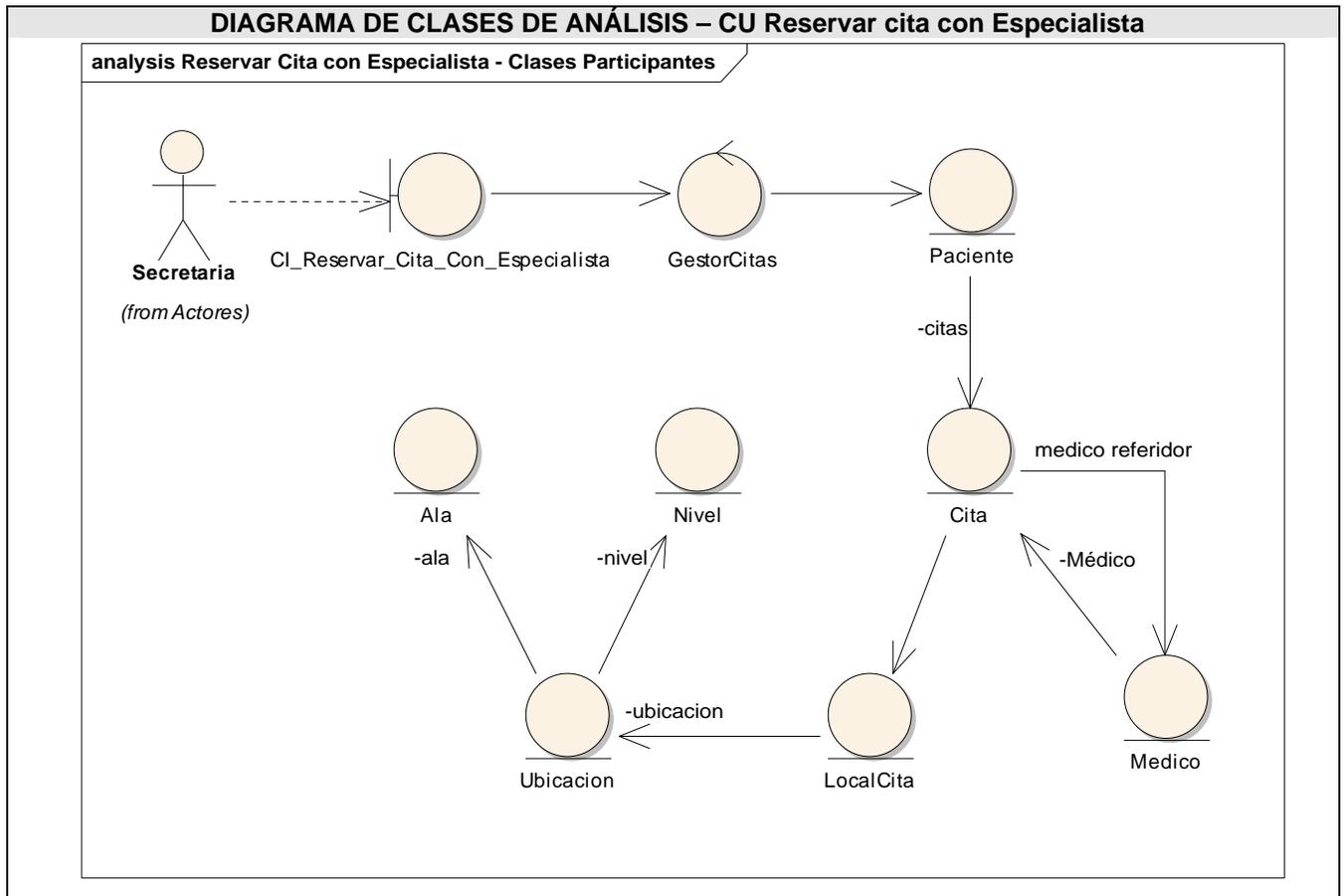
### 3.1.3 Diagrama de clases – Ingresar paciente



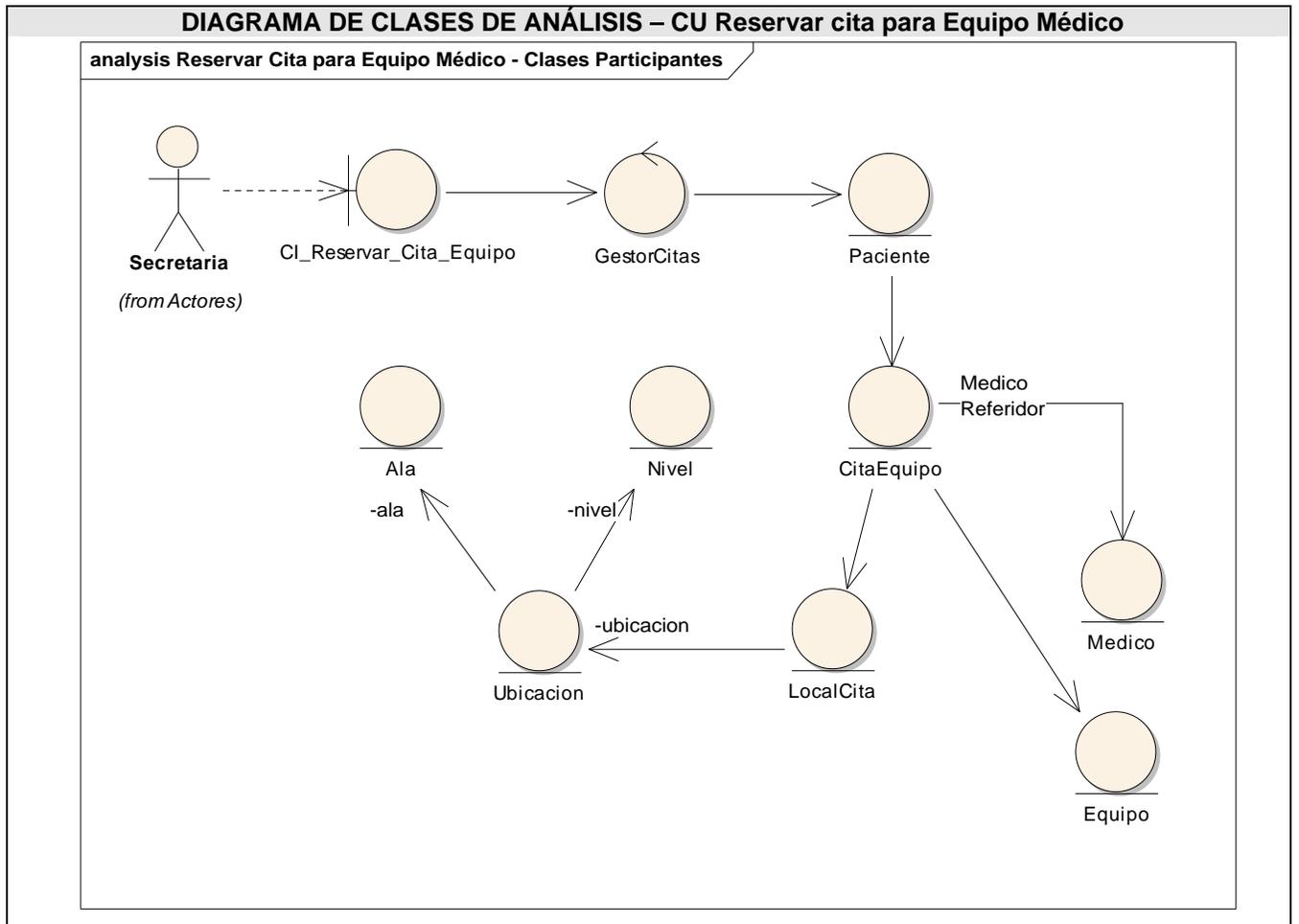
### 3.1.4 Diagrama de clases – Egresar paciente



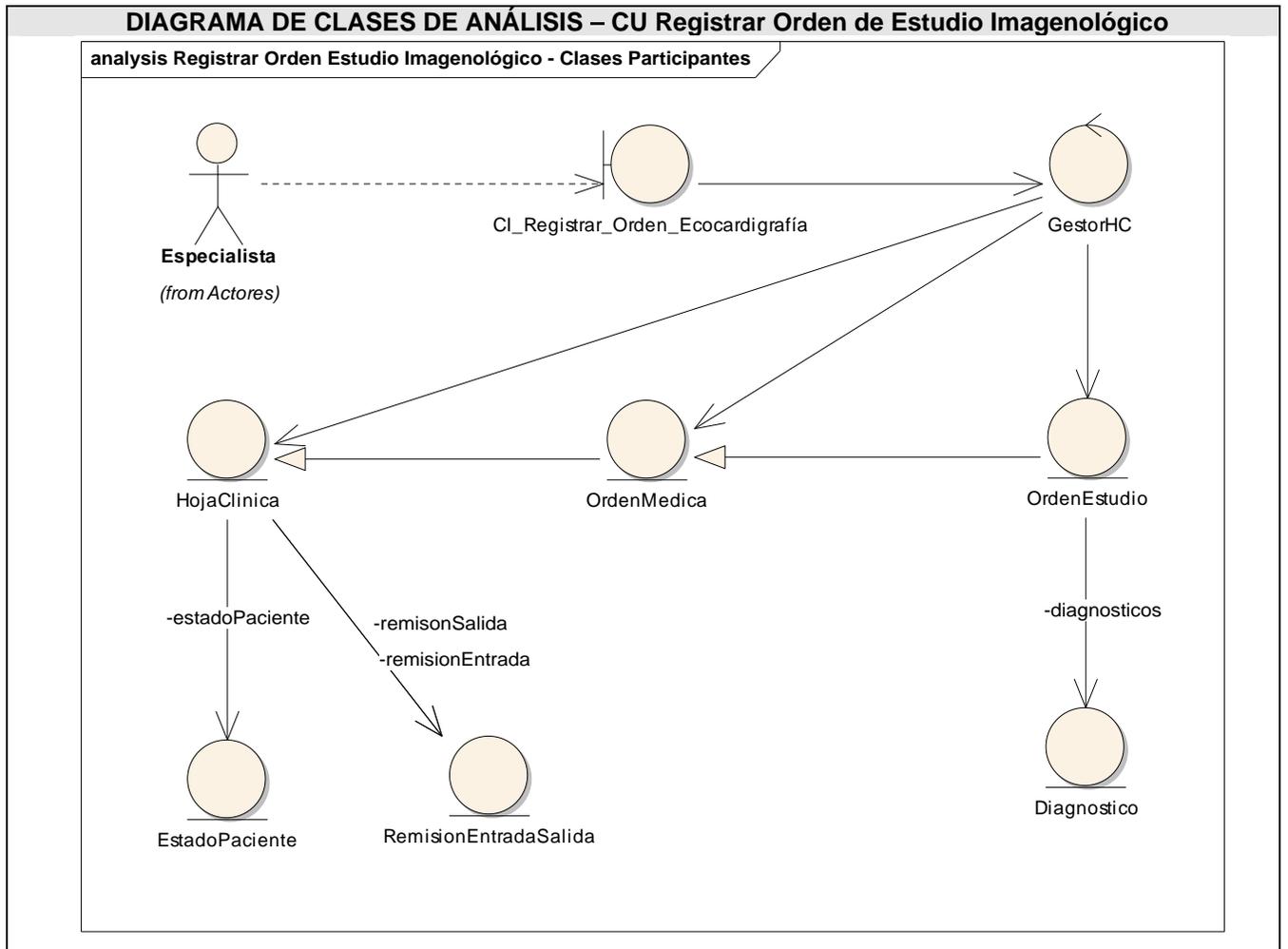
### 3.1.5 Diagrama de clases – Reservar cita con Especialista



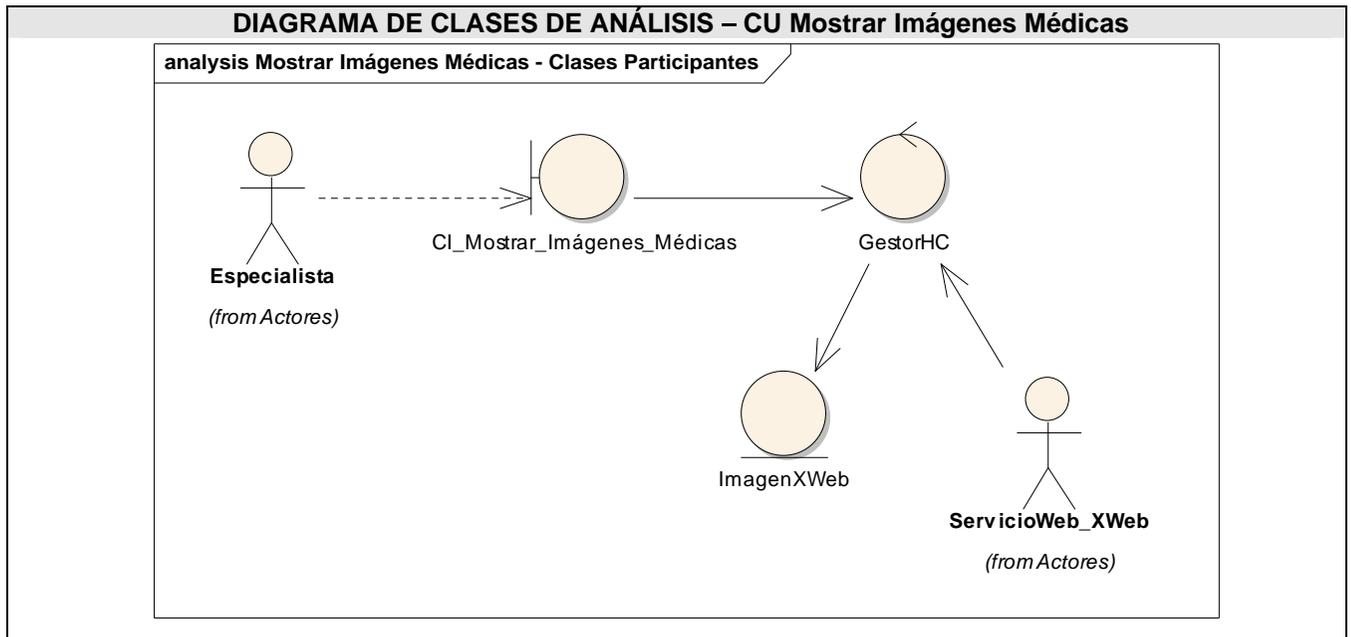
### 3.1.6 Diagrama de clases – Reservar cita para Equipo Médico



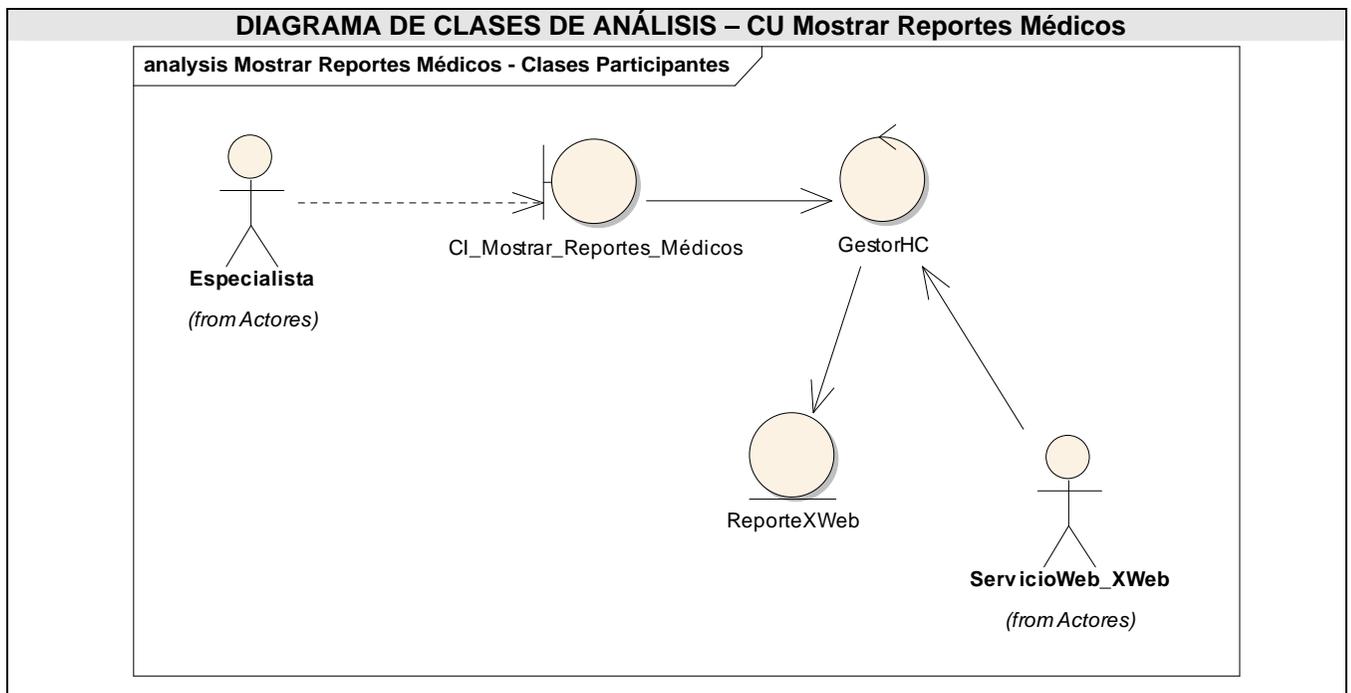
### 3.1.7 Diagrama de clases – Registrar Orden de Estudio Imagenológico



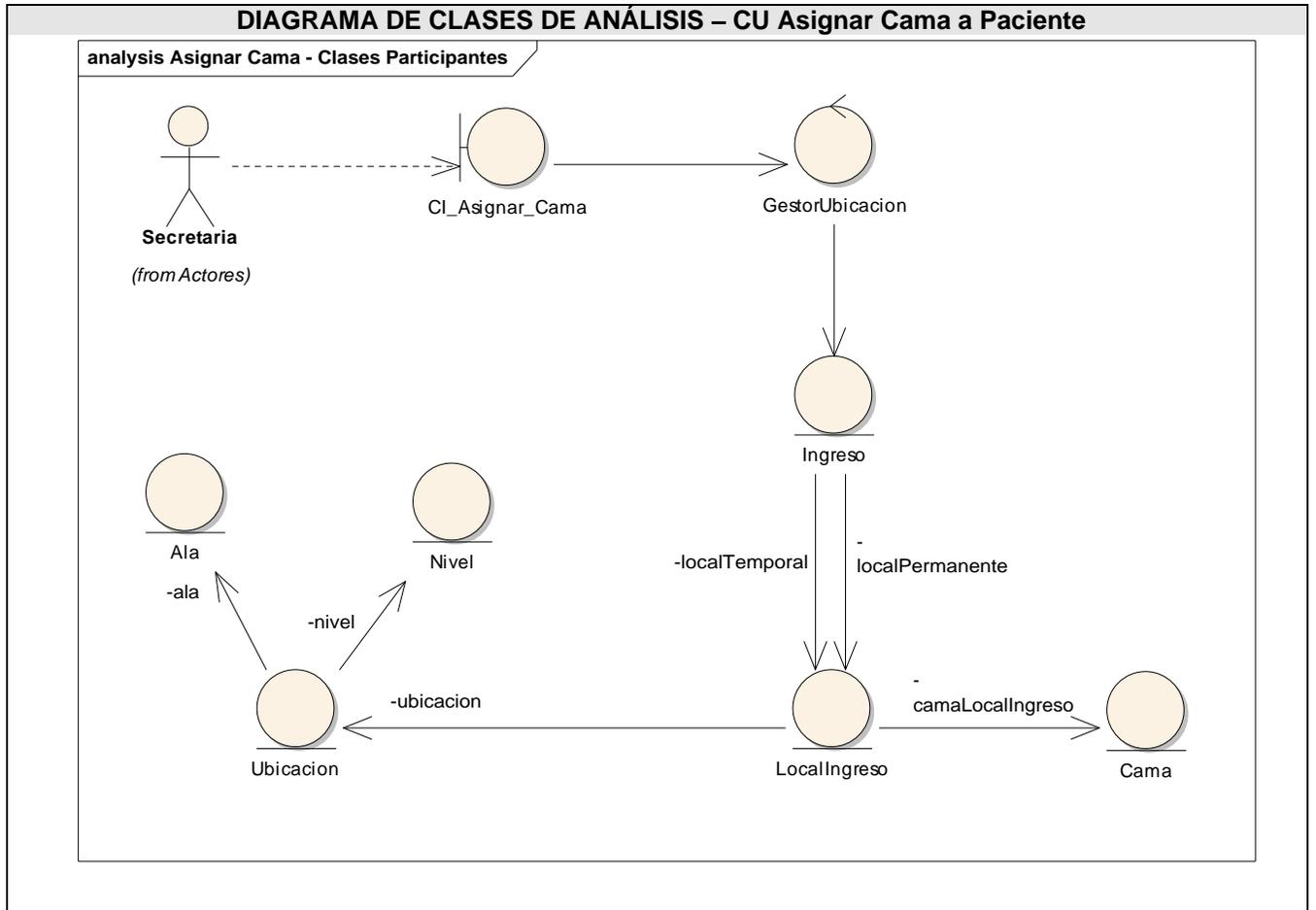
### 3.1.8 Diagrama de clases – Mostrar Imágenes Médicas



### 3.1.9 Diagrama de clases – Mostrar Reportes Médicos

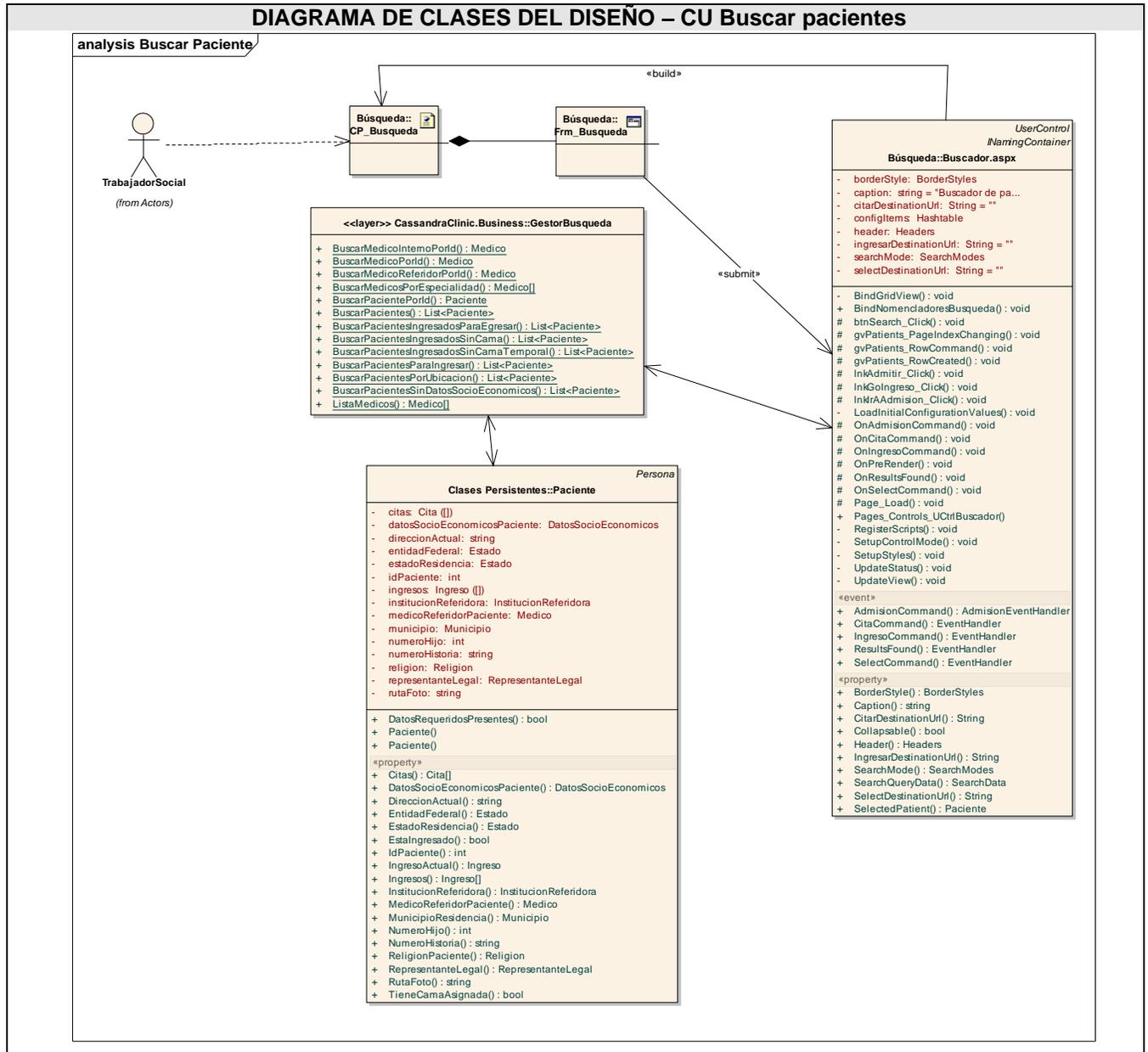


### 3.1.10 Diagrama de clases – Asignar Cama a Paciente



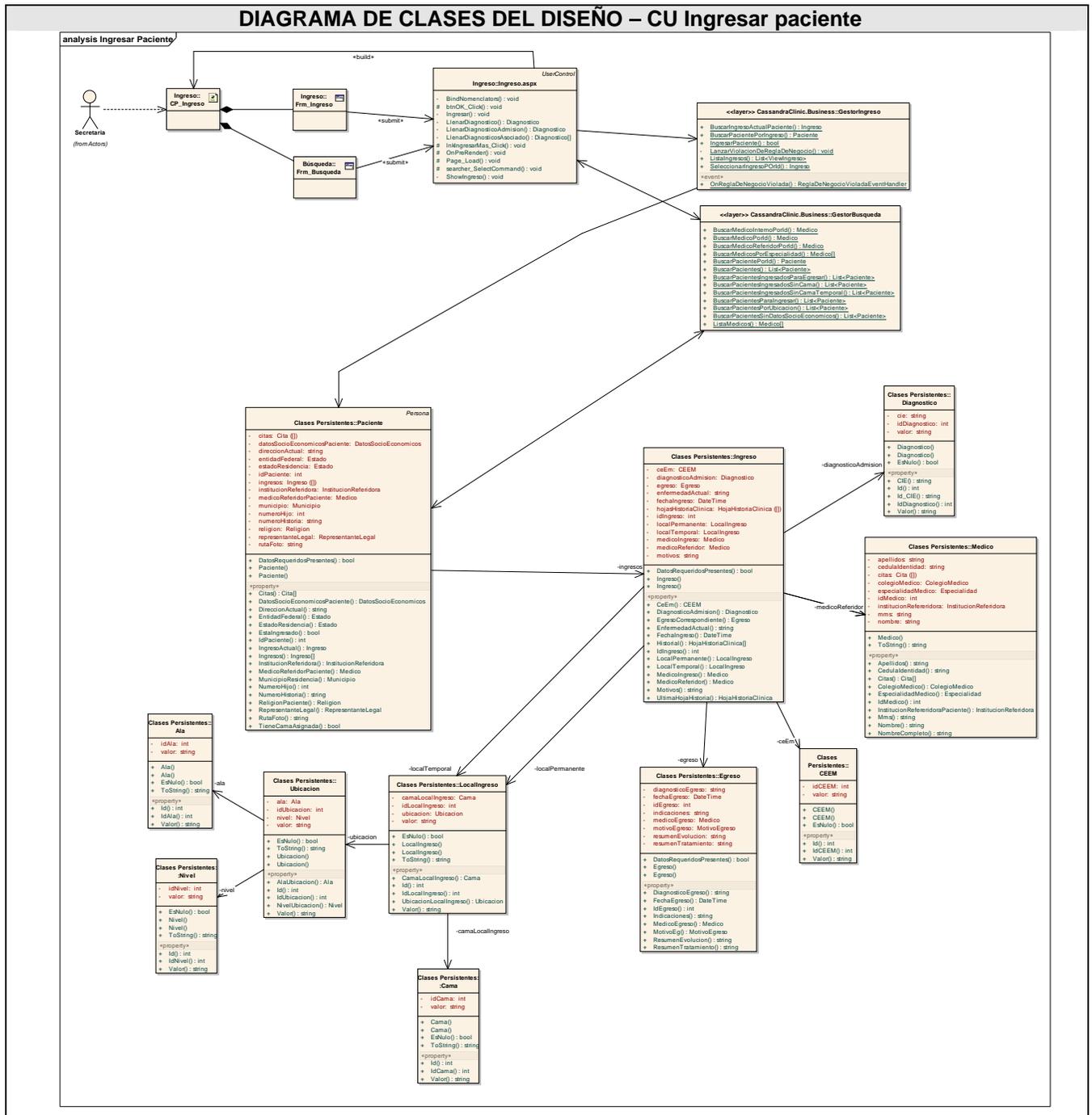
### Anexo 3 Diagramas de clases del diseño

### 3.2.1 Diagrama de clases del diseño – Buscar pacientes

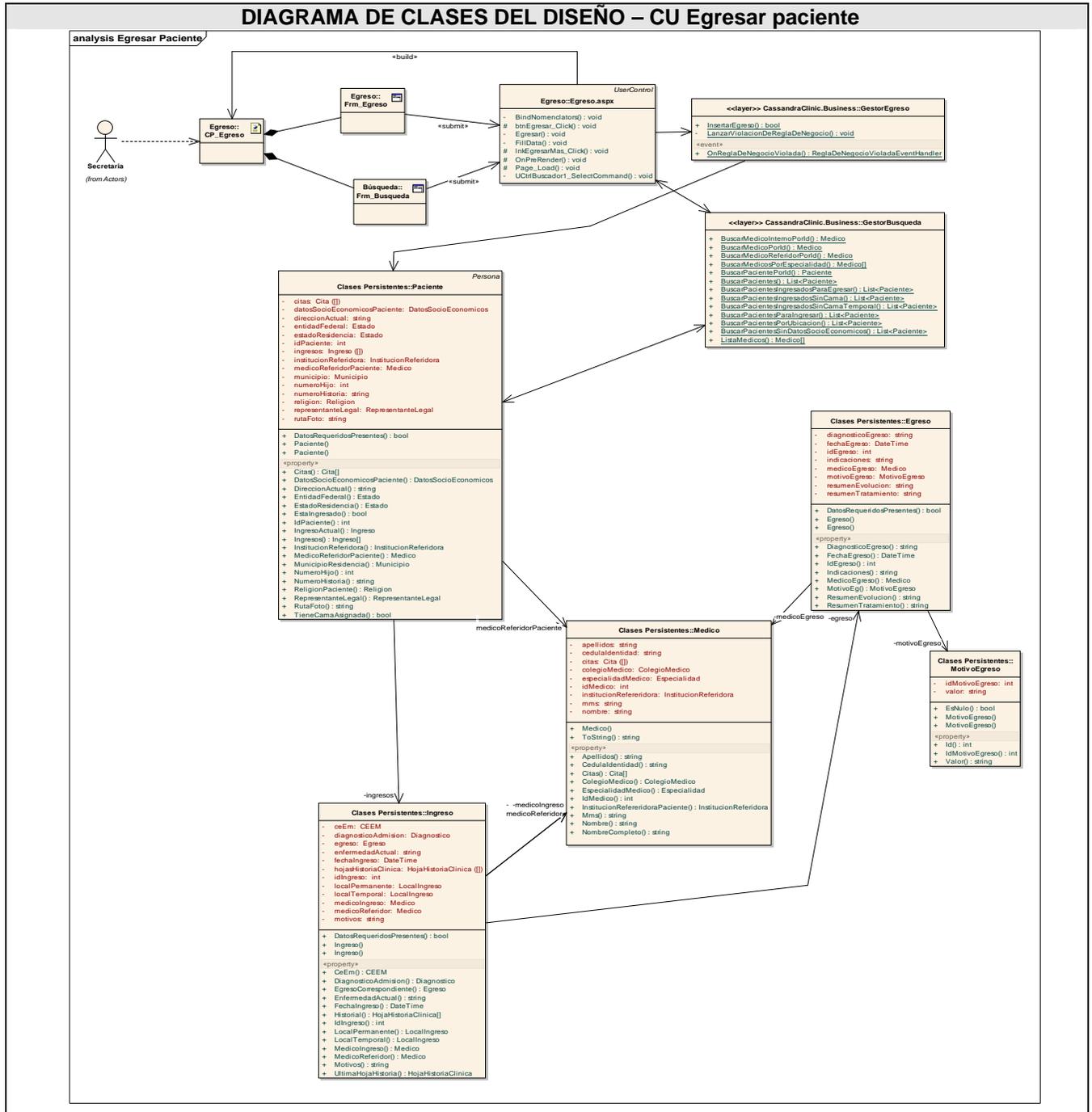




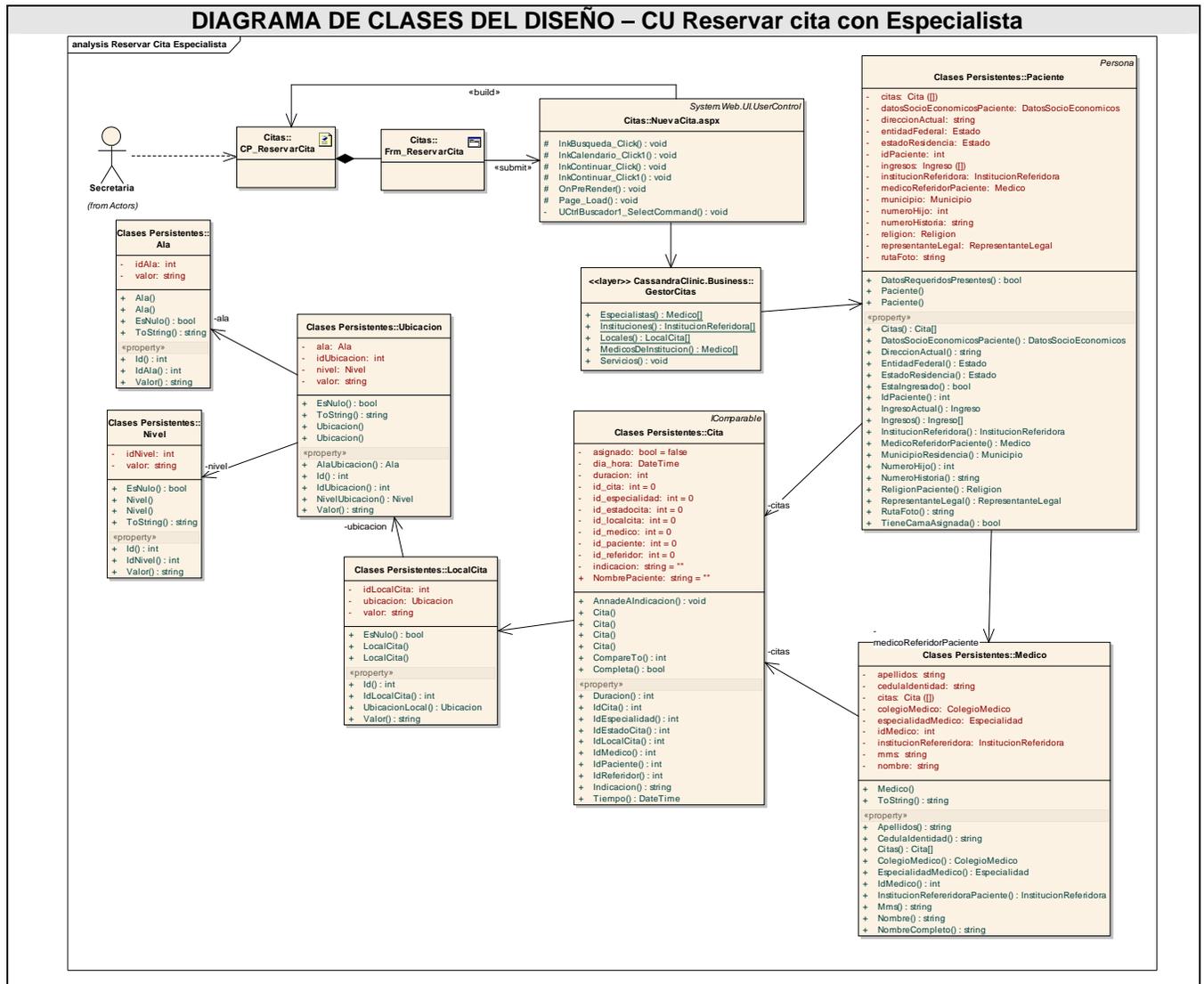
### 3.2.3 Diagrama de clases del diseño – Ingresar paciente



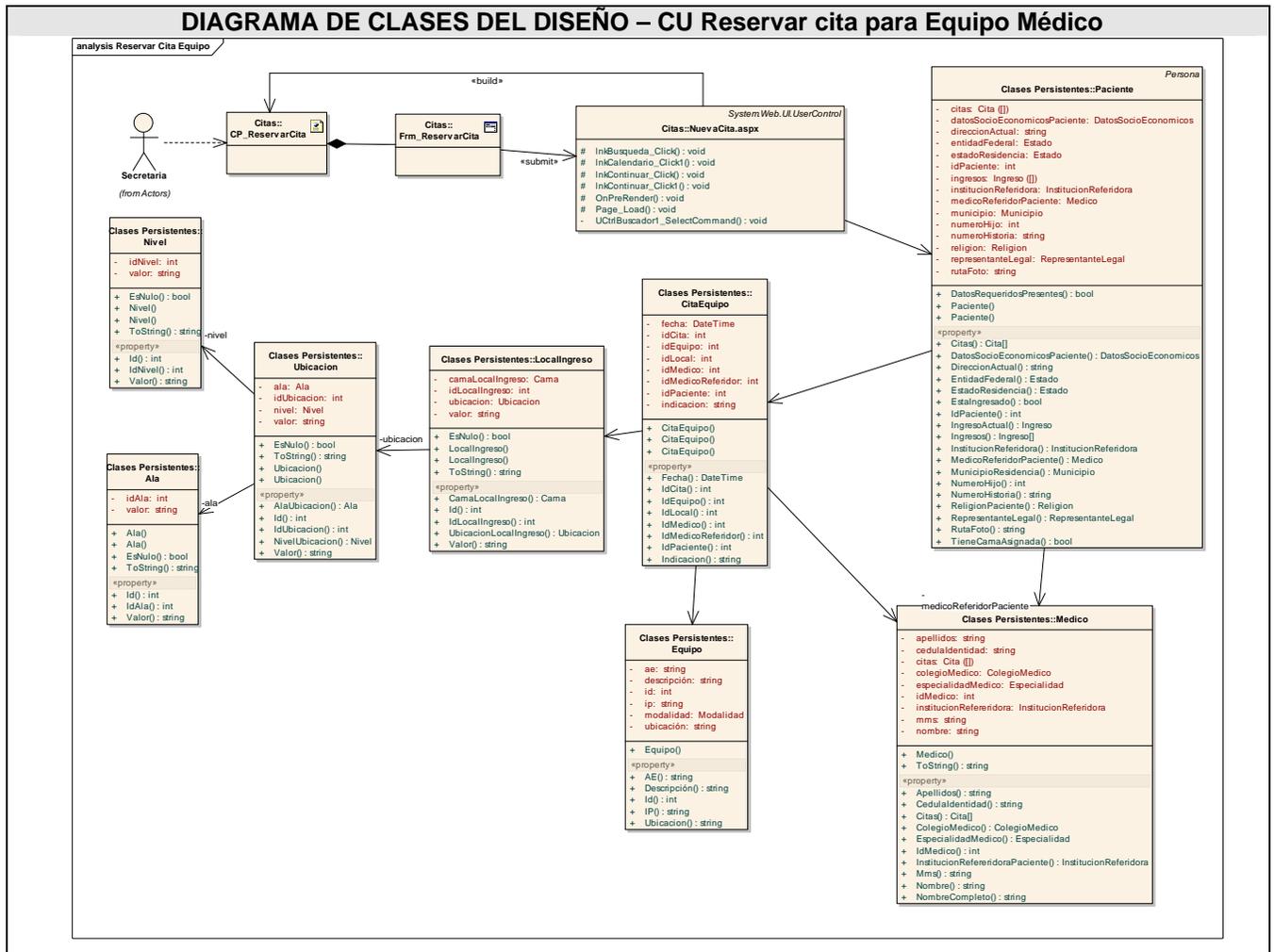
### 3.2.4 Diagrama de clases del diseño – Egresar paciente



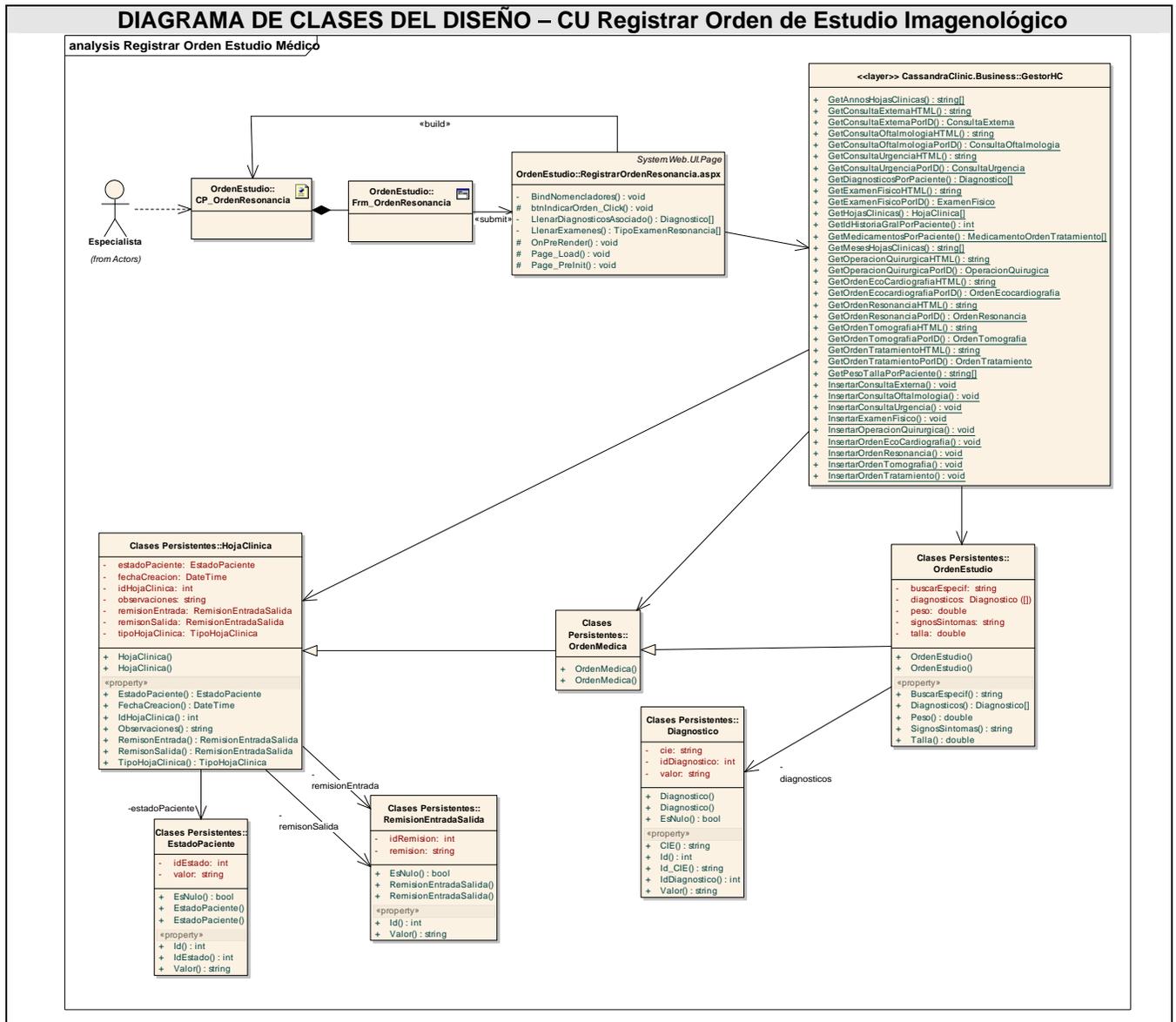
### 3.2.5 Diagrama de clases del diseño – Reservar cita con Especialista



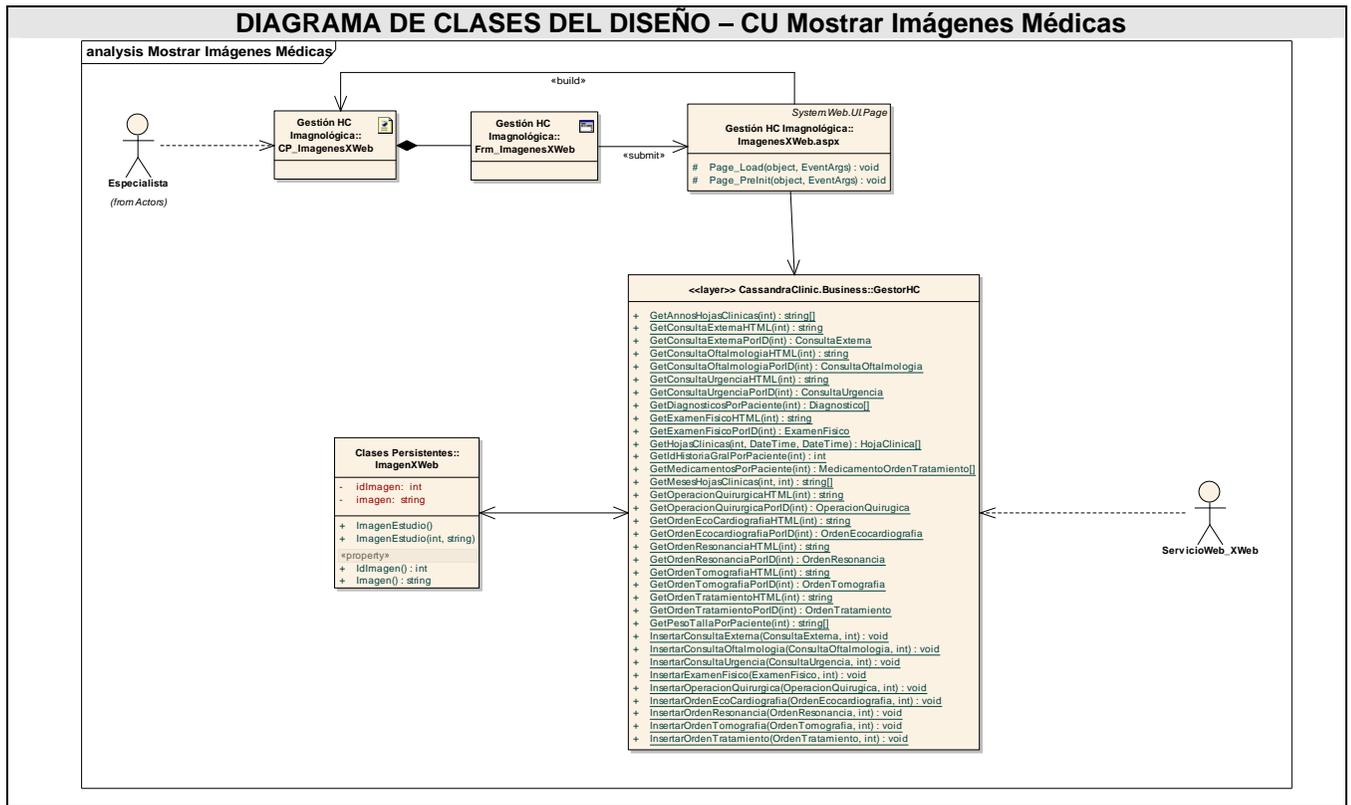
### 3.2.6 Diagrama de clases del diseño – Reservar cita para Equipo Médico



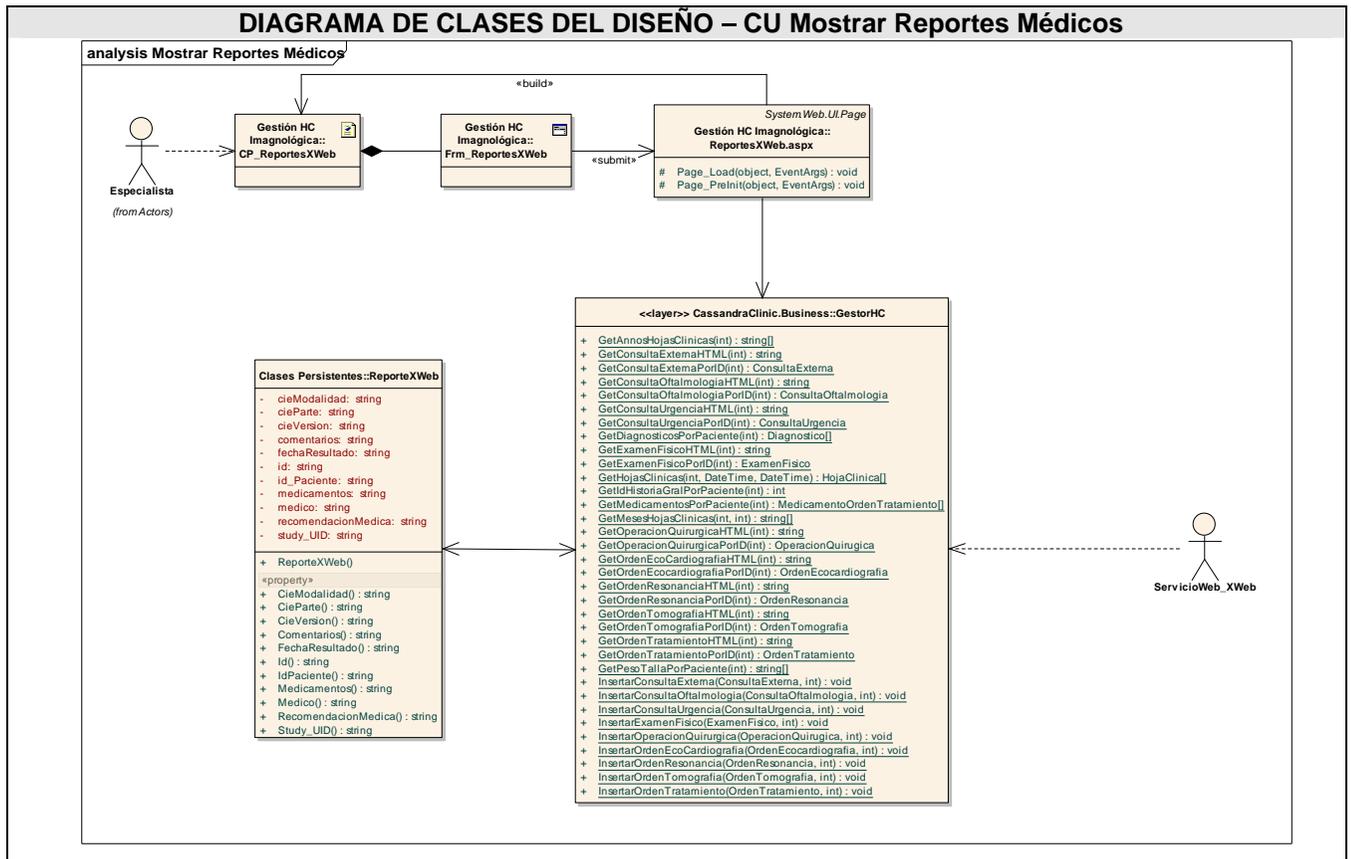
### 3.2.7 Diagrama de clases del diseño – Registrar Orden de Estudio Imagenológico



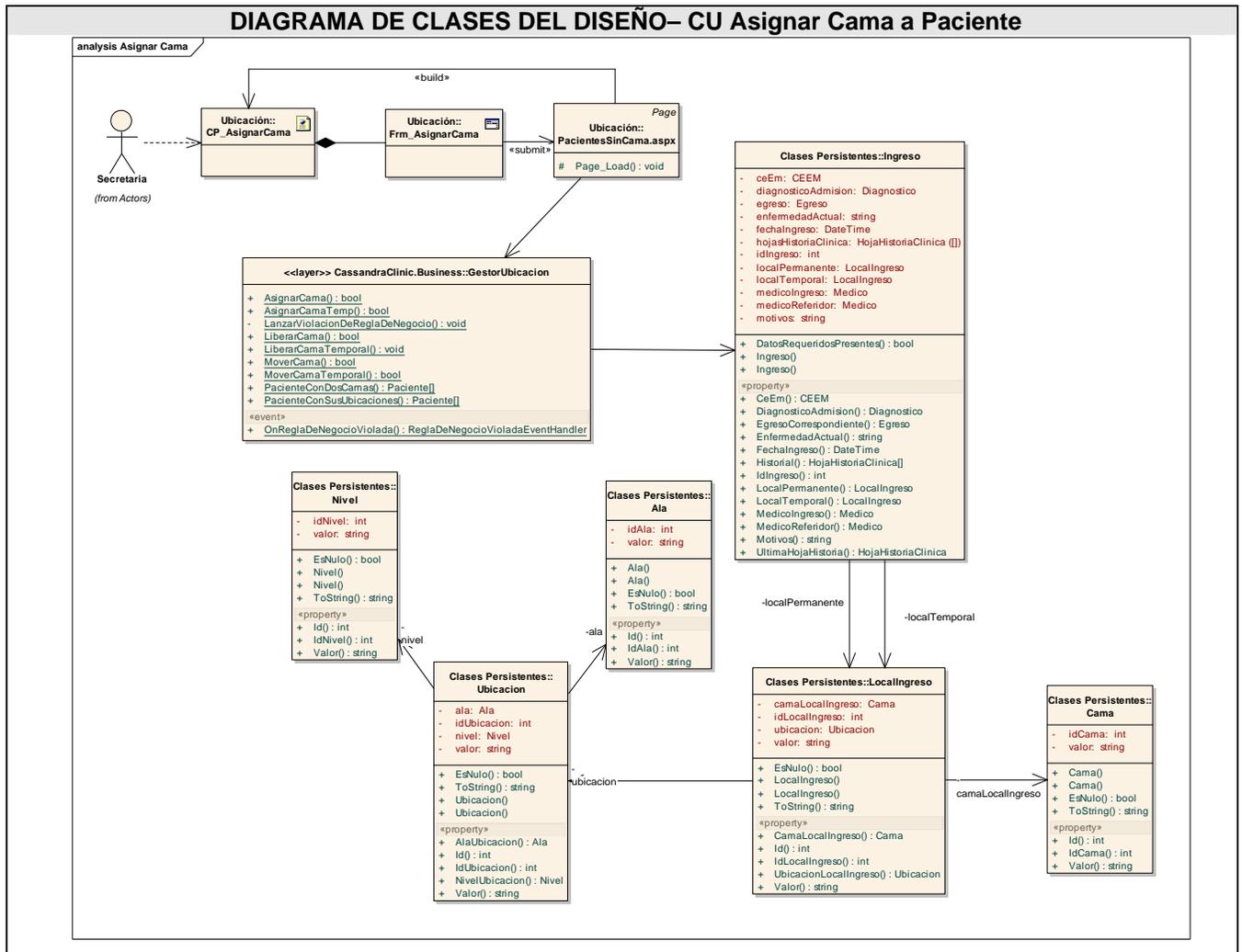
### 3.2.8 Diagrama de clases del diseño – Mostrar Imágenes Médicas



### 3.2.9 Diagrama de clases del diseño – Mostrar Reportes Médicos

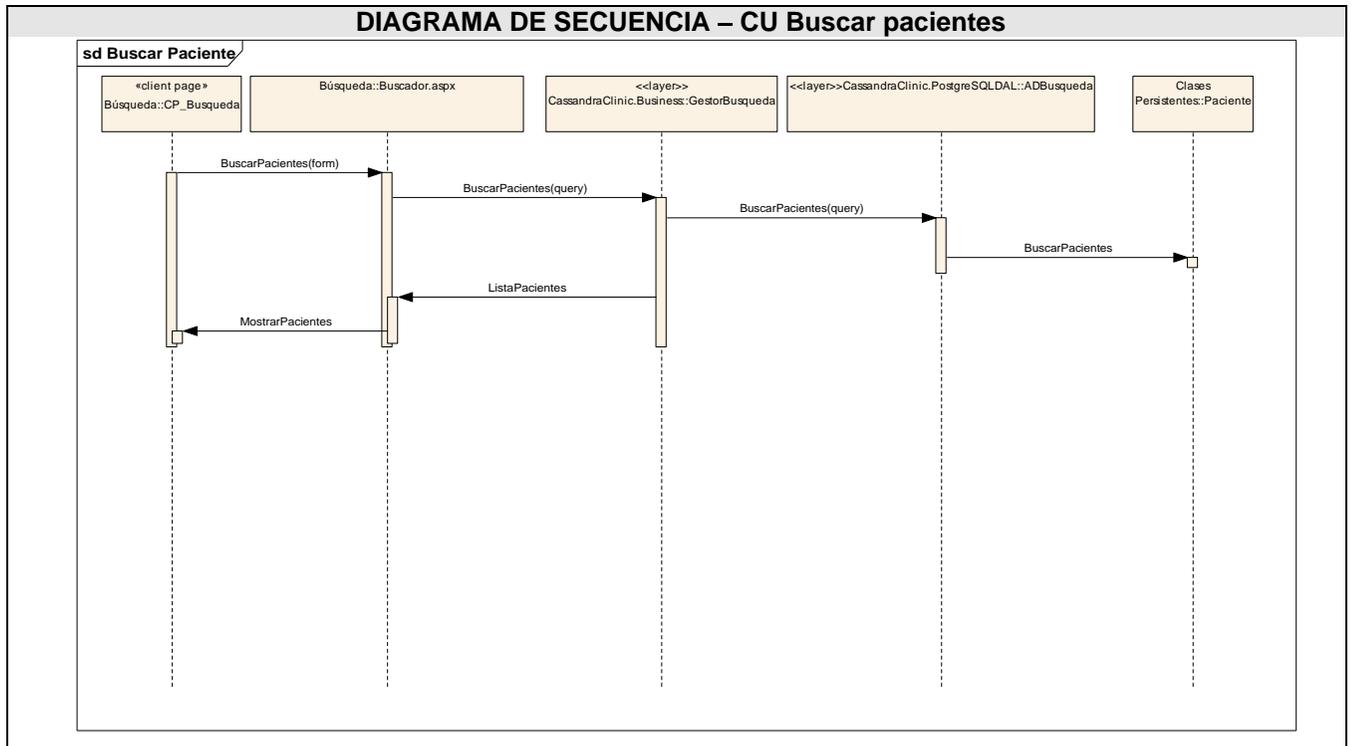


### 3.2.10 Diagrama de clases del diseño – Asignar Cama a Paciente

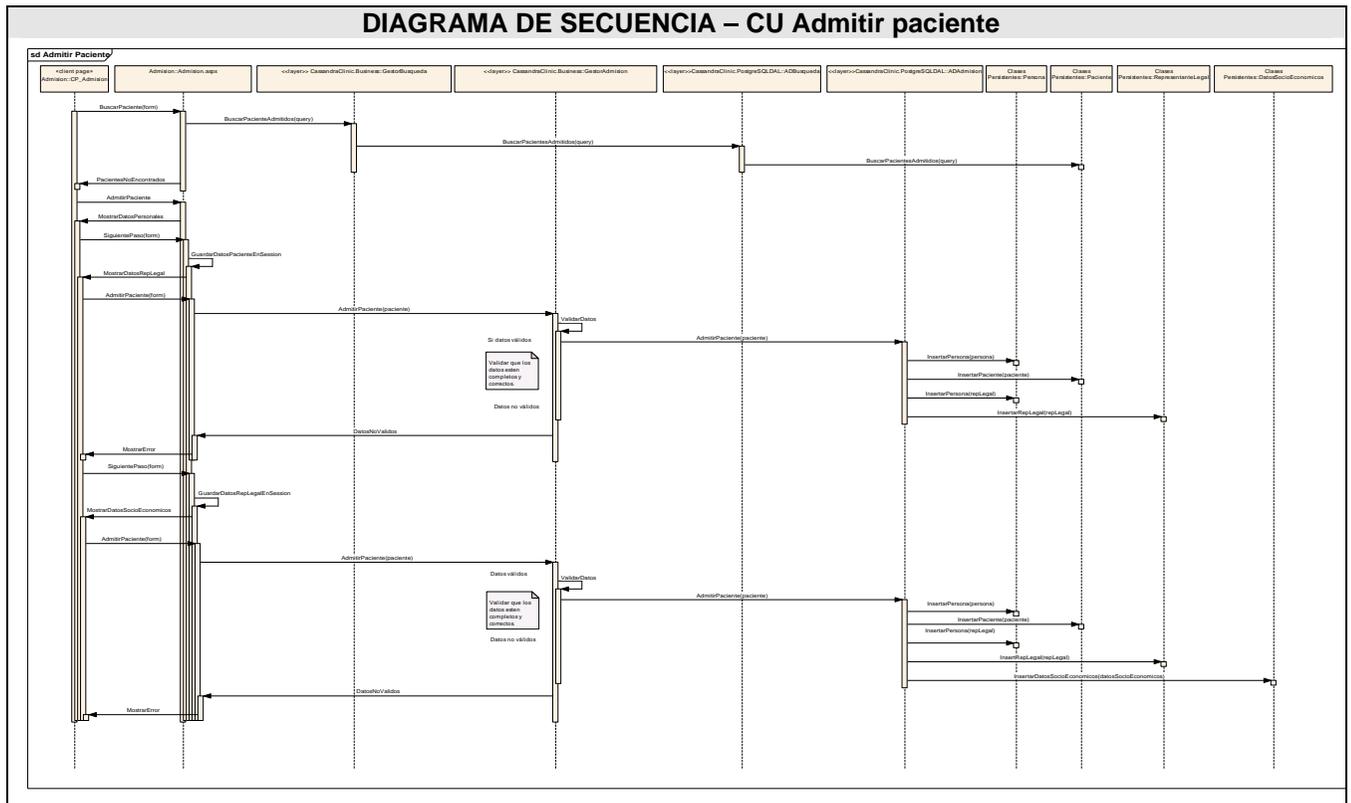


## Anexo 4 Diagramas de secuencia

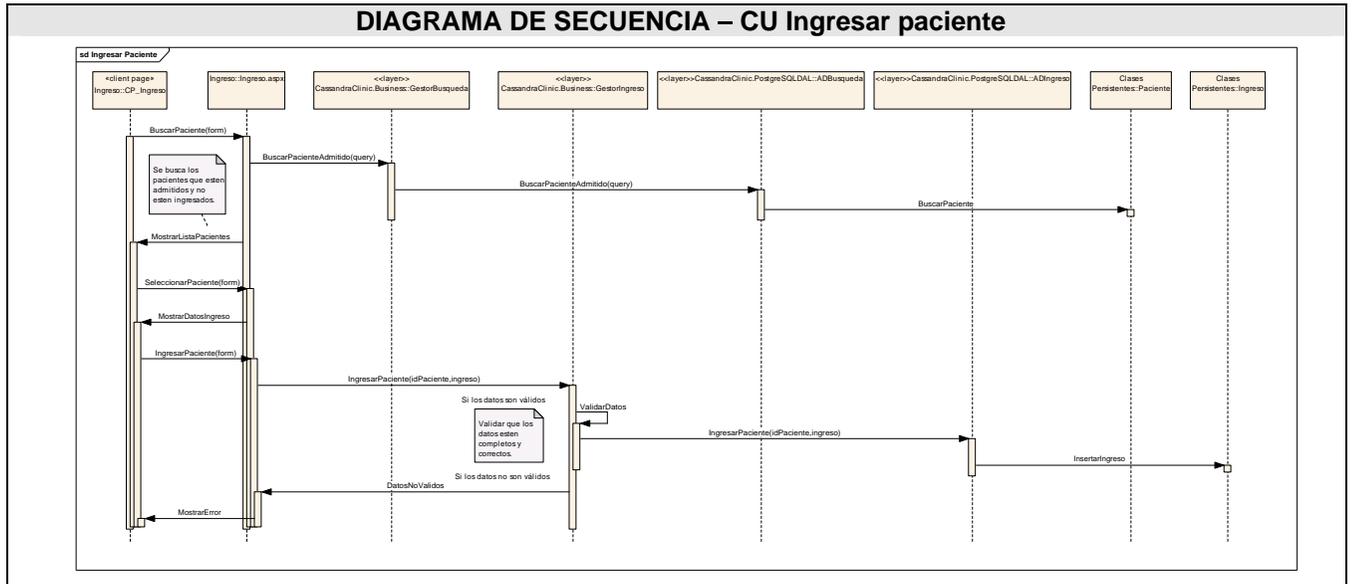
### 3.3.1 Diagrama de secuencia – Buscar pacientes



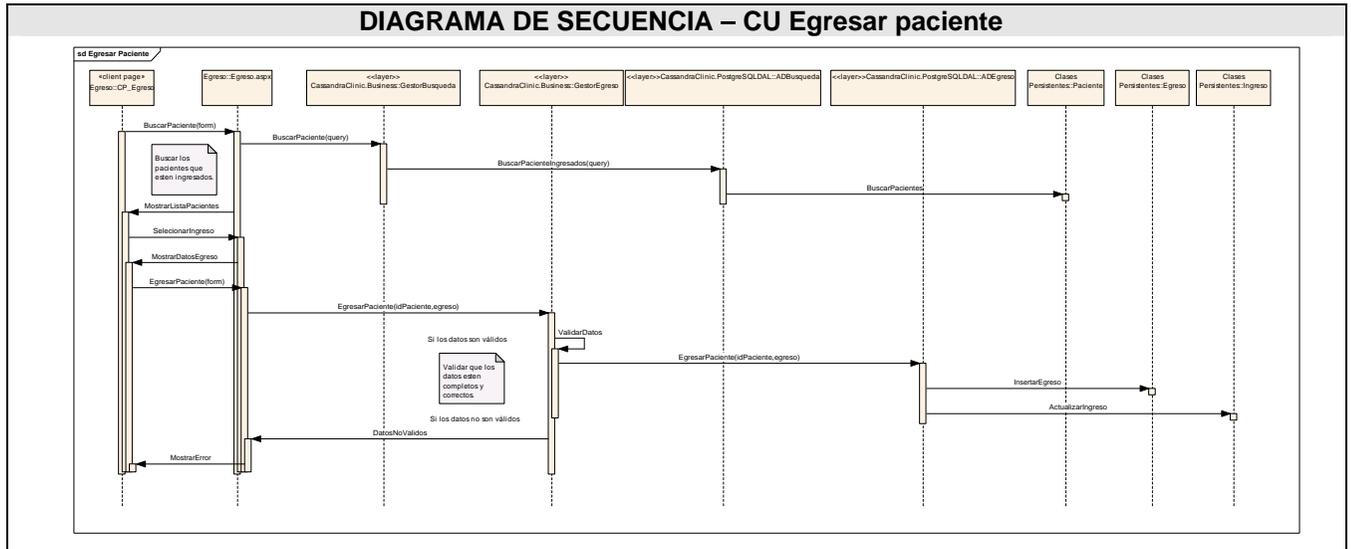
### 3.3.2 Diagrama de secuencia - Admitir paciente



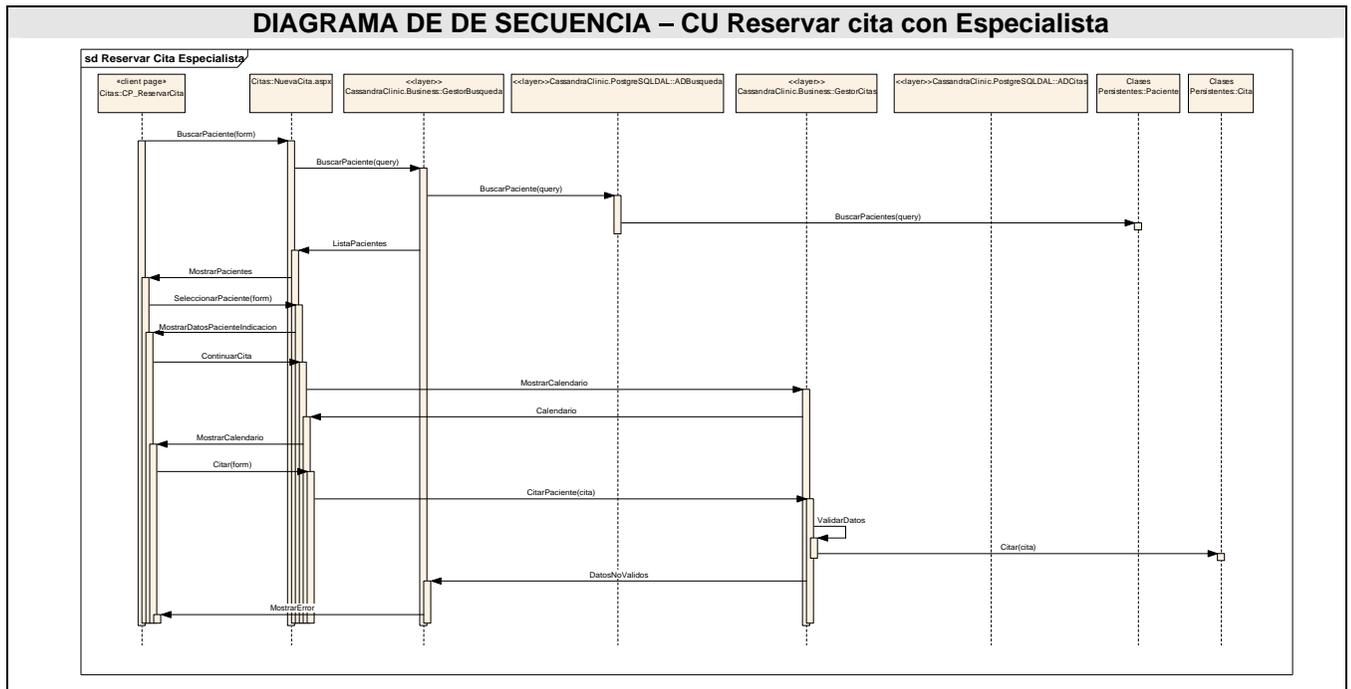
### 3.3.3 Diagrama de secuencia – Ingresar paciente



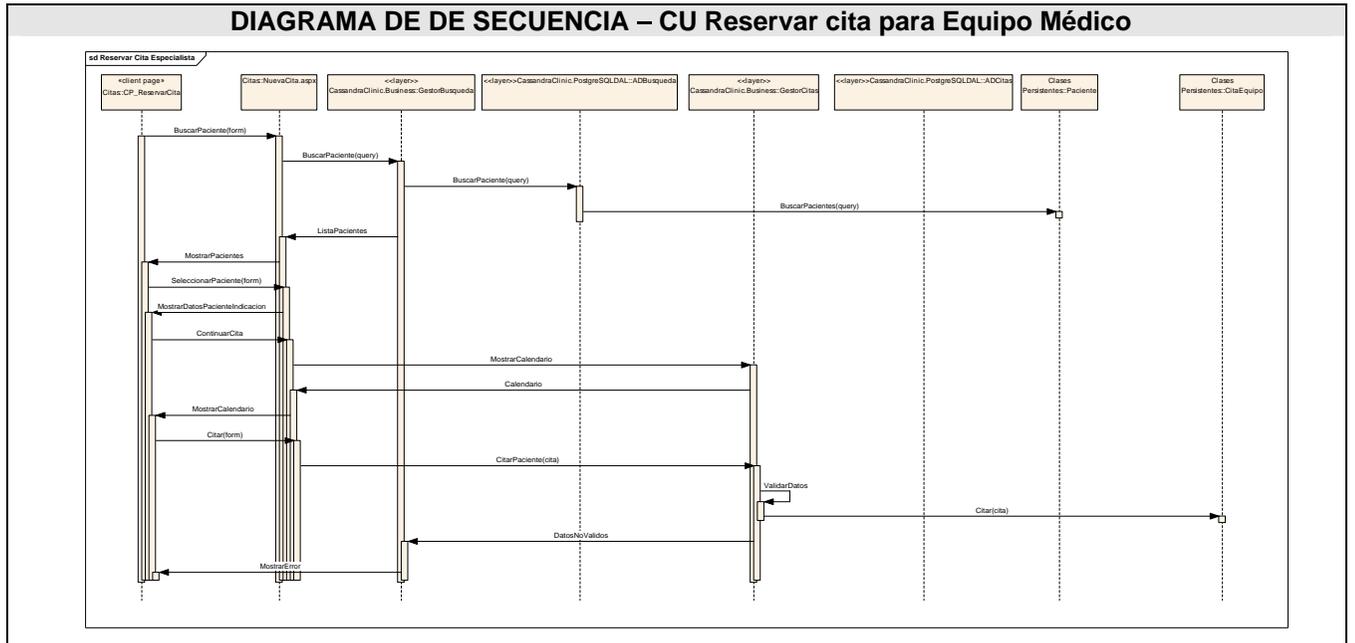
### 3.3.4 Diagrama de secuencia – Egresar paciente



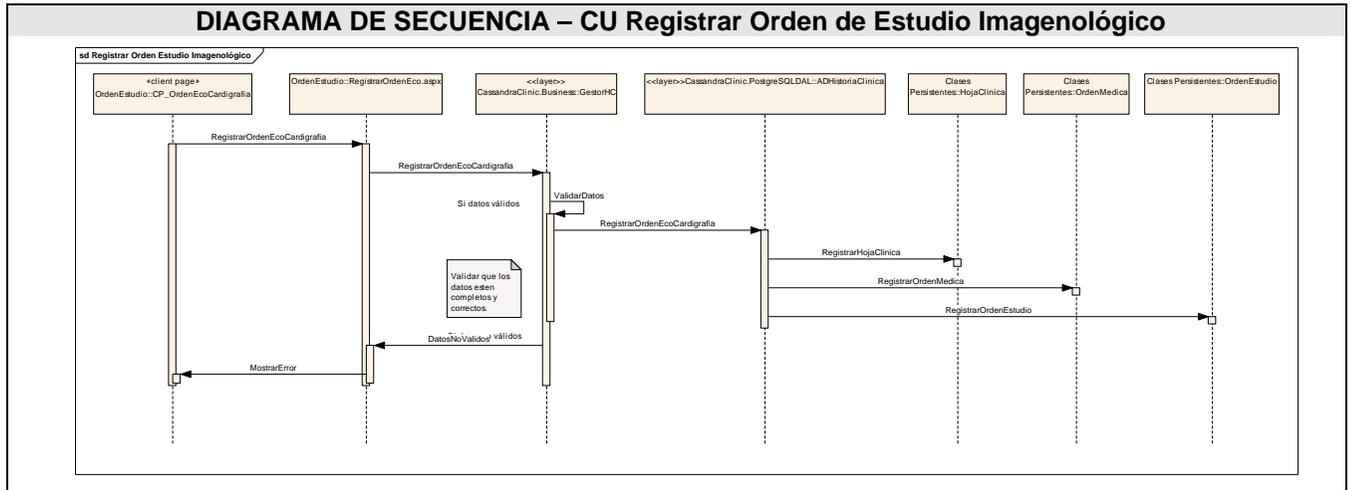
### 3.3.5 Diagrama de secuencia – Reservar cita con Especialista



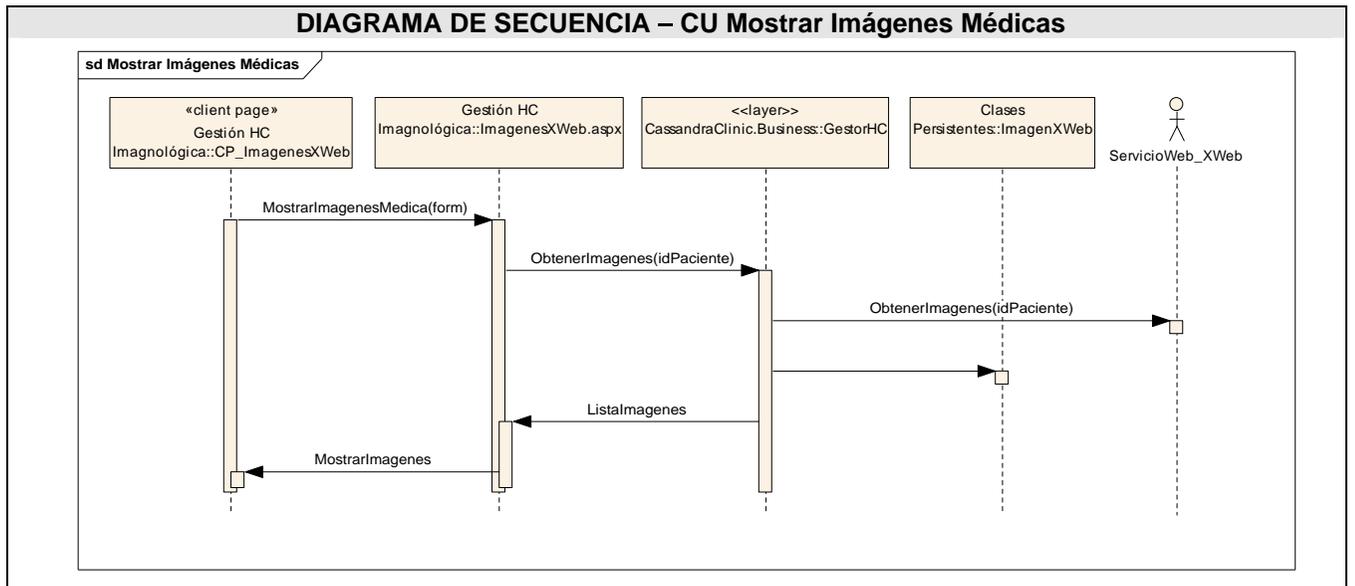
### 3.3.6 Diagrama de secuencia – Reservar cita para Equipo Médico



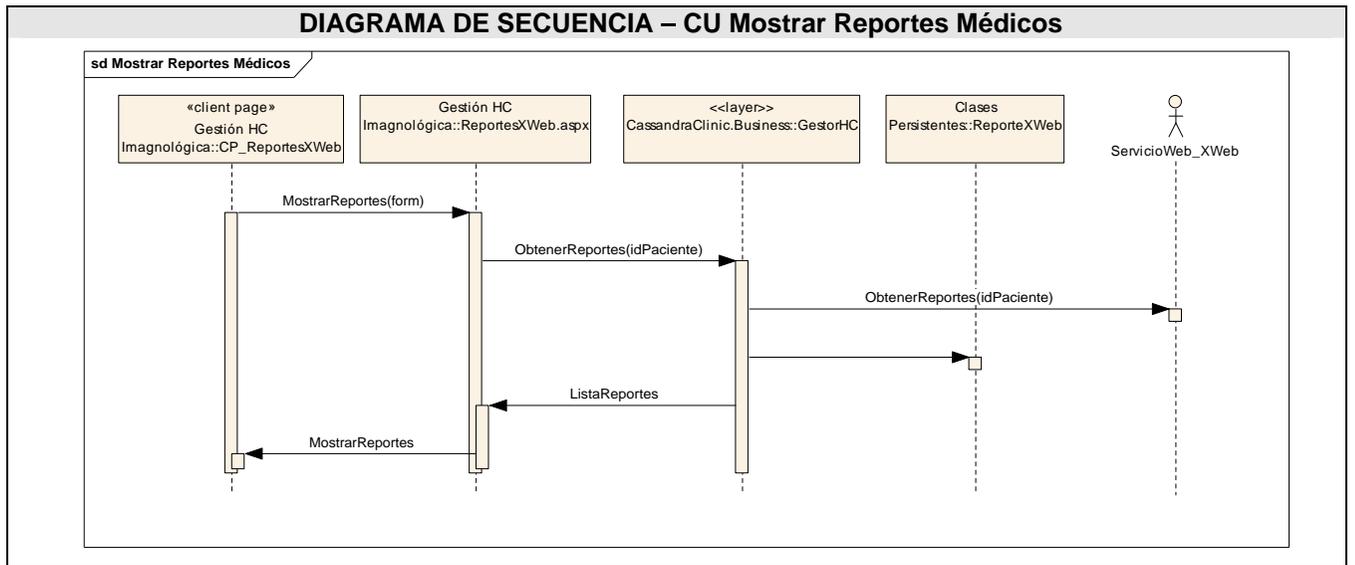
### 3.3.7 Diagrama de secuencia – Registrar Orden de Estudio Imagenológico



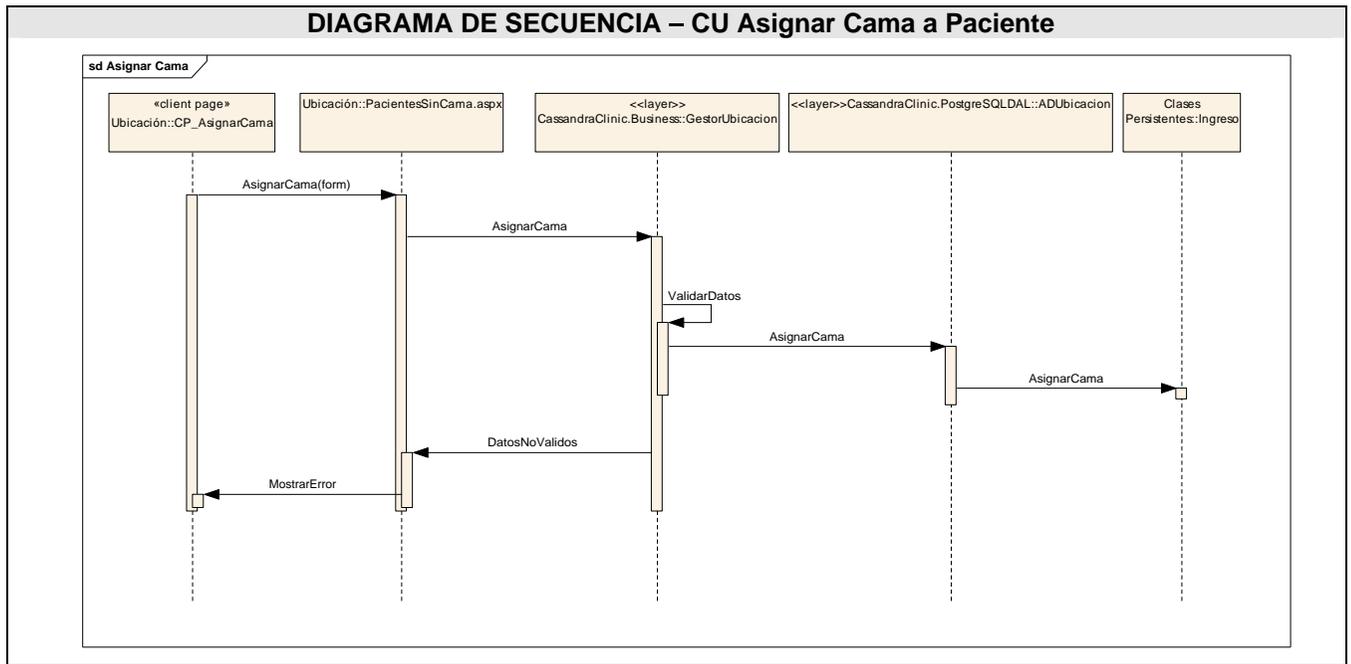
### 3.3.8 Diagrama de secuencia – Mostrar Imágenes Médicas



### 3.3.9 Diagrama de secuencia – Mostrar Reportes Médicos



### 3.3.10 Diagrama de secuencia – Asignar Cama a Paciente



## Anexo 5 Descripción de las clases

### 3.4.1 Clases entidades

<b>Nombre:</b> Paciente	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
citas	Cita
datosSocioEconomicosPaciente	DatosSocioEconomicos
direccionActual	string
entidadFederal	Estado
estadoResidencia	Estado
idPaciente	int
ingresos	Ingreso
institucionReferidora	InstitucionReferidora
medicoReferidorPaciente	Medico
municipio	Municipio
numeroHijo	int
numeroHistoria	string
religion	Religion
representanteLegal	RepresentanteLegal
rutaFoto	string
<b>Nombre:</b>	Citas()
<b>Descripción:</b>	Citas del paciente.
<b>Nombre:</b>	DatosRequeridosPresentes()
<b>Descripción:</b>	Verifica que todos los datos requeridos estén presentes.
<b>Nombre:</b>	DatosSocioEconomicosPaciente()
<b>Descripción:</b>	Datos socioeconómicos del paciente.
<b>Nombre:</b>	DireccionActual()
<b>Descripción:</b>	Dirección actual del paciente
<b>Nombre:</b>	EntidadFederal()
<b>Descripción:</b>	Entidad federal del paciente
<b>Nombre:</b>	EstadoResidencia()
<b>Descripción:</b>	Estado en el que reside actualmente el paciente.
<b>Nombre:</b>	EstaIngresado()
<b>Descripción:</b>	Retorna True si el paciente está ingresado.
<b>Nombre:</b>	IdPaciente()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	IngresoActual()
<b>Descripción:</b>	Ingreso actual del paciente.
<b>Nombre:</b>	Ingresos()
<b>Descripción:</b>	Ingresos del paciente.

<b>Nombre:</b>	InstitucionReferidora()
<b>Descripción:</b>	Institución referidora del paciente
<b>Nombre:</b>	MedicoReferidorPaciente()
<b>Descripción:</b>	Médico referidor del paciente.
<b>Nombre:</b>	MunicipioResidencia()
<b>Descripción:</b>	Municipio del paciente
<b>Nombre:</b>	NumeroHijo()
<b>Descripción:</b>	Representa la posición en la lista de hijos de los padres del paciente Se usa para generar el número de historia clínica.
<b>Nombre:</b>	NumeroHistoria()
<b>Descripción:</b>	Numero de Historia del paciente
<b>Nombre:</b>	Paciente()
<b>Descripción:</b>	Inicializa una nueva instancia de Paciente con los valores por defecto.
<b>Nombre:</b>	Paciente(int idPersona,int idPaciente,string nombre,string apellidos,Sexo sexo,string cedulaIdentidad,Nacionalidad nacionalidad,DateTime fechaNacimiento,Estado estadoNacimiento,EstudiosRealizados estudiosRealizados,RepresentanteLegal representanteLegal,string numeroHistoria,string direccionActual,string rutaFoto,Religion religion,InstitucionReferidora institucionReferidora,Estado entidadFederal,Municipio municipio,Cita[] citas,Ingreso[] ingresos,)
<b>Descripción:</b>	Sobrecargado. Inicializa una nueva instancia de Paciente con los valores especificados.
<b>Nombre:</b>	ReligionPaciente()
<b>Descripción:</b>	Religión del paciente
<b>Nombre:</b>	RepresentanteLegal()
<b>Descripción:</b>	Representante legal del paciente
<b>Nombre:</b>	RutaFoto()
<b>Descripción:</b>	Ruta de la foto del paciente
<b>Nombre:</b>	TieneCamaAsignada()
<b>Descripción:</b>	Retorna True si el paciente esta ingresado y tiene cama asignada.

<b>Nombre:</b> Estado	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idEstado	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Estado()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Municipio con los valores por defecto.
<b>Nombre:</b>	Estado(int idEstado,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Estado con los valores suministrados.
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdEstado()
<b>Descripción:</b>	

<b>Nombre:</b>	ToString()
<b>Descripción:</b>	
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> Medico	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
apellidos	string
cedulaidentidad	string
citas	Cita
colegio médico	Colegio médico
especialidad médico	Especialidad
id médico	int
institucionRefereridora	InstitucionReferidora
mms	string
nombre	string
<b>Nombre:</b>	Apellidos()
<b>Descripción:</b>	Apellidos del médico
<b>Nombre:</b>	Cedulaidentidad()
<b>Descripción:</b>	Cedula de identidad del médico
<b>Nombre:</b>	Citas()
<b>Descripción:</b>	Lista de citas que el medico tiene para atender.
<b>Nombre:</b>	ColegioMedico()
<b>Descripción:</b>	Colegio medico del médico
<b>Nombre:</b>	EspecialidadMedico()
<b>Descripción:</b>	Especialidad del médico
<b>Nombre:</b>	IdMedico()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	InstitucionRefereridoraPaciente()
<b>Descripción:</b>	Institución referidora a la que pertenece el medico
<b>Nombre:</b>	Medico()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Parentesco con los valores por defecto.
<b>Nombre:</b>	Mms()
<b>Descripción:</b>	MMS.
<b>Nombre:</b>	Nombre()
<b>Descripción:</b>	Nombre del médico.
<b>Nombre:</b>	NombreCompleto()
<b>Descripción:</b>	Nombre completo del medico Nombre + Apellidos
<b>Nombre:</b>	ToString()
<b>Descripción:</b>	

<b>Nombre:</b> Municipio	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>

idMunicipio	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdMunicipio()
<b>Descripción:</b>	
<b>Nombre:</b>	Municipio()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Municipio con los valores por defecto.
<b>Nombre:</b>	Municipio(int idMunicipio,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Municipio con los valores suministrados.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> Religion	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idReligion	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdReligion()
<b>Descripción:</b>	
<b>Nombre:</b>	Religion()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores por defecto.
<b>Nombre:</b>	Religion(int idReligion,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Religion con los valores suministrados.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> RepresentanteLegal	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
celular	string
estadoCivil	EstadoCivil
idRepresentanteLegal	int
ocupacion	Ocupacion
parentesco	Parentesco
telefono	string

telefonoEmergencia	string
<b>Nombre:</b>	Celular()
<b>Descripción:</b>	
<b>Nombre:</b>	EstadoCivilRepresentate()
<b>Descripción:</b>	
<b>Nombre:</b>	IdRepresentanteLegal()
<b>Descripción:</b>	
<b>Nombre:</b>	OcupacionRepLegal()
<b>Descripción:</b>	
<b>Nombre:</b>	ParentescoRepresentante()
<b>Descripción:</b>	
<b>Nombre:</b>	RepresentanteLegal()
<b>Descripción:</b>	Constructor. Inicializa una instancia de Representante Legal con los valores por defecto.
<b>Nombre:</b>	RepresentanteLegal(int idPersona,int idRepresentanteLegal,string nombre,string apellidos,Sexo sexo,string cedulaIdentidad,Nacionalidad nacionalidad,DateTime fechaNacimiento,Estado lugarNacimiento,EstudiosRealizados estudiosRealizados,Parentesco parentesco,Ocupacion ocupacion,string telefono,string celular,string telefonoEmergencia,EstadoCivil estadoCivil,)
<b>Descripción:</b>	
<b>Nombre:</b>	Telefono()
<b>Descripción:</b>	
<b>Nombre:</b>	TelefonoEmergencia()
<b>Descripción:</b>	

<b>Nombre:</b> Sexo	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idSexo	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdSexo()
<b>Descripción:</b>	
<b>Nombre:</b>	Sexo()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores por defectos.
<b>Nombre:</b>	Sexo(int idSexo,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores suministrados.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> EstudiosRealizados
<b>Clase Entidad</b>

<b>Atributo</b>		<b>Tipo</b>
idEstudiosRealizados		int
valor		string
<b>Nombre:</b>	EsNulo()	
<b>Descripción:</b>		
<b>Nombre:</b>	EstudiosRealizados()	
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de EstudiosRealizados con los valores por defecto.	
<b>Nombre:</b>	EstudiosRealizados(int idEstudiosRealizados,string valor,)	
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de EstudiosRealizados con los valores por defecto.	
<b>Nombre:</b>	Id()	
<b>Descripción:</b>		
<b>Nombre:</b>	IdEstudiosRealizados()	
<b>Descripción:</b>		
<b>Nombre:</b>	Valor()	
<b>Descripción:</b>		

<b>Nombre:</b> Nacionalidad		
<b>Clase Entidad</b>		
<b>Atributo</b>		<b>Tipo</b>
idNacionalidad		int
valor		string
<b>Nombre:</b>	EsNulo()	
<b>Descripción:</b>		
<b>Nombre:</b>	Id()	
<b>Descripción:</b>		
<b>Nombre:</b>	IdNacionalidad()	
<b>Descripción:</b>	Id.	
<b>Nombre:</b>	Nacionalidad()	
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores por defectos.	
<b>Nombre:</b>	Nacionalidad(int idNacionalidad,string valor,)	
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Nacionalidad con los valores suministrados.	
<b>Nombre:</b>	Valor()	
<b>Descripción:</b>	Valor posible.	

<b>Nombre:</b> Parentesco		
<b>Clase Entidad</b>		
<b>Atributo</b>		<b>Tipo</b>
idParentesco		int
valor		string
<b>Nombre:</b>	EsNulo()	
<b>Descripción:</b>		
<b>Nombre:</b>	Id()	
<b>Descripción:</b>		

<b>Nombre:</b>	IdParentesco()
<b>Descripción:</b>	Id
<b>Nombre:</b>	Parentesco()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Parentesco con los valores por defecto.
<b>Nombre:</b>	Parentesco(int idParentesco,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Parentesco con los valores suministrados.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Valor

<b>Nombre:</b> EstadoCivil	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idEstadoCivil	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	EstadoCivil()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de EstadoCivil con los valores por defecto.
<b>Nombre:</b>	EstadoCivil(int idEstadoCivil,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de EstadoCivil con los valores suministrados.
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdEstadoCivil()
<b>Descripción:</b>	
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> Persona	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
apellidos	string
cedulaIdentidad	string
estadoNacimiento	Estado
estudiosRealizados	EstudiosRealizados
fechaNacimiento	Date Time
idPersona	int
nacionalidad	Nacionalidad
nombre	string
sexo	Sexo
<b>Nombre:</b>	Apellidos()
<b>Descripción:</b>	Apellidos
<b>Nombre:</b>	CedulaIdentidad()

<b>Descripción:</b>	Cedula de Identidad
<b>Nombre:</b>	Edad()
<b>Descripción:</b>	Obtiene la edad de un paciente
<b>Nombre:</b>	EstadoNacimientoPersona()
<b>Descripción:</b>	Lugar de nacimiento
<b>Nombre:</b>	EstudiosRealizadosPaciente()
<b>Descripción:</b>	Estudios realizados
<b>Nombre:</b>	FechaNacimiento()
<b>Descripción:</b>	Fecha de nacimiento
<b>Nombre:</b>	IdPersona()
<b>Descripción:</b>	Id
<b>Nombre:</b>	NacionalidadPersona()
<b>Descripción:</b>	Nacionalidad
<b>Nombre:</b>	Nombre()
<b>Descripción:</b>	Nombre
<b>Nombre:</b>	NombreCompleto()
<b>Descripción:</b>	Nombre completo. Unión de nombre y apellidos separados por espacio
<b>Nombre:</b>	Persona()
<b>Descripción:</b>	Inicializa una nueva instancia de Persona con los valores por defecto.
<b>Nombre:</b>	Persona(int idPersona,string nombre,string apellidos,Sexo sexo,string cedulaIdentidad,Nacionalidad nacionalidad,DateTime fechaNacimiento,Estado estadoNacimiento,EstudiosRealizados estudiosRealizados,)
<b>Descripción:</b>	Sobrecargado. Inicializa una nueva instancia de Persona con los valores suministrados.
<b>Nombre:</b>	SexoPersona()
<b>Descripción:</b>	Sexo

<b>Nombre:</b> DatosSocioEconomicos	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
comiteSaludPaciente	ComiteSalud
condicionesAlojamientoPaciente	CondicionesAlojamiento
consejoComunalPaciente	ConsejoComunal
id	int
misionBarrioAdentro	bool
misionMadreBarrio	bool
misionMilagro	bool
misionNegraHipolita	bool
misionRivas	bool
misionRobinson	bool
misionSucre	bool
misionVuelvanCaras	bool
nivelInstruccionMadrePaciente	NivelInstruccionMadre
principalFuenteIngresosPaciente	PrincipalFuenteIngreso

profesionJefeFamiliaPaciente		ProfesionJefeFamilia
<b>Nombre:</b>	ComiteSaludPaciente()	
<b>Descripción:</b>	Comité de salud del paciente.	
<b>Nombre:</b>	CondicionesAlojamientoPaciente()	
<b>Descripción:</b>	Condiciones de alojamiento del paciente.	
<b>Nombre:</b>	ConsejoComunalPaciente()	
<b>Descripción:</b>	Consejo Comunal del paciente.	
<b>Nombre:</b>	DatosSocioEconomicos()	
<b>Descripción:</b>	Inicializa una nueva instancia de DatosSocioEconomicos con los valores por defecto.	
<b>Nombre:</b>	DatosSocioEconomicos(int id,bool misionRobinson,bool misionRivas,bool misionSucre,bool misionVuelvanCaras,bool misionNegraHipolita,bool misionBarrioAdentro,bool misionMilagro,bool misionMadreBarrio,ComiteSalud comiteSaludPaciente,ConsejoComunal consejoComunalPaciente,ProfesionJefeFamilia profesionJefeFamiliaPaciente,NivellInstruccionMadre nivellInstruccionMadrePaciente,PrincipalFuenteIngreso principalFuenteIngresosPaciente,CondicionesAlojamiento condicionesAlojamientoPaciente,)	
<b>Descripción:</b>	Sobrecargado. Inicializa una nueva instancia de Paciente con los valores especificados.	
<b>Nombre:</b>	Id()	
<b>Descripción:</b>	Id.	
<b>Nombre:</b>	MisionBarrioAdentro()	
<b>Descripción:</b>	Integración a la Misión Barrio Adentro de al menos un miembro de la familia.	
<b>Nombre:</b>	MisionMadreBarrio()	
<b>Descripción:</b>	Integración a la Misión Madre Barrio de al menos un miembro de la familia.	
<b>Nombre:</b>	MisionMilagro()	
<b>Descripción:</b>	Integración a la Misión Milagro de al menos un miembro de la familia.	
<b>Nombre:</b>	MisionNegraHipolita()	
<b>Descripción:</b>	Integración a la Misión Negra Hipólita de al menos un miembro de la familia.	
<b>Nombre:</b>	MisionRivas()	
<b>Descripción:</b>	Integración a la misión Rivas de al menos un miembro de la familia.	
<b>Nombre:</b>	MisionRobinson()	
<b>Descripción:</b>	Integración a la misión Robinson de al menos un miembro de la familia.	
<b>Nombre:</b>	MisionSucre()	
<b>Descripción:</b>	Integración a la Misión Sucre de al menos un miembro de la familia.	
<b>Nombre:</b>	MisionVuelvanCaras()	
<b>Descripción:</b>	Integración a la Misión Vuelva Caras de al menos un miembro de la familia.	
<b>Nombre:</b>	NivellInstruccionMadrePaciente()	
<b>Descripción:</b>	Nivel de instrucción de la madre del paciente.	
<b>Nombre:</b>	PrincipalFuenteIngresosPaciente()	
<b>Descripción:</b>	Principal fuente de ingresos del paciente.	
<b>Nombre:</b>	ProfesionJefeFamiliaPaciente()	
<b>Descripción:</b>	Profesión del jefe de familia del paciente.	

<b>Nombre:</b> ComiteSalud	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>

idComiteSalud	int
valor	string
<b>Nombre:</b>	ComiteSalud()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de ComiteSalud con los valores por defectos.
<b>Nombre:</b>	ComiteSalud(int idComiteSalud,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de ComiteSalud con los valores suministrados.
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdComiteSalud()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Nombre del Comité de Salud.

<b>Nombre:</b> CondicionesAlojamiento	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idCondicionesAlojamiento	int
valor	string
<b>Nombre:</b>	CondicionesAlojamiento()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores por defectos.
<b>Nombre:</b>	CondicionesAlojamiento(int idCondicionesAlojamiento,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores suministrados.
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdCondicionesAlojamiento()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Valor posible de sexo.

<b>Nombre:</b> ConsejoComunal	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idConsejoComunal	int
valor	string
<b>Nombre:</b>	ConsejoComunal()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de ConsejoComunal con los valores por defectos.
<b>Nombre:</b>	ConsejoComunal(int idConsejoComunal,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de ConsejoComunal con los

	valores suministrados.
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdConsejoComunal()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Valor posible de sexo.

<b>Nombre:</b> NivelInstruccionMadre	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idNivelInstruccionMadre	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdNivelInstruccionMadre()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	NivelInstruccionMadre()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores por defectos.
<b>Nombre:</b>	NivelInstruccionMadre(int idNivelInstruccionMadre,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores suministrados.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Valor posible de sexo.

<b>Nombre:</b> PrincipalFuenteIngreso	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idPrincipalFuenteIngreso	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdPrincipalFuenteIngreso()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	PrincipalFuenteIngreso()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores por defectos.
<b>Nombre:</b>	PrincipalFuenteIngreso(int idPrincipalFuenteIngreso,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Sexo con los valores suministrados.
<b>Nombre:</b>	Valor()

<b>Descripción:</b>	Valor posible de sexo.
---------------------	------------------------

<b>Nombre:</b> ProfesionJefeFamilia	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idProfesionJefeFamilia	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdProfesionJefeFamilia()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	ProfesionJefeFamilia()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de ProfesionJefeFamilia con los valores por defectos.
<b>Nombre:</b>	ProfesionJefeFamilia(int idProfesionJefeFamilia,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de ProfesionJefeFamilia con los valores suministrados.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Valor posible de sexo.

<b>Nombre:</b> Ingreso	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
ceEm	CEEM
diagnosticoAdmision	Diagnostico
egreso	Egreso
enfermedadActual	string
fechaIngreso	DateTime
hojasHistoriaClinica	HojaHistoriaClinica
idIngreso	int
localPermanente	LocalIngreso
localTemporal	LocalIngreso
medicoIngreso	Medico
medicoReferidor	Medico
motivos	string
<b>Nombre:</b>	CeEm()
<b>Descripción:</b>	Si el ingreso es por CE o EM
<b>Nombre:</b>	DatosRequeridosPresentes()
<b>Descripción:</b>	Se asegura de que los datos que son obligatorios en el ingreso estén presentes.
<b>Nombre:</b>	DiagnosticoAdmision()
<b>Descripción:</b>	Diagnóstico de admisión del paciente en el momento del ingreso.
<b>Nombre:</b>	EgresoCorrespondiente()

<b>Descripción:</b>	Egreso correspondiente y que pone fin al ingreso actual.
<b>Nombre:</b>	EnfermedadActual()
<b>Descripción:</b>	Historia de la enfermedad actual.
<b>Nombre:</b>	FechaIngreso()
<b>Descripción:</b>	Fecha y hora del ingreso.
<b>Nombre:</b>	Historial()
<b>Descripción:</b>	Hojas de la historia clínica correspondientes a este ingreso
<b>Nombre:</b>	IdIngreso()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	Ingreso()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Ingreso con los valores por defecto.
<b>Nombre:</b>	Ingreso(int idIngreso,CEEM ceEm,string motivos,string enfermedadActual,Diagnostico diagnosticoAdmision,DateTime fechaIngreso,Egreso egreso,Medico medicoIng,HojaHistoriaClinica[] historial,Medico medicoReferidor,LocalIngreso localPermanente,LocalIngreso localTemporal,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Ingreso con los valores suministrados
<b>Nombre:</b>	LocalPermanente()
<b>Descripción:</b>	Local permanente de ingreso del paciente
<b>Nombre:</b>	LocalTemporal()
<b>Descripción:</b>	Local temporal de que se le asigna al paciente confines clínicos
<b>Nombre:</b>	MedicoIngreso()
<b>Descripción:</b>	Médico que ingresa
<b>Nombre:</b>	MedicoReferidor()
<b>Descripción:</b>	Médico que ingresa
<b>Nombre:</b>	Motivos()
<b>Descripción:</b>	Motivos de ingreso
<b>Nombre:</b>	UltimaHojaHistoria()
<b>Descripción:</b>	Hoja de historia clínica actual

<b>Nombre:</b> Egreso	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
diagnosticoEgreso	string
fechaEgreso	DateTime
idEgreso	int
indicaciones	string
medicoEgreso	Medico
motivoEgreso	MotivoEgreso
resumenEvolucion	string
resumenTratamiento	string
<b>Nombre:</b>	DatosRequeridosPresentes()
<b>Descripción:</b>	
<b>Nombre:</b>	DiagnosticoEgreso()
<b>Descripción:</b>	

<b>Nombre:</b>	Egreso()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Egreso con los valores por defecto.
<b>Nombre:</b>	Egreso(int idEgreso,DateTime fechaEgreso,string resumenEvolucion,string diagnosticoEgreso,string resumenTratamiento,string indicaciones,MotivoEgreso ME,Medico medicoEgreso,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Egreso con los valores suministrados.
<b>Nombre:</b>	FechaEgreso()
<b>Descripción:</b>	
<b>Nombre:</b>	IdEgreso()
<b>Descripción:</b>	
<b>Nombre:</b>	Indicaciones()
<b>Descripción:</b>	
<b>Nombre:</b>	MedicoEgreso()
<b>Descripción:</b>	
<b>Nombre:</b>	MotivoEg()
<b>Descripción:</b>	
<b>Nombre:</b>	ResumenEvolucion()
<b>Descripción:</b>	
<b>Nombre:</b>	ResumenTratamiento()
<b>Descripción:</b>	

<b>Nombre:</b> CEEM	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
ídem	int
valor	string
<b>Nombre:</b>	CEEM()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de CEEM con los valores por defecto.
<b>Nombre:</b>	CEEM(int idCEEM,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de CEEM con los valores suministrados.
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdCEEM()
<b>Descripción:</b>	
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> Diagnostico	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
cie	string
idDiagnostico	int

valor	string
<b>Nombre:</b>	CIE()
<b>Descripción:</b>	Código del Calificador Internacional de Enfermedades del diagnostico
<b>Nombre:</b>	Diagnostico()
<b>Descripción:</b>	
<b>Nombre:</b>	Diagnostico(int IdDiag,string Diag,string Cie,)
<b>Descripción:</b>	
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	Id_CIE()
<b>Descripción:</b>	Cadena resultante de concatenar el id mas el código CIE-10
<b>Nombre:</b>	IdDiagnostico()
<b>Descripción:</b>	Id del diagnostico en la base de datos
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Diagnostico

<b>Nombre:</b> LocalIngreso	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
camaLocalIngreso	Cama
idLocalIngreso	int
ubicacion	Ubicacion
valor	string
<b>Nombre:</b>	CamaLocalIngreso()
<b>Descripción:</b>	Una cama que se encuentra en el local.
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdLocalIngreso()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	LocalIngreso()
<b>Descripción:</b>	Constructor. Inicializa una instancia de LocalIngreso con los valores por defecto.
<b>Nombre:</b>	LocalIngreso(int idLocalIngreso,string valor,Ubicacion ubicacion,Cama cama,)
<b>Descripción:</b>	Sobrecargado. Inicializa una instancia de LocalIngreso con los valores suministrados
<b>Nombre:</b>	ToString()
<b>Descripción:</b>	
<b>Nombre:</b>	UbicacionLocalIngreso()
<b>Descripción:</b>	Ubicacion del local de ingreso
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Nombre del local.

<b>Nombre:</b> Cama	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idCama	int
valor	string
<b>Nombre:</b>	Cama()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Cama con los valores por defectos.
<b>Nombre:</b>	Cama(int idCama,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Cama con los valores suministrados.
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdCama()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	ToString()
<b>Descripción:</b>	
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Valor posible.

<b>Nombre:</b> Ala	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idAla	int
valor	string
<b>Nombre:</b>	Ala()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Ala con los valores por defectos.
<b>Nombre:</b>	Ala(int idAla,string valor,)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Ala con los valores suministrados.
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdAla()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	ToString()
<b>Descripción:</b>	
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Valor posible.

<b>Nombre:</b> Ubicacion	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>

ala	Ala
idUbicacion	int
nivel	Nivel
valor	string
<b>Nombre:</b>	AlaUbicacion()
<b>Descripción:</b>	Ala de la ubicacion
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdUbicacion()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	NivelUbicacion()
<b>Descripción:</b>	Nivel de la ubicacion
<b>Nombre:</b>	ToString()
<b>Descripción:</b>	
<b>Nombre:</b>	Ubicacion()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Ubicacion con los valores por defectos.
<b>Nombre:</b>	Ubicacion(int idUbicacion,string valor,Nivel nivel,Ala ala)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Ubicacion con los valores suministrados.
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Valor posible.

<b>Nombre:</b> Nivel	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idNivel	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdNivel()
<b>Descripción:</b>	Id.
<b>Nombre:</b>	Nivel()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Ala con los valores por defectos.
<b>Nombre:</b>	Nivel(int idNivel,string valor)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de Ala con los valores suministrados.
<b>Nombre:</b>	ToString()
<b>Descripción:</b>	
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	Nombre de Nivel

<b>Nombre:</b> MotivoEgreso	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idMotivoEgreso	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdMotivoEgreso()
<b>Descripción:</b>	
<b>Nombre:</b>	MotivoEgreso()
<b>Descripción:</b>	Inicializa un objeto MotivoEgreso con los valores por defecto
<b>Nombre:</b>	MotivoEgreso(int IdME,string valor)
<b>Descripción:</b>	Inicializa un objeto MotivoEgreso con valores suministrados
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> HojaClinica	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
estadoPaciente	EstadoPaciente
fechaCreacion	DateTime
idHojaClinica	int
observaciones	string
remisionEntrada	RemisionEntradaSalida
remisionSalida	RemisionEntradaSalida
tipoHojaClinica	TipoHojaClinica
<b>Nombre:</b>	EstadoPaciente()
<b>Descripción:</b>	
<b>Nombre:</b>	FechaCreacion()
<b>Descripción:</b>	
<b>Nombre:</b>	HojaClinica()
<b>Descripción:</b>	
<b>Nombre:</b>	HojaClinica(int idHojaClinica,DateTime fechaCreacion,EstadoPaciente estadoPaciente,string observaciones,RemisionEntradaSalida remisionEntrada,RemisionEntradaSalida remisionSalida,TipoHojaClinica tipoHojaClinica,)
<b>Descripción:</b>	
<b>Nombre:</b>	IdHojaClinica()
<b>Descripción:</b>	
<b>Nombre:</b>	Observaciones()
<b>Descripción:</b>	
<b>Nombre:</b>	RemisionEntrada()
<b>Descripción:</b>	
<b>Nombre:</b>	RemisionSalida()

<b>Descripción:</b>	
<b>Nombre:</b>	TipoHojaClinica()
<b>Descripción:</b>	

<b>Nombre:</b> RemisionEntradaSalida	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idRemision	int
remision	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	RemisionEntradaSalida()
<b>Descripción:</b>	
<b>Nombre:</b>	RemisionEntradaSalida(int idRemision,string remision)
<b>Descripción:</b>	
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> EstadoPaciente	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idEstado	int
valor	string
<b>Nombre:</b>	EsNulo()
<b>Descripción:</b>	
<b>Nombre:</b>	EstadoPaciente()
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de EstadoPaciente con los valores por defecto.
<b>Nombre:</b>	EstadoPaciente(int idEstado,string valor)
<b>Descripción:</b>	Constructor. Inicializa una nueva instancia de EstadoPaciente con los valores suministrados.
<b>Nombre:</b>	Id()
<b>Descripción:</b>	
<b>Nombre:</b>	IdEstado()
<b>Descripción:</b>	
<b>Nombre:</b>	Valor()
<b>Descripción:</b>	

<b>Nombre:</b> OrdenEstudio	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
buscarEspecif	string
diagnosticos	Diagnostico
peso	double

signosSintomas	string
talla	double
<b>Nombre:</b>	BuscarEspecif()
<b>Descripción:</b>	
<b>Nombre:</b>	Diagnosticos()
<b>Descripción:</b>	
<b>Nombre:</b>	OrdenEstudio()
<b>Descripción:</b>	
<b>Nombre:</b>	OrdenEstudio(int idHojaClinica,DateTime fechaCreacion,EstadoPaciente estadoPaciente,string observaciones,RemisionEntradaSalida remisionEntrada,RemisionEntradaSalida remisionSalida,TipoHojaClinica tipoHojaClinica,string buscarEspecif,string signosSintomas,double peso,double talla,Diagnostico[] diagnosticos)
<b>Descripción:</b>	
<b>Nombre:</b>	Peso()
<b>Descripción:</b>	
<b>Nombre:</b>	SignosSintomas()
<b>Descripción:</b>	
<b>Nombre:</b>	Talla()
<b>Descripción:</b>	

<b>Nombre:</b> OrdenMedica	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	OrdenMedica()
<b>Descripción:</b>	
<b>Nombre:</b>	OrdenMedica(int idHojaClinica,DateTime fechaCreacion,EstadoPaciente estadoPaciente,string observaciones,RemisionEntradaSalida remisionEntrada,RemisionEntradaSalida remisionSalida,TipoHojaClinica tipoHojaClinica,)
<b>Descripción:</b>	

<b>Nombre:</b> ImagenXWeb	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
idImagen	int
imagen	string
<b>Nombre:</b>	IdImagen()
<b>Descripción:</b>	
<b>Nombre:</b>	Imagen()
<b>Descripción:</b>	
<b>Nombre:</b>	ImagenEstudio()
<b>Descripción:</b>	
<b>Nombre:</b>	ImagenEstudio(int idImagen,string imagen)
<b>Descripción:</b>	

<b>Nombre:</b> ReporteXWeb	
<b>Clase Entidad</b>	

<b>Atributo</b>		<b>Tipo</b>
cieModalidad		string
cieParte		string
cieVersion		string
comentarios		string
fechaResultado		string
id		string
id_Paciente		string
medicamentos		string
medico		string
recomendacionMedica		string
study_UID		string
<b>Nombre:</b>	CieModalidad()	
<b>Descripción:</b>		
<b>Nombre:</b>	CieParte()	
<b>Descripción:</b>		
<b>Nombre:</b>	CieVersion()	
<b>Descripción:</b>		
<b>Nombre:</b>	Comentarios()	
<b>Descripción:</b>		
<b>Nombre:</b>	FechaResultado()	
<b>Descripción:</b>		
<b>Nombre:</b>	Id()	
<b>Descripción:</b>		
<b>Nombre:</b>	IdPaciente()	
<b>Descripción:</b>		
<b>Nombre:</b>	Medicamentos()	
<b>Descripción:</b>		
<b>Nombre:</b>	Medico()	
<b>Descripción:</b>		
<b>Nombre:</b>	RecomendacionMedica()	
<b>Descripción:</b>		
<b>Nombre:</b>	ReporteXWeb()	
<b>Descripción:</b>		
<b>Nombre:</b>	Study_UID()	
<b>Descripción:</b>		

### 3.4.2 Clases controladoras

<b>Nombre:</b> GestorAdmision		
<b>Clase Control</b>		
<b>Atributo</b>		<b>Tipo</b>
<b>Nombre:</b>	AdmitirPaciente(Paciente paciente)	
<b>Descripción:</b>	Inserta un paciente en la BD	

<b>Nombre:</b>	AdmitirPaciente(Paciente paciente,int idPaciente,)
<b>Descripción:</b>	Inserta un paciente en la BD y devuelve su id como un parámetro de salida
<b>Nombre:</b>	GenerarNumeroHistoria(string cedula,int numeroHijo,)
<b>Descripción:</b>	
<b>Nombre:</b>	InsertarDatosSocioEconomicos(Paciente paciente,)
<b>Descripción:</b>	Inserta los datos socioeconómicos al paciente específico
<b>Nombre:</b>	LanzarViolacionDeReglaDeNegocio(string message,object objetoInvolucrado,)
<b>Descripción:</b>	Lanza un evento de Violación de Regla de Negocio siempre que existan suscriptores del evento OnReglaDeNegocioViolada, de lo contrario, se lanza una excepción.
<b>Nombre:</b>	OnReglaDeNegocioViolada()
<b>Descripción:</b>	Evento estático que se lanza cuando se produce una violación de una regla del negocio.

<b>Nombre:</b> GestorBusqueda	
<b>Clase Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	BuscarMedicoInternoPorId(int idMedicoInterno,)
<b>Descripción:</b>	Busca un medico interno en la base de datos.
<b>Nombre:</b>	BuscarMedicoPorId(int idMedico,)
<b>Descripción:</b>	Busca un medico interno en la base de datos.
<b>Nombre:</b>	BuscarMedicoReferidorPorId(int idMedicoReferidor,)
<b>Descripción:</b>	Busca un medico referidor en la base de datos.
<b>Nombre:</b>	BuscarMedicosPorEspecialidad(int idEspecialidad,)
<b>Descripción:</b>	Busca los médicos por la especialidad dada
<b>Nombre:</b>	BuscarPacientePorId(int idPaciente,)
<b>Descripción:</b>	Busca un paciente por su Id en la base de datos.
<b>Nombre:</b>	BuscarPacientes(SearchQuery searchQuery,)
<b>Descripción:</b>	Busca los pacientes que coincidan con los criterios de búsqueda en searchQuery
<b>Nombre:</b>	BuscarPacientesIngresadosParaEgresar(SearchQuery searchQuery)
<b>Descripción:</b>	Busca los pacientes ingresados para egresarlos
<b>Nombre:</b>	BuscarPacientesIngresadosSinCama(SearchQuery searchQuery)
<b>Descripción:</b>	Extrae de la base de datos los pacientes ingresados sin camas asignadas
<b>Nombre:</b>	BuscarPacientesIngresadosSinCamaTemporal()
<b>Descripción:</b>	Busca los pacientes ingresados que tengan la cama permanente asignada pero no la temporal
<b>Nombre:</b>	BuscarPacientesParaIngresar(SearchQuery searchQuery)
<b>Descripción:</b>	Busca los pacientes que van a ser ingresados que coincidan con los criterios de búsqueda en searchQuery
<b>Nombre:</b>	BuscarPacientesPorUbicacion(int IdLocal,)
<b>Descripción:</b>	Busca los pacientes que se encuentran en una ubicación dada
<b>Nombre:</b>	BuscarPacientesSinDatosSocioEconomicos(SearchQuery sq,)
<b>Descripción:</b>	Busca los paciente pendientes para llenar datos socioeconómicos
<b>Nombre:</b>	ListaMedicos(TipoMedico tipoMedico,)
<b>Descripción:</b>	Extrae de la base de datos la lista de médicos internos

<b>Nombre:</b> GestorCitas
----------------------------

<b>Clase Control</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	Especialistas(int idEspecialidad,)
<b>Descripción:</b>	Devuelve los especialistas de una especialidad
<b>Nombre:</b>	Instituciones()
<b>Descripción:</b>	De vuelve las instituciones referidoras
<b>Nombre:</b>	Locales()
<b>Descripción:</b>	Devuelve los locales para las citas
<b>Nombre:</b>	MedicosDeInstitucion(int idInstitucion,)
<b>Descripción:</b>	Devuelve los médicos referidores
<b>Nombre:</b>	Servicios()
<b>Descripción:</b>	

<b>Nombre:GestorEgreso</b>	
<b>Clase Control</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	InsertarEgreso(Egreso egreso,int idPaciente,)
<b>Descripción:</b>	Inserta un egreso en la BD
<b>Nombre:</b>	LanzarViolacionDeReglaDeNegocio(string message,object objetoInvolucrado,)
<b>Descripción:</b>	Lanza un evento de Violación de Regla de Negocio siempre que existan suscriptores del evento OnReglaDeNegocioViolada, de lo contrario, se lanza una excepción.
<b>Nombre:</b>	OnReglaDeNegocioViolada()
<b>Descripción:</b>	

<b>Nombre:GestorHC</b>	
<b>Clase Control</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	GetAnnosHojasClinicas(int idPaciente,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetConsultaExternaHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetConsultaExternaPorID(int idConsulta,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetConsultaOftalmologiaHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetConsultaOftalmologiaPorID(int idConsulta,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetConsultaUrgenciaHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetConsultaUrgenciaPorID(int idConsulta,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetDiagnosticosPorPaciente(int idPaciente,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetExamenFisicoHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetExamenFisicoPorID(int idExamen,)

<b>Descripción:</b>	
<b>Nombre:</b>	GetHojasClinicas(int idPaciente,DateTime fecha1,DateTime fecha2,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetIdHistoriaGralPorPaciente(int idPaciente,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetMedicamentosPorPaciente(int idPaciente,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetMesesHojasClinicas(int idPaciente,int anno,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOperacionQuirurgicaHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOperacionQuirurgicaPorID(int idCirugia,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOrdenEcoCardiografiaHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOrdenEcocardiografiaPorID(int idOrden,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOrdenResonanciaHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOrdenResonanciaPorID(int idOrden,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOrdenTomografiaHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOrdenTomografiaPorID(int idOrden,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOrdenTratamientoHTML(int idHoja,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetOrdenTratamientoPorID(int idOrden,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetPesoTallaPorPaciente(int idPaciente,)
<b>Descripción:</b>	
<b>Nombre:</b>	InsertarConsultaExterna(ConsultaExterna consulta,int idHistoriaGral,)
<b>Descripción:</b>	
<b>Nombre:</b>	InsertarConsultaOftalmologia(ConsultaOftalmologia consulta,int idHistoriaGral,)
<b>Descripción:</b>	
<b>Nombre:</b>	InsertarConsultaUrgencia(ConsultaUrgencia consulta,int idHistoriaGral,)
<b>Descripción:</b>	
<b>Nombre:</b>	InsertarExamenFisico(ExamenFisico examen,int idHistoriaGral,)
<b>Descripción:</b>	Lanza un evento de Violación de Regla de Negocio siempre que existan suscriptores del evento OnReglaDeNegocioViolada, de lo contrario, se lanza una excepción.
<b>Nombre:</b>	InsertarOperacionQuirurgica(OperacionQuirurgica cirugia,int idHistoriaGral,)
<b>Descripción:</b>	
<b>Nombre:</b>	InsertarOrdenEcoCardiografia(OrdenEcocardiografia orden,int idHistoriaGral,)
<b>Descripción:</b>	
<b>Nombre:</b>	InsertarOrdenResonancia(OrdenResonancia orden,int idHistoriaGral,)
<b>Descripción:</b>	

<b>Nombre:</b>	InsertarOrdenTomografia(OrdenTomografia orden,int idHistoriaGral,)
<b>Descripción:</b>	
<b>Nombre:</b>	InsertarOrdenTratamiento(OrdenTratamiento orden,int idHistoriaGral,)
<b>Descripción:</b>	

<b>Nombre:</b> GestorIngreso	
<b>Clase Control</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	BuscarIngresoActualPaciente(int IdPaciente,)
<b>Descripción:</b>	Listar los ingresos de un paciente
<b>Nombre:</b>	BuscarPacientePorIngreso(int idIngreso,)
<b>Descripción:</b>	Busca el paciente al que le corresponde el ingreso dado
<b>Nombre:</b>	IngresarPaciente(Ingreso ingreso,int idPaciente,HojaHistoriaClinica HC,)
<b>Descripción:</b>	Insertar Ingreso en la BD retorna:
<b>Nombre:</b>	LanzarViolacionDeReglaDeNegocio(string message,object objetoInvolucrado,)
<b>Descripción:</b>	Lanza un evento de Violación de Regla de Negocio siempre que existan suscriptores del evento OnReglaDeNegocioViolada, de lo contrario, se lanza una excepción.
<b>Nombre:</b>	ListalIngresos()
<b>Descripción:</b>	Busca la lista de ingresos.
<b>Nombre:</b>	OnReglaDeNegocioViolada()
<b>Descripción:</b>	
<b>Nombre:</b>	SeleccionarIngresoPORId(int IdIngreso,)
<b>Descripción:</b>	Selecciona los datos de un ingreso

<b>Nombre:</b> GestorNomencladores	
<b>Clase Control</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	GetIdNomencladorByValor(string valorNomenclador,Type tipoNomenclador,)
<b>Descripción:</b>	Obtiene el id del nomenclador pasado como cadena.
<b>Nombre:</b>	GetNomencladorById(int idNomenclador,Type tipoNomenclador,)
<b>Descripción:</b>	
<b>Nombre:</b>	GetValorNomencladorById(int id,Type tipoNomenclador,)
<b>Descripción:</b>	Obtiene el id del nomenclador pasado como cadena.
<b>Nombre:</b>	InsertarDiagnosticosAsociados(Diagnostico[] diagnosticos,)
<b>Descripción:</b>	Inserta los diagnosticos asociados al paciente
<b>Nombre:</b>	InsertarInstitucionReferidora(string institucion,)
<b>Descripción:</b>	Inserta una nueva institucion referidora en la BD
<b>Nombre:</b>	ListaClasificacionCirugia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaCompresaCirugia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaDiagnosticoCirugia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaEstadoSaludPaciente()
<b>Descripción:</b>	

<b>Nombre:</b>	ListaLiquidosCirugia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaMedicamentos()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaMotivosCirugia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaRemisionEntrada()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaRemisionSalida()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaTipoAnestecia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaTipoCirugia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaTipoExamenResonancia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaTipoExamenTomografia()
<b>Descripción:</b>	
<b>Nombre:</b>	ListaValores_Ala()
<b>Descripción:</b>	Devuelve todos los valores del nomenclador Ala
<b>Nombre:</b>	ListaValores_CamaPorLocalIngreso(int idLocalIngreso,)
<b>Descripción:</b>	Devuelve la lista de camas que se encuentran en un nivel, un ala y un local de ingreso
<b>Nombre:</b>	ListaValores_CamasNoAsignadasPorLocalIngreso(int idLocalIngreso,)
<b>Descripción:</b>	Devuelve la lista de camas que se encuentran disponibles dentro de un local de ingreso en un nivel, un ala y un local de ingreso
<b>Nombre:</b>	ListaValores_CEEM()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_ColegioMedico()
<b>Descripción:</b>	Lista los colegios médicos.
<b>Nombre:</b>	ListaValores_ComiteSalud()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_CondicionesAlojamiento()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_ConsejoComunal()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_Diagnostico()
<b>Descripción:</b>	Devuelve los valores de los diagnosticos de CIE - 10.
<b>Nombre:</b>	ListaValores_DiagnosticoAsociado()
<b>Descripción:</b>	Devuelve los valores de los diagnosticos asociados de CIE - 10.
<b>Nombre:</b>	ListaValores_EntidadFederal()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_Especialidad()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_Estado()
<b>Descripción:</b>	Devuelve los Estados como una lista.
<b>Nombre:</b>	ListaValores_EstadoCita()
<b>Descripción:</b>	Devuelve los valores de los estados de una cita de paciente.
<b>Nombre:</b>	ListaValores_EstadoCivil()

<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_EstadoPaciente()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_EstudiosRealizados()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_InstitucionReferidora()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_LocallIngresoPorAlaYNivel(int idNivel,int idAla,)
<b>Descripción:</b>	Busca la lista de habitaciones de la base de datos dados los id de ala y nivel
<b>Nombre:</b>	ListaValores_LocallIngresoSinAsignarPorAlaYNivel(int idNivel,int idAla,)
<b>Descripción:</b>	Busca la lista de habitaciones que no han sido asignados de la base de datos dados los id de ala y nivel
<b>Nombre:</b>	ListaValores_Modalidad()
<b>Descripción:</b>	Devuelve la lista de modalidades de estudios
<b>Nombre:</b>	ListaValores_MotivoEgreso()
<b>Descripción:</b>	Devuelve la lista de egresos que se encuentran en la base de datos.
<b>Nombre:</b>	ListaValores_Municipio()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_MunicipioPorEstado(int idEstado,)
<b>Descripción:</b>	Retorna los municipios de un estado.
<b>Nombre:</b>	ListaValores_Nacionalidad()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_Nivel()
<b>Descripción:</b>	Obtiene la lista de niveles desde la bd
<b>Nombre:</b>	ListaValores_NivelInstruccionMadre()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_Ocupacion()
<b>Descripción:</b>	Devuelve las ocupaciones posibles del representante legal
<b>Nombre:</b>	ListaValores_Parentesco()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_PrincipalFuenteIngreso()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_ProfesionJefeFamilia()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_Religion()
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_Sexo(TipoDeEnlace tipoEnlace,)
<b>Descripción:</b>	Obtiene la lista de valores del nomenclador desde la capa de datos.
<b>Nombre:</b>	ListaValores_TipoEquipo()
<b>Descripción:</b>	Devuelve la lista de equipos de la base de datos.

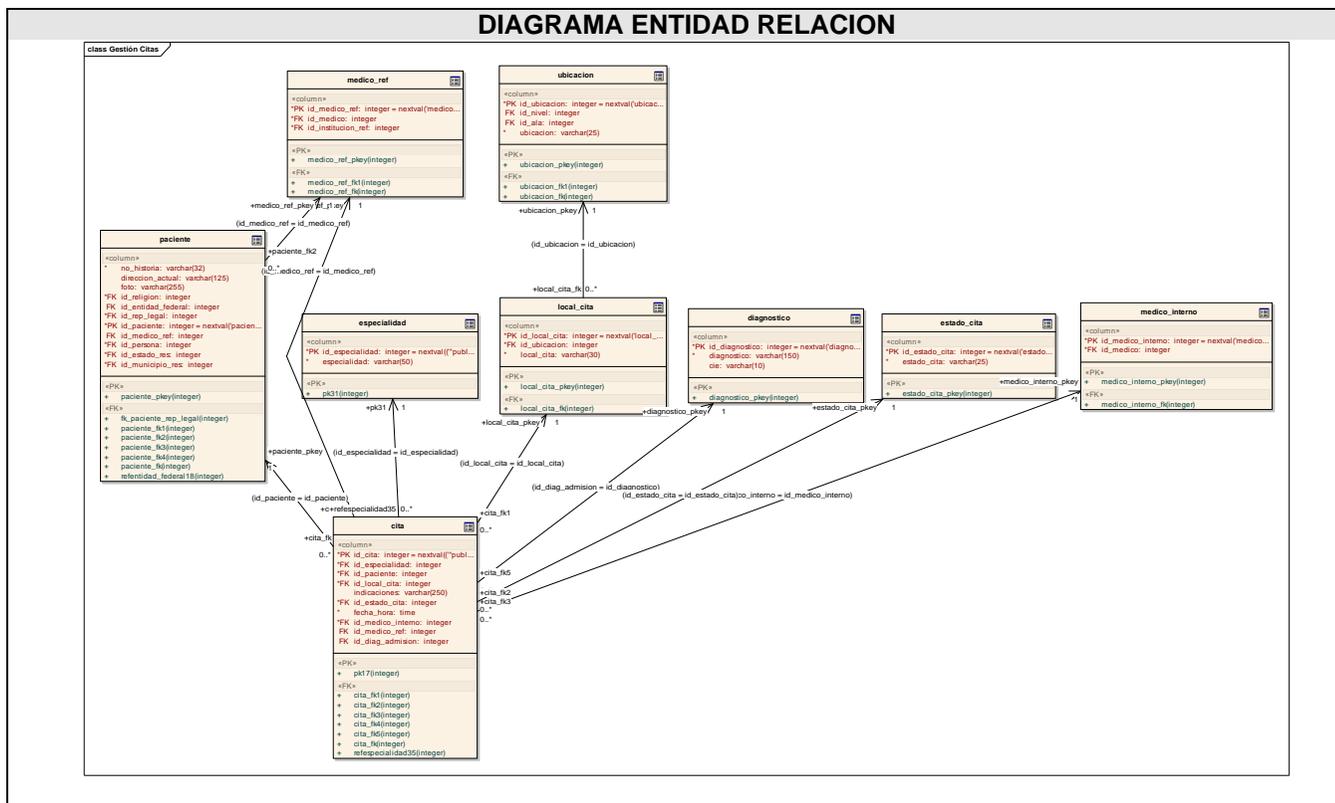
<b>Nombre:</b> GestorUbicacion	
<b>Clase Control</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	AsignarCama(int IdPaciente,int IdCama,)
<b>Descripción:</b>	Asignar una cama a un paciente ingresado
<b>Nombre:</b>	AsignarCamaTemp(int IdPaciente,int IdCamaTemp,)
<b>Descripción:</b>	Asignar Cama Temporal
<b>Nombre:</b>	LanzarViolacionDeReglaDeNegocio(string message,object

	objetoInvolucrado,)
<b>Descripción:</b>	Lanza un evento de Violación de Regla de Negocio siempre que existan suscriptores del evento OnReglaDeNegocioViolada, de lo contrario, se lanza una excepción.
<b>Nombre:</b>	LiberarCama(int IdPaciente,int IdCama,)
<b>Descripción:</b>	Liberar cama de paciente ingresado
<b>Nombre:</b>	LiberarCamaTemporal(int IdIngreso,)
<b>Descripción:</b>	Libera la cama temporal de un ingreso.
<b>Nombre:</b>	MoverCama(int IdPaciente,int IdCamaDestino,)
<b>Descripción:</b>	Mover un paciente de cama
<b>Nombre:</b>	MoverCamaTemporal(int IdPaciente,int IdCamaDestino,)
<b>Descripción:</b>	Mover un paciente de cama
<b>Nombre:</b>	OnReglaDeNegocioViolada()
<b>Descripción:</b>	
<b>Nombre:</b>	PacienteConDosCamas()
<b>Descripción:</b>	Devuelve los pacientes que están ingresados con dos camas asignadas
<b>Nombre:</b>	PacienteConSusUbicaciones()
<b>Descripción:</b>	Devuelve los pacientes que están ingresados con sus dos ubicaciones, la permanente y la temporal

<b>Nombre:</b> GestorXWeb	
<b>Clase Control</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	GetHTMLImágenes(int idPaciente,)
<b>Descripción:</b>	Devuelve una cadena en forma de HTML con el código para representar las imágenes.
<b>Nombre:</b>	GetReportes(int idPaciente,)
<b>Descripción:</b>	Devuelve una lista de reportes desde el WebService de XWeb

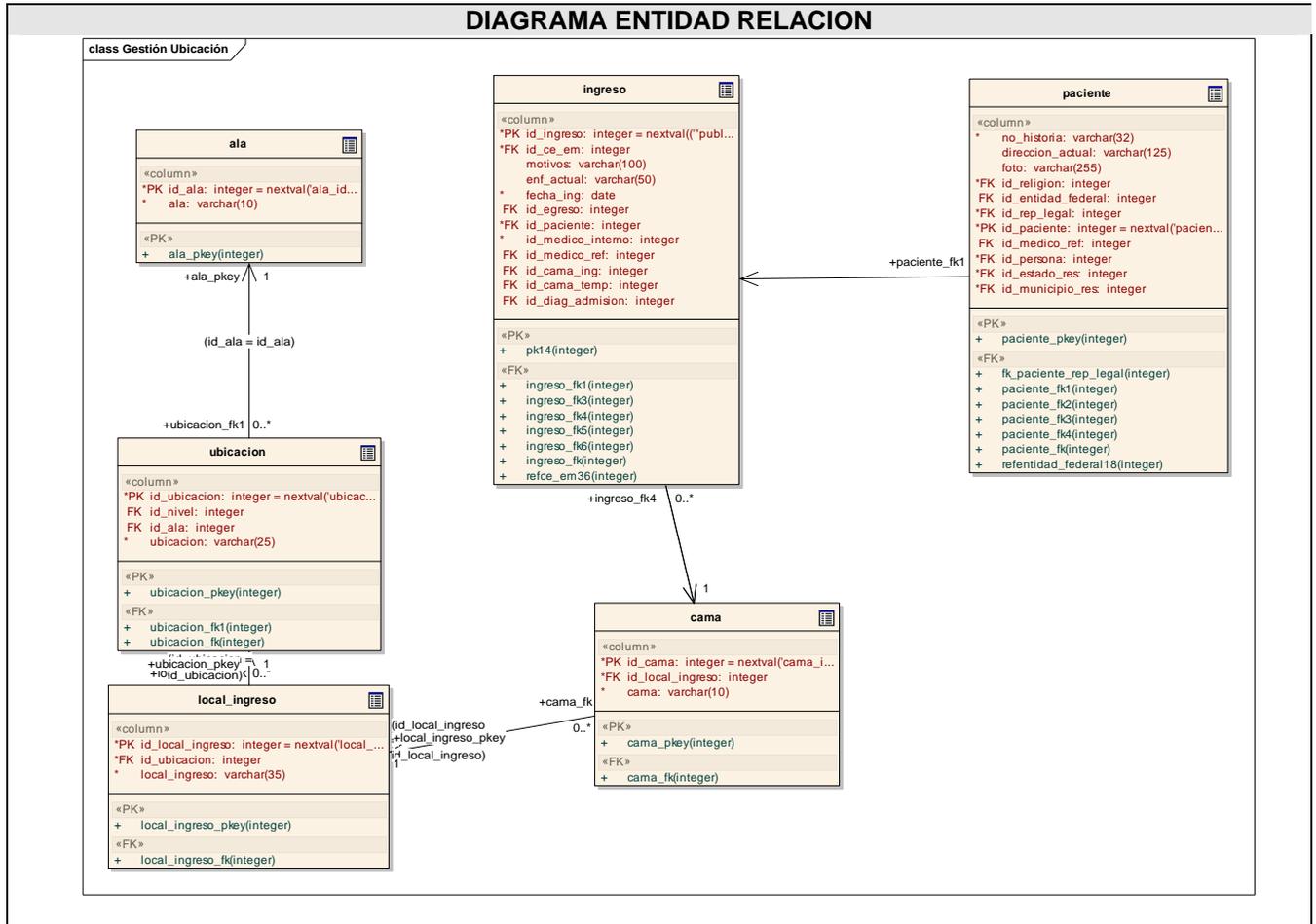
## Anexo 6 Diagramas Entidad – Relación.

### 3.5.1 Diagrama Entidad Relación - Paquete Gestión de Citas





### 3.5.3 Diagrama Entidad Relación - Paquete Gestión de Ubicación



### Anexo 7 Descripción de las tablas de la Base de Datos.

#### 3.6 Descripción de las tablas.

Nombre: ala (Nomenclador)		
Descripción: Almacena las alas de una institución (norte, sur, este, oeste)		
Atributo	Tipo	Descripción
id_ala	integer	
ala	varchar	

<b>Nombre: cama (Nomenclador)</b>		
<b>Descripción:</b> Almacena las camas de los locales de una institución hospitalaria.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_cama	integer	
id_local_ingreso	integer	
cama	varchar	

<b>Nombre: ce_em (Nomenclador)</b>		
<b>Descripción:</b> Almacena las vías de ingreso consulta externa, emergencia		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_ce_em	integer	
ce_em	varchar	

<b>Nombre: cita</b>		
<b>Descripción:</b> Almacena las cita con un especialista.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_cita	integer	
id_especialidad	integer	
id_paciente	integer	
id_local_cita	integer	
indicaciones	varchar	
id_estado_cita	integer	
fecha_hora	time	
id_medico_interno	integer	
id_medico_ref	integer	
id_diag_admision	integer	

<b>Nombre: comite_salud (Nomenclador)</b>		
<b>Descripción:</b> Almacena el comité de salud de un paciente.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_comite_salud	integer	
comite_salud	varchar	

<b>Nombre: cond_alojam (Nomenclador)</b>		
<b>Descripción:</b> Almacena las condiciones de alojamiento de un paciente.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_cond_alojam	integer	
cond_alojam	varchar	

<b>Nombre: consejo_comunal (Nomenclador)</b>		
<b>Descripción:</b> Almacena el consejo comunal de un paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_consejo_comunal	integer	
consejo_comunal	varchar	

<b>Nombre: datos_socioeconom</b>		
<b>Descripción:</b> Almacena los datos socioeconómicos de los pacientes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_datos_socioeconom	integer	
m_robinson	boolean	
m_rivas	boolean	
m_sucre	boolean	
m_vuelvan_caras	boolean	
m_negra_hipolita	boolean	
m_barrio_adentro	boolean	
m_milagro	boolean	
m_madre_barrio	boolean	
id_comite_salud	integer	
id_consejo_comunal	integer	
id_prof_jefe_fam	integer	
id_nivel_inst_madre	integer	
id_princ_fuente_ing	integer	
id_cond_alojam	integer	
id_paciente	integer	

<b>Nombre: diag_asociado (Nomenclador)</b>		
<b>Descripción:</b> Almacena los posibles diagnósticos asociados de los pacientes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_diag_asociado	integer	
diag_asociado	varchar	
cie	varchar	

<b>Nombre: diagnostico (Nomenclador)</b>		
<b>Descripción:</b> Almacena los diagnósticos de los pacientes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_diagnostico	Integer	
diagnostico	varchar	
cie	varchar	

<b>Nombre: egreso</b>		
<b>Descripción:</b> Almacena el Egreso de un paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_egreso	integer	
fecha_egreso	date	
resumen_evol	varchar	
diag_egreso	varchar	
resumen_trat	varchar	
indicaciones	varchar	
id_motivo_egreso	integer	
id_medico_interno	integer	

<b>Nombre: especialidad (Nomenclador)</b>		
<b>Descripción:</b> Almacena la especialidad de un médico o equipo médico		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_especialidad	integer	
especialidad	varchar	

<b>Nombre: estado (Nomenclador)</b>		
<b>Descripción:</b> Almacena los estados (provincias) de un país		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_estado	integer	
estado	varchar	

<b>Nombre: estado_cita (Nomenclador)</b>		
<b>Descripción:</b> Almacena los posibles estados en los que puede encontrarse una cita.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_estado_cita	integer	
estado_cita	varchar	

<b>Nombre: estado_civil (Nomenclador)</b>		
<b>Descripción:</b> Almacena el estado civil de una persona.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_estado_civil	integer	
estado_civil	varchar	

<b>Nombre: estado_paciente (Nomenclador)</b>		
<b>Descripción:</b> Almacena el estado de salud de un paciente.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_estado_paciente	integer	
estado_paciente	varchar	

<b>Nombre: estudios_realiz (Nomenclador)</b>		
<b>Descripción:</b> Almacena los diferentes niveles culturales de una persona.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_estudios_realiz	integer	
estudios_realiz	varchar	

<b>Nombre: historia_gral</b>		
<b>Descripción:</b> Almacena la historia clínica general de un paciente.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_historia_gral	integer	
id_cita	integer	
id_ingreso	integer	
id_paciente	integer	
id_orden_medica	integer	

<b>Nombre: hoja_clinica</b>		
<b>Descripción:</b> Almacena una hoja electrónica en la historia clínica general.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_hoja_clinica	integer	
id_historia_general	integer	
fecha_creacion	time	
estado_paciente	integer	
observaciones	varchar	
tipo_hoja_clinica	integer	

<b>Nombre: ingreso</b>		
<b>Descripción:</b> Almacena los datos de los ingresos de los pacientes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_ingreso	integer	
id_ce_em	integer	
motivos	varchar	
enf_actual	varchar	
fecha_ing	date	
id_egreso	integer	
id_paciente	integer	
id_medico_interno	integer	
id_medico_ref	integer	
id_cama_ing	integer	
id_cama_temp	integer	
id_diag_admision	integer	

<b>Nombre: institucion_ref (Nomenclador)</b>		
<b>Descripción:</b> Almacena las posibles instituciones referidoras de los pacientes.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_institucion_ref	integer	
institucion_ref	varchar	

<b>Nombre: local_cita(Nomenclador)</b>		
<b>Descripción:</b> Local para realizar citas con especialistas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_local_cita	integer	
id_ubicacion	integer	
local_cita	varchar	

<b>Nombre: local_ingreso(Nomenclador)</b>		
<b>Descripción:</b> Local para ingresar a los pacientes		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_local_ingreso	integer	
id_ubicacion	integer	
local_ingreso	varchar	

<b>Nombre: medico</b>		
<b>Descripción:</b> de esta tabla heredan la tabla "medico_interno" y "medico_ref"		
Atributo	Tipo	Descripción
id_medico	integer	
nombre	varchar	
apellidos	varchar	
cedula_ident	varchar	
id_especialidad	integer	
id_colegio_medico	integer	
mms	varchar	

<b>Nombre: medico_interno</b>		
<b>Descripción:</b> Especialistas de la clínica		
Atributo	Tipo	Descripción
id_medico_interno	integer	
id_medico	integer	

<b>Nombre: medico_ref</b>		
<b>Descripción:</b> Médicos que refieren a los pacientes a la clínica		
Atributo	Tipo	Descripción
id_medico_ref	integer	
id_medico	integer	
id_institucion_ref	integer	

<b>Nombre: motivo_egreso(Nomenclador)</b>		
<b>Descripción:</b> Motivos por el cual se dan de alta a los pacientes		
Atributo	Tipo	Descripción
id_motivo_egreso	integer	
motivo_egreso	varchar	

<b>Nombre: municipio(Nomenclador)</b>		
<b>Descripción:</b> Municipios de residencia de los pacientes		
Atributo	Tipo	Descripción
id_municipio	integer	
municipio	varchar	
id_estado	integer	

<b>Nombre: nacionalidad(Nomenclador)</b>		
<b>Descripción:</b> Nacionalidad de los pacientes		
Atributo	Tipo	Descripción
id_nacionalidad	integer	
nacionalidad	varchar	

<b>Nombre: nivel</b>		
<b>Descripción:</b> Niveles de la clínica		
Atributo	Tipo	Descripción
id_nivel	integer	
nivel	varchar	

<b>Nombre: nivel_inst_madre(Nomenclador)</b>		
<b>Descripción:</b> Nivel de instrucción de la madre del paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_nivel_inst_madre	integer	
nivel_inst_madre	varchar	

<b>Nombre: ocupacion(Nomenclador)</b>		
<b>Descripción:</b> Ocupación del representante legal del paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_ocupacion	integer	
ocupacion	varchar	

<b>Nombre: orden_estudio</b>		
<b>Descripción:</b> Tabla para guardar la ordenes de estudios imagenológicos		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_orden_estudio	integer	
buscar_especificamente	varchar	
signos_sintomas	varchar	
peso	double precision	
talla	double precision	

<b>Nombre: orden_medica</b>		
<b>Descripción:</b> Tabla para guardar las ordenes médicas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_orden_medica	integer	

<b>Nombre: paciente</b>		
<b>Descripción:</b> Tabla para guardar los datos de los pacientes admitidos en la clínica		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
no_historia	varchar	
direccion_actual	varchar	
foto	varchar	
id_religion	integer	
id_entidad_federal	integer	
id_rep_legal	integer	
id_paciente	integer	
id_medico_ref	integer	
id_persona	integer	
id_estado_res	integer	
id_municipio_res	integer	

<b>Nombre: parentesco(Nomenclador)</b>		
<b>Descripción:</b> Parentesco entre el representante legal y el paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_parentesco	integer	
parentesco	varchar	

<b>Nombre: persona</b>		
<b>Descripción:</b> Tabla para guardar los datos personales de los pacientes y representantes legales		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_persona	integer	
nombre	varchar	
apellidos	varchar	
id_sexo	integer	
cedula_ident	varchar	
id_nacionalidad	integer	
fecha_nac	date	
id_estudios_realiz	integer	
id_estado_nac	integer	

<b>Nombre: princ_fuente_ing(Nomenclador)</b>		
<b>Descripción:</b> Principal fuente de ingreso de la familia del paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_princ_fuente_ing	integer	
princ_fuente_ing	varchar	

<b>Nombre: prof_jefe_fam(Nomenclador)</b>		
<b>Descripción:</b> Profesión del jefe de la familia del paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_prof_jefe_fam	integer	
prof_jefe_fam	varchar	

<b>Nombre: religion(Nomenclador)</b>		
<b>Descripción:</b> Religión que practica el representante legal del paciente		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_religion	integer	
religion	varchar	

<b>Nombre: remision_entrada(Nomenclador)</b>		
<b>Descripción:</b> Remisiones de entrada en la clínica		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_remision	integer	
remision	varchar	

<b>Nombre: remision_salida(Nomenclador)</b>		
<b>Descripción:</b> Remisiones de salidas existentes en la clínica		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_remision	integer	
remision	varchar	

<b>Nombre: rep_legal</b>		
<b>Descripción:</b> Tabla para guardar los datos del representante legal de los pacientes		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_parentesco	integer	
telefono	varchar	
celular	varchar	
telef_emergencia	varchar	
id_estado_civil	integer	
id_rep_legal	integer	
id_persona	integer	
id_ocupacion	integer	

<b>Nombre: sexo(Nomenclador)</b>		
<b>Descripción:</b> Sexo de las personas		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_sexo	integer	
sexo	varchar	

<b>Nombre: tipo_hoja_clinica(Nomenclador)</b>		
<b>Descripción:</b>		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_tipo_hoja_clinica	integer	
tipo_hoja_clinica	varchar	
ruta_icono	varchar	

<b>Nombre: ubicacion</b>		
<b>Descripción:</b> Ubicación que tendrán los pacientes al ser ingresados y citados		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
id_ubicacion	integer	
id_nivel	integer	
id_ala	integer	
ubicacion	varchar	

## GLOSARIO

<b>Término</b>	<b>Definición</b>
ASCII	American Standard Code for Information Interchange. Código Estándar Americano para el Intercambio de Información.
CMS	Content Management System. Sistema para la creación, almacenamiento, modificación y eliminación de recursos de un repositorio en una organización. Incluye herramientas para publicación, administración de presentación, control de revisión, indexado, búsqueda y obtención de información.
DICOM	Estandar para la visualización, almacenamiento y transmisión de imágenes en medicina. Norma las especificaciones que deben cumplir los sistemas para la transmisión, almacenamiento y visualización de imágenes médicas.
DICOM Standards Committee	Comité que se creó por ACR y NEMA para la elaboración del estándar DICOM y que en 1985 lanza su primera versión.
DOM	DOM o Document Object Model. Es una interfaz multi lenguaje y multiplataforma que permite el acceso dinámico y la modificación de los elementos que conforman el contenido, la estructura y el estilo de los documentos en la web.
E-BUSINESS	Comercio electrónico. Se entiende como la realización de procesos de comercio sobre una red telemática (ej. Internet)
Ensamblado (.NET)	Es una biblioteca de código parcialmente compilada que puede usarse para el despliegue e instalación de sistemas. Existen dos tipos: procesos (.EXE) y bibliotecas (.DLL).
FTP	FTP o File Transfer Protocol. Es un protocolo para la transferencia de ficheros entre nodos de una red.
HC	Abreviatura comúnmente usada para referirse a la Historia Clínica de un paciente.
HIS	Sistema de Información Hospitalaria. Son sistemas de software que trabajan acoplados y permiten la informatización de los distintos servicios de las instituciones de salud.
HL7	Es una especificación para un estándar de intercambio de datos electrónicos en el área de la salud, con especial énfasis en las comunicaciones intra - hospitalarias. en el área de la información clínica y administrativa.
HTML	HyperText Markup Language. Es el lenguaje para la representación de la información en la web (ver Web).
ISAPI	Internet Server Application Programming Interface. Es un conjunto de estándares que proveen funciones de seguridad para los servidores Windows NT y conexiones a bases de datos a través de Internet Information Server (IIS).
LIS	Sistema de Información de Laboratorio (Laboratory Information System). Es un sistema de gestión de la información generada y manipulada en los procesos de negocio de Laboratorio Clínico.
PACS	Sistema para el almacenamiento y comunicación de imágenes médicas.
PDF	Portable Document Format. Es un formato de fichero que simula un documento impreso como una imagen electrónica que se puede leer, crear, navegar, imprimir, etc.
PostScript	Es un lenguaje de descripción de páginas creado por Adobe Systems, que es un estándar independiente de dispositivos, para la visualización de documentos y gráficos.

RIS	Sistema de Información Radiológica (Radiological Information System). Es un sistema encargado de la gestión de la información generada y manipulada como resultado de los procesos de negocio de carácter radiológico (imagenológico)
RTF	Es un método de codificación de formatos de texto, usando el conjunto de caracteres ASCII.
SMTP	Simple Mail Transfer Protocol. Es un protocolo para el envío y recepción de mensajería de correo electrónico.
UNIX	Sistema operativo creado en 1960 por los Laboratorio Bell de AT&T, Dennis Ritchie, Ken Thompson y Douglas Mcllroy y que fue licenciado a las universidades. A partir del cual han surgido varios proyectos de código abierto como Linux.
W3C	W3C o World Wide Web Consortium. Es un consorcio internacional de compañías, involucradas con Internet y la Web. Su objetivo fundamental es el desarrollo de estándares abiertos para asegurar la evolución de la web en una sola dirección.
WIN32	Plataforma del sistema operativo Microsoft Windows de 32 bits.
WorkList Server	Servidor de Listas de Trabajo. Componente de un PACS-RIS en una institución que se encarga de gestionar la información relativa a las listas de trabajo de los dispositivos de adquisición de imágenes médicas.
XHTML	XHTML o EXtensible HyperText Markup Language es una reformulación de HTML escrito en XML. Puede ser considerado como una versión más estricta y limpia de HTML.
XML	XML o Extensible Markup Language. Es un metalenguaje extensible para la especificación de datos en documentos.