

Universidad de las Ciencias Informáticas

Facultad 7



Título: Sistema para la telemedicina online

Trabajo de Diploma para optar por el título de

Ingeniero Informático

Autor: Yeinier Ferrás Cecilio

Tutor: Lic. Rolando Bonal Cáceres

Ciudad de la Habana, Julio 2007

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 5 días del mes de julio del año 2007.

Autor: Yeinier Ferrás Cecilio

Tutor: Lic. Rolando Bonal Cáceres

DATOS DE CONTACTO

TUTOR: Lic. Rolando Bonal Cáceres (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).

AGRADECIMIENTOS

Agradezco a las personas que han contribuido de una forma u otra a la culminación de este trabajo.

Mis agradecimientos son para: Alejandro Mario Velázquez, Anisleidy Fernández Hernández, Lucelia Jiménez, Karel Tamayo, Jublar García, Yanoksy Durañona, Lázaro Rodríguez, Heliodoro Rodríguez, Alejandro Arias, Nuvia A. Estévez, Aylin Parra, Yoeny Caballero, José Alejandro Segura, Yilen Pons, Karelis Mesa, Darlen Horia, a mis compañeros del proyecto y también a los que olvido mencionar en este momento.

Además debo agradecer a mis padres, a mi hermano Elser, a mis abuelos Melba y Arlan.

Al resto de mi familia, amigos y vecinos que aunque separados a varios cientos de kilómetros siempre me apoyaron, aconsejaron y ayudaron.

DEDICATORIA

Dedico este trabajo a mis padres, mi hermano, mis abuelos y familia en general.

RESUMEN

En Cuba, el Sistema Nacional de Salud Pública se ha planteado la necesidad de informatizar y mejorar sus servicios. Para ello, se han ido adquiriendo modernos equipos médicos de obtención de imágenes distribuidos mayormente en hospitales y centros especializados de investigaciones. Estos, al estar computarizados son capaces de comunicarse con ordenadores y otros equipos mediante redes telemáticas de transmisión de datos. El siguiente trabajo tiene como objetivo fundamental el desarrollo e implementación de un sistema para la teleconsulta online.

Para su desarrollo, se empleó la metodología de desarrollo de software RUP. Así como, UML como lenguaje de modelamiento de software; Visual C# .Net como lenguaje del lado del servidor y del lado del cliente se empleó JavaScript. Como herramientas de desarrollo y modelamiento de software Visual Studio .Net Team Suite 2005 y Enterprise Architect respectivamente. Por último, se empleó la tecnología ASP.NET para el desarrollo de aplicaciones web.

Con el presente trabajo se espera facilitar la obtención de segundas opiniones de médicos y especialistas en el proceso de diagnósticos de pacientes, ya que será posible el intercambio de imágenes médicas. Además, será posible habilitar de forma progresiva servicios especializados de atención a la salud donde no existan. También permitirá ayudar a la formación de los futuros médicos.

PALABRAS CLAVE

Telemedicina, teleconsulta, DICOM.

ÍNDICE

INTRODUCCIÓN	11
CAPÍTULO I FUNDAMENTACIÓN TEÓRICA.....	14
1.1 DEFINIENDO LA TELEMEDICINA.....	14
1.2 EVOLUCIÓN DE LA TELEMEDICINA EN EL MUNDO.	15
1.3 TELEMEDICINA EN CUBA.....	16
1.4 SOFTWARE EN TELEMEDICINA.	17
1.5 METODOLOGÍA DE DESARROLLO.....	18
1.5.1 RUP	18
1.6 LENGUAJES.	21
1.6.1 Visual C#.Net	21
1.6.2 JavaScript.....	22
1.6.3 UML	23
1.7 HERRAMIENTAS.....	24
1.7.1 Visual Studio .NET	24
1.7.2 Enterprise Architect.....	25
1.8 TECNOLOGÍAS.....	25
1.8.1 AJAX.....	25
1.8.2 ASP.NET.....	31
1.9 CONCLUSIONES.....	34
CAPÍTULO II CARACTERÍSTICAS DEL SISTEMA	35
2.1 OBJETO DE AUTOMATIZACIÓN	35
2.2 PROPUESTA DE SISTEMA	36
2.3 ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA	36
2.3.1 Funciones del sistema.....	36
2.3.2 Atributos del sistema.	40
2.4 MODELO CONCEPTUAL DEL SISTEMA.....	41
2.5 DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA.....	42

2.5.1 Actores del sistema.....	42
2.5.2 Lista de los casos de uso del sistema.	42
2.5.3 Diagramas de los casos de uso del sistema.	46
2.6 CONCLUSIONES	54
CAPÍTULO III ANÁLISIS Y DISEÑO DEL SISTEMA	55
3.1 ANÁLISIS	55
3.2 DISEÑO	55
3.2.1 Descripción de las clases.	55
3.2.2 Diagrama entidad-relación.....	67
3.3 CONCLUSIONES	70
CAPÍTULO IV IMPLEMENTACIÓN	71
4.1 DESPLIEGUE	71
4.2 COMPONENTES	73
4.3 CONCLUSIONES	75
CONCLUSIONES	76
RECOMENDACIONES	77
REFERENCIAS BIBLIOGRÁFICAS	78
BIBLIOGRAFÍA	79
ANEXOS	81
ANEXO I. CASOS DE USO EXPANDIDOS	81
CU-01– Subir DICOM para el servidor.....	81
CU-02 – Seleccionar paciente.....	82
CU-03 – Eliminar paciente	82
CU-04 – Realizar anotaciones en el paciente.....	83
CU-05 – Seleccionar estudio.....	83
CU-06 – Eliminar estudio	84
CU-07 – Realizar anotaciones en el estudio	84
CU-08 – Seleccionar serie.....	85

CU-09 – Eliminar serie	85
CU-10 – Realizar anotaciones en la serie	86
CU-11 – Seleccionar imagen	86
CU-12 – Eliminar imagen	87
CU-13 – Realizar anotaciones en la imagen	87
CU-14 – Buscar usuarios	88
CU-15 – Añadir contactos.....	88
CU-16 – Eliminar contacto	89
CU-17 – Renombrar contacto	89
CU-18 – Teleconsultar contacto	90
CU-19 – Hacer zoom sobre la imagen.....	91
CU-20 – Mover la imagen	91
CU-21 – Editar perfil	92
ANEXO II. DIAGRAMAS DE CLASES DEL ANÁLISIS	93
Subir DICOM para el servidor.....	93
Seleccionar paciente.....	94
Eliminar paciente	94
Realizar anotaciones al paciente	95
Seleccionar estudio	95
Eliminar estudio	96
Realizar anotaciones al estudio	96
Seleccionar serie	97
Eliminar serie	97
Realizar anotaciones a la serie	98
Seleccionar imagen	98
Eliminar imagen	99
Realizar anotaciones a la imagen	99
Buscar usuario.....	100
Añadir contacto	101
Eliminar contacto	102

Renombrar contacto	102
Teleconsultar contacto	103
Hacer zoom sobre la imagen	104
Mover la imagen	104
Editar perfil	105
ANEXO III. DIAGRAMAS DE CLASES DEL DISEÑO	106
Subir DICOM para el servidor	107
Seleccionar paciente	108
Eliminar paciente	109
Realizar anotaciones en el paciente	109
Seleccionar estudio	110
Eliminar estudio	111
Realizar anotaciones en el estudio	111
Seleccionar serie	112
Eliminar serie	113
Realizar anotaciones en la serie	113
Seleccionar imagen	114
Eliminar imagen	115
Realizar anotaciones en la imagen	115
Buscar usuarios	115
Añadir contacto	116
Añadir contacto	117
Eliminar contacto	118
Renombrar contacto	119
Teleconsultar contacto	120
Hacer zoom sobre la imagen	121
Mover la imagen	122
Editar perfil	122

INTRODUCCIÓN

Rápida y vertiginosa ha sido la evolución de la informática desde las primeras máquinas programables manualmente (maquina diferencial de Babbage) o procedimientos electrónicos (ENIAC, con tubos al vacío, en 1947), hasta la actualidad, donde existen potentes computadoras digitales. Estas han sido introducidas en casi todas las esferas de la vida del ser humano (industria, comercio, comunicaciones, transporte, militar, etc.). Con todos estos avances tecnológicos y nuevas necesidades, la comunicación o transmisión de datos fue tomando más y más auge, surgiendo así las redes de computadoras y evolucionando a escala mundial hasta el nacimiento de Internet. Naturalmente la medicina no estuvo ajena a tales avances de la tecnología y conjugó sus propios avances científicos con las telecomunicaciones y la informática, hasta tal punto que hoy en día existen equipos médicos de adquisición de imágenes computarizados capaces de comunicarse entre sí y con otros dispositivos de cómputo.

Cuba, para brindar mejores servicios de salud ha ido adquiriendo moderna técnica especializada, dentro de la cual se encuentran equipos de adquisición de imágenes médica (Tomógrafos, equipos de Ultrasonidos, Resonancia Magnética y Rayos X). Estos se encuentran localizados fundamentalmente en centros especializados de investigaciones médicas y hospitales, para brindar sus servicios.

Después de realizado un estudio a un paciente con alguno de estos equipos, se obtienen una o varias imágenes embebidas en un fichero regido por un estándar médico internacional llamado DICOM (Digital Imaging and Communications in Medicine) y que actualmente se encuentra en su versión 3. Este es un estándar para el almacenamiento, transmisión y comunicación de imágenes médicas.

Inmediatamente al ser obtenido un estudio puede ser examinado en un computador por médicos y especialistas con el objetivo de diagnosticar a los pacientes o, se pueden almacenar en algún dispositivo físico (CD, DVD, etc.) o en algún servidor especializado en este fin, para después ser analizados. Estos se van acumulando y forman el historial clínico imagenológico de los pacientes, que posteriormente pueden ser utilizados para nuevos diagnósticos, tratamientos, investigaciones o la educación de los futuros médicos.

Anteriormente se había mencionado que la mayor concentración de la moderna técnica se encuentra distribuida entre los centros de atención secundaria (principalmente en los grandes hospitales) y terciaria de salud. Lo que provoca, que a menudo especialistas tengan que trasladarse hasta lugares remotos para brindar sus servicios o que pacientes con determinadas afecciones o padecimientos, para poder ser diagnosticados y tratados tengan que trasladarse hasta donde se brinden los servicios especializados que él requiere. Es evidente, que esto trae una serie de problemas, entre los cuales se encuentra el de ¿cómo habilitar los servicios especializados de salud en centros que no los posean?

El **objeto de estudio** de este trabajo son los procesos de la telemedicina, y más específico el **campo de acción** son los procesos de la teleconsulta. El **objetivo general** del trabajo: implementar una aplicación web que permitirá la teleconsulta online; para ello se proponen los siguientes **objetivos específicos**:

- Obtener información sobre la telemedicina mediante una búsqueda bibliográfica.
- Investigar situación actual de las tecnologías y herramientas existentes para el desarrollo de la aplicación.
- Analizar sistemas que brinden servicios de telemedicina.
- Proponer un sistema para darle solución al problema científico.
- Realizar la captura de requisitos.
- Realizar el análisis y el diseño del sistema
- Implementar el sistema.

El presente trabajo está estructurado en cuatro capítulos:

Capítulo I: Fundamentación teórica. Respalda teóricamente la investigación realizada.

Capítulo II: Características del sistema. Se propone el sistema y se exponen las funcionalidades, atributos, casos de uso y los respectivos diagramas de los casos de uso del sistema.

Capítulo III: Análisis y Diseño. Se exponen varios diagramas del análisis y del diseño del sistema, así como también las descripciones de las clases y las descripciones de las tablas de la base de datos.

Capítulo IV: Implementación. Se muestran la distribución de los elementos hardware en el diagrama de despliegue, y se muestran los componentes que componen valga la redundancia el sistema.

CAPÍTULO I

FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se definen el término de telemedicina y se explica de forma sencilla como funciona un sistema de teleconsultas. Además se mencionan una serie de acontecimientos que muestran la evolución y desarrollo de la telemedicina en el mundo. Se nombran software desarrollados por empresas extranjeras y nacionales los cuales se usan o pueden utilizarse en servicios de telemedicina. Por último, se describen una serie de tecnologías, herramientas, lenguajes y técnicas de desarrollo empleadas para el desarrollo del sistema de teleconsultas propuesto.

1.1 DEFINIENDO LA TELEMEDICINA.

El término telemedicina se usa para designar la prestación de servicios de medicina a distancia; pueden ser tan simples como dos profesionales de la salud discutiendo un caso por vía telefónica o tan compleja como realizar una operación quirúrgica mediante el brazo de un robot controlado a miles de kilómetros de distancia. La telemedicina, es un recurso tecnológico que posibilita la optimización de los servicios de atención en salud, ahorra tiempo y dinero, facilita el acceso a zonas remotas de la geografía para tener la atención de especialistas.

Todo esto puede traducirse en una disminución de tiempos entre la toma de exámenes y la obtención de resultados, o entre la atención y diagnóstico remoto acertado de un especialista, el cual no necesita moverse de su lugar para tratar al paciente; o el paciente no tiene necesidad de moverse para recibir un diagnóstico.

En la actualidad la telemedicina se usa más comúnmente en dos áreas de trabajo: la práctica y la educativa. Dentro del área práctica se puede mencionar la siguiente forma: teleconsulta; y dentro del área educativa se puede mencionar el caso de las clases a distancia. Básicamente la educación médica a distancia hace uso de las técnicas de videoconferencias, de esta manera se le saca mejor provecho a los materiales presentados.

La teleconsulta: no es más que el diagnóstico a distancia o remoto, es la técnica de mayor impacto dadas las múltiples ventajas que presenta, y posee un amplio margen para la explotación de los medios tecnológicos. Consiste en evaluar o asistir la evaluación médica de un paciente desde un centro hospitalario remoto, haciendo uso de las telecomunicaciones para llevar a cabo esta acción.

Un sistema de telemedicina funciona básicamente de la siguiente forma: existe un centro hospitalario menor que presenta una carencia de especialistas en determinadas áreas específicas, este centro será asistido por uno de mayor envergadura, el cual dispondrá de los especialistas y del tiempo necesario para la atención de los pacientes de forma “remota”, quienes se encontrarán físicamente en el lugar donde se encuentra el centro de menor tamaño. Como ya se ha mencionado anteriormente esto conlleva a un ahorro considerable de tiempo, dinero y recursos; adicionándole además otros servicios de salud a centros apartados en la geografía.

Lógicamente para que un sistema de telemedicina funcione de forma correcta son necesarios recursos y medios técnicos pertinentes como por ejemplo: equipos capaces de comunicarse (computadores, equipos para las videoconferencias, etc.) que se encontrarán en los locales de los hospitales, vías de comunicación (Internet, LAN¹, pequeñas redes inalámbricas, etc.), aplicaciones de computador específicas para estas tareas y, por supuesto, el personal que se dispondrá para dar asistencia remota.

1.2 EVOLUCIÓN DE LA TELEMEDICINA EN EL MUNDO.

Existen antecedentes de que se intentaron diagnósticos a distancia desde casi la invención del teléfono, por lo que no es una nueva arista de la globalización de las comunicaciones o la Internet:

- 1924 – Aparece en la revista Radio News, un artículo titulado “Doctor por Radio”, el cual describía el esquema de la circuitería necesaria para lograrlo.
- 1955 – En Montreal, el Dr. Albert Jutras realiza Teleradiología, a fin de evitar las altas dosis de radiación que incidían en las fluoroscopias; se hizo uso de un interfono convencional.

¹ LAN (Local Area Network) y se refiere a las redes locales de computadoras.

- 1959 – Nebraska, Cecil Wittson comienza sus primeros cursos de Teleeducación y de Telepsiquiatría, entre su Hospital y el Hospital del Estado en Norfolk, Virginia, a 180 kilómetros de distancia.
- 1971 – Se inicia la era de los satélites, en especial el ATS (lanzado en 1966), con el fin de mejorar las prestaciones de una comunidad de nativos de Alaska.
- 1972 – Inicio de STARPAHC, programa de asistencia médica para nativos de Papago Arizona. Se realizó electrocardiografía y radiología y se transmitió por medio de microondas.
- 1975 – Finaliza el programa STARPAHC, el cual fue adaptado de un programa de atención médica para astronautas por la compañía Lockheed.
- 1988 – La NASA lanza el programa “Space Bridge” a fin de colaborar con Armenia y Ufa (en esa época pertenecientes a la unión soviética), Armenia fue devastada por un terremoto. Las conexiones se hicieron usando vídeo en una dirección y voz y fax bidireccionales entre el Centro Médico de Yerevan, Armenia y cuatro Hospitales en Estados Unidos, extendiéndose posteriormente el programa a Ufa, para socorrer a los quemados en un terrible accidente de tren.
- 1995 – La Clínica Mayo pone en marcha una conexión permanente con el Hospital Real de Ammán en Jordania, se realizaban consultas diarias entre un médico Hachemita y otros de Estados Unidos, el médico Hachemita presentaba, como si de una sesión clínica del hospital se tratase, a los pacientes de forma sucesiva; en directo los médicos americanos preguntaban o pedían al médico jordano que preguntara a su vez al paciente por sus dolencias. En otros casos eran interpretaciones de radiografías o problemas dermatológicos.
- 2001 – Un doctor en New York elimina la vesícula enferma de un paciente en Estrasburgo, Francia, por medio de un brazo robot. (1)

1.3 TELEMEDICINA EN CUBA.

El primer programa de telemedicina en la especialidad de Oftalmología entre Cuba y el mundo se realizó en el mes de marzo de 1999, entre el Hospital Oftalmológico Docente “Ramón Pando Ferrer” y la

Universidad de Indiana después de la visita del Dr. Eugene Helveston, profesor de la mencionada Universidad.

Para hacer posible la consulta se utilizaron una cámara digital SONY, una computadora con procesador de 450 MHz y memoria RAM de 64 MB, y un MODEM. Una vez realizadas todas las pruebas a un paciente, en las cuales se obtenían una serie de imágenes digitales y otros datos de interés que posteriormente serían enviados a la Universidad de Indiana donde el Dr. Eugene Helveston las analizaría y enviaría su criterio a cerca del caso en cuestión al equipo médico cubano para después emitir un diagnóstico al paciente. Como resultado se atendieron 77 pacientes; todos, menos 4 fueron diagnosticados correctamente. (2)

1.4 SOFTWARE EN TELEMEDICINA.

A nivel internacional se pueden encontrar varias compañías con aristas dedicadas al desarrollo de aplicaciones para la Telemedicina, por ejemplo:

- *Second Opinion Software*, dedicada a la formación de una red global de información médica. Esta compañía ha desarrollado un producto que se encuentra en dos ediciones:
 - *Second Opinion Standard 4.0*, permite la captura de imágenes y de clips de video, almacenarlos, así como también enviarlos a sitios remotos a través del correo electrónico; además incluye comunicación punto-a-punto.
 - *Second Opinion Professional 6.2*, posee la captura de imágenes y video, así como también otras informaciones del paciente; tiene una base de datos diseñada para ser compartida por varios usuarios ubicados en diferentes lugares.

Estos dos productos no soportan el estándar DICOM.

- *GlobalMedia*, fundada en el 2002 en Scottsdale, Arizona. Ha emergido como una de las más respetadas innovadoras en el diseño, desarrollo y fabricación de soluciones en comunicaciones digitales. Cuenta con varios productos para la industria y la medicina. Dentro de los productos para la telemedicina se pueden mencionar los siguientes:
 - *CapSure Pro Lab 100*.

- *CapSure Pro Lab 200.*
- *CapSure Pro Lab 300.*

Sus productos no están pensados en el estándar DICOM, solo se emplean para la transmisión de imágenes, audio y video, lo que los hace muy útiles para las videoconferencias.

- *Vitech*, se especializa en el diseño, instalación y venta de sistemas de videoconferencias, dentro de sus servicios se incluyen venta y renta de equipos y software.

A nivel nacional se encuentran algunos softwares que brindan servicios de telemedicina:

- *IMAGIS*, desarrollado en el Centro de Biofísica Médica de Santiago de Cuba. Dentro de sus prestaciones están recibir y almacenar imágenes en formato DICOM, así como también buscar y recuperar las imágenes desde una red.
- *CardioWeb*, diseñado en el 2003 por la Empresa de Servicios Informáticos de Holguín (ESI) y el Hospital “Vladimir Ilich Lenin”, dirigido a los servicios de cardiología y cirugía cardiovascular y que además permite la telemedicina.
- *Cassandra DICOM-Mail*, es uno de los componentes del PACS (Picture Archiving and Communication System) desarrollado por el Grupo de Procesamiento de Imágenes y Señales (GPI) de la Universidad de las Ciencias Informáticas (UCI); este componente permite la teleconsulta, pero esta no es interactiva.

1.5 METODOLOGÍA DE DESARROLLO.

1.5.1 RUP

El RUP (Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

Los orígenes de RUP se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuidores claves de RUP colaboró con Boehm en la investigación. En 1995 Rational Software es comprada por una compañía sueca llamada Objectory AB. El Rational Unified Process fue el resultado de una convergencia de Rational Approach y Objectory, proceso desarrollado por el fundador de Objectory Ivan Jacobson. El primer resultado de esta fusión fue el Rational Objectory Process, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe Philippe Kruchten.

RUP posee como características principales:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización; se basa en los siguientes seis principios:

1. *Adaptar el proceso.* El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico.
2. *Balancear prioridades.* Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.
3. *Colaboración entre equipos.* El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc.
4. *Demostrar valor iterativamente.* Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto.
5. *Elevar el nivel de abstracción.* Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL² o esquemas (frameworks³) por nombrar algunos. Esto previene a los ingenieros de software ir directamente de los requisitos a la codificación de software a la medida del cliente. Un nivel alto de abstracción también permite discusiones sobre diversos niveles arquitectónicos. Éstos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.
6. *Enfocarse en la calidad.* El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

² Lenguaje de programación que usa instrucciones de alto nivel para recuperar y dar forma a la información.

³ En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

- *Concepción*: se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos
- *Elaboración*: se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos
- *Construcción*: se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario
- *Transición*: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

En la siguiente figura (ver Fig. 1) se muestra el perfil típico de proyecto, mostrando el tamaño relativo de las cuatro fases. (3)

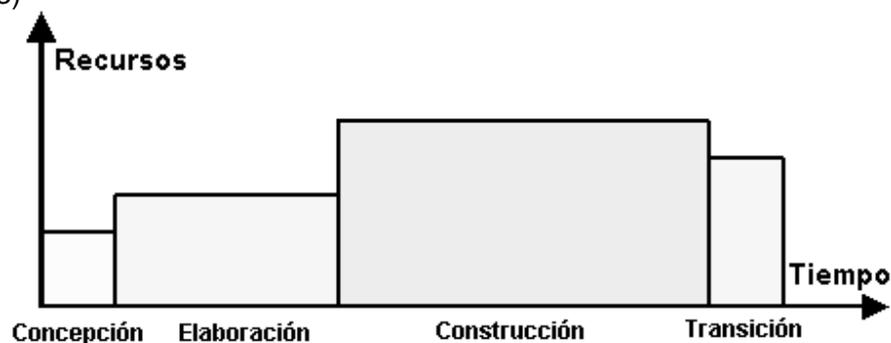


Fig. 1

1.6 LENGUAJES.

1.6.1 Visual C#.Net

Visual C#.Net es un lenguaje de POO (Programación Orientada a Objetos) desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que posteriormente fue aprobado como un estándar por la ECMA⁴ y la ISO⁵. Su sintaxis básica se deriva C/C++ y emplea el modelado de objetos de la plataforma

⁴ La ECMA (European Computer Manufacturers Association) es una asociación fundada en 1961 dedicada a la estandarización de sistemas de información.

⁵ ISO (International Organization for Standardization) es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización (ONs) nacionales, que produce normas internacionales industriales y comerciales.

.NET, el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes. C# fue diseñado para combinar el control a bajo nivel de C y la velocidad de programación de lenguajes como Visual Basic.

Es un lenguaje relativamente sencillo para aquellos programadores con conocimientos en lenguajes como C++ o Java. (4)

1.6.2 JavaScript

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM⁶ (Document Object Model).

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores web comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML⁷, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995.

⁶ DOM (Document Object Model) es una forma de representar documentos estructurados (tales como una página web HTML o un documento XML) que es independiente de cualquier lenguaje orientado a objetos.

⁷ HTML (Hyper Text Markup Language) es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también lo fue como un estándar ISO. (5)

1.6.3 UML

Lenguaje Unificado de Modelado 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 apoyado en gran manera por la OMG⁸. 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.

Es importante remarcar que UML es un "lenguaje" para especificar y no un método o proceso. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para soportar una metodología de desarrollo de software, pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas:

- *Diagramas de estructura*, enfatizan en los elementos que deben existir en el sistema modelado:
 - Diagrama de clases
 - Diagrama de componentes
 - Diagrama de objetos
 - Diagrama de estructura compuesta (UML 2.0)
 - Diagrama de despliegue

⁸ OMG (Object Management Group) es una organización NO lucrativa que promueve el uso de tecnología orientada a objetos mediante guías y especificaciones para tecnologías orientadas a objetos.

- Diagrama de paquetes
- *Diagramas de comportamiento*, enfatizan en lo que debe suceder en el sistema modelado:
 - Diagrama de actividades
 - Diagrama de casos de uso
 - Diagrama de estados
- *Diagramas de Interacción*, un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:
 - Diagrama de secuencia
 - Diagrama de comunicación
 - Diagrama de tiempos (UML 2.0)
 - Diagrama de vista de interacción (UML 2.0) (6)

1.7 HERRAMIENTAS

1.7.1 Visual Studio .NET

Visual Studio .NET es un IDE⁹ desarrollado por Microsoft a partir del 2002. Es para el sistema operativo Microsoft Windows y está pensado, principal pero no exclusivamente, para desarrollar para plataformas Win32.

Además de poderse desarrollar aplicaciones de escritorio, pueden ser desarrolladas aplicaciones Web usando ASP.NET y también aplicaciones para dispositivos portátiles usando el .NET Compact Framework.

La característica más notable del IDE es su soporte de varios lenguajes .NET (Visual Basic .NET, C#, J# y Visual C++). Los programas desarrollados en esos lenguajes no se compilan a código máquina ejecutable (como por ejemplo hace C++) sino que son compilados a un código intermedio llamado MSIL (Microsoft Intermediate Language). Cuando los programas ejecutan la aplicación MSIL, ésta es compilada en ese momento al código de máquina apropiado para la plataforma en la que se está ejecutando. Mediante este

⁹ IDE (Integrated Development Environment), provee un entorno de desarrollo como una aplicación simple, típicamente formado por un compilador, un depurador y una interfaz grafica de usuario.

método, Microsoft espera poder soportar varias implementaciones de sus sistemas operativos Windows (como Windows CE). Los programas compilados a MSIL pueden ejecutarse sólo en plataformas que tengan una implementación de .NET Framework. Es posible ejecutar programas MSIL en Linux o en MacOS X utilizando algunas implementaciones .NET que no pertenecen a Microsoft, como Mono y DotGNU. (7)

1.7.2 Enterprise Architect

Enterprise Architect 6.5 desarrollado por Sparx System, ofrece herramientas de modelado UML muy cómodas, flexibles y potentes para la plataforma de Windows. Es una herramienta de análisis de negocio y UML orientada a objetos para el desarrollo completo del ciclo de vida del software. Enterprise Architect provee el límite competitivo para el desarrollo de software, administración de proyecto, administración de requerimientos y análisis de negocio a un precio muy conveniente. Dentro de sus características más notables se destacan:

- Soporta la última especificación de UML 2.1
- Capacidad de Importación/Exportación XMI¹⁰ 2.1
- Motor de reporte HTML.
- Transformaciones MDA¹¹.
- Perfiles y soporte de tecnologías.
- Pruebas y rastreo de recursos.

1.8 TECNOLOGÍAS

1.8.1 AJAX

AJAX (Asynchronous JavaScript and XML¹²) nació con la idea de cambiar ciertos aspectos en contra de las aplicaciones web como son:

¹⁰ XMI (XML Metadata Interchange) es un estándar de la OMG para el intercambio de información de metadatos por XML

¹¹ Es un estándar propuesto por el OMG (Object Management Group) que promete acelerar el desarrollo de aplicaciones, simplificar la integración entre distintas tecnologías y reducir el coste de la migración de las aplicaciones a nuevas plataformas.

- *Pobre interactividad*, las aplicaciones web requieren que los usuarios esperen un tiempo hasta que las páginas se vuelvan a cargar después de cada interacción con el servidor. Durante el tiempo de carga estos deben mirar una pantalla blanca, lo cual disturba toda experiencia. Aunque el ancho de banda de las conexiones de Internet se están volviendo un estándar, las aplicaciones web se están convirtiendo cada vez más complejas (“pesadas”), así es que el conjunto de tiempos de espera permanece relativamente igual.
- *Apatía*, las clásicas aplicaciones web transfieren toda la información del formulario para el servidor, el cual a su vez envía de regreso todo el código HTML de la página para el navegador del cliente. Esto pasa durante cada postback¹³ y en la mayoría de los casos es altamente ineficiente, ya que solo una pequeña parte de la interfaz es cambiada en realidad. No obstante, partes del ancho de banda es consumido y el rendimiento está significativamente dificultado. Esto deja con la idea a los usuarios que las aplicaciones web son lentas por naturaleza. Aún peor, el usuario a menudo encontrará la página en una posición diferente, causando confusión.
- *Interfaces simplistas*, el requerimiento para un postback completo de la página en cualquier momento que la interfaz de usuario tiene que ser cambiada impone fuertes limitaciones en el grado de sofisticación de una interfaz web de usuario. Ricas y refinadas interfaces con actualización a pedido solo pueden ser implementadas utilizando tecnología Flash. Esta aproximación, sin embargo, es poco práctica para uso general puesto que es muy compleja y requiere de un conjunto diferente de habilidades que las poseídas por un típico desarrollador web. También puede causar problemas al usuario final como un plug-in¹⁴ requerido, en muchos casos.
- *Baja usabilidad*, si una aplicación web vuelve a cargar toda página porque el usuario hace una nueva selección en el formulario, ellos quedarán confundidos. Es con frecuencia el caso que aplicaciones web trabajan en una confusa y esotérica forma porque han sido construidas en torno a la vista estándar, de los protocolos de Internet. ASP.NET dio a entender que se pueden construir

¹² XML (eXtensible Markup Language) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

¹³ Peticiones subsiguientes al servidor después de cargada una página web.

¹⁴ Un plugin (o plug-in) es un programa de ordenador que interactúa con otro programa para aportarle una función o utilidad específica, generalmente muy específica

aplicaciones con mayor funcionalidad, más rápidas, la usabilidad tiene una forma de la que salir aún.

La nueva generación de aplicaciones basadas en AJAX posibilita un funcionamiento cercano a lo instantáneo, ricas interfaces y muy mejoradas a la experiencia del usuario. Abre nuevos horizontes para la interacción mucho más cercana con la aplicación y demuestra en la práctica lo que hasta hace poco parecía imposible:

- Google Maps o Virtual Earth muestran los mapas en tiempo real tal y como lo hace el Adobe Photoshop.
- La hojeada a carpetas, la vista previa de mensajes, etc. en Microsoft (MS) Outlook Web Access es muy similar a la versión de escritorio de Outlook.
- La validación de información compleja en los campos de entrada puede procesarse por el servidor, sin volver a cargar la página.

AJAX no es de hecho tan nuevo como una tecnología, ha sido usado por vez primera después que Microsoft implementó objetos COM MicrosoftXMLHTTP que fue parte de el MS XML Parser distribuido. Como un objeto ActiveX¹⁵ en el Internet Explorer 5.0, fue usado para crear el famoso MS Outlook Web Access. Lo que si es nuevo es el nombre AJAX, que fue aceptado en el 2005. También existen otras etiquetas con las cuales se le conoce: Load on Demand, Asynchronous Request, Callbacks, Out-of-band Calls.

Lo que es aún más interesante es que AJAX no es una tecnología, es más como una técnica de desarrollo que utiliza de forma única un número de tecnologías ya maduras: XML/XHTML, XML, DHTML, los objetos XmlHttpRequest, y JavaScript.

El centro de la idea detrás de AJAX es hacer la comunicación con el servidor de forma asíncrona, así que la información es transferida y procesada en el trasfondo. Como resultado el usuario puede seguir interactuando con otras partes de la página sin interrupción. En una aplicación basada en AJAX solo los elementos más relevantes de la página son actualizados, solo si esto es necesario.

¹⁵ Aplicación desarrollada por Microsoft que ofrece dinamismo en páginas web.

En contraste, la tradicional comunicación sincrónica con el servidor requerirá volver a cargar la página cada vez que la información deba ser transferida desde o hasta este. Esto induce los siguientes efectos negativos:

- La interacción del usuario con la aplicación es interrumpida cada vez que una llamada al servidor es necesaria, ya que un postback tiene que ser hecho.
- El usuario tiene que esperar y mirar una pantalla blanca durante cada postback.
- La página completa está siendo suministrada y transferida para el cliente después de cada postback, lo cual es tiempo consumido y tráfico intensivo.
- Alguna información suministrada por el usuario será enviada para el servidor, quizá prematuramente.

Las aplicaciones basadas en AJAX, por otro lado, con un nuevo método de comunicación asíncrona cliente-servidor, es implementado como un motor de JavaScript que es cargado en el cliente durante la carga inicial de la página. Desde allí, este motor sirve como un mediador que solo envía datos relevantes para el servidor como XML y subsecuentemente procesada, el servidor responde para actualizar los elementos relevantes de la página.

En la siguiente figura (Fig. 2) se refleja el funcionamiento de la comunicación asíncrona entre el navegador y el servidor.

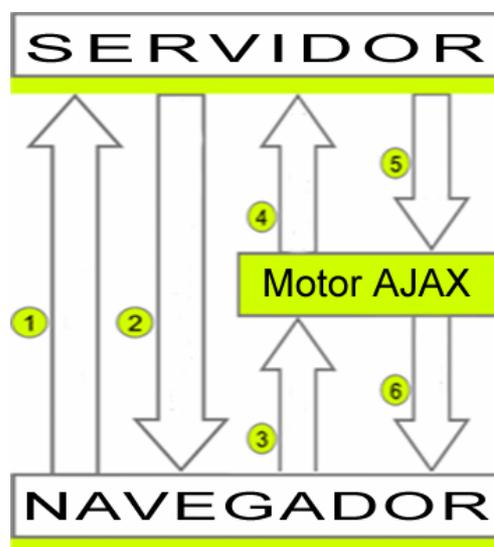


Fig. 2

1. Solicitud inicial por el navegador – el usuario realiza una petición de una URL¹⁶ en particular.
2. La página completa es suministrada por el servidor (junto con el motor de JavaScript) y envía para el cliente (HTML, CSS¹⁷, motor de JavaScript).
3. Todas las solicitudes ulteriores para el servidor son iniciadas como función de las llamadas del motor de JavaScript.
4. El motor de JavaScript entonces hace los XMLHttpRequest al servidor.
5. El servidor procesa las solicitudes y envía las respuestas en formato XML al cliente (documento XML). Contiene solo la información de los elementos que necesitan ser cambiados. En la mayoría de los casos esta información comprende solamente una fracción del total del marcado de la página.

¹⁶ URL (Uniform Resource Locator). Denominación que no solo representa una dirección de Internet sino que apunta a un recurso concreto dentro de esa dirección.

¹⁷ CSS (Cascading Style Sheets) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML).

6. EL motor de JavaScript procesa la respuesta del servidor, actualiza el contenido relevante de la página o ejecuta otra operación con los datos recibidos desde el servidor (HTML + CSS).

Dificultades a las que se enfrentan los desarrolladores a la hora de trabajar con AJAX:

- *Escritura y mantención de JavaScript complejo* – la construcción de aplicaciones basadas en AJAX requiere de habilidades sustanciales de JavaScript, que puede comenzar a ser un problema para un gran número de desarrolladores .NET. Adicionalmente, la falta de buenas herramientas de corrección de errores para los scripts del lado del cliente hace el proceso aún más difícil.
- *Gestión del estado de vista* – los controles Web ASP.NET adecuadamente mantienen su estado entre los postbacks. El mismo, sin embargo, no se aplica para los callbacks de AJAX. Como un resultado, los desarrolladores necesitan imaginar una manera para la administración idónea del estado de vista de las páginas.
- *Ruptura del paradigma de la página* – AJAX requiere una forma diferente de pensar a cerca de un sitio web, puesto que el concepto de “Página” ya no es válido. En efecto, las aplicaciones con AJAX pueden ser consideradas como acercamiento al desarrollo de las aplicaciones de escritorio. El hecho que una página ya no sostenga información constante conduce a dos consecuencias importantes: el botón de atrás y la preservación de cierta información ya no funcionara como se esperaba. Por lo tanto, los desarrolladores necesitan implementar mecanismos específicos para la superación de estos dos temas.
- *Accesibilidad* – las técnicas de desarrollo con AJAX violan fundamentalmente los requerimientos de accesibilidad. Como el contenido de la página está siendo actualizado de forma dinámica, los cambios no pueden ser detectados por herramientas de accesibilidad como lectores de pantalla. Además, algunos estándares de accesibilidad prohíben enteramente el uso de JavaScript, lo cual elimina prácticamente toda posibilidad de usar AJAX.
- *Las nuevas interfaces de usuario interactivas requieren de conocimientos* – las riquezas de las interfaces de usuario en las aplicaciones con AJAX presenta a los usuarios con una nueva e inesperada funcionalidad. A pesar que esta es la principal razón e primer lugar para usar AJAX, esto requiere de conocimientos.

Principales ventajas de AJAX:

- *Mejor rendimiento y eficiencia* – la principal ventaja de las aplicaciones con AJAX es su significativo alto rendimiento, el cual es el resultado de la pequeña cantidad de información transferida desde el servidor. Esto hace a la técnica de AJAX especialmente beneficiosa para aplicaciones de datos intensivos, así como también para redes de pequeña banda ancha.
- *Interfaces más sensitivas* – el mejoramiento del rendimiento conduce a interfaces mucho más sensitivas, lo que crea la ilusión de que la información se actualiza de forma instantánea. Como un resultado las aplicaciones Web que utilizan AJAX aparecen comportándose mucho como su contrapartida, las aplicaciones de escritorio.
- *Reduce o elimina el tiempo de espera* – en las aplicaciones basadas en AJAX solo los elementos más relevantes de la página son actualizados, y el resto de la página permanece inalterable. Este acercamiento elimina las pantallas blancas (o el parpadeo en conexiones rápidas) y disminuye de forma significativa el tiempo de espera.
- *Incremento de la usabilidad* – como las aplicaciones basadas en AJAX son más eficientes a través de AJAX, y el cliente puede comunicarse con el servidor sin la necesidad de cargar la página nuevamente. Se vuelve posible para edificar maravillosas interfaces de usuario que se adecuen a las necesidades y expectativas del usuario.
- *Los usuarios pueden trabajar con el resto de la página* – algunas aplicaciones con AJAX permiten que se continúe trabajando con el resto de la página, mientras la información está siendo transferida en trasfondo. Esto por otro lado suma de forma ininterrumpida la experiencia del usuario final. (8)

1.8.2 ASP.NET

ASP.NET fue desarrollado por Microsoft como parte de la estrategia .NET para el desarrollo Web, con el objetivo de resolver las limitaciones de ASP¹⁸ y posibilitar la creación de software como servicio. ASP.NET

¹⁸ ASP (Active Server Pages) es una tecnología del lado servidor desarrollado por Microsoft para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Server (IIS).

es la plataforma unificada de desarrollo web que proporciona a los desarrolladores los servicios necesarios para crear aplicaciones web empresariales.

Algunas de las características más notables de ASP.NET se mencionan a continuación:

- *Soporte de múltiples lenguajes compilados.* ASP.NET no descarta totalmente la idea de los lenguajes de scripting aunque esto suponga un problema a la hora de la interpretación del lenguaje; e introduce el soporte para lenguajes completamente compilados, dándole la posibilidad al desarrollador de escribir sus aplicaciones en cualquier lenguaje .NET propio de la plataforma (Visual C++, Visual Basic.NET, Visual C#.NET) o de escribirla en algún otro lenguaje compatible con la plataforma .NET (Perl, Eiffel, SmallTalk).
- *Mejor rendimiento.* Cuando es solicitada una página ASP.NET al servidor por primera vez, se precompila la clase derivada generada dinámicamente que representa a esa página a MSIL (Microsoft Intermediate Language). Después de compilar, la clase derivada es almacenada en un caché de salida, de modo que las peticiones subsiguientes a esa misma página se leen del caché, incrementándose el rendimiento después de la primera ejecución. Este mismo proceso ocurre cuando se solicitan otras páginas ASP.NET.
- *Mejora de la seguridad.* ASP.NET permite varios tipos de identificación y autenticación (Windows, Passport y Cookies). Mediante ASP.NET se puede programáticamente verificar el rol de un usuario determinado y de forma condicional permitirle ejecutar algunas tareas, si tiene autorización. Además, se pueden crear sistemas de autenticación basadas en formularios personalizados por el desarrollador de la aplicación, de forma muy sencilla.
- *Mayor escalabilidad.* La Web es fundamentalmente un modelo sin estado y sin correlación entre solicitudes HTTP¹⁹. Esto puede dificultar el proceso de desarrollo de aplicaciones Web, pues en muchas oportunidades es necesario mantener el estado entre varias peticiones. ASP.NET mejora

¹⁹ HTTP (Hyper Text Transfer Protocol) es decir, protocolo de transferencia de hipertexto. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla.

y extiende los servicios de gestión de estados introducidos por su antecesora ASP al proporcionar tres tipos de estado en las aplicaciones Web: aplicación, sesión y usuario.

- El estado de aplicación, como en ASP, es específico de la instancia de una aplicación y no es persistente.
- El estado de sesión es específico de una sesión de usuario con la aplicación, este estado en ASP.NET se almacena en un proceso separado que incluso puede configurarse para ser almacenado en una base de datos, en un proceso o en una máquina separada. Esto hace que se pueda utilizar el estado de sesión cuando la aplicación se implanta en una granja de servidores.
- El estado de usuario es similar al estado de sesión, pero generalmente no expira y es persistente. El estado de usuario es útil para almacenar preferencias de usuario y demás información de personalización. Todos los servicios de gestión de estado se implementan como módulos HTTP, de modo que pueden ser fácilmente añadidos o eliminados del flujo de una aplicación. Un módulo de terceros puede proporcionar servicios de gestión de estado con independencia de los proporcionados por ASP.NET si hubiese necesidad de ello.
- *Fácil configuración e implantación.* La configuración e implantación se simplifican con el uso de archivos de configuración directamente legibles que no necesitan ser registrados. ASP.NET utiliza una arquitectura de configuración jerárquica. La información de configuración de una aplicación ASP.NET (como el nombre de la aplicación que gestiona las peticiones HTTP, la información de permisos o las cadenas de conexión a bases de datos) se almacena en archivos textuales de configuración en formato XML denominados: web.config, que pueden ser ubicados en los mismos directorios que los archivos de aplicación. Si la aplicación tiene gran tamaño, directorios hijos pueden heredar la configuración del directorio padre a menos que sean sobrescritos por otros archivos web.config en el propio directorio hijo. Para implantar un sitio Web, todo lo que se necesita hacer es copiar de un sitio a otro la carpeta raíz de la aplicación web.

- Otro beneficio importante de ASP.NET es el soporte para la actualización "en vivo" de las aplicaciones. Un administrador no necesita detener el servidor Web ni la aplicación para actualizar la configuración. Los archivos de la aplicación nunca se bloquean, de modo que pueden sobrescribirse incluso cuando la aplicación está ejecutándose. El runtime²⁰ recogerá y propagará automáticamente cualquier cambio que se haya hecho. Cuando se actualizan los archivos, el sistema conmuta transparentemente a la nueva versión. El sistema detecta los cambios de los archivos, lanza una nueva instancia de la aplicación utilizando el nuevo código y empieza a rutear las nuevas peticiones entrantes a esa aplicación. Cuando todas las peticiones pendientes procesadas por la instancia de la aplicación existente han sido gestionadas, esa instancia se elimina. (9)

1.9 CONCLUSIONES

En este capítulo, se han definido los términos teleconsulta y telemedicina. Además, se ha mostrado de forma muy básica el funcionamiento de un sistema de teleconsultas. También se analizaron softwares que brindan servicios de telemedicina. Y para concluir se ha dado una caracterización de las herramientas, tecnologías y lenguajes a emplear para la creación e implementación de un sistema que posibilite la teleconsulta.

²⁰ Cuando un programa está corriendo o ejecutándose, se dice que está en runtime.

CAPÍTULO II

CARACTERÍSTICAS DEL SISTEMA

El objetivo fundamental de este capítulo es la propuesta de un sistema que resuelva el problema científico planteado en este trabajo. Se exponen las funcionalidades y atributos del sistema. Se muestran además los casos de uso y los diagramas de casos de uso del sistema.

2.1 OBJETO DE AUTOMATIZACIÓN

El objeto de automatización de este trabajo es el proceso de la teleconsulta, el cual se detalla a continuación, aunque ya se había dado una breve explicación del mismo en el [Epígrafe 1.1](#) del capítulo anterior.

El Sistema de Salud Pública de Cuba está constituido por tres niveles de atención: primaria, secundaria y terciaria. El primer nivel lo constituyen los policlínicos y médicos de familia; el segundo está constituido por los hospitales y el tercero por centros especializados de investigaciones médicas. La mejor forma para que se desarrolle el proceso de teleconsultas sería que todos los niveles brinden y reciban estos servicios entre ellos mismos y de unos a otros, en la figura 3 se muestra como debería ocurrir el proceso.

De esta forma, cualquier centro podría brindar o recibir servicios de teleconsulta y se aprovecharían al máximo los conocimientos y experiencias de todos, en función de brindar un mejor diagnóstico al paciente.

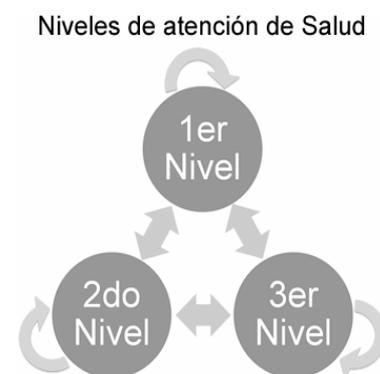


Fig. 3

2.2 PROPUESTA DE SISTEMA

En materia de telemedicina existe una tendencia a realizar sistemas para las videoconferencias, transmisión, visualización y almacenamiento de imágenes médica, también existen los sistemas que brindan servicios de teleconsulta pero estos son o muy específicos a alguna rama de la medicina, o no permiten que la teleconsulta se realice en tiempo real, además prevalecen las aplicaciones de escritorio sobre las aplicaciones web. Por tal motivo se propone realizar un sistema para la teleconsulta online.

Este sistema debe ser lo más general posible, para que varias ramas de la medicina que emplean el diagnóstico a través de estudios imagenológicos puedan hacer uso del sistema sin problemas. Además, debe estar disponible a toda hora todo el año y ser implementado para el entorno web, de forma tal que resuelva los detalles en contra de las aplicaciones web, para aprovechar la experiencia del usuario en las aplicaciones de escritorio. Esto se logra utilizando AJAX.

2.3 ESPECIFICACIÓN DE LOS REQUISITOS DEL SISTEMA

2.3.1 Funciones del sistema.

Las funciones del sistema o requisitos funcionales del sistema son todas aquellas funciones valga la redundancia, que el sistema debe poseer para que se cumpla con las necesidades para las que él fue creado.

A continuación, se identifican y se listan los requerimientos funcionales en grupos o paquetes de requerimientos cohesivos y lógicos:

Paquete	Bandeja de casos	
Referencia		Descripción
RF-1.1	Estado: Implementado Dificultad: Baja Prioridad: Alta	El sistema debe permitir que los ficheros DICOM sean trasladados hacia el servidor.
RF-1.2	Estado: Implementado Dificultad: Alta Prioridad: Alta	El sistema debe extraer información contenida dentro del fichero DICOM, y almacenar estos datos en una base de datos.

Paquete	Bandeja de casos	
Referencia		Descripción
RF-1.1	Estado: Implementado Dificultad: Baja Prioridad: Alta	El sistema debe permitir que los ficheros DICOM sean trasladados hacia el servidor.
RF-1.2	Estado: Implementado Dificultad: Alta Prioridad: Alta	El sistema debe extraer información contenida dentro del fichero DICOM, y almacenar estos datos en una base de datos.
RF-1.3	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe mostrar un listado de los pacientes que el Médico atiende (Bandeja de casos).
RF-1.4	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe mostrar los estudios que se ha realizado un paciente.
RF-1.5	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe mostrar por cada estudio la lista de series que este posea.
RF-1.6	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe mostrar un listado de las imágenes que contiene cada estudio.

Paquete	Paciente	
Referencia		Descripción
RF-2.1	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe mostrar un conjunto de datos relacionados al paciente seleccionado.
RF-2.2	Estado: Aprobado Dificultad: Media Prioridad: Alta	El sistema debe permitir que se elimine el Paciente seleccionado.
RF-2.3	Estado: Implementado Dificultad: Media Prioridad: Baja	El sistema debe permitir que se realicen anotaciones sobre el paciente.

Paquete	Estudio	
Referencia		Descripción
RF-3.1	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe mostrar un conjunto de datos relacionados al estudio seleccionado.
RF-3.2	Estado: Aprobado Dificultad: Media Prioridad: Alta	El sistema debe permitir que se elimine el Estudio seleccionado.
RF-3.3	Estado: Implementado Dificultad: Media Prioridad: Baja	El sistema debe permitir que se realicen anotaciones sobre el Estudio.

Paquete	Serie	
Referencia		Descripción
RF-4.1	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe mostrar un conjunto de datos relacionados a la Serie seleccionada.
RF-4.2	Estado: Aprobado Dificultad: Media Prioridad: Alta	El sistema debe permitir que se elimine la Serie seleccionada.
RF-4.3	Estado: Implementado Dificultad: Media Prioridad: Baja	El sistema debe permitir que se realicen anotaciones sobre la serie.

Paquete	Imagen	
Referencia		Descripción
RF-5.1	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe mostrar un conjunto de datos relacionados a la imagen seleccionada.
RF-5.2	Estado: Aprobado Dificultad: Media Prioridad: Alta	El sistema debe permitir que se elimine la imagen seleccionada.
RF-5.3	Estado: Implementado Dificultad: Media Prioridad: Baja	El sistema debe permitir que se realicen anotaciones sobre la imagen.

Paquete	Gestionar Contactos	
Referencia		Descripción
RF-6.1	Estado: Implementado Dificultad: Media Prioridad: Alta	El sistema debe permitir que se realicen búsquedas con el objetivo de encontrar a usuarios registrados en el sistema.
RF-6.2	Estado: Implementado Dificultad: Media Prioridad: Alta	El Sistema permitirá que se seleccionen los nombres de una lista de Usuarios con el objetivo de poder ser añadidos a la lista de contactos personal.
RF-6.3	Estado: Implementado Dificultad: Media Prioridad: Media	El Sistema debe permitir que se eliminen Usuarios de la Listas de Contactos.
RF-6.4	Estado: Aprobado Dificultad: Media Prioridad: Media	El Sistema permitirá que se renombren los contactos ya existentes en la lista.
RF-6.5	Estado: Implementado Dificultad: Alta Prioridad: Alta	El sistema debe permitir que el usuario pueda teleconsultar a sus contactos.

Paquete	Operaciones sobre las imágenes.	
Referencia		Descripción
RF-7.1	Estado: Implementado Dificultad: Alta Prioridad: Alta	El sistema debe permitir que se dibuje sobre la imagen.
RF-7.2	Estado: Implementado Dificultad: Media Prioridad: Media	El sistema debe permitir realizar zoom sobre las imágenes.
RF-7.3	Estado: Implementado Dificultad: Media Prioridad: Baja	El sistema debe permitir que se mueva la imagen por toda el área de trabajo.

Referencia		Descripción
RF-8	Estado: Implementado Dificultad: Media Prioridad: Baja	El sistema permitirá que el Médico edite su perfil de usuario y este sea almacenado de forma persistente.

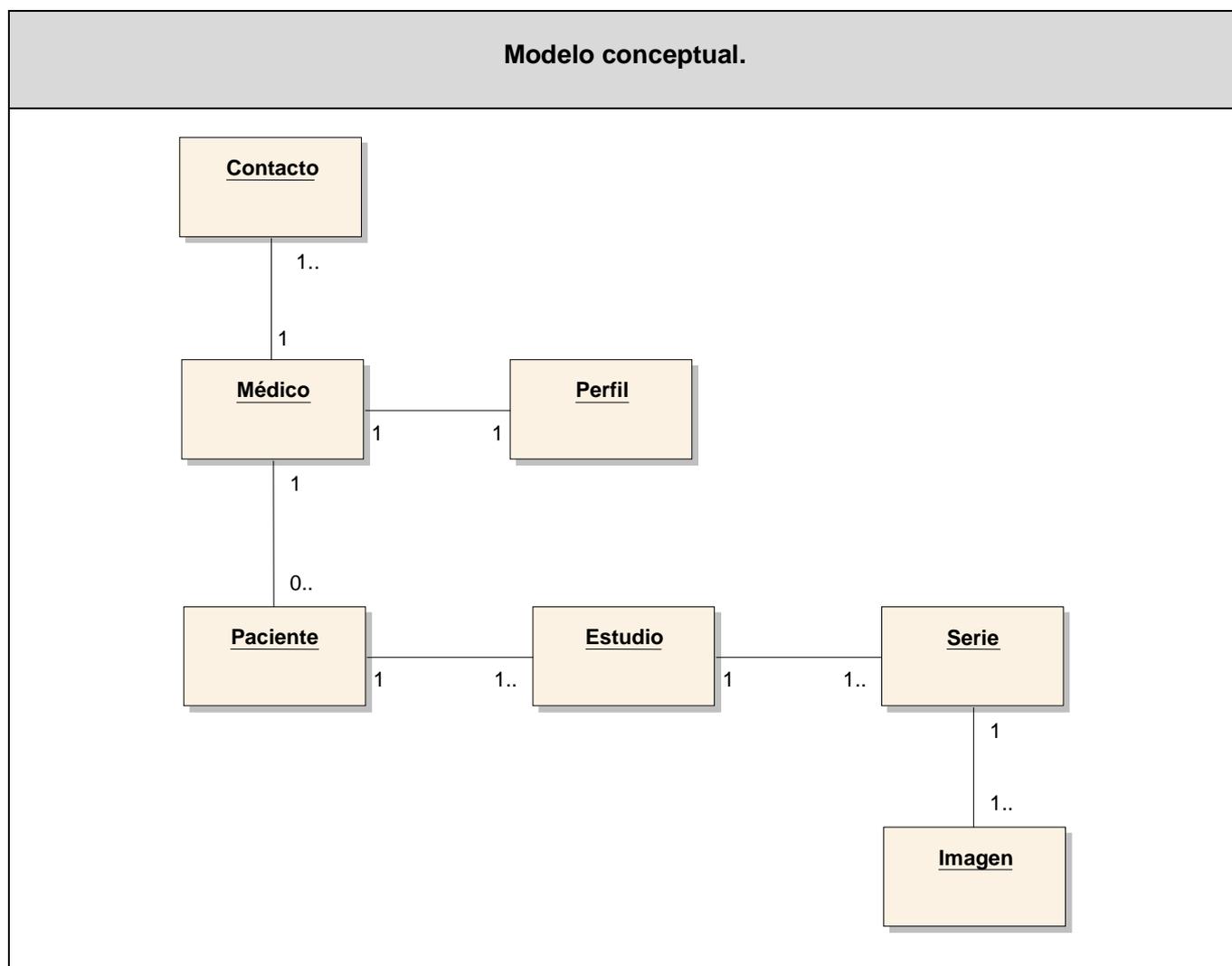
2.3.2 Atributos del sistema.

Los atributos del sistema son cualidades no funcionales que este tiene, es decir si el sistema no posee alguno de ellos el sistema continuará siendo funcional.

Atributo	Descripción
Usabilidad	Su acceso debe ser fácil y rápido.
Rendimiento	El sistema debe responder a un gran número de solicitudes en cortos instantes de tiempo (ej.: 1s).
Seguridad	El sistema será accedido por los siguientes usuarios: <ul style="list-style-type: none">• Administrador(es)• Usuarios registrados (Médicos)
Confiabilidad	El sistema debe estar disponible las 24 horas del día, y compartir entre usuarios solo la información que debe ser compartida.
Interfaz	La interfaz del sistema debe estar inspirada en las interfaces de las aplicaciones de escritorio.
Hardware	Servidor de aplicaciones: CPU Intel Xeon 2.8 GHz Dual; 2 Gb RAM Dual Channel y Gigabit Ethernet network adapter. El servidor que funcionara como fuente de datos, tendrá las mismas características que el primero.

2.4 MODELO CONCEPTUAL DEL SISTEMA

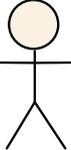
El modelo conceptual: es una representación de conceptos u objetos en el dominio del problema. Y para su construcción, se ha centrado la atención en identificar los conceptos obvios expresados en los requerimientos; de manera que por esta estrategia generalmente no se obtiene un modelo conceptual muy completo, pero esta primera versión servirá de base para posteriores modificaciones, en caso que sean necesarias.



2.5 DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA.

2.5.1 Actores del sistema.

Actores del sistema: son entidades externas del sistema que de alguna forma participa en la historia del caso de uso. Por lo regular estimula el sistema con eventos de entrada o recibe algo de el.

Actor	Descripción
	El Médico es el actor fundamental en el sistema ya que es quien inicia todos los casos de uso del sistema.

2.5.2 Lista de los casos de uso del sistema.

Lo que a continuación aparece es una breve descripción de los casos de uso del sistema, los casos expandidos de uso aparecen en el [Anexo I](#). Se han agrupado por paquetes, para mantener cierto orden y lógica.

Paquete: Bandeja de casos.

CU-01	Subir DICOM para el servidor
Actor	Médico
Descripción	Permite al Médico subir para el servidor los ficheros DICOM
Referencia	RF-1.1, RF-1.2, RF-1.3, RF-1.4, RF-1.5, RF-1.6

Paquete: Paciente.

CU-02	Seleccionar Paciente
Actor	Médico
Descripción	El Médico selecciona a uno de los Pacientes para obtener información a cerca de él.
Referencia	RF-2.1

CU-03	Eliminar Paciente
Actor	Médico
Descripción	El Médico elimina a un Paciente.
Referencia	CU-02, RF-2.2

CU-04	Realizar anotaciones en el Paciente
Actor	Médico
Descripción	El Médico realiza anotaciones al Paciente.
Referencia	CU-02, RF-2.3

Paquete: Estudio.

CU-05	Seleccionar Estudio
Actor	Médico
Descripción	El Médico selecciona a uno de los Estudios de algún paciente para obtener información a cerca del Estudio seleccionado.
Referencia	RF-3.1

CU-06	Eliminar Estudio
Actor	Médico
Descripción	El Médico elimina un Estudio.
Referencia	CU-05, RF-3.2

CU-07	Realizar anotaciones en el Estudio
Actor	Médico
Descripción	El Médico realiza anotaciones en el Estudio.
Referencia	CU-05, RF-3.3

Paquete: Serie.

CU-08	Seleccionar Serie
Actor	Médico
Descripción	El Médico selecciona una de las Serie para obtener información a cerca de esta.
Referencia	RF-4.1

CU-09	Eliminar Serie
Actor	Médico
Descripción	El Médico elimina una Serie.
Referencia	CU-08, RF-4.2

CU-10	Realizar anotaciones en la Serie
Actor	Médico
Descripción	El Médico realiza anotaciones a la Serie.
Referencia	CU-08, RF-4.3

Paquete: Imagen.

CU-11	Seleccionar Imagen
Actor	Médico
Descripción	El Médico selecciona a una de las Imágenes para obtener información a cerca de esta.
Referencia	RF-5.1

CU-12	Eliminar Imagen
Actor	Médico
Descripción	El Médico elimina una Imagen.
Referencia	CU-11, RF-5.2

CU-13	Realizar anotaciones en la Imagen
Actor	Médico
Descripción	El Médico realiza anotaciones a la Imagen.
Referencia	CU-11, RF-5.3

Paquete: Contactos.

CU-14	Buscar usuario
Actor	Médico
Descripción	Buscar usuarios registrados en el sistema por la categoría: nombre de usuario.
Referencia	RF-6.1

CU-15	Añadir contacto
Actor	Médico
Descripción	El Médico añade a su lista de contactos a otros usuarios.
Referencia	RF-6.2

CU-16	Eliminar contacto
Actor	Médico
Descripción	El Médico elimina de su lista de contactos a otros usuarios.
Referencia	RF-6.3

CU-17	Renombrar contacto
Actor	Médico
Descripción	El Médico cambia el nombre a los contactos de su lista.
Referencia	RF-6.4

CU-18	Teleconsultar contacto
Actor	Médico
Descripción	El Médico pide una segunda opinión a otro.
Referencia	RF-6.5, CU-11, RF-7.1

Paquete: Operaciones con las imágenes.

CU-19	Hacer zoom sobre la imagen.
Actor	Médico
Descripción	El Médico aumenta o disminuye el tamaño de la imagen.
Referencia	CU-11, RF-7.2

CU-20	Mover la imagen
Actor	Médico
Descripción	El Médico arrastra la imagen por toda el área de trabajo.
Referencia	CU-11, RF-7.3

CU-21	Editar perfil.
Actor	Médico
Descripción	El Médico edita sus datos personales.
Referencia	RF-8

2.5.3 Diagramas de los casos de uso del sistema.

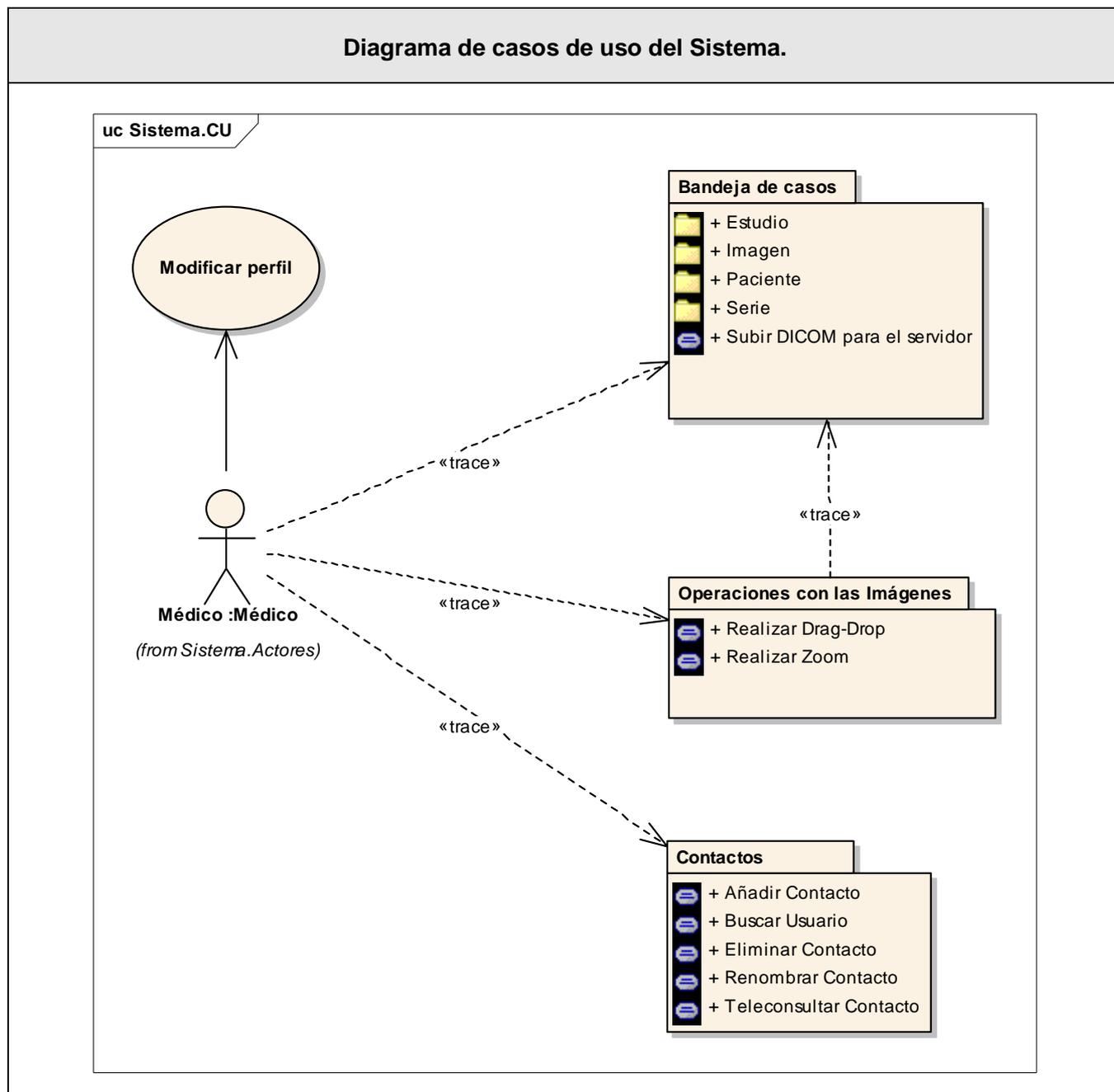


DIAGRAMA DE CASOS DE USO (Paquete: Bandeja de casos)

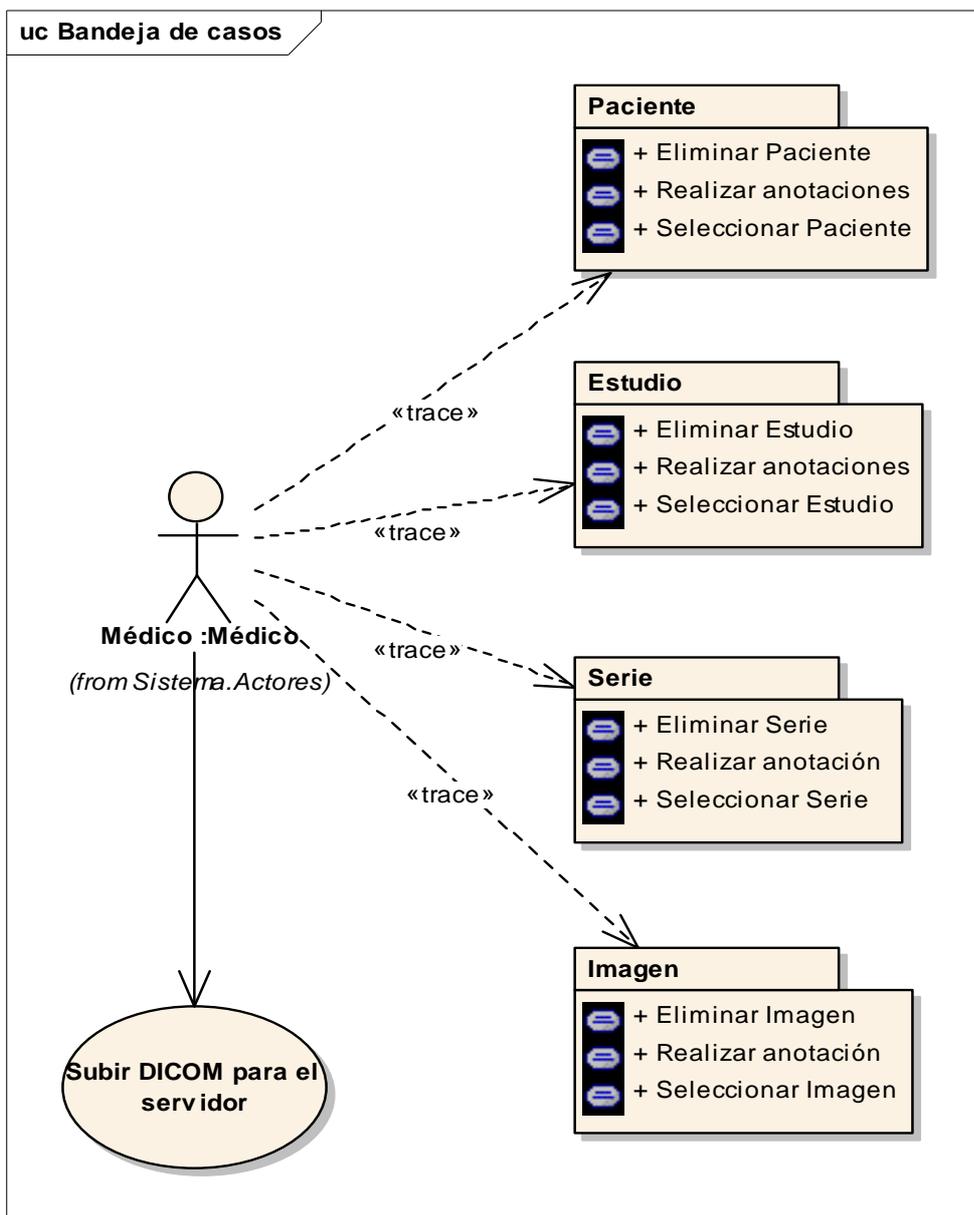


DIAGRAMA DE CASOS DE USO (Paquete: Paciente)

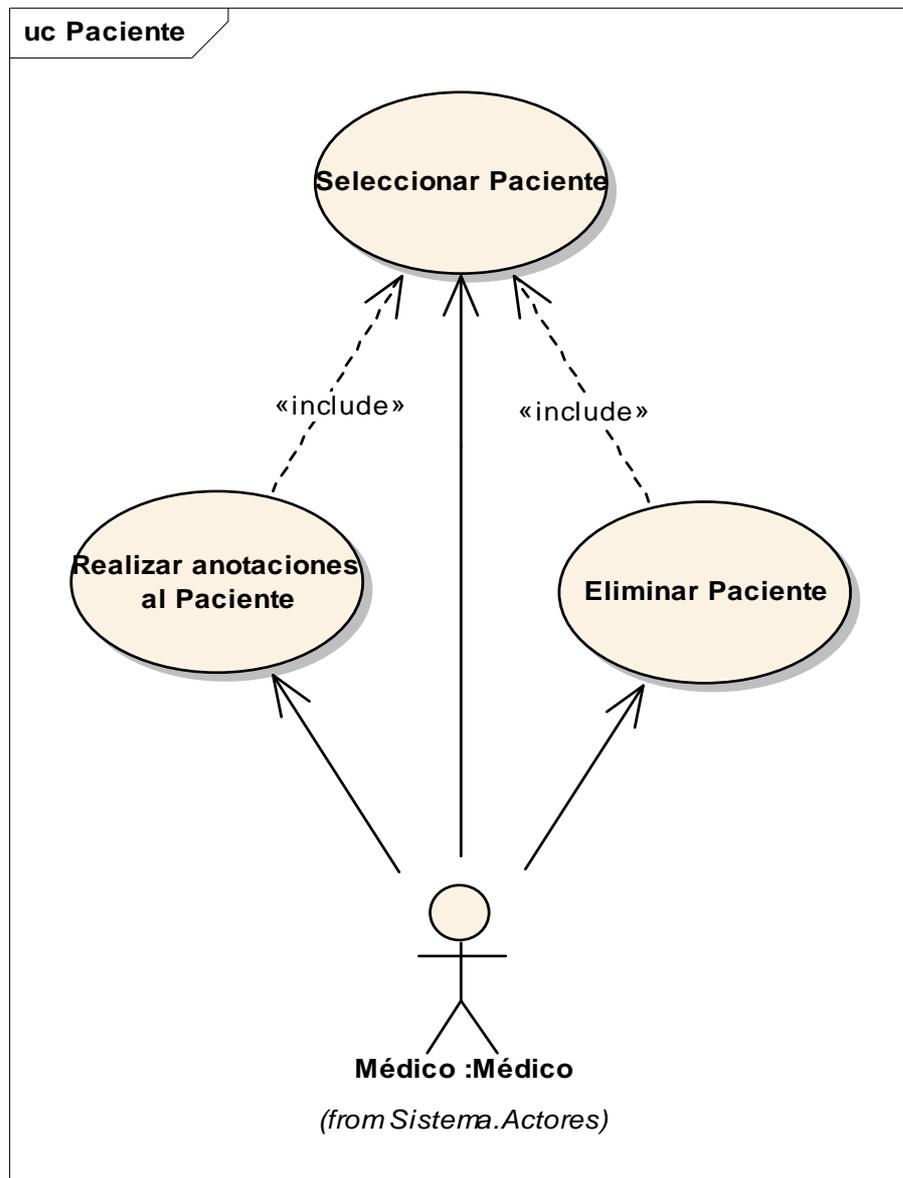


DIAGRAMA DE CASOS DE USO (Paquete: Estudio)

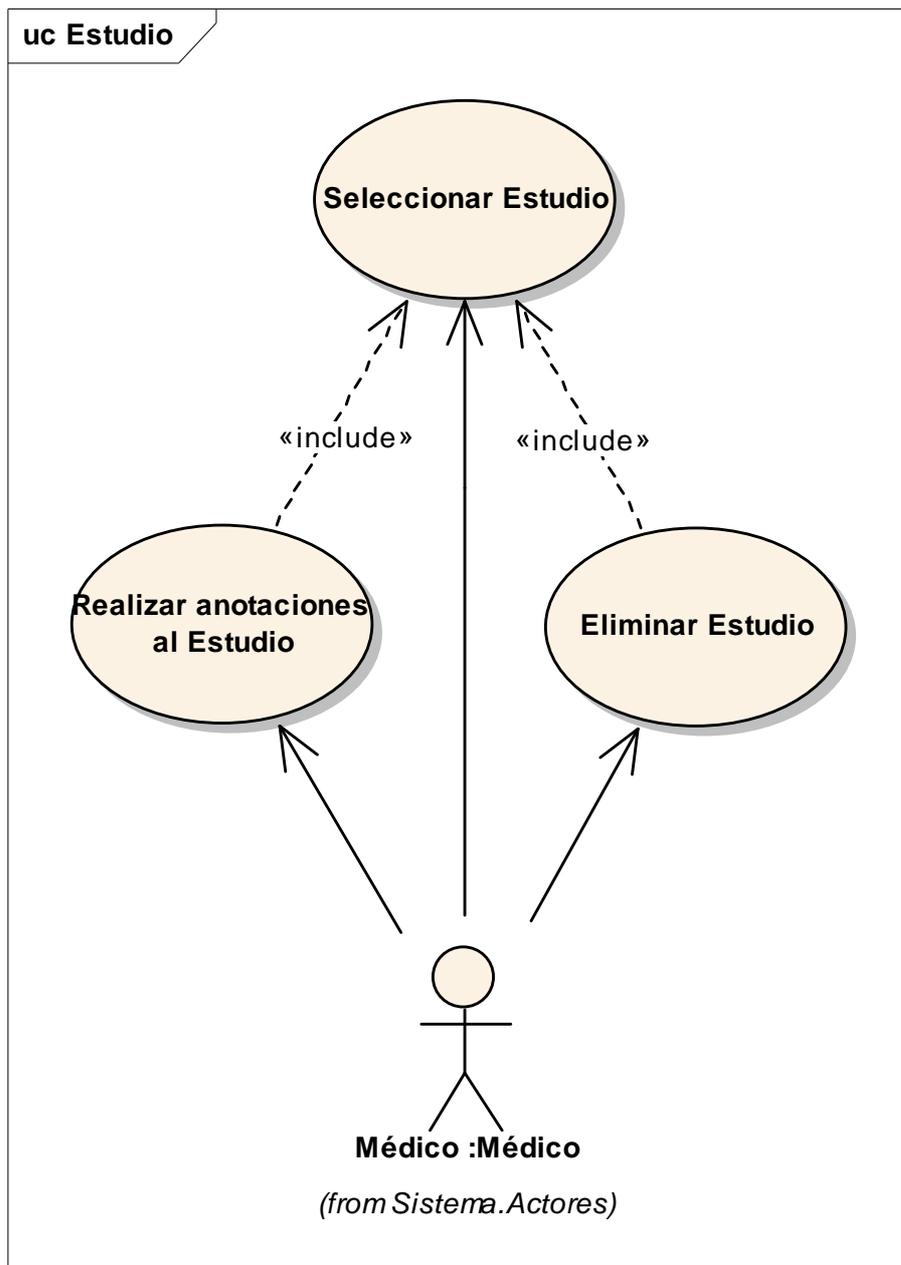


DIAGRAMA DE CASOS DE USO (Paquete: Serie)

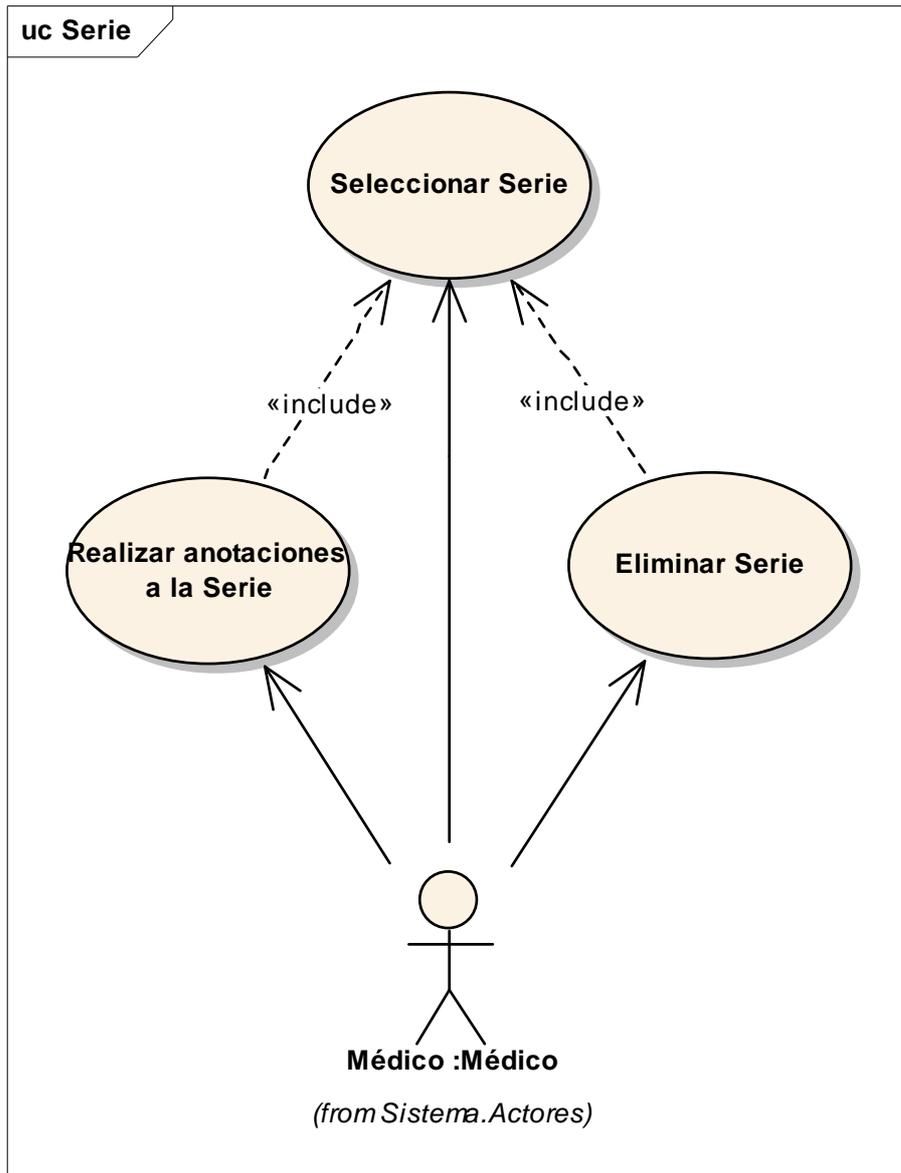


DIAGRAMA DE CASOS DE USO (Paquete: Imagen)

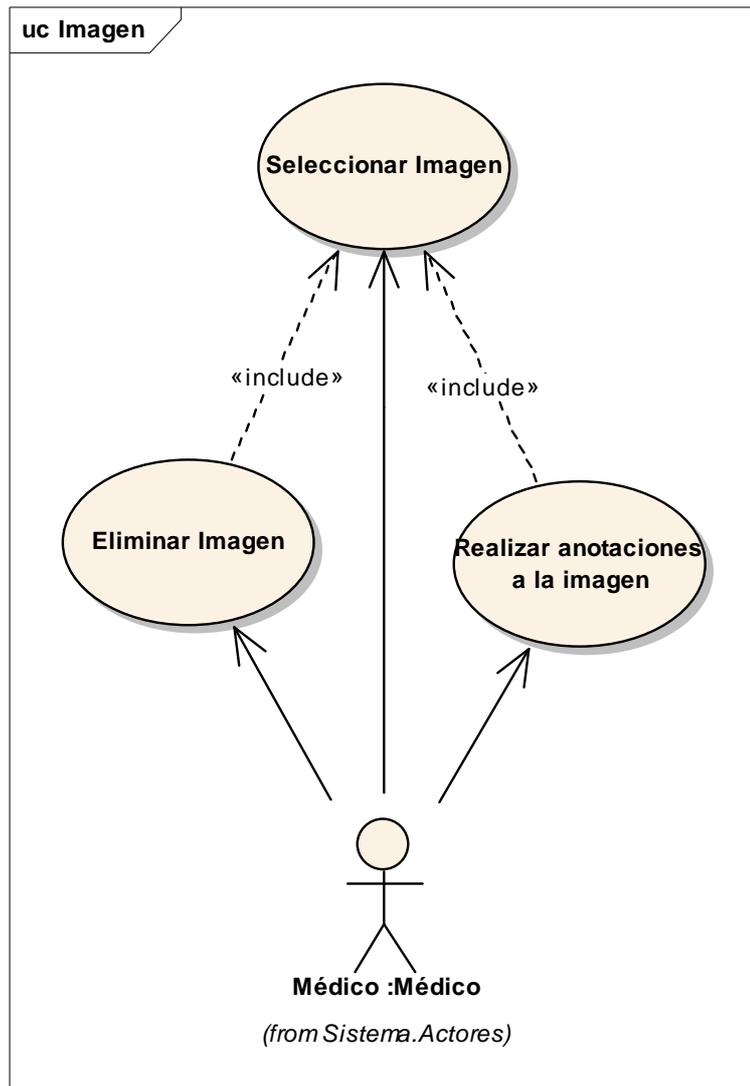


DIAGRAMA DE CASOS DE USO (Paquete: Contactos)

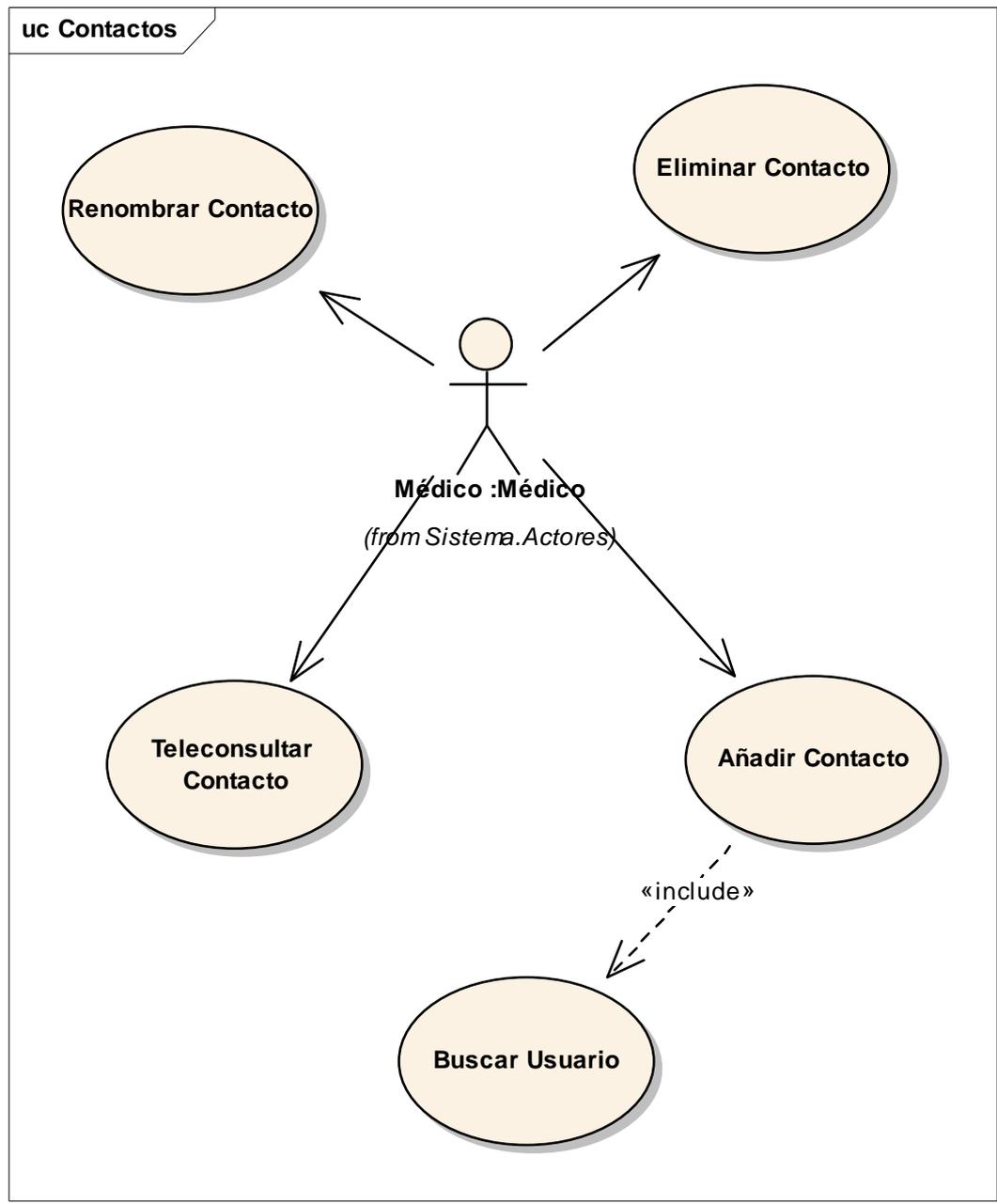
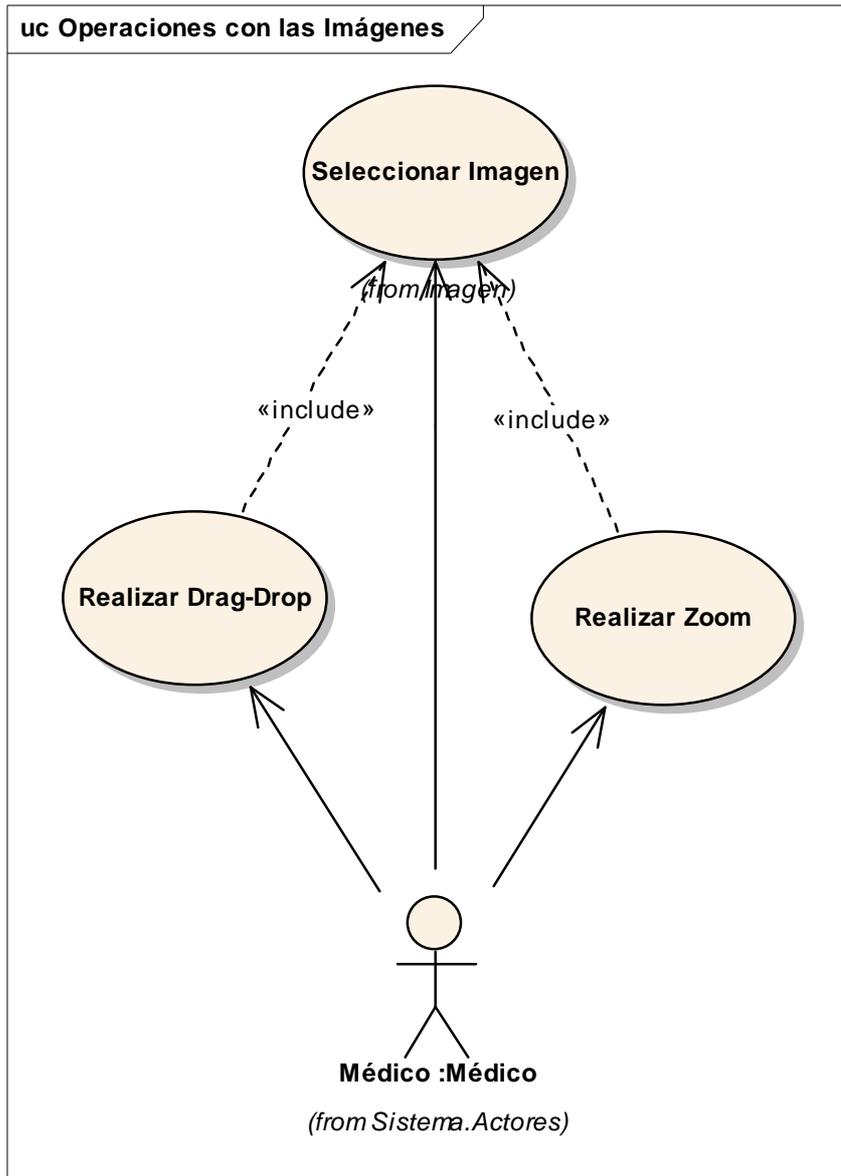


DIAGRAMA DE CASOS DE USO (Paquete: Operaciones con las Imágenes)



2.6 CONCLUSIONES

En este capítulo se propuso la solución al problema planteado, mediante el sistema para la teleconsulta online presentado. Se expusieron las funcionalidades y atributos, los casos de uso, así como los diagramas de casos de uso del sistema.

CAPÍTULO III

ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se realiza el análisis y el diseño del sistema, además se muestran diferentes diagramas correspondientes a estas dos fases como por ejemplo los diagramas de clases del análisis y los diagramas de clases del diseño, también se muestran las descripciones de las clases y las descripciones de las tablas del modelo de datos.

3.1 ANÁLISIS

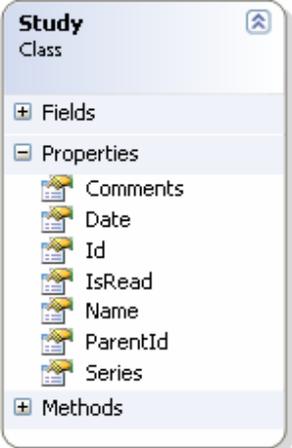
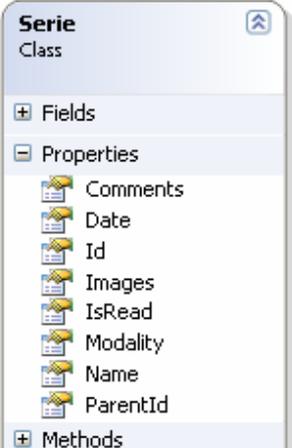
Los diagramas de clases correspondientes al análisis se pueden ver en el [Anexo II](#).

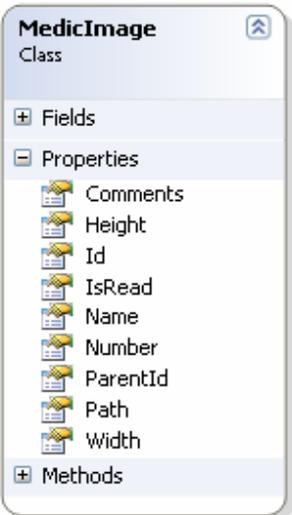
3.2 DISEÑO

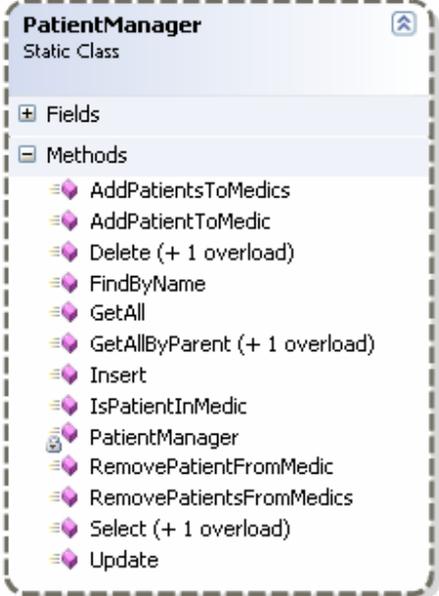
Los diagramas de clases del diseño aparecen en el [Anexo III](#)

3.2.1 Descripción de las clases.

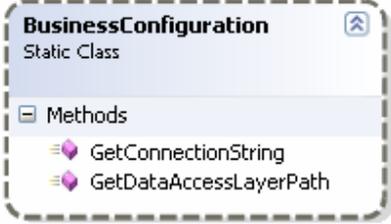
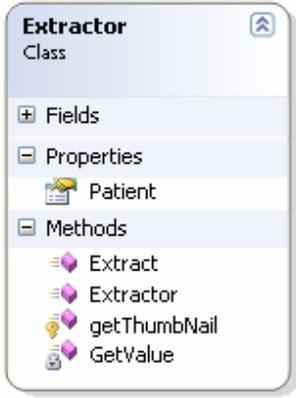
Entidades del Negocio																	
Clase	Descripción																
 <p>The screenshot shows a class named 'Patient' with the following properties listed: Birthdate, Comments, Id, IsRead, Name, ParentId, Sex, and Studies. There are also expandable sections for 'Fields' and 'Methods'.</p>	<p>Encapsula la información referente al paciente.</p> <table border="1"> <thead> <tr> <th colspan="2">Propiedad</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Birthdate</td> <td>Devuelve o recibe la fecha de nacimiento del paciente.</td> </tr> <tr> <td>Comments</td> <td>Devuelve o recibe los comentarios o anotaciones realizadas por le Médico.</td> </tr> <tr> <td>Id</td> <td>Devuelve o recibe el identificador del paciente.</td> </tr> <tr> <td>IsRead</td> <td>Devuelve verdadero si todos los estudios que tiene el paciente han sido revisados, de lo contrario devuelve falso</td> </tr> <tr> <td>Name</td> <td>Devuelve o recibe el nombre del paciente.</td> </tr> <tr> <td>ParentId</td> <td>Devuelve o recibe el identificador del</td> </tr> </tbody> </table>	Propiedad		Nombre	Descripción	Birthdate	Devuelve o recibe la fecha de nacimiento del paciente.	Comments	Devuelve o recibe los comentarios o anotaciones realizadas por le Médico .	Id	Devuelve o recibe el identificador del paciente.	IsRead	Devuelve verdadero si todos los estudios que tiene el paciente han sido revisados, de lo contrario devuelve falso	Name	Devuelve o recibe el nombre del paciente.	ParentId	Devuelve o recibe el identificador del
Propiedad																	
Nombre	Descripción																
Birthdate	Devuelve o recibe la fecha de nacimiento del paciente.																
Comments	Devuelve o recibe los comentarios o anotaciones realizadas por le Médico .																
Id	Devuelve o recibe el identificador del paciente.																
IsRead	Devuelve verdadero si todos los estudios que tiene el paciente han sido revisados, de lo contrario devuelve falso																
Name	Devuelve o recibe el nombre del paciente.																
ParentId	Devuelve o recibe el identificador del																

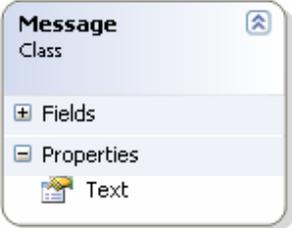
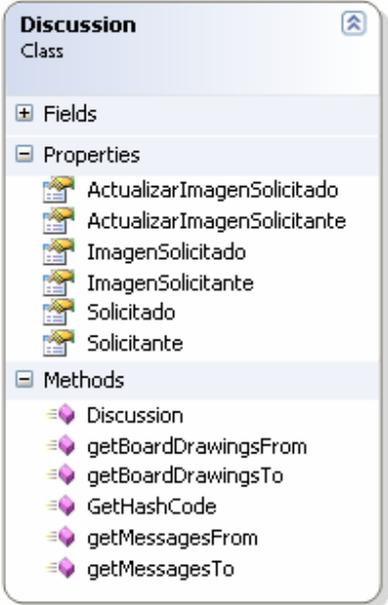
	<table border="1"> <tr> <td></td> <td>Médico que atiende al paciente.</td> </tr> <tr> <td>Sex</td> <td>Devuelve o recibe el sexo del paciente.</td> </tr> <tr> <td>Studies</td> <td>Devuelve o recibe una colección de estudios.</td> </tr> </table>		Médico que atiende al paciente.	Sex	Devuelve o recibe el sexo del paciente.	Studies	Devuelve o recibe una colección de estudios.												
	Médico que atiende al paciente.																		
Sex	Devuelve o recibe el sexo del paciente.																		
Studies	Devuelve o recibe una colección de estudios.																		
	<p>Encapsula la información referente a un estudio del paciente.</p> <table border="1"> <thead> <tr> <th colspan="2">Propiedad</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Comments</td> <td>Devuelve o recibe los comentarios o anotaciones realizadas por le Médico.</td> </tr> <tr> <td>Date</td> <td>Devuelve o recibe la fecha de realización del estudio</td> </tr> <tr> <td>Id</td> <td>Devuelve o recibe el identificador del estudio.</td> </tr> <tr> <td>IsRead</td> <td>Devuelve verdadero si todas las series que tiene el estudio han sido revisadas, en caso contrario devuelve falso.</td> </tr> <tr> <td>Name</td> <td>Devuelve o recibe el nombre del estudio.</td> </tr> <tr> <td>ParentId</td> <td>Devuelve o recibe el identificador del paciente al que pertenece el estudio.</td> </tr> <tr> <td>Series</td> <td>Devuelve o recibe una colección de series.</td> </tr> </tbody> </table>	Propiedad		Nombre	Descripción	Comments	Devuelve o recibe los comentarios o anotaciones realizadas por le Médico .	Date	Devuelve o recibe la fecha de realización del estudio	Id	Devuelve o recibe el identificador del estudio.	IsRead	Devuelve verdadero si todas las series que tiene el estudio han sido revisadas, en caso contrario devuelve falso.	Name	Devuelve o recibe el nombre del estudio.	ParentId	Devuelve o recibe el identificador del paciente al que pertenece el estudio.	Series	Devuelve o recibe una colección de series.
Propiedad																			
Nombre	Descripción																		
Comments	Devuelve o recibe los comentarios o anotaciones realizadas por le Médico .																		
Date	Devuelve o recibe la fecha de realización del estudio																		
Id	Devuelve o recibe el identificador del estudio.																		
IsRead	Devuelve verdadero si todas las series que tiene el estudio han sido revisadas, en caso contrario devuelve falso.																		
Name	Devuelve o recibe el nombre del estudio.																		
ParentId	Devuelve o recibe el identificador del paciente al que pertenece el estudio.																		
Series	Devuelve o recibe una colección de series.																		
	<p>Encapsula la información referente a una serie de un estudio.</p> <table border="1"> <thead> <tr> <th colspan="2">Propiedad</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Comments</td> <td>Devuelve o recibe los comentarios o anotaciones realizadas por le Médico.</td> </tr> <tr> <td>Date</td> <td>Devuelve o recibe la fecha de realización de la serie.</td> </tr> <tr> <td>Id</td> <td>Devuelve o recibe el identificador de la serie.</td> </tr> <tr> <td>Images</td> <td>Devuelve o recibe una colección de imágenes.</td> </tr> <tr> <td>IsRead</td> <td>Devuelve verdadero si todas las imágenes que tiene la serie han sido revisadas, en caso contrario devuelve falso.</td> </tr> <tr> <td>Modality</td> <td>Devuelve o recibe la modalidad de la serie.</td> </tr> <tr> <td>Name</td> <td>Devuelve o recibe el nombre de la serie.</td> </tr> </tbody> </table>	Propiedad		Nombre	Descripción	Comments	Devuelve o recibe los comentarios o anotaciones realizadas por le Médico .	Date	Devuelve o recibe la fecha de realización de la serie.	Id	Devuelve o recibe el identificador de la serie.	Images	Devuelve o recibe una colección de imágenes.	IsRead	Devuelve verdadero si todas las imágenes que tiene la serie han sido revisadas, en caso contrario devuelve falso.	Modality	Devuelve o recibe la modalidad de la serie.	Name	Devuelve o recibe el nombre de la serie.
Propiedad																			
Nombre	Descripción																		
Comments	Devuelve o recibe los comentarios o anotaciones realizadas por le Médico .																		
Date	Devuelve o recibe la fecha de realización de la serie.																		
Id	Devuelve o recibe el identificador de la serie.																		
Images	Devuelve o recibe una colección de imágenes.																		
IsRead	Devuelve verdadero si todas las imágenes que tiene la serie han sido revisadas, en caso contrario devuelve falso.																		
Modality	Devuelve o recibe la modalidad de la serie.																		
Name	Devuelve o recibe el nombre de la serie.																		

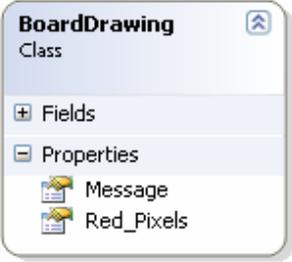
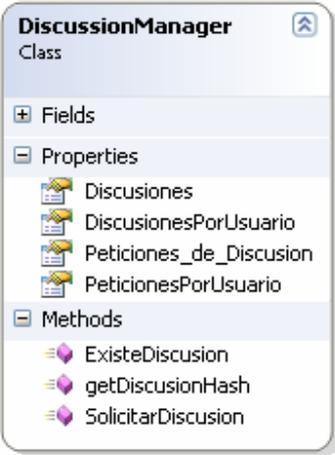
	ParentId	Devuelve o recibe el identificador del estudio al que pertenece la serie.
	Encapsula la información referente a una imagen de una serie.	
	Propiedad	
	Nombre	Descripción
	Comments	Devuelve o recibe los comentarios o anotaciones realizadas por le Médico .
	Height	Altura de la imagen en píxeles.
	Id	Devuelve o recibe el identificador de la imagen.
	IsRead	Devuelve verdadero si la imagen ha sido revisada, en caso contrario devuelve falso.
	Name	Devuelve o recibe el nombre de la imagen.
	Number	Número de orden de la imagen en el fichero DICOM
	ParentId	Devuelve o recibe el identificador de la serie a la que pertenece la imagen.
	Path	Ruta para la imagen respecto a la raíz de la aplicación.
	Width	Ancho de la imagen en píxeles.

Clases del Negocio																															
Clase	Descripción																														
	<p>Clase estática que se especializa en el manejo de de la clase entidad: <i>Patient</i>.</p> <table border="1"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>AddPatientsToMedics</td> <td>Adicionar varios pacientes a varios médicos.</td> </tr> <tr> <td>AddPatientToMedic</td> <td>Adiciona un paciente a un médico.</td> </tr> <tr> <td>Delete</td> <td>Eliminar pacientes</td> </tr> <tr> <td>FindByName</td> <td>Buscar pacientes por sus nombres. Devuelve una colección con el resultado</td> </tr> <tr> <td>GetAll</td> <td>Devolver una colección con todos los pacientes.</td> </tr> <tr> <td>GetAllByParent</td> <td>Devolver todos los pacientes que pertenecen a un médico.</td> </tr> <tr> <td>Insert</td> <td>Adicionar un paciente.</td> </tr> <tr> <td>IsPatientInMedic</td> <td>Devolver verdadero si un paciente es atendido por un médico determinado.</td> </tr> <tr> <td>PatientManager</td> <td>Constructor.</td> </tr> <tr> <td>RemovePatientFromMedic</td> <td>Eliminar a un paciente de la lista de pacientes de un médico determinado.</td> </tr> <tr> <td>RemovePatientsFromMedics</td> <td></td> </tr> <tr> <td>Select</td> <td>Devolver a un paciente dado su identificador.</td> </tr> <tr> <td>Update</td> <td>Actualizar los datos del paciente.</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	AddPatientsToMedics	Adicionar varios pacientes a varios médicos.	AddPatientToMedic	Adiciona un paciente a un médico.	Delete	Eliminar pacientes	FindByName	Buscar pacientes por sus nombres. Devuelve una colección con el resultado	GetAll	Devolver una colección con todos los pacientes.	GetAllByParent	Devolver todos los pacientes que pertenecen a un médico.	Insert	Adicionar un paciente.	IsPatientInMedic	Devolver verdadero si un paciente es atendido por un médico determinado.	PatientManager	Constructor.	RemovePatientFromMedic	Eliminar a un paciente de la lista de pacientes de un médico determinado.	RemovePatientsFromMedics		Select	Devolver a un paciente dado su identificador.	Update	Actualizar los datos del paciente.
Métodos																															
Nombre	Descripción																														
AddPatientsToMedics	Adicionar varios pacientes a varios médicos.																														
AddPatientToMedic	Adiciona un paciente a un médico.																														
Delete	Eliminar pacientes																														
FindByName	Buscar pacientes por sus nombres. Devuelve una colección con el resultado																														
GetAll	Devolver una colección con todos los pacientes.																														
GetAllByParent	Devolver todos los pacientes que pertenecen a un médico.																														
Insert	Adicionar un paciente.																														
IsPatientInMedic	Devolver verdadero si un paciente es atendido por un médico determinado.																														
PatientManager	Constructor.																														
RemovePatientFromMedic	Eliminar a un paciente de la lista de pacientes de un médico determinado.																														
RemovePatientsFromMedics																															
Select	Devolver a un paciente dado su identificador.																														
Update	Actualizar los datos del paciente.																														

 <p>StudyManager Static Class</p> <p>Fields</p> <p>Methods</p> <ul style="list-style-type: none"> Delete (+ 1 overload) FindByName GetAll GetAllByParent (+ 1 overload) Insert Select (+ 1 overload) StudyManager Update 	<p>Clase estática que se especializa en el manejo de de la clase entidad: <i>Study</i>.</p> <table border="1"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Delete</td> <td>Eliminar un estudio.</td> </tr> <tr> <td>FindByName</td> <td>Buscar estudios por sus nombres. Devuelve una colección con el resultado.</td> </tr> <tr> <td>GetAll</td> <td>Devolver una colección con todos los estudios.</td> </tr> <tr> <td>GetAllByParent</td> <td>Devolver una colección con todos los estudios de un paciente.</td> </tr> <tr> <td>Insert</td> <td>Adicionar un estudio.</td> </tr> <tr> <td>Select</td> <td>Devolver un estudio dado su identificador</td> </tr> <tr> <td>StudyManager</td> <td>Constructor</td> </tr> <tr> <td>Update</td> <td>Actualizar un estudio.</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	Delete	Eliminar un estudio.	FindByName	Buscar estudios por sus nombres. Devuelve una colección con el resultado.	GetAll	Devolver una colección con todos los estudios.	GetAllByParent	Devolver una colección con todos los estudios de un paciente.	Insert	Adicionar un estudio.	Select	Devolver un estudio dado su identificador	StudyManager	Constructor	Update	Actualizar un estudio.
Métodos																					
Nombre	Descripción																				
Delete	Eliminar un estudio.																				
FindByName	Buscar estudios por sus nombres. Devuelve una colección con el resultado.																				
GetAll	Devolver una colección con todos los estudios.																				
GetAllByParent	Devolver una colección con todos los estudios de un paciente.																				
Insert	Adicionar un estudio.																				
Select	Devolver un estudio dado su identificador																				
StudyManager	Constructor																				
Update	Actualizar un estudio.																				
 <p>SerieManager Static Class</p> <p>Fields</p> <p>Methods</p> <ul style="list-style-type: none"> Delete (+ 1 overload) FindByName GetAll GetAllByParent (+ 1 overload) Insert Select (+ 1 overload) SerieManager Update 	<p>Clase estática que se especializa en el manejo de de la clase entidad: <i>Serie</i>.</p> <table border="1"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Delete</td> <td>Eliminar una serie.</td> </tr> <tr> <td>FindByName</td> <td>Buscar series por sus nombres. Devuelve una colección con el resultado.</td> </tr> <tr> <td>GetAll</td> <td>Devolver una colección con todas las series.</td> </tr> <tr> <td>GetAllByParent</td> <td>Devolver una colección con todas las series de un estudio.</td> </tr> <tr> <td>Insert</td> <td>Adicionar una serie.</td> </tr> <tr> <td>Select</td> <td>Devolver una serie dado su identificador</td> </tr> <tr> <td>SerieManager</td> <td>Constructor</td> </tr> <tr> <td>Update</td> <td>Actualizar una serie.</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	Delete	Eliminar una serie.	FindByName	Buscar series por sus nombres. Devuelve una colección con el resultado.	GetAll	Devolver una colección con todas las series.	GetAllByParent	Devolver una colección con todas las series de un estudio.	Insert	Adicionar una serie.	Select	Devolver una serie dado su identificador	SerieManager	Constructor	Update	Actualizar una serie.
Métodos																					
Nombre	Descripción																				
Delete	Eliminar una serie.																				
FindByName	Buscar series por sus nombres. Devuelve una colección con el resultado.																				
GetAll	Devolver una colección con todas las series.																				
GetAllByParent	Devolver una colección con todas las series de un estudio.																				
Insert	Adicionar una serie.																				
Select	Devolver una serie dado su identificador																				
SerieManager	Constructor																				
Update	Actualizar una serie.																				

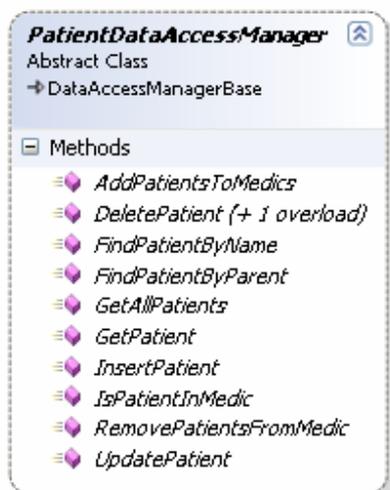
	<p>Clase estática que se especializa en el manejo de de la clase entidad: <i>MedicImage</i>.</p> <table border="1"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Delete</td> <td>Eliminar una imagen.</td> </tr> <tr> <td>FindByName</td> <td>Buscar imágenes por sus nombres. Devuelve una colección con el resultado.</td> </tr> <tr> <td>GetAll</td> <td>Devolver una colección con todas las imágenes.</td> </tr> <tr> <td>GetAllByParent</td> <td>Devolver una colección con todas las imágenes de una serie.</td> </tr> <tr> <td>Insert</td> <td>Adicionar una imagen.</td> </tr> <tr> <td>Select</td> <td>Devolver una imagen dado su identificador</td> </tr> <tr> <td>MedicImageManager</td> <td>Constructor</td> </tr> <tr> <td>Update</td> <td>Actualizar una imagen.</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	Delete	Eliminar una imagen.	FindByName	Buscar imágenes por sus nombres. Devuelve una colección con el resultado.	GetAll	Devolver una colección con todas las imágenes.	GetAllByParent	Devolver una colección con todas las imágenes de una serie.	Insert	Adicionar una imagen.	Select	Devolver una imagen dado su identificador	MedicImageManager	Constructor	Update	Actualizar una imagen.
Métodos																					
Nombre	Descripción																				
Delete	Eliminar una imagen.																				
FindByName	Buscar imágenes por sus nombres. Devuelve una colección con el resultado.																				
GetAll	Devolver una colección con todas las imágenes.																				
GetAllByParent	Devolver una colección con todas las imágenes de una serie.																				
Insert	Adicionar una imagen.																				
Select	Devolver una imagen dado su identificador																				
MedicImageManager	Constructor																				
Update	Actualizar una imagen.																				
	<p>Clase estática que se encarga de la lectura de las configuraciones en el fichero de configuración de la aplicación (web.config)</p> <table border="1"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>GetConnectionString</td> <td>Devolver la cadena de conexión de la base de datos.</td> </tr> <tr> <td>GetDataAccessLayerPath</td> <td>Devolver la dirección física de la capa de acceso a datos.</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	GetConnectionString	Devolver la cadena de conexión de la base de datos.	GetDataAccessLayerPath	Devolver la dirección física de la capa de acceso a datos.												
Métodos																					
Nombre	Descripción																				
GetConnectionString	Devolver la cadena de conexión de la base de datos.																				
GetDataAccessLayerPath	Devolver la dirección física de la capa de acceso a datos.																				
	<p>Clase que encapsula la funcionalidad de extraer la información referente al estudio de un paciente de un fichero DICOM</p> <table border="1"> <thead> <tr> <th colspan="2">Propiedades</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Patient</td> <td>Devuelve los datos de paciente, así como toda la información que de él depende.</td> </tr> </tbody> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Extract</td> <td>Extrae la información contenida en un fichero DICOM.</td> </tr> <tr> <td>Extractor</td> <td>Constructor</td> </tr> <tr> <td>getThumbNail</td> <td>Devuelve una imagen en miniatura.</td> </tr> </tbody> </table>	Propiedades		Nombre	Descripción	Patient	Devuelve los datos de paciente, así como toda la información que de él depende.	Métodos		Nombre	Descripción	Extract	Extrae la información contenida en un fichero DICOM.	Extractor	Constructor	getThumbNail	Devuelve una imagen en miniatura.				
Propiedades																					
Nombre	Descripción																				
Patient	Devuelve los datos de paciente, así como toda la información que de él depende.																				
Métodos																					
Nombre	Descripción																				
Extract	Extrae la información contenida en un fichero DICOM.																				
Extractor	Constructor																				
getThumbNail	Devuelve una imagen en miniatura.																				

	<p>Clase que representa un mensaje.</p>																																
	<p>Clase que representa un intercambio de opiniones entre el teleconsultante y el teleconsultor.</p>																																
<table border="1"> <thead> <tr> <th colspan="2">Propiedades</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Text</td> <td>Devuelve o recibe una cadena de caracteres.</td> </tr> </tbody> </table>		Propiedades		Nombre	Descripción	Text	Devuelve o recibe una cadena de caracteres.																										
Propiedades																																	
Nombre	Descripción																																
Text	Devuelve o recibe una cadena de caracteres.																																
<table border="1"> <thead> <tr> <th colspan="2">Propiedades</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>ActualizarImagenSolicitado</td> <td>Devuelve o recibe verdadero si es necesaria la actualización de la imagen del médico solicitado, de lo contrario falso.</td> </tr> <tr> <td>ActualizarImagenSolicitante</td> <td>Devuelve o recibe verdadero si es necesaria la actualización de la imagen del médico solicitante, de lo contrario falso.</td> </tr> <tr> <td>ImagenSolicitado</td> <td>Devuelve o recibe la ruta de la imagen del médico solicitado</td> </tr> <tr> <td>ImagenSolicitante</td> <td>Devuelve o recibe la ruta de la imagen del médico solicitante</td> </tr> <tr> <td>Solicitado</td> <td>Devuelve o recibe el nombre Médico solicitado</td> </tr> <tr> <td>Solicitante</td> <td>Devuelve o recibe el nombre Médico solicitante</td> </tr> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> <tr> <td>Discussion</td> <td>Constructor</td> </tr> <tr> <td>getBoardDrawingsFrom</td> <td></td> </tr> <tr> <td>getBoardDrawingsTo</td> <td></td> </tr> <tr> <td>GetHashCode</td> <td></td> </tr> <tr> <td>getMessagesFrom</td> <td>Devuelve la cola de mensajes desde quien la envía.</td> </tr> <tr> <td>getMessagesTo</td> <td>Devuelve la cola de mensajes a quien la envía.</td> </tr> </tbody> </table>		Propiedades		Nombre	Descripción	ActualizarImagenSolicitado	Devuelve o recibe verdadero si es necesaria la actualización de la imagen del médico solicitado, de lo contrario falso.	ActualizarImagenSolicitante	Devuelve o recibe verdadero si es necesaria la actualización de la imagen del médico solicitante, de lo contrario falso.	ImagenSolicitado	Devuelve o recibe la ruta de la imagen del médico solicitado	ImagenSolicitante	Devuelve o recibe la ruta de la imagen del médico solicitante	Solicitado	Devuelve o recibe el nombre Médico solicitado	Solicitante	Devuelve o recibe el nombre Médico solicitante	Métodos		Nombre	Descripción	Discussion	Constructor	getBoardDrawingsFrom		getBoardDrawingsTo		GetHashCode		getMessagesFrom	Devuelve la cola de mensajes desde quien la envía.	getMessagesTo	Devuelve la cola de mensajes a quien la envía.
Propiedades																																	
Nombre	Descripción																																
ActualizarImagenSolicitado	Devuelve o recibe verdadero si es necesaria la actualización de la imagen del médico solicitado, de lo contrario falso.																																
ActualizarImagenSolicitante	Devuelve o recibe verdadero si es necesaria la actualización de la imagen del médico solicitante, de lo contrario falso.																																
ImagenSolicitado	Devuelve o recibe la ruta de la imagen del médico solicitado																																
ImagenSolicitante	Devuelve o recibe la ruta de la imagen del médico solicitante																																
Solicitado	Devuelve o recibe el nombre Médico solicitado																																
Solicitante	Devuelve o recibe el nombre Médico solicitante																																
Métodos																																	
Nombre	Descripción																																
Discussion	Constructor																																
getBoardDrawingsFrom																																	
getBoardDrawingsTo																																	
GetHashCode																																	
getMessagesFrom	Devuelve la cola de mensajes desde quien la envía.																																
getMessagesTo	Devuelve la cola de mensajes a quien la envía.																																

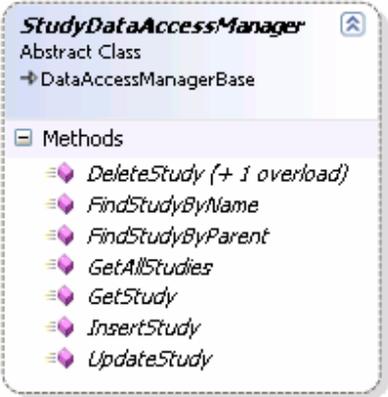
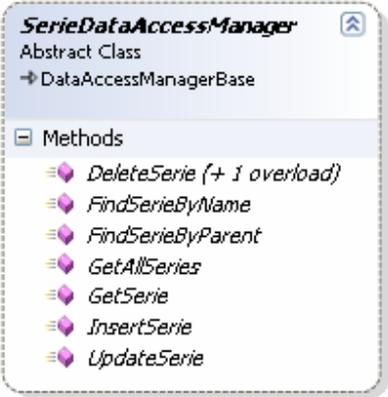
	<p>Clase que representa los datos del dibujo enviado de un cliente a otro y además el mensaje asociado.</p> <table border="1" data-bbox="655 517 1469 658"> <thead> <tr> <th colspan="2">Propiedades</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Message</td> <td>Mensaje</td> </tr> <tr> <td>Red_Pixels</td> <td>Datos del dibujo</td> </tr> </tbody> </table>	Propiedades		Nombre	Descripción	Message	Mensaje	Red_Pixels	Datos del dibujo														
Propiedades																							
Nombre	Descripción																						
Message	Mensaje																						
Red_Pixels	Datos del dibujo																						
	<p>Clase que se encarga de la gestión de las discusiones.</p> <table border="1" data-bbox="655 792 1469 1207"> <thead> <tr> <th colspan="2">Propiedades</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Discusiones</td> <td>Todas las discusiones que se están realizando</td> </tr> <tr> <td>DiscusionesPorUsuario</td> <td></td> </tr> <tr> <td>Peticiones_de_Discusion</td> <td></td> </tr> <tr> <td>PeticionesPorUsuario</td> <td></td> </tr> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> <tr> <td>ExisteDiscusion</td> <td></td> </tr> <tr> <td>getDiscusionHash</td> <td></td> </tr> <tr> <td>SolicitarDiscusion</td> <td></td> </tr> </tbody> </table>	Propiedades		Nombre	Descripción	Discusiones	Todas las discusiones que se están realizando	DiscusionesPorUsuario		Peticiones_de_Discusion		PeticionesPorUsuario		Métodos		Nombre	Descripción	ExisteDiscusion		getDiscusionHash		SolicitarDiscusion	
Propiedades																							
Nombre	Descripción																						
Discusiones	Todas las discusiones que se están realizando																						
DiscusionesPorUsuario																							
Peticiones_de_Discusion																							
PeticionesPorUsuario																							
Métodos																							
Nombre	Descripción																						
ExisteDiscusion																							
getDiscusionHash																							
SolicitarDiscusion																							

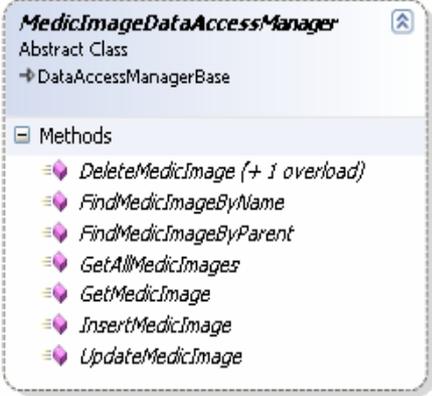
<p>Clases de acceso a datos.</p>															
<table border="1" data-bbox="204 1379 592 1570"> <thead> <tr> <th colspan="2">DataAccessManagerBase</th> </tr> <tr> <th colspan="2">Abstract Class</th> </tr> </thead> <tbody> <tr> <td colspan="2">Methods</td> </tr> <tr> <td colspan="2">Initialize() : void</td> </tr> </tbody> </table>	DataAccessManagerBase		Abstract Class		Methods		Initialize() : void		<p>Clase abstracta base para el acceso a datos.</p> <table border="1" data-bbox="655 1451 1461 1563"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td>Initialize</td> <td>Inicializar la cadena de conexión.</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	Initialize	Inicializar la cadena de conexión.
DataAccessManagerBase															
Abstract Class															
Methods															
Initialize() : void															
Métodos															
Nombre	Descripción														
Initialize	Inicializar la cadena de conexión.														

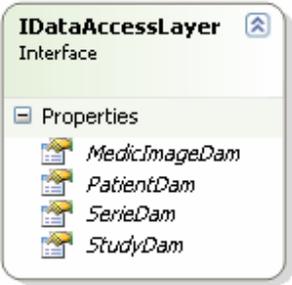
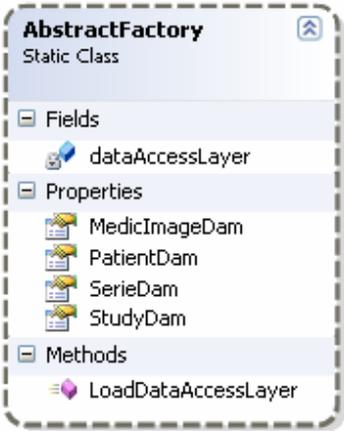
Clase abstracta que especifica y encapsula las funcionalidades que debe tener una clase que se especialice en el acceso a datos del tipo *Patient*. Hereda de *DataAccessManagerBase*.



Métodos	
Nombre	Descripción
<i>AddPatientsToMedics</i>	Método abstracto para adicionar varios pacientes a varios médicos.
<i>DeletePatient</i>	Método abstracto para eliminar pacientes de la fuente de datos.
<i>FindPatientByName</i>	Método abstracto para buscar pacientes por sus nombres.
<i>FindPatientByParent</i>	Método abstracto para obtener todos los pacientes que le pertenecen a un médico específico.
<i>GetAllPatients</i>	Método abstracto que devuelve todos los pacientes almacenados en la fuente de datos.
<i>GetPatient</i>	Método abstracto que devuelve todos los datos del paciente.
<i>InsertPatient</i>	Método abstracto que inserta un paciente en la fuente de datos.
<i>IsPatientInMedic</i>	Método abstracto que devuelve verdadero si el paciente es atendido por un médico dado.
<i>RemovePatientsFromMedic</i>	Método abstracto que elimina a un grupo de pacientes de un grupo de médicos.
<i>UpdatePatient</i>	Método abstracto que actualiza los datos del paciente en la fuente de datos

	<p>Clase abstracta que especifica y encapsula las funcionalidades que debe tener una clase que se especialice en el acceso a datos del tipo <i>Study</i>. Hereda de <i>DataAccessManagerBase</i>.</p> <table border="1" data-bbox="655 546 1465 1151"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><i>DeleteStudy</i></td> <td>Método abstracto que elimina un estudio de la fuente de datos.</td> </tr> <tr> <td><i>FindStudyByName</i></td> <td>Método abstracto que permite buscar estudios por sus nombres.</td> </tr> <tr> <td><i>FindStudyByParent</i></td> <td>Método abstracto que devuelve todos los estudios de un paciente determinado.</td> </tr> <tr> <td><i>GetAllStudies</i></td> <td>Método abstracto que devuelve todos los estudios de la fuente de datos.</td> </tr> <tr> <td><i>GetStudy</i></td> <td>Método abstracto que devuelve un estudio de la fuente de datos.</td> </tr> <tr> <td><i>InsertStudy</i></td> <td>Método abstracto que inserta un estudio en la fuente de datos.</td> </tr> <tr> <td><i>UpdateStudy</i></td> <td>Método abstracto que actualiza un estudio en la fuente de datos.</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	<i>DeleteStudy</i>	Método abstracto que elimina un estudio de la fuente de datos.	<i>FindStudyByName</i>	Método abstracto que permite buscar estudios por sus nombres.	<i>FindStudyByParent</i>	Método abstracto que devuelve todos los estudios de un paciente determinado.	<i>GetAllStudies</i>	Método abstracto que devuelve todos los estudios de la fuente de datos.	<i>GetStudy</i>	Método abstracto que devuelve un estudio de la fuente de datos.	<i>InsertStudy</i>	Método abstracto que inserta un estudio en la fuente de datos.	<i>UpdateStudy</i>	Método abstracto que actualiza un estudio en la fuente de datos.
Métodos																			
Nombre	Descripción																		
<i>DeleteStudy</i>	Método abstracto que elimina un estudio de la fuente de datos.																		
<i>FindStudyByName</i>	Método abstracto que permite buscar estudios por sus nombres.																		
<i>FindStudyByParent</i>	Método abstracto que devuelve todos los estudios de un paciente determinado.																		
<i>GetAllStudies</i>	Método abstracto que devuelve todos los estudios de la fuente de datos.																		
<i>GetStudy</i>	Método abstracto que devuelve un estudio de la fuente de datos.																		
<i>InsertStudy</i>	Método abstracto que inserta un estudio en la fuente de datos.																		
<i>UpdateStudy</i>	Método abstracto que actualiza un estudio en la fuente de datos.																		
	<p>Clase abstracta que especifica y encapsula las funcionalidades que debe tener una clase que se especialice en el acceso a datos del tipo <i>Serie</i>. Hereda de <i>DataAccessManagerBase</i>.</p> <table border="1" data-bbox="655 1308 1465 1836"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><i>DeleteSerie</i></td> <td>Método abstracto que elimina una serie de la fuente de datos</td> </tr> <tr> <td><i>FindSerieByName</i></td> <td>Método abstracto que permite buscar series por sus nombres.</td> </tr> <tr> <td><i>FindSerieByParent</i></td> <td>Método abstracto que devuelve todas las series que pertenecen a un estudio.</td> </tr> <tr> <td><i>GetAllSeries</i></td> <td>Método abstracto que devuelve todas las series desde la fuente de datos.</td> </tr> <tr> <td><i>GetSerie</i></td> <td>Método abstracto que devuelve los datos de una serie en específico.</td> </tr> <tr> <td><i>InsertSerie</i></td> <td>Método abstracto que permite insertar una serie en la fuente de</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	<i>DeleteSerie</i>	Método abstracto que elimina una serie de la fuente de datos	<i>FindSerieByName</i>	Método abstracto que permite buscar series por sus nombres.	<i>FindSerieByParent</i>	Método abstracto que devuelve todas las series que pertenecen a un estudio.	<i>GetAllSeries</i>	Método abstracto que devuelve todas las series desde la fuente de datos.	<i>GetSerie</i>	Método abstracto que devuelve los datos de una serie en específico.	<i>InsertSerie</i>	Método abstracto que permite insertar una serie en la fuente de		
Métodos																			
Nombre	Descripción																		
<i>DeleteSerie</i>	Método abstracto que elimina una serie de la fuente de datos																		
<i>FindSerieByName</i>	Método abstracto que permite buscar series por sus nombres.																		
<i>FindSerieByParent</i>	Método abstracto que devuelve todas las series que pertenecen a un estudio.																		
<i>GetAllSeries</i>	Método abstracto que devuelve todas las series desde la fuente de datos.																		
<i>GetSerie</i>	Método abstracto que devuelve los datos de una serie en específico.																		
<i>InsertSerie</i>	Método abstracto que permite insertar una serie en la fuente de																		

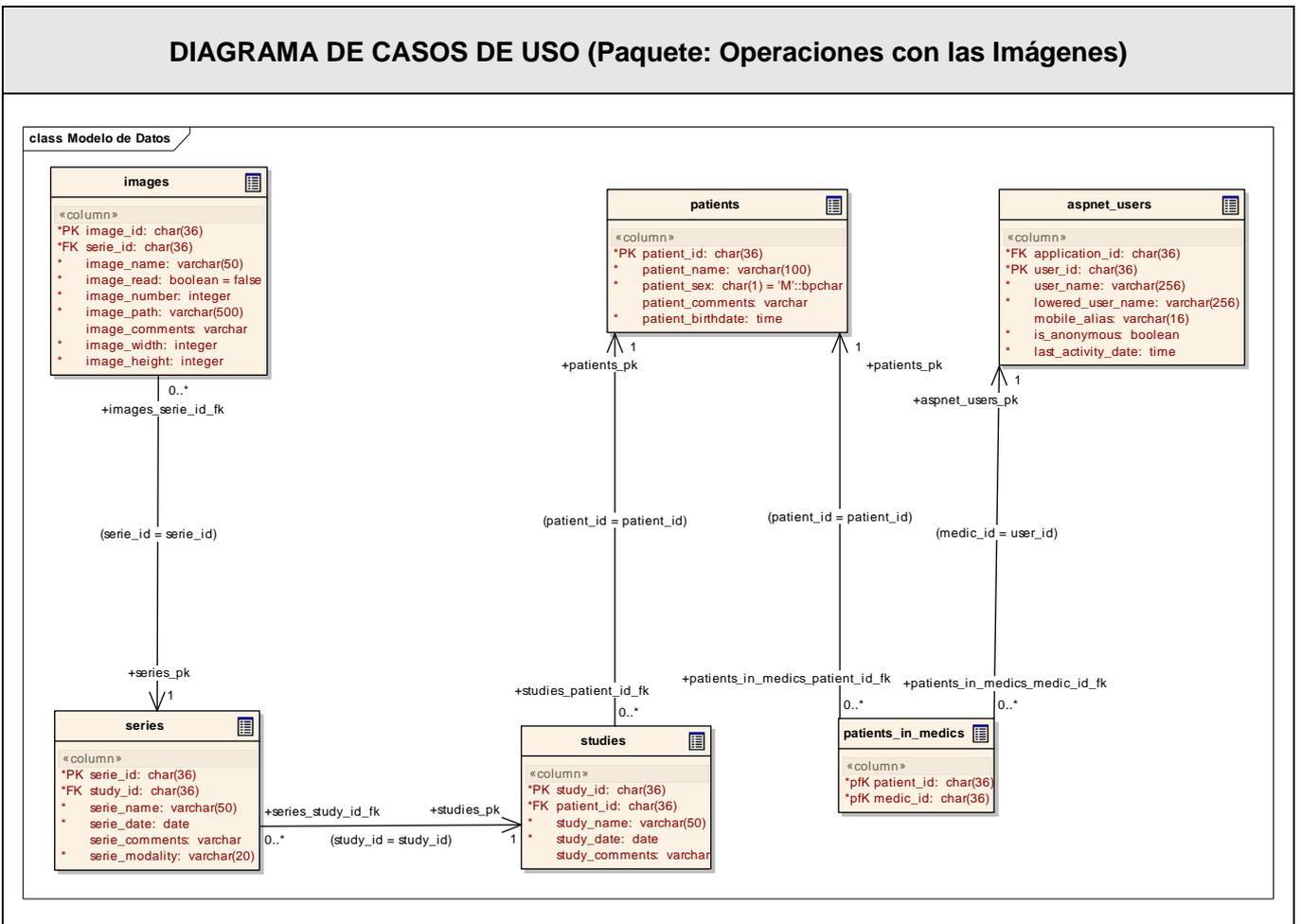
	<p>datos.</p> <p><i>UpdateSerie</i> Método abstracto que permite actualizar los datos de una serie en la fuente de datos.</p>																		
	<p>Clase abstracta que especifica y encapsula las funcionalidades que debe tener una clase que se especialice en el acceso a datos del tipo <i>MedicImage</i>. Hereda de <i>DataAccessManagerBase</i>.</p> <table border="1" data-bbox="651 725 1465 1489"> <thead> <tr> <th colspan="2">Métodos</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><i>DeleteMedicImage</i></td> <td>Método abstracto que elimina una imagen de la fuente de datos.</td> </tr> <tr> <td><i>FindMedicImageByName</i></td> <td>Método abstracto que permite buscar imágenes por sus nombres</td> </tr> <tr> <td><i>FindMedicImageByParent</i></td> <td>Método abstracto que devuelve todas las imágenes de una serie.</td> </tr> <tr> <td><i>GetAllMedicImages</i></td> <td>Método abstracto que devuelve todas las imágenes desde la fuente de datos.</td> </tr> <tr> <td><i>GetMedicImage</i></td> <td>Método abstracto que devuelve los datos de una imagen en específico.</td> </tr> <tr> <td><i>InsertMedicImage</i></td> <td>Método abstracto que inserta una imagen en la fuente de datos.</td> </tr> <tr> <td><i>UpdateMedicImage</i></td> <td>Método abstracto que permite actualizar los datos de una imagen en la fuente de datos.</td> </tr> </tbody> </table>	Métodos		Nombre	Descripción	<i>DeleteMedicImage</i>	Método abstracto que elimina una imagen de la fuente de datos.	<i>FindMedicImageByName</i>	Método abstracto que permite buscar imágenes por sus nombres	<i>FindMedicImageByParent</i>	Método abstracto que devuelve todas las imágenes de una serie.	<i>GetAllMedicImages</i>	Método abstracto que devuelve todas las imágenes desde la fuente de datos.	<i>GetMedicImage</i>	Método abstracto que devuelve los datos de una imagen en específico.	<i>InsertMedicImage</i>	Método abstracto que inserta una imagen en la fuente de datos.	<i>UpdateMedicImage</i>	Método abstracto que permite actualizar los datos de una imagen en la fuente de datos.
Métodos																			
Nombre	Descripción																		
<i>DeleteMedicImage</i>	Método abstracto que elimina una imagen de la fuente de datos.																		
<i>FindMedicImageByName</i>	Método abstracto que permite buscar imágenes por sus nombres																		
<i>FindMedicImageByParent</i>	Método abstracto que devuelve todas las imágenes de una serie.																		
<i>GetAllMedicImages</i>	Método abstracto que devuelve todas las imágenes desde la fuente de datos.																		
<i>GetMedicImage</i>	Método abstracto que devuelve los datos de una imagen en específico.																		
<i>InsertMedicImage</i>	Método abstracto que inserta una imagen en la fuente de datos.																		
<i>UpdateMedicImage</i>	Método abstracto que permite actualizar los datos de una imagen en la fuente de datos.																		

	<p>Interfaz que especifica el contrato para las capas de acceso a datos.</p> <table border="1"> <thead> <tr> <th colspan="2">Propiedades</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><i>MedicImageDam</i></td> <td>Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i></td> </tr> <tr> <td><i>PatientDam</i></td> <td>Devuelve una instancia de la clase abstracta <i>PatienDataAccessManager</i></td> </tr> <tr> <td><i>SerieDam</i></td> <td>Devuelve una instancia de la clase abstracta <i>SerieDataAccessManager</i></td> </tr> <tr> <td><i>StudyDam</i></td> <td>Devuelve una instancia de la clase abstracta <i>StudyDataAccessManager</i></td> </tr> </tbody> </table>	Propiedades		Nombre	Descripción	<i>MedicImageDam</i>	Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i>	<i>PatientDam</i>	Devuelve una instancia de la clase abstracta <i>PatienDataAccessManager</i>	<i>SerieDam</i>	Devuelve una instancia de la clase abstracta <i>SerieDataAccessManager</i>	<i>StudyDam</i>	Devuelve una instancia de la clase abstracta <i>StudyDataAccessManager</i>						
Propiedades																			
Nombre	Descripción																		
<i>MedicImageDam</i>	Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i>																		
<i>PatientDam</i>	Devuelve una instancia de la clase abstracta <i>PatienDataAccessManager</i>																		
<i>SerieDam</i>	Devuelve una instancia de la clase abstracta <i>SerieDataAccessManager</i>																		
<i>StudyDam</i>	Devuelve una instancia de la clase abstracta <i>StudyDataAccessManager</i>																		
	<p>Clase estática que es la responsable de cargar una instancia de la interfaz <i>IDataAccessLayer</i> de la capa de acceso a datos y devolver las respectivas instancias de las clases de acceso a datos.</p> <table border="1"> <thead> <tr> <th colspan="2">Atributo</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><i>dataAccessLayer</i></td> <td>Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">Propiedades</th> </tr> <tr> <th>Nombre</th> <th>Descripción</th> </tr> </thead> <tbody> <tr> <td><i>MedicImageDam</i></td> <td>Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i></td> </tr> <tr> <td><i>PatientDam</i></td> <td>Devuelve una instancia de la clase abstracta <i>PatienDataAccessManager</i></td> </tr> <tr> <td><i>SerieDam</i></td> <td>Devuelve una instancia de la clase abstracta <i>SerieDataAccessManager</i></td> </tr> <tr> <td><i>StudyDam</i></td> <td>Devuelve una instancia de la clase abstracta</td> </tr> </tbody> </table>	Atributo		Nombre	Descripción	<i>dataAccessLayer</i>	Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i>	Propiedades		Nombre	Descripción	<i>MedicImageDam</i>	Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i>	<i>PatientDam</i>	Devuelve una instancia de la clase abstracta <i>PatienDataAccessManager</i>	<i>SerieDam</i>	Devuelve una instancia de la clase abstracta <i>SerieDataAccessManager</i>	<i>StudyDam</i>	Devuelve una instancia de la clase abstracta
Atributo																			
Nombre	Descripción																		
<i>dataAccessLayer</i>	Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i>																		
Propiedades																			
Nombre	Descripción																		
<i>MedicImageDam</i>	Devuelve una instancia de la clase abstracta <i>MedicImageDataAccessManager</i>																		
<i>PatientDam</i>	Devuelve una instancia de la clase abstracta <i>PatienDataAccessManager</i>																		
<i>SerieDam</i>	Devuelve una instancia de la clase abstracta <i>SerieDataAccessManager</i>																		
<i>StudyDam</i>	Devuelve una instancia de la clase abstracta																		

	<i>StudyDataAccessManager</i>	
Métodos		
Nombre	Descripción	
LoadDataAccessLayer	Responsable de inicializar el atributo <i>dataAccessLayer</i>	

3.2.2 Diagrama entidad-relación.

El presente diagrama representa las tablas de la base de datos donde serán almacenados los datos de las clases entidades y las relaciones que existen entre ellas.



Descripción de las tablas del modelo de entidad relación.

Tabla	Descripción												
<div style="border: 1px solid black; padding: 5px;"> <p>class Modelo de Datos</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">aspnet_users</p> <p>«column»</p> <p>*FK application_id: char(36)</p> <p>*PK user_id: char(36)</p> <p>* user_name: varchar(256)</p> <p>* lowered_user_name: varchar(256)</p> <p>mobile_alias: varchar(16)</p> <p>* is_anonymous: boolean</p> <p>* last_activity_date: time</p> </div> </div>	<p>Esta tabla pertenece a la base de datos de seguridad de ASP.NET, empleada para almacenar información relacionada con el registro de los usuarios.</p>												
<div style="border: 1px solid black; padding: 5px;"> <p>class Modelo de Datos</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">patients</p> <p>«column»</p> <p>*PK patient_id: char(36)</p> <p>* patient_name: varchar(100)</p> <p>* patient_sex: char(1) = 'M':::bpchar</p> <p>patient_comments: varchar</p> <p>* patient_birthdate: time</p> </div> </div>	<p>Tabla que se emplea para el almacenamiento de la información relacionada al paciente.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Atributo</th> <th style="text-align: left;">Descripción</th> </tr> </thead> <tbody> <tr> <td>patient_id</td> <td>Único identificador</td> </tr> <tr> <td>patient_name</td> <td>Nombre del paciente</td> </tr> <tr> <td>patient_birthdate</td> <td>Fecha de nacimiento</td> </tr> <tr> <td>patient_sex</td> <td>Sexo</td> </tr> <tr> <td>patient_comments</td> <td>Este campo es el designado para almacenar las anotaciones que el Médico le realiza al paciente.</td> </tr> </tbody> </table>	Atributo	Descripción	patient_id	Único identificador	patient_name	Nombre del paciente	patient_birthdate	Fecha de nacimiento	patient_sex	Sexo	patient_comments	Este campo es el designado para almacenar las anotaciones que el Médico le realiza al paciente.
Atributo	Descripción												
patient_id	Único identificador												
patient_name	Nombre del paciente												
patient_birthdate	Fecha de nacimiento												
patient_sex	Sexo												
patient_comments	Este campo es el designado para almacenar las anotaciones que el Médico le realiza al paciente.												
<div style="border: 1px solid black; padding: 5px;"> <p>class Modelo de Datos</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">studies</p> <p>«column»</p> <p>*PK study_id: char(36)</p> <p>*FK patient_id: char(36)</p> <p>* study_name: varchar(50)</p> <p>* study_date: date</p> <p>study_comments: varchar</p> </div> </div>	<p>Tabla que se emplea para el almacenamiento de la información relacionada al estudio.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">Atributo</th> <th style="text-align: left;">Descripción</th> </tr> </thead> <tbody> <tr> <td>patient_id</td> <td>Llave extranjera, para identificar los estudios que pertenecen a un paciente.</td> </tr> <tr> <td>study_id</td> <td>Único identificador</td> </tr> <tr> <td>study_name</td> <td>Nombre del estudio</td> </tr> <tr> <td>study_date</td> <td>Fecha de realización del estudio</td> </tr> <tr> <td>study_comments</td> <td>Este campo es el designado para almacenar las anotaciones que el Médico le realiza al estudio.</td> </tr> </tbody> </table>	Atributo	Descripción	patient_id	Llave extranjera, para identificar los estudios que pertenecen a un paciente.	study_id	Único identificador	study_name	Nombre del estudio	study_date	Fecha de realización del estudio	study_comments	Este campo es el designado para almacenar las anotaciones que el Médico le realiza al estudio.
Atributo	Descripción												
patient_id	Llave extranjera, para identificar los estudios que pertenecen a un paciente.												
study_id	Único identificador												
study_name	Nombre del estudio												
study_date	Fecha de realización del estudio												
study_comments	Este campo es el designado para almacenar las anotaciones que el Médico le realiza al estudio.												

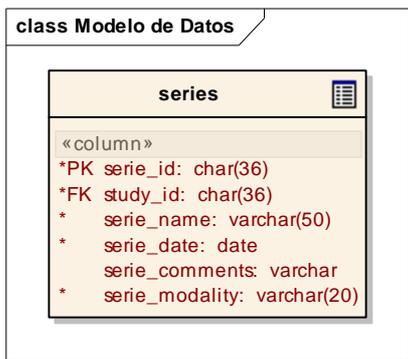


Tabla que se emplea para el almacenamiento de la información relacionada a las series.

Atributo	Descripción
study_id	Llave extranjera, para identificar las series que pertenecen a un estudio.
serie_id	Único identificador
serie_name	Nombre de la serie
serie_date	Fecha de realización de la serie
serie_modality	Modalidad de la serie, es decir si es una Tomografía Axial Computarizada (TAC), Resonancia Magnética (RM), etc.
serie_comments	Este campo es el designado para almacenar las anotaciones que el Médico le realiza a la serie.

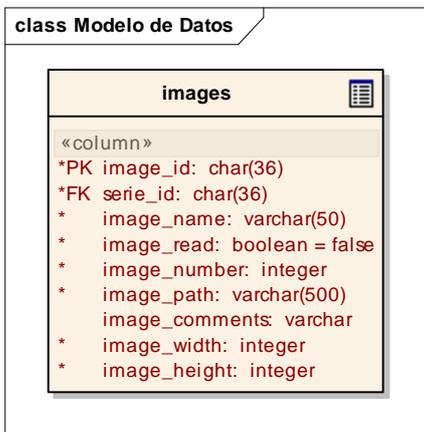


Tabla que se emplea para el almacenamiento de la información relacionada a las imágenes.

Atributo	Descripción
serie_id	Llave extranjera, para identificar las imágenes que pertenecen a una serie.
image_id	Único identificador
image_name	Nombre de la imagen
image_read	Solo puede tomar valores verdaderos (true) o falsos (false) y sirve para identificar las imágenes que se han visto por lo menos una vez.
image_number	Posición de la imagen en el fichero DICOM
image_path	Ruta para la imagen respecto a la raíz de la aplicación.
image_width	Ancho original de la imagen.
image_height	Alto original de la imagen.

3.3 CONCLUSIONES

En este capítulo se realizó el análisis y el diseño del sistema, además se mostraron los principales diagramas correspondientes a estas dos fases como los diagramas de clases del análisis y los diagramas de clases del diseño. También se mostraron las descripciones de las clases y de las tablas del modelo de datos.

CAPÍTULO IV

IMPLEMENTACIÓN

En el presente capítulo se definen los diagramas de despliegue y de componentes.

4.1 DESPLIEGUE

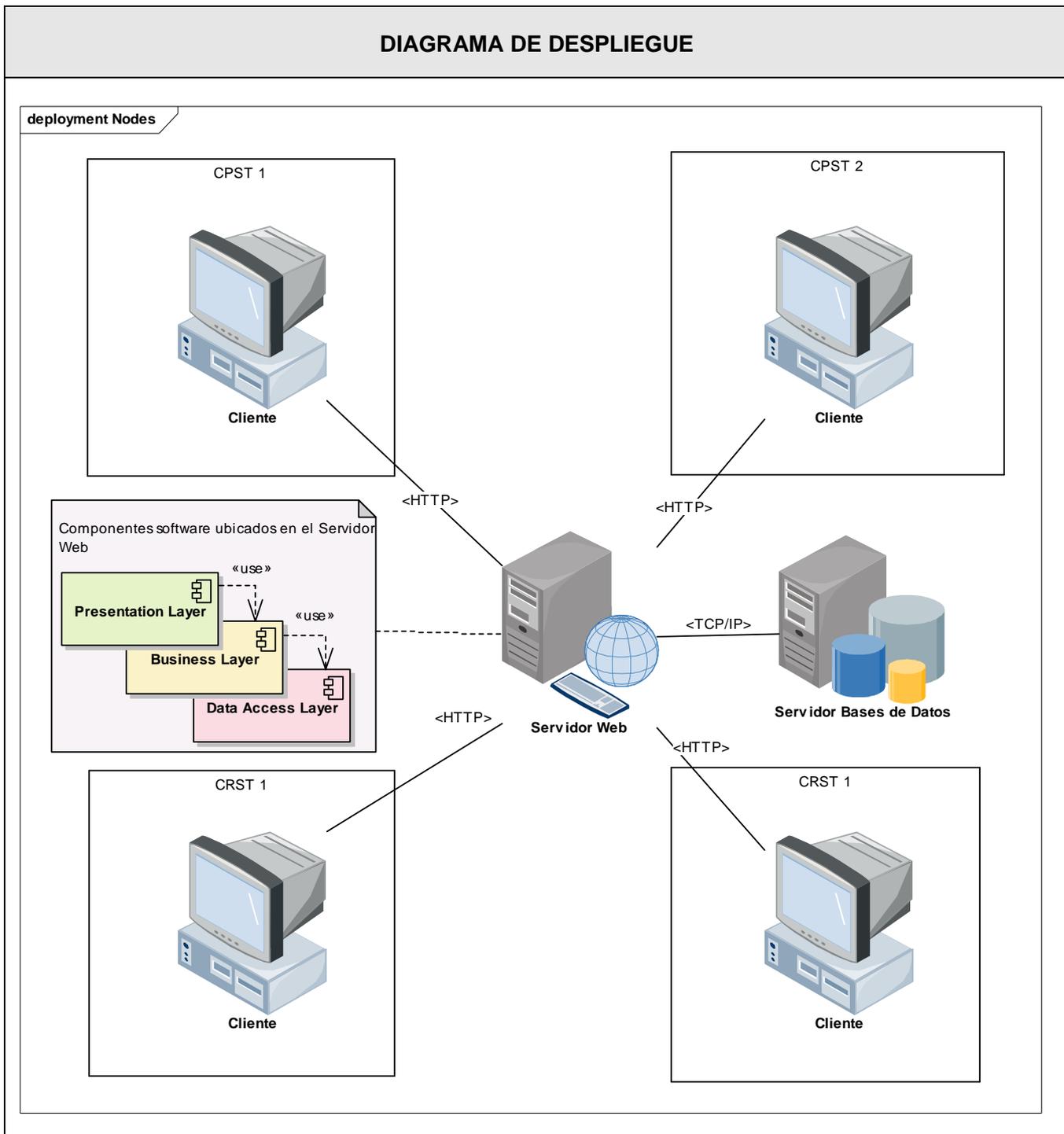
Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo será una unidad de computación de algún tipo, desde un sensor hasta un mainframe. Un nodo puede contener instancias de componentes software, objetos, procesos (caso particular de un objeto).

En el caso de las características de los servidores (Web y de bases de datos) sus características aparecen en la sección [2.2.2 Atributos del sistema](#).

En el diagrama de despliegue que aparece a continuación se muestra la distribución de los dispositivos de cómputo del sistema de teleconsultas. Los clientes que aparecen definidos con CPST (Clientes que Proveen Servicios de Teleconsulta), son computadoras clientes que se pueden encontrar en los centros médicos de atención secundaria o terciaria de salud y desde las cuales se brindan los servicios de teleconsulta. Los clientes con la definición CRST (Centros que Requieren Servicios de Teleconsulta) serán aquellos centros médicos los cuales requieren de los servicios especializados que ofrecen los primeros.

Por la flexibilidad del sistema los clientes no están fijos en la definición inicial dada a cada uno, sino que según sus necesidades pueden asumir una u otra.

DIAGRAMA DE DESPLIEGUE

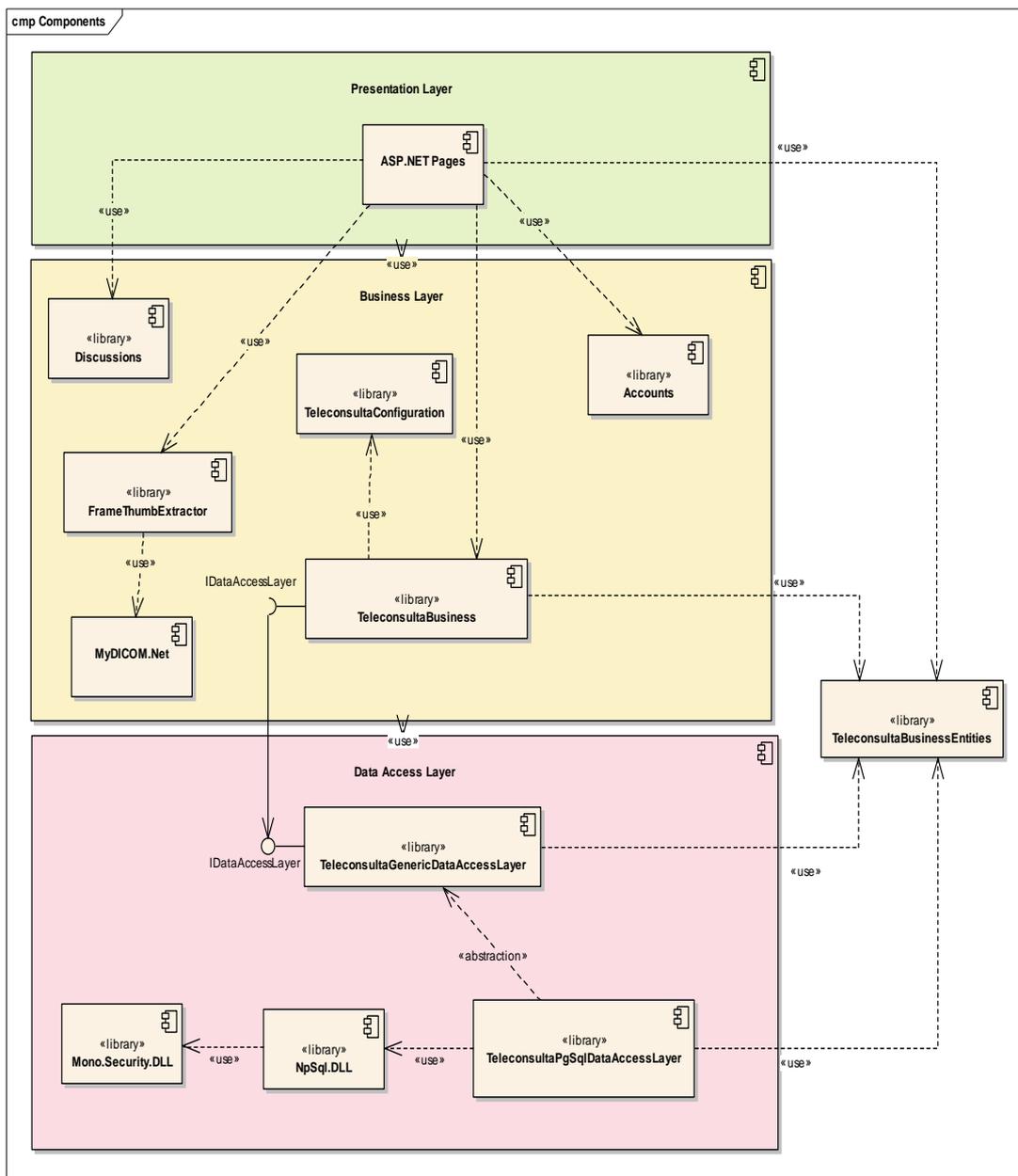


4.2 COMPONENTES

Lo que distingue un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener, paquetes utilizados para agrupar elementos del modelo.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un diagrama de componentes serán componentes y paquetes. En cuanto a los componentes, sólo aparecen tipos de componentes, ya que las instancias específicas de cada tipo se encuentran en el diagrama de despliegue.

DIAGRAMA DE COMPONENTES



4.3 CONCLUSIONES

Se definieron los diagramas de despliegue y de componentes. En el caso del primero, se observó la distribución de los dispositivos hardware, y el de componentes mostró las relaciones existentes entre un conjunto de estos que se organizan como piezas de un rompecabezas para que el sistema funcione como un todo.

CONCLUSIONES

Con la realización de este trabajo se concluye de forma general lo siguiente: el desarrollo actual de la informática y las telecomunicaciones, y su aplicación en el campo de la medicina ha favorecido en gran medida, el nivel el desarrollo y evolución de la telemedicina. Dentro de las múltiples aristas que posee la telemedicina, las que más aplicación tiene en el entorno de la salud cubana es la teleconsulta.

Además, para poder realizar este trabajo, se analizaron sistemas que prestan servicios de telemedicina a escala internacional y nacional; se propuso un sistema para la teleconsulta en línea, se realizó la captura de requisitos, posteriormente el análisis y el diseño del sistema propuesto y por último se materializó el sistema en una aplicación web.

RECOMENDACIONES

Para la continuación de este trabajo se hacen las siguientes recomendaciones:

- Incorporarle transmisión de audio y video, para facilitar otros tipos de diagnósticos.
- Realizar un estudio para una posible ampliación del sistema para brindar servicios de educación a distancia.
- Integrar este sistema al Cassandra PACS²¹, para de esta forma poder obtener los estudios de los pacientes desde el servidor del PACS.
- Someter el sistema a pruebas de calidad.
- Realizar un estudio para la implementación del sistema en plataforma libre.

²¹ PACS (Picture Archive Communication System) Sistemas que facilitan la visualización de imágenes médicas, así como también almacenar estas en dispositivos magnéticos u ópticos. Los PACS permiten además la comunicación usando las redes telemáticas.

REFERENCIAS BIBLIOGRÁFICAS

1. **Anónimo.** Telemedicina. *Wikipedia, la enciclopedia libre*. [En línea] 2007. <http://es.wikipedia.org/wiki/Telemedicina>. 1.
2. **Méndez Sánchez, Dra. Teresa de J., et al.** Efectividad de la telemedicina en el estrabismo. *Rev Cubana Oftalmol* 2003. [En línea] 2003. http://bvs.sld.cu/revistas/oft/vol16_01_03/oft06103.htm.
3. —. Proceso Unificado de Rational. *Wikipedia, la enciclopedia libre*. [En línea] 2007. <http://es.wikipedia.org/wiki/RUP>.
4. —. C Sharp. *Wikipedia, la enciclopedia libre*. [En línea] 2007. http://es.wikipedia.org/wiki/C_Sharp.
5. —. JavaScript. *Wikipedia, la enciclopedia libre*. [En línea] 2007. <http://es.wikipedia.org/wiki/JavaScript>.
6. —. Lenguaje Unificado de Modelado. *Wikipedia, la enciclopedia libre*. [En línea] 2007. <http://es.wikipedia.org/wiki/Uml>.
7. —. Microsoft Visual Studio .Net. *Wikipedia, la enciclopedia libre*. [En línea] 2007. http://es.wikipedia.org/wiki/Microsoft_Visual_Studio_.Net.
8. **Telerik.** AJAX Revolution. Join in. *Telerik*. [En línea] 2006. En inglés.. <http://www.telerik.com/community/ajax-learning-resources.aspx>.
9. **Pey, Xavier.** ASP.NET, Características . *netveloper*. [En línea] 06 11, 2001. http://www.netveloper.com/contenido2.aspx?IDC=41_0.

BIBLIOGRAFÍA

- Second Opinion Software.** Second Opinion Software. *Second Opinion Software*. [En línea] Second Opinion Software, 2005. En inglés. <http://www.2opinion.com/products/index.htm>.
- GlobalMedia.** Telemedicine and Industrial Video Conferencing Solutions From GlobalMedia. *Telemedicine and Industrial Video Conferencing Solutions From GlobalMedia*. [En línea] 2005. En inglés. <http://www.globalmedia.com/index.php>.
- Vitech.** Vitech. *Vitech*. [En línea] <http://www.vitech.com.mx/perfil.html>.
- Larman, Craig.** *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. 2004.
- AIN.** Granma Internacional Digital. *Granma Internacional Digital*. [En línea] 05 16, 2003. <http://www.granma.cu/espanol/mayo03/vier16/cardio.html>.
- VS, Rajesh.** Abstract Factory Patterns in C#. *C# Corner*. [En línea] 12 10, 2001. En inglés.. <http://www.c-sharpcorner.com/UploadFile/rajeshvs/AbstractFactoryPatternsinCS11142005001854AM/AbstractFactoryPatternsinCS.aspx>.
- Ikram, Saif.** Design Patterns Part - I. *C# Corner*. [En línea] 08 26, 2005. En inglés. http://www.c-sharpcorner.com/UploadFile/saif_ikram/DesignPatternsPart108262005090030AM/DesignPatternsPart1.aspx.
- VS, Rajesh.** Factory Patterns in C#. *C# Corner*. [En línea] 12 10, 2001. En inglés. <http://www.c-sharpcorner.com/UploadFile/rajeshvs/FactoryPatternsinCS11202005231940PM/FactoryPatternsinCS.aspx>
- Tamimi, Waleed El.** Data Access Component and the Factory Design Pattern. *C# Corner*. [En línea] 04 05, 2006. En inglés. <http://www.c-sharpcorner.com/UploadFile/wtamimi/dacfactory04052006061823AM/dacfactory.aspx>.

Cochran, Matthew. Wrapper Patterns in C#, Part IV: The Adapter Pattern. *C# Corner*. [En línea] 02 16, 2006. En inglés. http://www.sharpcorner.com/UploadFile/rmcochran/csharp_wrapper402122006091959AM/csharp_wrapper4.aspx.

ANEXOS

ANEXO I. CASOS DE USO EXPANDIDOS

CU-01– Subir DICOM para el servidor

CU-01	Subir DICOM para el servidor	
Actores:	Médico	
Propósito:	Insertar un nuevo fichero DICOM en el servidor	
Resumen:	El Médico sube para el servidor. Posteriormente se extrae la información necesaria del DICOM y esta es almacenada en una base de datos. Además se extraen todas las imágenes que el DICOM posee y se almacenan en una carpeta específica del disco duro del servidor.	
Referencias:	RF-1.1, RF-1.2, RF-1.3, RF-1.4, RF-1.5, RF-1.6	
Precondiciones:		
Poscondiciones:	1 – Infamación almacenada en la base de datos. 2 – Imágenes localizadas en una ruta específica.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – El Médico selecciona subir un fichero para el servidor.	1.1 – Muestra el diálogo correspondiente para la búsqueda del fichero DICOM en la PC del Médico .	
2 – Localiza el fichero DICOM, lo selecciona y luego presiona aceptar.	2.1 – Sube el fichero seleccionado hacia una carpeta específica. 2.2 – Extrae la información necesaria. 2.3 – Extrae las imágenes. 2.4 – Guarda la información. 2.5 – Muestra la información.	

CU-02 – Seleccionar paciente

CU-02	Seleccionar Paciente	
Actores:	Médico	
Propósito:	Mostrar la información relativa al paciente seleccionado.	
Resumen:	El Médico selecciona a uno de los Pacientes para obtener información a cerca de este.	
Referencias:	RF-2.1	
Precondiciones:	1 – Existir al menos un paciente. 2 – Paciente seleccionado.	
Poscondiciones:	1 – Datos del paciente mostrados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona al paciente.	1.1 – Muestra la información referente al elemento seleccionado.	

CU-03 – Eliminar paciente

CU-03	Eliminar paciente	
Actores:	Médico	
Propósito:	Eliminar la información relativa al paciente seleccionado.	
Resumen:	El Médico selecciona a uno de los Pacientes que aparecen en la lista y lo elimina.	
Referencias:	CU-02, RF-2.2	
Precondiciones:	1 – Existir al menos un paciente. 2 – Paciente seleccionado.	
Poscondiciones:	1 – Datos del paciente eliminados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona al paciente.	1.1 – Muestra la información referente al elemento seleccionado.	
2 – Presiona eliminar	2.1 – Elimina toda la información referente a ese paciente. 2.2 – Actualiza la información en pantalla	

CU-04 – Realizar anotaciones en el paciente

CU-04	Realizar anotaciones en el Paciente	
Actores:	Médico	
Propósito:	Realizar las anotaciones que el Médico crea pertinente sobre el Paciente	
Resumen:	El Médico selecciona a uno de los Pacientes que aparecen en la lista y edita sus anotaciones.	
Referencias:	CU-02, RF-2.3	
Precondiciones:	1 – Existir al menos un paciente. 2 – Paciente seleccionado.	
Poscondiciones:		
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona al paciente.	1.1 – Muestra la información referente al elemento seleccionado.	
2 – Presiona editar anotaciones.	2.1 – Habilita el cuadro de texto correspondiente a las anotaciones.	
3 – Realiza sus anotaciones y presiona actualizar	3.1 – Se actualiza la información.	
Flujo alterno		
3 – Realiza sus anotaciones y presiona cancelar.	3.1 – No se realiza ninguna actualización.	

CU-05 – Seleccionar estudio

CU-05	Seleccionar Estudio	
Actores:	Médico	
Propósito:	Mostrar la información relativa al estudio seleccionado.	
Resumen:	El Médico selecciona uno de los estudios de algún paciente para obtener información a cerca de este.	
Referencias:	RF-3.1	
Precondiciones:	1 – Existir al menos un estudio contenido en el paciente. 2 – Estudio seleccionado.	
Poscondiciones:	1 – Datos del estudio mostrados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona al estudio.	1.1 – Muestra la información referente al elemento seleccionado.	

CU-06 – Eliminar estudio

CU-06	Eliminar estudio	
Actores:	Médico	
Propósito:	Eliminar la información relativa al estudio seleccionado.	
Resumen:	El Médico selecciona uno de los estudios que aparecen contenidos dentro de un paciente en la lista y lo elimina.	
Referencias:	CU-05, RF-3.2	
Precondiciones:	1 – Existir al menos un paciente con al menos un estudio. 2 – Estudio seleccionado.	
Poscondiciones:	1 – Datos del estudio eliminados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona al estudio.	1.1 – Muestra la información referente al elemento seleccionado.	
2 – Presiona eliminar	2.1 – Elimina toda la información referente a ese estudio. 2.2 – Actualiza la información en pantalla.	

CU-07 – Realizar anotaciones en el estudio

CU-07	Realizar anotaciones en el estudio	
Actores:	Médico	
Propósito:	Realizar las anotaciones que el Médico crea pertinente sobre el estudio	
Resumen:	El Médico selecciona uno de los estudios que aparecen contenidos dentro de un paciente en la lista y le realiza anotaciones	
Referencias:	CU-05, RF-3.3	
Precondiciones:	1 – Existir al menos un paciente con al menos un estudio. 2 – Estudio seleccionado.	
Poscondiciones:		
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona al estudio.	1.1 – Muestra la información referente al elemento seleccionado.	
2 – Presiona editar anotaciones.	2.1 – Habilita el cuadro de texto correspondiente a las anotaciones.	
3 – Realiza sus anotaciones y presiona actualizar	3.1 – Se actualiza la información.	
Flujo alterno		
3 – Realiza sus anotaciones y presiona cancelar.	3.1 – No se realiza ninguna actualización.	

CU-08 – Seleccionar serie

CU-08	Seleccionar serie	
Actores:	Médico	
Propósito:	Mostrar la información relativa a la serie seleccionada.	
Resumen:	El Médico selecciona una de las series contenidas en algún estudio para obtener información a cerca de esta.	
Referencias:	RF-4.1	
Precondiciones:	1 – Existir al menos una serie en el estudio. 2 – Serie seleccionada.	
Poscondiciones:	1 – Datos de la serie mostrados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona la serie.	1.1 – Muestra la información referente al elemento seleccionado.	

CU-09 – Eliminar serie

CU-09	Eliminar serie	
Actores:	Médico	
Propósito:	Eliminar la información relativa a la serie seleccionada.	
Resumen:	El Médico selecciona una de las series contenidas en algún estudio y la elimina	
Referencias:	CU-08, RF-4.2	
Precondiciones:	1 – Existir al menos una serie en el estudio. 2 – Serie seleccionada.	
Poscondiciones:	1 – Datos de la serie eliminados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona la serie.	1.1 – Muestra la información referente al elemento seleccionado.	
2 – Presiona eliminar	2.1 – Elimina toda la información referente a esa serie. 2.2 – Actualiza la información en pantalla.	

CU-10 – Realizar anotaciones en la serie

CU-07	Realizar anotaciones en la serie	
Actores:	Médico	
Propósito:	Realizar las anotaciones que el Médico crea pertinente sobre la serie	
Resumen:	El Médico selecciona una de las series contenidas en algún estudio y le realiza anotaciones.	
Referencias:	CU-08, RF-4.3	
Precondiciones:	1 – Existir al menos una serie en el estudio. 2 – Serie seleccionada.	
Poscondiciones:		
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona la serie.	1.1 – Muestra la información referente al elemento seleccionado.	
2 – Presiona editar anotaciones.	2.1 – Habilita el cuadro de texto correspondiente a las anotaciones.	
3 – Realiza sus anotaciones y presiona actualizar	3.1 – Se actualiza la información.	
Flujo alterno		
3 – Realiza sus anotaciones y presiona cancelar.	3.1 – No se realiza ninguna actualización.	

CU-11 – Seleccionar imagen

CU-11	Seleccionar imagen	
Actores:	Médico	
Propósito:	Mostrar la información relativa a la imagen seleccionada.	
Resumen:	El Médico selecciona una de las imágenes contenidas en alguna serie para obtener información a cerca de esta.	
Referencias:	RF-5.1	
Precondiciones:	1 – Existir al menos una imagen en la serie. 2 – Imagen seleccionada.	
Poscondiciones:	1 – Datos de la imagen mostrados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona la imagen.	1.1 – Muestra la información referente al elemento seleccionado.	

CU-12 – Eliminar imagen

CU-12	Eliminar imagen	
Actores:	Médico	
Propósito:	Eliminar la información relativa a la imagen seleccionada.	
Resumen:	El Médico selecciona una de las imágenes contenidas en alguna serie y la elimina.	
Referencias:	CU-11, RF-5.2	
Precondiciones:	1 – Existir al menos una imagen en la serie. 2 – Imagen seleccionada.	
Poscondiciones:	1 – Datos de la imagen eliminados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona la imagen.	1.1 – Muestra la información referente al elemento seleccionado.	
2 – Presiona eliminar	2.1 – Elimina toda la información referente a esa imagen. 2.2 – Actualiza la información en pantalla.	

CU-13 – Realizar anotaciones en la imagen

CU-13	Realizar anotaciones en la imagen	
Actores:	Médico	
Propósito:	Realizar las anotaciones que el Médico crea pertinente sobre la imagen	
Resumen:	El Médico selecciona una de las imágenes contenidas en alguna serie y realiza las anotaciones.	
Referencias:	CU-11, RF-5.3	
Precondiciones:	1 – Existir al menos una imagen en la serie. 2 – Imagen seleccionada.	
Poscondiciones:		
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona la imagen.	1.1 – Muestra la información referente al elemento seleccionado.	
2 – Presiona editar anotaciones.	2.1 – Habilita el cuadro de texto correspondiente a las anotaciones.	
3 – Realiza sus anotaciones y presiona actualizar	3.1 – Se actualiza la información.	
Flujo alterno		
3 – Realiza sus anotaciones y presiona cancelar.	3.1 – No se realiza ninguna actualización.	

CU-14 – Buscar usuarios

CU-14	Buscar usuarios	
Actores:	Médico	
Propósito:	Buscar usuarios registrados en el sistema	
Resumen:	El Médico busca a otros usuarios registrados en el sistema.	
Referencias:	RF-6.1	
Precondiciones:		
Poscondiciones:		
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona buscar usuarios.	1.1 – Muestra la ventana correspondiente a las búsquedas.	
2 – Introduce el nombre o parte del nombre del usuario que desea buscar.	2.1 – Busca todos los usuarios que sus nombres tengan coincidencia con el entrada por el Médico y los adiciona a una lista. 2.2 – Muestra una lista con el resultado de la búsqueda.	

CU-15 – Añadir contactos

CU-15	Añadir contacto	
Actores:	Médico	
Propósito:	Adicionar nuevos usuarios a su lista de contactos.	
Resumen:	El Médico busca a otros usuarios registrados en el sistema para luego agregar los que le interese.	
Referencias:	CU-14, RF-6.2	
Precondiciones:		
Poscondiciones:	1 – Contactos adicionados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona añadir contactos.	1.1 – Muestra la ventana correspondiente a adicionar contactos.	
2 – Introduce el nombre o parte del nombre del usuario que desea buscar para añadirlo a su lista de contactos y presiona buscar.	2.1 – Busca todos los usuarios que sus nombres tengan coincidencia con el entrada por el Médico y los adiciona a una lista. 2.2 – Muestra una lista con el resultado de la búsqueda.	
3 – Selecciona el o los usuarios que desea aparezcan dentro de su lista de contactos.	3.1 – Agrega los nuevos contactos. 3.2 – Actualiza la lista de contactos del Médico .	

CU-16 – Eliminar contacto

CU-16	Eliminar contacto	
Actores:	Médico	
Propósito:	Eliminar contactos de su lista de contactos.	
Resumen:	El Médico selecciona los contactos con el fin de eliminarlos de su lista de contactos.	
Referencias:	RF-6.3	
Precondiciones:	1 – Contactos seleccionados.	
Poscondiciones:	1 – Contactos eliminados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona la opción de eliminar contactos.	1.1 – Muestra la ventana correspondiente para eliminar contactos.	
2 – Selecciona los contactos a eliminar de la lista.	2.1 – Elimina todos los contactos seleccionados. 2.2 – Actualiza la lista de contactos.	

CU-17 – Renombrar contacto

CU-17	Renombrar contacto	
Actores:	Médico	
Propósito:	Cambiar el nombre a los contactos de su lista de contactos.	
Resumen:	El Médico selecciona a alguno de sus contactos con el fin de renombrarlo y así poder identificarlo con mucha mayor rapidez.	
Referencias:	RF-6.4	
Precondiciones:	1 – Contacto seleccionado.	
Poscondiciones:	1 – Contacto renombrado.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona a algún contacto y presiona renombrar	1.1 – Muestra el nombre actual del contacto en una caja de texto.	
2 – Escribe en nuevo nombre del contacto y presiona actualizar.	2.1 – Actualiza el nuevo nombre del contacto. 2.2 – Deshabilita la caja de texto.	

CU-18 – Teleconsultar contacto

CU-17	Teleconsultar contacto
Actores:	Médico
Propósito:	Pedir una segunda opinión a un especialista.
Resumen:	El Médico selecciona a alguno de sus contactos con el fin de establecer una comunicación con este, e intercambiar opiniones sobre algún caso determinado.
Referencias:	RF-6.5
Precondiciones:	1 – Aparece un caso que requiere ser teleconsultado.
Poscondiciones:	1 – Caso diagnosticado.
Escenario1	
Acciones del Actor	Respuestas del Sistema
1 – Selecciona a algún contacto de la lista y realiza un doble clic con el Mouse.	1.1 – Muestra la interfaz correspondiente a la teleconsulta.
2 – Escribe sus ideas o dudas en una caja de texto y presiona la tecla “Enter”. (Y así sucesivamente)	2.1 – Establece un canal de comunicación entre ambos (Médico y contacto teleconsultado) si este no existe. 2.2 – Levanta en la PC del contacto teleconsultado una interfaz similar a la del Médico que teleconsulta. 2.3 – Muestra el mensaje al contacto seleccionado. 2.4 –Muestra el mensaje respuesta proveniente del contacto al cual se le ha solicitado la teleconsulta. (Y así sucesivamente)
3 – Diagnostica el caso y cierra la ventana correspondiente a la teleconsulta.	3.1 Elimina el canal de comunicación que se había establecido entre el Médico y el contacto teleconsultado.
Escenario2	
Acciones del Actor	Respuestas del Sistema
1 – Selecciona a algún contacto de la lista y realiza un doble clic con el Mouse.	1.1 – Muestra la interfaz correspondiente a la teleconsulta.
2 – Marca detalles que le resultan de interés sobre la imagen y presiona enviar. (Y así sucesivamente)	2.1 – Establece un canal de comunicación entre ambos (Médico y contacto teleconsultado), si este no existe. 2.2 – Levanta en la PC del contacto teleconsultado una interfaz similar a la del Médico que teleconsulta. 2.3 – Muestra las marcas realizadas por el Médico al contacto teleconsultado en la imagen que es común para ambos. 2.4 –Muestra el mensaje respuesta proveniente

	del contacto al cual se le ha solicitado la teleconsulta. (Y así sucesivamente)
3 – Diagnostica el caso y cierra la ventana correspondiente a la teleconsulta.	3.1 Elimina el canal de comunicación que se había establecido entre el Médico y el contacto teleconsultado.

CU-19 – Hacer zoom sobre la imagen

CU-17	Hacer zoom sobre la imagen	
Actores:	Médico	
Propósito:	Incrementar o disminuir el tamaño de la imagen con el objetivo de encontrar detalles de interés.	
Resumen:	El Médico incrementa o disminuye el tamaño de la imagen con el fin de encontrar detalles que le ayuden al diagnóstico.	
Referencias:	CU-11, RF-7.2	
Precondiciones:	1 – Imagen seleccionada.	
Poscondiciones:	1 – Imagen escalada.	
Escenario 1		
Acciones del Actor	Respuestas del Sistema	
1 – Presiona la tecla “Shift” y arrastra con el Mouse la imagen con el objetivo de hacerla más grande.	1.1 – Incrementa el tamaño de la imagen.	
Escenario 2		
Acciones del Actor	Respuestas del Sistema	
1 – Presiona la tecla “Shift” y arrastra con el Mouse la imagen con el objetivo de hacerla más pequeña.	1.1 – Decrementa el tamaño de la imagen.	

CU-20 – Mover la imagen

CU-17	Mover la imagen	
Actores:	Médico	
Propósito:	Arrastrar o mover la imagen por toda el área de trabajo para obtener información.	
Resumen:	El Médico mueve la imagen hasta una posición en la cual puede observar detalles de esta, que por ser demasiado grande no observan del todo en el área de trabajo.	
Referencias:	CU-11, RF-7.3	
Precondiciones:	1 – Imagen seleccionada.	
Poscondiciones:	1 – Imagen cambiada de posición.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Da clic sobre el área de la imagen y con el	1.1 – Actualiza la posición de la imagen hasta	

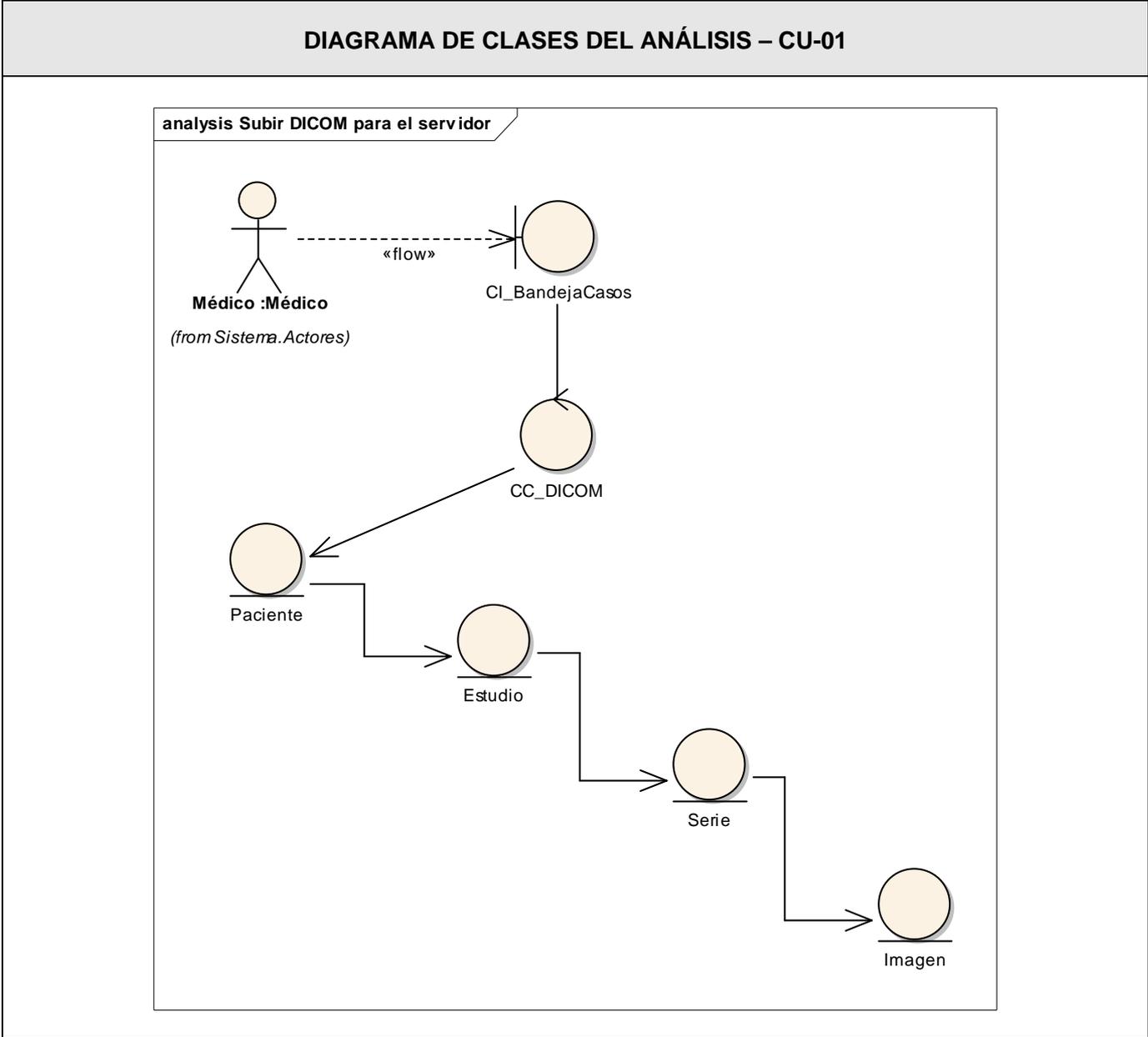
botón del Mouse oprimido arrastra la imagen.	las coordenadas actuales.
--	---------------------------

CU-21 – Editar perfil

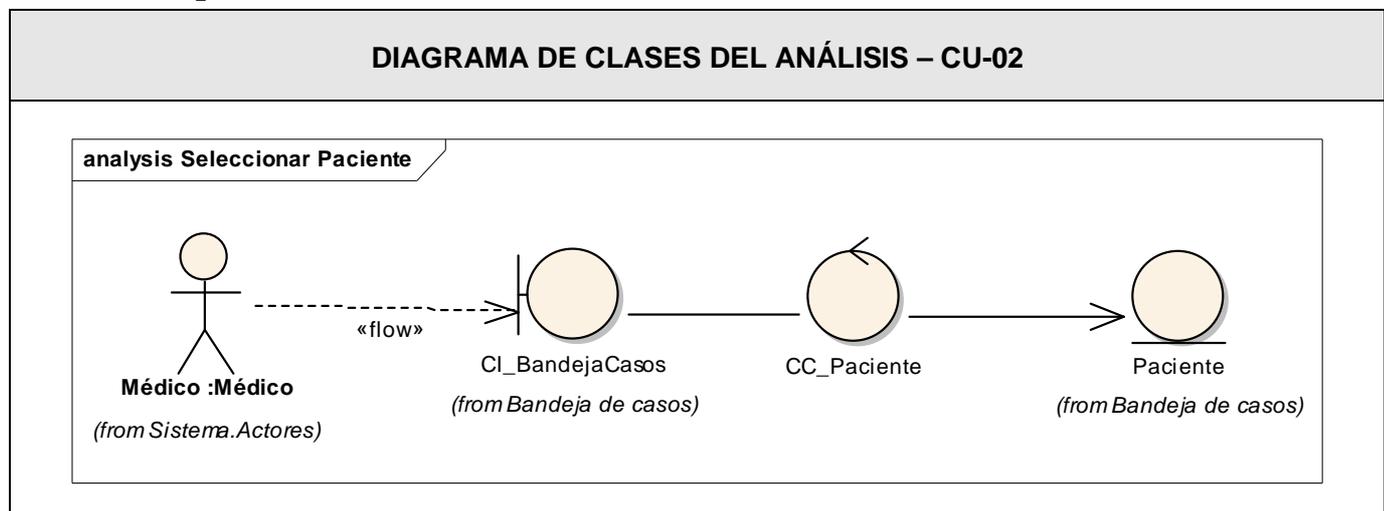
CU-17	Editar perfil.	
Actores:	Médico	
Propósito:	Cambiar los datos personales del Médico .	
Resumen:	El Médico edita su información personal	
Referencias:	RF-8	
Precondiciones:		
Poscondiciones:	1 – Datos personales modificados.	
Curso normal de los eventos		
Acciones del Actor	Respuestas del Sistema	
1 – Selecciona la opción de modificar su perfil.	1.1 – Muestra la interfaz correspondiente con los datos actuales del perfil del Médico .	
2 – Introduce los nuevos cambios a su perfil y presiona actualizar.	2.1 – Actualiza los datos.	

ANEXO II. DIAGRAMAS DE CLASES DEL ANÁLISIS

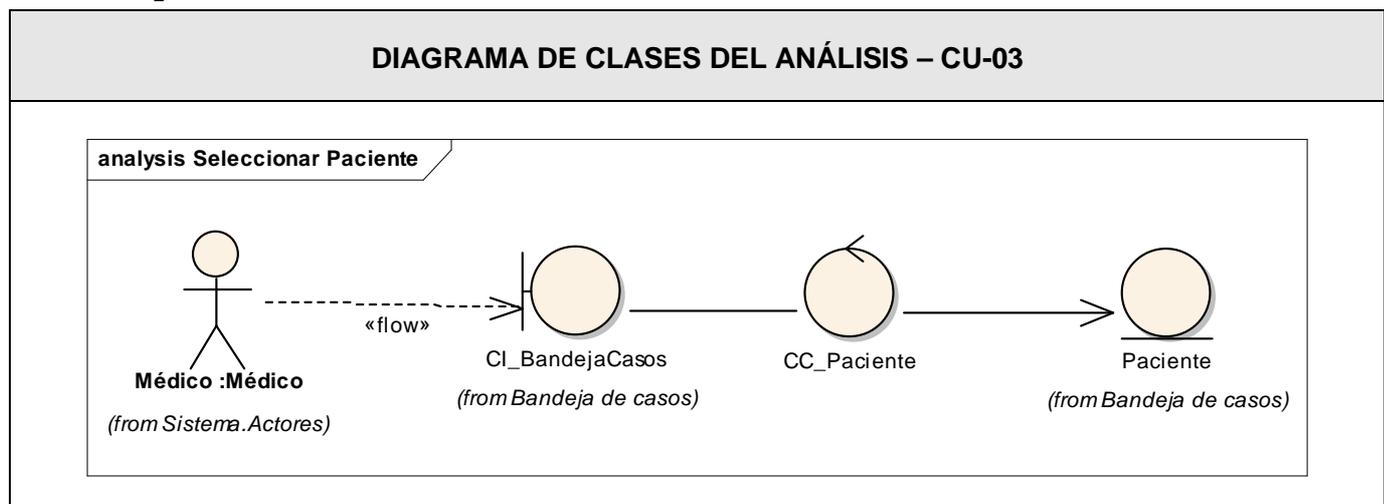
Subir DICOM para el servidor



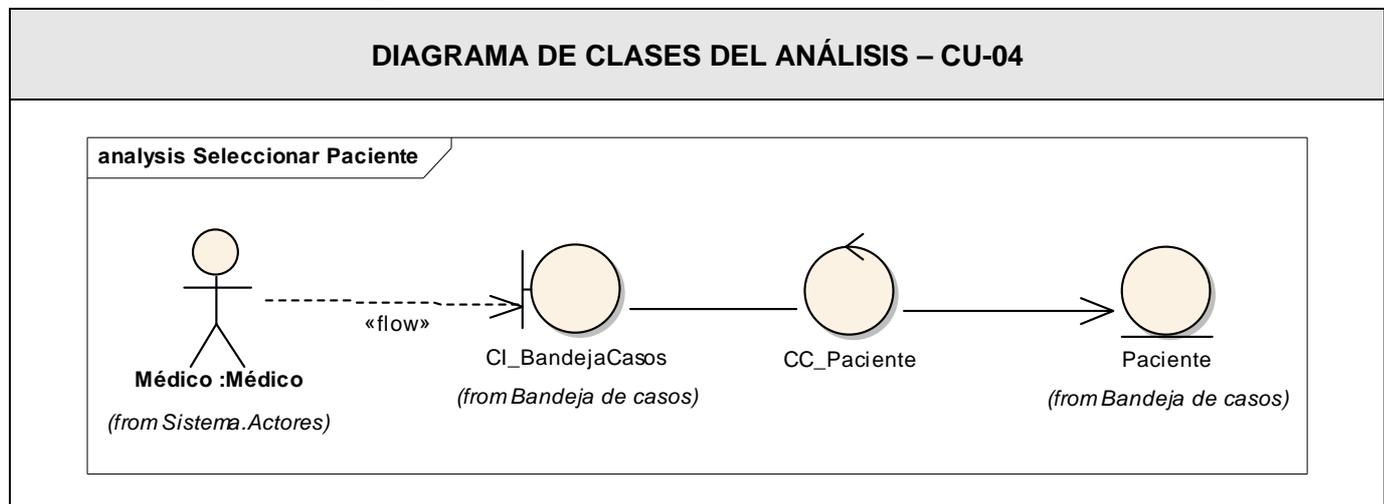
Seleccionar paciente



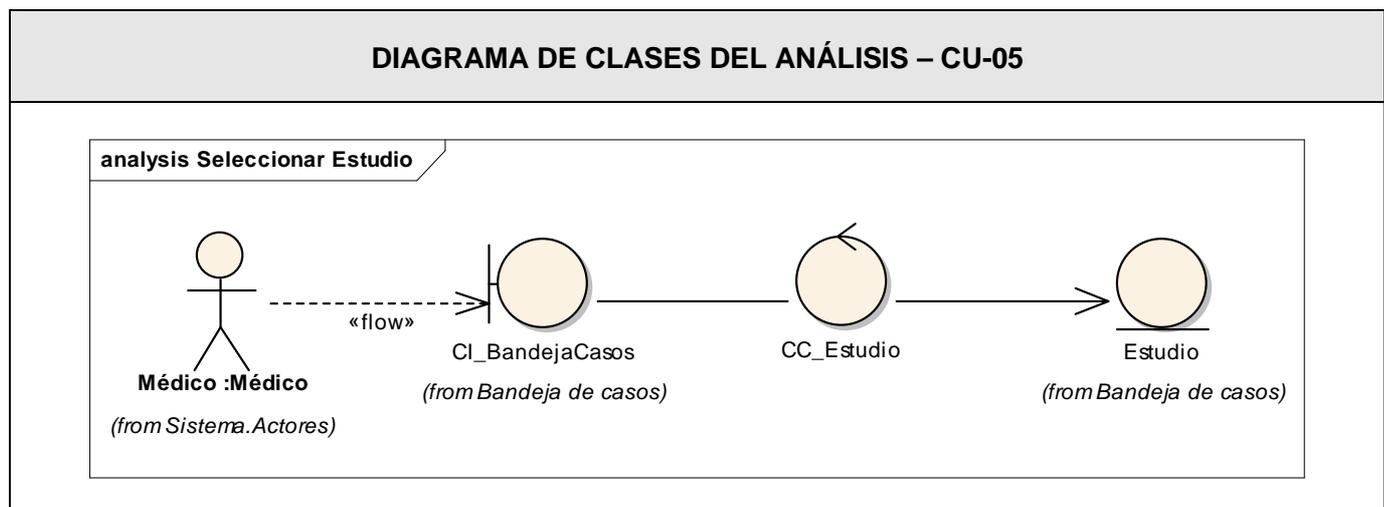
Eliminar paciente



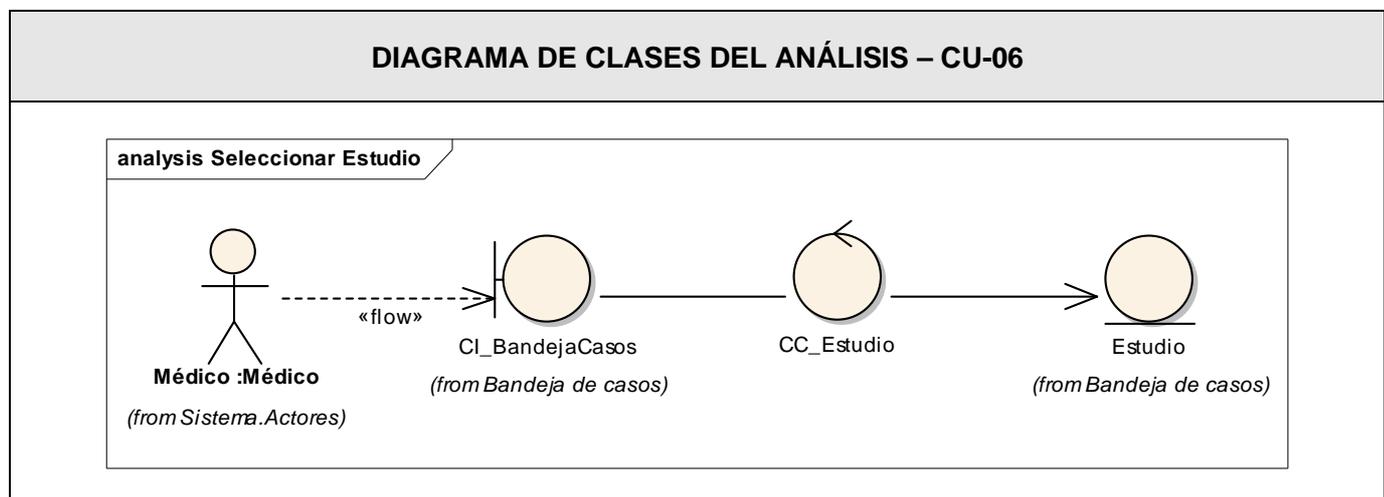
Realizar anotaciones al paciente



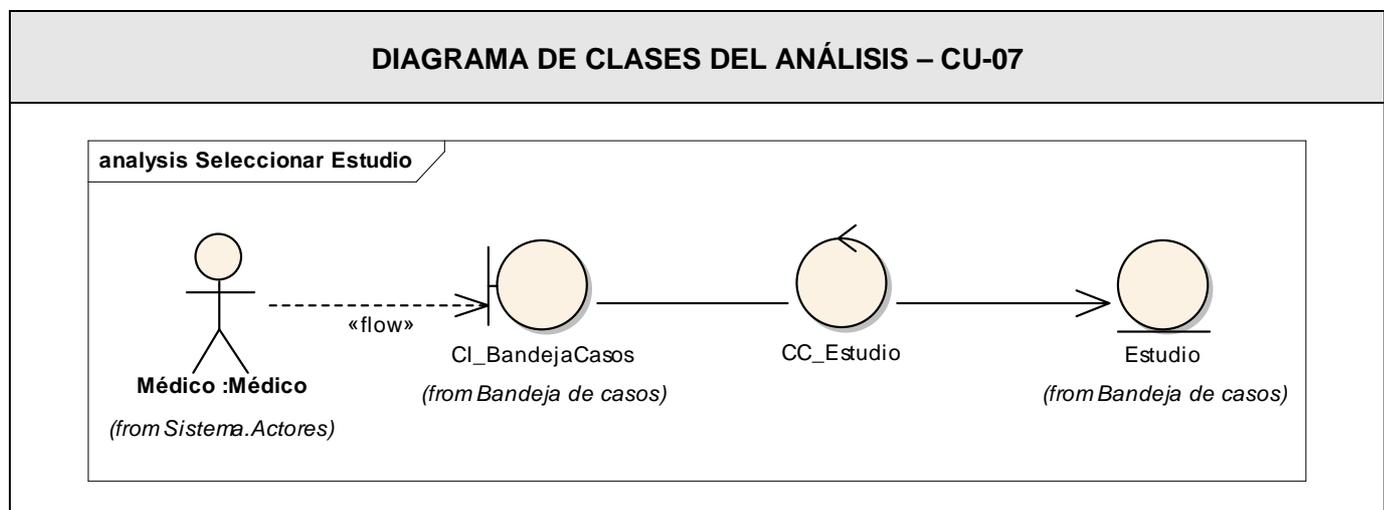
Seleccionar estudio



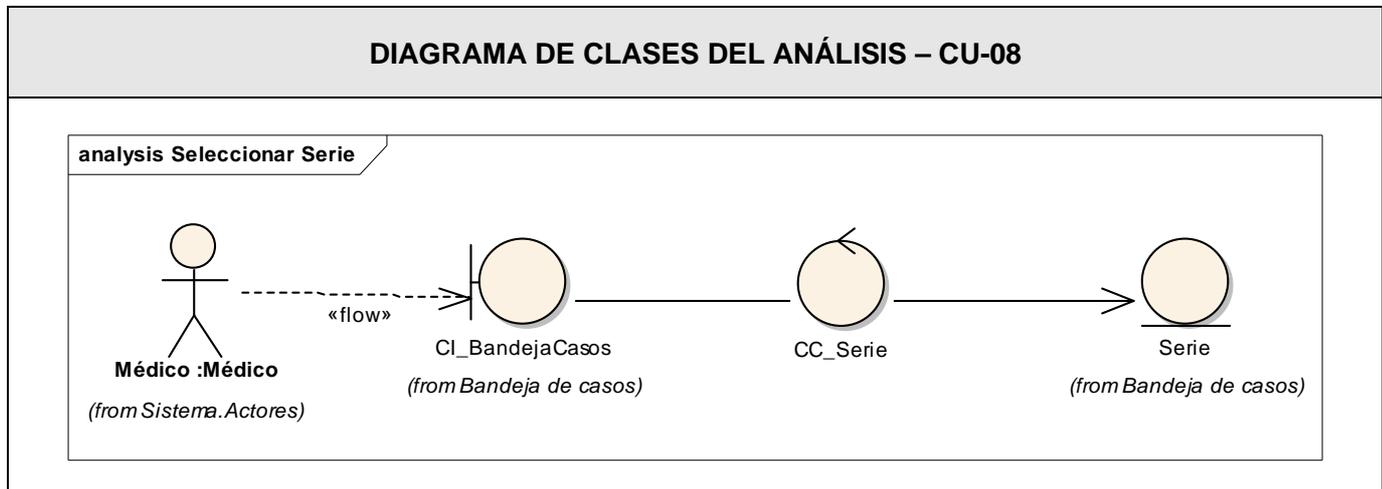
Eliminar estudio



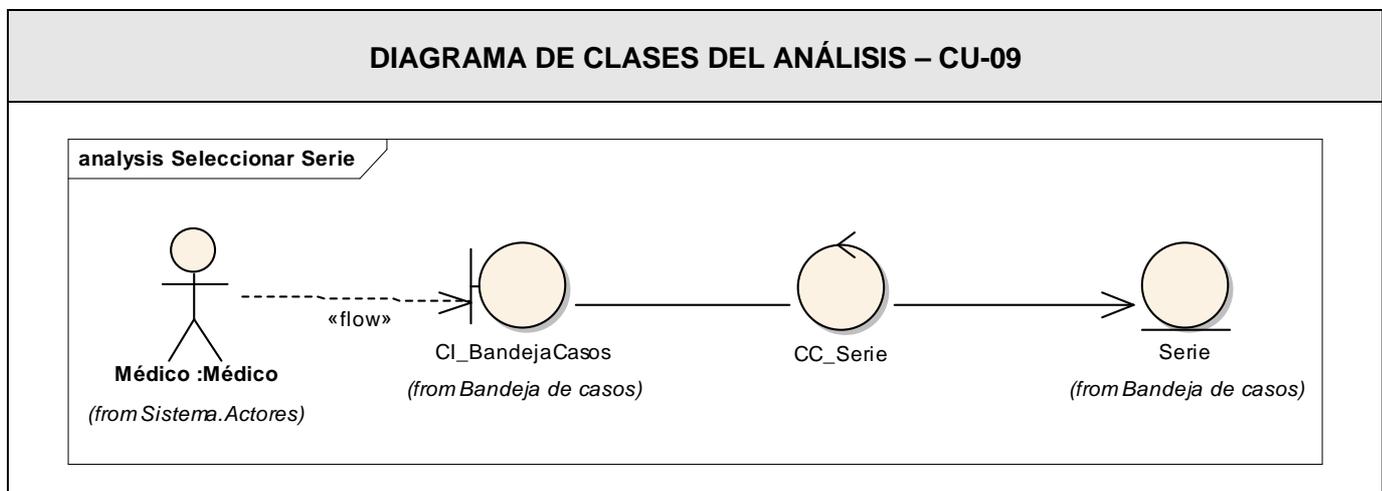
Realizar anotaciones al estudio



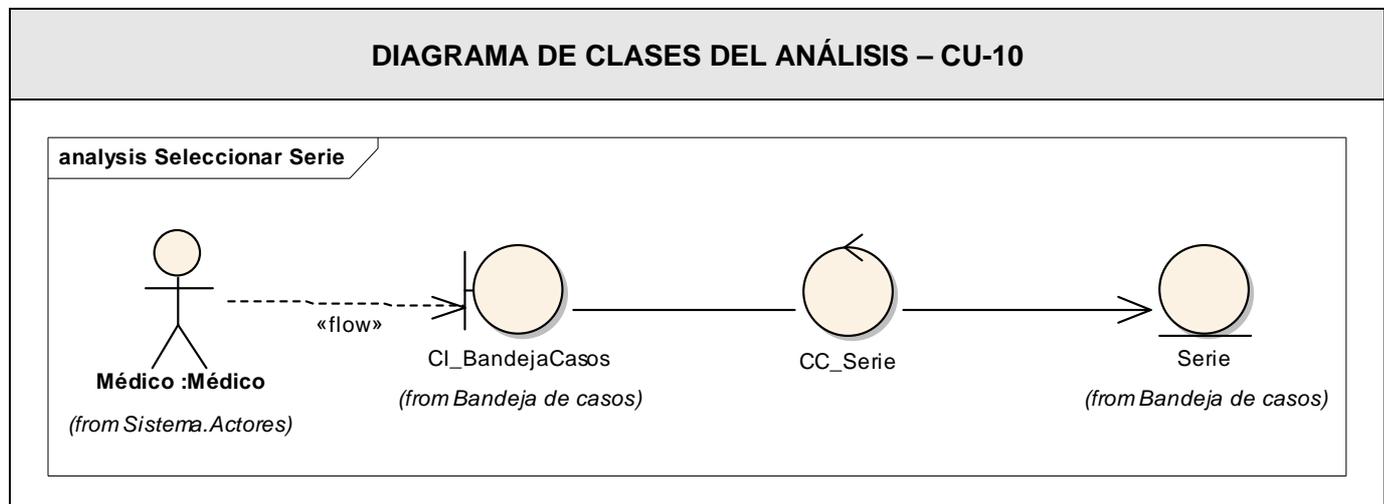
Seleccionar serie



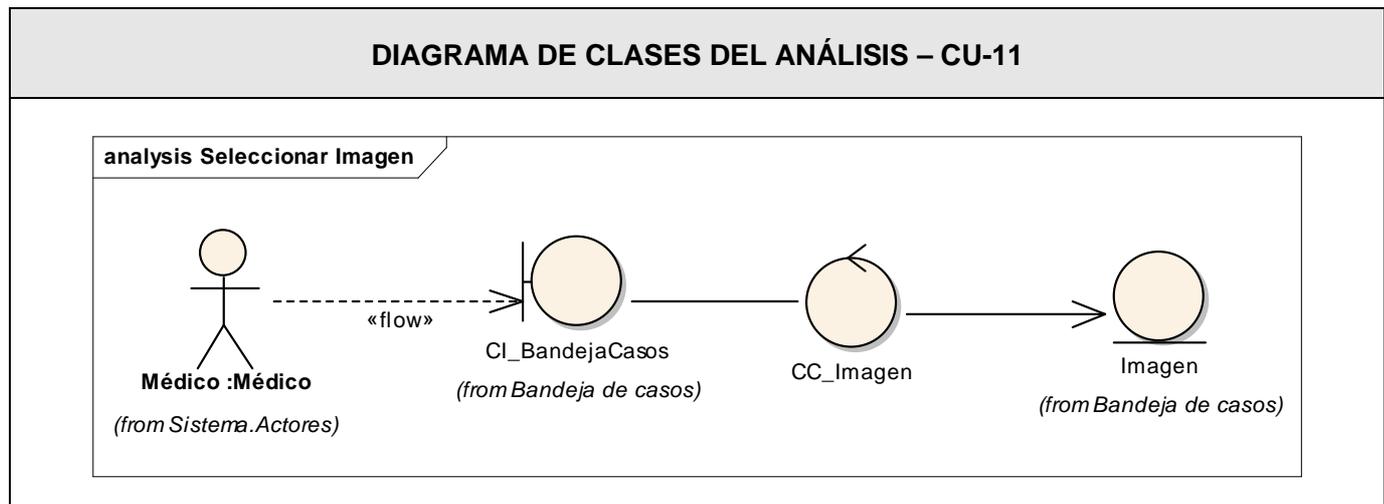
Eliminar serie



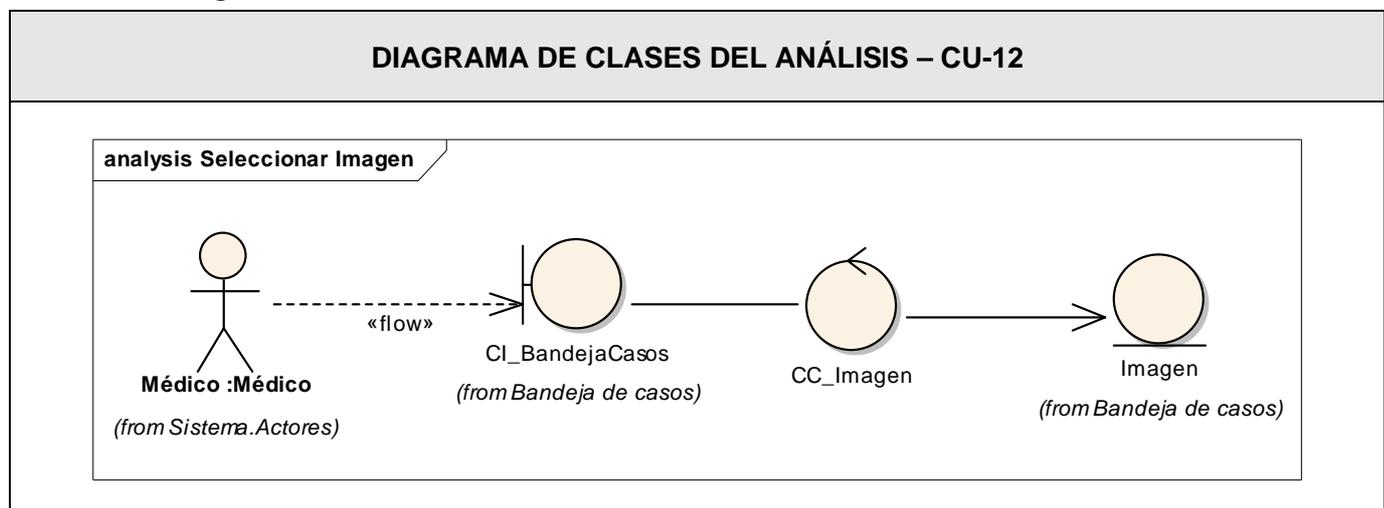
Realizar anotaciones a la serie



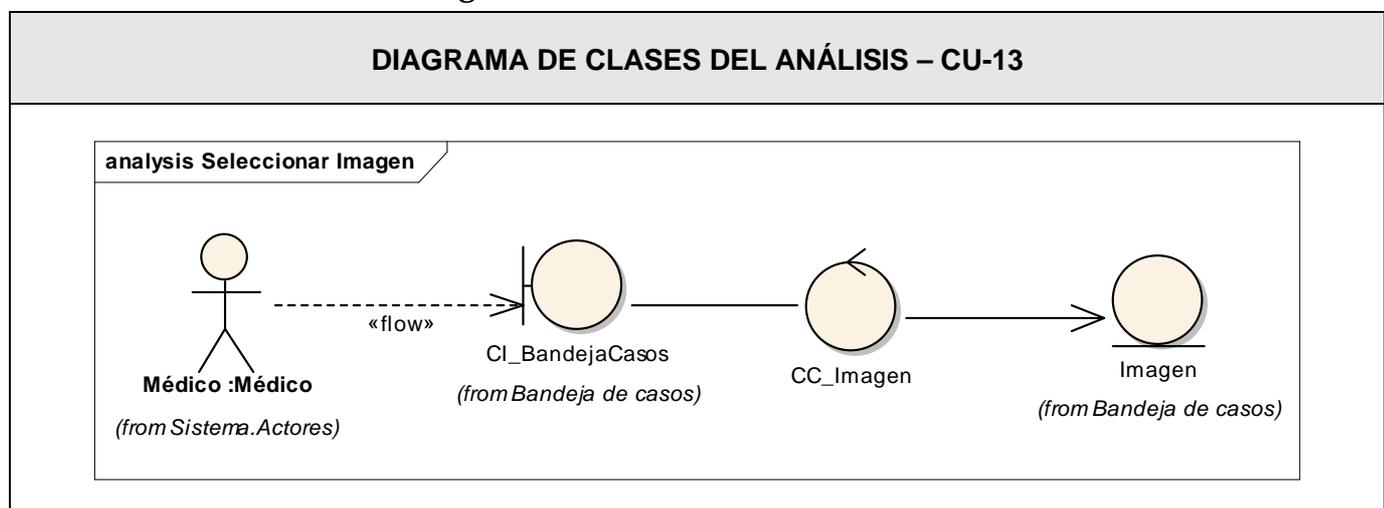
Seleccionar imagen



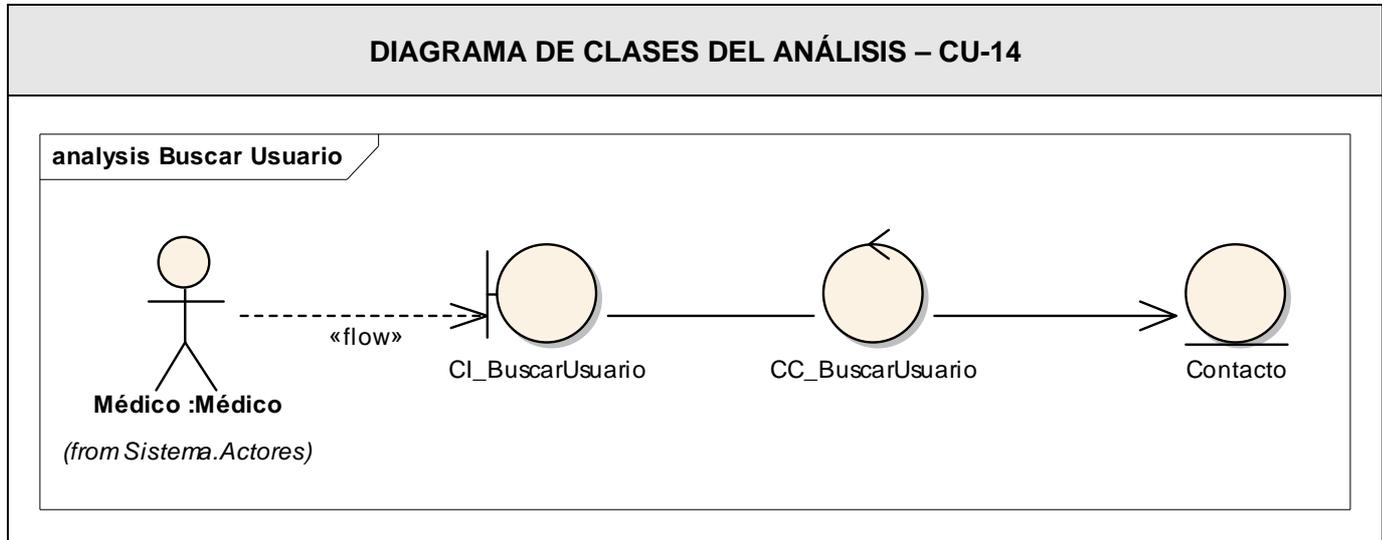
Eliminar imagen



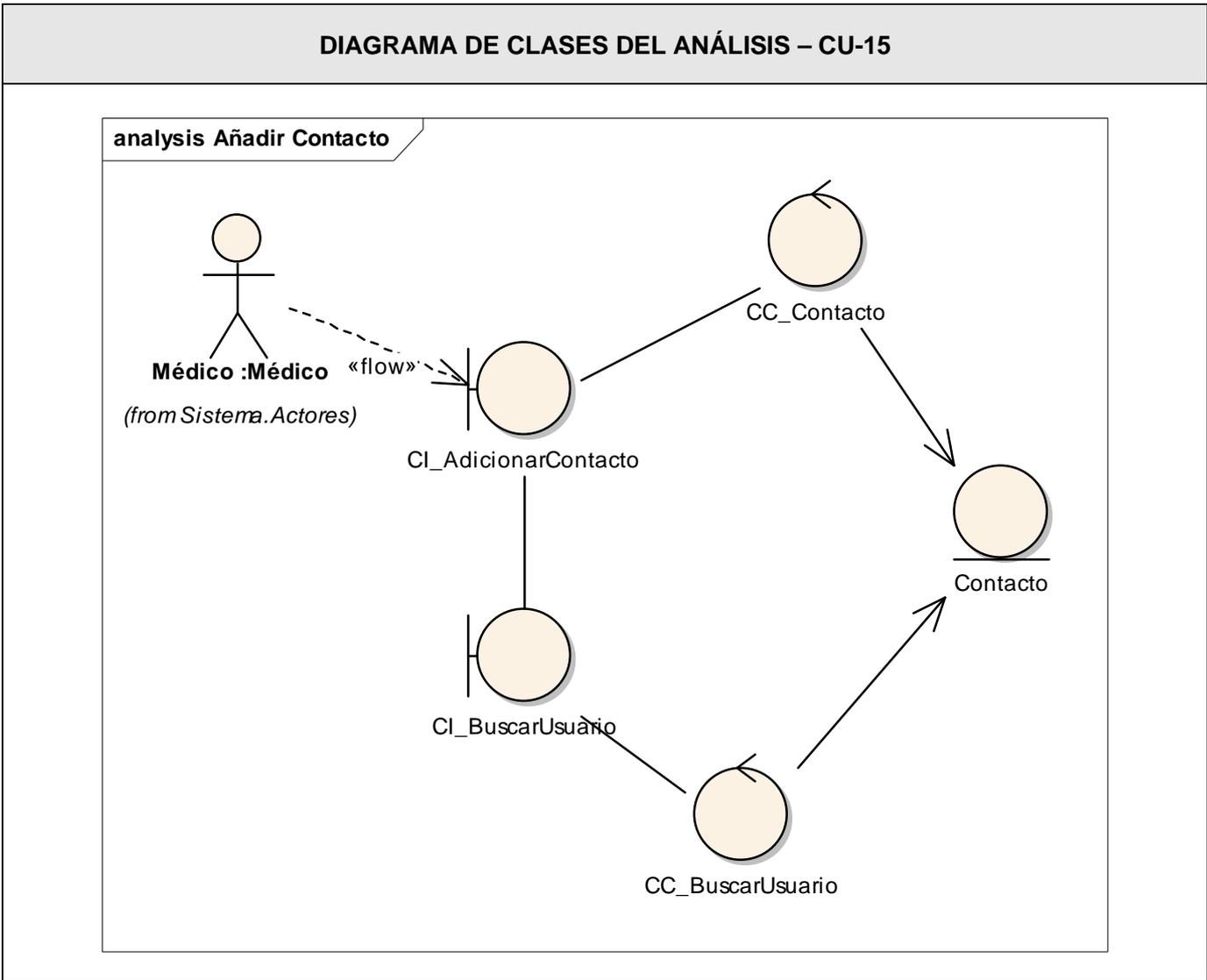
Realizar anotaciones a la imagen



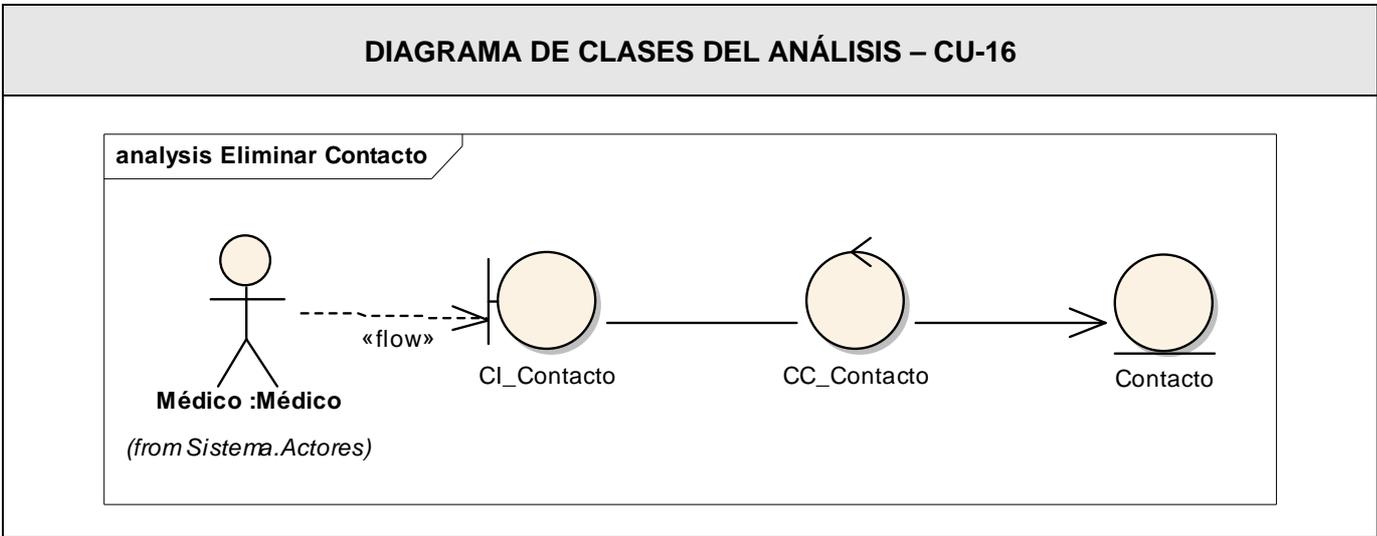
Buscar usuario



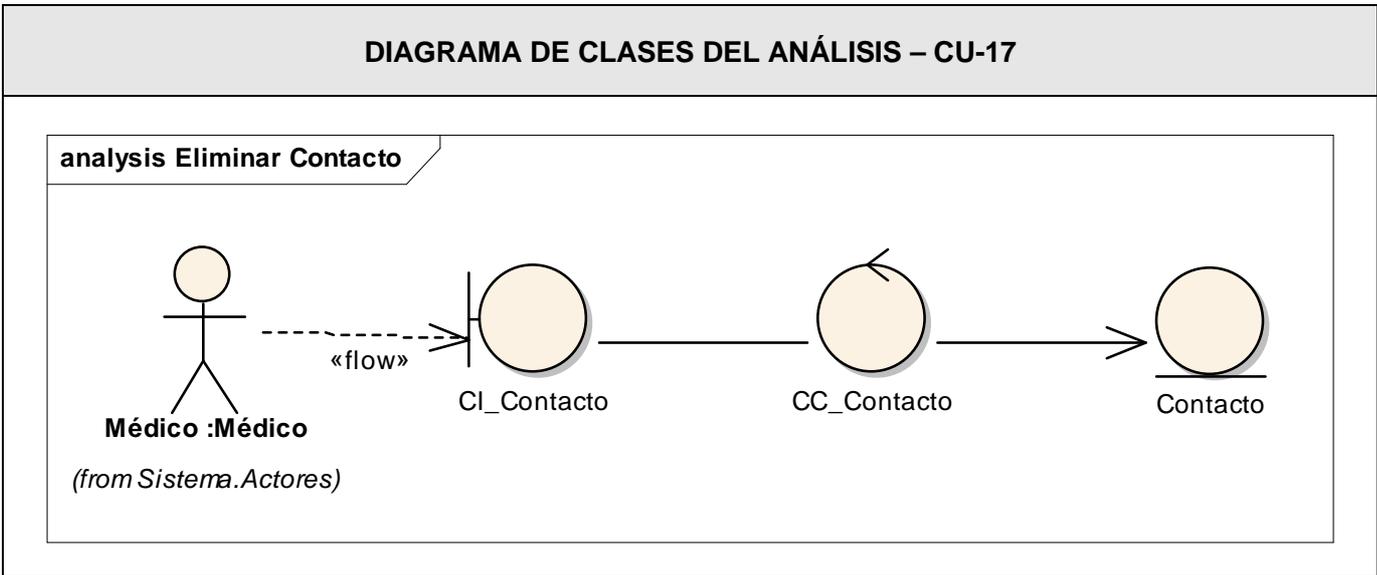
Añadir contacto



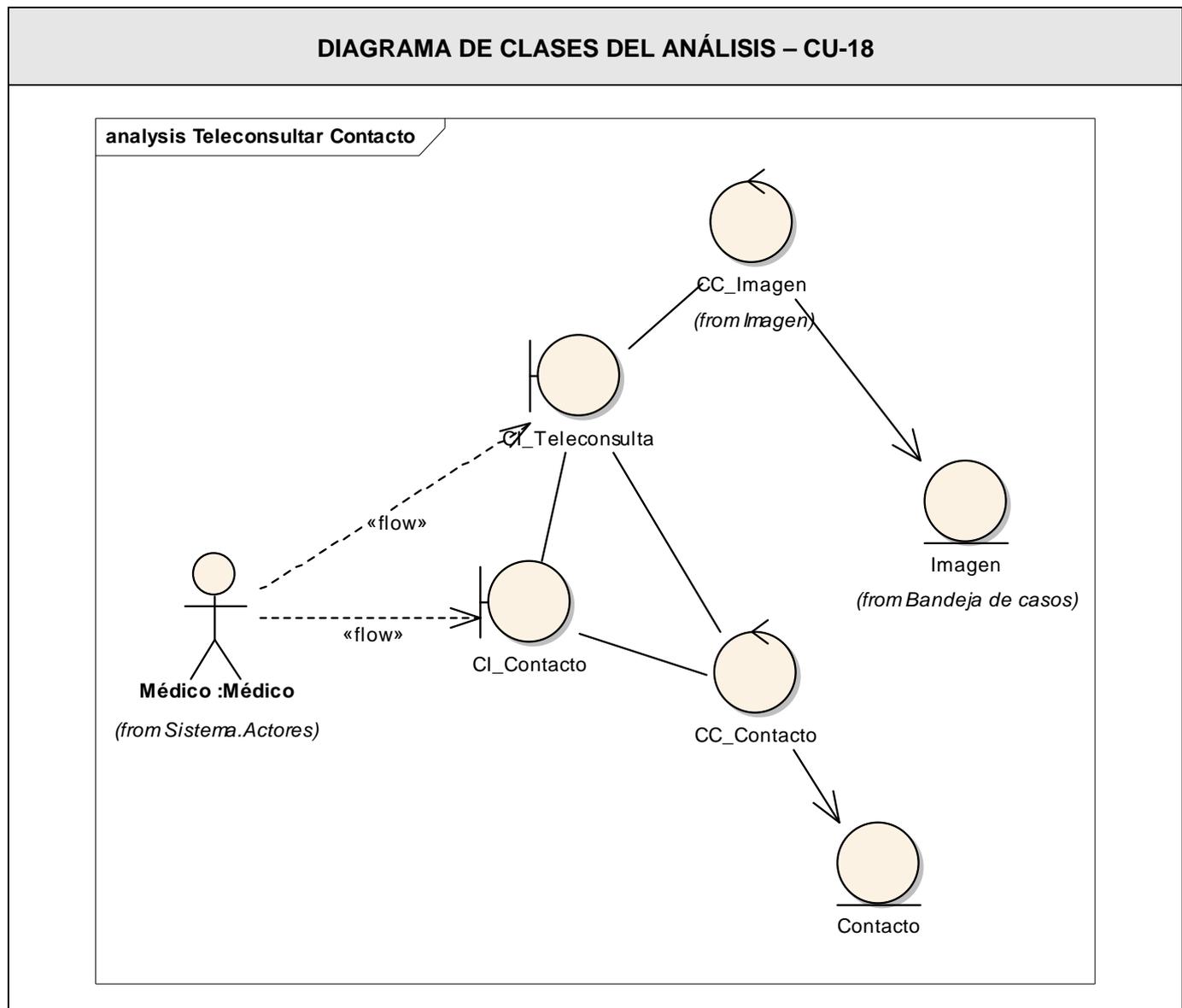
Eliminar contacto



Renombrar contacto



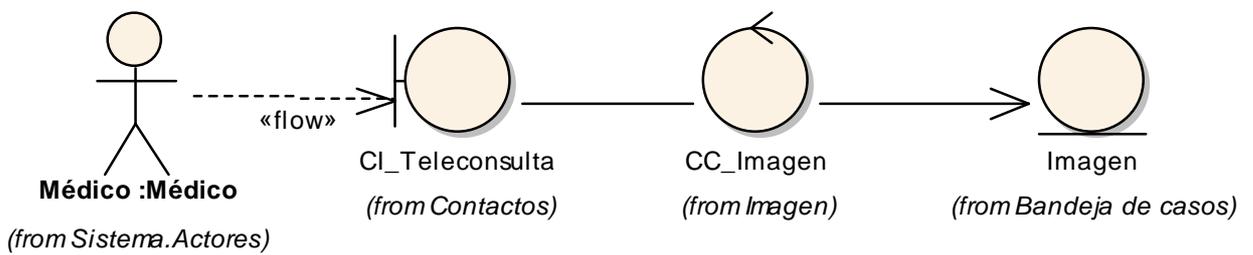
Teleconsultar contacto



Hacer zoom sobre la imagen

DIAGRAMA DE CLASES DEL ANÁLISIS – CU-19

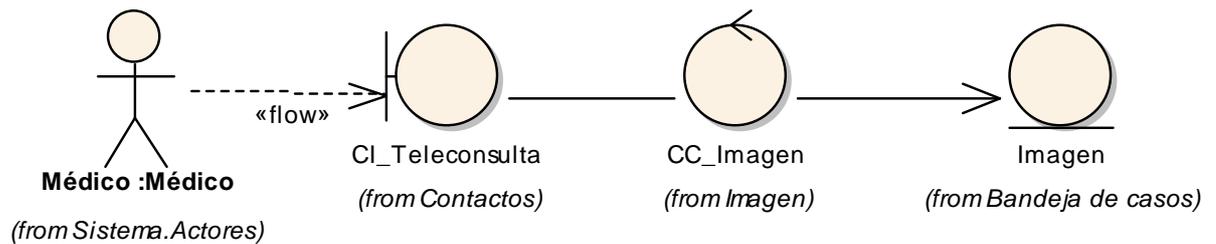
analysis Realizar Zoom



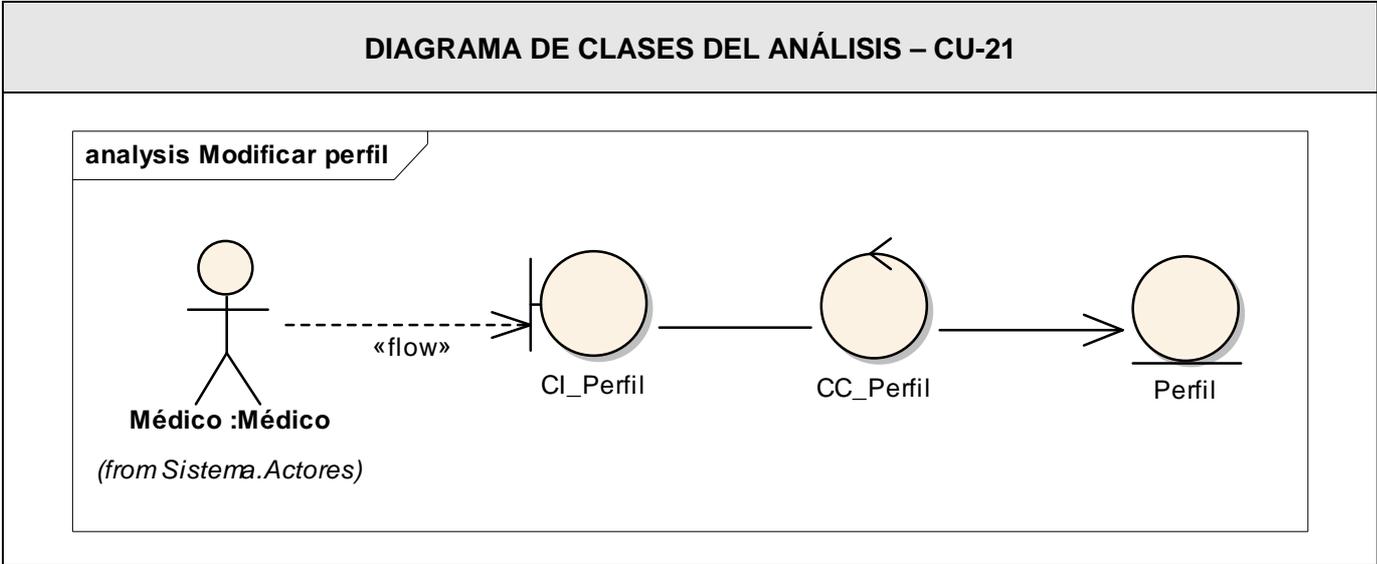
Mover la imagen

DIAGRAMA DE CLASES DEL ANÁLISIS – CU-20

analysis Realizar Zoom



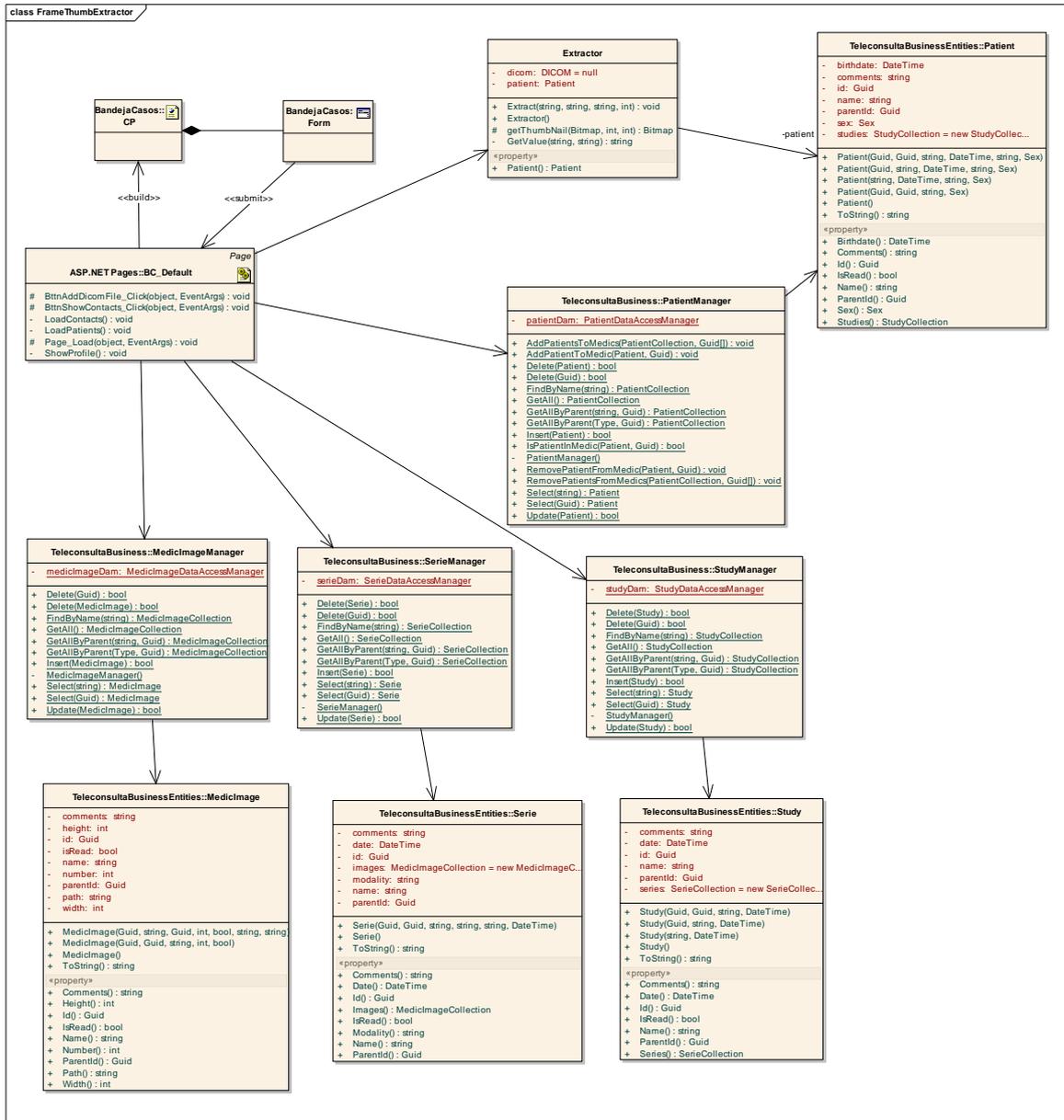
Editar perfil



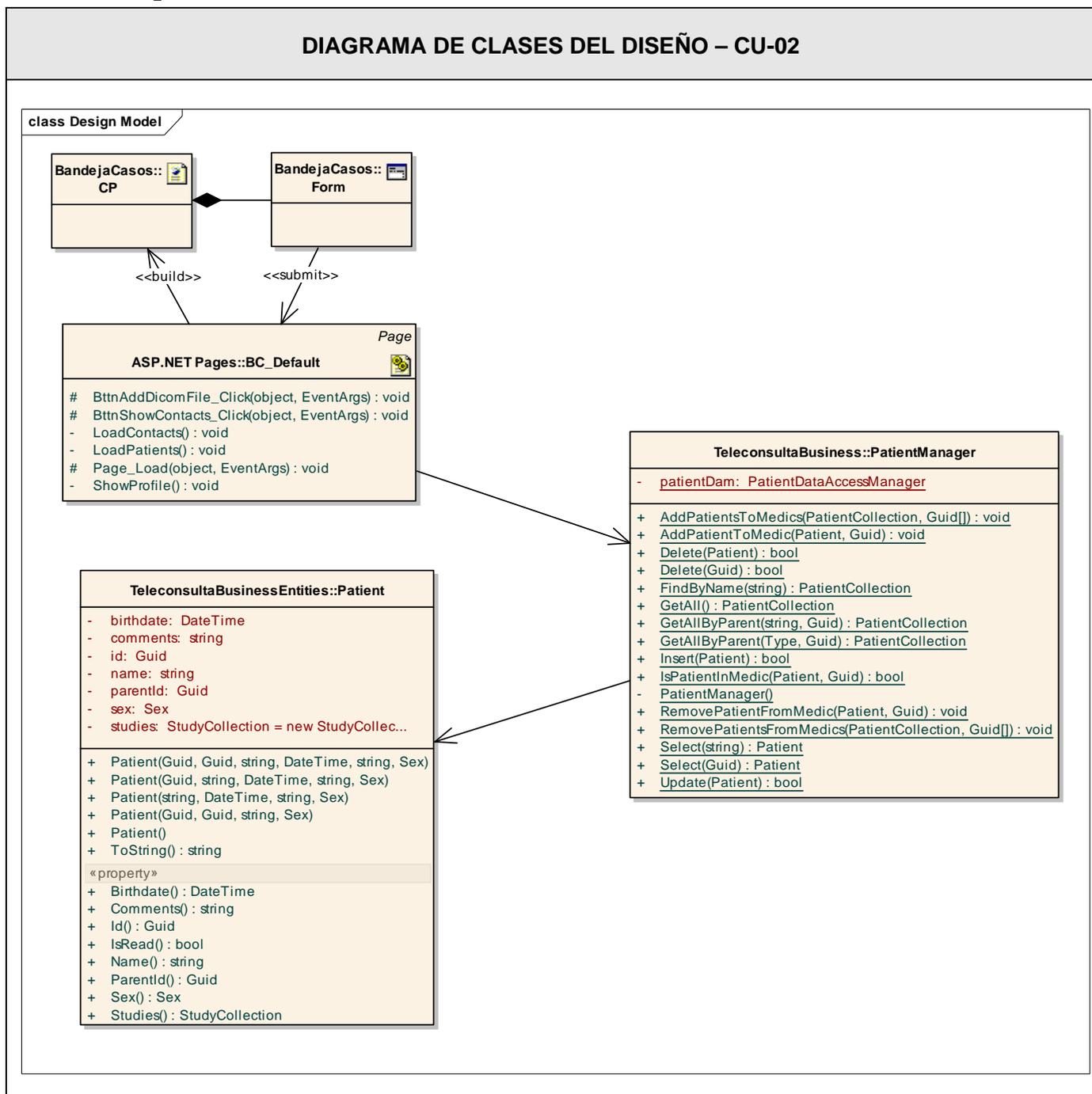
ANEXO III. DIAGRAMAS DE CLASES DEL DISEÑO

Subir DICOM para el servidor

DIAGRAMA DE CLASES DEL DISEÑO – CU-01



Seleccionar paciente



Eliminar paciente

DIAGRAMA DE CLASES DEL DISEÑO – CU-03

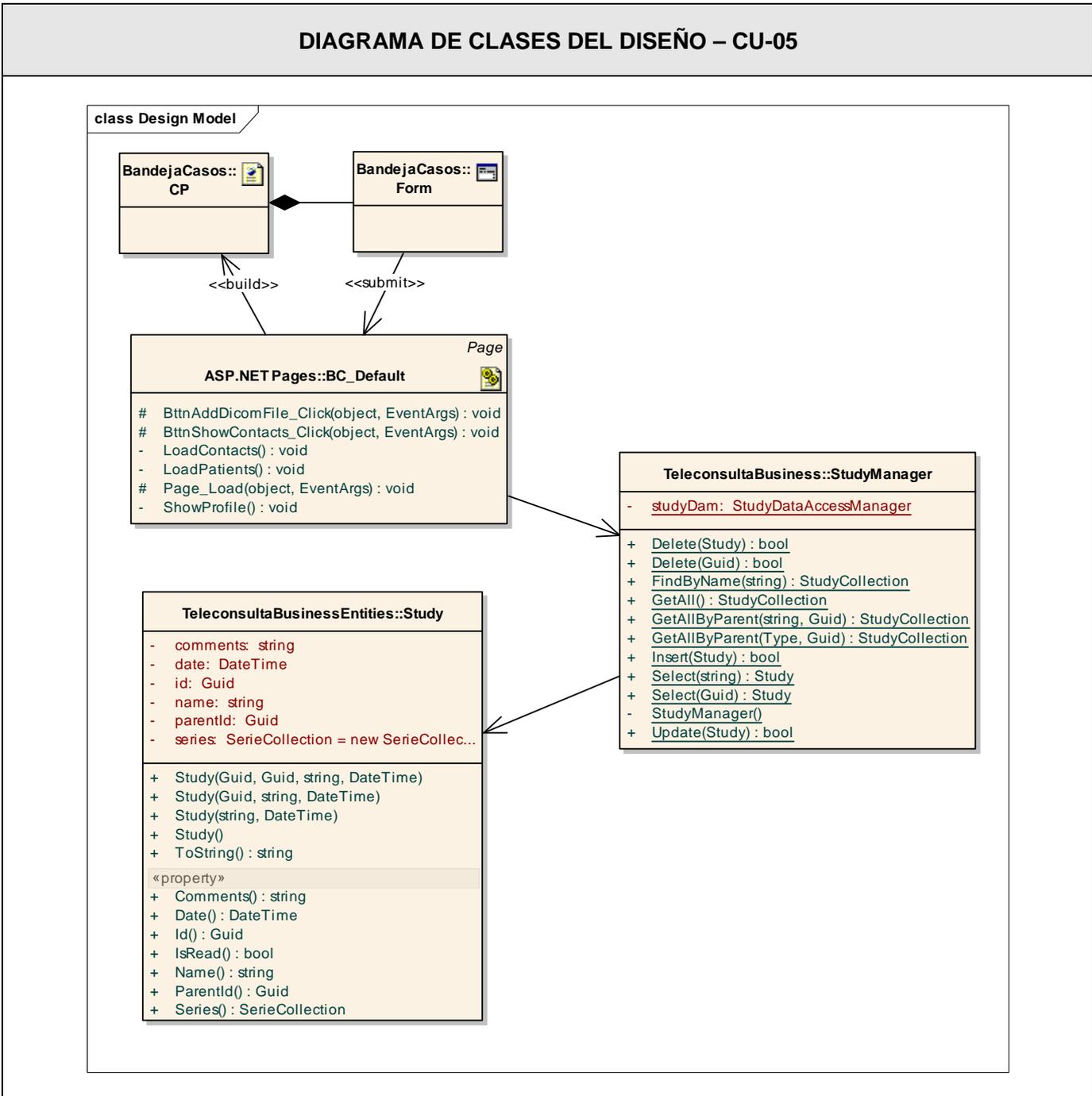
<Similar al Diagrama de clases del diseño: Seleccionar paciente>

Realizar anotaciones en el paciente

DIAGRAMA DE CLASES DEL DISEÑO – CU-04

<Similar al Diagrama de clases del diseño: Seleccionar paciente>

Seleccionar estudio



Eliminar estudio

DIAGRAMA DE CLASES DEL DISEÑO – CU-06

<Similar al Diagrama de clases del diseño: Seleccionar estudio>

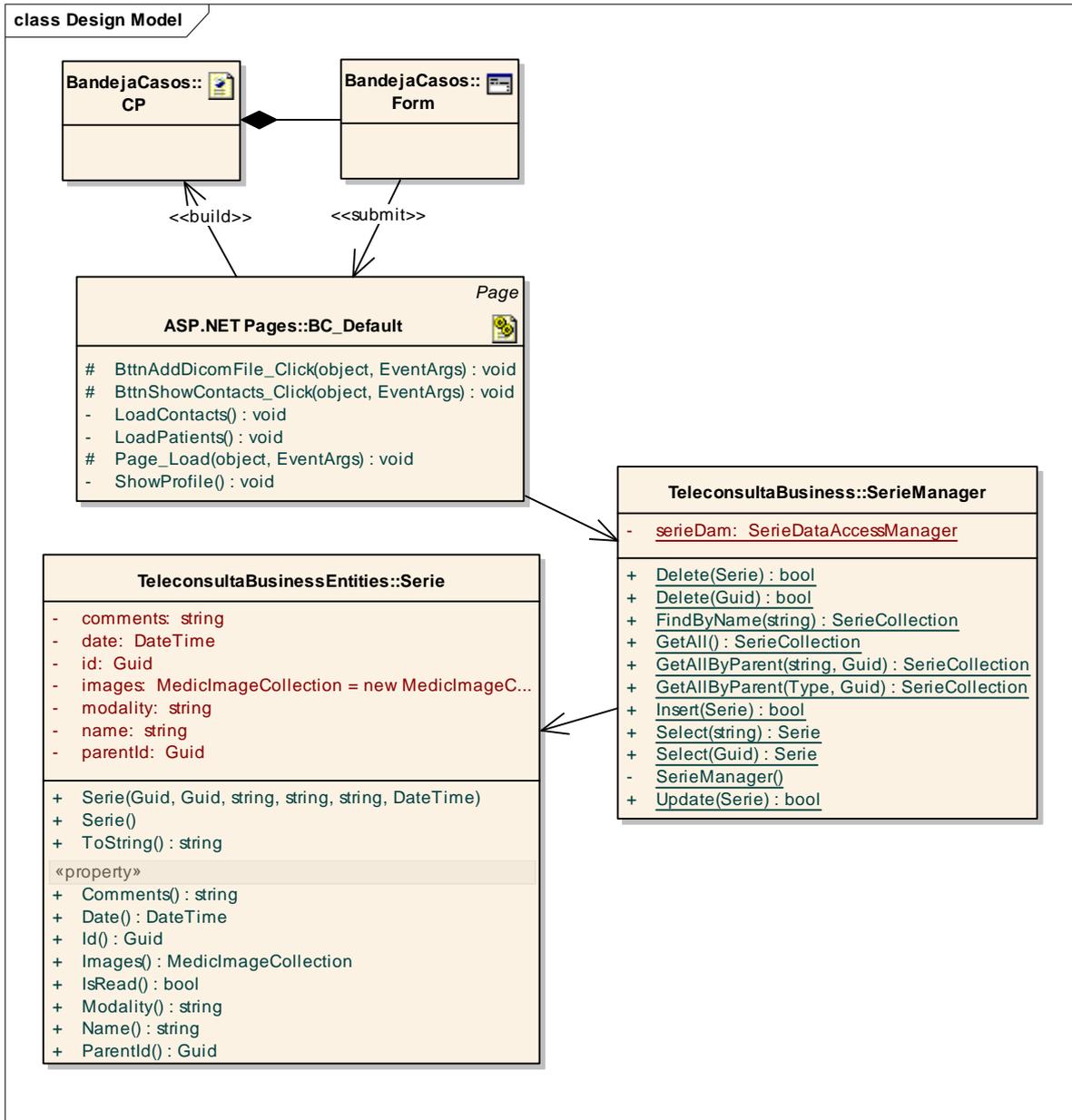
Realizar anotaciones en el estudio

DIAGRAMA DE CLASES DEL DISEÑO – CU-07

<Similar al Diagrama de clases del diseño: Seleccionar estudio>

Seleccionar serie

DIAGRAMA DE CLASES DEL DISEÑO – CU-08



Eliminar serie

DIAGRAMA DE CLASES DEL DISEÑO – CU-09

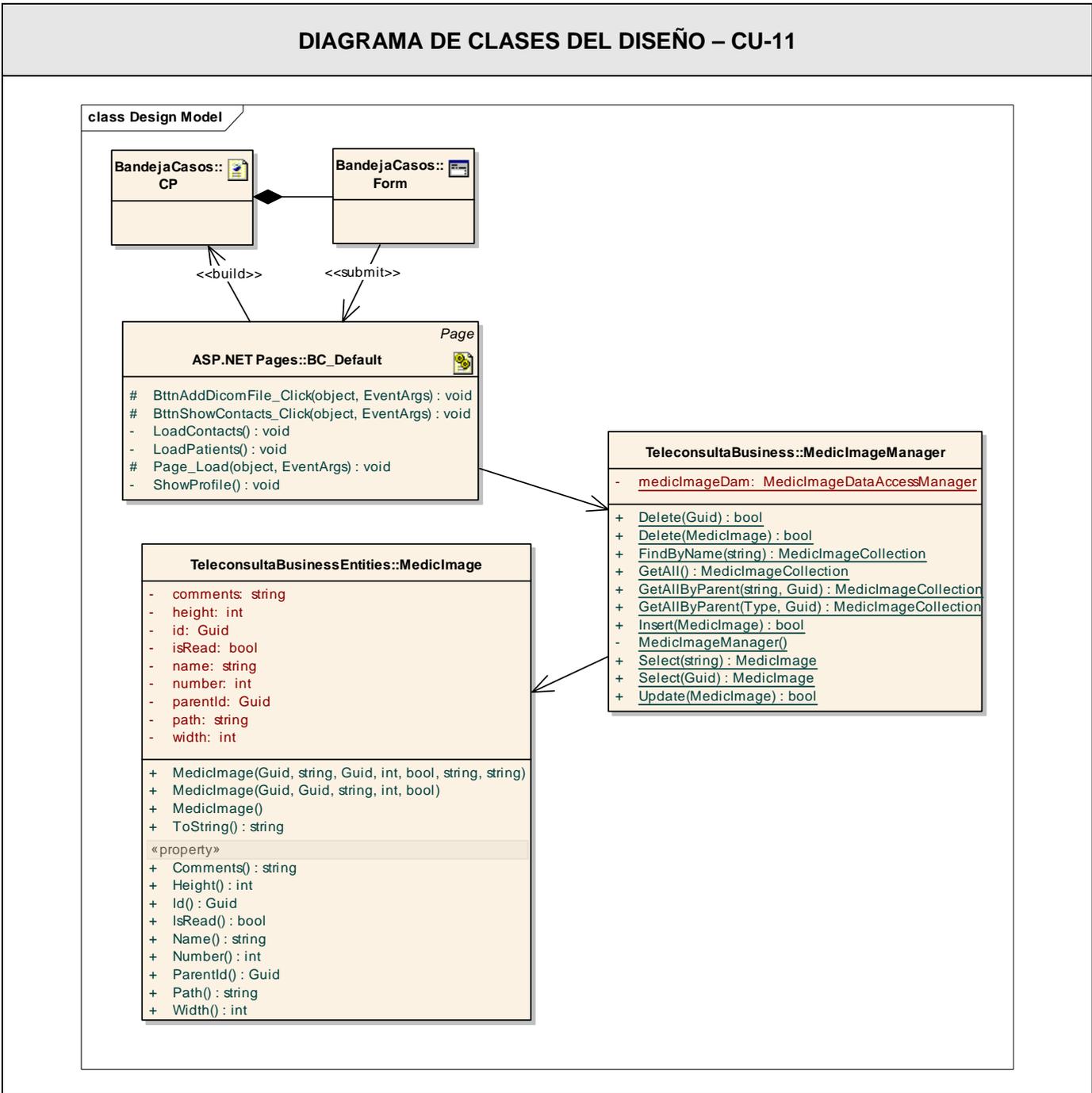
<Similar al Diagrama de clases del diseño: Seleccionar serie>

Realizar anotaciones en la serie

DIAGRAMA DE CLASES DEL DISEÑO – CU-10

<Similar al Diagrama de clases del diseño: Seleccionar serie>

Seleccionar imagen



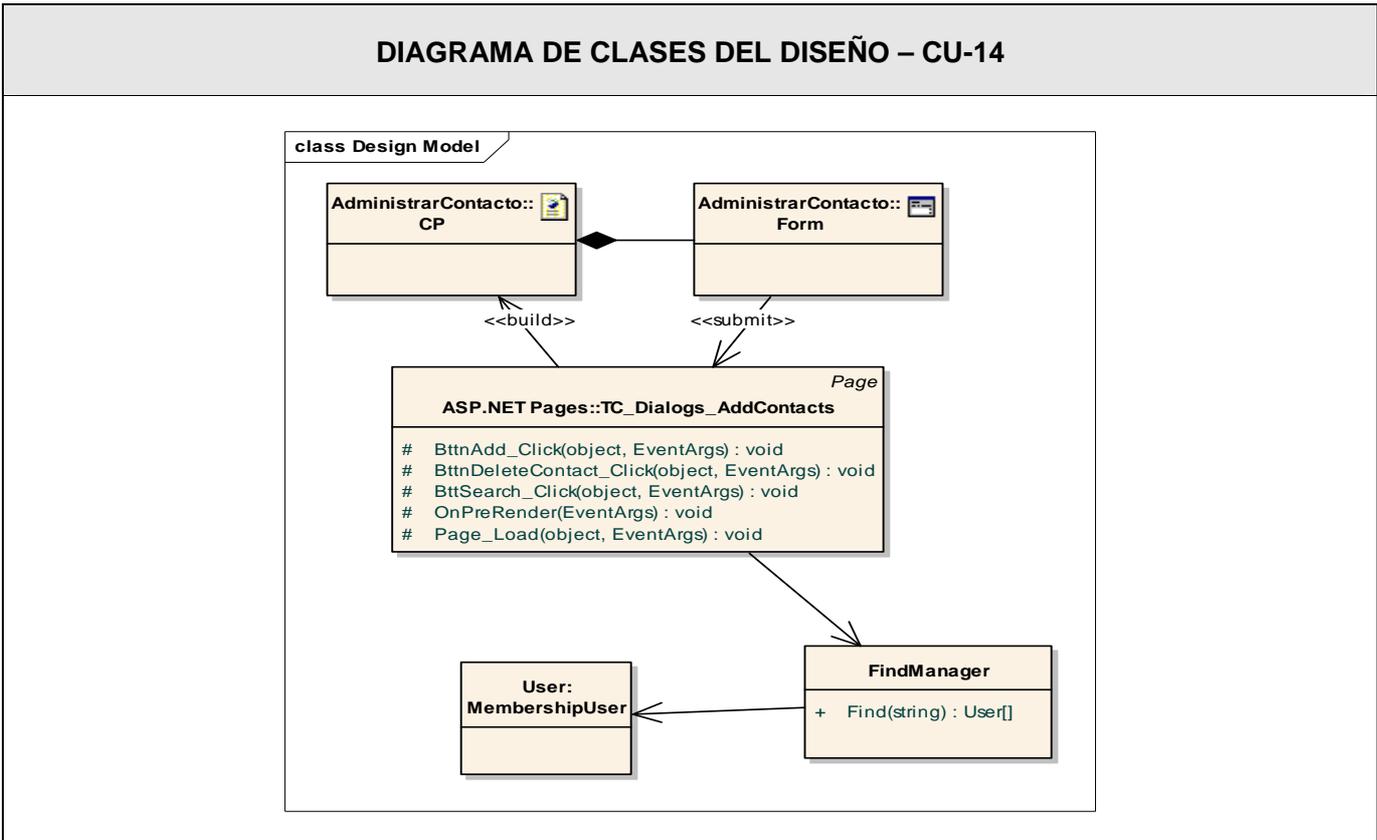
Eliminar imagen

DIAGRAMA DE CLASES DEL DISEÑO – CU-12
<Similar al Diagrama de clases del diseño: Seleccionar imagen>

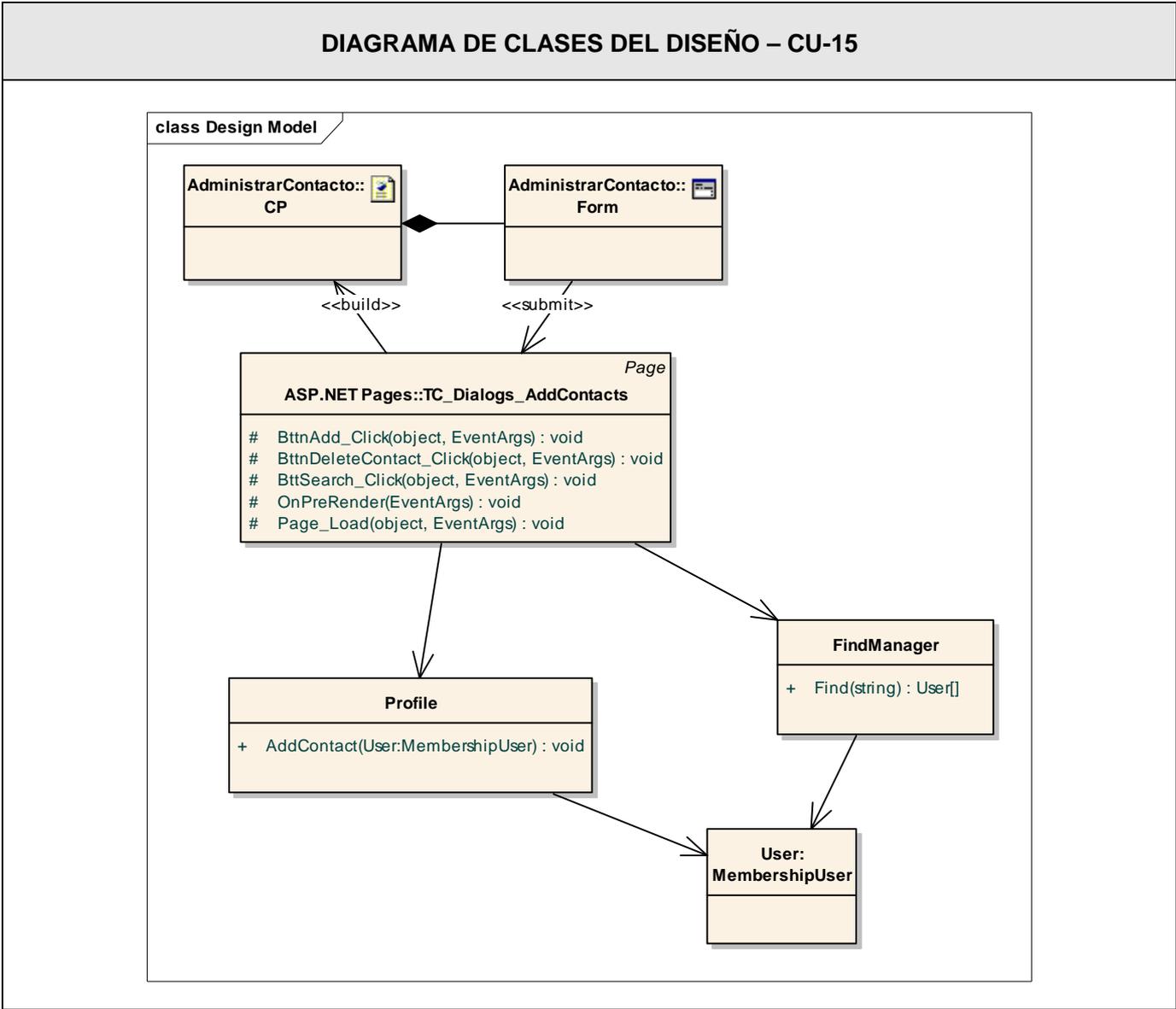
Realizar anotaciones en la imagen

DIAGRAMA DE CLASES DEL DISEÑO – CU-13
<Similar al Diagrama de clases del diseño: Seleccionar imagen>

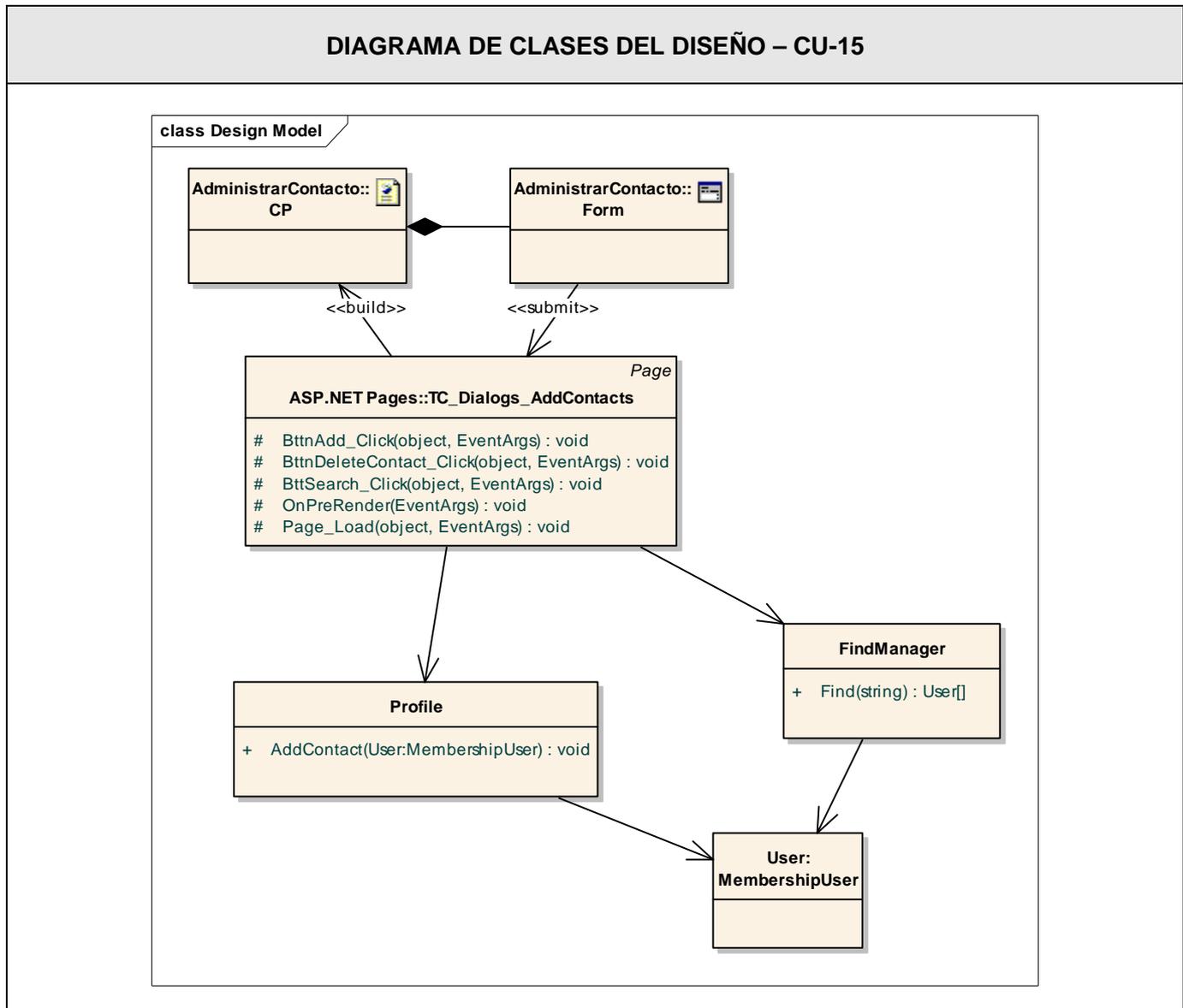
Buscar usuarios



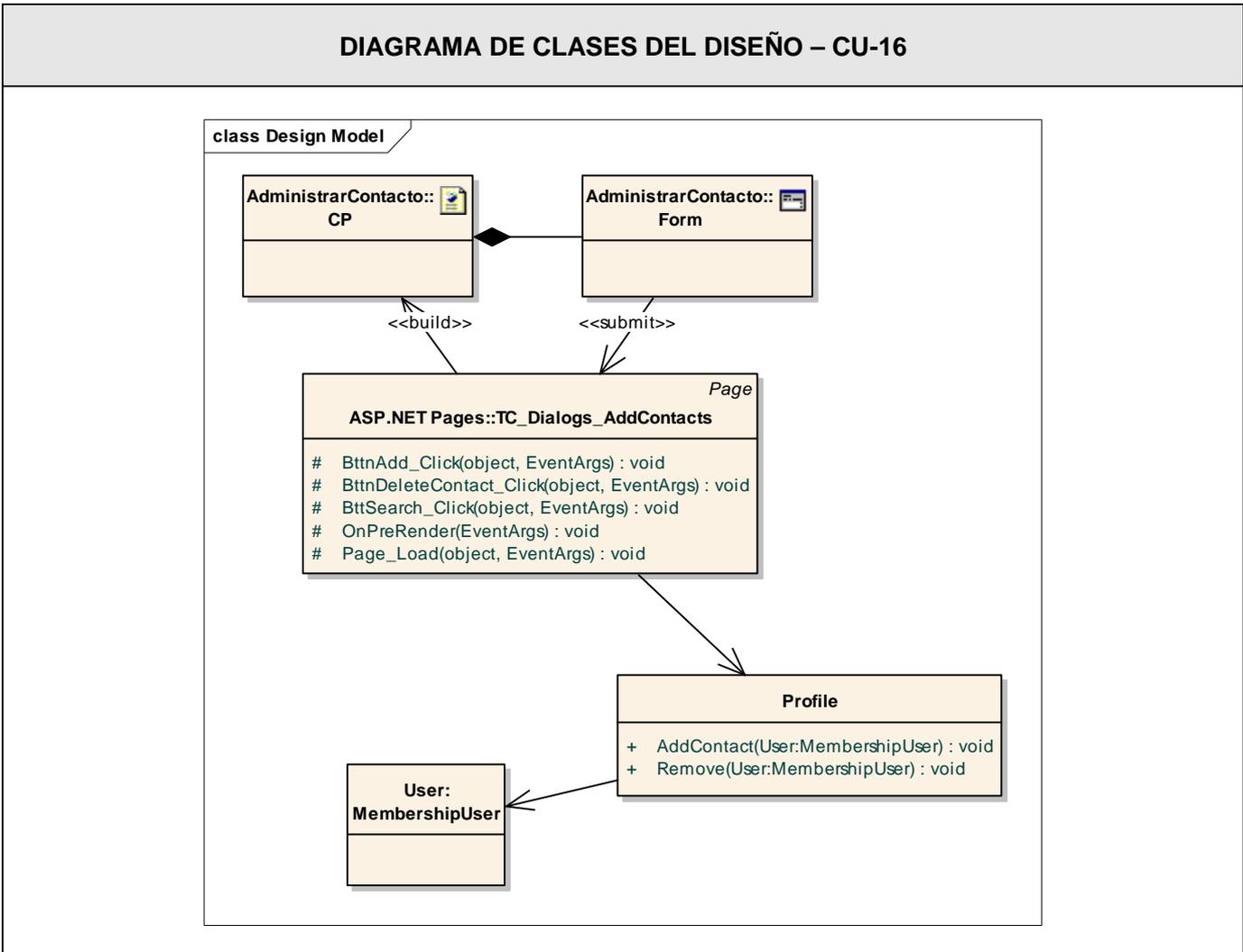
Añadir contacto



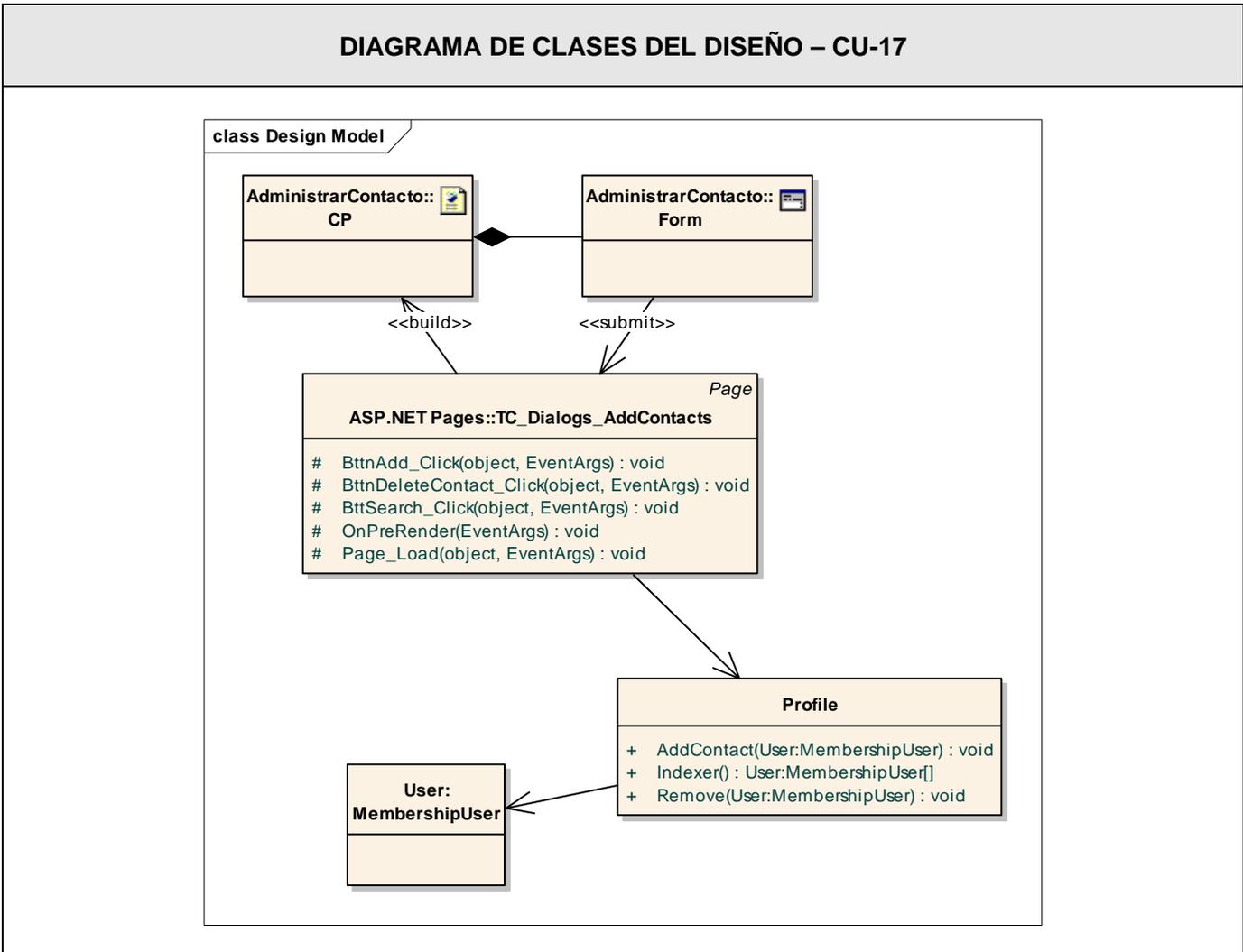
Añadir contacto



Eliminar contacto

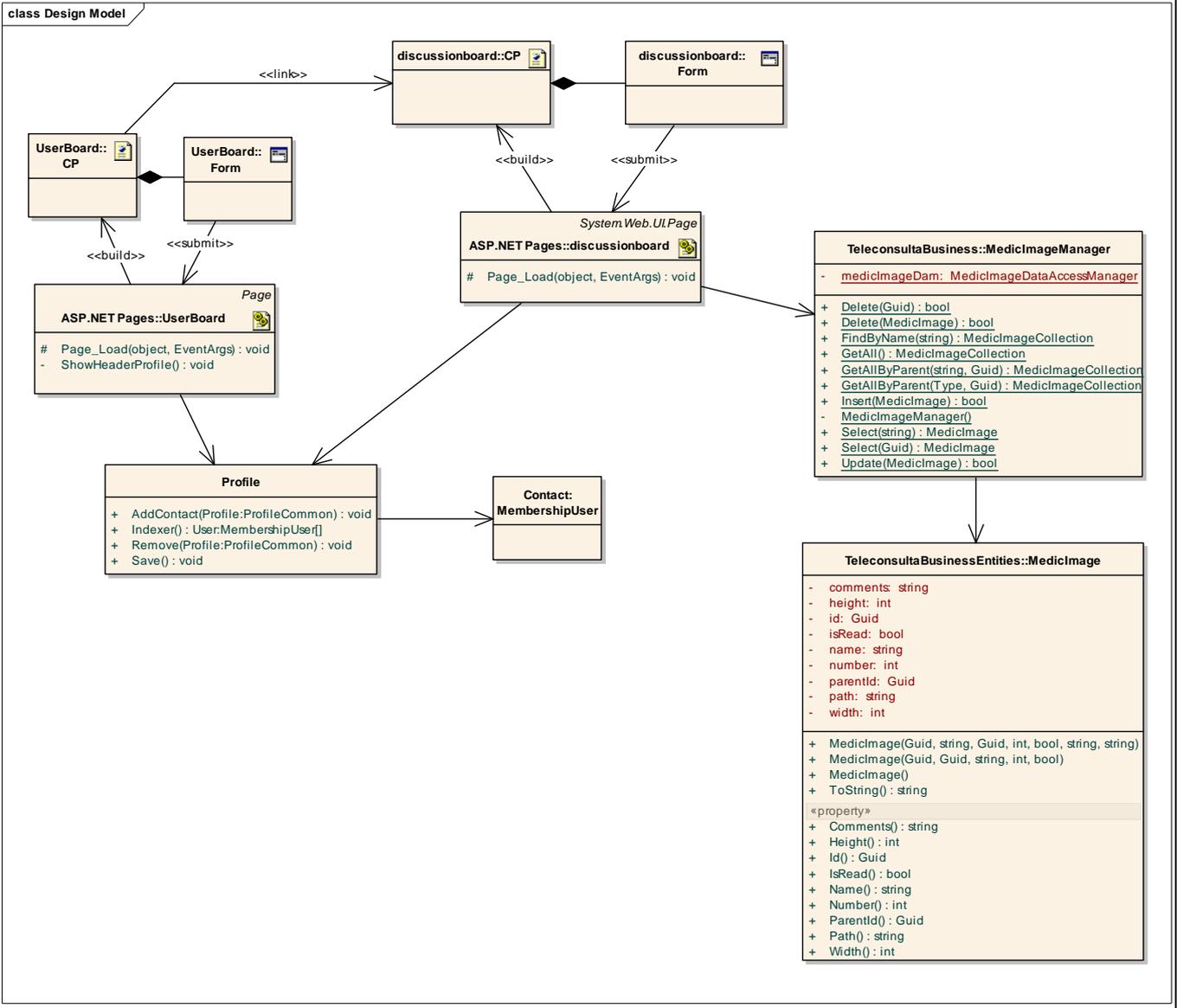


Renombrar contacto



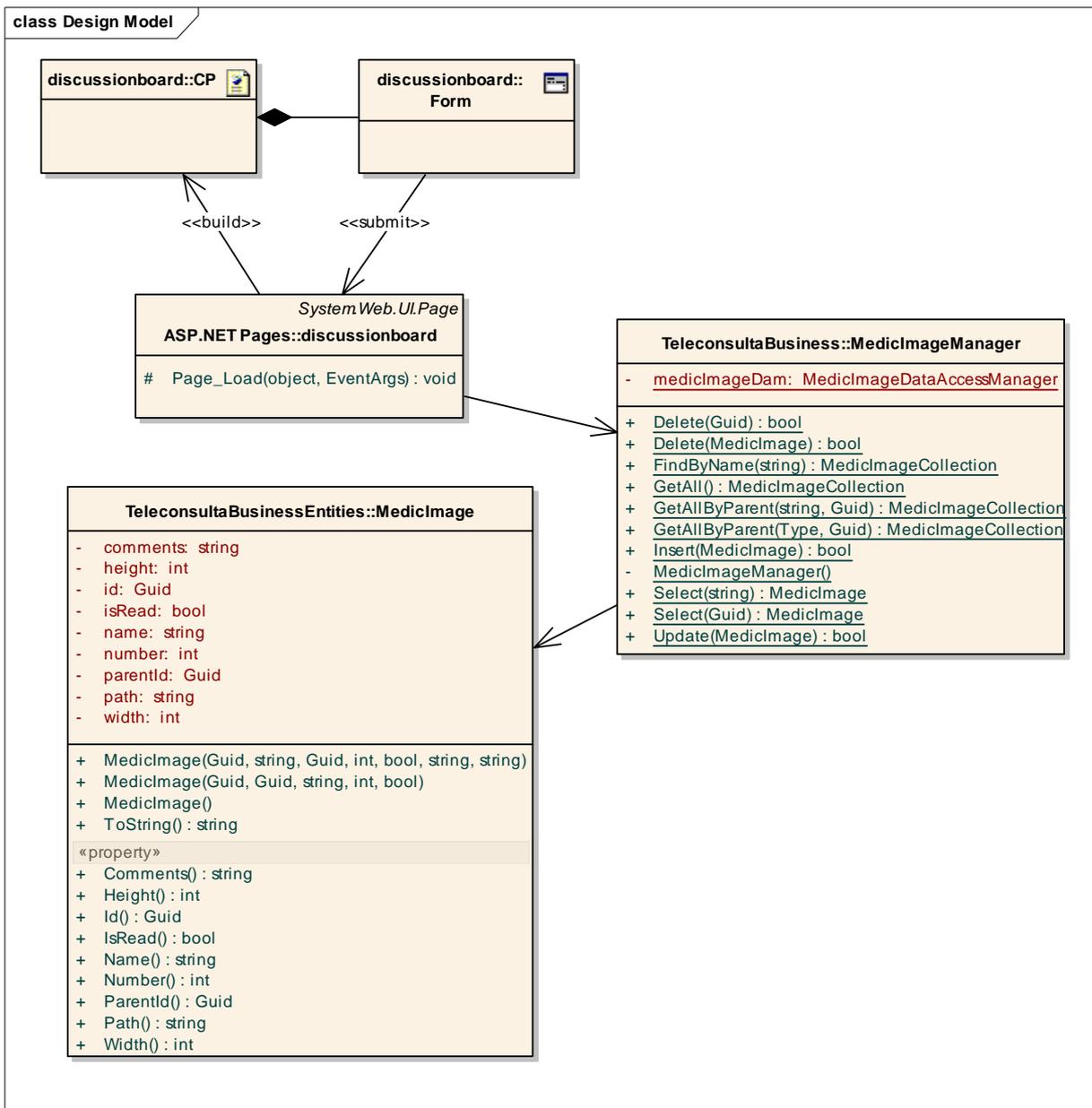
Teleconsultar contacto

DIAGRAMA DE CLASES DEL DISEÑO – CU-18



Hacer zoom sobre la imagen

DIAGRAMA DE CLASES DEL DISEÑO – CU-19



Mover la imagen

DIAGRAMA DE CLASES DEL DISEÑO – CU-20

<Similar al Diagrama de clases del diseño: Hacer zoom sobre la imagen>

Editar perfil

DIAGRAMA DE CLASES DEL DISEÑO – CU-21

