

Universidad de las Ciencias Informáticas

Facultad 7



Título: Análisis y diseño del módulo inscripción admisión del *Sistema de gestión hospitalaria.*

Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas

Autores:

Keila García Nogueira

Pedro Ernesto Salas Oliva

Tutor:

Lic. Maykell Sánchez Romero

Asesor:

Lic. Manuel Avelino Ricardo Hidalgo

Ciudad de La Habana

Junio de 2007

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 30 días del mes de junio del año 2007.

Keila García Nogueira

Pedro Ernesto Salas Oliva

Maykell Sánchez Romero

Firma del Autor

Firma del Autor

Firma del Tutor

“Lo que caracteriza a una inteligencia formada es que puede descansar satisfecha con el grado de precisión que la naturaleza de un asunto permite, y no buscar la exactitud cuando sólo una aproximación de la verdad es posible...”

Aristóteles

AGRADECIMIENTOS

A nuestros padres, que han dado todo de si para que nos convirtiéramos en profesionales y nos han guiado siempre.

A Rudel, por todo el apoyo que nos ha brindado y por no abandonarnos nunca durante el desarrollo de la tesis.

Al personal del departamento de inscripción admisión de los hospitales visitados, que nos brindaron toda la información necesaria para realizar este trabajo.

Agradecemos a todos los que durante toda una vida estudiantil han sembrado un granito de arena en nuestra formación para convertirnos en ingenieros.

Al grupo de las Lokytas, que siempre nos han apoyado en todo.

A Alain, Mainoldis, Ariesky, Soto, Yunaysy, Renier, Yenisel, Lourdes, Yubismel, Rafael, Avelino, Yudarys, entre otros amigos que han contribuido de una forma u otra en la realización del trabajo de diploma.

DEDICATORIA

De Keila:

A mis abuelas, que desde el cielo me han dado todas las fuerzas para seguir adelante.

A mi madre adorada, que merece todas las alegrías que yo pueda dar.

Al padre que me ha apoyado desde mi niñez y que siempre está conmigo en las buenas y las malas.

A mi AMORE, que ha sido la guía en estos tres años, y lo mejor que me ha pasado en la vida.

A mis suegros, que son maravillosos.

A los tíos que siempre me ayudaron a salir adelante: Freddy, Rubén y Chabelo.

A Martha Abreu, por haberme dado la oportunidad de aprender junto a ella.

A las amigas maravillosas que tengo: Yudith, Marita, Yenisel, Nadiezka, Yanelis, Kenia y Yadira.

Al Comandante Fidel, que hizo esta maravillosa universidad para que jóvenes de todo el país pudiéramos estudiar en ella.

De Pedro:

A mi abuelo, cuyo mayor sueño era verme graduado.

A mi madre, que me da todo el cariño del mundo y sin la cual nunca hubiera podido ser quien hoy soy.

A mi abuela, la cual ha sabido forjar mi carácter de la mejor forma posible.

A todos los familiares que me han apoyado y querido, mis tíos, primos, mi padre y tantos otros que el pecho me resulta pequeño.

A mis amistades, las de antes, de ahora y de siempre: Ygraine, Lili, Dayana, Kirenia, Yenni, Alejandro, Julio, Tami y muchos más que siempre me acompañaran, gracias por todo y tanto.

A todos los profesores que de una manera u otra, me han guiado en este camino que es el aprendizaje.

A nuestro comandante, por darnos la posibilidad de alcanzar nuestros sueños.

Resumen

La sección de inscripción admisión, en los centros de atención hospitalaria manipula gran cantidad de información. En nuestro país, el procesamiento de la información de este departamento, generalmente se realiza de forma manual y en los casos que este se automatiza, no existe un software estandarizado que se adapte a las necesidades del sistema nacional de salud y que permita gestionar dichos procesos. Por ello, en el presente trabajo de diploma se propone diseñar un sistema informático para mejorar los procesos de gestión de la información, relacionada con la inscripción admisión en los hospitales cubanos.

Para diseñar el sistema, se utiliza la metodología de desarrollo RUP con el lenguaje UML y la Suite de Rational como herramienta de modelado. Con la elaboración del diseño y el modelado de los patrones, se definen las clases fundamentales que permitirán que el sistema funcione satisfactoriamente, así como los atributos y métodos que deben tener las mismas, para brindarle al desarrollador una idea clara de lo que debe implementar.

Se espera que el sistema sea desarrollado, siguiendo el diseño propuesto. Además, que permita una mayor organización de la información, manejada en el departamento de inscripción admisión y menos gastos de recursos para el hospital y la economía del país.

Palabras claves

Sistema de información hospitalaria, inscripción y admisión, historia clínica.

Tabla de contenidos

Introducción	1
Capítulo 1. Fundamentación teórica	5
1.1 Conceptos básicos relacionados con el problema planteado.....	5
1.2 Sistemas automatizados existentes	6
1.3 Ventajas de los Sistemas de información hospitalaria.....	11
1.4 Tendencias y tecnologías	12
1.5 Lenguajes de programación	20
1.6 Metodologías de desarrollo	22
Capítulo 2. Características del Sistema	33
2.1 Objetivos estratégicos de la organización.....	33
2.2 Flujo actual de los procesos involucrados en el campo de acción.....	36
2.3 Objeto de automatización	38
2.4 Modelo de Negocio	43
2.4.1 Actores del Negocio.....	43
2.4.2 Trabajadores del Negocio.....	44
2.4.3 Diagrama de Casos de Uso del Negocio	45
2.4.4 Diagramas de Actividades	46
2.4.5 Modelo de objetos.....	52
2.5 Especificación de los requisitos de software.....	53
2.6 Diagrama de paquetes.....	57
2.6.1 Actores del sistema.....	58
2.6.2 Diagramas de Casos Uso.....	59
Capítulo 3. Análisis y Diseño	61
3.1 Diagramas de clases de análisis.....	61
3.2 Diagramas de clases de diseño	67
3.3 Diagramas de interacción.....	79
3.4 Descripción de las clases	93

Conclusiones.....	116
Recomendaciones.....	117
Referencias bibliográficas.....	118
Bibliografía	122
Anexos	123
Anexo 1. Modelo de casos de uso	123
4.1.1 Casos de uso por ciclos de desarrollo	123
4.1.2 Listado de casos de uso.....	125
4.1.3 Modelo de Casos de uso Paquete: Reportes	130
4.1.4 Descripciones textuales de los Casos de Uso del Sistema.....	131
Anexo 2. Prototipo no funcional.....	145
Glosario de términos.....	162

Introducción

Los sistemas han existido desde hace mucho tiempo, aunque no tuvieron siempre un alto grado de automatización, ni se estudiaban. Los datos que se generan día a día, se hallan dispersos, se procesan, analizan e interpretan para convertirse en información. A raíz del surgimiento de las computadoras, comenzaron a crearse sistemas sencillos de información, tanto con fines administrativos como financieros.

En la década de los 70's en el campo de la medicina, se perfilaron los primeros sistemas de información médica que posteriormente, dieron lugar a los Sistemas de Información Hospitalaria (SIH), tan indispensables en la actualidad. Su impacto en las instituciones de salud es fuerte, ya que buscan elevar la calidad de la atención del paciente, de los servicios brindados y aplicar la información obtenida a las áreas de la investigación, la clínica, la docencia, la administración, así como disminuir costos y elevar la productividad.

El mayor reto en salud es lograr equidad y calidad en los servicios que se prestan y para ello, debe alcanzarse un equilibrio entre la medicina preventiva y la curativa, con el uso de las tecnologías médicas, incluidas las tecnologías de información y la comunicación en los diferentes niveles de la atención en salud. Cerca del 30% de la información y los servicios que brinda internet se relacionan directa o indirectamente con la medicina.

Implementar sistemas informáticos en nuestros hospitales no está exento de dificultades, desde cuestiones elementales objetivas hasta aspectos subjetivos que entorpecen la gestión. En ocasiones solucionar un problema, genera otros mayores y nuestras expectativas, puestas en el éxito, se van diluyendo a medida que nos tropezamos con las imprevisibles situaciones que constituyen el costo de la modernidad que traemos a las instituciones.

La utilización de sistemas informáticos en nuestros hospitales consiste en articular el vasto continente de diferencias y particularidades de los datos que se manejan. De esta forma se optimizan los esfuerzos del personal de salud, brindando información calificada para tomar decisiones y disminuyendo trámites y tiempo para el paciente.

Aunque se han obtenido grandes progresos en esta empresa, los SIH actuales distan de ser perfectos y la realidad es que se desarrollan numerosas versiones para mejorar continuamente los ya implementados.

Los SIH en Cuba existen, aunque no ha habido la suficiente difusión e intercambio de experiencias. La meta es construir un SIH estandarizado donde el paciente sea el más beneficiado, y los profesionales de la salud encuentren en estos sistemas un recurso idóneo, amigable y flexible que responda a las necesidades de información de la institución hospitalaria o de salud.

Un SIH está compuesto por varios subsistemas, dentro de los cuales resalta el clínico, compuesto a su vez por los subsistemas clínicos departamentales y clínicos de apoyo. Los procesos de gestión de la inscripción admisión hospitalarias constituyen un subsistema dentro del SIH.

En Cuba, los procesos de gestión de la inscripción admisión hospitalarias no están automatizados, la mayoría se realizan de forma manual, o utilizando aplicaciones que no constituyen un software estándar para todo el país, lo que trae como consecuencia la demora, pérdida e inconsistencia de la información, además de hacer más engorroso el trabajo a los empleados del departamento. Los datos de los pacientes son guardados en formularios existentes en formato papel y llevados en algunos casos a las computadoras que generalmente no cumplen con los requisitos necesarios para el almacenamiento.

Es debido a esto, que en algunos casos se cometen errores a la hora del llenado de planillas y de la obtención de algunos datos, tales como en las estadísticas relacionadas con el tema, ya que es muy complicado extraer esa información y realizar los cálculos buscando en cada uno de los registros de los pacientes.

El problema del almacenamiento de los datos, se agrava debido al gran torrente de archivos que deben ser generados y guardados de manera organizada. Al realizarse este proceso de manera física, los locales se tornan grandes y los estantes se tornan inmensos. Esto trae consigo la lentitud en las solicitudes de Historias Clínicas (HC), entre otros documentos que se generan, como parte del proceso que se lleva a cabo en este departamento del hospital. Puede suceder, que se creen HC duplicadas por el desconocimiento de la existencia de las ya generadas, debido al gran flujo de información que no se encuentra amparado con ninguna tecnología.

El proceso manual tiene muchas desventajas para los trabajadores de salud pública, pues impide la tenencia de registros actualizados a causa de la demora del llenado de datos y a los errores que se pueden cometer durante este procedimiento. De igual manera, la cantidad de personas atendidas diariamente es reducida, en ocasiones, se inscriben pacientes en horario no laboral del departamento, provocando que se corra el riesgo de generar un gran volumen de información repetida, lo cual conlleva a

realizar despachos entre las diferentes oficinas y otros trámites engorrosos. La redundancia de información trae consigo que en la última HC, elaborada a un paciente, no existan sus datos anteriores debido a que se creó una nueva, esto impide el seguimiento de la evolución del paciente en relación a alguna patología, incluyendo la pérdida de tiempo al llenar nuevamente los mismos formularios; así mismo surgen problemas en la ubicación de su cama en el hospital y el consiguiente malgasto de materiales.

Por otra parte, existen registros que se generan para control y calidad del trabajo del departamento. Estos son llevados por los trabajadores de la sección y presentan inconsistencias y atrasos debido a la deficiencia del proceso manual.

Además, la información se desactualiza, debido a que en algunos casos las persona encargadas de transmitir estos partes deben trasladarse con documentos de un local a otro. Esto provoca demoras en el proceso y se plantea que es inminente la solución a este problema.

Después de realizar un análisis de la situación problemática, se define el siguiente **problema científico**:

¿Cómo mejorar los procesos de gestión de la información, relacionados con la inscripción admisión en los hospitales cubanos, mediante un sistema informático?

Este problema se enmarca en el **objeto de estudio** concerniente a los procesos de gestión de la información hospitalaria.

Basado en esa idea se define el **objetivo general** del presente trabajo:

Diseñar un sistema informático para mejorar los procesos de gestión de la información relacionada con la inscripción admisión en los hospitales cubanos.

Los **objetivos específicos** que se proponen para alcanzar el objetivo general son los siguientes:

- Realizar un estudio de los procesos de gestión de la información, relacionada con la inscripción admisión en los hospitales cubanos, que permita modelar el negocio.
- Investigar sobre los sistemas de gestión hospitalaria, que existen en Cuba y el mundo, que contengan el módulo estudiado.
- Determinar las funcionalidades que debe tener el sistema permitiendo definir un prototipo no funcional del mismo.

- Realizar un estudio detallado de los patrones de diseño existentes que puedan ser aplicados al sistema.
- Modelar el análisis y diseño del subsistema de inscripción admisión, utilizando patrones GRASP y GOF.

De manera que el **campo de acción** que se abarca está relacionado con los procesos de gestión de la información de la inscripción admisión en los hospitales cubanos.

El trabajo de diploma está compuesto por tres capítulos:

Capítulo 1. Fundamentación teórica. En el se estudian los conceptos y sistemas relacionados con la gestión hospitalaria, específicamente enfocados a los procesos de inscripción admisión; así como las tecnologías más utilizadas para la elaboración de los SIH.

Capítulo 2. Características del sistema. Se abordan los objetivos estratégicos de los hospitales, además de los procesos que ocurren en estos. Se modela el negocio y a partir de las funcionalidades identificadas, se realiza la propuesta del sistema.

Capítulo 3. Análisis y diseño. Teniendo en cuenta la metodología RUP, se elaboran un grupo de artefactos que pertenecen a la disciplina. Se describen las clases definidas en el diseño.

Así como de una serie de anexos que complementan la información presentada en dichos capítulos.

Capítulo 1. Fundamentación teórica

Existe un grupo de conceptos que pueden resultar desconocidos y que están estrechamente relacionados con los procesos que forman parte del campo de acción. Así como, algunos sistemas de gestión hospitalaria desarrollados a nivel nacional e internacional y que su estudio es de vital importancia, para ubicarnos en la necesidad concreta del diseño de un SIH. Al mismo tiempo, existe una gama de tendencias, tecnologías, lenguajes de programación y metodologías, muy utilizadas en la actualidad para la elaboración de estos sistemas y que evidencian las técnicas que se emplean para fabricar un producto, especificando cómo y con qué herramientas diseñarlo.

1.1 Conceptos básicos relacionados con el problema planteado

El proceso de inscripción es la actividad inicial en todo hospital. Cada paciente tiene que registrarse antes de conseguir cualquier consulta, tratamiento, o investigación. Este proceso involucra, información general y demográfica sobre el mismo; se le asigna un único número de inscripción central y le es entregada una tarjeta de identificación.

Existe un proceso estrechamente relacionado con la hospitalización de los pacientes al que generalmente se llama Admisión. Esta sección es la encargada de almacenar toda la información referente a los pacientes, durante su estancia en el hospital; así como manejar y mantener actualizados todos estos datos.

Como resultado de la inscripción y admisión hospitalaria, se elabora la historia clínica (HC). Es un documento que surge en el contacto entre el equipo de salud y los pacientes. Según el concepto de Castro y Gamez en el artículo Historia Clínica, esta debe tener las siguientes características:

La HC debe ser única, integrada y acumulativa para cada paciente en el hospital, debiendo existir un sistema eficaz de recuperación de la información clínica. Es creada en el proceso de inscripción en un hospital y su principal función es la asistencial, ya que permite la atención continuada a los pacientes [1].

Para lograr que estos procesos se efectúen con la mayor rapidez y calidad posible se propone la creación de un sistema de información o gestión hospitalario que está orientado a satisfacer las necesidades de generación de información, para almacenar, procesar y reinterpretar datos médico-administrativos de cualquier institución hospitalaria [2].

La idea de usar computadoras en hospitales, surgió debido a la gran cantidad de información que permiten gestionar. Persigue objetivos como la sistematización de la cuenta del paciente, con el propósito de manejar todos los datos concernientes al mismo a través de la HC. En otros casos, con la finalidad de obtener cálculos para estadísticas hospitalarias. Por lo que el uso de las mismas debe proporcionar un mejor funcionamiento, de los servicios dirigidos a los pacientes; además debe permitir el manejo de inventarios y sistemas financieros.

En nuestros días, los SIH encaminan sus esfuerzos a la descentralización. Cada departamento o sección hospitalaria, tiene su propia computadora y están conectados a través de una red local; esto permite: dispersar procesos para dedicar el tiempo del procesador únicamente a funciones comunes, obtener mejor tiempo de respuesta, compartir datos intra/extra departamentales y de esta forma se envía información únicamente a quien la necesite.

Se cuenta con poderosos SIH, que soportan:

- Administración integral y apoyo a la toma de decisiones
- Transacciones de intercambio de información en la atención al paciente
- Solicitud y suministro de medicamentos
- Apoyos a la elaboración de diagnósticos.

La utilización de esta tecnología elabora más del 80% de las tareas operativas, permitiendo optimizar funciones y dedicar más tiempo a la atención del paciente y la investigación.

Los SIH generalmente están divididos en módulos que se encargan de la gestión de los datos de las diferentes secciones departamentales de un hospital, tales como: inscripción admisión, bloque quirúrgico, archivos, laboratorio clínico, turnos, farmacia, consulta externa, banco de sangre, entre otros.

Como parte de la investigación en curso se identifican varios SIH, estos gestionan los procesos del módulo de inscripción admisión.

1.2 Sistemas automatizados existentes

SIVSA es una empresa especializada en el desarrollo e implantación de software para el sector de la salud que cuenta con más de 15 años de experiencia. Uno de los productos desarrollados por este

consorcio es el Hosix-V que está diseñado utilizando una *arquitectura Cliente – Servidor* accesible desde clientes Web, lo que permite ser ejecutado desde cualquier navegador ya sea vía Intranet o Internet.

Abarca todas las áreas de actividad de un hospital y pretende, a través de una utilización fácil, rentabilizar los recursos existentes para organizar el trabajo desarrollado diariamente. Está compuesto por un grupo de módulos orientados a la gestión específica de cada servicio o departamento, cubriendo los aspectos más importantes de una unidad de salud. Este sistema cuenta con una *base de datos* de usuarios centralizada, en la cual se encuentran todos los ciudadanos y permite una gestión integral de la información y su permanente actualización. Ha sido desarrollado utilizando como herramientas y lenguaje de programación Microsoft Visual Basic para aplicaciones, además requiere una aplicación Microsoft para su uso, su precio oscila entre los \$500,001 a 2, 000,000 de dólares.

Cuenta con módulos como el de historia clínica, gestión de citas, telemedicina, seguridad, urgencias y hospitalización, entre otros. Estos dos últimos son los equivalentes a lo que sería en el sistema de salud pública cubano los departamentos de inscripción admisión hospitalaria.

El módulo de Urgencias permite controlar y gestionar la entrada de pacientes en esta área. Su organización autoriza al sistema para registrar e informar de cualquier movimiento del paciente desde su llegada a urgencias hasta su salida, aunque se produzca esta por hospitalización, mejoría o traslado. Ofrece una gestión completa sobre todas aquellas actividades relacionadas con el paciente facilitando la optimización de recursos del hospital y la atención al mismo.

El módulo de hospitalización permite controlar los procesos correspondientes a los ingresos de los pacientes. Su organización autoriza al sistema a registrar e informar, de cualquier movimiento desde la llegada del paciente hasta su salida, así como de los preingresos existentes. Ofrece una gestión completa de las camas hospitalarias. Permite ver el nivel de ocupación, las camas libres por totales y por servicio, los traslados, facilitando con todo ello la optimización de recursos del hospital.

El módulo de codificación permite realizar todo el proceso de captura de datos referidos a la codificación de diagnósticos, procedimientos y complicaciones de todos los episodios de los pacientes (Urgencias, Ingresos, etc.). El sistema de codificación se basa en la atribución de los diagnósticos y procedimientos a los diferentes episodios de los pacientes [3].

El proyecto Care2x tiene un equipo de desarrollo que cuenta con más de 100 miembros de diferentes países con múltiples destrezas y conocimientos. El sistema Care2x integra datos, funciones y flujo de tareas en un entorno de cuidados de la salud. Está conformado por cuatro componentes principales que a su vez pueden funcionar de forma independiente.

- HIS - Sistema de Información Hospitalaria/servicios de la salud
- PM - Administración del ejercicio médico
- CDS - *Servidor* Central de Datos
- HXP - Protocolo de intercambio de datos de la salud

Care2x soluciona el problema del software múltiple, la redundancia e incompatibilidad de datos, los que están centralizados y se comparten a lo largo de toda la red mediante contraseñas. Los programas externos pueden acceder a los datos al utilizar el sistema SQL estándar para consultar la información. Esta implementado en PHP 4.0.4 pl1 - 4.3.4, utilizando como *gestor de Base de Datos* MySQL 3.2x - 4.0.x y corre sobre un *servidor* Web Apache 1.X.

Care2x soluciona el problema de la dependencia de plataforma, acelera las mejoras y la adaptación de módulos de programas existentes. Los costos de desarrollo y el tiempo requeridos se reducen sustancialmente. No es necesario contratar a varias compañías de software diferentes. No existe desconexión del sistema durante las actualizaciones de los módulos o durante la instalación de extensiones de los módulos. Los usuarios probablemente no se darán cuenta de que se han actualizado sus módulos de trabajo en caliente.

Cuenta con un módulo de admisión en el cual existen funcionalidades como la inscripción de un paciente nuevo, búsqueda de paciente ingresado, búsqueda de casos archivados. Permite el etiquetado con los datos del paciente y la impresión de códigos de barras. Existe además un módulo Persona, que se encarga de registrar los datos básicos de los pacientes de manera que almacena y procesa toda esta información de una manera rápida y consistente [4].

C-DAC es una corporación productora de software que radica en la India y cuenta con una gama de productos encaminados a la automatización de procesos no sólo del área de salud. Uno de los software que comercializa esta empresa es el C-DAC's Sushrut, que es un sistema de información hospitalario desarrollado con el objetivo de aerodinamizar el flujo del tratamiento de un paciente en el hospital,

permitiendo que los doctores y el otro personal se realicen a su capacidad máxima, de una forma optimizada y eficiente.

Este SIH implementado por C-DAC contiene varios módulos los cuales aseguran los beneficios obtenidos a través de la tecnología y para la salud. El sistema utiliza una red de computadoras para reunir procesos, y manejar toda la información administrativa del hospital para satisfacer los requisitos funcionales de los usuarios. Este sistema ha sido desarrollado utilizando una arquitectura cliente servidor, además puede ser utilizado en múltiples plataformas.

Incorpora un sistema de información clínica computarizada e integrada para mejorar la administración del hospital y el cuidado de la salud de los pacientes. Además provee un archivo médico electrónico muy preciso de los pacientes, incluyendo un almacén de datos (*data warehouse*) del cual los archivos pueden ser utilizados para requerimientos estadísticos y para la investigación.

El módulo registro del paciente es el primer proceso que se gestiona en el sistema de información total del hospital. El registro contiene información general y demográfica sobre el paciente, quien cuenta con un número de registro central único y además con una tarjeta de identificación paciente, conocida como tarjeta del registro.

El módulo hospitalización comienza cuando se asigna al paciente una cama en la sala, además se ocupa del tratamiento completo del paciente durante su estancia en el hospital [5].

El Proyecto PHRplus (Partners for Health Reform plus), con la asistencia del USAID (Agencia de Estados Unidos para el Desarrollo Internacional, por sus siglas en inglés) son los desarrolladores del Sistema Integrado de Gestión hospitalaria Galenhos (R), que ha sido diseñado con el propósito de apoyar a los establecimientos de salud en el correcto registro de información, clínica o administrativa, y la generación de información gerencial. Como sistema hospitalario para la gestión de su información, el Galenhos cuenta con ventajas apreciables, como:

- Información estandarizada
- Bases de datos consolidadas y exportables
- Generación de reportes clínicos
- Altos estándares de seguridad informática

- Diseño modular.

El desarrollo inicial de Galenhos se enfoca en el propósito de hacer más eficiente la gestión de cinco procesos operativos críticos de un hospital: consulta externa, facturación, hospitalización, emergencia y archivo clínico.

En el caso del módulo de Hospitalización se considera la filiación de pacientes y registra diagnósticos y procedimientos. Además, cuenta con un submódulo para la administración de camas hospitalarias.

El módulo de Emergencia al igual que consulta externa y hospitalización, cuenta con filiación y registro de diagnósticos y procedimientos. También permite la administración de las camas de observación de emergencia [6].

Otro sistema identificado es el MedFile 5.x que es un software de fácil uso, efectivo y con un precio accesible, diseñado para satisfacer las necesidades de archivo de HC y manejo de turnos de un consultorio o institución médica en el que se desempeñen uno o varios profesionales (hasta 200 en la versión multiusuario).

MedFile 5.x es una aplicación de escritorio que permite crear y mantener HC electrónicas de sus pacientes en un formato especial de *base de datos*, asignar turnos para la consulta con agenda personalizada para cada médico, y emitir prescripciones y órdenes médicas en forma altamente personalizadas y configurables [7].

El sistema de información hospitalario HIS que propone la Coordinación de Informática Médica de la Universidad Autónoma de Guadalajara (CIM UAG), se robustece en su nueva versión 4.0, la cual provee los avances en los procesos de información hospitalaria. El sistema corre en *sistemas operativos* Windows 95, 98, 2000, ME, XP y ha sido desarrollado utilizando el lenguaje de programación Delphi 7, con una *base de datos* SQL Server 7.

El HIS se divide en los siguientes sistemas:

- MIS (Sistema de Información Administrativo)
- NIS (Sistema de Información de Enfermería)
- LIS (Sistema de Información de Laboratorios)
- CIS (Sistema de Información Clínica).

El MIS o Sistema de Información Administrativo, se adapta a las necesidades de la sección de inscripción admisión en los hospitales. Permite agilizar y controlar las funciones operativas, además de automatizar la gestión de la información, en los diferentes módulos administrativos que apoyan a la organización, optimizando la productividad y el uso de recursos para apoyar la toma de decisiones.

Cuenta con módulos tales como Admisión y Ambulatorios. El primero de estos se encarga del ingreso de pacientes, censos hospitalarios y los movimientos cuarto-cama, mientras que el de Ambulatorios maneja los datos al registrar un paciente, censos realizados e incluye las altas o egresos, además de otras funcionalidades que se relacionan con cobros y facturas que deben pagarse por recibir estos servicios médicos. Dentro del Sistema de Información Clínica se elabora el expediente clínico que tiene notas relacionadas del paciente, así como su HC [8].

En nuestro país, se han desarrollado también sistemas de gestión hospitalarios, tal es el caso de la empresa Softel que brinda soluciones para la salud y que radica en la Universidad de las Ciencias Informáticas, este consorcio cuenta con sucursales en todo el país. De conjunto con la universidad se elaboró el Sistema de información hospitalaria SIH, que es una aplicación de escritorio que cuenta con varios módulos que gestionan la mayoría de los procesos que se desarrollan en un hospital, tales como la inscripción de un paciente, los ingresos, la creación de historias clínicas, así como los diferentes datos que se manejan al operar a un paciente o practicarle análisis en los laboratorios.

El módulo de inscripción admisión hospitalaria permite registrar o modificar los datos personales de un paciente, además brinda la opción de hacer búsquedas de historias clínicas existentes para realizar ingresos en sala. Permite además reubicar a un paciente que ha sido ingresado, conocer la disponibilidad de las camas, realizar egresos, entre otras funcionalidades.

Este sistema fue desarrollado utilizando el lenguaje de programación Visual Basic, con un *gestor de base de datos* Microsoft SQL Server [9].

1.3 Ventajas de los Sistemas de información hospitalaria

Las ventajas básicas de la implementación de los sistemas de información hospitalaria, se evidencian al mejorar la calidad en los servicios de salud. Estos proporcionan un mejor manejo de: información clínica, información administrativa, vigilancia epidemiológica, información científica, búsqueda de documentos. Además, para su aplicación efectiva requieren de: el compromiso, la responsabilidad, instalación de la

tecnología adecuada, entrenamiento del personal, actualización de la tecnología (software) y del mantenimiento (hardware). También requiere, basarse en las metas de la institución, contemplar la dinámica local y ser validado y compartido por los distintos actores.

El SIH se aplica en cinco campos:

- Información clínica
- Información administrativa
- Información científica
- Vigilancia epidemiológica
- Búsqueda de documentos.

La información debe ser íntegra, confidencial y disponible. No debe plantearse la informatización en forma aislada sino abierta, completa y universal, cubriendo las necesidades de obtención de datos, de generación de informes y de planeamiento de todo el sistema de salud del país, teniendo siempre presente la noción de conectividad, que implica la pertenencia a una red mas amplia.

Es obligatorio desde el punto de vista ético optimizar la calidad de la información en salud mediante la informática médica y utilizar esta información en beneficio de la Salud Pública.

1.4 Tendencias y tecnologías¹

En la actualidad, durante el proceso de desarrollo de un software para la salud, se toman en cuenta una serie de tendencias y tecnologías informáticas, de manera que se identifican un grupo de ellas para el desarrollo de la investigación.

Aplicaciones Web

En estos momentos, el uso de las aplicaciones Web está muy generalizado. Estas se han difundido con gran amplitud debido a que, una sola computadora (el servidor), debe tener mayor cantidad de recursos en cuanto a hardware. Las PC clientes pueden contar con un navegador ligero que se conecta al servidor

¹ Para la elaboración de este epígrafe se ha consultado: Wikipedia. Disponible en: <http://wikipedia.org> entre otras fuentes incluidas en las referencias bibliográficas.

de manera que las funciones de procesamiento se realicen del lado del mismo. Otra ventaja esta dada por la habilidad para actualizar y mantener aplicaciones Web, sin distribuir e instalar software en miles de potenciales clientes.

Desarrollar en WEB permite usar los elementos estándar de interfaz *HTML*. La mayoría de los usuarios potenciales tienen familiaridad en navegar por Internet y están acostumbrados con la apariencia de las páginas Web. También se cuenta con los beneficios de una instalación centralizada en un solo lugar, así como las actualizaciones sin tener que distribuir programa alguno. En vez de tener que repartir la aplicación a cientos ó miles de máquinas, únicamente colocamos el nuevo software en un solo sitio y de inmediato todos los usuarios acceden a éste.

Si en la aplicación a desarrollar hay usuarios remotos, es más fácil atravesar *firewalls* corporativos sobre *HTTP*, que sobre cualquier otro protocolo. Algunas aplicaciones requieren bastante poder de procesamiento. En lugar de tener que actualizar individualmente cada estación de trabajo, es mejor aumentar el poder de procesamiento en el *servidor* central. Esta será una solución mejor en costos que actualizar cientos o miles de estaciones de trabajo.

En WEB se tiene un control limitado sobre el lugar exacto donde aparecerán los elementos en la pantalla. El *HTML* estándar no permite el posicionamiento absoluto de los elementos. Esto podría hacer el diseño menos elegante que en una aplicación Windows.

El desempeño puede ser más bajo en una aplicación Web debido a que se están enviando los datos y el diseño de la pantalla cambia cada vez que se pide un documento *HTML*. En una aplicación Windows, la interfaz es armada localmente sólo una vez y únicamente los datos son traídos a través de una conexión de red rápida. De esta manera, no sería muy conveniente tener un aplicativo de atención en ventanilla de gran volumen de transacciones bajo WEB.

Con *HTML* no es muy fácil ir fuera de los límites del navegador. Esto significa que cualquier integración con otros productos está prácticamente desechada. Con el Microsoft Internet Explorer, es mucho más fácil integrar que con otros navegadores, pero aun así requiere demasiada programación. Hacerlo de esta manera, es casi como crear una aplicación Windows.

Internet

Otro de los factores que facilita la elección de desarrollar aplicaciones Web es la Internet que se puede definir como una red global de redes de ordenadores cuya finalidad es permitir el intercambio libre de información entre todos sus usuarios, utilizando un conjunto de protocolos, el más destacado, el *TCP/IP*. Una de las ventajas de Internet es que posibilita la conexión con todo tipo de ordenadores, desde los personales, hasta los más grandes que ocupan habitaciones enteras [10].

Navegadores Web

La difusión de los distintos sistemas a través de Internet se logra mediante los navegadores Web. Un navegador Web o *browser* es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en *HTML*, desde *servidores* Web de todo el mundo a través de Internet. La funcionalidad básica de un navegador Web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos. Existen diferentes navegadores Web tales como Mozilla, Windows Internet Explorer, Firefox, Netscape, Opera, Konqueror, entre otros [11].

Sistemas distribuidos

En lo que respecta a la arquitectura se observa la tendencia a la utilización de sistemas distribuidos, los cuales se definen como una colección de computadores autónomos conectados por una red, y con el software distribuido adecuado para que el sistema sea visto por los usuarios como una única entidad capaz de proporcionar facilidades de computación. Los sistemas distribuidos se implementan en diversas plataformas hardware, desde unas pocas estaciones de trabajo conectadas por una red de área local, hasta Internet [12].

Modelo Cliente/Servidor (dos capas)

El modelo cliente-servidor de un sistema distribuido es el modelo más conocido y ampliamente adoptado en la actualidad. Hay un conjunto de procesos servidores, cada uno actuando como un gestor de recursos para una colección de recursos de un tipo, y una colección de procesos clientes, cada uno llevando a cabo una tarea que requiere acceso a algunos recursos hardware y software compartidos. Los gestores de

recursos a su vez podrían necesitar acceder a recursos compartidos manejados por otros procesos, así que algunos procesos son ambos clientes y servidores. En el modelo, cliente-servidor, todos los recursos compartidos son mantenidos y manejados por los procesos servidores. Los procesos clientes realizan peticiones a los servidores cuando necesitan acceder a algún recurso. Si la petición es válida, entonces el *servidor* lleva a cabo la acción requerida y envía una respuesta al proceso cliente [13].

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El *servidor* presenta a todos sus clientes una interfaz única y bien definida
- El cliente no necesita conocer la lógica del *servidor*, sólo su interfaz externa
- El cliente no depende de la ubicación física del *servidor*, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo
- Los cambios en el *servidor* implican pocos o ningún cambio en el cliente [14].

Modelo cliente servidor (tres capas)

La arquitectura de tres capas se refiere a un diseño reciente que introduce una capa intermedia al proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final (el cliente). La arquitectura basada en WEB transforma la interfaz de búsqueda existente (el explorador de WEB), en la interfaz del usuario final.

La tercera capa generalmente es el sistema de administración de la *base de datos*. Es decir donde los datos requeridos por la capa intermedia son almacenados. La tercera capa se localiza en un *servidor* separado conocido como el *servidor de base de datos*. La parte funcional de la arquitectura de tres capas generalmente es conocida como la capa intermedia o el *servidor de aplicaciones*. En éste ocurren la mayoría de los procesos [15].

Servicios Web (Web services)

En el caso del manejo de datos se usa ampliamente lo que se conoce como Servicio Web (en inglés *Web service*) que no es más que una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar

datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

Ventajas de los servicios Web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en *HTTP*, los servicios Web pueden aprovecharse de los sistemas de seguridad *firewall* sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

Los servicios Web hacen uso de distintas tecnologías como son XML, SOAP, XSL.

XML

XML, sigla en inglés de eXtensible Markup Language («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium (W3C)*. Es una simplificación y adaptación del *SGML* y permite definir la gramática de lenguajes específicos (de la misma manera que *HTML* es a su vez un lenguaje definido por *SGML*). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, etc.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya

que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Ventajas de XML

- **Comunicación de datos.** Si la información se transfiere en XML cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
- **Migración de datos.** Si trabajamos en formato XML sería muy sencillo mover datos de una *base de datos* a otra.
- **Aplicaciones Web.** Con XML hay una sola aplicación que maneja los datos y para cada navegador podemos tener una hoja de estilo o similar para aplicarle el estilo adecuado.

SOAP

SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar creado por Microsoft, IBM y otros, está actualmente bajo el auspicio de la W3C que define cómo dos objetos, en diferentes procesos, pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

A diferencia de *DCOM* y *CORBA*, que son binarios, SOAP usa el código fuente en XML. Esto es una ventaja ya que facilita su lectura por parte de humanos, pero también es un inconveniente dado que los mensajes resultantes son más largos. El intercambio de mensajes se realiza mediante tecnología de componentes. El término *Object* en el nombre significa que se adhiere al paradigma de la programación orientada a objetos.

SOAP es un marco extensible y descentralizado que permite trabajar sobre múltiples pilas de protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.

XSL

XSL (siglas de Extensible Stylesheet Language, expresión inglesa traducible como "lenguaje extensible de *hojas de estilo*") es una familia de lenguajes basados en el estándar XML que permite describir cómo la

información contenida en un documento XML cualquiera debe ser transformada o formateada para su presentación en un medio específico.

XSLT

XSLT o XSL Transformaciones es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Las *hojas de estilo* XSLT realizan la transformación del documento utilizando una o varias reglas de plantilla: unidas al documento fuente a transformar, esas reglas de plantilla alimentan a un procesador de XSLT, el cual realiza las transformaciones deseadas colocando el resultado en un archivo de salida o, como en el caso de una página Web, directamente en un dispositivo de presentación, como el monitor de un usuario.

Actualmente, XSLT es muy usado en la edición Web, generando páginas *HTML* o *XHTML*. La unión de XML y XSLT permite separar contenido y presentación, aumentando así la productividad.

Apache

En lo que respecta a las tecnologías por parte del *servidor* sobresale Apache, software libre y de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo *HTTP/1.1*. Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

Apache es un *servidor* de red para el protocolo *HTTP*, elegido para poder funcionar como un proceso independiente, sin que eso solicite el apoyo de otras aplicaciones o directamente del usuario. Para poder hacer esto, una vez que se haya iniciado, crea unos subprocesos (que normalmente vienen llamados "*children processes*") para poder gestionar las solicitudes: estos, sin embargo, no podrán nunca interferir con el proceso mayor.

PostgreSQL²

² Para la realización de este sub epígrafe se ha consultado: Worsley, John; Drake, Joshua: *PostgreSQL*. Disponible en: <http://www.sobl.org/traduccion/practical-postgres/node19.html>

Dentro de los gestores de bases de datos existentes, se menciona como uno de los más representativos a PostgreSQL, un motor de *base de datos* que es *servidor de base de datos* relacional libre, liberado bajo la licencia BSD. PostgreSQL está considerado como la *base de datos* de código abierto más avanzada del mundo, proporcionando un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como *DB2* u *Oracle*. Ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema: *Clases*, *Herencia*, *Tipos*, *Funciones*. Cuenta además con características que aportan potencia y flexibilidad adicional:

- *Restricciones (Constraints)*
- *Disparadores (triggers)*
- *Reglas (rules)*
- Integridad transaccional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Soporta operadores, *funciones* métodos de acceso y tipos de datos definidos por el usuario. Soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la *base de datos*.

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de *Oracle*, PL/SQL. La característica de PostgreSQL conocida como *Write Ahead Logging* incrementa la dependencia de la *base de datos* al registro de cambios antes de que estos sean escritos en la misma. Esto garantiza que en el hipotético caso de no pueda ser accedido el servidor de BD, existirá un registro de las transacciones a partir del cual podremos restaurar la *base de datos*. Esto puede ser enormemente beneficioso en el caso de no accesibilidad, ya que cualquier transacción que no se haya registrado en la *base de datos* puede ser recuperada usando el dato que fue previamente registrado. Una vez el sistema ha quedado restaurado, un usuario puede continuar trabajando desde el punto en que lo dejó cuando el servidor dejó de estar disponible.

Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

1.5 Lenguajes de programación³

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente.

C#

C# es un lenguaje de programación *orientado a objetos* desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .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 lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

C#, como parte de la plataforma .NET, está normalizado por ECMA desde diciembre de 2001. El 7 de noviembre de 2005 acabó la beta y salió la versión 2.0 del lenguaje que incluye mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. Ya existe la versión 3.0 de C# en fase de beta destacando los tipos implícitos y el LINQ (Language Integrated Query).

Aunque C# forma parte de la plataforma .NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Aunque aún no existen, es posible implementar *compiladores* que no generen programas para dicha plataforma, sino para una plataforma diferente como Win32 o UNIX.

En la actualidad existen los siguientes *compiladores* para el lenguaje C#:

- Microsoft .NET *framework* SDK incluye un compilador de C#.
- Microsoft Visual C#, IDE por excelencia de este lenguaje, versión 2002, 2003 y 2005.
- #develop, es un IDE libre para C# bajo *licencia LGPL*, muy similar a Microsoft Visual C#.

³ Para la elaboración de este epígrafe se ha consultado: Wikipedia. Disponible en: <http://wikipedia.org> entre otras fuentes incluidas en las referencias bibliográficas.

- Mono, es una implementación GPL de todo el entorno .NET desarrollado por Novell. Como parte de esta implementación se incluye un compilador de C#.
- Delphi 2006, de Borland Software Corporation.
- dotGNU Portable.NET, de la Free Software Foundation.

ASP.NET

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios Web domésticos, aplicaciones Web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET es la plataforma unificada de desarrollo Web que proporciona a los desarrolladores los servicios necesarios para crear aplicaciones Web empresariales.

Java Script

Java Script 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, Java Script 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. La mayoría de los navegadores interpretan el código Java Script integrado dentro de las páginas Web.

AJAX

Acrónimo para **Asynchronous Java Script + XML**, en realidad no es una tecnología, sino la unión de varias tecnologías que juntas se encargan de cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al *servidor* buscando los datos que son usados para actualizarla sólo re-renderizando la misma y mostrando u ocultando porciones de ella. AJAX incorpora:

- Presentación basada en estándares usando XHTML y CSS
- Exhibición e interacción dinámicas usando el Document Object Model
- Intercambio y manipulación de datos usando XML and XSLT
- Recuperación de datos asincrónica usando XMLHttpRequest

- Java Script poniendo todo junto. [16]

1.6 Metodologías de desarrollo⁴

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Dentro de las clasificaciones existentes para las mismas sobresalen dos: las metodologías tradicionales y las ágiles.

Las metodologías tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, son generalmente aplicadas a grandes proyectos e indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Por otra parte se encuentran las llamadas metodologías ágiles que se centran más en la obtención del sistema sin importar cuán documentado esté el mismo, generalmente propone que los involucrados participen el proceso de desarrollo para ir mejorando las funcionalidades del sistema en producción, así como el trabajo en parejas [17].

Dentro de las metodologías tradicionales y que puede ser visto como una metodología ágil además, pues es tan configurable como deseemos, se encuentra el RUP haciendo uso del lenguaje UML.

RUP

El Proceso Racional Unificado (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, 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. Sus principales características son:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software

⁴ Para la elaboración de este epígrafe se ha consultado: JOSÉ H. CANÓS, PATRICIO LETELIER Y M^º CARMEN PENADÉS (enero 2007): *Todo Ágil*. Disponible en <http://www.willydev.net/descargas/prev/TodoAgil.Pdf> entre otras fuentes incluidas en las referencias bibliográficas.

- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

El 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) [18].

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

- **Inicio:** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen
- **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.

UML

UML es un lenguaje usado para especificar, visualizar y documentar los componentes de un sistema en desarrollo *orientado a objetos*. 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 el *OMG* (Object Management Group). 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.

- Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).
- UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

En el año 2001 se reunieron un grupo de impulsores de lo que hoy son las metodologías ágiles para debatir los puntos que tenía en común y el resultado fue el Manifiesto Ágil. Hay diversos métodos ágiles que recogen estas ideas como: eXtreme Programming(XP),Cristal Methods ,SCRUM ,etc.

eXtreme Programming (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuado para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Las características fundamentales de XP son:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias** continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- **Programación por parejas:** se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- **Frecuente interacción del equipo de programación con el cliente o usuario.** Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores antes de añadir nueva funcionalidad.** Hacer entregas frecuentes.
- **Refactorización del código,** es decir, reescribir ciertas partes del código para aumentar su legibilidad pero sin modificar su comportamiento
- **Propiedad del código compartida:** en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto.
- **Simplicidad en el código:** Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores [19].

SCRUM

Otra de las metodologías muy difundidas dentro de las ligeras o ágiles es Scrum, expuesta por Hirotaka Takeuchi e Ikujiro Nonaka y que define un marco para la gestión de proyectos, que se ha utilizado con

éxito durante los últimos 10 años. Esta es, después de XP, la metodología ágil mejor conocida y la que otros métodos ágiles recomiendan como complemento.

Además no está concebido como método independiente, sino que se promueve como complemento de otras metodologías, incluyendo XP, MSF o RUP. Como método, enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas; de allí su deliberada insuficiencia y su complementariedad. Esta metodología se define como un proceso de gestión y control que implementa técnicas de control de procesos; se lo puede considerar un conjunto de patrones organizacionales.

Scrum está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración [20].

Crystal Methodologies

Las metodologías Crystal fueron creadas por Alistair Cockburn. Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos.

La familia Crystal dispone un código de color para marcar la complejidad de una metodología: cuanto más oscuro un color, más pesado es el método. Cuanto más crítico es un sistema, más rigor se requiere. El código cromático se aplica a una forma tabular elaborada por Cockburn que se usa en muchas metodologías ágiles para situar el rango de complejidad.

El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas.

Los métodos se llaman de esta forma, evocando las facetas de una gema: cada faceta es otra versión del proceso, y todas se sitúan en torno a un núcleo idéntico. Hay cuatro variantes de metodologías: Crystal Clear (Claro como el cristal) para equipos de 8 o menos integrantes; Amarillo, para 8 a 20; Naranja, para

20 a 50; Rojo, para 50 a 100. Se promete seguir con Marrón, Azul y Violeta. La más exhaustivamente documentada es Crystal Clear que puede ser usada en proyectos pequeños de categoría D6, aunque con alguna extensión se aplica también a niveles E8 a D10. El otro método elaborado en profundidad es el Naranja, apto para proyectos de duración estimada en 2 años. Como casi todos los otros métodos, CC consiste en valores, técnicas y procesos. [21]

Dynamic Systems Development Method (DSDM)

Define el marco para desarrollar un proceso de producción de software. Nace en 1994 con el objetivo de crear una *metodología RAD* unificada. Originado en los trabajos de Jennifer Stapleton, directora del DSDM Consortium, DSDM se ha convertido en el *framework* de desarrollo rápido de aplicaciones (RAD) más popular de Gran Bretaña y se ha llegado a promover como el estándar del momento para desarrollo de soluciones de negocios sujetas a márgenes de tiempo estrechos.

Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio viabilidad, estudio del negocio, modelado funcional, diseño y construcción, y finalmente implementación. Las tres últimas son iterativas, además de existir realimentación a todas las fases.

Adaptive Software Development (ASD)

Su impulsor es Jim Highsmith. La estrategia entera se basa en el concepto de emergencia, una propiedad de los sistemas adaptativos complejos que describe la forma en que la interacción de las partes genera una propiedad que no puede ser explicada en función de los componentes individuales. Para Highsmith, los proyectos de software son sistemas adaptativos complejos y la optimización no hace más que sofocar la emergencia necesaria para afrontar el cambio.

Las principales características de ASD son: iterativo, orientado a los componentes software más que a las tareas y tolerante a los cambios. El ciclo de vida que propone tiene tres fases esenciales: especulación, colaboración y aprendizaje. En la primera de ellas se inicia el proyecto y se planifican las características del software; en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

ASD se concentra más en los componentes que en las tareas; en la práctica, esto se traduce en ocuparse más de la calidad que en los procesos usados para producir un resultado. En los ciclos adaptativos de la fase de Colaboración, el planeamiento es parte del proceso iterativo, y las definiciones de los componentes se refinan continuamente.

Feature -Driven Development (FDD)

FDD, es un método ágil, iterativo y adaptativo. A diferencia de otras metodologías ágiles, no cubre todo el ciclo de vida sino sólo las fases de diseño y construcción y se considera adecuado para proyectos mayores y de misión crítica. FDD no requiere un modelo específico de proceso y se complementa con otras metodologías. Enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso.

Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.

FDD consiste en cinco procesos secuenciales durante los cuales se diseña y construye el sistema. La parte iterativa soporta desarrollo ágil con rápidas adaptaciones a cambios en requerimientos y necesidades del negocio. Cada fase del proceso tiene un criterio de entrada, tareas, pruebas y un criterio de salida. Típicamente, la iteración de un rasgo insume de una a tres semanas. Las fases son:

- Desarrollo de un modelo general
- Construcción de la lista de rasgos
- Planeamiento por rasgo
- Diseño por rasgo y Construcción por rasgo

Lean Development (LD)

Lean Development (LD) es el método menos divulgado entre los reconocidamente importantes. Definida por Bob Charette a partir de su experiencia en proyectos con la industria japonesa del automóvil en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. En LD, los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en oportunidades que

mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.

Dado que LD es más una filosofía de administración que un proceso de desarrollo no hay mucho que decir del tamaño del equipo, la duración de las iteraciones, los roles o la naturaleza de sus etapas. Últimamente LD ha evolucionado como Lean Software Development (LSD). LD y LSD han sido pensados como complemento de otros métodos, y no como una metodología excluyente a implementar en la empresa. LD prefiere concentrarse en las premisas y modelos derivados de Lean Production, que hoy constituyen lo que se conoce como el canon de la Escuela de Negocios de Harvard. Para las técnicas concretas de programación, LD promueve el uso de otros métodos ágiles que sean consistentes con su visión, como XP o sobre todo Scrum.

Existen además herramientas para una vez decidida la metodología a utilizar en el proceso de desarrollo de un software, aplicar el método y poder modelar los artefactos correspondientes a las mismas. Dentro de estas herramientas existe un paquete representativo a nivel mundial: Rational, y que brinda una plataforma de trabajo que conecta todas sus funcionalidades permitiendo no sólo documentar los artefactos, sino que brinda otras características para administrar un proyecto en toda su totalidad.

Rational Suite 2003

Este paquete de Rational permite documentar un producto software brindando una serie de facilidades para la elaboración de los artefactos que propone el Proceso Unificado Racional. Está compuesto a su vez por un grupo de herramientas que se especializan en un aspecto en específico:

- **Rational Rose Enterprise Edition:** Esta herramienta permite modelar visualmente un grupo de artefactos que se generan como parte de la metodología. Dividido en cuatro vistas fundamentales: Vista de Casos de Uso, Vista Lógica, Vista de Componentes y Vista de Despliegue; propone un grupo de artefactos predefinidos que facilitan el desarrollo de software con calidad.
- **Rational Requisite Pro:** Esta herramienta gestiona los requerimientos funcionales y no funcionales del sistema a elaborar.
- **Rational Clear Case:** Rational ClearCase proporciona una gestión del ciclo de vida y control de los activos de desarrollo de software.

- **Rational ClearQuest:** proporciona un seguimiento flexible de defectos y cambios en toda la empresa. Soporte completo para consultas con generación de múltiples informes y gráficos.
- **Rational Soda:** Automatiza la documentación del proyecto de software a lo largo de todo el ciclo de vida. Genera documentos mediante la extracción de datos solicitados directamente de los repositorios de datos de herramientas.

Existen dentro del paquete otras herramientas con otras funcionalidades que gestionan y satisfacen todo el proceso de desarrollo de software.

Para el desarrollo de software de gestión existen un grupo de plataformas que permiten implementar un sistema brindando un grupo de facilidades para los desarrolladores.

MONO

Por definición, Mono es una *plataforma de desarrollo* de código abierto, basada en el *framework* .NET. Es una plataforma para ejecutar y desarrollar aplicaciones basadas en los estándares ECMA/ISO. Mono puede ejecutar aplicaciones hechas para los *framework* .NET y Java.

Mono permite a los desarrolladores construir aplicaciones Linux y multiplataforma de alta productividad. La implementación de Mono cumple los estándares ECMA para C# y la infraestructura de lenguaje común (Common Language Infrastructure, CLI).

Acoge varias licencias libres (X11, LGPL y GPL) y tiene una comunidad de desarrolladores activa, está esponsorizado por Novell y es la base para muchas aplicaciones.

Mono incluye *compiladores*, un intérprete compatible con el CLR de ECMA y un conjunto de *librerías*. Las *librerías* cubren la compatibilidad con Microsoft .NET (incluyendo ADO.NET, System.Windows.Forms y ASP.NET). Mono posee *librerías* adicionales y otras de terceras partes. Para el desarrollo de aplicaciones gráficas se incluye la librería GTK# que es un enlace sobre GTK+ y GNOME para permitir el desarrollo de aplicaciones nativas para Gnome utilizando Mono y cualquiera de los lenguajes soportados. El diseño de interfaces gráficas se puede realizar con Glade. Además está Monodevelop que es un IDE (entorno de desarrollo integrado) que integra la gestión de proyectos, la construcción de aplicaciones, depurado y toda la documentación (APIs, especificaciones, etc.) [22]

Los productos software encontrados como parte de la investigación en curso, no cumplen con todos los requisitos que el sistema de salud cubano desarrolla a diario en cada centro hospitalario del país. En alguno de los casos, existen sistemas que gestionan gran parte de los procesos de inscripción admisión, pero han sido desarrollados en un lenguaje de programación y plataformas de trabajo propietarios; de manera que no nos permite hacer uso de dichos productos, algunos son aplicaciones de escritorio y existen otros sistemas, que además de estas características mencionadas anteriormente, se les suma un elevado precio de venta incluyendo las licencias de los SW o *sistemas operativos* necesarios y que por cuestiones de bloqueo con nuestro país, se imposibilita la compra de los mismos.

A partir del estudio detallado de los diferentes sistemas de información hospitalarios relacionados, surge la necesidad de desarrollar un producto software que se adapte a las necesidades del sistema de salud cubano, que esté soportado por una *plataforma de desarrollo* de software libre y que permita a los clientes finales un entorno de trabajo amigable y flexible.

Para efectuar el diseño de dicho sistema, nos amparamos del Proceso unificado racional que hace uso de el lenguaje unificado de modelado UML, que en su conjunto conforman en la actualidad una de las metodologías más utilizadas en el desarrollo de grandes proyectos.

De las tecnologías, técnicas y metodologías que han sido objeto de estudio en este capítulo se seleccionó un grupo de ellas para conformar la propuesta tecnológica y así desarrollar un sistema de información hospitalario acorde con las posibilidades y necesidades de nuestro país, proponiendo la elaboración de un producto que al ser compilado en Mono pueda ser distribuido como software libre.

- Gestor de base datos: PostgreSQL 8.2
- Lenguajes de programación: ASP.net, C# y Java Script.
- Metodología AJAX.
- Metodología de Desarrollo de SW: Metodología RUP con notación UML
- Herramienta CASE de Modelado UML: Rational Suite 2003
- Servidor Web: Apache 2.0 (Mod_Mono)
- Navegador : IE6 y FireFox (compatibilidad absoluta)

- Servicios Web, con formato HL7.
- Chameleon 4.1.
- Framework .Net v2.0, Mono 2.0
- IDE's Desarrollo: VS2005.
- Herramienta Diseño Web: Macromedia Dreamweaver 8.0
- Herramienta de Control de Versiones: SubVersion v1.3.1, Tortoise 1.4.2, AnksVN 1.0
- Herramienta de Generación de Capa de Acceso a Datos: DAG.
- Herramienta de Diseño de Bases de Datos: Case Studio 2.2
- Herramienta de Análisis de Migración: MOMA.

Capítulo 2. Características del Sistema

El Sistema Nacional de Salud (SNS) dentro de sus niveles de atención médica, incluye a los hospitales como parte de la atención secundaria. El tema de inscripción admisión hospitalario es un eslabón fundamental en este nivel, que sirve de antesala para una serie de procesos que se encargan de gestionar todas las actividades delanteras en un hospital y que son imprescindibles para poder recibir cualquier servicio del mismo. Estos procesos a su vez generan un grupo de registros que soportan toda la información recuperada como parte de dichas acciones. Es por esto que se propone la elaboración de un sistema que informatice la gran mayoría de estos registros y se documenten los artefactos de desarrollo que se elaboren como resultado de esta tarea.

2.1 Objetivos estratégicos de la organización⁵

El Ministerio de Salud Pública (MINSAP) es el organismo rector del SNS y se encarga de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la salud pública, el desarrollo de las ciencias médicas y la industria médico farmacéutica.

El SNS, a su vez se compone por la forma y los métodos que sirven de base para la organización de la atención a la salud en un país determinado. Es el conjunto de unidades administrativas, de producción, investigación y servicios, responsabilizado con la atención integral de la salud de una población.

La estructura del SNS cubano se compone por 3 niveles, en correspondencia con los niveles administrativos del país:

- 1) El nivel nacional, representado por el MINSAP, como órgano rector con funciones metodológicas, normativas, de coordinación y de control.
- 2) El nivel provincial, representado por las direcciones provinciales de salud, las cuales están directamente subordinadas administrativa y financieramente a la Asamblea Provincial del Poder Popular (órgano de gobierno a esa instancia).
- 3) El nivel municipal, representado por las direcciones municipales de salud y dependientes, administrativa y financieramente de la Asamblea Municipal del Poder Popular.

⁵ Para la elaboración de este epígrafe se ha consultado: MARÍN DÍAZ, M. E. *Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de informática*. La Habana, 2006. p. 29.

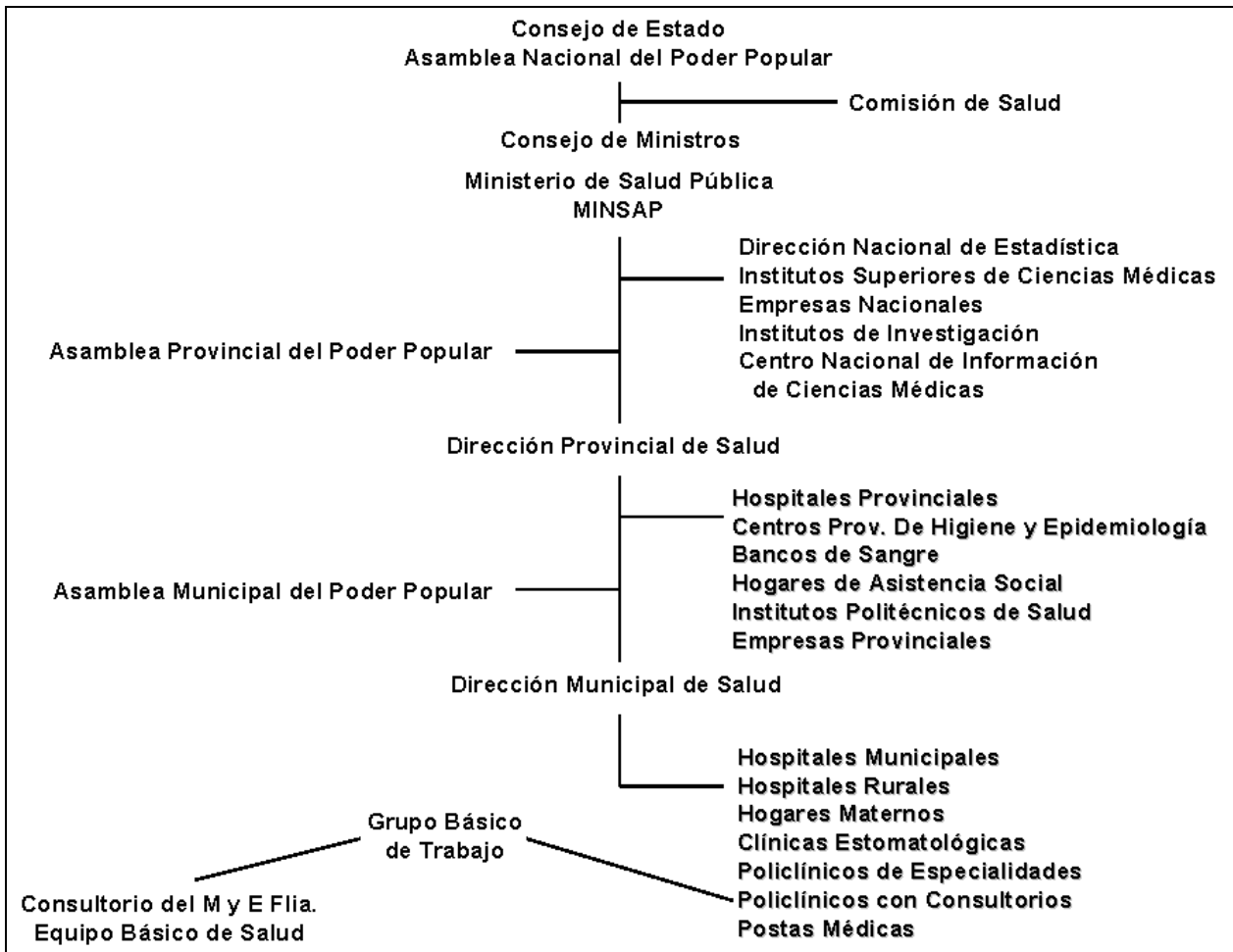


Figura 1. Organigrama del Sistema Nacional de Salud

Dentro de los objetivos básicos del SNS está elevar el nivel de salud de la población con estrategias encaminadas a fortalecer el enfoque de la atención primaria de salud; recuperar la vitalidad de los hospitales, perfeccionar la formación y capacitación de los recursos humanos, el desarrollo de la tecnología médica cubana, la consolidación del programa nacional de medicamentos y el uso de la medicina natural y tradicional; garantizar las especialidades y la introducción de tecnologías de punta en los programas; así como fortalecer la actividad de estomatología, ópticas, transporte sanitario, sistema integrado de urgencia médica y lograr una mayor participación de la comunidad, al organizar los Consejos de Salud a cada nivel e impulsar el movimiento de comunidades saludables.

Dentro de las funciones fundamentales del SNS se encuentra la atención médica hospitalaria que entra en la categoría de asistencia, para poder ejecutar esta función de forma correcta se hace necesario e imprescindible que en cada centro hospitalario exista un departamento de inscripción admisión que se encargue de inscribir a los pacientes y gestionar todos los movimientos del mismo, dentro del hospital.

El SNS define tres niveles de atención médica dentro de los cuales los hospitales forman parte del nivel secundario. Toda esta estructura por niveles para la atención médica también se divide en urbana y rural, de acuerdo con la ubicación de las unidades de salud y la población que atiende.

A medida que se hacen más complejos los servicios que se prestan estos niveles van aumentando y a su vez disminuye el volumen de pacientes que atiende.

El nivel secundario da cobertura a cerca del 15 % de los problemas de salud, su función fundamental es tratar al hombre ya enfermo, ya sea desde el punto de vista individual como colectivo pero también desempeña funciones de rehabilitación, promoción y prevención de la salud. Se llevan a cabo acciones de salud más complejas y especializadas (Especialidades). Comprende la atención médica brindada en los distintos hospitales.

Estructura organizativa de un hospital

Existen diversos criterios para clasificar los hospitales del SNS de Cuba como son:

Criterio de atención territorial:

Son provinciales cuando su atención se proyecta hacia la provincia e interprovinciales cuando brinda sus servicios a un grupo de provincias. Los hospitales nacionales permiten una cobertura a todo el país por sus características, recursos y nivel científico. Estos forman ya parte de un tercer nivel de atención.

Los hospitales provinciales reúnen todas las especialidades médicas con el desarrollo adecuado de las mismas para la atención eficiente y oportuna de la población que requiere de sus servicios.

Existen también hospitales municipales que contemplan los servicios básicos para la atención médica del territorio en el cual están enmarcados, así como hospitales rurales para las zonas alejadas de los poblados.

El SNS dispone en el ámbito secundario y terciario de centros de referencia y hospitales especializados e institutos de investigación de la salud que concentran mayor especialización y recursos de la tecnología

más avanzada. Los hospitales de cada provincia les remiten los casos que por su complejidad no puedan atender.

Por las características geográficas del país existen también hospitales de subordinación provincial que funcionan como hospitales de referencia en determinadas especialidades para las provincias vecinas.

Según perfil asistencial

Hospitales generales

Hospitales clínico-quirúrgicos

Hospitales pediátricos

Hospitales materno-infantiles

Hospitales especializados

Según la actividad docente

Hospitales docentes

Hospitales no docentes

Según ubicación

Hospitales urbanos

Hospitales rurales

En dependencia del tipo de hospital de que se trate, la estructura organizativa de un hospital variará en mayor o menor medida. Esta estructura está estrechamente relacionada con los diferentes servicios que brinda el centro asistencial.

2.2 Flujo actual de los procesos involucrados en el campo de acción

Dentro de los principales procesos que se llevan a cabo en un hospital, se encuentran los de inscripción admisión, estos a su vez involucran una serie de documentos y registros necesarios para los pacientes que se atienden en el hospital.

La inscripción del paciente es el primer proceso que se realiza en un hospital, siendo de vital importancia para poder asistir a una consulta especializada o al cuerpo de guardia, por lo que debe presentarse en

este departamento donde el auxiliar de inscripción le crea una nueva Historia Clínica y de esta forma queda registrado como paciente del hospital.

Luego de estar inscrito, el paciente puede asistir a consultas médicas que en alguno de los casos pueden orientarles un ingreso con el objetivo de intervenirlos quirúrgicamente, o para realizarle estudios; el ingreso también puede ser orientado por el médico en el cuerpo de guardia a raíz de una gravedad inmediata por un accidente, alguna causa traumática o simplemente cualquier afección que no sea tan grave pero requiera de cuidados médicos inmediatos para evitar posibles trastornos futuros.

De manera que el ingreso puede ser clasificado en dos tipos fundamentales: ingreso electivo e ingreso de urgencia. En el primero de los casos, el proceso surge como resultado de una consulta previa del paciente, al que se le orienta ser ingresado generalmente para ser intervenido quirúrgicamente. El paciente debe presentarse en el departamento de admisión, con su carné de identidad para que la auxiliar de admisión localice su HC, la cual debe contener en la hoja de datos generales la tenencia de donante, en caso de que esto no suceda el ingreso se suspende. En la HC se incluirá además, la orden de ingreso emitida por el médico. Luego de verificar las camas disponibles en la sala en que será internado el paciente, se procede a asignar la cama y a realizar el ingreso en sala.

El ingreso urgente sucede sin consulta previa, lo que implica que el proceso omita algunos pasos que en los caso de ingresos electivos se practican. Al presentarse el paciente, se le localiza la HC y de no tenerla o no encontrarla se crea una nueva, a la cual se le añade la orden de ingreso, luego la auxiliar de admisión asigna la cama al paciente y procede al ingreso en sala, registrando además todos los datos referentes al paciente que han sido manejados hasta ese momento.

Durante su estadía en el hospital la HC irá modificándose de manera continua e indicando la evolución o decadencia del paciente, estos datos ya no los maneja el departamento de inscripción admisión, quien sólo se encargará de gestionar traslados durante el ingreso, así como permutas entre salas o internas y finalmente se responsabiliza con los egresos de los pacientes ingresados.

Para ser dado de alta, el paciente primeramente debe estar ingresado y presentar las condiciones para ser egresado, ya sea por mejoría o por haber fallecido. En el primer caso luego de ser dado de alta su HC es enviada al departamento de Estadísticas que se encargará de tramitarla y gestionar datos de interés de esa sección. En el segundo de los casos, el paciente es dado de alta por fallecimiento y como

consecuencia, se generan el certificado de defunción y el registro de fallecidos, en caso de que la muerte haya sido traumática se envía al fallecido a Medicina Legal con los documentos correspondientes.

Al mismo tiempo que se realizan los procesos de inscripción admisión que se mencionan, se efectúa uno de los más importantes del departamento que es el llamado movimiento hospitalario.

Este proceso es realizado por el auxiliar de admisión diariamente, se genera el registro de movimiento hospitalario que es el resultado del registro de ingresos y egresos y que además guarda otros datos de interés. Luego de tener este registro creado, el auxiliar de admisión se encarga de elaborar el resumen diario de movimiento de salas donde se contabilizan la cantidad de ingresos, traslados, egresos, cantidad de camas, entre otros datos, para cada una de las salas del hospital. Como parte del movimiento hospitalario surgen otros documentos de vital importancia que constituyen estadísticas fundamentales para el buen funcionamiento del hospital, es el caso del registro diario de censo de pacientes y el informe de movimiento hospitalario que se elabora cada cierto período de tiempo.

Todos estos procesos que se realizan en los hospitales cubanos, son efectuados de forma manual, no existe un software estandarizado a nivel nacional. Esto afecta de gran manera la administración de los hospitales, pues como no se puede tener una información actualizada de todos los insumos necesarios se gastan más recursos. El departamento de inscripción admisión es uno de los principales generadores de datos del hospital, debido a que es el encargado de inscribir a todos los pacientes y brindarle el servicio de ingreso para que dentro del mismo, puedan recibir las atenciones de cualquier especialidad que exista en el hospital.

Al no tener un sistema automatizado se valen de documentos en formato Excel y que luego deben ser impresos para poder mantener actualizada la información en los diferentes niveles. En ocasiones se crean HC duplicadas a causa de que al paciente se le perdió el número de la anterior y es muy engorroso buscar, de forma rápida y eficiente, en un tarjetero que contenga miles de nombres.

2.3 Objeto de automatización

Como resultado de estos procesos se generan un grupo de documentos que serán el principal objeto de automatización de la investigación en curso; registros tales como la historia clínica, orden de ingreso, registro de camas vacantes, registro de ingresos, registro de movimiento hospitalario, junto a los procesos

que los generan dejarán de hacerse de forma manual con el desarrollo de un sistema de gestión hospitalario que se encargue de informatizar todas las actividades de la sección de inscripción admisión.

El proceso de inscripción en el hospital, se vale de un registro donde se guardan las inscripciones que se efectúan en el centro, este almacena todas las ocurrencias de esta actividad de manera que, luego se pueda manejar toda la información de las inscripciones realizadas y sacar estadísticas, así como realizar estudios de casos de importancia para el hospital.

Durante este proceso, se crea la HC del paciente que guardaría todos los datos del mismo en el momento de su registro en el hospital. Esta historia es manipulada en cada momento que sea atendido el paciente y en ella, se evidencian los datos evolutivos de los mismos y las dolencias o enfermedades padecidas, así como las intervenciones quirúrgicas realizadas y los ingresos tanto urgentes como electivos. La historia clínica cuenta con una hoja de datos generales, en la que se guarda justamente la información corriente del paciente y algunos datos de interés para el hospital.

El proceso de admisión engloba un grupo de registros para poder realizar los ingresos, egresos y traslados. El registro de ingresos se encarga de guardar el listado de los ingresos realizados, que incluyen una lista de HC con los datos de las mismas de forma global así como el lugar físico donde ha sido hospitalizado cada paciente del registro.

Para poder efectuar el ingreso se hace necesario saber la disponibilidad que tienen las camas de las diferentes salas que pertenecen a un servicio determinado. Conociendo esta información puede realizarse la asignación de camas a los pacientes que ingresan; para poder realizar esta tarea, el auxiliar de admisión cuenta con un registro de camas vacantes que muestra el estado de las camas por sala y por servicio.

Cuando el paciente debe ser ingresado por cuestiones quirúrgicas, el ingreso es de tipo electivo y se realiza en una fecha ya dada con antelación por un médico determinado en una consulta previa, de manera que a la hora de ingresar, en la orden de ingreso y la hoja de datos generales debe estar plasmado que el mismo cuenta con una donación de sangre. Cuando el paciente no cumple con este requisito no puede ingresar, pero puede suceder que sea ingresado de urgencia, para ser intervenido quirúrgicamente porque ha tenido un accidente o una situación que requiera una operación al instante.

Puede darse el caso de que el día correspondiente al ingreso del paciente, en el servicio que había sido asignado no existan salas con camas disponibles, en ese caso se ingresa en otro servicio y se especifica que este tipo de ingreso es fuera de servicio, generándose así un nuevo registro que se encargaría de listar todos los pacientes dentro del hospital que se encuentra ingresados fuera de servicio.

En ocasiones, los ingresos urgentes también pueden tener pacientes fuera de servicio, debido a que la causa por la que son ingresados requiere de un servicio que está lleno y en este caso se crearían las condiciones para ingresarlo en otro servicio hasta que pueda ser trasladado. Cuando se efectúa un ingreso urgente, en ocasiones no hay tiempo de buscar la HC del paciente debido a que la gravedad del mismo no permite demoras, de forma tal que se hace necesaria la creación de una nueva HC con la hoja de datos generales y la orden de ingreso para que se efectúe de forma urgente.

Los traslados pueden efectuarse dentro, o hacia otro hospital. Cuando se presenta el caso de un paciente ingresado fuera de servicio y una de las camas de una sala del servicio al que debía entrar desde que se efectuó el ingreso es liberada, entonces se procede a trasladar al paciente hacia esa cama. Pueden efectuarse traslados debido a decisiones internas del hospital y estos en su conjunto traerían como consecuencia la elaboración de un registro de traslados donde se guarden los datos fundamentales de los pacientes trasladados, de que sala venían y a cual han sido cambiados, para en caso de que se quiera anular luego el traslado, saber cómo llevar a su lugar de origen a un paciente trasladado. También se guardan registros que son estadísticos y que visualizan la cantidad de traslados que ha tenido una sala en un período de tiempo determinado. Cuando el traslado es hacia otro hospital se maneja como un egreso.

Cuando los pacientes presentan las condiciones necesarias son dados de alta, y este movimiento es contabilizado en el registro de egresos, que tendría un listado de todos los egresos realizados en el hospital, en las salas de los servicios existentes en el mismo, así como los nombres y números de HC de los pacientes egresados; se controla también el lugar donde estaban ingresados antes de ser dados de alta para en caso de que exista la necesidad de anular el egreso poder manejar la información previa al egreso practicado.

Los pacientes no sólo egresan en buenas condiciones, sino que pueden ser dados de alta por fallecimiento. Cuando esto sucede se generan un grupo de documentos en los que se guardan los datos, dentro de los cuales se encuentran el registro de fallecidos patológicos y el certificado de defunción, que se emiten sin importar las causas del fallecimiento, si la muerte fue por causas traumáticas, externas o

violentas se guardan los datos en el registro de fallecidos médico legal que contiene los números de HC y los nombres de los pacientes incluyendo además las causas específicas de la muerte, luego de ser llenado este registro, el hospital elabora la remisión a medicina legal donde se le practicarán los estudios pertinentes al fallecido, para determinar la información necesaria de tipo policial relacionada.

Existe un registro que se encarga de guardar los datos más importantes por cada sala y que brinda la información exacta de las condiciones del hospital, en la hora del cambio de guardia, que se efectúa cada mañana en los hospitales de nuestro país, este es el registro movimiento hospitalario y que se encarga de guardar por cada sala de un servicio los ingresos, traslados, los egresos ya sean vivos o fallecidos y la cantidad de camas vacantes.

Como parte del proceso de inscripción admisión se generan datos estadísticos que brindan la opción al hospital de realizar estudios pormenorizados de las fechas prioritarias de inscripciones, ingresos, etc. Estos registros brindarían solamente los cálculos estadísticos de las cantidades de inscripciones realizadas en un período de tiempo determinado, los ingresos y sus causas, los egresos y traslados, permitiendo al hospital una mayor preparación en determinadas épocas del año para afrontar un gran volumen de pacientes, ingresar a muchas personas por causas determinadas, egresar masivamente salas del hospital, entre otras conclusiones a las que permiten arribar estos estudios.

De manera que los registros que se obtienen durante todo el proceso de inscripción admisión hospitalario serían:

- Registro de inscripciones realizadas
- Historia clínica
- Orden de ingreso
- Registro de camas vacantes
- Registro de ingresos
- Hoja de datos generales
- Registro de casos fuera de servicio
- Registro de traslados

- Registro de egresos
- Registro de fallecidos patológicos
- Certificado de defunción
- Registro de fallecidos médico legal
- Remisión a medicina legal
- Registro movimiento hospitalario
- Resumen diario de movimiento de salas
- Registro diario de censo de pacientes
- Informe movimiento hospitalario

A partir de la identificación de este grupo de registros que se manejan durante la inscripción admisión hospitalaria, surge la necesidad de la producción de un sistema software que sea capaz de informatizar estos procesos a través de una interfaz fácil de usar y entendible para los trabajadores de esta sección del hospital.

El sistema debe permitir inscribir pacientes y mostrar un listado de todas las inscripciones realizadas en un rango de tiempo determinado por el usuario. En lo concerniente a los ingresos, se propone la elaboración de un producto que sea capaz de ingresar, reubicar y además anular el ingreso de un paciente, así como brindar otras funcionalidades propias del proceso, informatizando todos los registros que hoy se llenan de forma manual.

Los egresos y traslados deben ser tratados de manera prioritaria al igual que las demás actividades, el sistema debe permitir gestionar estos procesos y de forma general mostrar reportes estadísticos de los ingresos, inscripciones, traslados, egresos y casos fuera de servicio existente en el hospital. Se propone una aplicación Web que gestione los procesos antes mencionados y que permitirá que todos los usuarios puedan acceder al mismo sin necesidad de instalación de un ejecutable, además que cualquier cambio que se realice al mismo será visto sin necesidad de actualizaciones en cada puesto de trabajo.

Se plantea como parte del desarrollo del trabajo de diploma, realizar el análisis y diseño del sistema software propuesto, siguiendo la metodología de desarrollo RUP utilizando el lenguaje de modelado UML.

2.4 Modelo de Negocio

La disciplina de modelado de negocio es la primera que propone RUP dentro del ciclo de desarrollo de un software, tiene su mayor peso durante la fase de inicio debido a que permite conocer los procesos existentes actuales de cualquier entidad o empresa para la que se vaya a desarrollar el sistema. Es en este flujo de trabajo donde se conocen a fondo cómo son iniciadas cada una de las actividades de un sector determinado, y a través del modelado de estos procesos se obtiene una visión más amplia del negocio existente.

RUP propone, que para esta disciplina se modele el negocio siempre y cuando los procesos sean fácilmente identificables, en el caso de que no puedan percibirse se realizaría un modelo de dominio, que englobaría los principales conceptos que sean encontrados para a partir de este modelo, pasar al flujo de trabajo de requerimientos.

En el caso que nos embarga utilizamos modelo de negocio, para establecer una abstracción de la organización y con el objetivo de lograr un buen entendimiento del negocio para la construcción de un sistema correcto, a través del desarrollo de un conjunto de artefactos que permiten modelar íntegramente el mismo.

2.4.1 Actores del Negocio

Un actor del negocio es el principal beneficiado de los procesos existentes, es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actor	Descripción
Paciente	Principal beneficiado de los procesos de negocio pues recibe todos los privilegios del departamento de Inscripción Admisión hospitalaria.
JSAI(Jefe Sección Admisión Información)	Principal beneficiado de la actualización del movimiento hospitalario para brindar mejor

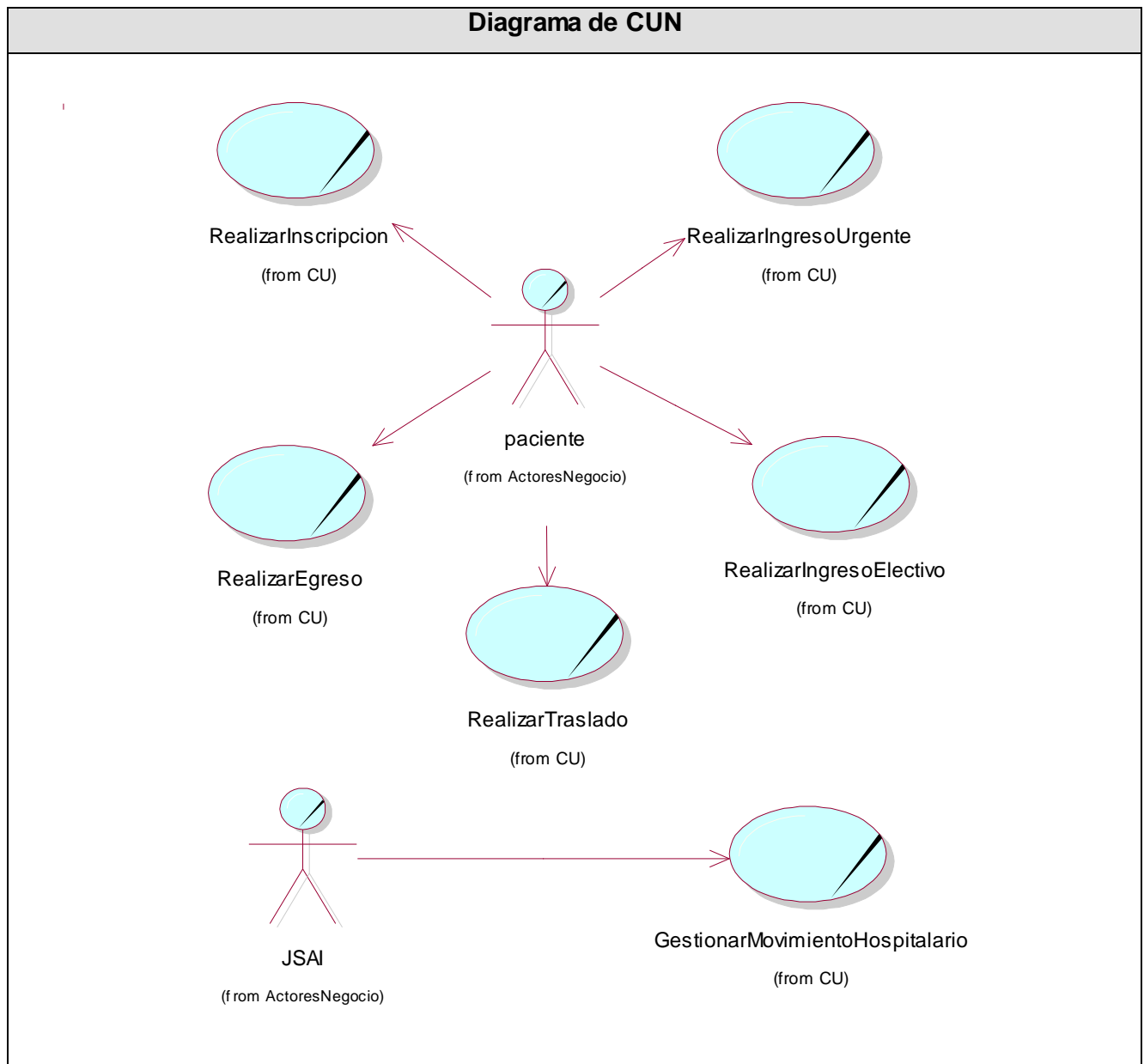
	servicio de la sección y para las entregas de guardias.
--	---

2.4.2 Trabajadores del Negocio

Un trabajador del negocio es la abstracción de una persona o sistema software que representa un rol desempeñado en las realizaciones de CUN. Un trabajador del negocio colabora con otros trabajadores, es notificado de los eventos del negocio y manipula las entidades del negocio para realizar sus responsabilidades.

Trabajador	Descripción
Auxiliar Admisión	Principal protagonista de los procesos de negocio en admisión, es el encargado de emitir reportes y documentos importantes así como interactuar con el paciente y otros trabajadores del negocio.
Auxiliar Inscripción	Principal protagonista de los procesos de negocio en Inscripción. Es el encargado de inscribir a los pacientes en este departamento de la sección y crear las HC.
Auxiliar Archivo	Es el encargado de brindar información del departamento de archivos y gestionar datos relacionados con la HC.

2.4.3 Diagrama de Casos de Uso del Negocio



2.4.4 Diagramas de Actividades

Diagrama de Actividades <Realizar Inscripción>

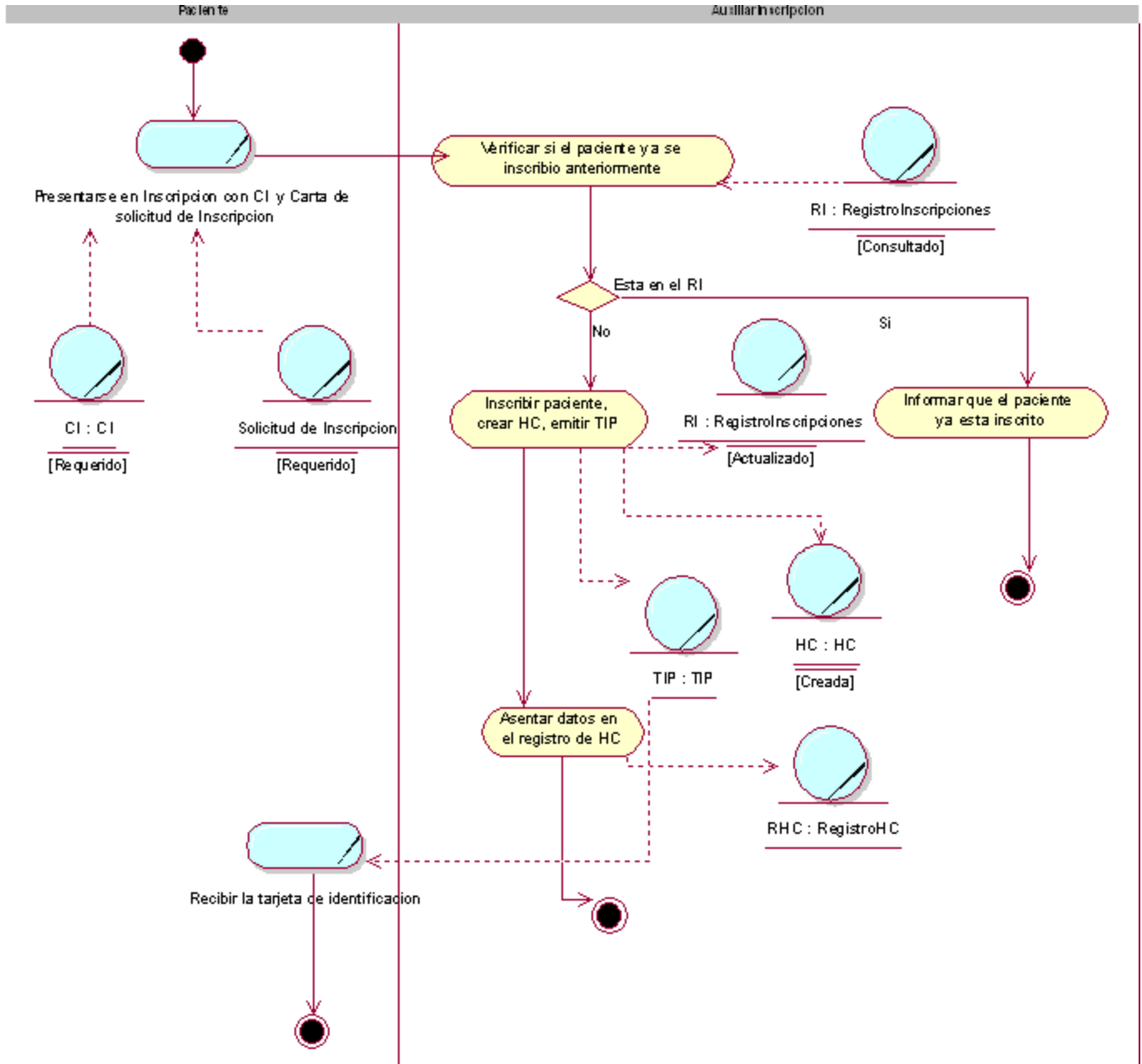


Diagrama de Actividades <Realizar Egreso>

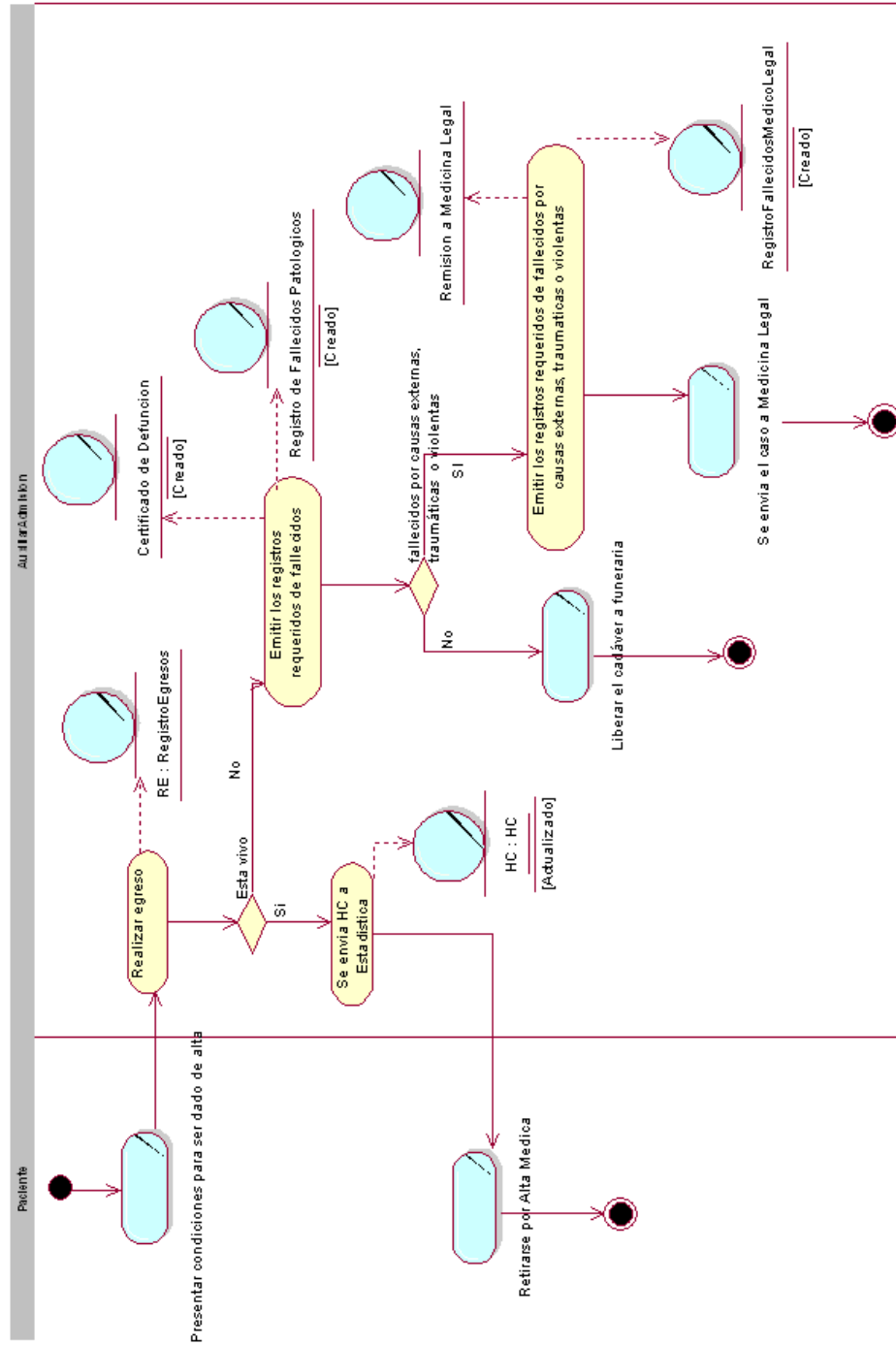


Diagrama de Actividades <Realizar Ingreso Electivo>

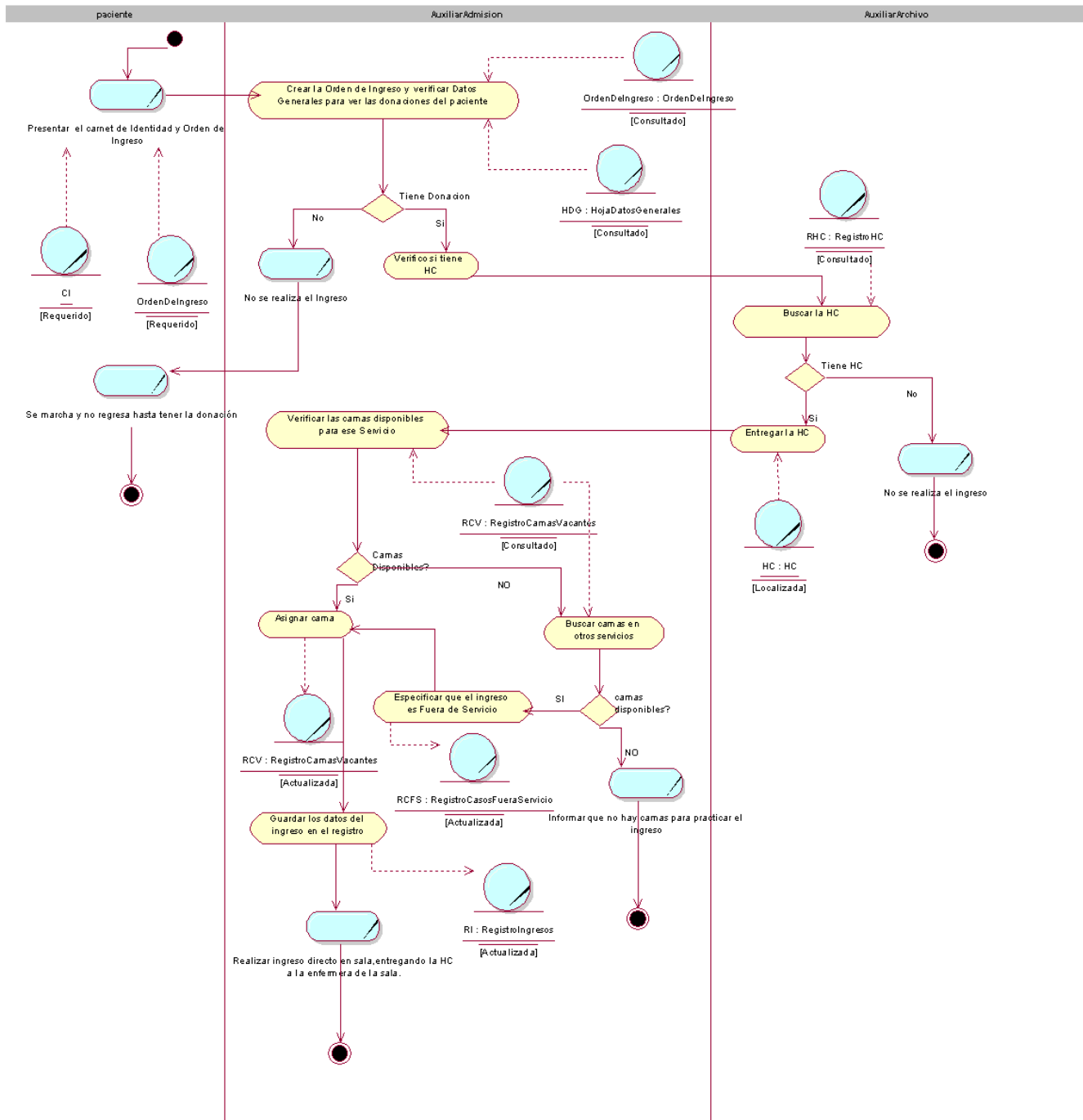


Diagrama de Actividades < Realizar Ingreso Urgente >

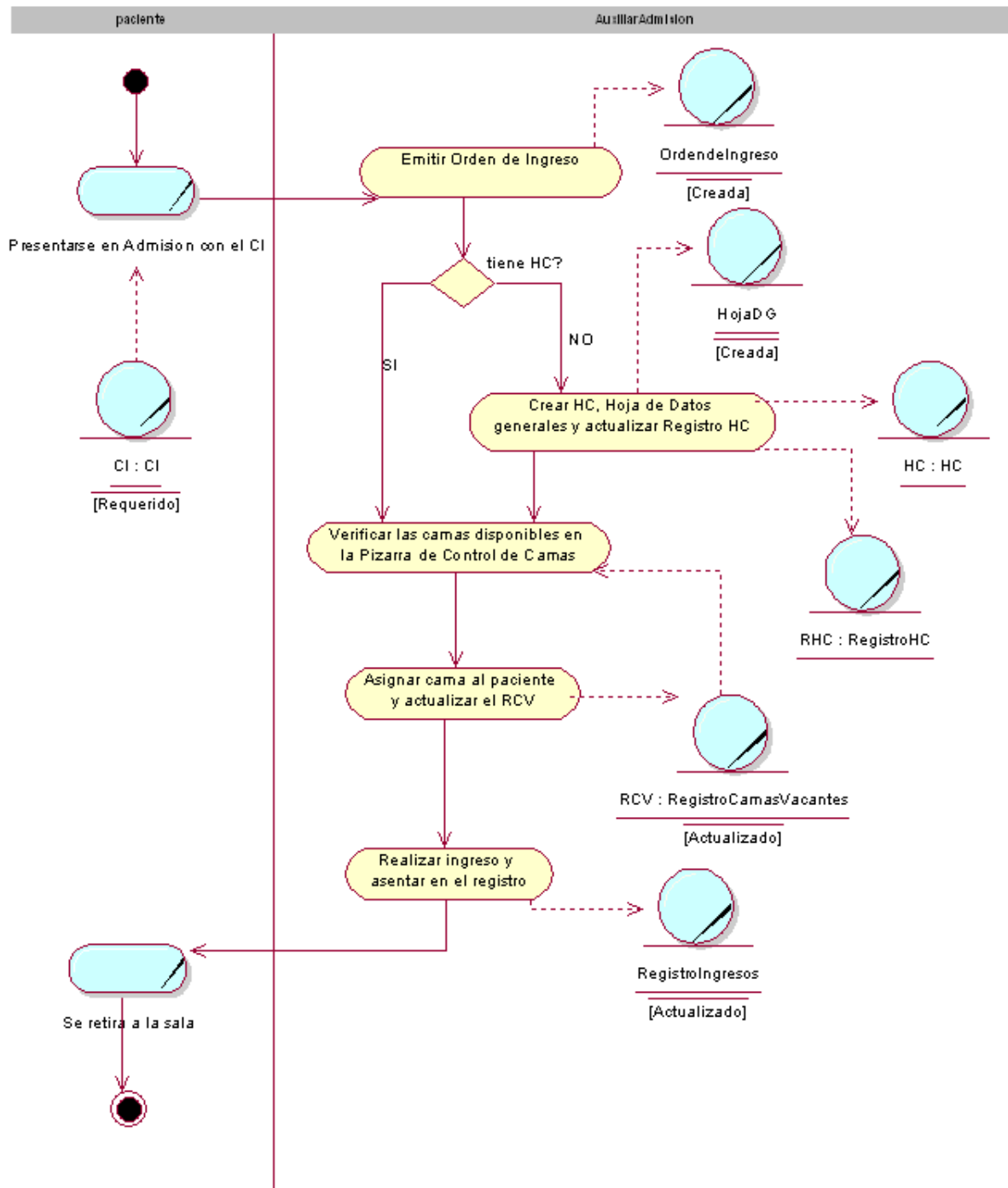


Diagrama de Actividades < Realizar traslado >

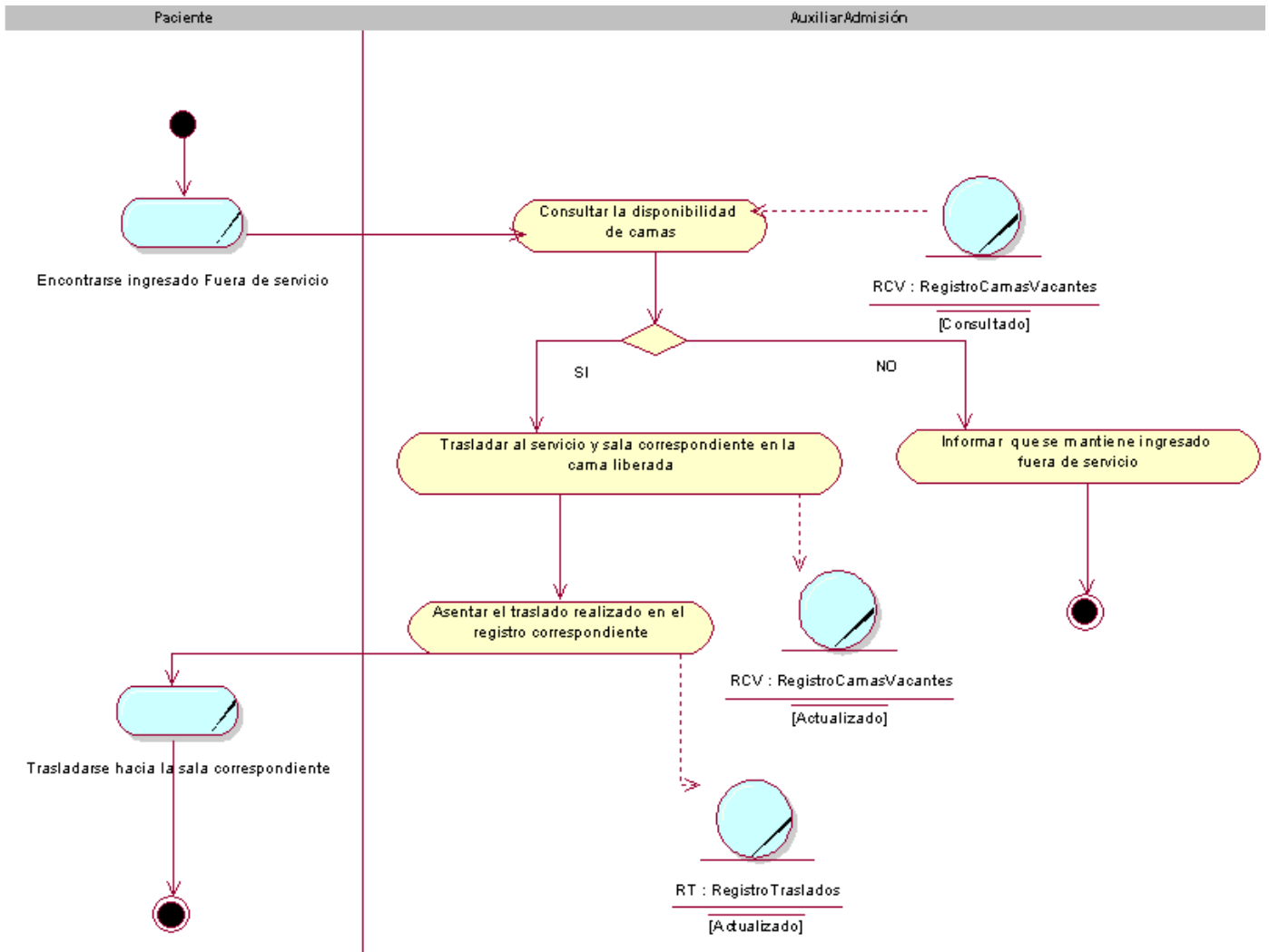
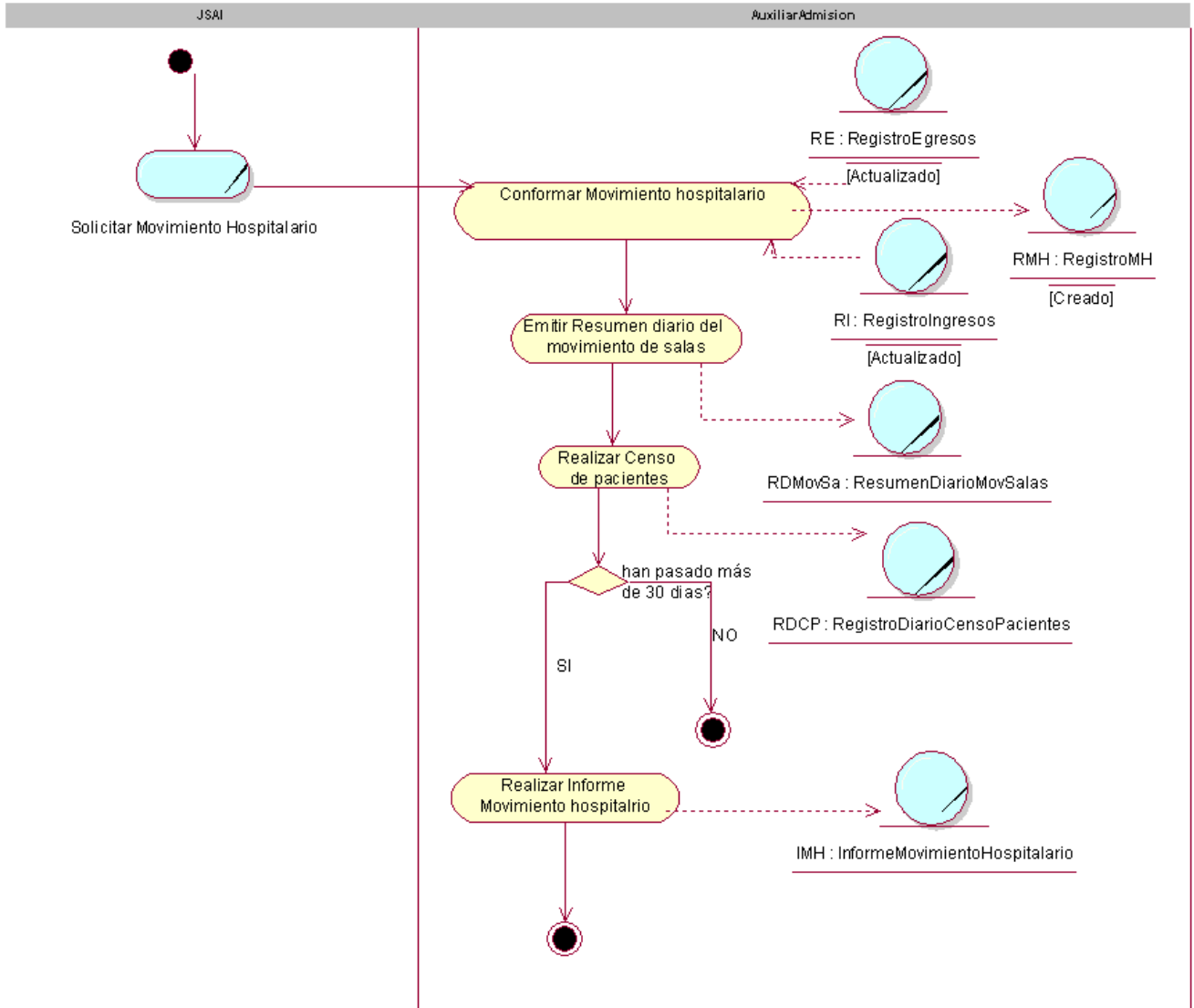
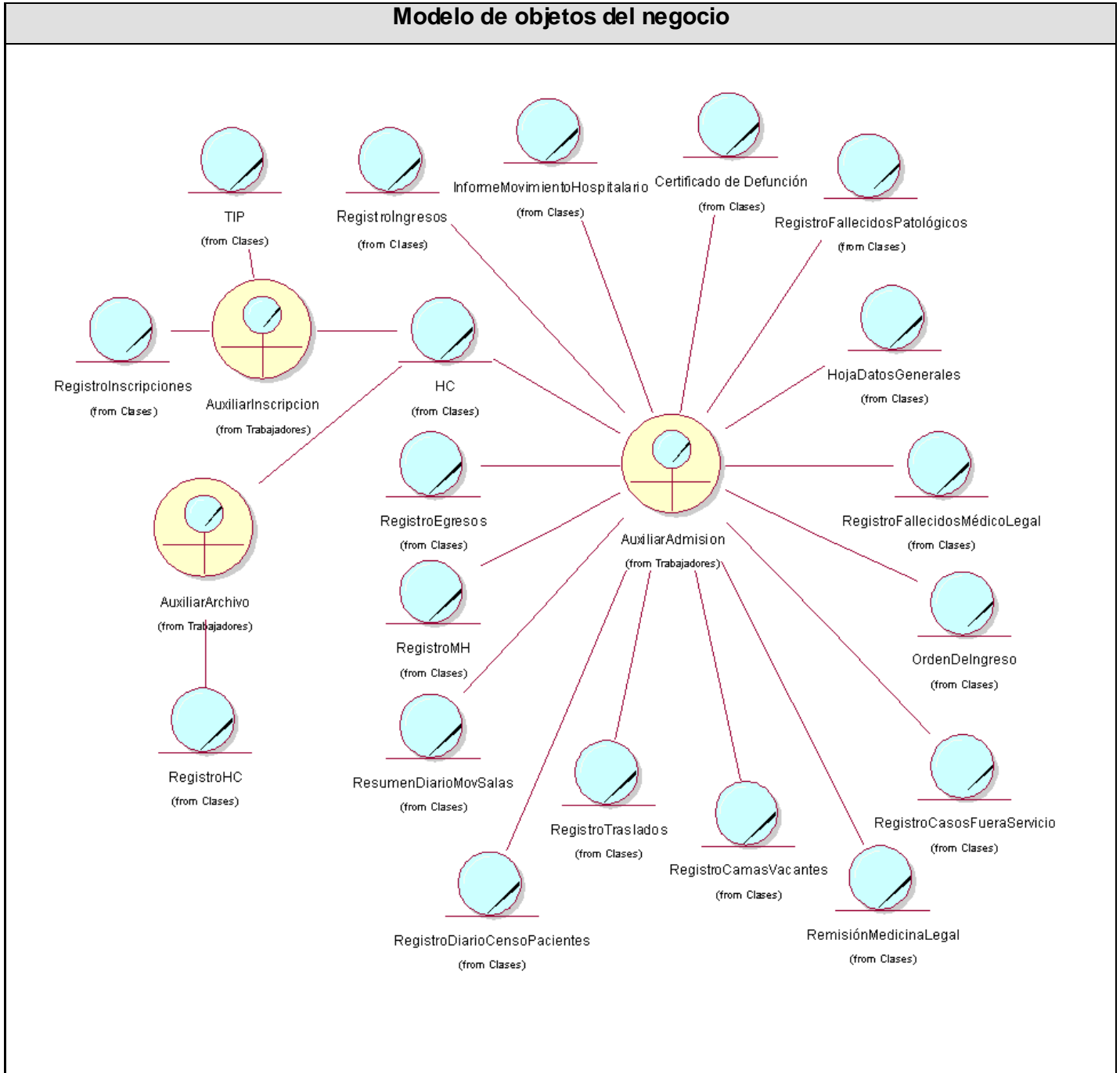


Diagrama de Actividades < Gestionar Movimiento Hospitalario >



2.4.5 Modelo de objetos



El proceso unificado racional propone, dentro de sus disciplinas, que a continuación de la elaboración del Modelado de Negocio, continúe el ciclo de desarrollo con el flujo de trabajo de requerimientos, que se encarga de extraer los requisitos funcionales que son las capacidades que debe tener el sistema y los requisitos no funcionales que evidenciarían cómo y con qué se elaboraría este software propuesto; a partir de estos se construye el Diagrama de casos de uso y se definen los que son arquitectónicamente significativos.

2.5 Especificación de los requisitos de software

El sistema que se propone forma parte del Sistema de gestión hospitalaria, siendo un subsistema del mismo y que al integrarse con otros conformarán el producto software esperado para brindar una solución informática a los hospitales del país. De esta forma el subsistema de inscripción admisión se nutre de funcionalidades de otros módulos tal es el caso del módulo de administración que implementa la funcionalidad referente a inscribir a los trabajadores del hospital, incluyendo a los médicos, quienes son los responsables de autorizar un ingreso, una inscripción y un egreso.

Existe además un subsistema de vital importancia para todos los módulos y para el producto software final, este es el módulo de seguridad que se encarga de implementar justamente todo lo referente a la seguridad del sistema.

Requisitos Funcionales

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas se pueden definir los siguientes requisitos funcionales:

1. Autenticarse
2. Gestionar Inscripción
 - a. Inscribir paciente
3. Gestionar ingresos

- a. Registrar Ingreso
 - b. Reubicar Ingreso
 - c. Anular Ingreso
4. Localizar camas disponibles
5. Buscar paciente
6. Gestionar Traslado
 - a. Realizar traslado
 - b. Permutar pacientes
 - c. Anular traslado
7. Gestionar Egreso
 - a. Realizar egreso
 - b. Anular Egresos
8. Realizar egresos masivos
9. Buscar ingresos x sala
10. Mostrar informe de inscripciones realizadas
11. Mostrar registro de pacientes
12. Mostrar registro de ingresos
13. Mostrar reporte de ingresos
14. Mostrar registro de casos ingresados fuera de servicio
15. Mostrar registro de camas vacantes
16. Mostrar registro de traslado
17. Mostrar egresos
18. Mostrar reporte de egresos

19. Mostrar registro de movimiento hospitalario
20. Mostrar resumen diario de movimiento de salas
21. Mostrar registro diario censo de pacientes
22. Mostrar informe de movimiento hospitalario

Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Muestran las características que hacen al producto atractivo, usable, rápido o confiable. No alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importarle con que propiedades o cualidades se relacionen.

Requerimientos de apariencia o interfaz externa

El sistema debe tener un ambiente amigable y entendible para los usuarios finales, de forma tal que no les sea muy complicado utilizar el software.

No debe utilizarse tecnología de *frames*.

Cada página no debe exceder los 500 Kb en imágenes.

Requerimientos de usabilidad

La aplicación debe cumplir con los principales principios de usabilidad, debe brindarse comodidad a la hora de acceder a las diferentes funcionalidades que brinda la aplicación mediante teclas de acceso rápido, la navegabilidad no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por el usuario.

Requerimientos de rendimiento

El tiempo de respuesta de una petición al *servidor* debe ser rápido para la toma de decisiones.

Requerimientos de soporte

Se le debe dar mantenimiento periódico a los *servidores* de bases de datos controlando la integridad de la información.

Requerimientos de portabilidad

El producto podrá ser usado bajo cualquier sistema operativo ya sea Linux o Windows.

Requerimientos de seguridad y privacidad

La información debe transmitirse de manera segura, se debe garantizar la seguridad a todos los niveles (Interfaz, negocio y Acceso a datos) restringiendo las funcionalidades mediante roles de usuarios garantizando que la información sea accesible al usuario autorizado.

Requerimientos de confiabilidad

La información debe transmitirse a través de canales seguros. Se debe chequear la integridad de los datos.

Requerimientos de ayudas y documentación en línea

Se debe brindar una interfaz amigable que explique las diferentes funcionalidades con que cuenta el sistema de manera rápida, además los manuales de usuario y toda la documentación actualizada de cada módulo de la aplicación.

Requerimientos de hardware

Requerimientos para una estación de trabajo: 256Mb RAM (Recomendado 512Mb), 1GHz, 10Gb HDD.

Requerimientos para un *servidor*: 512Mb RAM (Recomendado 1Gb RAM o superior), 1GHz o superior, 60Gb HDD.

Requerimientos de software

El sistema debe correr en sistemas operativos Windows, Unix y Linux utilizando la plataforma Mono 1.2.4 o superior. El cliente solo deberá disponer de un navegador Web (IE6 o Firefox).

Restricciones en el diseño y la implementación

El sistema será implementado utilizando como lenguaje de programación del lado del cliente java script auxiliándose del grupo de tecnologías AJAX.

Se utilizará la *plataforma de desarrollo* Microsoft .NET que brinda una gama de facilidades en su entorno y que da la posibilidad de utilizar el lenguaje C Sharp del lado del *servidor*.

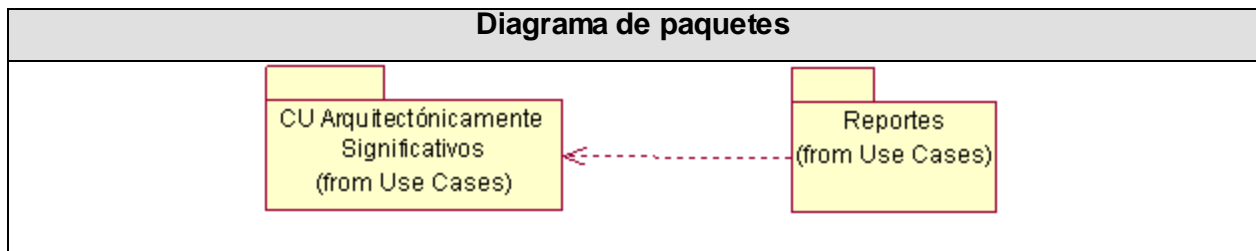
Se utilizará Chameleon 4.1 para la comunicación HL7 y Npgsql para la conexión al servidor de bases de datos.

2.6 Diagrama de paquetes

Para un desarrollo comprensible del sistema, se han conformado paquetes o subsistemas en los que se agrupan los casos de uso. Los criterios utilizados se basan en las funcionalidades que brindan los casos de uso, de manera que el diagrama se compone por 2 paquetes, uno de ellos se encarga de manejar los CU arquitectónicamente significativos o de prioridad crítica y el otro gestiona todos los reportes o salidas del sistema.

El paquete de CU Arquitectónicamente Significativos agrupa los casos de usos de prioridad para el desarrollo del producto software, este se vale del paquete de Administración para servirse de la funcionalidad que permite buscar a un médico y del paquete Módulo Seguridad que se encarga de gestionar la autenticación de usuarios así como tratar la seguridad del sistema.

El paquete de Reportes se encarga de listar un grupo de salidas que el sistema debe proporcionar a los usuarios finales, de manera que este se respalda en el subsistema de CU Arquitectónicamente Significativos.

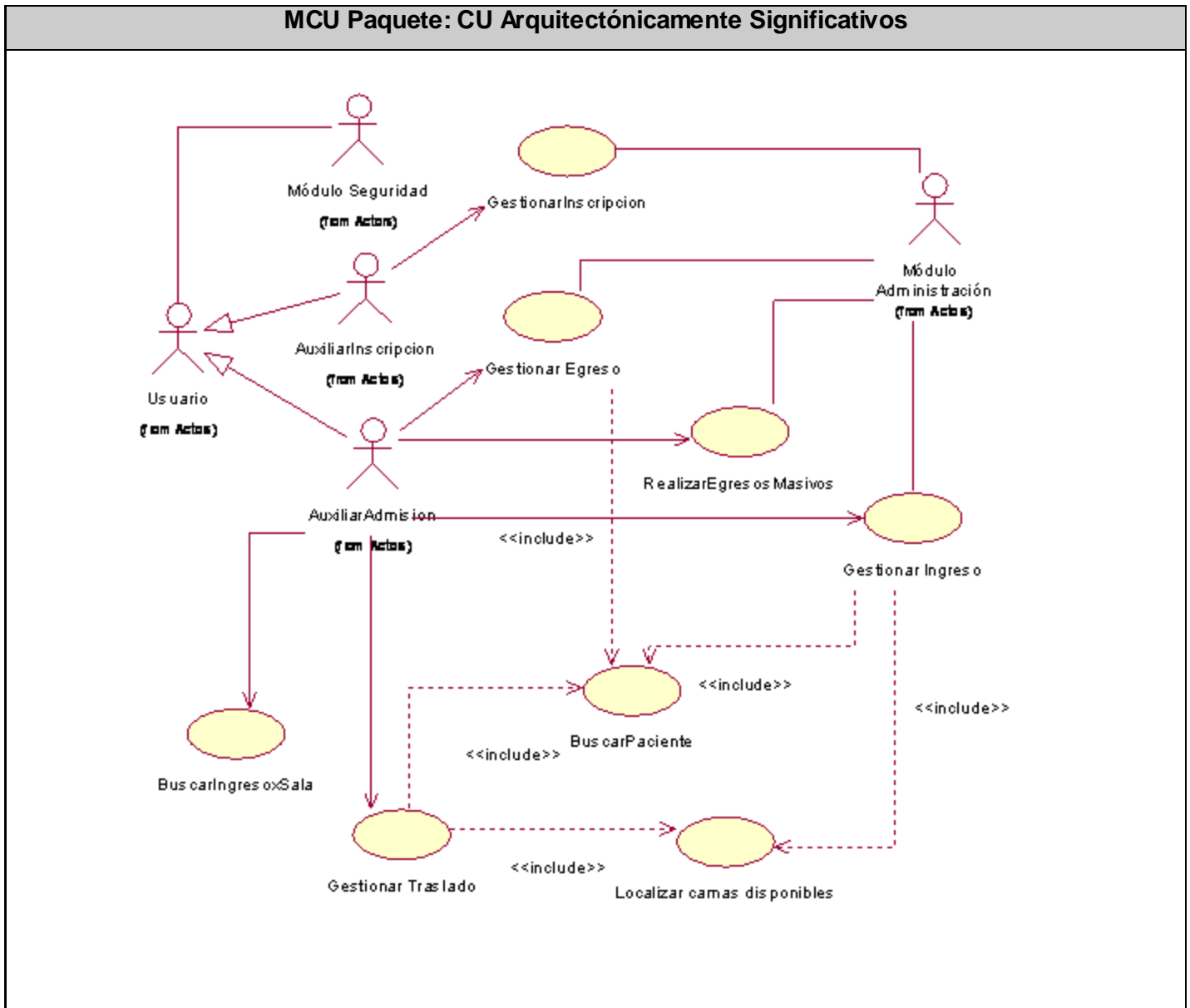


2.6.1 Actores del sistema

Actor	Descripción
Usuario	Usuario global que permite la autenticación en el sistema y que valida al mismo dándole un rol ya sea de auxiliar de inscripción como de admisión
Auxiliar de Inscripción	Encargado de inscribir un paciente y generar una nueva Historia Clínica.
Auxiliar de Admisión	Encargado de manejar los ingresos de un paciente, así como los egresos y los traslados de un paciente.
Módulo Archivo	Sistema automatizado que gestiona la HC, después de ser creada en inscripción, se encarga de manipular los datos referentes a la misma así como informar el estado en el que se encuentra la HC.
Módulo Administración	Sistema automatizado que gestiona los datos administrativos del hospital, registra a todos los trabajadores del centro y permite hacer búsquedas de médicos por especialidad entre otras funcionalidades.

2.6.2 Diagramas de Casos Uso

Modelo de Casos de uso Paquete: CU Arquitectónicamente Significativos



Para abundar más en el Modelo de casos de uso del sistema ver [anexo 2](#).

Para solucionar el problema existente en el sector hospitalario, específicamente en la sección de inscripción admisión se propone el diseño del Sistema de gestión hospitalaria que debe garantizar la informatización de los procesos referentes a este departamento.

Para lograr el desarrollo del sistema se realizaron los artefactos correspondientes a las disciplinas de negocio y requerimientos que propone la metodología RUP utilizando notación UML. La entidad para quien tributará el producto a desarrollar, solucionará los procesos manuales y agilizará la atención a los pacientes de este nivel de atención de la salud.

Capítulo 3. Análisis y Diseño

A partir del diagrama de casos de uso, obtenido como parte del flujo de trabajo de requerimientos, se puede establecer una traza con la disciplina que continúa el ciclo de vida de RUP, esta se encarga de describir en términos de clases cómo debe funcionar cada caso de uso y las colaboraciones que existen entre las clases definidas.

De esta forma se obtienen los diagramas de clases de análisis correspondientes a cada caso de uso, conformando las realizaciones de los mismos conjuntamente con los diagramas de interacción hasta llegar a refinar este análisis en el diseño esperado para la aplicación a desarrollar, con el objetivo de transformar los requerimientos, al diseño que el sistema debe tener y con la finalidad de adaptarlo al ambiente de implementación, trazando su funcionamiento. Para el desarrollo satisfactorio de esta disciplina la metodología propone el uso de patrones con el objetivo de lograr una arquitectura robusta para el sistema en cuestión.

3.1 Diagramas de clases de análisis

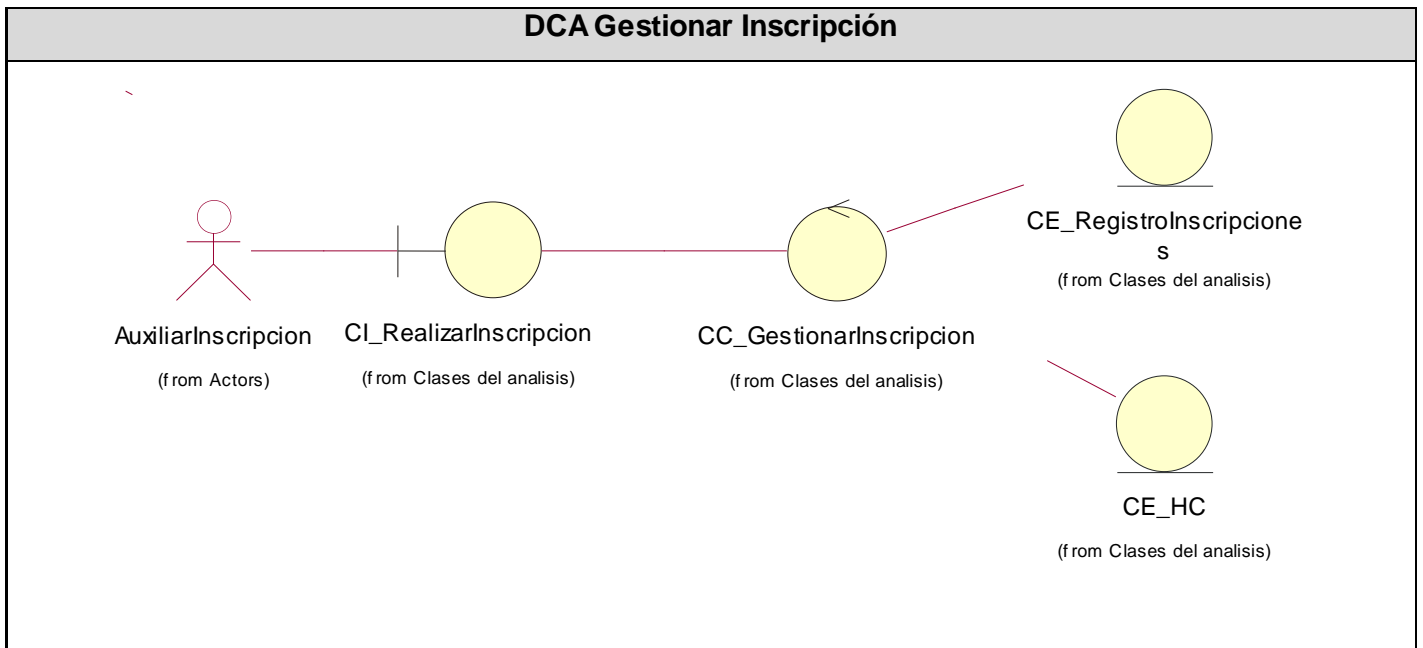
Los diagramas de clases de análisis, expresan la definición y relación entre las clases, están compuestos por clases que pueden ser de tres tipos fundamentales: interfaz, controladora y entidad.

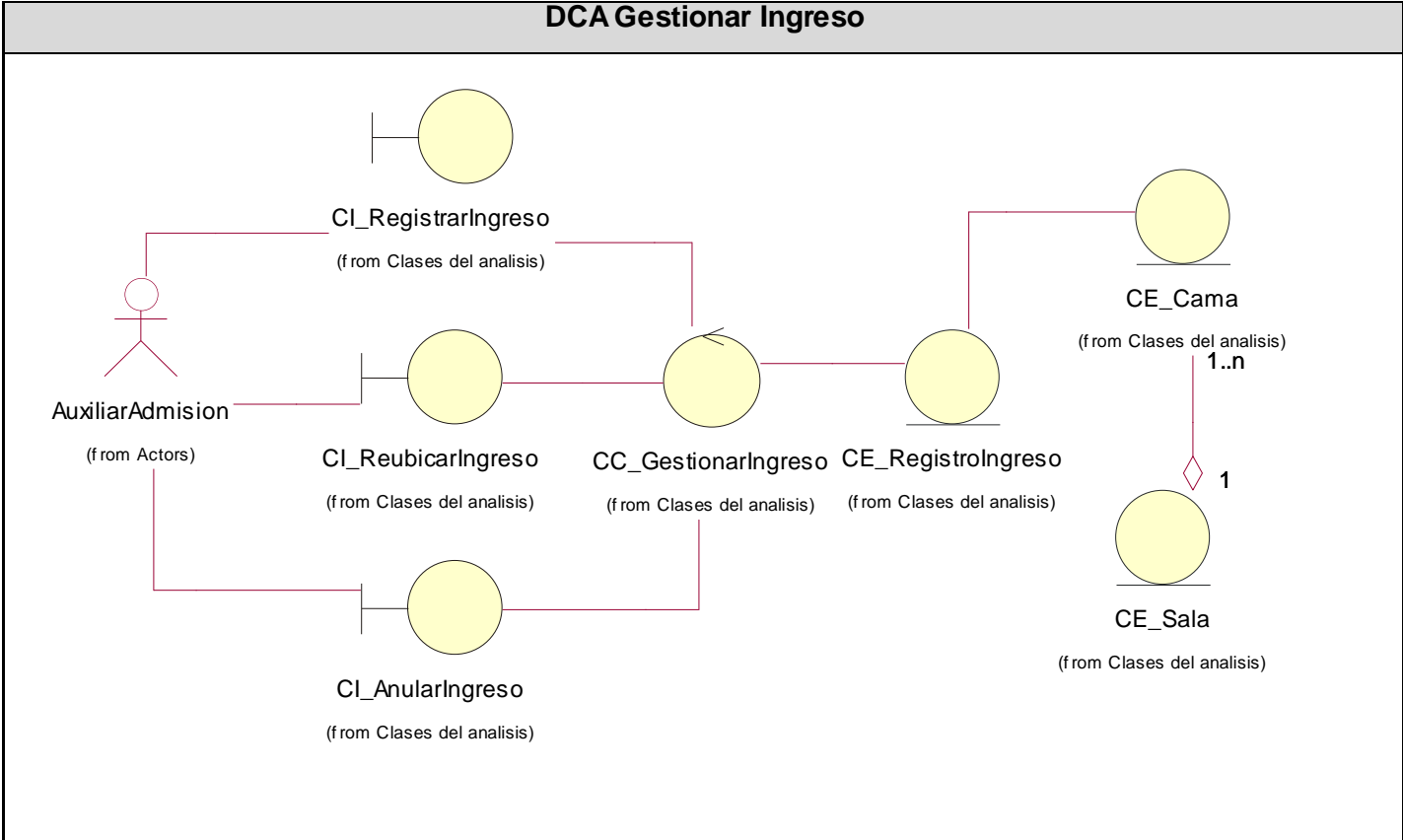
Las clases interfaz son las que se encargan de brindar un entorno gráfico amigable y entendible para que el usuario interactúe con el sistema de manera que modelan la interacción entre el sistema y sus actores.

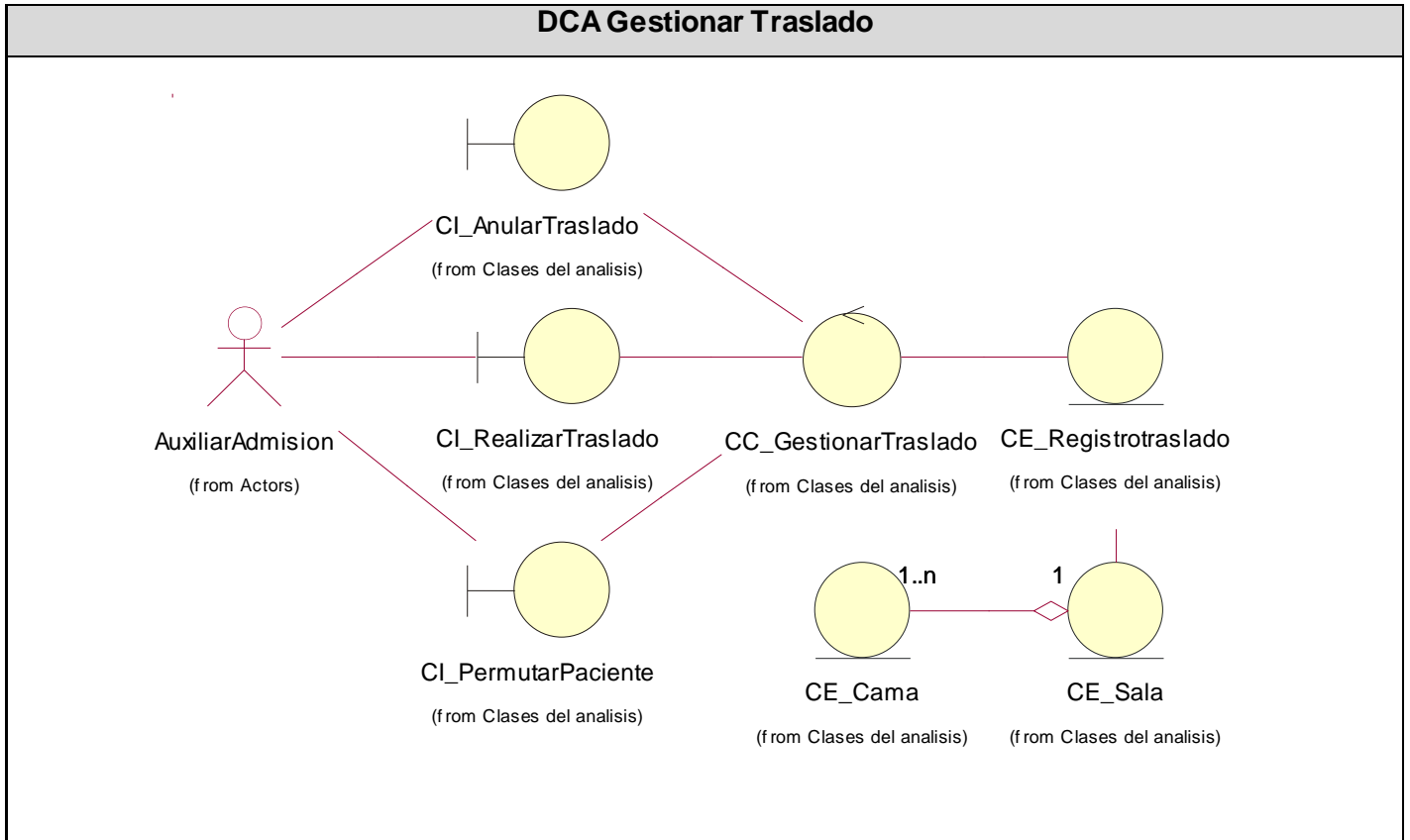
Las clases controladoras se encargan de captar toda la información que es enviada a través de las clases interfaz y guardar esos datos a través de métodos en la *Base de Datos* (BD), coordinando la realización de uno o unos pocos casos de uso así como las actividades de los objetos que implementan la funcionalidad del caso de uso.

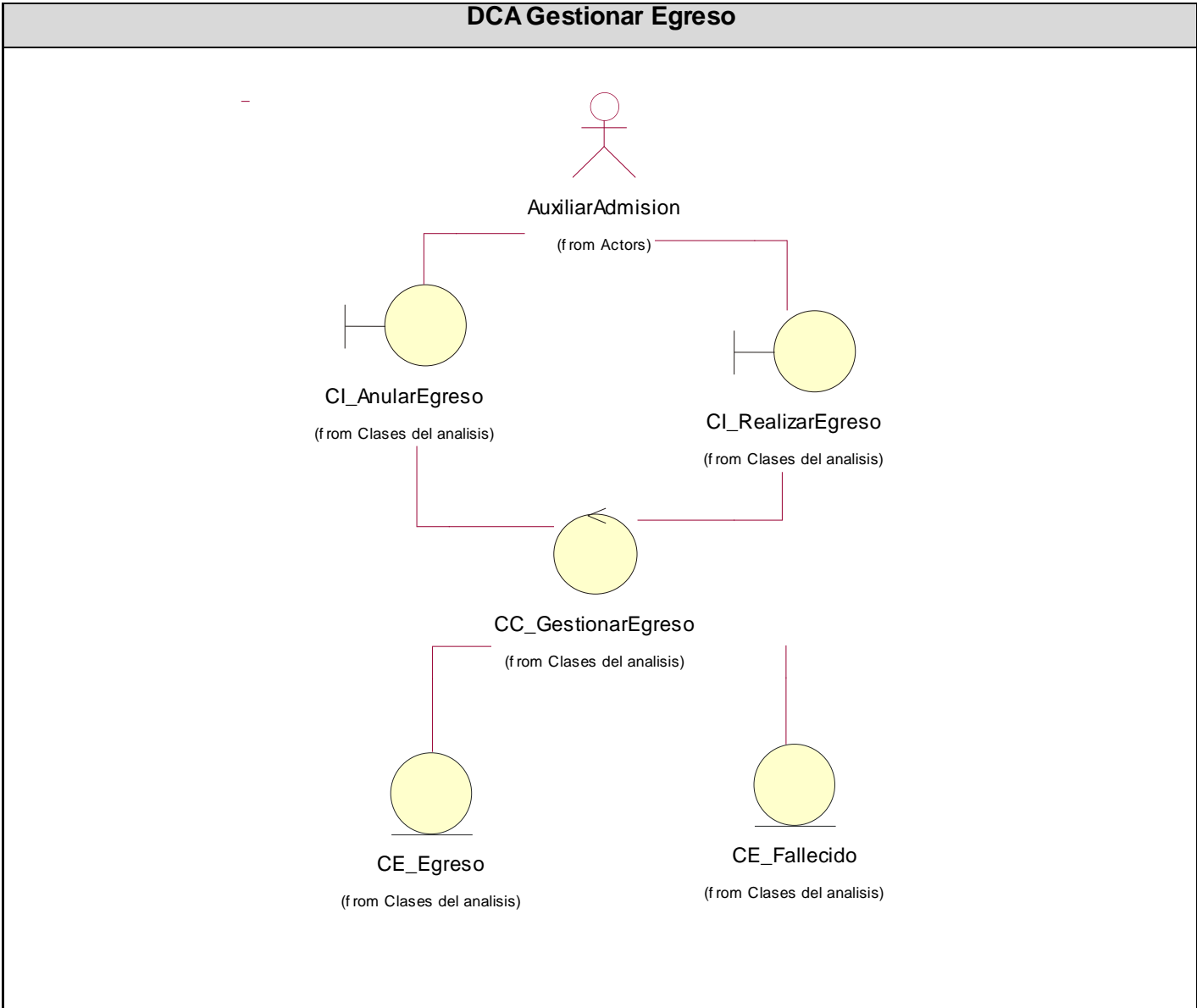
Las clases entidades vienen como resultado de las entidades definidas en el negocio estudiado y definido, aunque puede suceder que algunas de estas desaparezcan y surjan nuevas entidades, generalmente modelan información que posee larga vida y que es a menudo persistente, estas clases son las futuras tablas de la BD.

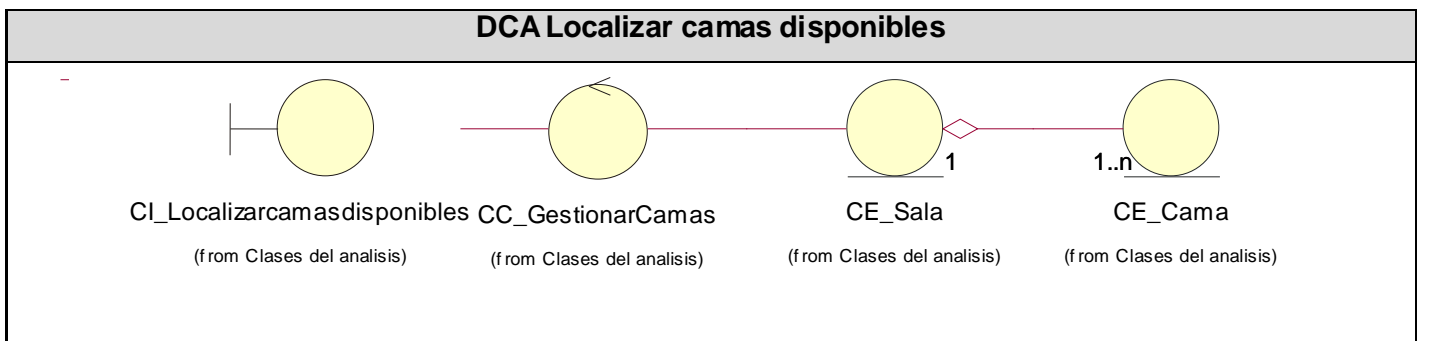
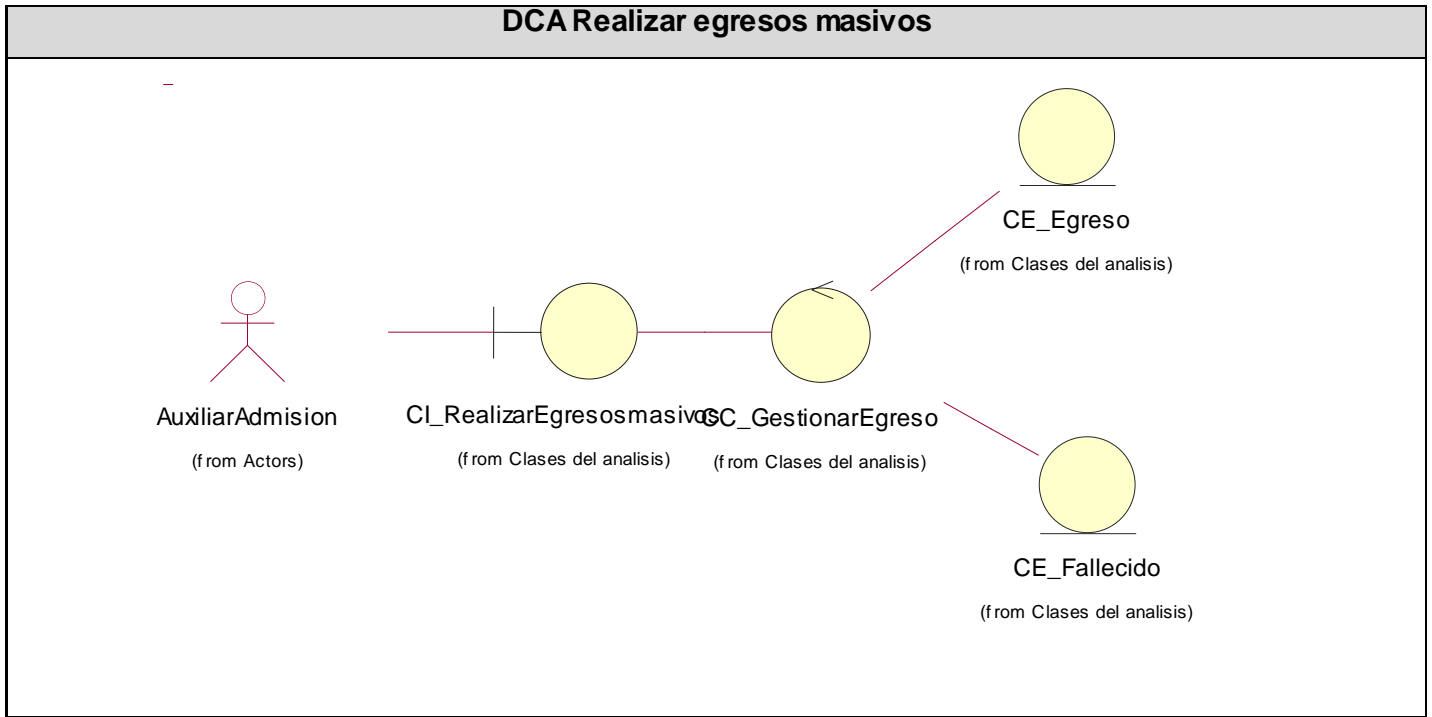
Los diagramas de clases de análisis además contienen relaciones que evidencian la navegabilidad y la dependencia entre clases, de esta forma se identifican relaciones de asociación, generalización, especialización, entre otras.

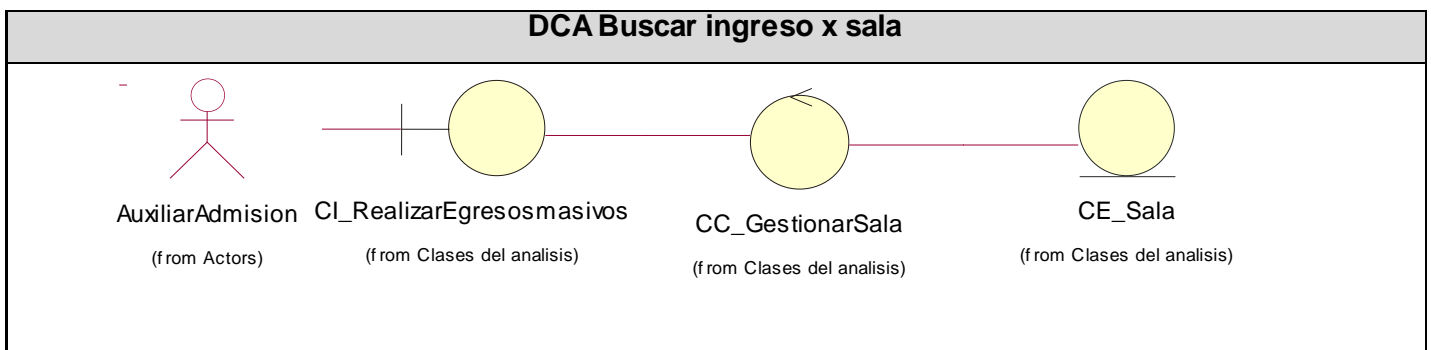
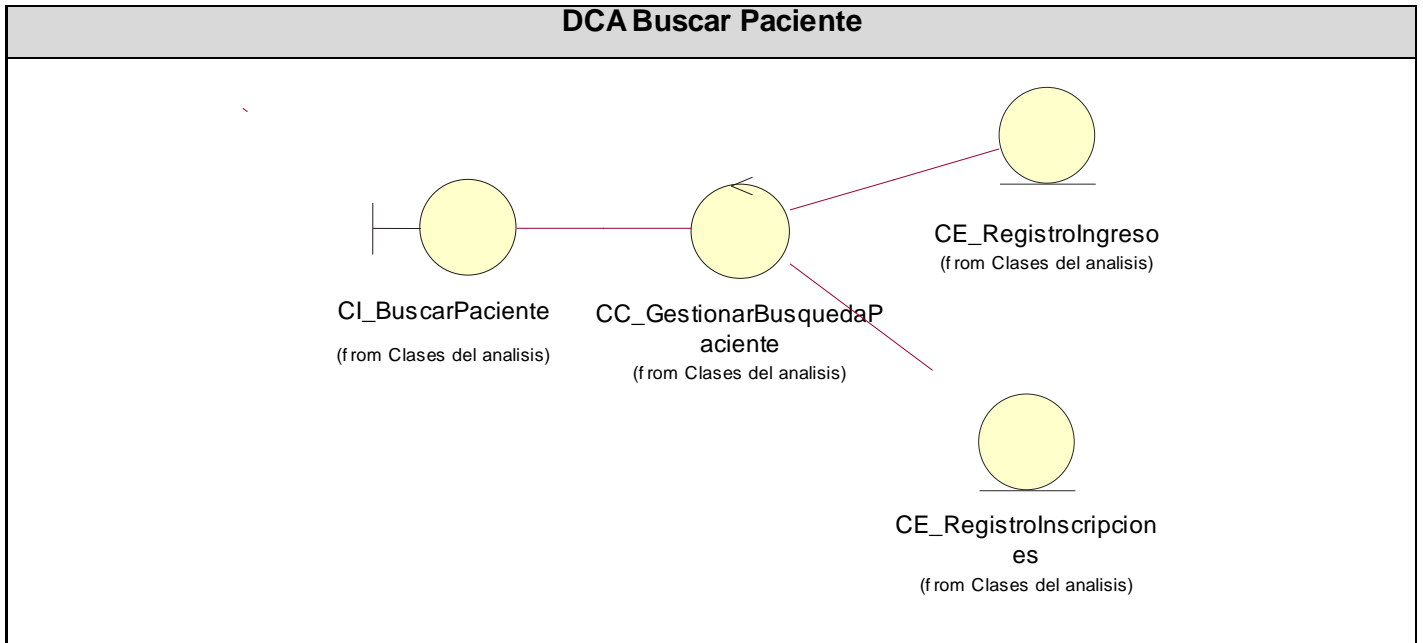












3.2 Diagramas de clases de diseño

Para la elaboración del diseño, generalmente se utilizan un grupo de patrones, o modelos para lograr objetivos específicos, dentro de los más conocidos están los GRASP (General Responsibility Assignment Software Patterns) que en su traducción al español significa patrones generales de software para asignar responsabilidades y los GOF (Gang of four) abreviatura que proviene del inglés y que significa pandilla de los cuatro, llamado de esta forma debido a cuatro autores que definieron estos patrones.

De manera general los patrones constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño *orientado a objetos*, basados en la experiencia y que se ha demostrado que funcionan. [23]

Los patrones GRASP se utilizan con el objetivo de asignar responsabilidades a las diferentes clases que se definen en el diseño. Dentro de este grupo se identifican cinco patrones muy utilizados: experto, creador, alta cohesión, bajo acoplamiento y el controlador. En los diagramas de clases que elaboramos no se aprecian directamente el uso de estos patrones, debido a que no hay necesidad de crear nuevas clases, simplemente se aplican a las clases definidas en el diseño, distribuyendo responsabilidades entre las mismas de forma tal que no existan muchas relaciones, que no se sobrecargue de métodos a una clase en específico pudiendo acomodarlos en otras, entre otras mejoras que brinda el uso de este grupo de patrones.

Los patrones GOF generalmente se evidencian en clases que son creadas debido al uso de un patrón en específico. Existe un grupo de patrones de este tipo definidos para el diseño de clases y con el propósito de crear una arquitectura robusta para el sistema a desarrollar. Del gran número de patrones propuestos por la pandilla de los cuatro o simplemente GOF, se propone el uso del patrón Fábrica Abstracta. Este tiene como principal función crear objetos sin importar en que momento y a que clase pertenecen y por lo general se implementa junto al patrón método de fabricación.

Estos son patrones de creación y en el caso que nos embarga al ser utilizados dentro del *framework* de .NET se modelan dentro de un paquete al que acceden todas las clases controladoras del negocio a través de un repositorio que gestiona los procesos fundamentales y que al hacer uso del subsistema fábrica estaría llamando a todas las interfaces de cada una de las entidades del negocio. Estas interfaces contienen métodos abstractos y se encargan de realizar las operaciones básicas sobre los datos, insertar, modificar, eliminar y seleccionar.

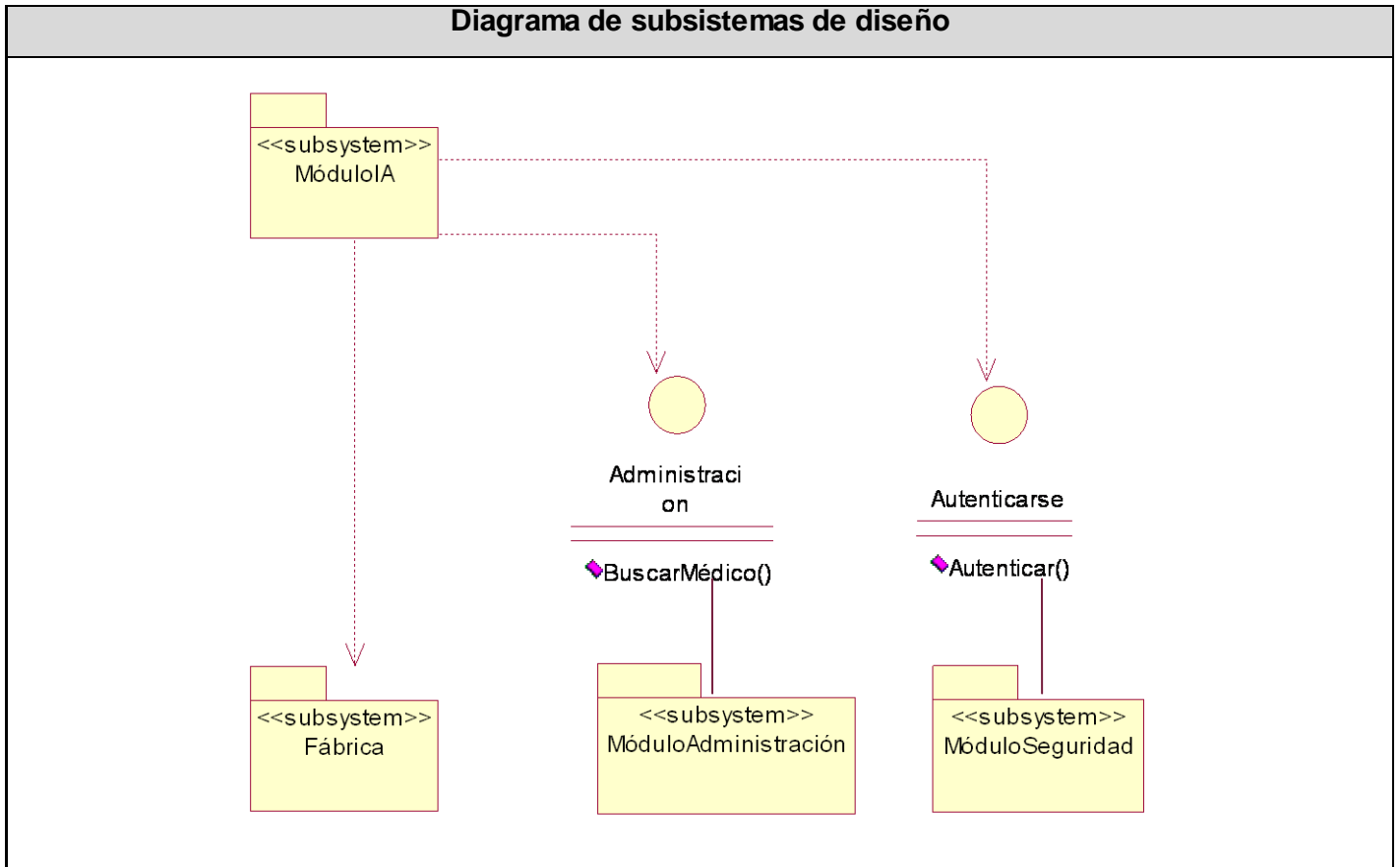
Para el desarrollo del módulo de inscripción admisión, hemos creado un paquete que contiene las realizaciones de cada uno de los casos de uso definidos en el flujo de trabajo de requerimientos, como arquitectónicamente significativos.

Estas realizaciones contienen los diagramas de clases del diseño que evidencian las clases que se definen para el desarrollo del sistema, así como sus correspondientes diagramas de interacción que en este caso se ha decidido optar por la elaboración de diagramas de secuencia que manejan objetos de las

diferentes clases y muestran una secuencia lógica de llamadas entre los objetos, brindándole a los futuros desarrolladores una imagen clara de cómo deben implementar el software propuesto.

Se utilizan además otros subsistemas para el diseño que aunque no son definidos en este módulo, si son utilizados y es necesario especificar de alguna forma cómo se relacionarían los mismos con el subsistema de inscripción admisión. Para esto se ha elaborado un diagrama que muestra la colaboración entre los subsistemas; este contiene el módulo en cuestión, que hace uso del subsistema fábrica que contiene un grupo de *interfaces* o *clases abstractas* con sus métodos que se encargan de crear los productos abstractos y concretos para poder efectuar las operaciones necesarias sobre las clases entidades.

Además del paquete fábrica se encuentra el subsistema de administración del cual hacen uso algunas de las realizaciones contenidas en el módulo de inscripción admisión, y que a través de una *interfaz* que contiene el método *Buscar Médico* posibilita esta funcionalidad a la hora de realizar una inscripción, ingreso o egreso. De la misma forma se encuentra el subsistema de seguridad que a través de una *interfaz* Autenticarse permite al paquete de inscripción admisión gestionar la autenticación de los actores, sin necesidad de tener que definir en el diseño de las clases, un escenario o diagrama para estas cuestiones.

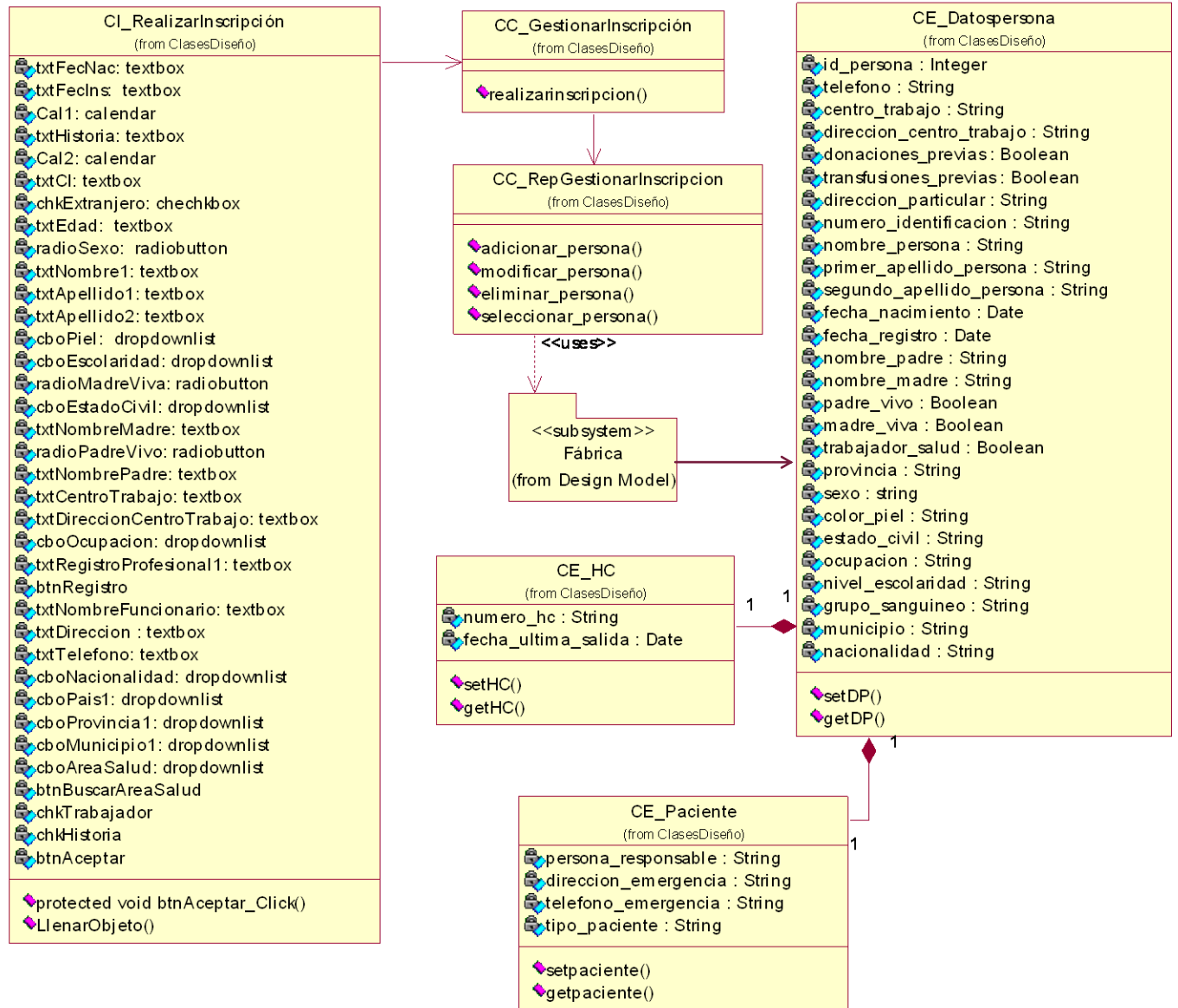


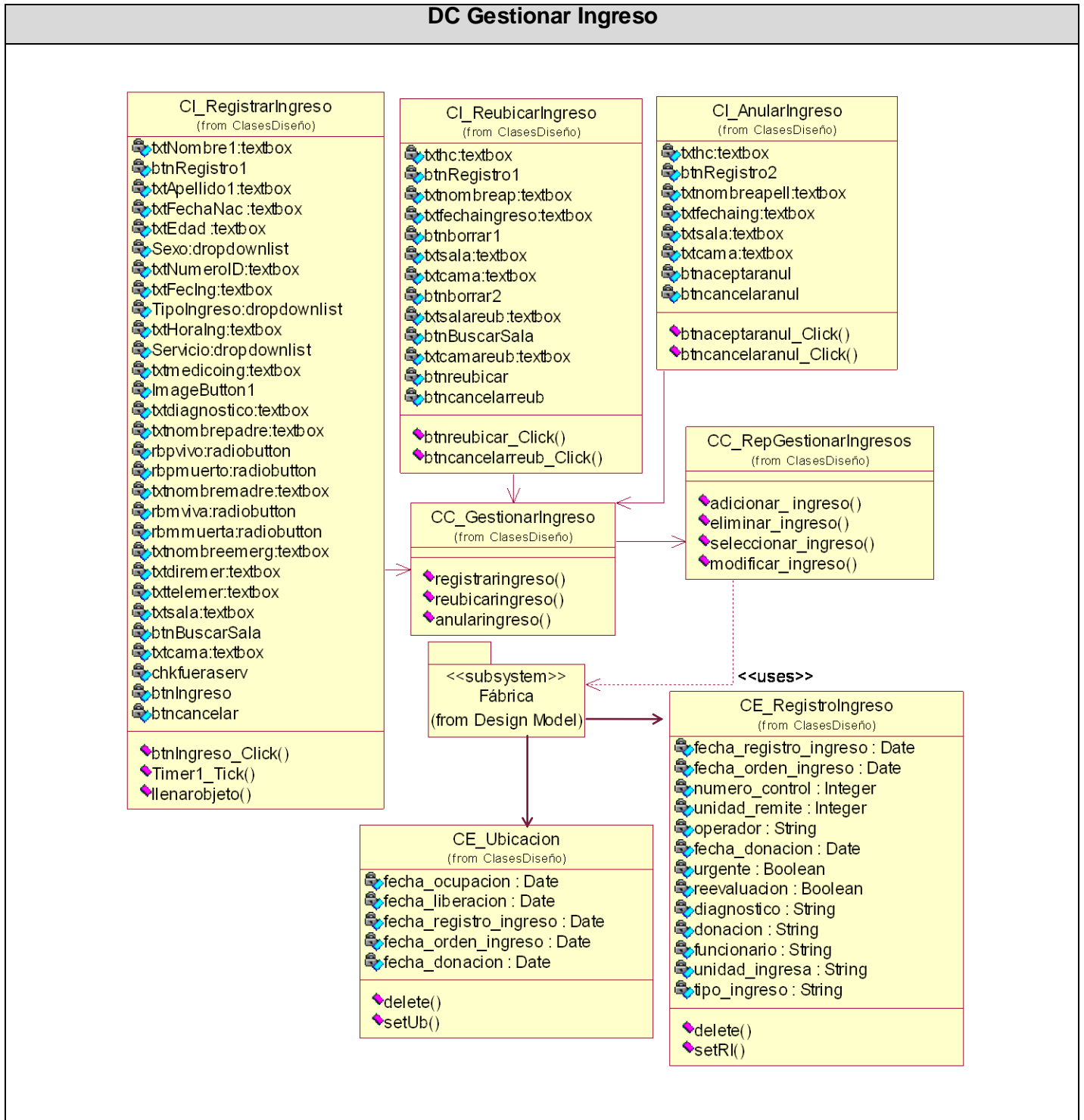
Dentro del paquete referente al módulo de inscripción admisión se encuentran las realizaciones de los casos de uso compuestas por los diagramas de clases conjuntamente con cada uno de los diagramas de secuencia de las diferentes secciones o escenarios.

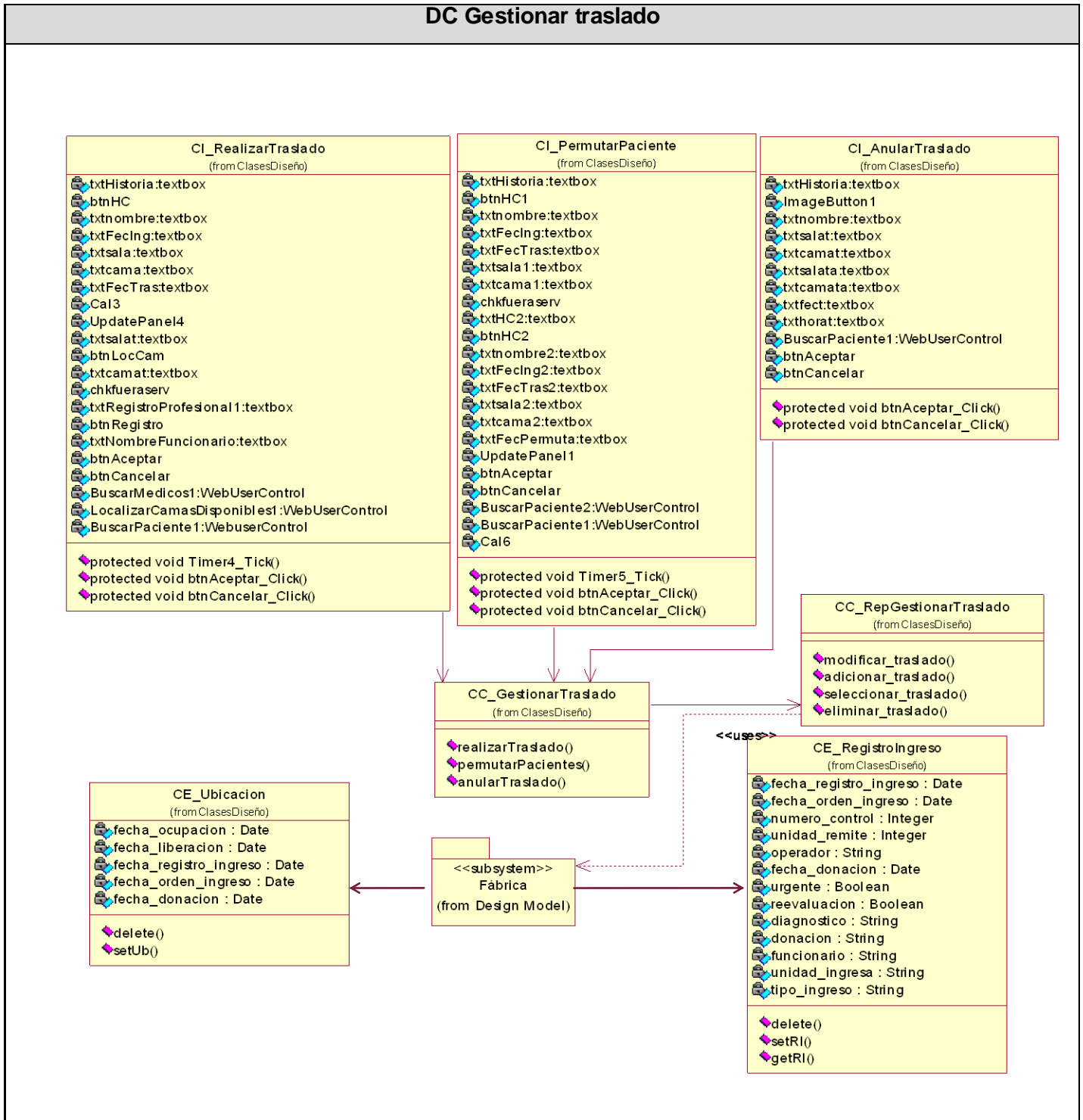
En cada uno de los diagramas que se muestran se incluye el paquete fábrica, necesario e indispensable para poder acceder a través de alguna de las interfaces del mismo a las entidades definidas en el diseño.

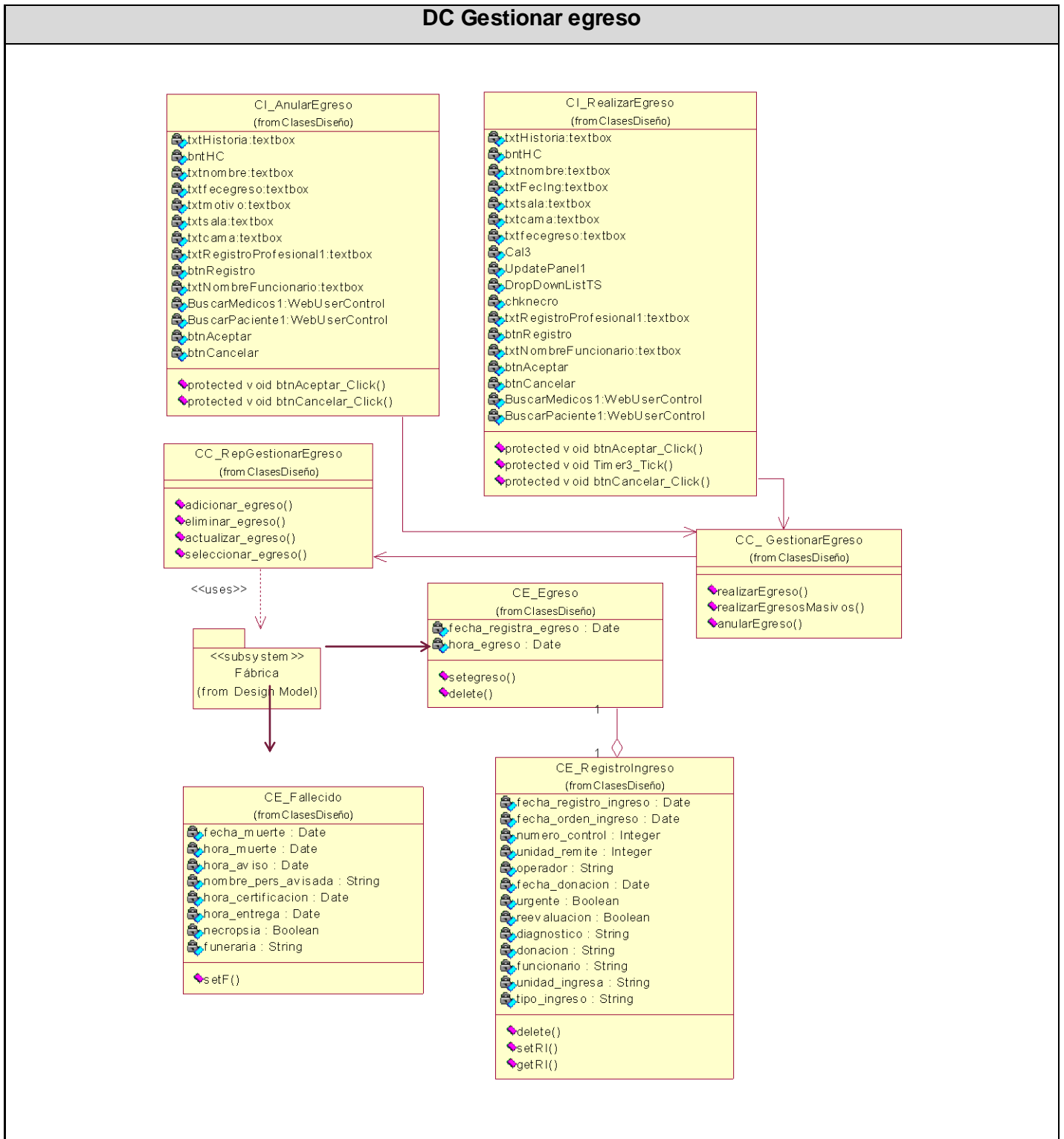
Cada caso de uso contiene un repositorio que actúa como una clase controladora y que es el encargado de seleccionar la interface exacta que va a utilizar para implementar cada proceso descrito.

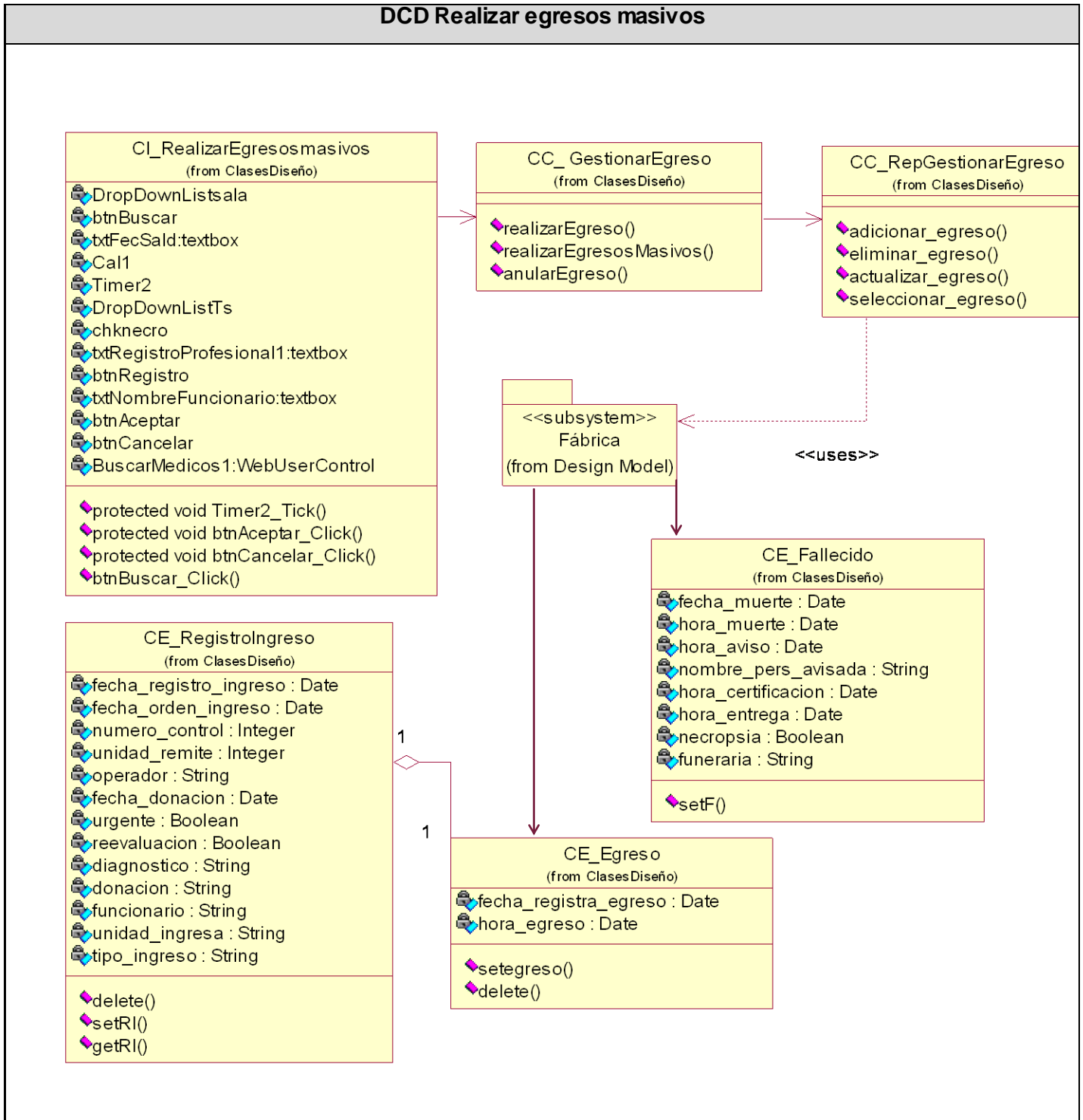
DCD Realizar inscripción

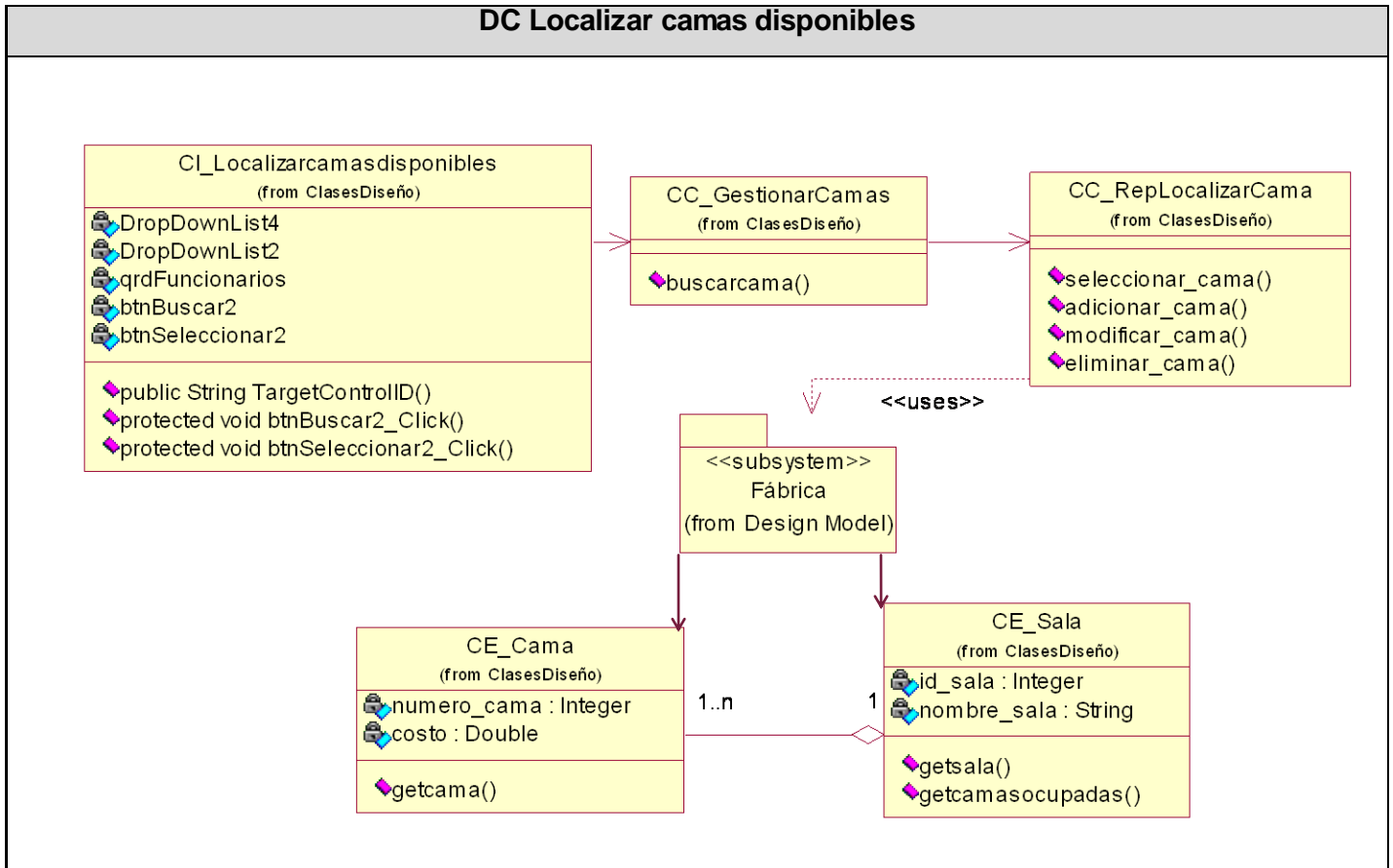


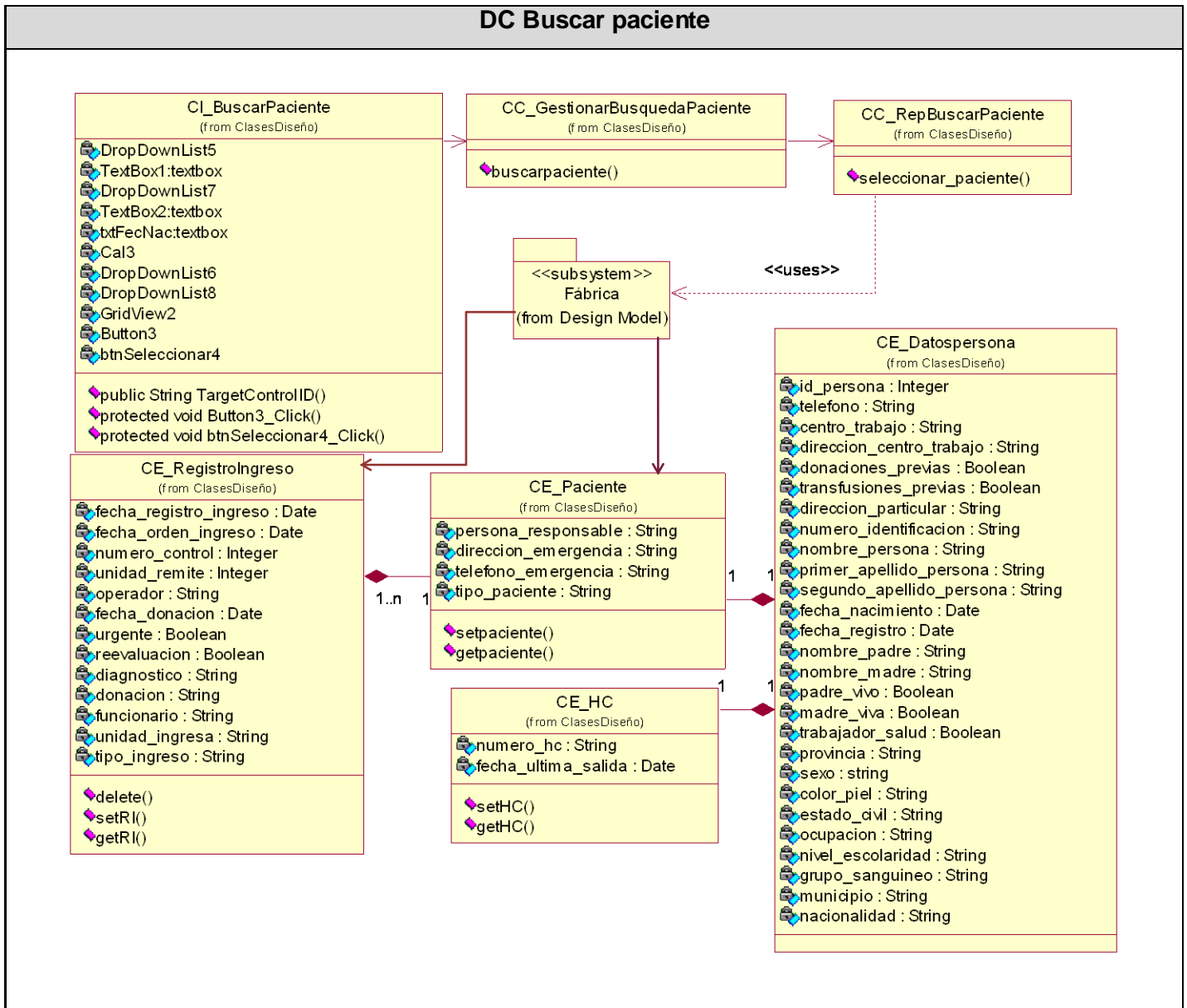


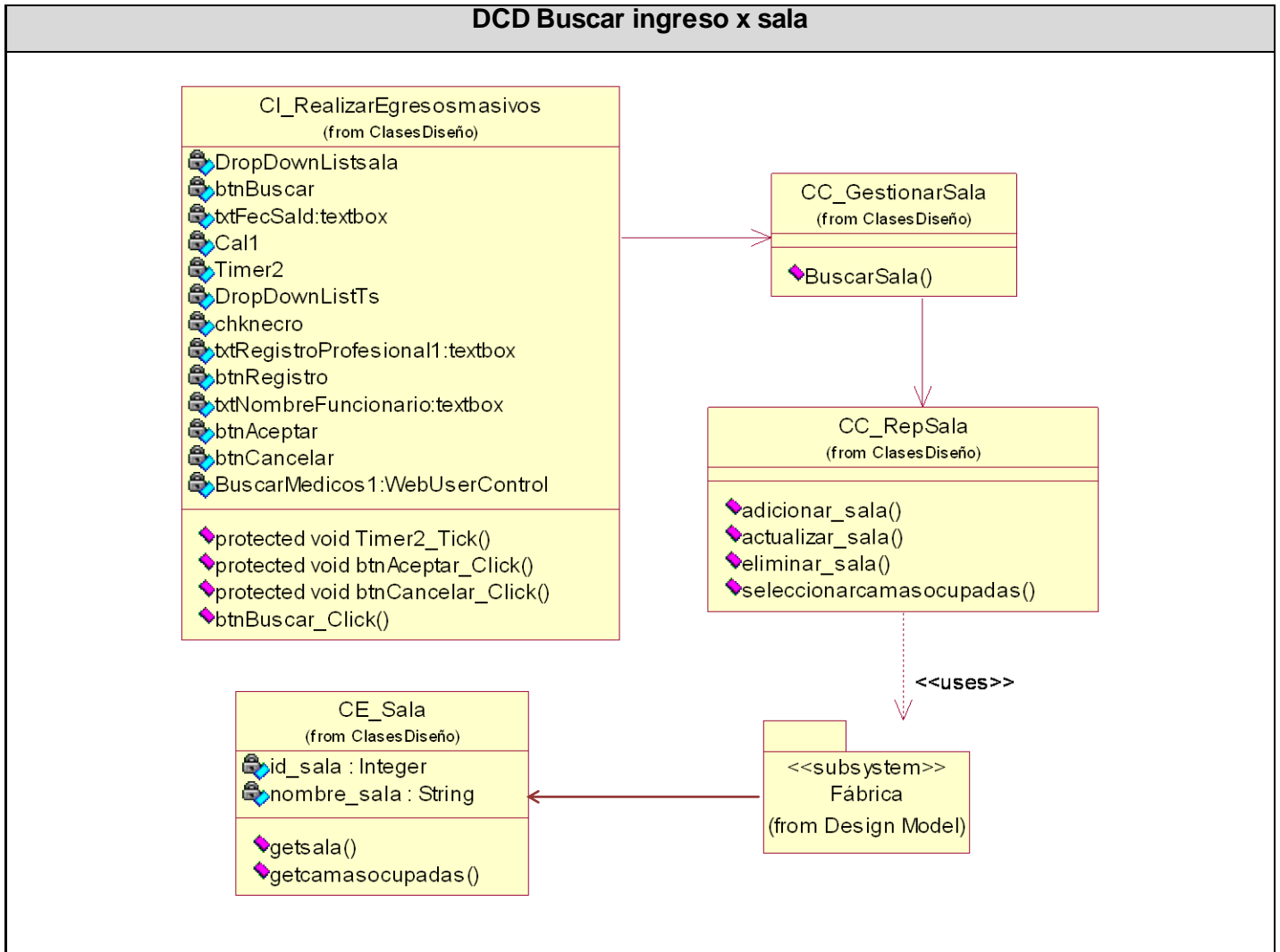




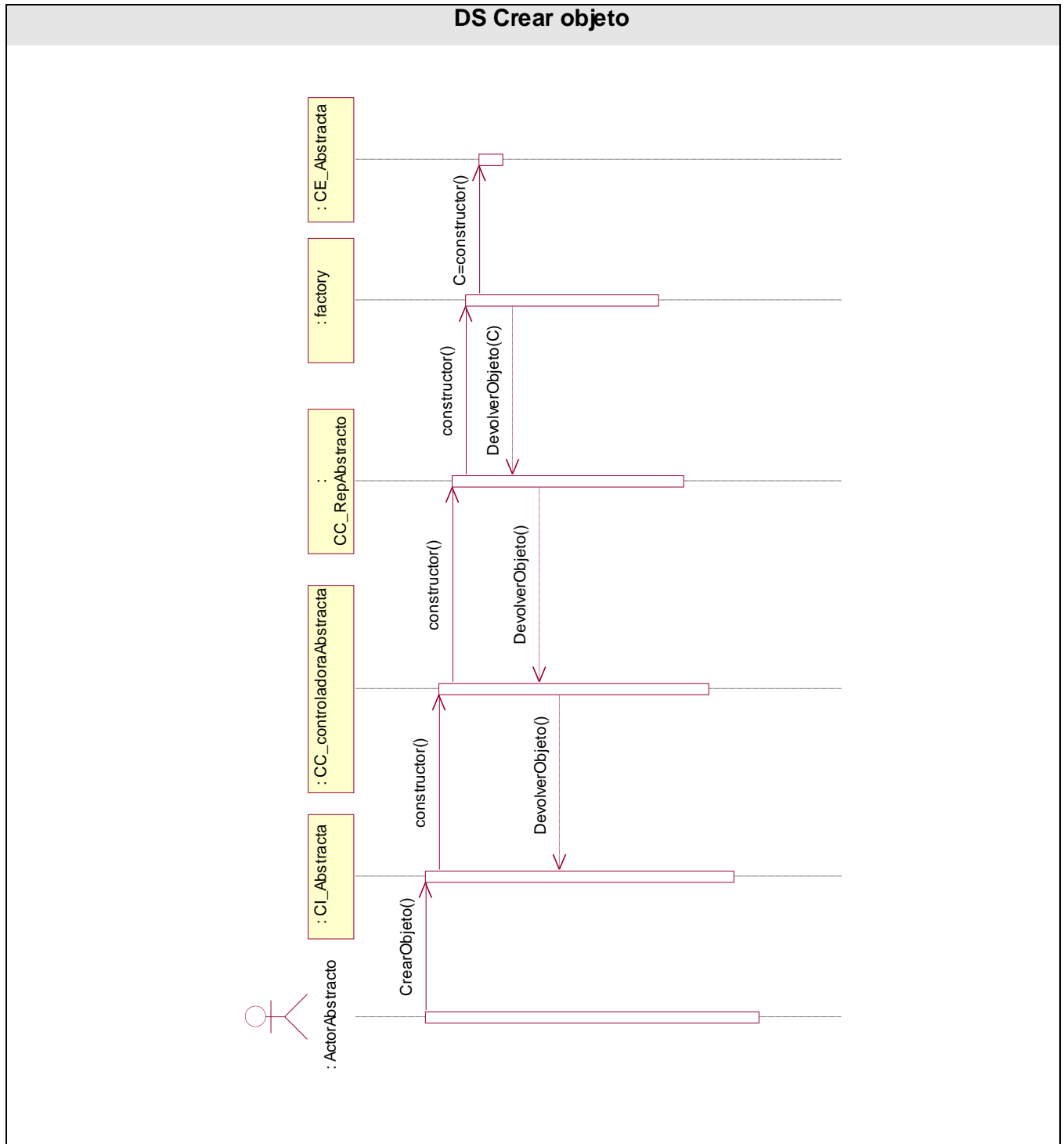


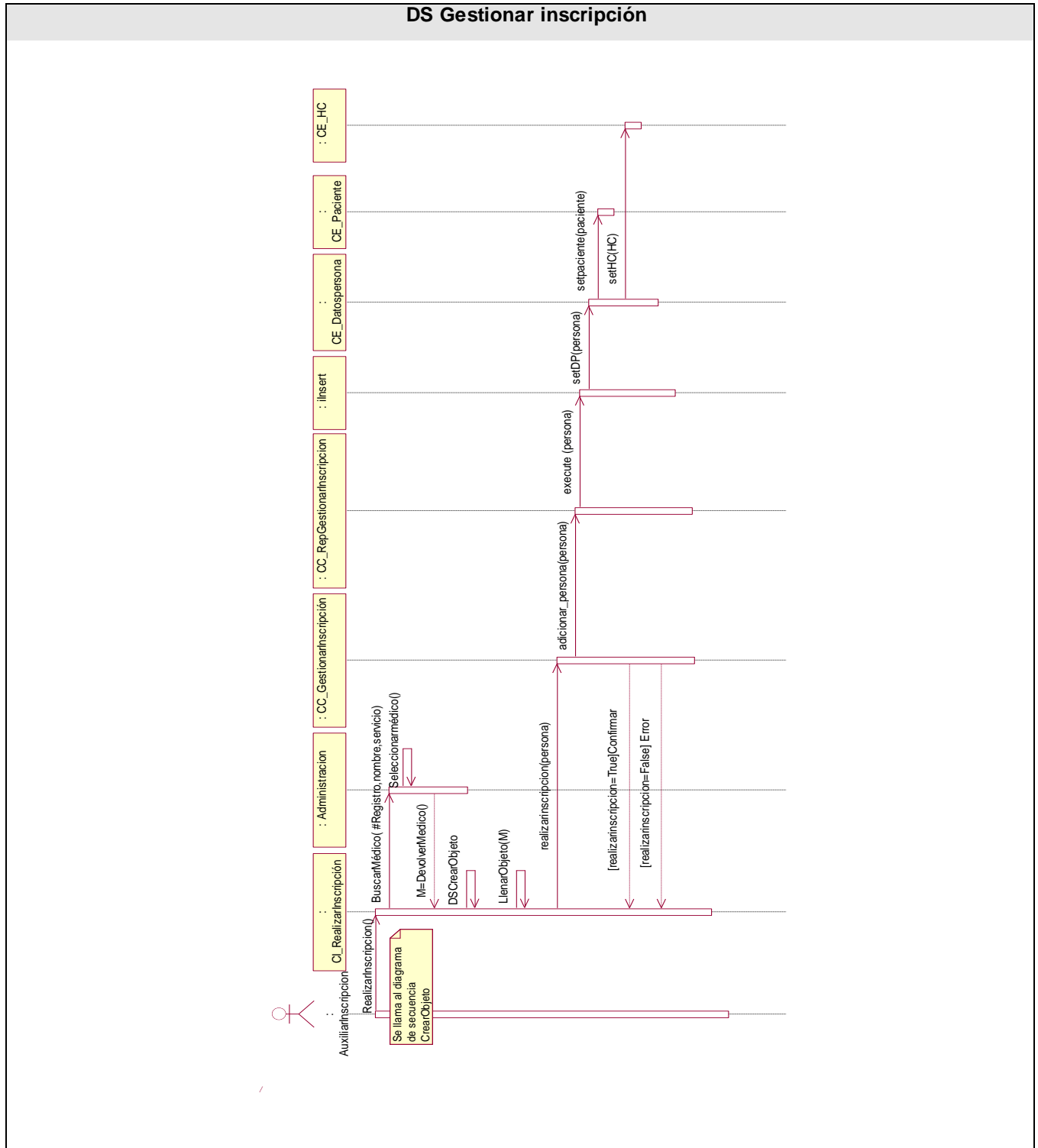


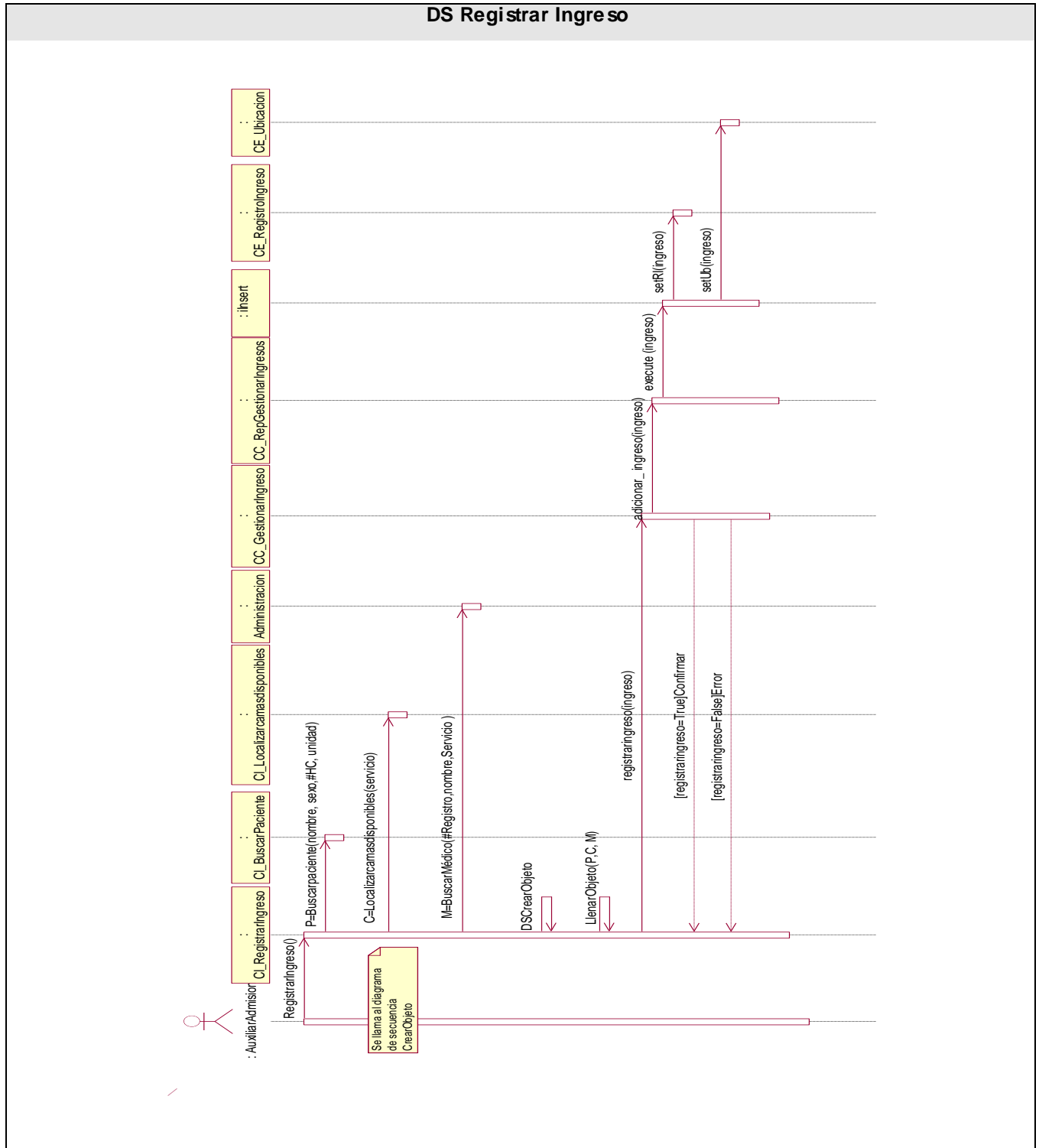


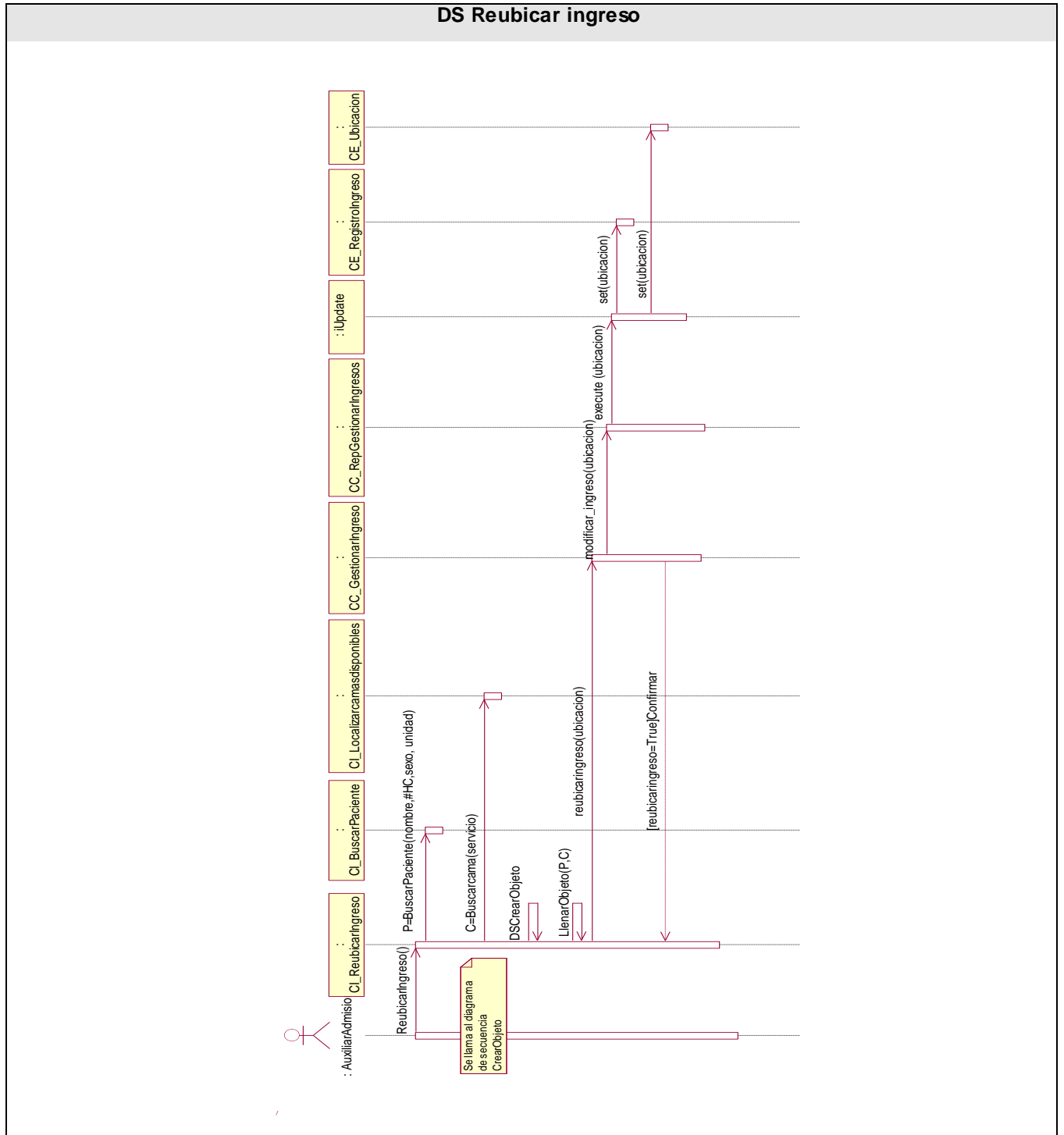


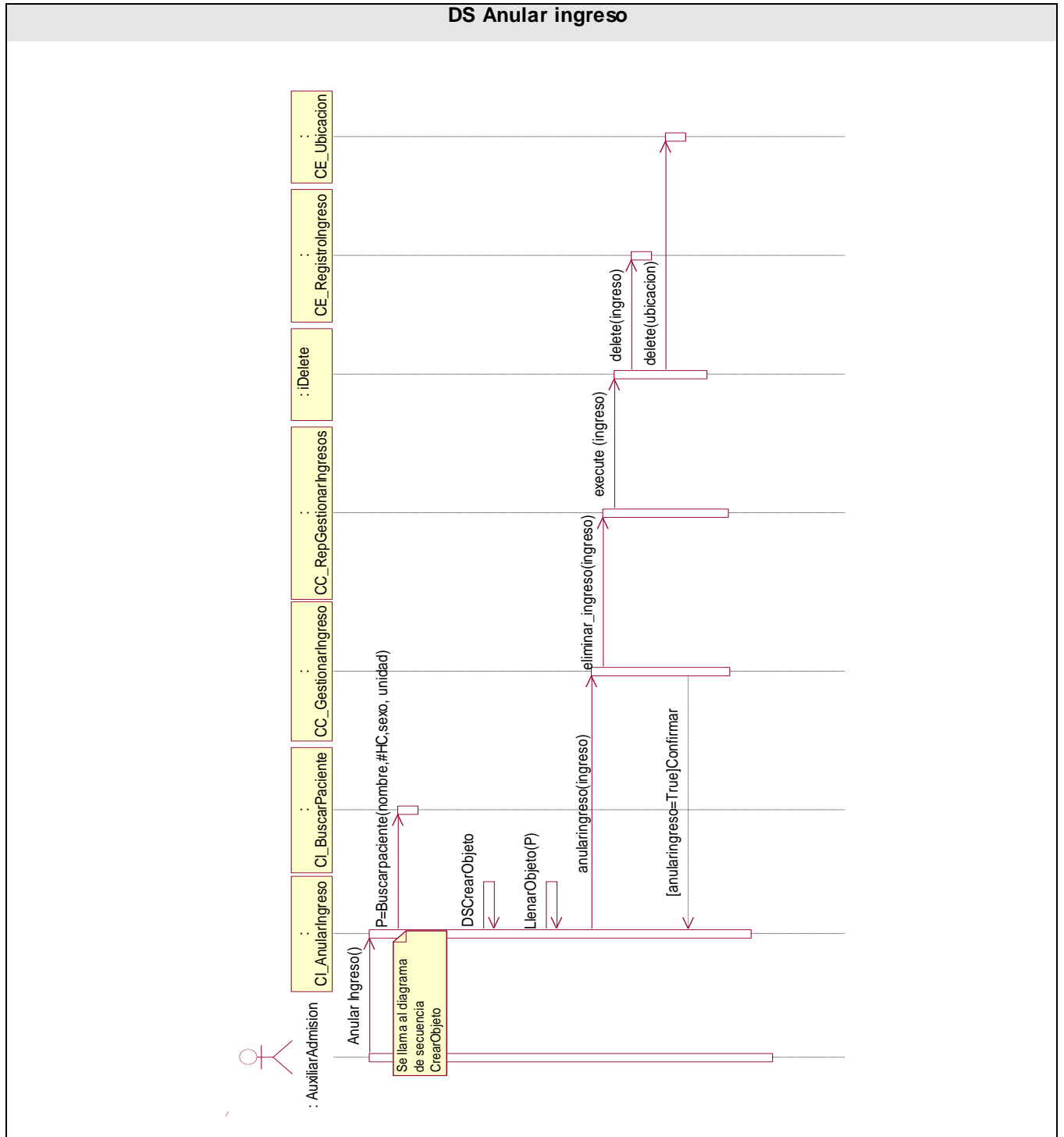
3.3 Diagramas de interacción

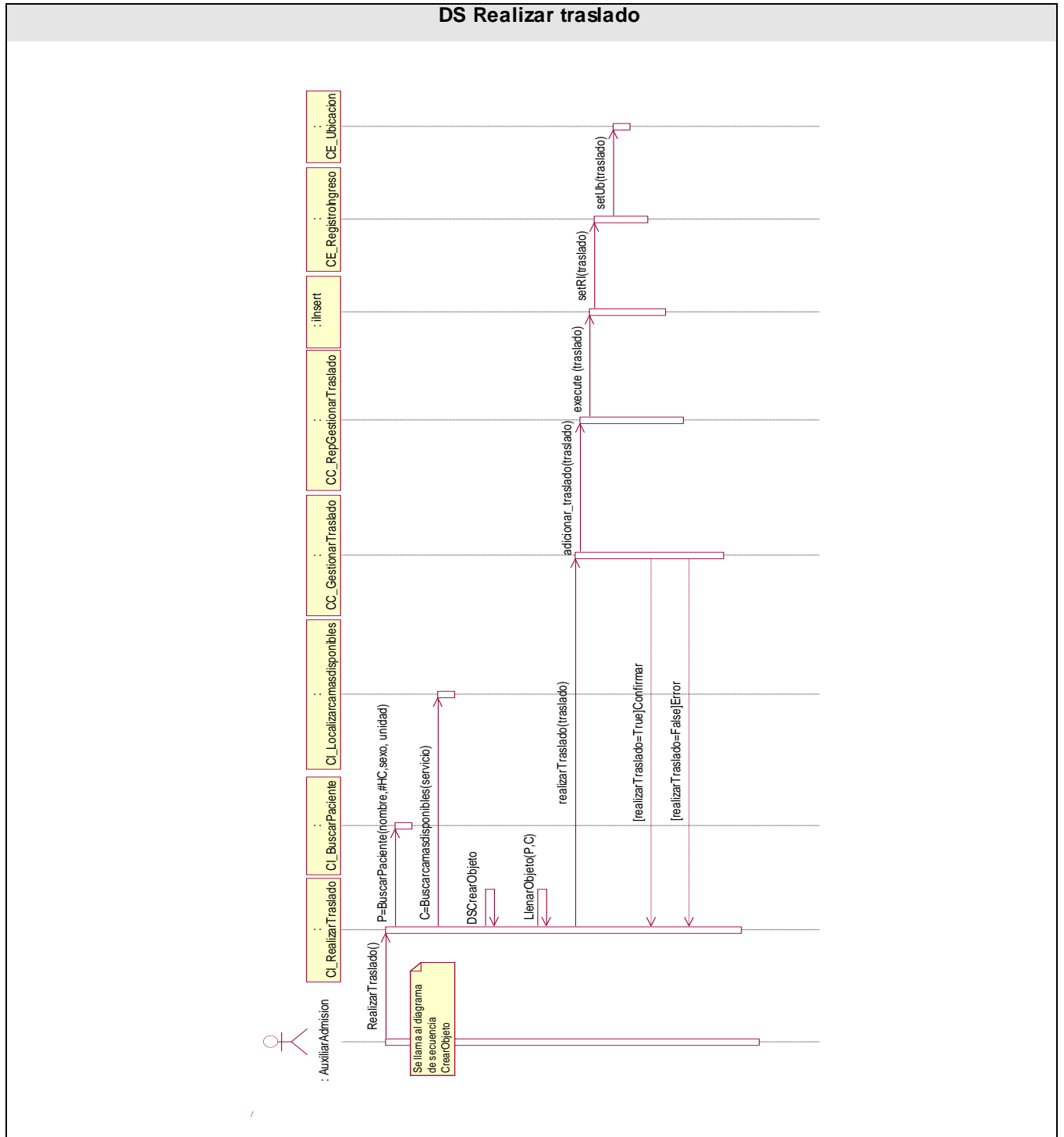


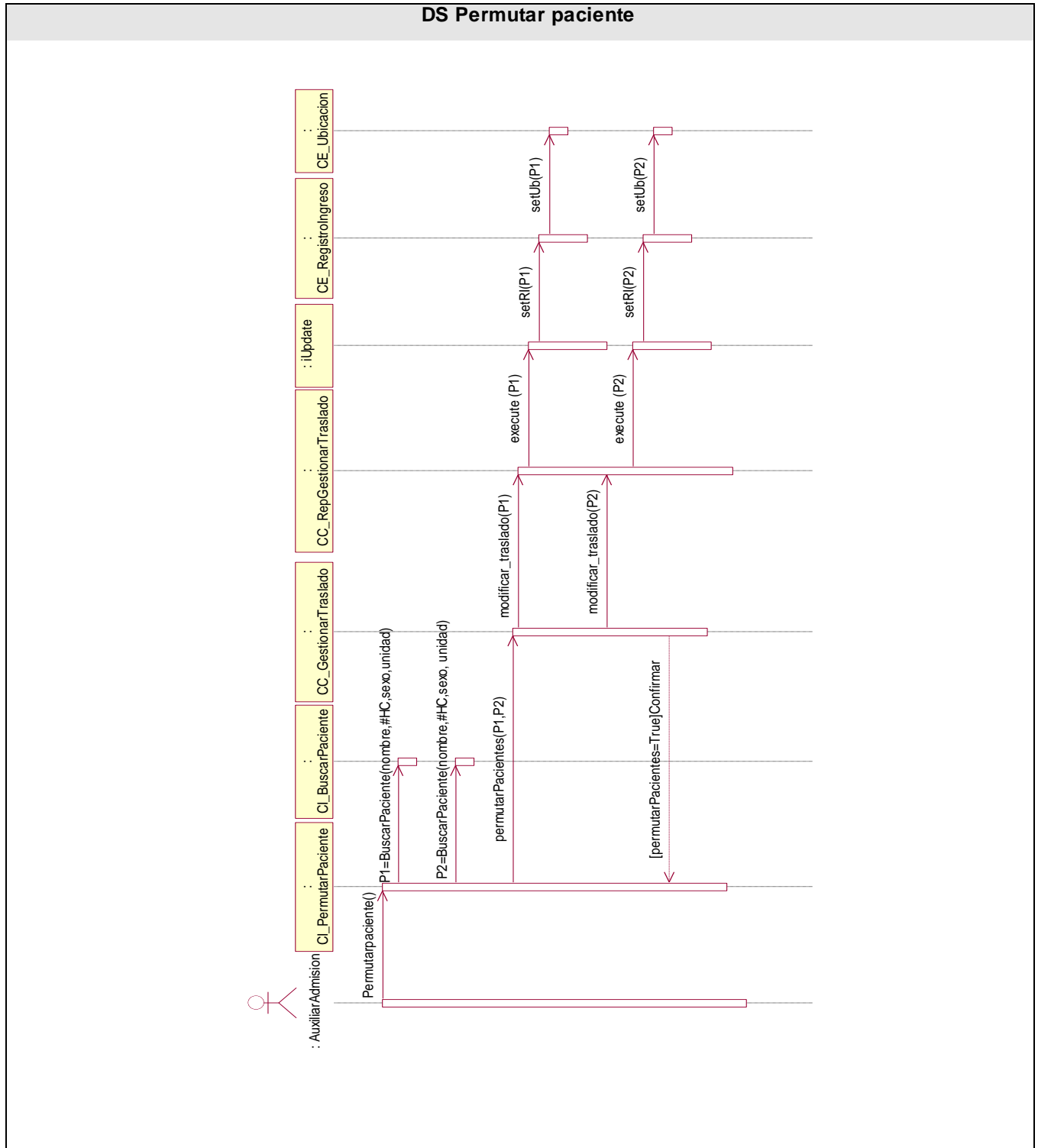


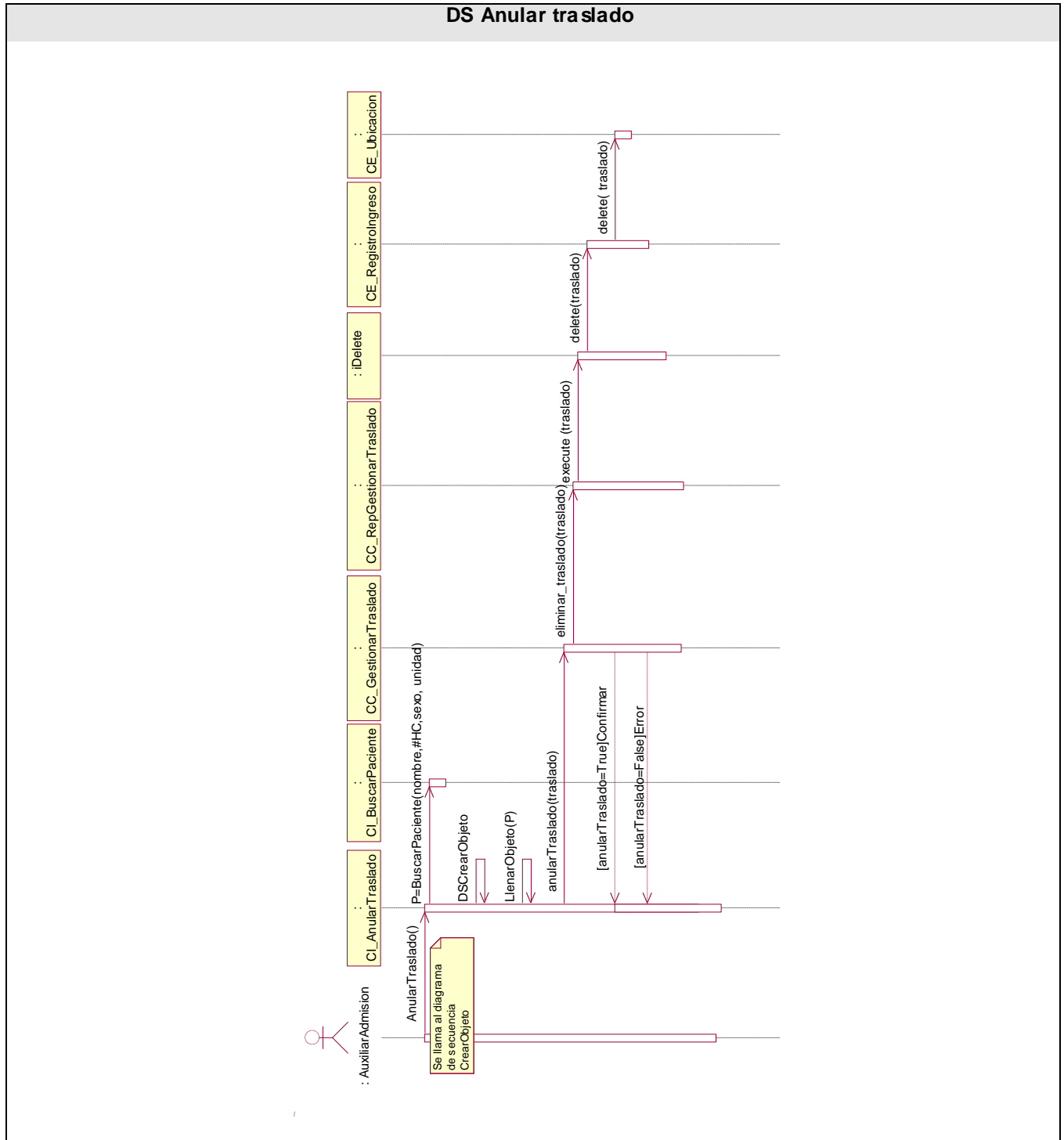


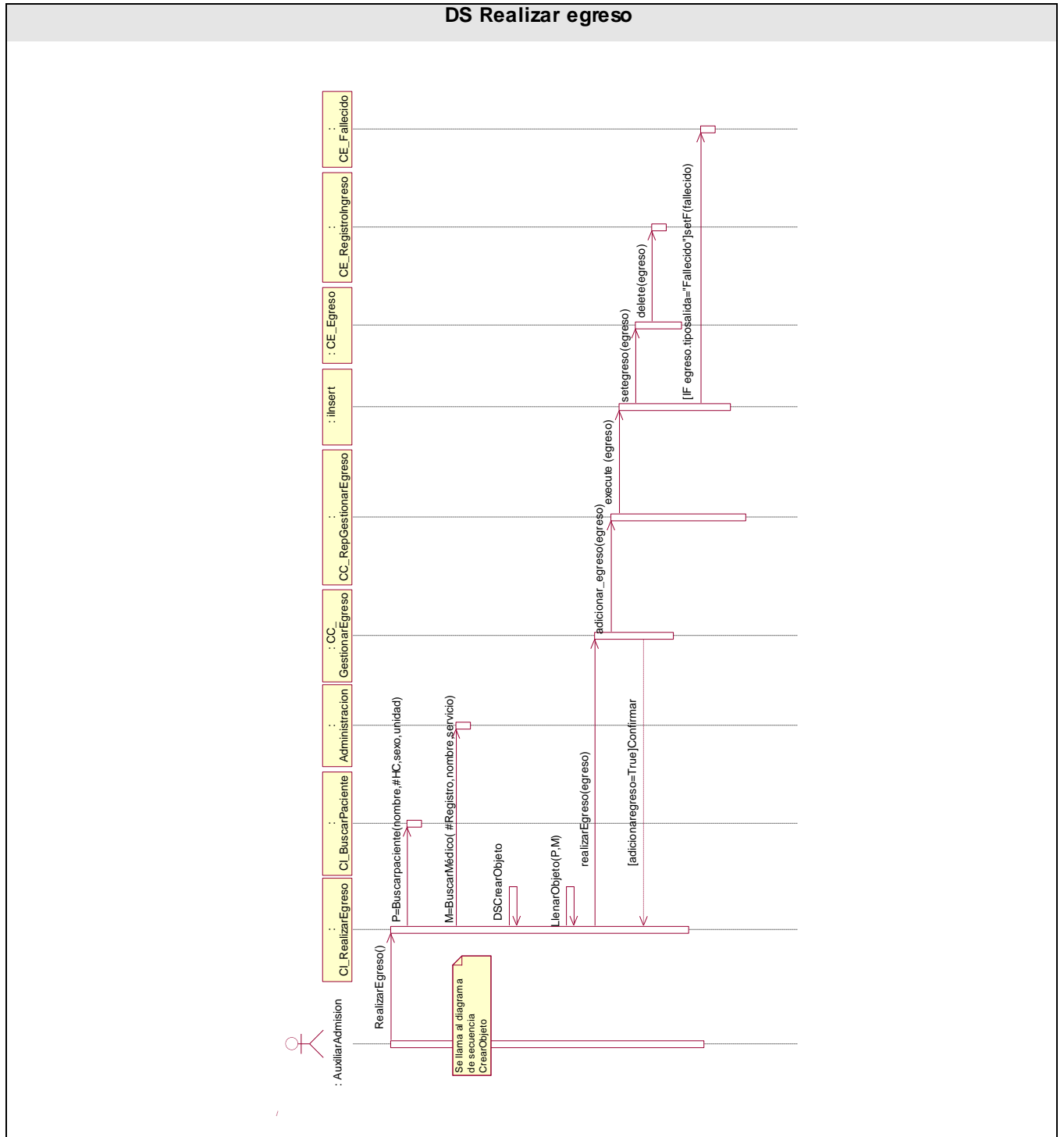


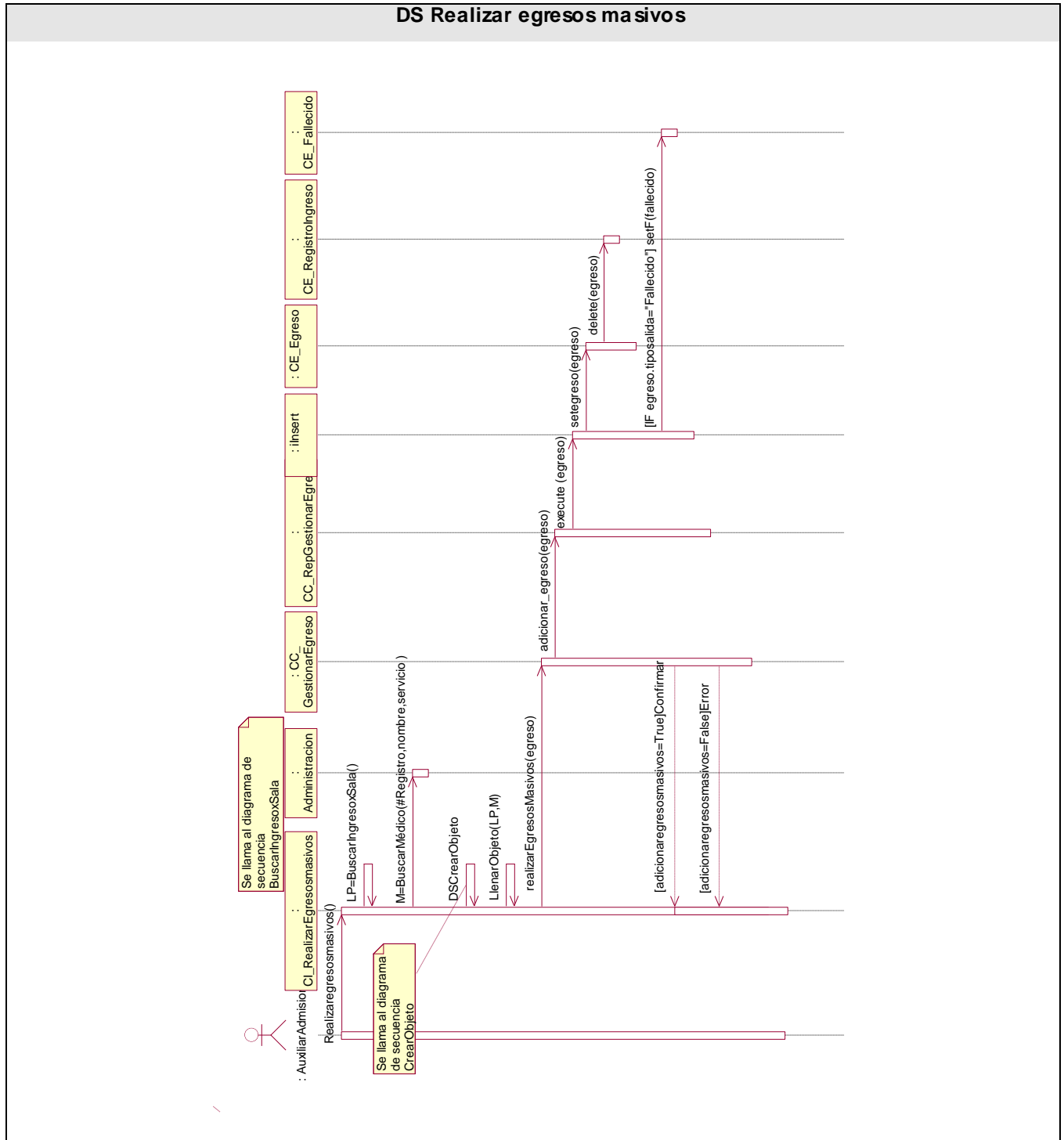


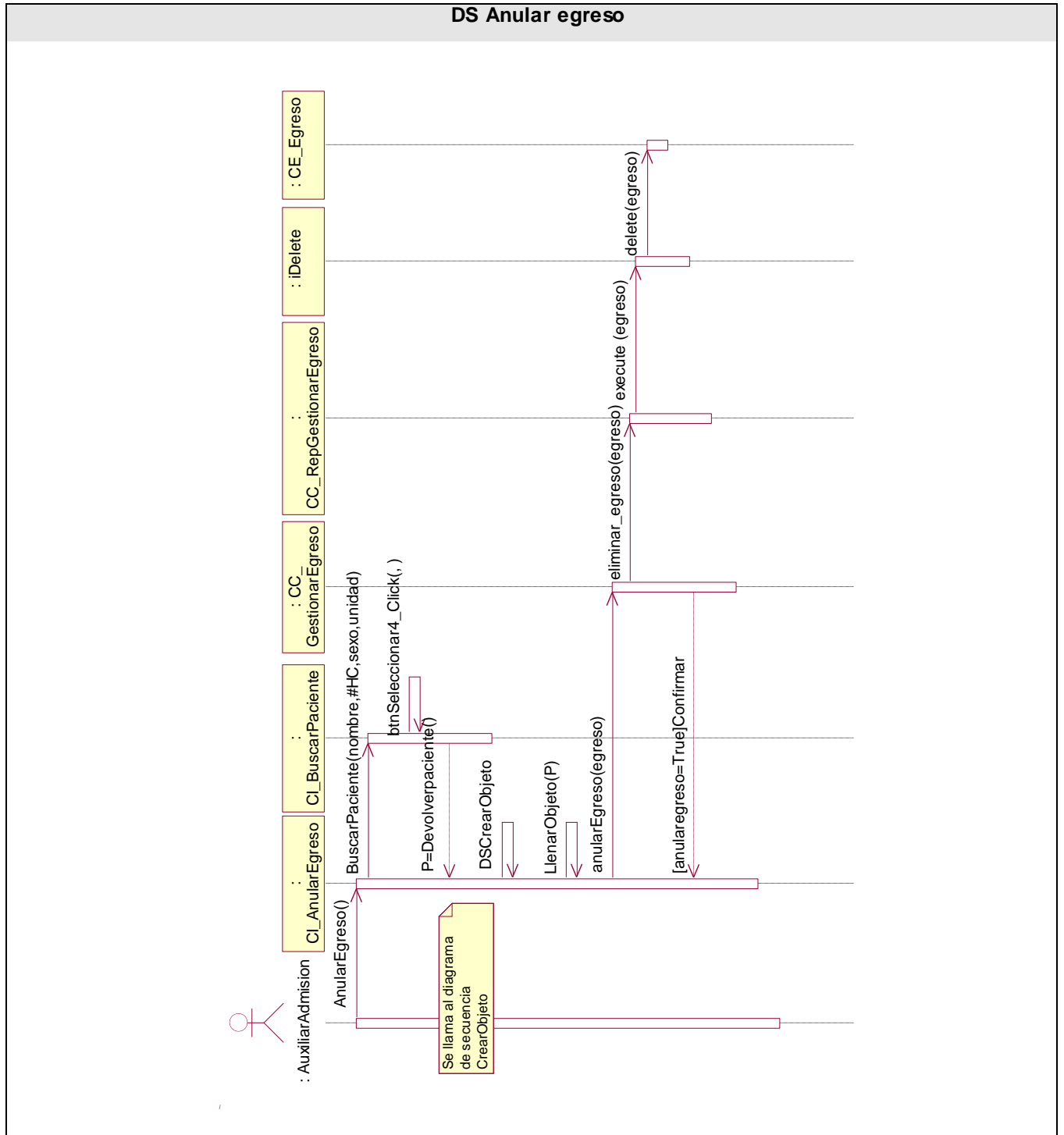


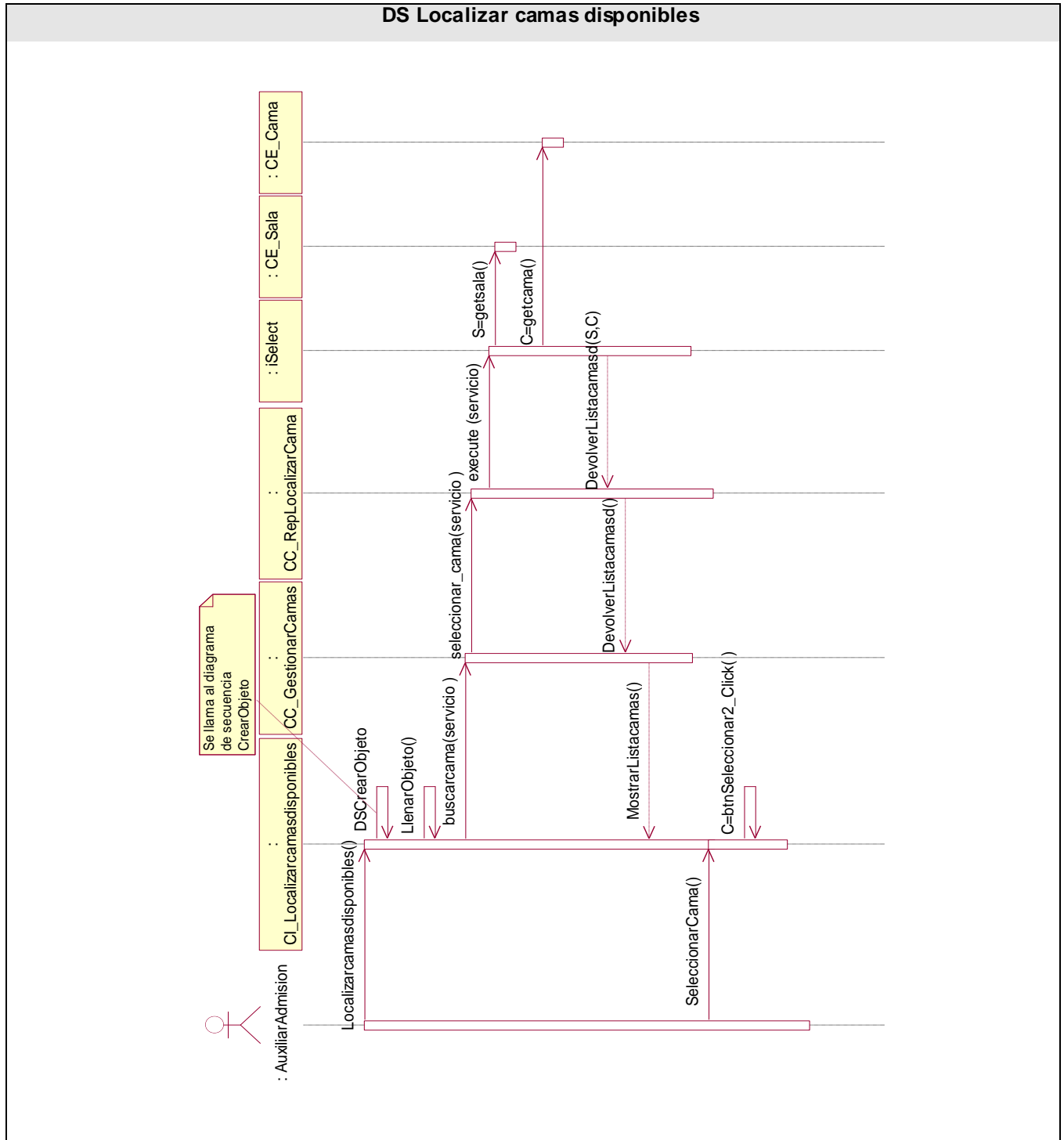


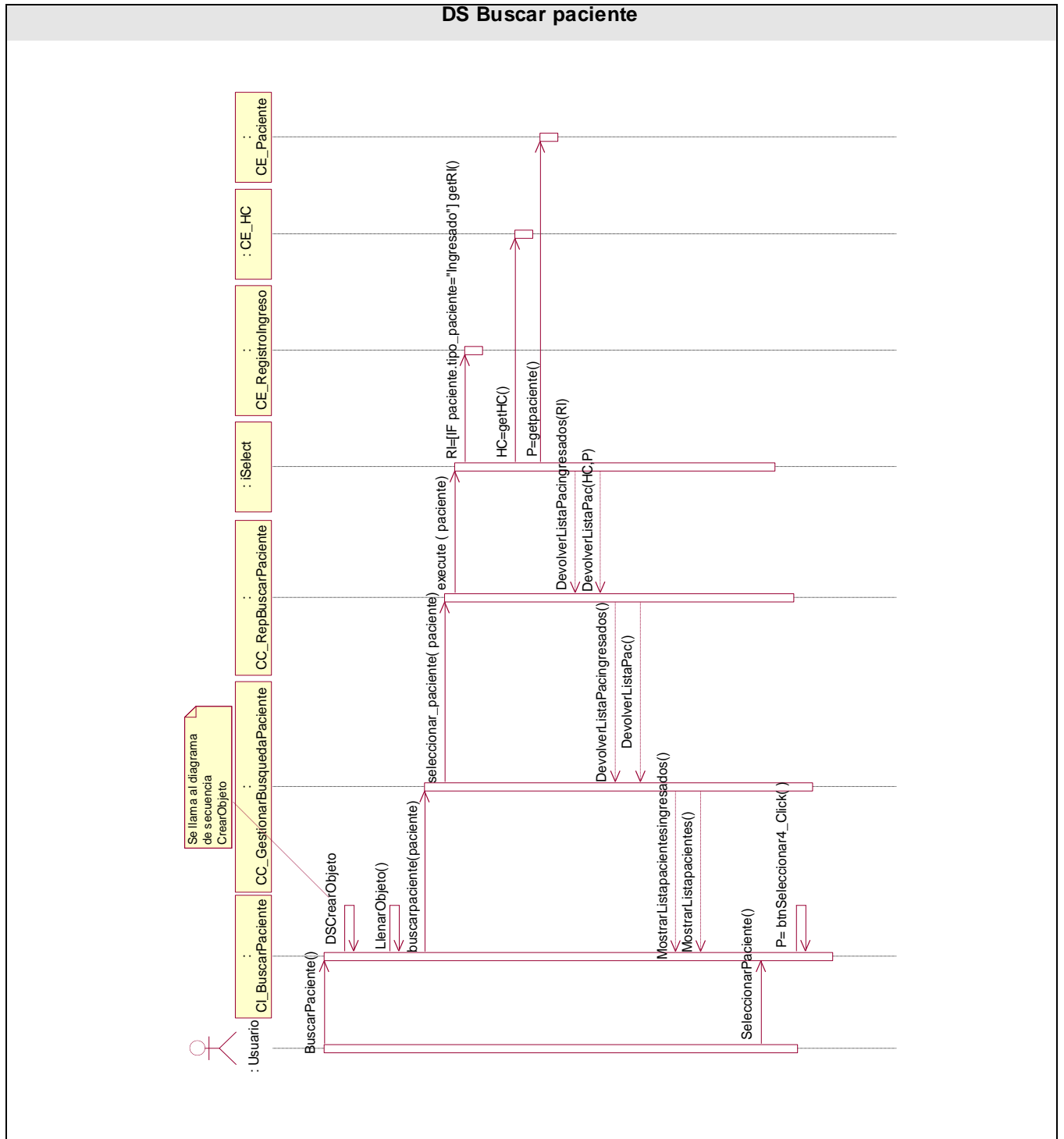


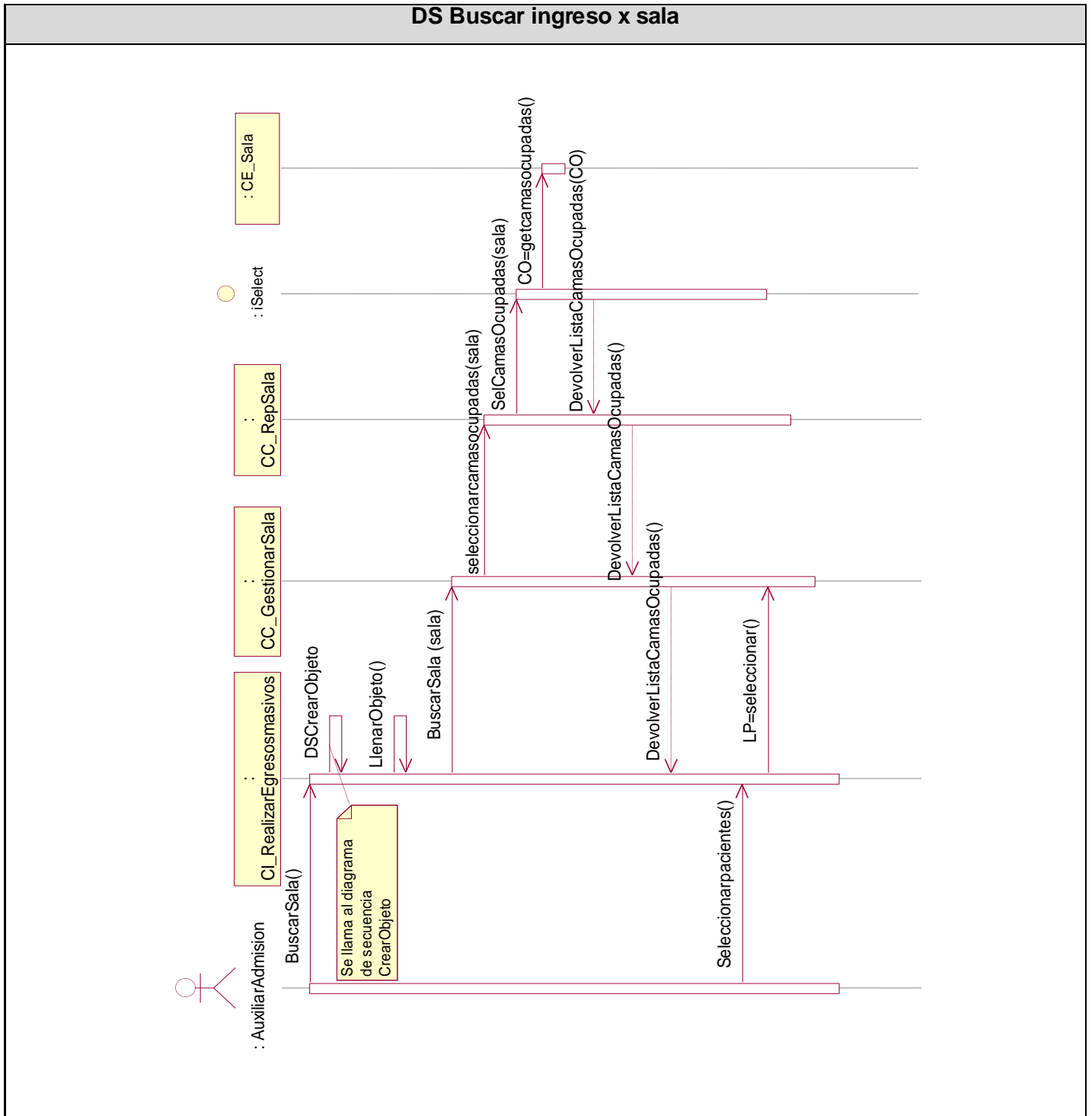












3.4 Descripción de las clases

Nombre: CI_BuscarPaciente	
Tipo de clase : Interfaz	
Atributo	Tipo
DropDownList5	DropDownList
TextBox1	textbox
DropDownList7	DropDownList
TextBox2	textbox
txtFecNac	textbox
Ca13	Button
DropDownList6	DropDownList
DropDownList8	DropDownList
GridView2	GridView
Button3	Button
btnSeleccionar4	Button
Para cada responsabilidad:	
Nombre:	public String TargetControlID
Descripción:	Función que permite definir la propiedad TargetControlId asignando el botón que disparará el WebUserControl en cada una de las interfaces que lo utilicen.
Nombre:	protected void Button3_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnSeleccionar4_Click
Descripción:	Función que define la programación del evento de un botón.

Nombre: CI_AnularEgreso	
Tipo de clase : Interfaz	
Atributo	Tipo
txtHistoria	textbox

bntHC	Button
txtnombre	textbox
txtfecegreso	textbox
txtmotivo	textbox
txtsala	textbox
txtcama	textbox
txtRegistroProfesional1	textbox
btnRegistro	Button
txtNombreFuncionario	textbox
BuscarMedicos1	WebUserController
BuscarPaciente1	WebUserController
btnAceptar	Button
btnCancelar	Button
Para cada responsabilidad:	
Nombre:	protected void btnAceptar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.

Nombre: CI_RealizarEgreso	
Tipo de clase : Interfaz	
Atributo	Tipo
txtHistoria	textbox
bntHC	Button
txtnombre	textbox
txtFeclng	textbox
txtsala	textbox
txtcama	textbox

txtfecegreso	textbox
Cal3	Button
UpdatePanel1	UpdatePanel
DropDownListTS	DropDownList
chknecro	CheckBox
txtRegistroProfesional1	textbox
btnRegistro	Button
txtNombreFuncionario	textbox
btnAceptar	Button
btnCancelar	Button
BuscarMedicos1	WebUserControl
BuscarPaciente1	WebUserControl
Para cada responsabilidad:	
Nombre:	protected void btnAceptar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void Timer3_Tick
Descripción:	Función que da la hora actual

Nombre: CI_RealizarEgresosmasivos	
Tipo de clase : Interfaz	
Atributo	Tipo
DropDownListsala	DropDownList
btnBuscar	Button
txtFecSald:textbox	textbox
Cal1	Button
Timer2	Timer

DropDownListTs	DropDownList
chknecro	CheckBox
txtRegistroProfesional1:textbox	textbox
btnRegistro	Button
txtNombreFuncionario:textbox	textbox
btnAceptar	Button
btnCancelar	Button
BuscarMedicos1:WebUserControl	WebUserControl
Para cada responsabilidad:	
Nombre:	protected void btnAceptar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void Timer2_Tick
Descripción:	Función que da la hora actual

Nombre: CI_RegistrarIngreso	
Tipo de clase : Interfaz	
Atributo	Tipo
txtNombre1	textbox
btnRegistro1	Button
txtApellido1	textbox
txtFechaNac	textbox
txtEdad	textbox
Sexo	dropdownlist
txtNumeroID	textbox
txtFeclng	textbox
TipoIngreso	dropdownlist

txtHoraIng	textbox
Servicio	dropdownlist
Txtmedicoing	textbox
ImageButton1	Button
txtdiagnostico	textbox
txtnombrepadre	textbox
rbpvivo	radiobutton
rbpmuerto	radiobutton
txtnombremadre	textbox
rbmviva	radiobutton
rbmmuerta	radiobutton
txtnombreemerg	textbox
txtdiremer	textbox
txttelemer	textbox
txtsala	textbox
btnBuscarSala	Button
txtcama	textbox
chkfueraserv	CheckBox
btnIngreso	Button
btncancelar	Button

Para cada responsabilidad:

Nombre:	btnIngreso_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	Timer1_Tick
Descripción:	Función que da la hora actual

Nombre: CI_ReubicarIngreso

Tipo de clase : Interfaz

Atributo	Tipo
txthc	textbox
btnRegistro1	Button
txtnombreap	textbox
txtfechaingreso	textbox
btnborrar1	Button
txtsala	textbox
txtcama	textbox
btnborrar2	Button
txtsalareub	textbox
btnBuscarSala	Button
txtcamareub	textbox
btnreubicar	Button
btncancelarreub	Button
Para cada responsabilidad:	
Nombre:	btnreubicar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	btncancelarreub_Click
Descripción:	Función que define la programación del evento de un botón.

Nombre: CI_AnularIngreso	
Tipo de clase : Interfaz	
Atributo	Tipo
txthc	textbox
btnRegistro2	Button
txtnombreapell	textbox
txtfechaing	textbox
txtsala	textbox

txtcamara	textbox
btnaceptaranul	Button
btncancelaranul	Button
Para cada responsabilidad:	
Nombre:	btnaceptaranul_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	btncancelaranul_Click
Descripción:	Función que define la programación del evento de un botón.

Nombre: CI_RegistrarInscripcion	
Tipo de clase : Interfaz	
Atributo	Tipo
txtFecNac	textbox
txtFecIns	textbox
Cal1	Button
txtHistoria	textbox
Cal2	Button
txtCI	textbox
chkExtranjero	checkbox
txtEdad	textbox
radioSexo	radiobutton
txtNombre1	textbox
txtApellido1	textbox
txtApellido2	textbox
cboPiel	dropdownlist
cboEscolaridad	dropdownlist
radioMadreViva	radiobutton
cboEstadoCivil	dropdownlist

txtNombreMadre	textbox
txtCentroTrabajo	textbox
txtDireccionCentroTrabajo	textbox
cboOcupacion	dropdownlist
txtRegistroProfesional1	textbox
btnRegistro	Button
txtNombreFuncionario	textbox
txtDireccion	textbox
txtTelefono	textbox
cboNacionalidad	dropdownlist
cboPais1	dropdownlist
cboProvincia1	dropdownlist
cboMunicipio1	dropdownlist
cboAreaSalud	dropdownlist
chkTrabajador	checkbox
chkHistoria	checkbox
btnAceptar	Button
Para cada responsabilidad:	
Nombre:	protected void btnAceptar_Click
Descripción:	Función que define la programación del evento de un botón.

Nombre: CI_RealizarTraslado	
Tipo de clase : Interfaz	
Atributo	Tipo
txtHistoria	textbox
btnHC	Button
txtnombre	textbox
txtFeclng	textbox

txtsala	textbox
txtcama	textbox
txtFecTras	textbox
Cal3	Button
UpdatePanel4	UpdatePanel
txtsalat	textbox
btnLocCam	Button
txtcamat	textbox
chkfueraserv	checkbox
txtRegistroProfesional1	textbox
btnRegistro	Button
txtNombreFuncionario	textbox
btnAceptar	Button
btnCancelar	Button
BuscarMedicos1	WebUserControl
LocalizarCamasDisponibles1	WebUserControl
BuscarPaciente1	WebuserControl

Para cada responsabilidad:

Nombre:	protected void btnAceptar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void Timer4_Tick
Descripción:	Función que da la hora actual

Nombre: CI_PermutarPaciente	
Tipo de clase : Interfaz	
Atributo	Tipo

txtHistoria	textbox
btnHC1	Button
txtnombre	textbox
txtFecIng	textbox
txtFecTras	textbox
txtsala1	textbox
txtcama1	textbox
chkfueraserv	checkbox
txtHC2	textbox
btnHC2	Button
txtnombre2	textbox
txtFecIng2	textbox
txtFecTras2	textbox
txtsala2	textbox
txtcama2	textbox
txtFecPermuta	textbox
UpdatePanel1	UpdatePanel
btnAceptar	Button
btnCancelar	Button
BuscarPaciente2	WebUserControl
BuscarPaciente1	WebUserControl
Cal6	Button

Para cada responsabilidad:

Nombre:	protected void btnAceptar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void Timer5_Tick

Descripción:	Función que da la hora actual
--------------	-------------------------------

Nombre: CI_AnularTraslado	
Tipo de clase : Interfaz	
Atributo	Tipo
txtHistoria	textbox
ImageButton1	button
txtnombre	textbox
txtcamat	textbox
txtsalat	textbox
txtsalata	textbox
txtcamata	textbox
txtfect	textbox
txthorat	textbox
BuscarPaciente1	WebUserControl
btnAceptar	Button
btnCancelar	button
Para cada responsabilidad:	
Nombre:	protected void btnAceptar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.

Nombre: CI_Localizarcama	
Tipo de clase : Interfaz	
Atributo	Tipo
DropDownList4	DropDownList
DropDownList2	DropDownList

qrdFuncionarios	GridView
btnBuscar2	Button
btnSeleccionar2	Button
Para cada responsabilidad:	
Nombre:	public String TargetControlID
Descripción:	Función que define el la relación de los complementos que se programen y los hipervínculos, botones o imágenes que los activan.
Nombre:	protected void btnBuscar2_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnSeleccionar2_Click
Descripción:	Función que define la programación del evento de un botón.

Nombre: CC_GestionarEgreso	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	realizarEgreso
Descripción:	Función que define la realización del egreso
Nombre:	realizarEgresosMasivos
Descripción:	Función que define la realización de egresos masivos
Nombre:	anularEgreso
Descripción:	Función que define la anulación de un egreso

Nombre: CC_GestionarBúsquedaPaciente	
Tipo de clase : Controladora	
Atributo	Tipo

Para cada responsabilidad:	
Nombre:	buscarpaciente
Descripción:	Función que define la búsqueda de un paciente mediante diferentes criterios

Nombre: CC_GestionarCamas	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	buscarama
Descripción:	Función que define la búsqueda de las camas.

Nombre: CC_GestionarIngreso	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	registrarIngreso
Descripción:	Función que define la realización del ingreso.
Nombre:	reubicarIngreso
Descripción:	Función que define la reubicación del ingreso.
Nombre:	anularIngreso
Descripción:	Función que define la anulación del ingreso.

Nombre: CC_GestionarInscripción	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	

Nombre:	realizarInscripción
Descripción:	Función que define la realización de la inscripción.

Nombre: CC_GestionarTraslado	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	realizarTraslado
Descripción:	Función que permite definir la realización del traslado de pacientes.
Nombre:	permutarPacientes
Descripción:	Función que permite definir la realización de permutas entre pacientes.
Nombre:	anularTraslado
Descripción:	Función que permite definir la anulación de traslado de pacientes.

Nombre: CC_GestionarSala	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	BuscarSala
Descripción:	Función que permite buscar pacientes ingresados en una sala.

Nombre: CC_RepBuscarPaciente	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	seleccionar_paciente
Descripción:	Función que permite definir la selección de pacientes.

Nombre: CC_RepGestionarEgreso	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	adicionar_egreso
Descripción:	Función que permite adicionar un nuevo egreso.
Nombre:	eliminar_egreso
Descripción:	Función que permite eliminar egresos.
Nombre:	actualizar_egreso
Descripción:	Función que permite actualizar los datos del egreso.
Nombre:	seleccionar_egreso
Descripción:	Función que permite definir la selección de egresos.

Nombre: CC_RepGestionarIngresos	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	adicionar_ingreso
Descripción:	Función que permite adicionar un nuevo ingreso.
Nombre:	eliminar_ingreso
Descripción:	Función que permite eliminar ingresos.
Nombre:	actualizar_ingreso
Descripción:	Función que permite actualizar los datos del ingreso.
Nombre:	seleccionar_ingreso
Descripción:	Función que permite definir la selección de ingresos.

Nombre: CC_RepGestionarInscripcion	
Tipo de clase : Controladora	

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	adicionar_persona
Descripción:	Función que permite adicionar una nueva persona.
Nombre:	eliminar_persona
Descripción:	Función que permite eliminar personas.
Nombre:	actualizar_persona
Descripción:	Función que permite actualizar los datos de una persona.
Nombre:	seleccionar_persona
Descripción:	Función que permite definir la selección de personas.

Nombre: CC_RepGestionarTraslado	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	adicionar_traslado
Descripción:	Función que permite adicionar un nuevo traslado.
Nombre:	eliminar_traslado
Descripción:	Función que permite eliminar traslado.
Nombre:	modificar_traslado
Descripción:	Función que permite actualizar los datos de un traslado.
Nombre:	seleccionar_traslado
Descripción:	Función que permite definir la selección de traslados.

Nombre: CC_RepLocalizarCama	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	

Nombre:	adicionar_cama
Descripción:	Función que permite adicionar una nueva cama.
Nombre:	eliminar_cama
Descripción:	Función que permite eliminar una cama.
Nombre:	modificar_cama
Descripción:	Función que permite actualizar los datos de una cama.
Nombre:	seleccionar_cama
Descripción:	Función que permite definir la selección de camas.

Nombre: CC_RepSala	
Tipo de clase : Controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	adicionar_sala
Descripción:	Función que permite adicionar una nueva sala.
Nombre:	eliminar_sala
Descripción:	Función que permite eliminar salas.
Nombre:	actualizar_sala
Descripción:	Función que permite actualizar los datos de las salas.
Nombre:	seleccionarcamasocupadas
Descripción:	Función que permite seleccionar las camas ocupadas.

Nombre: CE_Datospersona	
Tipo de clase : Entidad	
Atributo	Tipo
Id_persona	Integer
telefono	string
centro_trabajo	string

direccion_centro_trabajo	string
donaciones_previas	boolean
transfusiones_previas	boolean
direccion_particular	string
numero_identificacion	string
nombre_persona	string
primer_apellido_persona	string
segundo_apellido_persona	string
fecha_nacimiento	date
fecha_registro	date
nombre_padre	string
nombre_madre	string
padre_vivo	boolean
madre_viva	boolean
trabajador_salud	boolean
provincia	String
sexo	String
color_piel	String
estado_civil	String
ocupación	String
nivel_escolaridad	String
grupo_sanguineo	String
municipio	String
nacionalidad	String
Para cada responsabilidad:	
Nombre:	setDP()
Descripción:	Método que se encarga de modificar los valores de cada atributo de la clase.
Nombre:	getDP()

Descripción:	Devuelve los valores que contienen los atributos de la clase.
--------------	---

Nombre: CE_Paciente	
Tipo de clase : Entidad	
Atributo	Tipo
persona_responsable	string
direccion_emergencia	string
telefono_emergencia	string
tipo_paciente	String
Para cada responsabilidad:	
Nombre:	setpaciente()
Descripción:	Método que se encarga de modificar los valores de cada atributo de la clase.
Nombre:	getpaciente()
Descripción:	Devuelve los valores que contienen los atributos de la clase.

Nombre: CE_Ubicacion	
Tipo de clase : Entidad	
Atributo	Tipo
fecha_ocupacion	date
fecha_liberacion	date
fecha_orden_ingreso	date
fecha_registro_ingreso	date
fecha_donacion	date
Para cada responsabilidad:	
Nombre:	setUb()
Descripción:	Método que se encarga de modificar los valores de cada atributo de la clase.
Nombre:	delete()
Descripción:	Método que se encarga de eliminar los valores de cada atributo de la clase.

Nombre: CE_Cama	
Tipo de clase : Entidad	
Atributo	Tipo
numero_cama	integer
costo	double
Para cada responsabilidad:	
Nombre:	getcama()
Descripción:	Devuelve los valores que contienen los atributos de la clase.

Nombre: CE_Egreso	
Tipo de clase : Entidad	
Atributo	Tipo
fecha_registro_egreso	Date
hora_egreso	Date
Para cada responsabilidad:	
Nombre:	setegreso()
Descripción:	Método que se encarga de modificar los valores de cada atributo de la clase.
Nombre:	delete()
Descripción:	Método que se encarga de eliminar los valores de cada atributo de la clase.

Nombre: CE_Fallecido	
Tipo de clase : Entidad	
Atributo	Tipo
fecha_muerte	Date
hora_muerte	Date
hora_avisado	Date
nombre_pers_avisada	string
hora_certificacion	Date

hora_entrega	Date
necropsia	boolean
funeraria	string
Para cada responsabilidad:	
Nombre:	setF()
Descripción:	Método que se encarga de modificar los valores de cada atributo de la clase.

Nombre: CE_HC	
Tipo de clase : Entidad	
Atributo	Tipo
numero_hc	string
fecha_ultima_salida	Date
Para cada responsabilidad:	
Nombre:	setHC()
Descripción:	Método que se encarga de modificar los valores de cada atributo de la clase.
Nombre:	getHC()
Descripción:	Devuelve los valores que contienen los atributos de la clase.

Nombre: CE_RegistroIngreso	
Tipo de clase : Entidad	
Atributo	Tipo
fecha_registro_ingreso	Date
fecha_hora_ingreso	Date
unidad_remite	integer
numero_control	integer
operador	string
fecha_donacion	Date
urgente	boolean

reevaluacion	boolean
diagnostico	string
donacion	string
funcionario	string
unidad_ingresa	string
tipo_ingreso	string
Para cada responsabilidad:	
Nombre:	setRI()
Descripción:	Método que se encarga de modificar los valores de cada atributo de la clase.
Nombre:	getRI()
Descripción:	Devuelve los valores que contienen los atributos de la clase.
Nombre:	delete()
Descripción:	Método que se encarga de eliminar los valores de cada atributo de la clase.

Nombre: CE_Sala	
Tipo de clase : Entidad	
Atributo	Tipo
id_sala	integer
nombre_sala	string
Para cada responsabilidad:	
Nombre:	getsala()
Descripción:	Devuelve los valores que contienen los atributos de la clase.
Nombre:	getcamasocupadas()
Descripción:	Devuelve una lista con las camas ocupadas en una sala en específico.

A partir de las clases de análisis definidas en función de los casos de uso arquitectónicamente significativos, se arribó al modelo de análisis, que da entrada al diseño y que está elaborado en el

lenguaje del usuario, sin necesidad de especificar métodos ni atributos de las clases expuestas. Como resultado de refinar este artefacto se alcanzó la primera aproximación del modelo de diseño.

Con la elaboración del diseño, se definen las clases fundamentales que permitirán que el sistema funcione satisfactoriamente, así como los atributos y métodos que deben tener las mismas, para brindarle al desarrollador una idea clara de lo que debe implementar.

El uso de patrones de diseño contribuyó a reutilizar la experiencia de los desarrolladores, ya que clasifican y describen formas de solucionar problemas, que ocurren de forma frecuente en el desarrollo.

La realización de cada caso de uso es un artefacto muy importante que da entrada al flujo de trabajo que le sigue al diseño en la metodología RUP. La disciplina que continúa, es la de implementación cuyo principal objetivo es crear componentes más específicos y asignarles a estos las clases del diseño correspondientes, en dependencia de la trazabilidad definida.

Dentro de las realizaciones de los casos de uso se elaboraron los diagramas de interacción. Estos explican la colaboración que existe entre las clases y cómo son llamados los métodos y sentencias dentro de cada una. Con la obtención de estos artefactos se arriba a los resultados esperados en el diseño del Sistema de gestión hospitalaria.

Conclusiones

Con el desarrollo de este trabajo se arriba a las siguientes conclusiones:

- Se realizó un estudio de los procesos de inscripción admisión, lo que permitió elaborar el modelado de negocio.
- Se determinó que los SIH estudiados, no cubren las necesidades del sistema nacional de salud en unos casos y en otros, están elaborados utilizando tecnologías y software propietarios a los cuales nuestro país no tiene acceso.
- Se determinaron las principales funcionalidades del sistema a diseñar y se obtuvo el prototipo no funcional.
- Se realizó el análisis y diseño del sistema aplicando los patrones GRASP y GOF.

De esta forma se da cumplimiento a los objetivos trazados para la elaboración del trabajo de diploma, obteniendo el diseño de un sistema informático, que permita mejorar los procesos de gestión de la información, en el departamento de inscripción admisión en los hospitales cubanos.

Recomendaciones

En la realización del trabajo fue posible determinar algunas recomendaciones necesarias, a la dirección del proyecto:

- Implementar el sistema diseñado utilizando el patrón propuesto.
- Diseñar e implementar el segundo ciclo de desarrollo del producto.
- Desplegar la aplicación en los hospitales y otros centros del país.

Referencias bibliográficas

1. CASTRO, I; GAMEZ, M. *Historia Clínica*. 2007
2. *Soluciones para la Salud*. 2007. Disponible en:
http://www.telvent.com/es/soluciones/sector_publico/salud.asp
3. SIVS.A. *Hosix-V*, 2007. Disponible en: <http://www.sivsa.com/>
4. CARE2X. *CARE2X*, 2007. Disponible en: http://care2x.org/index.php?c2x_lang=es&chglang=1
5. C-DAC. *C-DAC'S Sushrut* 2006. Disponible en: <http://www.cdac.in/html/his/sushrut.asp>
6. SOBREVILLA, A. *Galenhos (R)*, Octubre 2006. Disponible en:
http://www.elhospital.com/eh/secciones/EH/ES/MAIN/IN/ESTUDIOS_CASO/doc_52343_HTML.html?idDocumento=52343%20-%2051k%20-%202007-04-21
7. MEDICAL-SOFT. *MedFile 5.x*. Disponible en: <http://www.medical-soft.com/>
8. (CIM_UAG), C. D. I. M. *HIS 4.0*. Disponible en: <http://cim.uag.mx/his.html>
9. SOFTEL UCI FAC7. *SIH*. Habana. Cuba: Sistema de Información Hospitalaria
10. *Internet*. Disponible en: <http://eia.udg.es/~ramon/ib/t3/tema3.htm#punto31>
11. *Navegadores Web*. Disponible en: http://es.wikipedia.org/wiki/Navegadores_web
12. *Sistemas distribuidos*. 2006. Disponible en: http://www.augcyl.org/glo1/old/N_1/sistemasdis.html
13. *Modelo Cliente/Servidor (dos capas)*. Disponible en:
http://www.augcyl.org/glo1/old/N_1/sistemasdis.html
14. *Modelo cliente-servidor* Disponible en: <http://www.csi.map.es/csi/silice/GlobaI71.html>

15. *Modelo Cliente/Servidor (tres capas)*. 2006. Disponible en: http://www.janium.com/phpws/index.php?module=documents&JAS_DocumentManager_op=downloadFile&JAS_File_id=1
16. *AJAX*. 2007. Disponible en: <http://www.maestrosdelweb.com/editorial/ajax/>
17. *Metodologías de Desarrollo*. 2007. Disponible en: <http://www.um.es/docencia/barzana/IAGP/Iagp2.html#BM1>
18. RATIONAL SOFTWARE. *RUP*. Disponible en: http://es.wikipedia.org/wiki/Rational_Unified_Process
19. *eXtreme Programming (XP)*. Disponible en: <http://www.willydev.net/descargas/prev/ToDoAgil.Pdf>,
http://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema
20. *Scrum*. Disponible en: <http://es.wikipedia.org/wiki/Scrum>
21. *Crystal Methodologies* 2007. Disponible en: http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/heterodox.asp#8
22. *Mono*. Disponible en: <http://www.pedromj.com/tecnologias/mono.aspx>
23. *Patrones de diseño*. 2007. Disponible en: <http://www.ingenierosoftware.com/analisisydiseño/patrones-diseño.php>
24. *Firewall*. 2007. Disponible en: <http://www.desarrolloweb.com/articulos/513.php>
25. *Data warehouse*. 2007. Disponible en: <http://www.abcdatos.com/tutoriales/tutorial/I7781.html>
26. *W3C*. 2007. Disponible en: <http://www.w3c.es/Consortio/>
27. *SGML*. 2007. Disponible en: <http://www.alegsa.com.ar/Dic/metalinguaje.php>
28. *DCOM*. 2007. Disponible en: <http://acsblog.es/svn/pfc/DCOM/DCOM.html>
29. *CORBA*. 2007. Disponible en: <http://www.osmosislatina.com/java/rmi.htm>
30. *Prototipo no funcional*. 2007. Disponible en: <http://www.poder-judicial.go.cr/proveeduria/archivos/licitaciones/anejo%202.doc>

31. *Arquitectura Cliente – Servidor*. 2007. Disponible en: <http://www.csi.map.es/csi/silice/Global71.html>
32. *Base de datos*. 2007. Disponible en: <http://www.mastermagazine.info/termino/4012.php>
33. *Gestor de Base de Datos*. 2007. Disponible en:
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php
34. *Sistemas operativos*. 2007. Disponible en: <http://www.mflor.mx/materias/comp/cursoso/sisope1.htm>
35. *TCP/IP*. 2007. Disponible en:
<http://www.monografias.com/trabajos/protocolotcpip/protocolotcpip.shtml>
36. *OASIS*. 2007. Disponible en: http://formacion.cnice.mec.es/materiales/43/cd/cap8/cap8_frame.htm
37. *Servidor*. 2007. Disponible en: <http://www.mty.itesm.mx/rectoria/dda/usols/concepto1.htm>
38. *DB2*. 2007. Disponible en: <http://www.bytepile.com/definiciones-d.php>
39. *Oracle*. 2007. Disponible en: <http://www.alegsa.com.ar/Diccionario/dic.php?palabra=oracle>
40. *Clases*. 2007. Disponible en: http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/I_1.htm
41. *Herencia*. 2007. Disponible en: <http://lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>
42. *Funciones*. 2007. Disponible en: <http://www.desarrolloweb.com/manuales/20/>
43. *Restricciones*. 2007. Disponible en: <http://www.varlena.com/GeneralBits/47es.php>
44. *Disparadores*. 2007. Disponible en:
<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=178>
45. *Reglas*. 2007. Disponible en: http://www.sqlmax.com/centro/moduloII_4.asp?MX=
46. *Integridad transaccional*. 2007. Disponible en: <http://support.microsoft.com/kb/188589/es>
47. *Orientado a objetos*. 2007. Disponible en: <http://www.monografias.com/trabajos5/softobj/softobj.shtml>

48. *ECMA*. 2007. Disponible en: <http://www.albertolacalle.com/disenostandares.htm>
49. *Compiladores*. 2007. Disponible en: <http://www.dimelo.com/GS/SeeTheme.aspx?item=60>
50. *Licencia LGPL*. 2007. Disponible en: <http://creativecommons.org/licenses/LGPL/2.1/>
51. *OMG*. 2007. Disponible en: <http://www-gris.det.uvigo.es/~avilas/UML/node9.html>
52. *Metodología RAD*. 2007. Disponible en:
http://www.planetacodigo.com/wiki/glosario:rapid_application_development
53. *Frames*. 2007. Disponible en: <http://www.desarrolloweb.com/articulos/791.php>
54. *Clases abstractas*. 2007. Disponible en: http://www.fi-b.unam.mx/pp/profesores/carlos/java/java_basico4_8.html

Bibliografía

- BLASI, EMANUEL (mayo del 2007): *Resumen de Patrones de Diseño*
- CORPORATION, R. S., Ayuda de Rational Suite 2003. Consultada en enero del 2007.
- EMPRESA SOFTEL; UCIFAC7: Sistema informático SIH, Ciudad Habana, Cuba, 2004- 2005.
- JACOBSON, IVAR; RUMBAUGH, JAMES; BOOCH, GRADY: *El proceso unificado de desarrollo*, Editorial Addison Wesley, 2000.
- JACOBSON, IVAR; RUMBAUGH, JAMES; BOOCH, GRADY: *El lenguaje unificado de modelado. Manual de referencia*, Editorial Addison Wesley, 2000.
- JOSÉ H. CANÓS, PATRICIO LETELIER Y M^a CARMEN PENADÉS (enero 2007): *Todo Ágil*. Disponible en <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- LARMAN, CRAIG: *Uml y patrones*, Editorial Prentice Hall Iberoamericana, 1999.
- LEÓN WELICKI (mayo del 2007): “Patrones de Fabricación: Fábricas de Objetos”, en http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/MTJ_3624.asp#M1,
- MARIN, MIGUEL: *Fundamentos del Sistema de Salud Pública en Cuba para estudiantes de informática*. La Habana, 2006.
- MOLPECERES, A: *Procesos de desarrollo: RUP, XP Y FDD*, 2002.
- PRESSMAN, ROGER: *Ingeniería del Software. Un enfoque práctico*, Editorial McGraw-Hill, España, 2002.
- SOBREVILLA, A. *Galenhos (R)*, 2006. Disponible en: http://www.elhospital.com/eh/secciones/EH/ES/MAIN/IN/ESTUDIOS_CASO/doc_52343_HTML.html?idDocumento=52343%20-%2051k%20-%202007-04-21
- WIKIPEDIA: <http://wikipedia.org/>
- WORSLEY, JOHN; DRAKE, JOSHUA: *PostgreSQL(2006)*. Disponible en: <http://www.sobl.org/traduccion/practical-postgres/node19.html>

Anexos

Anexo 1. Modelo de casos de uso

4.1.1 Casos de uso por ciclos de desarrollo

Para el primer ciclo de desarrollo se propone que se realicen los casos de uso que guían los procesos de inscripción admisión, y que tengan prioridad crítica de manera que quede diseñado el subsistema para que el mismo pueda ser implementado y se valga de estos casos de uso para su funcionamiento, de esta forma se definen los siguientes:

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-1	Gestionar inscripción	CU Arquitectónicamente significativos	CU imprescindible para el funcionamiento del módulo completo.
CU-2	Gestionar ingreso	CU Arquitectónicamente significativos	CU importante dentro del proceso de inscripción admisión.
CU-3	Localizar camas disponibles	CU Arquitectónicamente significativos	CU importante dentro del proceso de inscripción admisión.
CU-4	Buscar paciente	CU Arquitectónicamente significativos	CU importante dentro del proceso de inscripción admisión.
CU-5	Gestionar traslado	CU Arquitectónicamente significativos	CU importante dentro del proceso de inscripción admisión.
CU-6	Gestionar egreso	CU Arquitectónicamente significativos	CU importante dentro del proceso de inscripción admisión.
CU-7	Realizar egresos masivos	CU Arquitectónicamente significativos	CU importante dentro del proceso de inscripción admisión.
CU-8	Ingresos por salas	CU Arquitectónicamente significativos	CU importante dentro del proceso de inscripción admisión.

Para el segundo ciclo de desarrollo se plantean los casos de uso pertenecientes al paquete de reportes y que se encargan de mostrar estadísticas y listados que tienen importancia para el hospital pero que en estos momentos no forman parte del principal problema que es la automatización de los servicios a los pacientes.

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-9	Mostrar informe inscripciones realizadas	Reportes	CU definidos con prioridad secundaria.
CU-10	Mostrar registro pacientes	Reportes	CU definidos con prioridad secundaria.
CU-11	Mostrar Registro Ingresos	Reportes	CU definidos con prioridad secundaria.
CU-12	Mostrar Reporte de Ingresos	Reportes	CU definidos con prioridad secundaria.
CU-13	Mostrar Registros de Casos Fuera de Servicio	Reportes	CU definidos con prioridad secundaria.
CU-14	Mostrar disponibilidad de Camas	Reportes	CU definidos con prioridad secundaria.
CU-15	Mostrar registro de traslado	Reportes	CU definidos con prioridad secundaria.
CU-16	Mostrar Egresos	Reportes	CU definidos con prioridad secundaria.
CU-17	Mostrar Reporte de Egresos	Reportes	CU definidos con prioridad secundaria.
CU-18	Mostrar registro de movimiento hospitalario	Reportes	CU definidos con prioridad secundaria.
CU-19	Mostrar resumen diario de movimiento de salas	Reportes	CU definidos con prioridad secundaria.
CU-20	Mostrar registro diario censo de pacientes	Reportes	CU definidos con prioridad secundaria.
CU-21	Mostrar informe de	Reportes	CU definidos con prioridad

	movimiento hospitalario		secundaria.
--	-------------------------	--	-------------

4.1.2 Listado de casos de uso

CU-1	Gestionar inscripción
Actor	Auxiliar de inscripción
Descripción	Este Caso de uso tiene como principal objetivo inscribir a los pacientes en el hospital, luego de estar inscritos ya podrán atenderse en cualquiera de las secciones del mismo, así como recibir todos los servicios que brinda la institución.
Referencia	RF 2, CU incluido Buscar médico(Módulo Administración)

CU-2	Gestionar ingreso
Actor	Auxiliar de admisión
Descripción	Este caso de uso se inicia cuando se quiere realizar un ingreso, modificar o reubicar a un paciente o en caso de que se necesite anular el ingreso.
Referencia	RF 3.1, RF3.2, RF 3.3, CU incluido Buscar médico (Módulo Administración), CU incluido Buscar HC (Módulo Archivo).

CU-3	Localizar camas disponibles
Actor	Auxiliar de admisión
Descripción	El caso de uso se invoca cuando se va a efectuar un ingreso y hace falta ubicar a un paciente en un servicio determinado. Los servicios tienen varias salas y estas a su vez tienen camas que pueden estar disponibles u ocupadas, en este último caso el paciente tendría que ingresar en otra sala que tenga camas disponibles o en otro servicio, siendo el ingreso catalogado como fuera de servicio.
Referencia	RF4

CU-4	Buscar paciente
Actor	Auxiliar de admisión
Descripción	El caso de uso es iniciado cuando el auxiliar de admisión necesita ingresar, trasladar o egresar a un paciente y le hace falta localizarlo para poder realizar los movimientos pertinentes.
Referencia	RF5

CU-5	Gestionar traslado
Actor	Auxiliar de admisión
Descripción	El caso de uso se inicia cuando el auxiliar de admisión va a realizar un traslado de un paciente de un servicio o sala a otro. Este caso de uso permite trasladar a un paciente, permutar pacientes y anular el traslado.
Referencia	RF 6.1, RF 6.2, RF 6.3, CU incluido Buscar paciente, CU incluido Localizar camas disponibles.

CU-6	Gestionar egreso
Actor	Auxiliar de admisión
Descripción	El caso de uso se inicia cuando el auxiliar de admisión va a realizar un egreso de un paciente, ya sea por alta médica, por fallecimiento o por traslado. Permite además realizar egresos masivos y anular egresos.
Referencia	RF 7.1, RF 7.2, RF 7.3, CU incluido Buscar paciente, CU incluido Buscar médico (Módulo Administración).

CU-7	Realizar egresos masivos
Actor	Auxiliar de admisión
Descripción	El caso de uso se inicia cuando el auxiliar de admisión va a realizar egresos masivos, ya sea por alta médica, por fallecimiento o por traslado de pacientes.
Referencia	RF 8, CU incluido Buscar médico (Módulo Administración).

CU-8	Buscar ingresos por sala
Actor	Auxiliar de admisión
Descripción	El caso de uso se inicia cuando el auxiliar de admisión necesita seleccionar pacientes de una sala.
Referencia	RF 9

CU-9	Mostrar Informe inscripciones realizadas
Actor	Auxiliar de inscripción
Descripción	Permite consultar el número de inscripciones que se han realizado en un determinado período de tiempo.
Referencia	RF 10

CU-10	Mostrar registro pacientes
Actor	Auxiliar de inscripción
Descripción	Permite consultar el registro de los datos de los pacientes del hospital
Referencia	RF 11

CU-11	Mostrar Registro Ingresos
Actor	Auxiliar de admisión
Descripción	Permite consultar la información correspondiente a los pacientes ingresados en algún momento en el Hospital o que lo están actualmente, en dependencia de los criterios de búsqueda.
Referencia	RF 12

CU-12	Mostrar Reporte de Ingresos
Actor	Auxiliar de admisión
Descripción	Permite consultar la cantidad de ingresos efectuados en un determinado periodo de tiempo.
Referencia	RF 13

CU-13	Mostrar Registros de Casos Fuera de Servicio
Actor	Auxiliar de admisión
Descripción	Permite consultar la cantidad de ingresos efectuados en un determinado periodo de tiempo.
Referencia	RF 14

CU-14	Mostrar disponibilidad de Camas
Actor	Auxiliar de admisión
Descripción	Permite consultar la información correspondiente a la disponibilidad de camas por salas, estado de la cama y tipo de cama.
Referencia	RF 15

CU-15	Mostrar registro de traslado
Actor	Auxiliar de admisión
Descripción	Permite consultar los traslados realizados por varios criterios.
Referencia	RF 16

CU-16	Mostrar Egresos
Actor	Auxiliar de admisión
Descripción	Permite consultar el número de egresos que se han realizado.
Referencia	RF 17

CU-17	Mostrar Reporte de Egresos
Actor	Auxiliar de admisión
Descripción	Permite consultar la información correspondiente a los egresos realizados en el hospital.
Referencia	RF 18

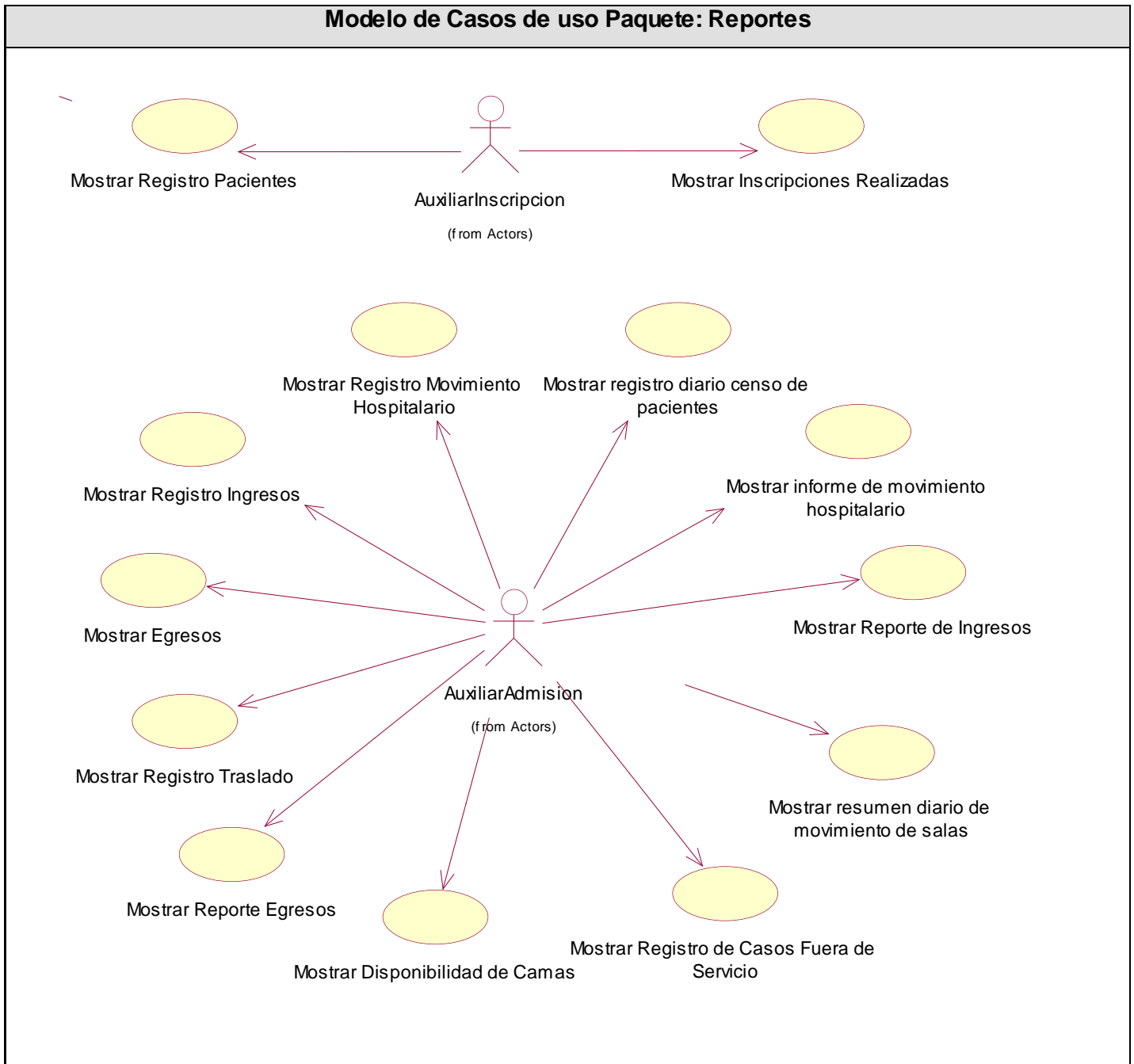
CU-18	Mostrar registro de movimiento hospitalario
Actor	Auxiliar de admisión
Descripción	Muestra un reporte con los ingresos por HC, conjuntamente con otros datos globales del mismo, así como del egreso en un rango de tiempo determinado.
Referencia	RF 19

CU-19	Mostrar resumen diario de movimiento de salas
Actor	Auxiliar de admisión
Descripción	Muestra un reporte detallado por salas de las entradas y salidas a las mismas, así como el estado en que se encuentran diariamente.
Referencia	RF 20

CU-20	Mostrar registro diario censo de pacientes
Actor	Auxiliar de admisión
Descripción	Muestra un reporte de los ingresos incluyendo los que vienen de traslado, de la misma forma los egresos, muestra además los que han egresado fallecidos, así como otros datos de interés.
Referencia	RF 21

CU-21	Mostrar informe de movimiento hospitalario
Actor	Auxiliar de admisión
Descripción	Muestra un reporte del estado del hospital en cuanto a camas ocupadas, ingresos, egresos, fallecidos en los diferentes servicios en un trimestre.
Referencia	RF 22

4.1.3 Modelo de Casos de uso Paquete: Reportes



4.1.4 Descripciones textuales de los Casos de Uso del Sistema

Las descripciones textuales de los casos de uso permiten una mejor comprensión de las transacciones que el sistema propuesto debe tener y el orden lógico en el que se realizarán las mismas. Muestran el flujo normal de los eventos así como el curso alternativo que tomaría el sistema en caso de no poder realizar una transacción satisfactoriamente.

Caso de uso: Gestionar Inscripción

Caso de Uso:	Gestionar Inscripción		
Actores:	Auxiliar de Inscripción		
Resumen:	Este caso de uso se inicia cuando el paciente necesita inscribirse en el hospital y se presenta en el mismo para inscribirse y que se le cree una historia clínica, luego de estar inscritos ya podrán atenderse en cualquiera de las secciones del mismo, así como recibir todos los servicios que brinda la institución. El caso de uso finaliza cuando se le crea al paciente la historia clínica.		
Referencias	RF2, CU incluido Buscar Médico (Módulo Administración).		
Precondiciones:	Antes de iniciarse el caso de uso el auxiliar de inscripción debe autenticarse en el sistema.		
Prioridad	Crítico		
Curso Normal de los eventos			
Acción del Actor		Respuesta del Sistema	
1	El auxiliar de inscripción accede al sistema y selecciona la opción realizar la inscripción de un paciente.	2	El sistema muestra una interfaz para registrar los datos personales del paciente.
3	El auxiliar de inscripción llena los datos personales y selecciona la provincia, el	4	El sistema carga las listas de las

	municipio y el área de salud a la que pertenece.		provincias y municipios.
5	El auxiliar de inscripción escoge el médico que autoriza la inscripción.	6	El sistema se conecta al módulo de administración que proporciona un listado de todos los médicos con privilegios para autorizar la inscripción.
7	El auxiliar de inscripción registra al paciente en el sistema.	8	El sistema verifica que los datos estén correctos.
		9	Se almacenan los datos en la BD del sistema y se muestra una página notificando el registro satisfactorio del paciente.
Cursos Alternos			
Sección “Inscribir paciente”			
Línea 8: El sistema muestra un mensaje de error porque los datos están incorrectos.			
Línea 9: El sistema muestra un mensaje de error porque el paciente ya se encontraba registrado.			

Caso de uso: Gestionar Ingreso

Caso de Uso:	Gestionar Ingreso
Actores:	Auxiliar de Admisión
Resumen:	Este caso de uso se inicia cuando se quiere realizar un ingreso, modificar o reubicar a un paciente o en caso de que se necesite anular el ingreso. Finaliza cuando el paciente ha sido ingresado, reubicado o simplemente se anula el ingreso efectuado.
Referencias	RF 3, CU incluido Buscar paciente, CU incluido Buscar médico (Módulo Administración), CU incluido Localizar camas disponibles.

Precondiciones:	Antes de iniciarse el caso de uso el auxiliar de admisión debe autenticarse en el sistema.		
Prioridad	Crítico		
Curso Normal de los eventos			
1. El auxiliar de admisión accede al sistema.	2. El sistema muestra las acciones que puede realizar: <ul style="list-style-type: none"> • Si desea realizar un ingreso ver sección: Realizar Ingreso. • Si desea cambiar de sala o de cama a un paciente ver sección: Reubicar Ingreso. • Si desea eliminar el ingreso, que no sería lo mismo que egresar al paciente, ver sección: Anular Ingreso. 		
Sección “Realizar Ingreso”			
Acción del Actor		Respuesta del Sistema	
3	El auxiliar de admisión selecciona la opción de ingresar un paciente.	4	El sistema muestra una interfaz para localizar al paciente que se desea ingresar, a través del número de la HC. Se ejecuta el caso de uso incluido Buscar paciente . Luego carga todos los datos generales del paciente.
5	El auxiliar de admisión selecciona el tipo de ingreso si es electivo o urgente. Selecciona el servicio en el que se va a internar al paciente.	6	El sistema carga un listado con los médicos que pertenecen al servicio seleccionado. Se ejecuta el caso de uso incluido Buscar Médico del Módulo de administración.
7	El auxiliar de admisión escoge el médico y llena otros datos necesarios de la orden de ingreso.	8	El sistema muestra una interfaz para ubicar al paciente en una de las salas del servicio en el que ingresa. Se ejecuta el caso de uso incluido Localizar camas disponibles .

9	Se le asigna la cama y la sala al paciente y en caso de que el ingreso sea fuera del servicio requerido se especifica en el ingreso. Finalmente el auxiliar de admisión presiona el botón Ingresar.	10	El sistema verifica y valida los datos.
		11	Se almacenan los datos en la BD del sistema. Se muestra un mensaje para afirmar la finalización de la transacción.
Sección "Reubicar Ingreso"			
Acción del Actor		Respuesta del Sistema	
12	El auxiliar de admisión selecciona la opción de Reubicar ingreso.	13	El sistema muestra una interfaz para localizar al paciente que se desea ingresar, a través del número de la HC. Se ejecuta el caso de uso incluido Buscar paciente . Luego carga los datos del paciente y los del ingreso en específico.
		14	El sistema muestra todas las salas con sus camas disponibles. Se ejecuta el caso de uso incluido Localizar camas disponibles .
15	El auxiliar selecciona la sala y la cama en la que va a reubicar al paciente en dependencia de la disponibilidad de las mismas.	16	El sistema muestra un mensaje diciendo que la reubicación ha sido satisfactoriamente finalizada.
Sección "Anular Ingreso"			
Acción del Actor		Respuesta del Sistema	
Cursos Alternos			
17	El auxiliar de admisión selecciona la opción de Anular ingreso.	18	El sistema muestra una interfaz para localizar al paciente que se le desea anular el ingreso.
19	El auxiliar introduce el número de la historia clínica para buscar al	20	El sistema hace una búsqueda interna ejecutando el caso de uso incluido Buscar

	paciente ingresado al que se le va a anular el ingreso.		paciente.
		21	El sistema carga los datos generales del paciente ingresado.
22	El auxiliar de inscripción presiona el botón anular para de esta forma cancelar el ingreso realizado.	23	El sistema muestra un mensaje para preguntar si realmente quiere anular el ingreso.
24	El auxiliar selecciona el botón si anulando definitivamente el ingreso.	25	El sistema elimina el ingreso en la BD finalizando la transacción.
Sección “Inscribir paciente”			
Línea 7: El sistema muestra un mensaje de error porque los datos están incorrectos.			
Línea 8: El sistema muestra un mensaje de error porque el paciente ya se encontraba registrado.			
Sección “Reubicar Ingreso”			
Línea 14: El sistema muestra un mensaje de error porque no hay camas disponibles en la sala de ese servicio.			
Sección “Anular Ingreso”			
Línea 25: El auxiliar seleccionó el botón No y el ingreso no se anula.			

Caso de uso: Localizar camas disponibles

Caso de Uso:	Localizar camas disponibles
Actores:	Auxiliar de admisión
Resumen:	El caso de uso se inicia cuando se requiere conocer una lista de camas disponibles ya sea para efectuar un ingreso o para reubicar a un paciente en un servicio determinado. Los servicios tienen varias salas y estas a su vez tienen camas que pueden estar disponibles u ocupadas, en este último caso el paciente tendría que ingresar en otra sala que tenga camas disponibles o

	en otro servicio, siendo el ingreso catalogado como fuera de servicio.		
Referencias	RF 4		
Precondiciones:	Antes de iniciarse el caso de uso el auxiliar de admisión debe autenticarse en el sistema.		
Prioridad	Crítico		
Curso Normal de los eventos			
Acción del Actor		Respuesta del Sistema	
1	Selecciona la opción de ubicar al paciente	2	El sistema muestra una interfaz que carga el servicio seleccionado en el ingreso.
3	El auxiliar de admisión selecciona la sala en la que desea ubicar al paciente y presiona el botón buscar para saber si existen camas disponibles en la misma.	4	El sistema muestra el listado de camas disponibles de esa sala.
5	El auxiliar debe seleccionar la cama en la que va a ubicar al paciente.	6	El sistema carga la sala y la cama en la que se ha ubicado al paciente.
Cursos Alternos			
<p>Línea 3: Si el auxiliar no selecciona la sala, el sistema devuelve todas las salas del servicio que tienen camas disponibles.</p> <p>Línea 4: Si el servicio no tiene camas disponibles, el sistema permite escoger otro servicio para hacer la búsqueda, devolviendo las salas que tiene las camas disponibles en una lista.</p> <p>Línea 6: En el caso de que el ingreso se haga en otro servicio que no corresponda al que debe ser, se especifica que el ingreso es Fuera de servicio y se muestra una pantalla para seleccionar el servicio en el que debería ingresar el paciente.</p>			

Caso de uso: Buscar paciente

Caso de Uso:	Buscar paciente		
Actores:	Auxiliar de admisión		
Resumen:	El caso de uso se inicia cuando se hace necesario buscar un paciente por cualquier criterio, ya sea por su nombre, HC, sexo entre otros criterios de búsqueda, generalmente se busca un paciente para que el mismo reciba un servicio determinado del hospital.		
Referencias	RF 5		
Precondiciones:	Antes de iniciarse el caso de uso el auxiliar de admisión debe autenticarse en el sistema.		
Prioridad	Crítico		
Curso Normal de los eventos			
Acción del Actor		Respuesta del Sistema	
1	El usuario del sistema selecciona la opción buscar HC.	2	El sistema muestra una interfaz permitiendo la búsqueda por varios criterios.
3	El usuario del sistema determina los criterios a través de los cuales realizará la búsqueda y presiona el botón Buscar.	4	El sistema devuelve una lista de pacientes, así como su número de HC, entre otros campos para que sea seleccionado el paciente en cuestión.
5	El usuario del sistema selecciona de la lista el paciente o la historia clínica deseada.	6	El sistema carga en la interfaz correspondiente los datos del paciente seleccionado.
Cursos Alternos			
Línea 4: El sistema no devuelve ninguna información debido a que no se encontró el paciente o la			

HC especificada.

Caso de uso: Gestionar Traslado

Caso de Uso:	Gestionar Traslado	
Actores:	Auxiliar de admisión	
Resumen:	El caso de uso se inicia cuando el auxiliar de admisión va a realizar un traslado de un paciente de un servicio o sala a otro. Este caso de uso permite trasladar a un paciente, permutar pacientes y anular el traslado.	
Referencias	RF 6.1, RF 6.2, RF 6.3, CU incluido Buscar paciente, CU incluido Localizar camas disponibles.	
Precondiciones:	Antes de iniciarse el caso de uso el auxiliar de admisión debe autenticarse en el sistema.	
Prioridad	Crítico	
Curso Normal de los eventos		
Acción del Actor		Respuesta del Sistema
1. El auxiliar de admisión accede al sistema.		2. El sistema muestra las acciones que puede realizar: <ul style="list-style-type: none"> • Si desea realizar un traslado ver sección: Realizar Traslado. • Si desea permutar dos pacientes ver sección: Permutar pacientes. • Si desea anular el traslado ver sección: Anular traslado.
Sección "Realizar Traslado"		
Acción del Actor		Respuesta del Sistema
3	El auxiliar de admisión selecciona la opción de trasladar un paciente.	4 El sistema muestra una interfaz para realizar el traslado. Esta interfaz

			permite buscar al paciente que será trasladado a través del CU Buscar paciente .
5	El auxiliar de admisión selecciona el paciente que será trasladado.	6	Muestra los datos del ingreso del paciente que será trasladado.
		7	Carga los datos de la fecha y hora de la PC cliente en la que está corriendo el sistema.
		8	El sistema busca las salas y camas disponibles para el servicio especificado, invocando al caso de uso incluido Localizar camas disponibles .
9	El auxiliar de admisión selecciona el servicio y la sala para la cual será trasladado el paciente.	10	El sistema carga los datos seleccionados.
11	El auxiliar presiona el botón aceptar para finalmente trasladar al paciente.	12	El sistema verifica los datos y los envía a la BD mostrando un mensaje de finalización de la transacción.
Sección "Permutar pacientes"			
Acción del Actor		Respuesta del Sistema	
13	El auxiliar de admisión selecciona la opción de permutar pacientes.	14	El sistema muestra una interfaz que carga la hora y fecha de la PC cliente.
		15	El sistema muestra una interfaz para seleccionar los pacientes que se van a permutar invocando al caso de uso

			incluido Buscar paciente .
16	El auxiliar selecciona los pacientes que va a permutar	17	El sistema carga los datos de los pacientes que han sido seleccionados.
18	El auxiliar de admisión oprime el botón aceptar para realizar la permuta.	19	El sistema cambia de servicio, sala y cama a ambos pacientes finalizando la permuta.
Sección “Anular traslado”			
Acción del Actor		Respuesta del Sistema	
20	El auxiliar de admisión selecciona la opción de anular traslado.	21	El sistema muestra una interfaz que permite localizar al paciente al que se le quiere anular el traslado. Se invoca al caso de uso incluido Buscar paciente .
22	El auxiliar selecciona el paciente al que se le anulará el traslado.	23	El sistema carga los datos del traslado realizado al paciente con anterioridad.
23	El auxiliar presiona el botón aceptar para anular el traslado.	24	El sistema ubica al paciente en el servicio sala y cama en que estaba antes de realizar el traslado.
Cursos Alternos			
Sección “Realizar Traslado”			
Línea 9: En el caso de que el auxiliar seleccione la opción Fuera de servicio el sistema muestra una pantalla para seleccionar el servicio en el que debería estar el paciente.			
Sección “Permutar pacientes”			
Sección “Anular traslado”			
Línea 24: Si la cama que ocupaba el paciente trasladado esta siendo ocupada no se puede			

deshacer el traslado.

Caso de uso: Gestionar Egreso

Caso de Uso:	Gestionar Egreso		
Actores:	Auxiliar de admisión		
Resumen:	El caso de uso se inicia cuando el auxiliar de admisión va a realizar un egreso de un paciente, ya sea por alta médica, por fallecimiento o por traslado. Permite además realizar egresos masivos y anular egresos.		
Referencias	RF 7.1, RF 7.2, CU incluido Buscar paciente, CU incluido Buscar médico (Módulo Administración).		
Precondiciones:	Antes de iniciarse el caso de uso el auxiliar de admisión debe autenticarse en el sistema.		
Prioridad	Crítico		
Curso Normal de los eventos			
Acción del Actor		Respuesta del Sistema	
1. El auxiliar de admisión accede al sistema.		2. El sistema muestra las acciones que puede realizar: <ul style="list-style-type: none"> • Si desea realizar un egreso ver sección: Realizar egreso. • Si desea anular el egreso ver sección: Anular egreso. 	
Sección "Realizar egreso"			
Acción del Actor		Respuesta del Sistema	
3	El auxiliar de admisión selecciona la opción de egresar un paciente.	4	El sistema muestra una interfaz para realizar el egreso. Esta interfaz permite buscar al paciente que será egresado, a través del CU Buscar

			paciente.
5	El auxiliar de admisión selecciona el paciente que será egresado.	6	Muestra los datos del ingreso del paciente que será egresado.
		7	Carga los datos de la fecha y hora de la PC cliente en la que está corriendo el sistema.
8	El auxiliar de admisión selecciona el tipo de egreso y el médico que da de alta al paciente.	9	El sistema invoca al caso de uso incluido Buscar médico del Módulo de Administración.
10	El auxiliar presiona el botón aceptar para finalmente egresar al paciente.	11	El sistema egresa al paciente liberando la cama de la sala en la que se encontraba ingresado el paciente.
		12	Muestra un mensaje que valida la finalización de la transacción.
Sección "Anular egreso"			
Acción del Actor		Respuesta del Sistema	
13	El auxiliar de admisión selecciona la opción de anular egreso.	14	El sistema muestra una interfaz que permite localizar al paciente al que se le quiere anular el egreso. Se invoca al caso de uso incluido Buscar paciente.
15	El auxiliar selecciona el paciente al que se le anulará el egreso.	16	El sistema carga los datos del egreso realizado al paciente con anterioridad.
17	El auxiliar presiona el botón aceptar para anular el egreso.	18	El sistema ubica al paciente en el servicio sala y cama en que estaba antes de realizar el egreso.
Cursos Alternos			

Sección “Realizar egreso”
Línea 8: Si el egreso es por causa de que falleció el paciente se activa la opción para especificar si tiene necropsia o no.
Sección “Anular egreso”
Línea 18: Si la cama que ocupaba el paciente egresado ya se asignó a otro paciente no se puede deshacer el egreso.

Caso de uso: Realizar egresos masivos

Caso de Uso:	Realizar Egresos Masivos		
Actores:	Auxiliar de admisión		
Resumen:	El caso de uso se inicia cuando el auxiliar de admisión va a realizar un egreso de varios pacientes, ya sea por alta médica, por fallecimiento o por traslado.		
Referencias	RF 8, CU incluido Buscar médico (Módulo Administración).		
Precondiciones:	<p>Antes de iniciarse el caso de uso el auxiliar de admisión debe autenticarse en el sistema.</p> <p>Antes de iniciarse el caso de uso se debe haber realizado previamente el CU Ingresos por Salas.</p>		
Prioridad	Crítico		
Curso Normal de los eventos			
Acción del Actor		Respuesta del Sistema	
1	El auxiliar de admisión selecciona la opción de realizar egresos masivos.	2	El sistema muestra los pacientes seleccionados para egresar masivamente.
3	El auxiliar de admisión especifica el tipo de egreso y el médico que autoriza.	4	El sistema muestra una interfaz para seleccionar el médico invocando el caso de uso incluido Buscar

			médico del Módulo de administración.
5	El auxiliar de admisión presiona el botón aceptar para finalizar el egreso.	6	El sistema muestra un mensaje que valida la finalización de la transacción.

Caso de uso: Buscar ingresos por salas

Caso de Uso:	Buscar ingresos por salas		
Actores:	Auxiliar de admisión		
Resumen:	El caso de uso se inicia cuando es necesario seleccionar un grupo de pacientes de una sala determinada		
Referencias	RF 9		
Precondiciones:	Antes de iniciarse el caso de uso el auxiliar de admisión debe autenticarse en el sistema.		
Prioridad	Crítico		
Curso Normal de los eventos			
Acción del Actor		Respuesta del Sistema	
1	El auxiliar de admisión selecciona la sala de la cual se van a seleccionar los pacientes y presiona el botón buscar.	2	El sistema muestra la lista de pacientes ingresados en esa sala.
3	El auxiliar de admisión selecciona los pacientes que desea egresar.	4	El sistema guarda los pacientes seleccionados.

Anexo 2. Prototipo no funcional

Inscripción

- Datos de la inscripción

Inscripción Inicio : Inscripción-Admisión : Inscribir paciente admin | Salir

Inscripción

- Inscripción-Admisión
 - Inscribir paciente
 - Realizar Ingreso
 - Reubicar Ingreso
 - Anular Ingreso
 - Realizar Traslado
 - Permutar Pacientes
 - Anular Traslado
 - Realizar Egreso
 - Realizar Egresos Masivos
 - Anular Egreso

Datos de la Inscripción

Historia Clínica

Fecha de Nacimiento

Fecha de Inscripción

Extranjero

No. Identificación

Edad

Sexo Masculino Femenino

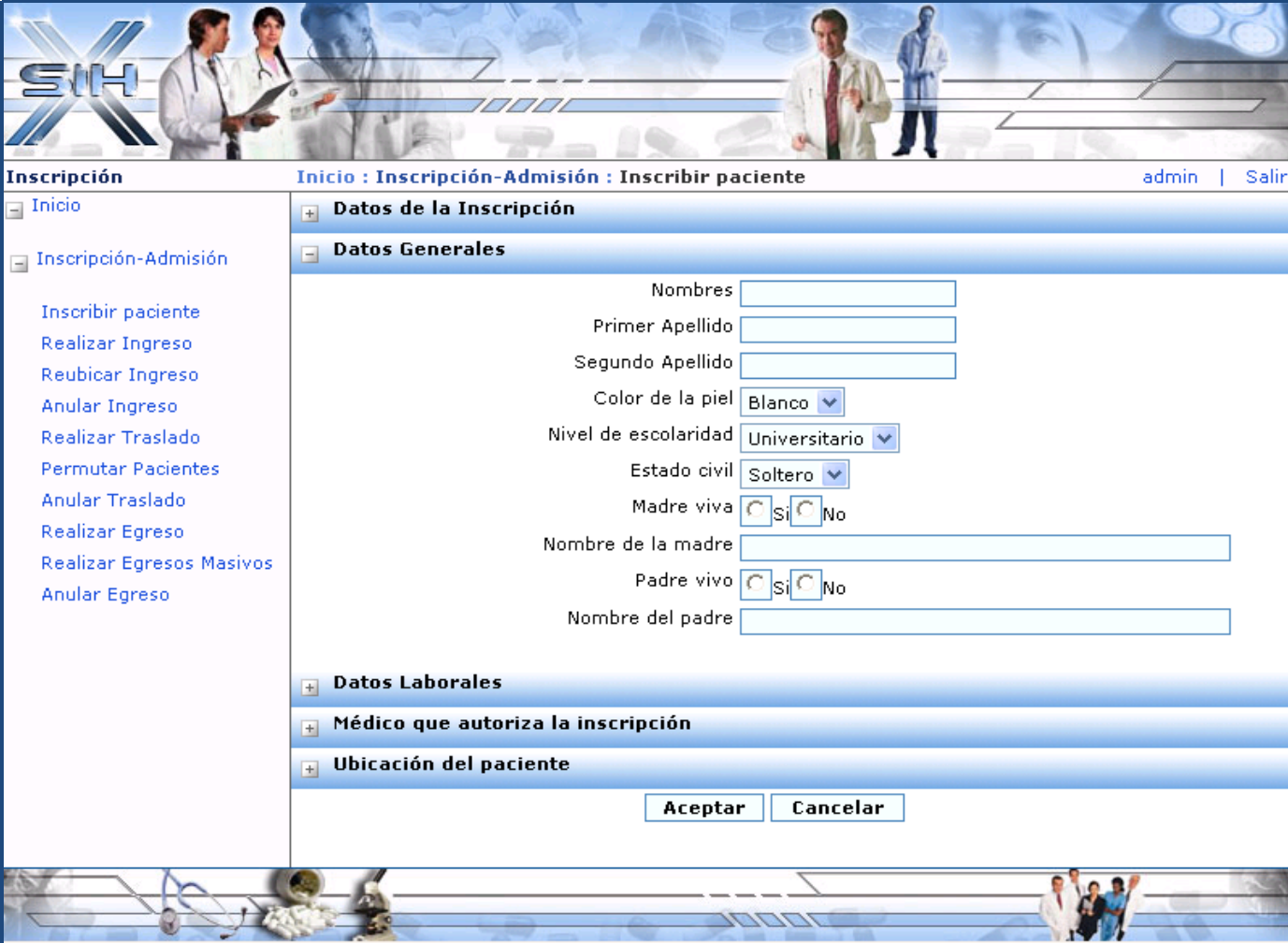
Datos Generales

Datos Laborales

Médico que autoriza la inscripción

Ubicación del paciente

- Datos generales



The screenshot shows a web-based interface for patient registration. At the top, there is a banner with the logo 'SIH' and images of medical professionals. Below the banner, the page title is 'Inicio : Inscripción-Admisión : Inscribir paciente', with 'admin | Salir' on the right. A left sidebar contains a menu with 'Inicio' and 'Inscripción-Admisión' expanded, listing various actions like 'Inscribir paciente', 'Realizar Ingreso', etc. The main content area is titled 'Datos de la Inscripción' and contains several sections: 'Datos Generales' with fields for names, surnames, skin color, education level, marital status, and parental status; 'Datos Laborales'; 'Médico que autoriza la inscripción'; and 'Ubicación del paciente'. At the bottom of the form are 'Aceptar' and 'Cancelar' buttons.

Inscripción Inicio : Inscripción-Admisión : Inscribir paciente admin | Salir

Datos de la Inscripción

Datos Generales

Nombres

Primer Apellido

Segundo Apellido

Color de la piel

Nivel de escolaridad

Estado civil

Madre viva Si No

Nombre de la madre

Padre vivo Si No

Nombre del padre

Datos Laborales

Médico que autoriza la inscripción

Ubicación del paciente

- Datos laborales y médico que autoriza la inscripción

Inscripción Inicio : Inscripción-Admisión : Inscribir paciente admin | Salir

Inscripción-Admisión

- Inscribir paciente
- Realizar Ingreso
- Reubicar Ingreso
- Anular Ingreso
- Realizar Traslado
- Permutar Pacientes
- Anular Traslado
- Realizar Egreso
- Realizar Egresos Masivos
- Anular Egreso

Datos de la Inscripción

Datos Generales

Datos Laborales

Centro de trabajo

Dirección del centro de trabajo

Ocupación

Médico que autoriza la inscripción

Registro Profesional

Ubicación del paciente

- Ubicación del paciente

Inscripción Inicio : Inscripción-Admisión : Inscribir paciente admin | Salir

Inscripción-Admisión

- Inscribir paciente
- Realizar Ingreso
- Reubicar Ingreso
- Anular Ingreso
- Realizar Traslado
- Permutar Pacientes
- Anular Traslado
- Realizar Egreso
- Realizar Egresos Masivos
- Anular Egreso

Datos de la Inscripción

Datos Generales

Datos Laborales

Médico que autoriza la inscripción

Ubicación del paciente

Dirección

Teléfono

Nacionalidad

País

Provincia

Municipio

Área de salud

Trabajador del Hospital

Con Historia en archivo

Aceptar **Cancelar**

Realizar ingreso

- Registrar ingreso y datos de ingreso

The screenshot shows a web-based interface for a hospital management system. At the top, there is a banner with the logo 'SIH' and images of medical professionals. Below the banner, the page title is 'Inscripción-Admisión : Realizar Ingreso' and the user is logged in as 'admin'. A left sidebar contains a menu with options like 'Inscribir paciente', 'Realizar Ingreso', and 'Reubicar Ingreso'. The main content area is titled 'Registrar ingreso' and contains several form fields: 'Historia clínica' (with a search icon), 'Nombre y apellidos', 'Fecha de nacimiento', 'Edad', 'Sexo' (set to 'Femenino'), and 'Nro Identidad'. Below this is a 'Datos Ingreso' section with 'Fecha ingreso' (with a calendar icon), 'Tipo de ingreso' (set to 'Programado'), and 'Hora' (set to '4:14:27 PM'). There are also expandable sections for 'Orden de ingreso' and 'Ubicación Paciente'. At the bottom of the form are 'Aceptar' and 'Cancelar' buttons.

Inscripción Inicio : Inscripción-Admisión : Realizar Ingreso admin | Salir

Registrar ingreso

Historia clínica 🔍

Nombre y apellidos

Fecha de nacimiento

Edad

Sexo **Femenino** ▼

Nro Identidad

Datos Ingreso

Fecha ingreso 📅

Tipo de ingreso **Programado** ▼

Hora **4:14:27 PM**

Orden de ingreso

Ubicación Paciente

- **Buscar paciente**

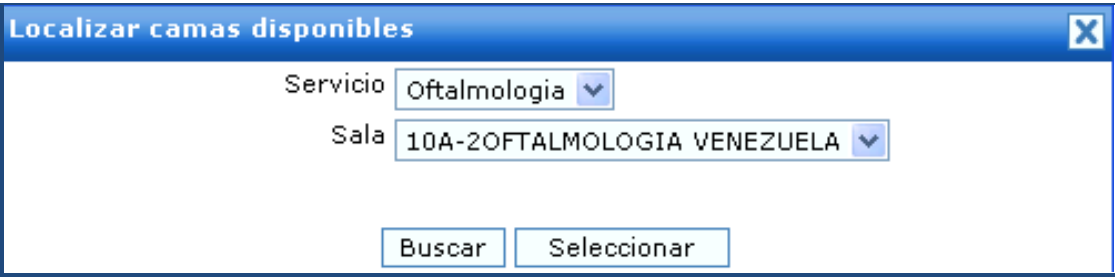


The screenshot shows a dialog box titled "Buscar paciente" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Nombre:** A dropdown menu showing "x..." and an adjacent text input field.
- Historia clínica:** A dropdown menu showing "=" and an adjacent text input field.
- Fecha de nacimiento:** A text input field followed by a calendar icon.
- Area de salud:** A dropdown menu showing "#1 Trinidad".
- Sexo:** A dropdown menu showing "Femenino".

At the bottom of the dialog, there are two buttons: "Buscar" and "Seleccionar".

- **Localizar camas disponibles**

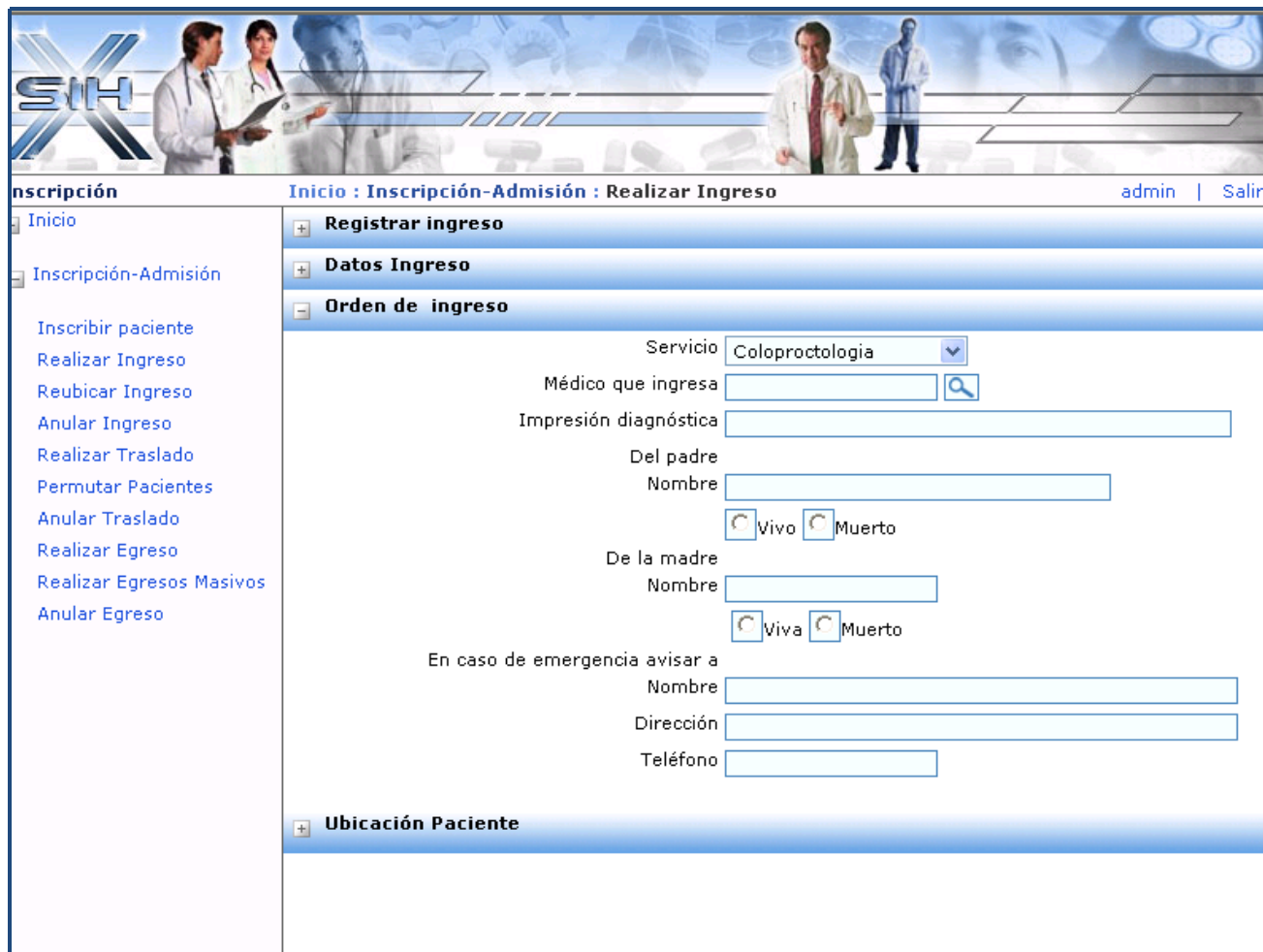


The screenshot shows a dialog box titled "Localizar camas disponibles" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Servicio:** A dropdown menu showing "Oftalmologia".
- Sala:** A dropdown menu showing "10A-2OFTALMOLOGIA VENEZUELA".

At the bottom of the dialog, there are two buttons: "Buscar" and "Seleccionar".

- Orden de ingreso



The screenshot shows a web-based interface for a hospital management system. At the top, there is a banner with the letters 'SIH' and images of medical professionals. Below the banner, the page title is 'Inicio : Inscripción-Admisión : Realizar Ingreso'. In the top right corner, there are links for 'admin' and 'Salir'. On the left side, there is a navigation menu with the following items: 'Inicio', 'Inscripción-Admisión', 'Inscribir paciente', 'Realizar Ingreso', 'Reubicar Ingreso', 'Anular Ingreso', 'Realizar Traslado', 'Permutar Pacientes', 'Anular Traslado', 'Realizar Egreso', 'Realizar Egresos Masivos', and 'Anular Egreso'. The main content area is titled 'Registrar ingreso' and contains several sections: 'Datos Ingreso' and 'Orden de ingreso'. The 'Orden de ingreso' section includes the following fields: 'Servicio' (a dropdown menu set to 'Coloproctología'), 'Médico que ingresa' (a text input field with a search icon), 'Impresión diagnóstica' (a long text input field), 'Del padre' (a section with a 'Nombre' text input field and radio buttons for 'Vivo' and 'Muerto'), 'De la madre' (a section with a 'Nombre' text input field and radio buttons for 'Viva' and 'Muerto'), 'En caso de emergencia avisar a' (a section with 'Nombre', 'Dirección', and 'Teléfono' text input fields), and 'Ubicación Paciente' (a section with a text input field).

- Ubicación ingreso

The screenshot shows a web application interface for a hospital management system. At the top, there is a banner with the logo 'SIH' and images of medical professionals. Below the banner, the page title is 'Inicio : Inscripción-Admisión : Realizar Ingreso' and the user is logged in as 'admin'. A left sidebar menu contains options like 'Inscripción-Admisión', 'Inscribir paciente', 'Realizar Ingreso', 'Reubicar Ingreso', 'Anular Ingreso', 'Realizar Traslado', 'Permutar Pacientes', 'Anular Traslado', 'Realizar Egreso', 'Realizar Egresos Masivos', and 'Anular Egreso'. The main content area has a navigation pane with 'Registrar ingreso', 'Datos Ingreso', 'Orden de ingreso', and 'Ubicación Paciente' (selected). The 'Ubicación Paciente' section contains input fields for 'Sala' (with a search icon) and 'Cama', a checkbox for 'Fuera de servicio', and 'Aceptar' and 'Cancelar' buttons.

Inicio : Inscripción-Admisión : Realizar Ingreso admin | Salir

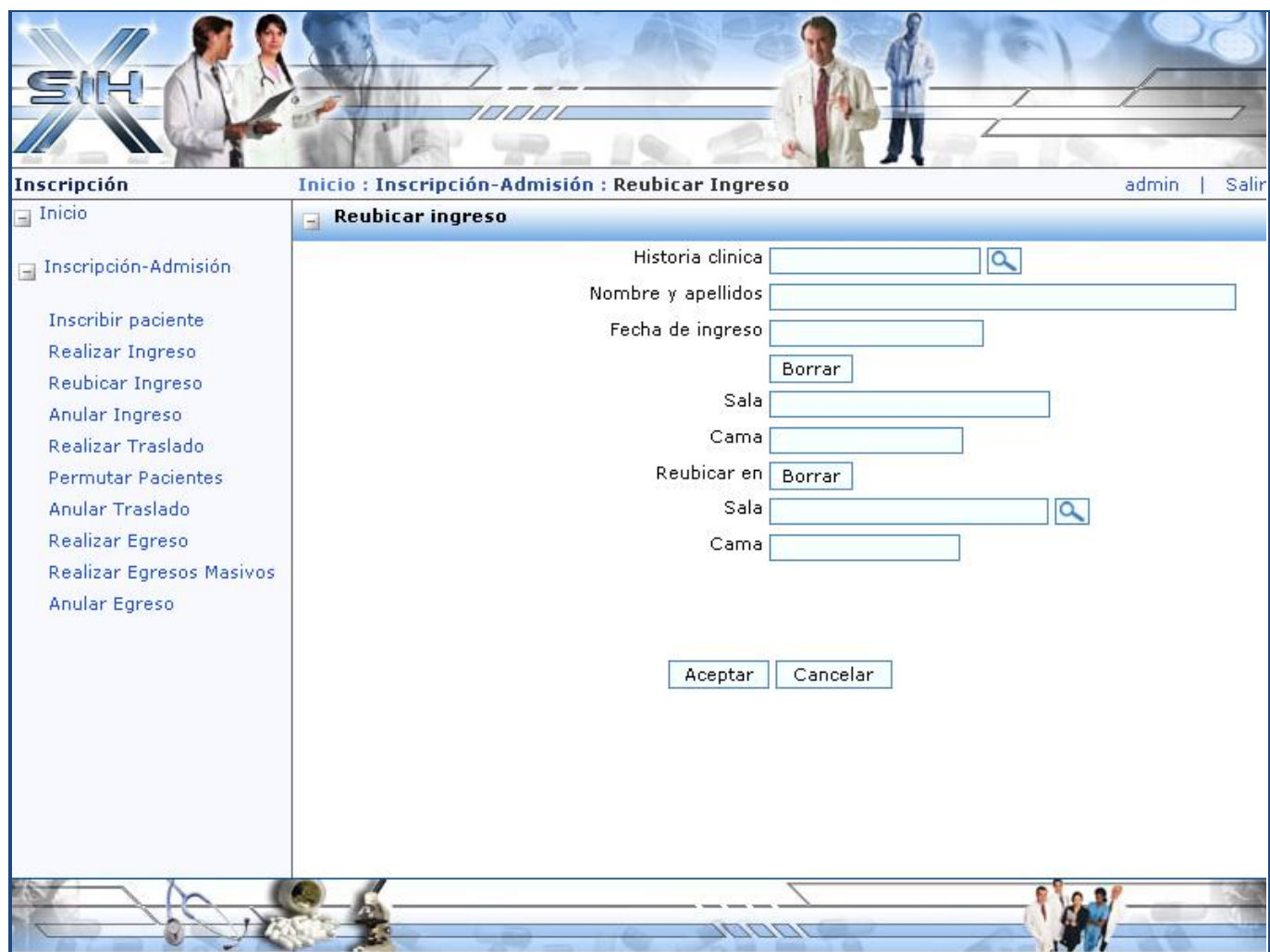
Ubicación Paciente

Sala

Cama

Fuera de servicio

- Reubicar ingreso



Inscripción Inicio : Inscripción-Admisión : Reubicar Ingreso admin | Salir

Inscripción-Admisión

- Inscribir paciente
- Realizar Ingreso
- Reubicar Ingreso
- Anular Ingreso
- Realizar Traslado
- Permutar Pacientes
- Anular Traslado
- Realizar Egreso
- Realizar Egresos Masivos
- Anular Egreso

Reubicar ingreso

Historia clinica 🔍

Nombre y apellidos

Fecha de ingreso

Sala

Cama

Reubicar en

Sala 🔍

Cama

- Anular ingreso

The screenshot displays a web-based interface for a hospital management system. At the top, there is a banner with a blue background and images of medical professionals. Below the banner, the page title is "Inscripción" and the breadcrumb trail is "Inicio : Inscripción-Admisión : Anular Ingreso". The user is logged in as "admin" and has a "Salir" link. A left sidebar menu is expanded to "Inscripción-Admisión", showing a list of actions: Inscribir paciente, Realizar Ingreso, Reubicar Ingreso, Anular Ingreso, Realizar Traslado, Permutar Pacientes, Anular Traslado, Realizar Egreso, Realizar Egresos Masivos, and Anular Egreso. The main content area is titled "Anular ingreso" and contains a form with the following fields: "Historia clinica" (with a search icon), "Nombre y apellidos", "Fecha de ingreso", "Ubicación actual" (with sub-fields for "Sala" and "Cama"). At the bottom of the form are "Aceptar" and "Cancelar" buttons. The interface is framed by a decorative border with medical icons at the bottom.

- Realizar traslado

The screenshot shows a web-based interface for a hospital management system. At the top, there is a banner with the letters 'SIH' and images of medical professionals. Below the banner, the page title is 'Inicio : Inscripción-Admisión : Realizar Traslado'. In the top right corner, there are links for 'admin' and 'Salir'. On the left side, there is a navigation menu with the following items: Inicio, Inscripción-Admisión, Inscribir paciente, Realizar Ingreso, Reubicar Ingreso, Anular Ingreso, Realizar Traslado, Permutar Pacientes, Anular Traslado, Realizar Egreso, Realizar Egresos Masivos, and Anular Egreso. The main content area is titled 'Datos del Traslado' and contains the following fields: Historia Clínica (with a search icon), Nombre (text input), Fecha de Ingreso (text input), Ubicación actual (text input), Sala (text input), Cama (text input), Fecha del Traslado (calendar icon), Hora del Traslado (text input, showing '10:35:59 PM'), Trasladar a (text input), Sala (with a search icon), Cama (text input), and a checkbox for 'Fuera de servicio'. Below these fields is a section titled 'Médico que autoriza el Traslado' with a 'Registro Profesional' field (with a search icon) and a text input field. At the bottom of the form, there are two buttons: 'Aceptar' and 'Cancelar'. The interface is decorated with medical-themed images at the top and bottom.

- Permutar pacientes

The screenshot shows a web-based interface for a hospital management system. At the top, there is a banner with the logo 'SIH' and images of medical professionals. Below the banner, the navigation bar includes 'Inscripción', 'Inicio : Inscripción-Admisión : Permutar Pacientes', and user options 'admin | Salir'. A left sidebar menu is expanded to 'Inscripción-Admisión', listing various actions such as 'Inscribir paciente', 'Realizar Ingreso', 'Reubicar Ingreso', 'Anular Ingreso', 'Realizar Traslado', 'Permutar Pacientes', 'Anular Traslado', 'Realizar Egreso', 'Realizar Egresos Masivos', and 'Anular Egreso'. The main content area is titled 'Datos de Pacientes a permutar de cama' and is divided into two sections: 'Primer Paciente' and 'Segundo Paciente'. Each section contains input fields for 'Historia Clínica' (with a search icon), 'Nombre', 'Fecha de Ingreso', 'Fecha del Traslado', 'Sala', and 'Cama'. At the bottom of the form, there is a section for 'Datos de la permuta' with 'Aceptar' and 'Cancelar' buttons.

Inscripción Inicio : Inscripción-Admisión : Permutar Pacientes admin | Salir

Datos de Pacientes a permutar de cama

Primer Paciente

Historia Clínica

Nombre

Fecha de Ingreso

Fecha del Traslado

Sala

Cama

Segundo Paciente

Historia Clínica

Nombre

Fecha de Ingreso

Fecha del Traslado

Sala

Cama

Datos de la permuta

- Datos de la permuta

The screenshot shows a web-based interface for a hospital management system. At the top, there is a banner with the logo 'SIH' and images of medical professionals. Below the banner, the page title is 'Inicio : Inscripción-Admisión : Permutar Pacientes' and the user is logged in as 'admin' with a 'Salir' link.

The main content area is divided into two sections:

- Datos de Pacientes a permutar de cama**: This section is currently collapsed.
- Datos de la permuta**: This section is expanded and contains the following fields:
 - Fecha de la Permuta**: A text input field with a calendar icon to its right.
 - Hora**: A text input field containing the value '10:42:37 PM'.

At the bottom of the form, there are two buttons: **Aceptar** and **Cancelar**.

On the left side of the interface, there is a navigation menu under the heading 'Inscripción-Admisión' with the following options:

- Inscribir paciente
- Realizar Ingreso
- Reubicar Ingreso
- Anular Ingreso
- Realizar Traslado
- Permutar Pacientes
- Anular Traslado
- Realizar Egreso
- Realizar Egresos Masivos
- Anular Egreso

- Anular traslado

The screenshot shows a web-based interface for a hospital management system. At the top, there is a banner with the logo 'SIH' and images of medical professionals. Below the banner, the page title is 'Inicio : Inscripción-Admisión : Anular Traslado'. In the top right corner, there are links for 'admin' and 'Salir'. On the left side, there is a navigation menu with the following items: 'Inicio', 'Inscripción-Admisión', 'Inscribir paciente', 'Realizar Ingreso', 'Reubicar Ingreso', 'Anular Ingreso', 'Realizar Traslado', 'Permutar Pacientes', 'Anular Traslado', 'Realizar Egreso', 'Realizar Egresos Masivos', and 'Anular Egreso'. The main content area is titled 'Datos del Traslado' and contains the following form fields: 'Historia Clínica' (with a search icon), 'Nombre', 'Trasladado desde' (with sub-fields for 'Sala' and 'Cama'), 'Trasladado hacia' (with sub-fields for 'Sala' and 'Cama'), 'Fecha del Traslado', and 'Hora del Traslado'. At the bottom of the form, there are two buttons: 'Aceptar' and 'Cancelar'. The interface is decorated with medical-themed images at the top and bottom.

- Realizar egreso

The screenshot shows a web-based interface for a hospital management system. At the top, there is a banner with the logo 'SIH' and images of medical professionals. Below the banner, the page title is 'Inicio : Inscripción-Admisión : Realizar Egreso' and the user is logged in as 'admin'. A navigation menu on the left lists various actions under 'Inscripción-Admisión', with 'Realizar Egreso' highlighted. The main content area is titled 'Datos del Egreso' and contains several input fields: 'Historia Clínica' (with a search icon), 'Nombre', 'Fecha de Ingreso', 'Sala', 'Cama', 'Fecha del Egreso' (with a calendar icon), 'Hora del Egreso' (set to 10:52:50 PM), and 'Tipo de Salida' (set to 'Alta'). There is also a checkbox for 'Necropsia'. Below this section is another section titled 'Médico que autoriza el egreso' with a 'Registro Profesional' field and a search icon. At the bottom of this section are 'Aceptar' and 'Cancelar' buttons. The interface is decorated with medical-themed images at the top and bottom.

Inscripción Inicio : Inscripción-Admisión : Realizar Egreso admin | Salir

Datos del Egreso

Historia Clínica

Nombre

Fecha de Ingreso

Sala

Cama

Fecha del Egreso

Hora del Egreso

Tipo de Salida

Necropsia

Médico que autoriza el egreso

Registro Profesional

- Realizar egresos masivos

The screenshot shows a web-based interface for a hospital management system. At the top, there is a banner with the logo 'SIH' and images of medical professionals. Below the banner, the page title is 'Inicio : Inscripción-Admisión : Realizar Egresos Masivos'. In the top right corner, there are links for 'admin' and 'Salir'. On the left side, there is a navigation menu with the following items: Inicio, Inscripción-Admisión, Inscribir paciente, Realizar Ingreso, Reubicar Ingreso, Anular Ingreso, Realizar Traslado, Permutar Pacientes, Anular Traslado, Realizar Egreso, Realizar Egresos Masivos, and Anular Egreso. The main content area is titled 'Datos del egreso masivo' and contains the following fields: 'Sala' with a dropdown menu set to '10A-2OFTALMOLOGIA VENEZUELA' and a 'Buscar' button; 'Fecha salida' with a text input field and a calendar icon; and 'Hora' with a text input field set to '10:55:26 PM'. Below this, there is a section titled 'Médico que autoriza el egreso' with a 'Registro Profesional' field containing a search icon and a long empty text input field. At the bottom of this section are two buttons: 'Aceptar' and 'Cancelar'. The interface is decorated with medical-themed images at the top and bottom.

- Anular egreso

Inscripción Inicio : Inscripción-Admisión : Anular Egreso admin | Salir

Inscripción-Admisión

- Inscribir paciente
- Realizar Ingreso
- Reubicar Ingreso
- Anular Ingreso
- Realizar Traslado
- Permutar Pacientes
- Anular Traslado
- Realizar Egreso
- Realizar Egresos Masivos
- Anular Egreso

Anular Egreso

Historia Clínica 🔍

Nombre

Fecha del Egreso

Motivo del Egreso

Sala

Cama

Médico que autoriza el egreso

Registro Profesional 🔍

Aceptar **Cancelar**

Glosario de términos⁶

Firewall: Un firewall es un dispositivo que funciona como cortafuegos entre redes, permitiendo o denegando las transmisiones de una red a la otra. Un uso típico es situarlo entre una red local y la red Internet, como dispositivo de seguridad para evitar que los intrusos puedan acceder a información confidencial. [24]

Browser: Un navegador web o explorador web (del inglés, navigator o browser) es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en HTML, desde servidores web de todo el mundo a través de Internet.

Framework: 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 software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Interface: Se le llama así a las clases que definen métodos y atributos abstractos y que luego son utilizados por otras clases para su funcionamiento. En lenguajes de programación como Java son muy utilizados.

Data warehouse: Un Data Warehouse es una colección de datos en la cual se encuentra integrada la información de una Institución y que se usa como soporte para el proceso de toma de decisiones gerenciales. [25]

HTTP: El protocolo de transferencia de hipertexto (*HTTP, HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web (WWW). 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 acceso a una página y la respuesta con el contenido. También sirve el protocolo para enviar 'información adicional en ambos sentidos, como formularios con campos de texto.

⁶ Para elaborar este glosario fue utilizada la Wikipedia <http://es.wikipedia.org> y otras fuentes incluidas en las referencias bibliográficas.

HTML: El HTML, acrónimo inglés de HyperText Markup Language (lenguaje de marcas hipertextuales), lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web.

World Wide Web Consortium: El Consorcio World Wide Web (W3C) es un consorcio internacional donde las organizaciones miembros, personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web. [26]

SGML: (Standard Generalized Markup Language - Lenguaje de Marcado de Anotaciones Generales). Es un metalenguaje (Lenguaje utilizado para referirse a otro lenguaje) de donde deriva el HTML y el XML. [27]

DCOM: El Modelo de Objetos de Componentes Distribuidos (Distributed Component Object Model), originalmente conocido como “Network OLE”, es un protocolo de interconexión que permite que los componentes software se comuniquen de una forma fiable, seguro y eficiente sobre redes de ordenadores utilizando protocolos tales como HTTP. [28]

CORBA: (Common Object Request Broker Architecture) establece estándares para la comunicación de objetos a través de procedimientos/métodos remotos. [29]

Prototipo no funcional: El prototipo no funcional corresponde al diseño de las pantallas acorde con el estándar definido. Dicho prototipo deberá incluirse dentro del menú principal del sistema, en la ubicación correspondiente según al módulo que corresponda. [30]

Arquitectura Cliente – Servidor: La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. [31]

Base de datos: (DataBase). Conjunto de datos relacionados que se almacenan de forma que se pueda acceder a ellos de manera sencilla, con la posibilidad de relacionarlos, ordenarlos en base a diferentes criterios, etc. [32]

Gestor de Base de Datos: Es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. [33]

Sistemas operativos: Conjunto de programas que se integran con el hardware para facilitar al usuario, el aprovechamiento de los recursos disponibles. [34]

TCP/IP: Transmission Control Protocol (TCP) y el Internet Protocol (IP). Protocolos que sirven para enlazar computadoras que utilizan diferentes sistemas operativos. [35]

OASIS: OASIS es un consorcio internacional sin fines de lucro que orienta el desarrollo, la convergencia y la adopción de estándares en el desarrollo de aplicaciones web y ofimáticas. [36]

Hojas de estilo: Las hojas de estilo (style sheets) son conjuntos de instrucciones, a veces en forma de archivo anexo, que se asocian a los archivos de texto y se ocupan de los aspectos de formato y de presentación de los contenidos: tipo, fuente y tamaño de letras, justificación del texto, colores y fondos, etc.

Servidor: es una computadora central, de gran capacidad, compartida por las otras computadoras de la red, llamadas Clientes o estaciones de trabajo (workstations), ya que reciben el servicio de almacenar, controlar y compartir la información contenida en el servidor. [37]

DB2: Es una familia de productos gestores de base de datos relacionales libres de la IBM que sirven a varias plataformas de sistemas operativos. [38]

Oracle: Tipo servidor para bases de datos. [39]

Clases: abstracciones que representan a un conjunto de objetos con un comportamiento e interfaz común. [40]

Herencia: se define la herencia como una jerarquía de extracciones, y la relación entre clases, donde se comparte la estructura y el comportamiento de una o más clase considerada como clases superiores o una superclase, con lo cual se resume que la herencia es una unidad independiente por sí misma heredada de una abstracción o superclase. Un ejemplo cotidiano lo encontramos en las aplicaciones que existen actualmente en el mercado, donde un formulario cualquiera hereda las características de una ventana del sistema operativo Windows (Maximizar, Minimizar, Cerrar). [41]

Funciones: conjunto de instrucciones que permiten procesar las variables para obtener un resultado. [42]

Restricciones (Constraints): Constraints son restricciones en alguna columna en la definición de una tabla. [43]

Disparadores (triggers): Un disparador define una acción que la base de datos debe llevar a cabo cuando se produce algún suceso relacionado con la misma. [44]

Reglas (rules): Las reglas son capacidades que realizan algunas de las mismas funcionalidades que las restricciones CHECK. Las restricciones CHECK son el modo preferido y estándar de restringir valores para una columna. [45]

Integridad transaccional: recuperación de error de sistema no planeado. [46]

Orientado a objetos: Significa que el software se organiza como una colección de objetos discretos que contiene tanto estructura de datos como también un comportamiento. [47]

ECMA: organismos de estándares fundado en 1961. [48]

Compiladores: Un compilador es una aplicación que toma un programa escrito en un lenguaje de alto nivel y lo convierte a un programa escrito en un lenguaje que la maquina puede entender. [49]

Licencia LGPL: Es una licencia de software libre (LESSER GENERAL PUBLIC LICENCE). [50]

Lenguaje interpretado: un lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados. También se les conoce como lenguajes de script.

OMG: (Object Management Group) se formó en 1989 con el propósito de crear una arquitectura estándar para objetos distribuidos en redes (componentes). En Octubre de 1989 empezó a funcionar como una corporación independiente y sin ánimo de lucro. El compromiso asumido por el OMG busca el desarrollo de especificaciones para la industria del software que sean técnicamente ``excelentes'', comercialmente viables e independientes del vendedor. [51]

Metodología RAD: (Rapid Application Development) Denominación genérica para técnicas y herramientas de desarrollo de software que permiten el desarrollo rápido de aplicaciones. [52]

Plataforma de desarrollo: es el entorno común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada directamente a un sistema operativo, sin embargo, también es posible encontrarlas ligadas a una familia de lenguajes de programación o a una Interfaz de programación de aplicaciones o API por sus siglas en inglés.

Librerías: una librería o biblioteca es un conjunto de procedimientos y *funciones* (subprogramas) agrupadas en un archivo con el fin de ser aprovechadas por otros programas.

Frames: en castellano marcos- son una manera de partir la página en distintos espacios independientes los unos de los otros, de modo que en cada espacio se coloca una página distinta que se codifica en un fichero HTML distinto. [53]

Clases abstractas: Una clase que declara la existencia de métodos pero no la implementación de dichos métodos. [54]