



Universidad de las Ciencias Informáticas

Análisis y Diseño del Módulo de Archivos del
Sistema de Información Hospitalaria.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

AUTORES:

M^a Isabel Morales Galañena.

Dariel L. Naveda Díaz.

TUTOR:

Ing. Paúl Pérez Zurita.

Ciudad de La Habana
Julio, 2007

DECLARACIÓN DE AUTORÍA

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

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

Firma del Autor

Ma. Isabel Morales Galañena

Firma del Autor

Dariel L. Naveda Díaz

Firma del Tutor

Ing. Paul Pérez Zurita

*En la sociedad de la información y del conocimiento,
una sanidad de calidad tiene como requisito imprescindible
una gestión excelente de ambos.*

Anónimo

***A todas aquellas personas que han hecho posible
el logro de un sueño.***

María Isabel

A toda mi familia.

Dariel L.

Agradecimientos

De María Isabel

A mis padres y a abuelos por la vida y el amor, por su apoyo en cada paso y los desvelos causados.

A Titico por estar atento en cada etapa importante de mi vida como estudiante, por preocuparse por temas casi desconocidos para él y estar siempre en todo lo que yo necesite.

A mi tía Norma por sus incontables consejos.

A Normi por enseñarme la clave del éxito.

A Yamil por estar a mi lado en cada adversidad y demostrarme que todo llega algún día.

A Javier y Doraidita por siempre estar recordándome la meta de esta vida.

A Mechi y Norma quienes también han sido espectadoras de esta odisea.

A Reynel Fals del Pedro por su ayuda incondicional.

A Maylén, Yanelys, Yudelky, Dayamis y Yudy por su amistad.

En fin a toda mi familia y amigos que han aportado su granito de arena para poder lograr subir este escalón.

De Dariel L.

A mi mamá por haberme dado la luz, por estar siempre a mi lado y por quererme como a nadie.

A mi papá por siempre ayudarme en los peores momentos.

A mi novia por estar conmigo en los momentos más difíciles.

A mi abuela (Nena) por quererme mucho y por ser siempre comprensiva conmigo.

A Homero y Titi porque siempre me han ayudado en todo lo que he necesitado, y siempre me han querido como a un hijo.

A mis tíos Elenita y Fernando por quererme como a sus hijos.

Y especialmente a mi abuelo (Tomasín) que aunque no este presente físicamente, siempre lo recuerdo con mucho cariño.

Resumen

El presente documento recoge los resultados de la investigación realizada para el desarrollo posterior de un sistema informático. El objetivo de esta es realizar el análisis y diseño de las principales actividades que componen el módulo de registros médicos del paciente, que actualmente no cuenta con un soporte informático en los hospitales cubanos.

Para el desarrollo de estos flujos de trabajo se siguieron los pasos que propone el *Rational Unified Process* (RUP), como metodología de desarrollo y como herramienta Rational Rose con el lenguaje de modelado UML. Por lo que este trabajo contiene los principales artefactos obtenidos de los flujos de trabajo de Modelación del Negocio, Requerimientos, Análisis y Diseño. Permitiendo así la abstracción a una futura versión de la implementación y un futuro diseño de la base de dato.

Con el sistema se espera cumplir las políticas de migración del MINSAP y la UCI, empleando herramientas de *software* libre y de código abierto con el objetivo de disminuir los costos en licencias. Se especifica todo el flujo de trabajo que se realiza en los departamentos de Archivo/HC, garantizando una mejor gestión y rapidez en la búsqueda de las mismas, así como un mejor control de los movimientos de entrada/salida en el departamento. La aplicación contribuirá a lograr mayor eficiencia y control.

INTRODUCCIÓN.....	1
1. FUNDAMENTACIÓN TEÓRICA.....	5
1.1 SISTEMAS DE GESTIÓN DE ARCHIVOS.....	5
1.2 SISTEMAS DE INFORMACIÓN HOSPITALARIA EXISTENTES.	6
1.3 APLICACIONES.	9
1.3.1 Aplicaciones de escritorio.	9
1.3.2 Aplicaciones Web.	10
1.4 PLATAFORMAS Y LENGUAJES.	11
1.5 SOFTWARE LIBRE.	12
1.6 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.	12
1.3.5.1 Características de RUP.	14
1.3.5.2 La vida de un producto de software según RUP.	15
1.3.5.3 El Lenguaje Unificado del Modelado (UML).....	16
1.4 JUSTIFICACIÓN DE LA PROPUESTA DE SOLUCIÓN.	17
1.5 CONCLUSIONES.....	18
2. CARACTERÍSTICAS DEL SISTEMA.....	19
2.1 FLUJO ACTUAL DE LOS PROCESOS INVOLUCRADOS EN EL CAMPO DE ACCIÓN.	19
2.2 OBJETO DE AUTOMATIZACIÓN.....	20
2.3 MODELO DEL NEGOCIO PROPUESTO.....	21
2.4 ESPECIFICACIÓN DE LOS REQUERIMIENTOS DE SOFTWARE.....	23
2.4.1 Requerimientos funcionales.....	23
2.4.2 Requerimientos no funcionales.....	24
2.5 MODELO DEL SISTEMA.....	25
2.6 CONCLUSIONES.....	26
3. ANÁLISIS Y DISEÑO DEL SISTEMA.....	27
3.1 MODELO DE ANÁLISIS.	27
3.1.1 Diagramas de Clases del Análisis.....	27
3.2 MODELO DE DISEÑO.....	28
3.2.1 Diagrama de clases de diseño.....	28
3.2.2 Diagramas de interacción.....	29
3.3 PATRONES.....	29
3.3.1 Patrón de Diseño.....	29
3.3.2 Patrón de Arquitectura.....	30
3.3.2.1 Modelo Cliente servidor.....	30
3.3.2.2 Arquitectura de 3 capas.....	30
3.4 DESCRIPCIÓN DE CLASES.....	30
3.4.1 Clases Interfaz.....	30
3.4.2 Clases Controladoras.....	35
3.4.2 Clases entidad.....	40

3.5 PROTOTIPO DE INTERFAZ DE USUARIO.....	49
3.6 CONCLUSIONES	50
CONCLUSIONES GENERALES.....	51
RECOMENDACIONES.....	52
BIBLIOGRAFÍA.....	53
ANEXOS.....	54
ANEXO I MODELADO DEL NEGOCIO PROPUESTO	54
<i>Actores / Trabajadores del negocio.....</i>	54
<i>Diagrama de Casos de Uso del Negocio.....</i>	55
<i>Diagramas de Actividades.....</i>	56
<i>Modelo de Objetos.....</i>	59
ANEXO II MODELADO DEL SISTEMA	61
<i>Actores del sistema. Justificación</i>	61
<i>Casos de uso por ciclos de desarrollo.....</i>	61
<i>Listado de los casos de uso del sistema.....</i>	62
<i>Modelo de casos de uso del Sistema.....</i>	64
<i>Descripción de los Casos de Uso.....</i>	65
ANEXO III ANÁLISIS Y DISEÑO DEL SISTEMA.....	71
<i>Diagrama de Clases del Análisis.....</i>	71
<i>Diagramas de Clases del Diseño.....</i>	76
<i>Diagramas de Interacción del Diseño.....</i>	84
Diagramas de Secuencia del Diseño.....	84
ANEXO IV INTERFAZ DE LA APLICACIÓN.....	106
GLOSARIO.....	112

Introducción

Durante los últimos 20 años un grupo de instituciones cubanas han desarrollado sistemas encaminados a lograr determinados niveles de informatización de la salud. Estas soluciones carecían de integración y de una definición generalizable, aparte de que no existían los recursos tecnológicos necesarios para su ejecución en el Sistema Nacional de Salud (SNS).

A partir de 1997 se concibe una primera estrategia de informatización como respuesta del sector de la salud a los lineamientos estratégicos para la informatización de la sociedad cubana, con la finalidad de coordinar esfuerzos para el desarrollo de este proceso en el SNS.

En estos momentos se trabaja integradamente en el desarrollo de un grupo de aplicaciones básicas para la informatización del mismo.

Actualmente el Ministerio de Salud Pública (MINSAP) ha definido a la informatización como una de sus prioridades y ha convocado para ello a un grupo de instituciones propias de sector, del Ministerio de Informática y Comunicaciones y de otros organismos de la administración central del estado, para definir de conjunto la estrategia a desarrollar.

Algunas de las estrategias anteriormente empleadas se basaban en la reutilización de sistemas previos desarrollados en el país, trazándose un grupo de premisas y requisitos.

”Durante las décadas de los 80’s y los 90’s se realizaron múltiples esfuerzos para perfeccionar los Sistemas de Información Hospitalaria (SIH), específicamente en los procesos de recolección, distribución, interpretación y almacenamiento de los datos.” (MESSERLE 1999)

Aunque se han obtenido grandes progresos en esta empresa, los SIH actuales presentan todavía déficit con el cumplimiento de las actividades propias de los hospitales cubanos y la realidad es que se desarrollan numerosas versiones para mejorar continuamente los ya implementados.

Los SIH están compuestos por varios sistemas, entre ellos resaltan subsistemas que permiten acceder a la Historia Clínica (HC) del paciente, a recursos bibliográficos, así como también se incluyen actividades ambulatorias, monitorización de pacientes, alertas y herramientas de apoyo a la toma de decisiones tanto clínicas como administrativas. A continuación se muestra la **figura 1** para una mejor comprensión.

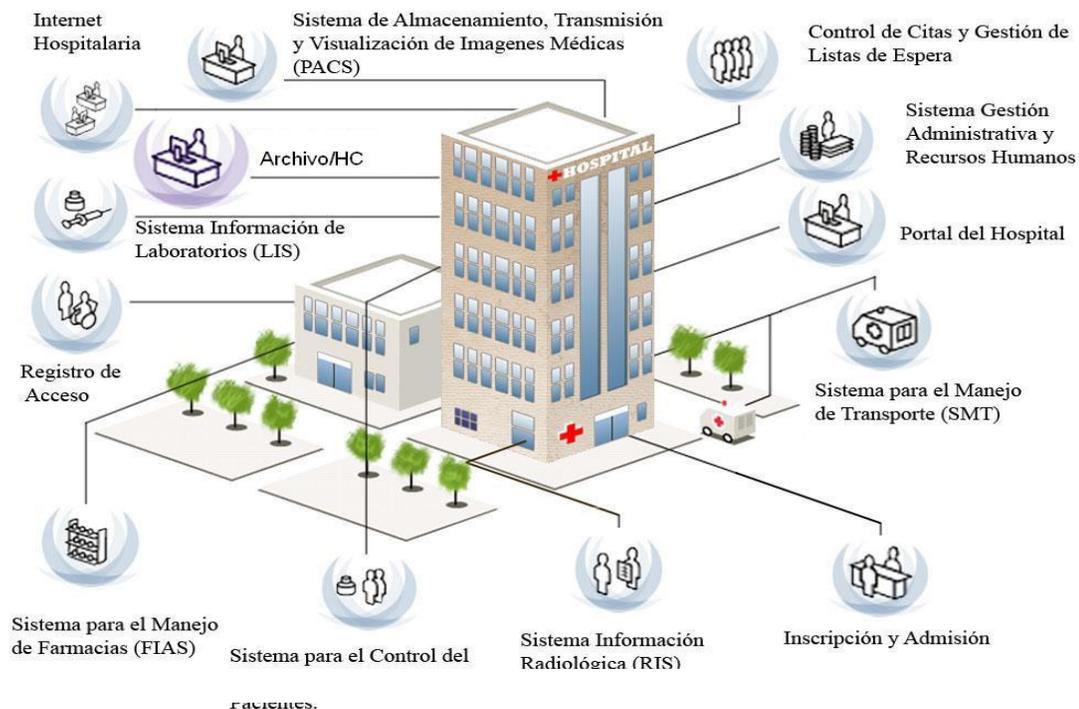


Figura 1 Sistemas que componen al SIH.

Al aplicar las Tecnologías de la Información y las Comunicaciones (TIC) en este campo, se pone en marcha la idea de un crecimiento y mejoramiento de las herramientas de trabajo del personal médico, reflejándose este avance en el crecimiento de las inversiones y prospectos de trabajos, fundiéndose así una sociedad capaz de dar respuesta a sus emergentes necesidades.

El eje fundamental y centro del proceso de informatización del sector lo constituye el paciente, quién será el principal beneficiado al garantizar las aplicaciones, la calidad, oportunidad y consistencia de la

información, lo que incrementará la efectividad y eficiencia de los procesos relacionados con la salud, lo que conlleva en un incremento continuo y sostenido de la calidad en la atención médica.

Esta investigación ayuda a la gestión de la información para el control de registros médicos a un sistema informático, llamado SIH hasta lograr una automatización de la HC.

La misma consiste en el análisis y diseño de una aplicación que controle las solicitudes tanto de entrada/salida de una HC correspondiente al Departamento de Archivos, como la clasificación y búsqueda de las HC. Además de su organización a partir del tiempo dentro de archivo y la ubicación que puede tener dentro del mismo para un mejor trabajo organizativo. Siendo imprescindible un estudio de los diferentes artefactos de análisis y diseño en términos de ingeniería de *software* y lograr una aplicación que solucione las necesidades actuales de nuestros centros hospitalarios.

Por eso a partir de las bondades de las TIC y enfrentar dentro de los archivos de HC, la proliferación vertiginosa y deterioro de los documentos generados en la práctica asistencial que influye en las instalaciones destinadas a archivos, estos se ven en riesgo de una mala organización de la información, pérdida de la misma por lo que su acceso se torna difícil, además de estar digitalizados no cuentan con tecnologías avanzadas para desarrollar estas acciones; todo esto trae como consecuencia problemas en el desarrollo del proceso médico del paciente, pues de esta información dependen las decisiones del médico.

De lo anteriormente expuesto, se deriva la necesidad del análisis y diseño de un sistema que automatice los procesos propios pertenecientes al módulo de registros médicos del paciente en los hospitales del país, en virtud de lo cual el **problema científico** puede definirse de la siguiente forma: *¿Cómo diseñar un sistema informático que optimice la gestión de la información en el departamento de registros médicos de los hospitales cubanos?*

En correspondencia con el problema, el **objeto de estudio** lo constituye *“Los procesos de gestión de la información de los pacientes en los centros hospitalarios de Cuba”*.

El **campo de acción** de este trabajo está centrado en la optimización de la gestión de la información de los pacientes en los departamentos de Archivo/HC en los centros hospitalarios de Cuba.

El **objetivo general** es diseñar una aplicación para el módulo de archivos que facilite la obtención de información de los pacientes en los centros hospitalarios de Cuba.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes tareas:

- ❖ Determinar el problema que existe y la situación actual. Diagnosticar los procesos de gestión de las HC en los hospitales cubanos.
- ❖ Realizar entrevistas al personal médico y laboral de los centros hospitalarios.
- ❖ Obtener los requisitos del sistema.
- ❖ Generar los Artefactos necesarios para realizar el análisis y diseño.

Estructuración del Contenido:

El Capítulo I titulado “Fundamentación teórica” ofrece los conceptos básicos asociados al negocio y los métodos científicos que se utilizaron para llevar a cabo el conocimiento del negocio.

Se brinda el estado del arte en cuanto a técnicas, tecnologías, metodologías y *software* usados en la actualidad o en las que se apoya para la solución del problema que se enfrenta.

El Capítulo II denominado “Características del sistema” plantea el modelo de negocio, los requisitos y casos de uso del sistema, así como la expansión de los casos de uso.

En el Capítulo III “Análisis y diseño del sistema” se muestra el diagrama de clases del análisis y se exponen los detalles relacionados con el diseño del sistema propuesto.

Cada capítulo se inicia con una breve introducción y finaliza con las conclusiones, en las que se plantean los resultados obtenidos.

Fundamentación Teórica

En el capítulo se darán a conocer conceptos básicos relacionados con el tema y el estado del arte en cuanto a metodologías y herramientas a utilizar en el desarrollo del trabajo.

1.1 Sistemas de Gestión de Archivos

Un sistema de gestión de archivos es aquel sistema informático que brinda servicios a los usuarios en el uso de archivos., haciendo cumplir una serie de objetivos, como optimizar el rendimiento, proveer un soporte de entrada/salida y así tener control de los datos archivados y minimizar o eliminar la pérdida o destrucción de datos.

Un sistema de gestión de archivo hospitalario no se diferencia de los objetivos principales de la gestión de un archivo pero como bien lo dice está guiado sobre el ambiente de los hospitales donde el principal objeto a organizar, almacenar y adquirir adecuadamente son las Historias Clínicas (HC) siendo estas “ (...) el elemento esencial de acreditación por parte del médico de su conducta con el paciente en todo momento, al reflejar toda la información relacionada con la asistencia dispensada al propio paciente.” (*Historia Clínica y Registros Médicos Informatizados*).

La HC presenta una gran importancia para la práctica médica donde la tecnología de la información nos ofrece una opción para hacer de nuestra HC un documento realmente eficiente y por tanto útil lo que nos proporciona la ventaja de resolver los problemas pertenecientes a los archivos de HC.

1.2 Sistemas de Información Hospitalaria existentes.

En la actualidad existen una gama de producciones diferenciadas de *software* para la salud, encaminadas a resolver disímiles problemáticas dentro de esta esfera.

La informatización de los hospitales ha favorecido el establecimiento de sistemas de información que, a la vez que aseguran la recogida ágil y segura de todos los datos que históricamente han permitido cuantificar la actividad realizada en los hospitales, también están facilitando el registro sistemático y la posterior recuperación con fines diversos de datos sobre las características de los procesos atendidos en ellos.

En otras palabras la posibilidad de disponer de tecnología con una capacidad de almacenamiento, tratamiento y recuperación de la información creciente, ha sentado las bases para incorporar a los sistemas informativos hospitalarios datos de tipo clínico, hasta hace muy poco casi siempre ausentes de los mismos.

Como ejemplos de *software* que se han desarrollado y comercializado a nivel internacional tenemos:

eMedicalRecs.com

Es un *software* en el que se dispone de un sistema de gestión de información que puede ser utilizado en consultorios o clínicas médicas donde podrá consultar y actualizar tanto las HC de sus pacientes como movimientos de dinero.

Gowin HC - Historia clínica electrónica

“GoWin está desarrollado sobre Windows y sigue sus estándares, presentando una apariencia familiar e intuitiva que facilita la adaptación del usuario a la nueva aplicación, lo que redundará en una reducción de los costes de formación.”(*Características del sistema Gowin HC*)

“GoWin.Net, por su parte, permite la búsqueda y consulta de historias clínicas y la citación online a través de un portal Web de Internet o Intranet. Su principal ventaja es que, al ser un producto basado en la Web, permite la realización remota de todas las operaciones por parte de los usuarios autorizados,

independientemente de la plataforma que éstos utilicen. Por supuesto, se han extremado las medidas para garantizar la seguridad del sistema.”

Gestión de Archivos BIOCOT

“BIOCOT posibilita organizar procesos de la información y de la documentación clínica, codificándola y garantizando el cumplimiento de las normas de la Administración sanitaria y de los sistemas de clasificación y codificación internacionales”(Característica del sistema de Gestión de Archivos BIOCOT.).

Care2x

“Care2x integra datos, funciones y flujo de tareas en un entorno de cuidados de la salud. Al momento, está compuesto de cuatro componentes principales. Cada uno de estos componentes también puede funcionar de manera individual.

- ❖ SIH - 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.

Se esfuerza por desarrollar el sistema de cuidados de la salud más útil y práctico y que se encuentra abierto para que otros lo desarrollen aún más.” (Características del Sistema Care2x Entorno Integrado de Cuidados de la Salud)

En Cuba se ha realizado un cambio fundamental en el conocimiento de la actividad que se lleva a cabo en los sistemas de gestión y está permitiendo por una parte a los profesionales sanitarios la recuperación de la información con fines asistenciales, docentes y de investigación clínica y por otra parte está facilitando a los gestores una herramienta que les ayuda a evaluar, comprender y analizar los recursos disponibles y las tomas de decisiones.

Además se ha mantenido desarrollando este tipo de sistemas controlado por el Centro de Desarrollo Informático para la Salud Pública (CEDISAP), contando con la ayuda de la Universidad de Ciencias Informáticas.

Estos sistemas permiten mejorar la calidad de la asistencia, optimizar el funcionamiento y gestión de las áreas, además permite que las aplicaciones informáticas de ámbito hospitalario sean dirigidas a la planificación óptima de recursos humanos, a la elaboración y el control de los planes de cuidados de los pacientes, a la evaluación de las cargas de trabajo y el control de costos operativos, a la optimización de la logística de las unidades de hospitalización y las comunicaciones intrahospitalarias.

Ejemplo de sistemas de información a nivel nacional tenemos:

Galen

Es un sistema automatizado para los egresos hospitalarios y además permite el cálculo de indicadores hospitalarios ajustados y de otros indicadores de calidad y eficiencia de los servicios brindados en las instituciones hospitalarias.

Entre sus desventajas consta una arquitectura orientada a servicios que lo convierte en un sistema creado en una isla de información con utilidad y beneficios muy limitados, lo que imposibilita que el sistema se integre por si solo a la Red Telemática de Infomed.

Tiene una escasa o ninguna documentación el proceso de análisis, diseño e implementación del sistema, lo que hace extremadamente difícil el mantenimiento.

No posee buena escalabilidad, ya que al aumentar el tamaño de la base de de datos, la aplicación no puede adaptarse a los volúmenes de datos generados por las consultas y el trabajo se dificulta, perdiendo de esta forma la rapidez en las prestaciones de servicios.

1.3 Aplicaciones.

La construcción de *software* ha evolucionado a pasos agigantados, tanto conceptual como tecnológicamente. En esta última década dicha evolución se produjo en gran medida gracias al paradigma de orientación a objetos (OO), al acogimiento de la Internet y de las redes de datos como medio global de comunicación, y a la unificación de estándares, tanto de metodologías como de plataformas de desarrollo de *software*.

Cuando se desea desarrollar una aplicación se debe decidir entre otras cosas, qué tipo de aplicación se realizará: Web o de escritorio y cuál tecnología será empleada.

Es fundamental la existencia de una arquitectura bien definida a la hora de encarar el desarrollo de una aplicación.

Cada desarrollo tiene una serie de características que hacen que una tecnología sea más adecuada o menos adecuada que otra. Ni las interfaces Web son buenas para todo, ni las interfaces de escritorio solucionan todos los problemas.

1.3.1 Aplicaciones de escritorio.

Estas aplicaciones se basan en la eficacia del equipo de escritorio al procesar y presentar el contenido de alto rendimiento. En este caso hay que tener la aplicación instalada en cada uno de los clientes.

Entre sus ventajas tenemos la rapidez de acceso a los datos, el uso de interfaces enriquecidas, con controles inteligentes que dan mayor fluidez al trabajo del usuario.

El problema de las aplicaciones de escritorio generalmente radica en:

- ❖ Duplicidad de datos por la falta de unificación de los mismos.
- ❖ Diseminación de la información y lógica en muchas partes (cada computador que la use).
- ❖ Falta de portabilidad de la aplicación a diferentes sistemas operativos.

- ❖ Traumas a la hora de realizar actualizaciones o correcciones al programa ya que las instalaciones están diseminadas.
- ❖ Dificultad para configurar cada una de las instalaciones.(deployments) dependiendo de las necesidades de cada usuario.

Las aplicaciones de escritorio tienen un campo enorme (sistemas CAD, CAM, suite de oficina, aplicaciones gráficas, juegos, utilidades o el mismo sistema operativo), no todo esta en Web, hay cosas que se necesita que se ejecuten estrictamente en su máquina para aprovechar el poder que tiene al alcance.

1.3.2 Aplicaciones Web.

Existe ahora una forma de distribuir *software* que evita que los usuarios se conviertan en administradores de sistemas; las aplicaciones basadas en Web. Son programas que corren en servidores Web y utilizan páginas Web como interfaz de usuario.

Una ventaja importante de cualquier aplicación Web es que no tiene ningún costo de distribución. Los usuarios ya tienen instalada la única parte de la aplicación que necesitan en el cliente, el explorador.

Son, por definición, independientes de la plataforma; es decir, son aplicaciones "de envergadura".

Los usuarios pueden interactuar con la aplicación independientemente del tipo de explorador que tengan e, incluso, del tipo de equipo que utilicen.

Todas estas obvias ventajas dejan claro el potencial de las aplicaciones Web, pero dentro de sus desventajas están:

- ❖ Acceso limitado, la necesidad de conexión permanente y rápida a Internet hacen que el acceso a estas aplicaciones no esté al alcance de todos.
- ❖ La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera excesivo hasta que se obtiene la reacción del sistema.

- ❖ Elementos de interacción muy limitados. En comparación con el *software* de escritorio, las posibilidades de interacción con el usuario que ofrecen las aplicaciones Web (mediante formularios principalmente) son muy escasas.
- ❖ Diferencias de presentación entre plataformas y navegadores. La incompatibilidad de estándares ampliamente soportados dificulta el desarrollo de las aplicaciones.

1.4 Plataformas y lenguajes.

Una plataforma es una combinación de *hardware* y *software* usada para ejecutar aplicaciones; en su forma más simple consiste únicamente de un sistema operativo, una arquitectura, o una combinación de ambos.

Las plataformas de *software* pueden ser un sistema operativo, un entorno de programación, o (más comúnmente) una combinación de ambos.

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de *software* con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de aplicaciones.

Mono es la plataforma de desarrollo de *software* libre basada en .NET que permite a los desarrolladores de *software* construir aplicaciones GNU/Linux y multiplataforma con una productividad sin precedentes.

Mono incluye las herramientas de desarrollo y la infraestructura necesarias para crear aplicaciones del cliente y el servidor. Posee importantes componentes útiles de desarrollo de *software*, entre los cuales está un compilador para el lenguaje C #; una máquina virtual de lenguaje común de infraestructura (CLI) que contiene un cargador de clases, un compilador en tiempo de ejecución (JIT), y unas rutinas de recolección de memoria.

Tiene un entorno de desarrollo integrado (IDE), MonoDevelop, gratuito, diseñado primordialmente para C#.

El lenguaje C# (C Sharp) es parte de la plataforma .NET. Es un lenguaje orientado a objetos simple, seguro, moderno, de alto rendimiento y con especial énfasis en Internet y sus estándares (como XML). Es también la principal herramienta para programar en la plataforma .NET.

1.5 Software Libre.

“Software Libre” se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Su libertad la precisa:

- ❖ Usar el programa con cualquier propósito.
- ❖ Estudiar cómo funciona el programa, y adaptarlo a tus necesidades.
- ❖ Distribuir copias.
- ❖ Mejorar el programa y hacer públicas las mejoras a los demás.

En la actualidad existe una creciente tendencia hacia la utilización de *software* libre. Cada vez se promociona más la migración desde los sistemas con licencia comercial que están en poder de unos pocos monopolios de la rama de la informática que se reservan el derecho de vender sus productos a quienes consideren pertinente, hacia aquellos que se denominan “libres”.

En la Universidad de las Ciencias Informáticas (UCI), y en nuestro país en general, se ha comenzado a seguir una política de migración hacia el *software* libre y partiendo de esta premisa, el presente trabajo propone hacer uso de herramientas y tecnologías libres para la elaboración de la aplicación propuesta.

1.6 Metodologías de Desarrollo de Software.

“(…) la dirección con éxito del negocio o la ejecución de la misión pública depende del *software* que lo soporta. Estos sistemas *software* se hacen más complejos, su tiempo de salida al mercado necesita reducirse, y su desarrollo, por tanto, se hace más difícil. Por razones como éstas, la industria del *software* necesita un proceso para guiar a los desarrolladores (…)” (JACOBSON 2000)

Es labor del proceso de desarrollo hacer que las medidas para aumentar la calidad de los productos sean reproducibles en cada desarrollo.

El resultado final de un proyecto de *software* es un producto donde intervienen las personas a través de un proceso de desarrollo de *software* que guía los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla, que explica los pasos necesarios para terminar el proyecto.

En un proceso de desarrollo de *software* la metodología define QUIEN debe hacer QUE, CUANDO Y COMO debe hacerlo.

No existe una metodología de *software* universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc) exigen que el proceso sea configurable.

Entre las metodologías mas conocidas tenemos:

Dynamic Systems Development Method define el marco para desarrollar un proceso de producción de *software*. Nace en 1994 con el objetivo de crear una metodología unificada. Sus principales características son: es un proceso iterativo e incremental y el equipo de desarrollo y el usuario trabajan juntos. Propone cinco fases: estudio de la 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.(STAPLETON 1997)

Adaptive Software Development su impulsor es Jim Highsmith. Sus principales características 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.(HIGHSMITH J 2000)

Programación Extrema o **eXtreme Programing (XP)** es una la ingeniería de *software* desarrollada por Kent Beck. “ (...) se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.”(G. 9/12/2002). Siendo técnica adecuada para proyectos con requisitos imprecisos y muy cambiantes. Es un proceso ágil de desarrollo de *software* formado por la simplicidad, la comunicación, la retroalimentación y la refactorización de código. Tiene como objetivos la satisfacción del cliente y potenciar el trabajo en grupo, todos están involucrados en el desarrollo del *software* y como actividades fundamentales codificar, hacer pruebas, escuchar y diseñar. Consta de 7 roles en todas sus fases de desarrollo los cuales son: programador, cliente, encargado de Pruebas (*Tester*), entrenador (*Couch*), consultor, gestor (*Big Boss*) y el encargado del Seguimiento.

Rational Unified Process (RUP) es la herramienta líder en el mundo de modelación visual, análisis de requerimientos y diseño de sistemas orientados a objetos. Es una herramienta *CASE* (acrónimo de *Computer Aided Software Engineering*) que se basa en una combinación de herramientas de *software* y metodologías de desarrollo siendo herramientas individuales que ayudan al desarrollados de *software* o administrador del proyecto durante una o más fases del desarrollo del *software* (o mantenimiento).

1.3.5.1 Características de RUP.

Se caracteriza por:

- ❖ Ser iterativo e incremental: La alta complejidad de los sistemas actuales hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos o versiones del producto donde a cada uno de estos se le denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo.
- ❖ Estar centrado en la arquitectura: La arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.
- ❖ Guiado por los casos de uso: RUP utiliza los casos de uso tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígame diseño, implementación y prueba.

RUP es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de *software*.

1.3.5.2 La vida de un producto de *software* según RUP.

RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo. Cada ciclo se divide en cuatro fases:

- ❖ Inicio. Define el alcance del proyecto.
- ❖ Elaboración. Se hace el plan del proyecto, especificación de características, arquitectura base.
- ❖ Construcción. Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- ❖ Transición: Transición del producto a la comunidad del usuario.

El proceso puede describirse en dos dimensiones, o a lo largo de dos ejes (Ver **figura 2**).

- ❖ El eje horizontal representa tiempo y muestra el aspecto dinámico del proceso, expresado en términos de ciclos, fases, iteraciones, y metas.
- ❖ El eje vertical representa el aspecto estático del proceso; como está descrito en términos de actividades, artefactos, trabajadores y flujos de trabajo.

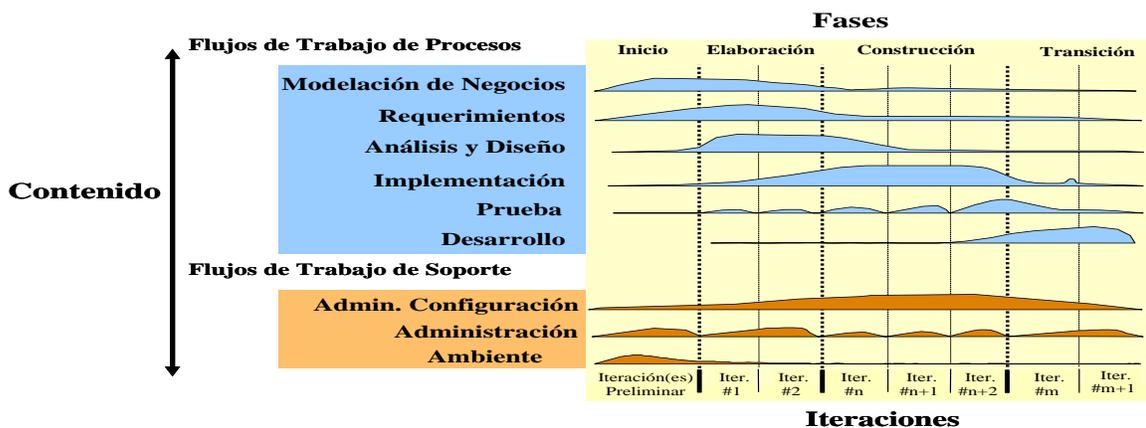


Figura 2 Estructura del RUP.

Otra característica a tener en cuenta es que RUP incluye artefactos que no son más que los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.

Todos estos artefactos se generan según el flujo de trabajo que se este llevando a cabo también se debe saber que estos artefactos se obtienen gracias de una serie de actividades realizadas por los trabajadores de ese flujo de trabajo.

Para el flujo de trabajo de modelación del negocio se generan los siguientes artefactos de modelado: actores del negocio, trabajadores del negocio, casos de uso del negocio, modelo del negocio, realización de casos de uso del negocio(descripción conceptual), diagrama de actividad de cada casos de uso del negocio y modelo de objeto del negocio.

En el flujo de trabajo de requerimientos se generan los siguientes: actores del sistema, casos de uso del sistema, modelo de casos de uso del sistema, requerimientos del *software*, requerimientos del *hardware*, especificación de los requerimientos del *software*, etc.

Para el flujo de trabajo de análisis y diseño se obtiene los artefactos de modelado que le siguen: diagrama de clases de análisis, diagrama de clases de diseño y diagramas de interacción.

1.3.5.3 El Lenguaje Unificado del Modelado (UML)

RUP y UML están estrechamente relacionados entre sí, pues mientras el primero establece las actividades y los criterios para conducir un sistema desde su máximo nivel de abstracción (la idea en la cabeza del cliente), hasta su nivel más concreto (un programa ejecutándose en las instalaciones del cliente), el segundo ofrece la notación gráfica necesaria para representar los sucesivos modelos que se obtienen en el proceso de refinamiento.

UML surge como respuesta a algunos problemas reseñados para contar con un lenguaje estándar para escribir planos de *software*. Es el resultado de una propuesta de estandarización promovida por el consorcio OMG (Object Management) Group) del cual forman parte las empresas más importantes que se dedican al desarrollo de *software*.

UML no es una metodología ni un lenguaje de programación es un lenguaje para modelos.

Los modelos en UML capturan una vista de un sistema del mundo real y describen completamente aquellos aspectos del sistema que son relevantes al propósito del modelo, y a un apropiado nivel de detalle.

Un modelo es expresado en un lenguaje de modelado compuesto por vistas, elementos de modelos y un conjunto de mecanismos generales o reglas que indican como utilizar los elementos.

Las vistas muestran diferentes aspectos del sistema modelado. Una vista no es una gráfica, pero sí una abstracción que consiste en un número de diagramas y todos esos diagramas juntos muestran una "fotografía" completa del sistema

Los diagramas son la representación gráfica de un conjunto de elementos describiendo el contenido de una vista y visualizan un sistema desde diferentes perspectivas.

1.4 Justificación de la propuesta de solución.

A partir de un análisis exhaustivo en este capítulo, como propuesta de solución se ha decidido por parte de los arquitectos y los analistas del sistema, desarrollar una aplicación de escritorio, siendo un *software* que resolvería las urgencias que presenta el departamento de archivos actualmente.

Como plataformas de desarrollo se han definido las siguientes: en la aplicación la programación se realizará con C# como lenguaje, compilado en Mono.

El Gestor de Bases de Datos a emplear será PostgreSQL.

Se utilizará como metodología de desarrollo de software el RUP.

El uso de estas tecnologías están en correspondencia con las políticas de migración hacia *software* libre en la UCI y del Sistema Nacional de Salud, debido a las prestaciones y ventajas que brindan ambos en la

construcción de *software* cubano, que lo hace evolucionar sin tener que pagar licencia. Además esto facilita una mejora en todo el flujo de trabajo del usuario.

1.5 Conclusiones.

En este capítulo se hizo un estudio de las tecnologías a utilizar en el desarrollo de la propuesta de solución dentro de la cuales se encuentran RUP, el gestor de base de datos utilizado PostgreSQL 8.2, y la plataforma de desarrollo .Net, así como algunos conceptos dentro de los cuales se encuentran las aplicaciones escritorios, aplicaciones Web y tendencias que se deben tener en cuenta la hora de realizar una aplicación de este tipo. Se fundamentó la metodología de desarrollo de *software* (RUP); así como el uso de técnicas, la programación orientada a objeto, basada en una arquitectura en tres capas lógicas (presentación, negocio, acceso a datos).

Características del sistema

En el presente capítulo se modela el negocio propuesto, identificándose los actores, trabajadores y los casos de uso correspondientes. Se plantean los requerimientos funcionales y no funcionales de la aplicación a desarrollar y se describe la solución propuesta en términos de casos de uso de sistema.

2.1 Flujo actual de los procesos involucrados en el campo de acción.

Dentro de los procesos fundamentales que se llevan a cabo en el departamento de archivos está el control de la entrada/salida de las HC; proceso que se hace mediante un registro que recoge datos como la fecha, a quien entregada/recibida la HC así como el departamento que la solicita.

Otro proceso a realizar por el personal de archivo es la clasificación de las HC que se da a partir del uso y tiempo que lleven dentro del archivo, esta clasificación oscila en un estado que puede ser activo, pasivo o de baja:

- ❖ El caso de una HC activa es cuando en criterio de uso ha sido utilizada fuera del archivo y en cuanto a tiempo lleva un período de 0 a 3 años dentro del archivo.
- ❖ Cuando se refiere a una HC es pasiva la HC cumplió su tiempo como activa y no fue modificada o consultada fuera del archivo. Al ser pasiva se le da un tiempo de estadía dentro del archivo de 5 años.
- ❖ En caso de que sea clasificada como de baja debe cumplirse primeramente que la HC a tratar sea una HC pasiva y cumpla con un período de estadía sin haber sido modificada o consultada fuera del archivo.

La organización que se le pueda dar a la HC, es otro proceso involucrado en el campo de acción donde su ubicación está dada, por la clasificación que tenga la HC (activo, pasivo o de baja). Además se necesita la inicial del primer apellido del paciente, para así destinarla a una gaveta y estante específicos.

En caso de que existan dos HC (una HC provisional porque el paciente fue hospitalizado por urgencia y la otra su HC normal). El personal de archivo procede a la unión de estas, donde dentro de la HC normal se adjunta la HC provisional siendo esta acción otro proceso a tener en cuenta.

Por último otro proceso a efectuar es el reporte estadístico de cada parámetro calculable con que se trabaje en el departamento. Dicho reporte se precisa cada mes o en un tiempo determinado por el administrador del departamento.

2.2 Objeto de automatización.

Como objeto de automatización inicial se tiene la autenticación de los usuarios al sistema, dando la posibilidad a los administradores de registrar usuarios y darle los permisos a las diferentes partes de la aplicación según la responsabilidad del usuario.

Luego ya como función esencial se comienza por la localización de las HC donde su búsqueda es según los criterios especificados los cuales pueden ser por número de la HC, nombre del paciente, fecha en que fue ingresada al archivo, fecha en que fue extraída del archivo, *etc.*

Otros puntos a tener en cuenta: registrar su entrada y salida de los estantes, consultar información de Entrada/Salida del Archivo, listar las HC que por un período determinado están fuera del archivo, listar del registro de pacientes las HC pasivas.

Otro parámetro a cumplir es proporcionarle una clasificación a las HC de forma tal que facilite el trabajo de este proceso en la actualidad.

En caso de que el paciente tenga dos HC se efectúa la unión de estas consolidando en un solo documento todos los episodios hospitalarios del mismo.

Y por último se brinda un resumen estadístico del movimiento de las HC donde se le da la posibilidad al jefe del departamento de realizar, en un tiempo determinado por el mismo, un resumen donde recoge todo lo que ha pasado con el movimiento de las HC.

2.3 Modelo del negocio propuesto.

A partir del uso de la metodología RUP como artefactos a generar en el primer flujo de trabajo Modelado del Negocio tenemos:

- ❖ Actores del negocio: es el rol o función que asume una persona, sistema o entidad que interactúa con el negocio, donde este actor se beneficia de los resultados de los procesos existentes.
- ❖ Trabajadores del negocio: son aquellas personas, grupo de personas, sistema automatizado o máquina, que trabajan en conjunto como un equipo realizando las actividades involucradas en uno o más procesos del negocio.
- ❖ Diagrama de actividad: Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro del negocio. Donde participan los actores/trabajadores del negocio, las actividades a realizar por ellos y las relaciones entre estas.
- ❖ Modelo de negocio: se realiza con el propósito de comprender las características y actividades que se llevan a cabo en el contexto a automatizar. En él, se modelan los actores y trabajadores del negocio y su interacción con los procesos y objetos que lo componen.
- ❖ Modelo de objetos del negocio: Es un modelo de objetos que describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo del proceso de negocio.

La **figura 3** muestra la representación de los actores y trabajadores involucrados en el negocio.

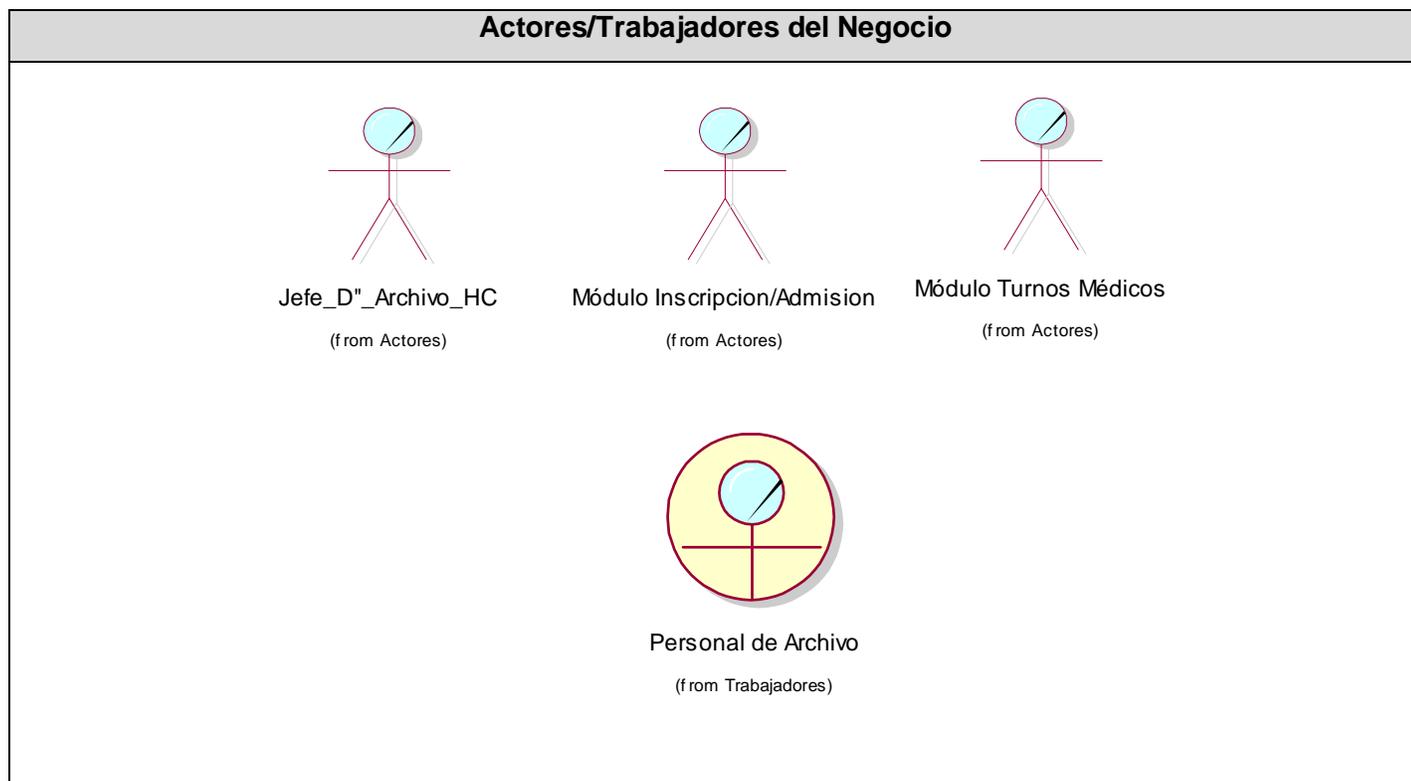


Figura 3 Actores / Trabajadores del Negocio

A continuación se describen los procesos del negocio propuesto, en el **Anexo I Modelado del Negocio Propuesto** pueden consultarse el Diagrama de Casos de Uso del Negocio, los Diagramas de Actividades y Modelos de Objetos del Negocio correspondientes a cada caso de uso del negocio.

CUN Analizar HC Este caso de uso se inicia cuando el Módulo de Inscripción/Admisión, quienes a partir de las peticiones del paciente, hacen la petición al departamento de Archivo la HC o la envía si el objetivo del paciente es la devolución de la misma.

CUN Solicitar HC: Este caso de uso se inicia cuando el Módulo de Turnos Médicos les hace llegar al personal del Archivo los datos del paciente para que el personal de archivos busque la HC correspondiente, luego de haber utilizado el departamento de Turnos Médicos esta HC procede a la

entrega de la HC donde el personal de archivo registra la entrada de la HC en el departamento y la organiza en los estantes.

CUN Gestionar Mov. Estadístico HC Este caso de uso se inicia cuando el jefe de departamento accede al archivo cada fin de mes, para controlar todo el movimiento estadístico de las HC.

2.4 Especificación de los requerimientos de *software*.

En éste epígrafe se relaciona la lista de requerimientos funcionales y no funcionales del sistema a desarrollar.

2.4.1 Requerimientos funcionales.

Un requerimiento funcional son capacidades o condiciones que el sistema debe cumplir definiendo el comportamiento interno del *software* así como los comportamientos del sistema. Luego de conocer los procesos de negocio y las actividades a automatizar identificadas se pueden definir los siguientes requisitos funcionales:

R1 -Autenticar usuario: permitir a los usuarios acceder a la información que le corresponde.

R1.1- Comparar Usuario y contraseña con los usuarios del sistema.

R1.2- Asignar privilegios.

R2 Gestionar Usuario: permitir realizar actividades sobre determinados usuarios del sistema.

R2.1 Insertar Usuarios

R2.2 Modificar Rol

R2.3 Eliminar Usuario

R3 Gestionar HC: permite realizar actividades al personal de archivo.

R3.1 Clasificar HC.

R3.2 Asignar ubicación a la HC.

R3.3 Unir HC.

R3.4 Registrar HC.

R3.5 Insertar nueva HC.

R4 Buscar HC: permite buscar las HC.

R5 Crear Reporte: permite crear reportes al jefe del departamento de archivo.

2.4.2 Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto.

Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes, representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros.

Apariencia o interfaz externa:

- ❖ Diseño sencillo, permitiendo la utilización del sistema sin mucho entrenamiento.
- ❖ Diseñado para la resolución desde 800x600 hasta 1024x768 píxel.

Soporte:

- ❖ Garantía de instalación y prueba del sistema, además de un breve entrenamiento a los futuros usuario.
- ❖ Crear un puesto de mando para en determinados periodos de tiempo darle mantenimiento al *software*.
- ❖ Crear nuevas versiones con el objetivo de solucionar las posibles desventajas que pueda presentar la aplicación.

Portabilidad:

- ❖ Independencia de la plataforma.

Seguridad:

- ❖ Existencia de distintos permisos que establezcan las acciones que pueden realizar los usuarios.
- ❖ Verificación sobre acciones irreversibles (por ejemplo las eliminaciones).

Confiabilidad:

- ❖ Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.
- ❖ Garantía de que se realice las operaciones necesarias, y que la información que se maneje en la aplicación sea real.

2.5 Modelo del sistema

En este epígrafe se muestra el modelo del sistema. Para ello se identifican los actores del sistema y se especifican los casos de uso del sistema.

- ❖ Los actores de un sistema son agentes externos, es decir, aquellas personas o sistemas que interactúan con él. En la **figura 4** se puede observar la representación de los actores del sistema propuesto.
- ❖ Los casos de uso son un conjunto de secuencia de acciones que un sistema ejecuta y produce un resultado observable para un actor.
- ❖ El modelo del sistema es utilizado para representar los de casos de uso y actores con sus relaciones entre sí.

Puede consultar el **Anexo II Modelado del Sistema** donde se muestra el Diagrama de casos de uso del sistema además de la descripción de los actores del sistema así como la de los casos de uso del sistema propuesto.

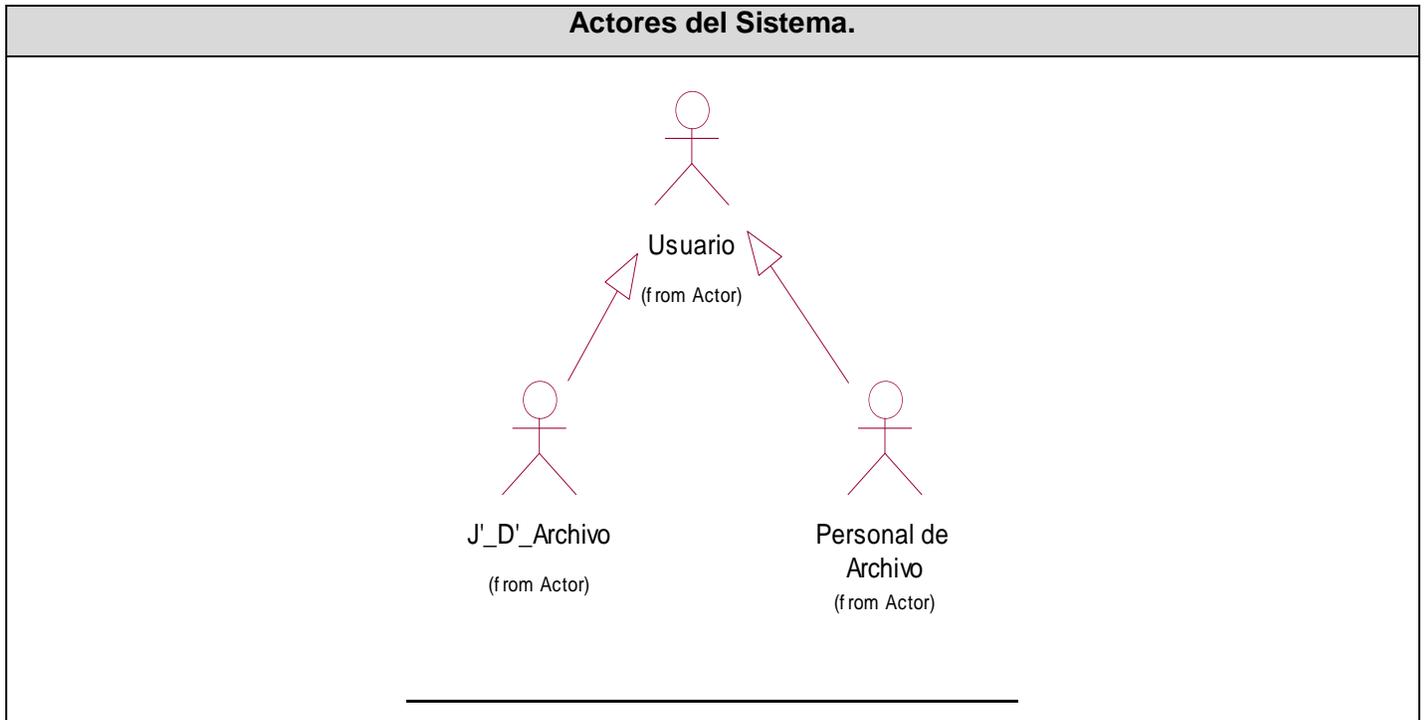


Figura 4 Actores del Sistema.

2.6 Conclusiones.

En este capítulo se realizó una modelación del negocio propuesto, identificándose los actores, trabajadores y los casos de uso correspondientes; se plantearon los requerimientos funcionales y no funcionales de la aplicación a desarrollar y se modeló la misma en términos de casos de uso de sistema.

Análisis y diseño del sistema

En el siguiente capítulo se analiza y diseña la propuesta de solución abordando conceptos básicos para una mejor explicación del tema y diagramas expuestos en este capítulo para finalizar con el prototipo de interfaz de usuario.

3.1 Modelo de Análisis.

El Modelo de Análisis está compuesto por propósitos fundamentales:

- ❖ Refinar los casos de uso con más detalles.
- ❖ Establecer la asignación inicial de la funcionalidad del sistema a un conjunto de objetos que proporcionan el comportamiento.
- ❖ No se toman en cuenta el lenguaje de programación a usar en la construcción, la plataforma en la que se ejecutará la aplicación, entre otras características que afectan al sistema.

3.1.1 Diagramas de Clases del Análisis.

Las clases de análisis se centran en los requerimientos funcionales y se clasifican en Interfaz, de Control o Entidad.

- ❖ Clase de Interfaz: Modelan la interacción entre el sistema y sus actores.
- ❖ Clase de Control: Representan coordinación, secuencia, transacciones, y control de otros objetos y a menudo encapsula a un caso de uso en concreto.
- ❖ Clase de Entidad: Estas clases modelan información que posee una larga vida y que a menudo es persistente y fenómenos, conceptos y sucesos que ocurren en el mundo real.

Los diagramas de clases del análisis muestran un resumen del sistema en términos de sus clases y las relaciones entre ellas. Son diagramas estáticos que muestran **qué** es lo que interactúa, pero no cómo interactúa o qué pasa cuando ocurre la interacción.

Para ver los diagramas de clases de análisis consulte el [Anexo III Análisis y diseño del sistema](#).

3.2 Modelo de diseño

El modelo de diseño describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar.

Sirve de abstracción de la implementación y es utilizada como entrada fundamental de las actividades de implementación.

3.2.1 Diagrama de clases de diseño

“Una clase de diseño es una abstracción de una clase o construcción en la implementación del sistema.
“(JACOBSON 2000)

Un diagrama de clases de diseño es un diagrama que muestra un conjunto de interfaces, colaboraciones y sus relaciones.

Los diagramas de clases de diseño se utilizan para modelar la vista de diseño estática de un sistema principalmente. Esto incluye modelar el vocabulario del sistema, las colaboraciones o esquemas.

Los diagramas de clases son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables, aplicando ingeniería directa e inversa. Para ver los diagramas de clases de diseño consulte el [Anexo III Análisis y diseño del sistema](#).

3.2.2 Diagramas de interacción

Un diagrama de interacción muestra una interacción, que consta de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

El diagrama de interacción puede ser de dos tipos: un diagrama de secuencia que destaca la ordenación temporal de los mensajes o un diagrama de colaboración que destaca la organización estructural de los objetos que envían y reciben mensajes.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Los diagramas de interacción pueden contener notas y restricciones. Consulte el **Anexo III Análisis y diseño del Sistema** para ver los diagramas referentes a los casos de uso del sistema.

3.3 Patrones

Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos.

Los patrones capturan la experiencia existente y probada para promover buenas prácticas.

3.3.1 Patrón de Diseño

Los patrones son empleados en los diagramas de clases de diseño para refinar el diseño y hacerlo de forma eficiente. Un patrón es un modelo que podemos seguir para realizar algo. Surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos capturando la experiencia existente y probada para promover buenas prácticas.

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia y que se ha demostrado que funcionan.

Entre los patrones de diseño tenemos los de patrones de diseño GOF (Gang of Four) (que en español es la pandilla de los cuatro) entre ellos se encuentra el uso de los patrones *Abstract Factory*.

El patrón de diseño *Abstract Factory* aborda el problema de la creación de familias de objetos que comparten toda una serie de características comunes en los objetos que componen dichas familias.

3.3.2 Patrón de Arquitectura

“ La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad ” (KRUCHTEN Noviembre 1995).

3.3.2.1 Modelo Cliente servidor

La arquitectura Cliente/Servidor requiere una determinada especialización de cada uno de los diferentes componentes que la integran aprovechando diferentes estándares, convenciones, reglas y procesos. En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes.

3.3.2.2 Arquitectura de 3 capas.

El objetivo primordial aplicado en esta arquitectura es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

Esta arquitectura ha sido definida por los arquitectos Mainoldis Fuentes y Alain Ramos del Sistema de Gestión Hospitalaria.

3.4 Descripción de clases

3.4.1 Clases Interfaz

Tabla 1 Descripción de la CI del CUS <Autenticar Usuario>

Nombre: CI_AutenticarUsuario	
Tipo de clase: interfaz	
Atributo	Tipo

txt_Usuario	textbox
txt_Contraseña	textbox
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.

Tabla 2 Descripción de la CI Eliminar Usuario del CUS <Autenticar Usuario>

Nombre: CI_EliminarUsuario	
Tipo de clase: interfaz	
Atributo	Tipo
txt_CI	textbox
txt_Usuario	textbox
txt_Rol	textbox
btnCancelar	button
btnEliminarUsuario	button
Para cada responsabilidad:	
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnEliminar_Click
Descripción:	Función que define la programación del evento de un botón.

Tabla 3 Descripción de la CI Insertar Usuario del CUS <Autenticar Usuario>

Nombre: CI_InsertarUsuario	
Tipo de clase: interfaz	
Atributo	Tipo
txt_Nombre_Usuario	textbox
txt_Contraseña	textbox
txt_Confirmar_Contraseña	textbox
txt_Num_Identidad	textbox
txt_Rol	textbox
btnCancelar	button
btnAdicionar	button
Para cada responsabilidad:	
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnAdicionar_Click

Descripción:	Función que define la programación del evento de un botón.
--------------	--

Tabla 4 Descripción de la CI Modificar Rol del CUS <Autenticar Usuario>

Nombre: CI_ModificarRol	
Tipo de clase: interfaz	
Atributo	Tipo
txt_Nombre_Usuario	textbox
txt_CI	textbox
txt_Rol	textbox
txt_Permisos	textbox
btnCancelar	button
btnModificar	button
Para cada responsabilidad:	
Nombre:	protected void btnCancelar_Click
Descripción:	Función que define la programación del evento de un botón.
Nombre:	protected void btnAdicionar_Click
Descripción:	Función que define la programación del evento de un botón.

Tabla 5 Descripción de la CI del CUS <Clasificar HC>

Nombre: CI_Clasificar HC	
Tipo de clase: interfaz	
Atributo	Tipo
txt_num_HC	textbox
txt_nombre	textbox
txt_CI	textbox
cbx_estado	combobox
txt_ubicacion	textbox
btnBuscar	button
btnAceptar	button
btnCancelar	button
Para cada responsabilidad:	
Nombre:	protected void btnBuscar_Click
Descripción:	Función que define la programación del evento de un botón.
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.

Tabla 6 Descripción de la CI del CUS <Insertar HC>

Nombre: CI_Insertar HC	
Tipo de clase: interfaz	
Atributo	Tipo
txt_num_HC	textbox
txt_nombre	textbox
txt_1er_apellido	textbox
txt_2do_apellido	textbox
txt_CI	textbox
cbx_estado	combobox
txt_ubicacion	textbox
btnBuscar	button
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.

Tabla 7 Descripción de la CI del CUS <Unir HC>

Nombre: CI_Unir HC	
Tipo de clase: interfaz	
Atributo	Tipo
txt_num_HC	textbox
txt_direccion	textbox
txt_edad	textbox
txt_ubicacion	textbox
btnAceptar	button
btnCancelar	button
Para cada responsabilidad:	
Nombre:	protected void btnBuscar_Click
Descripción:	Función que define la programación del evento de un botón.
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.

Tabla 8 Descripción de la CI del CUS <Registrar E/S HC>

Nombre: CI_Registrar E/S HC	
Tipo de clase: interfaz	
Atributo	Tipo
date_fecha_hora_entrega	date
txt_prestado_a	textbox
date_fecha_hora_retorno	date
txt_entregada_por	textbox
txt_observaciones	textbox
cbx_motivo_salida	combobox
cbx_departamento	combobox
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.

Tabla 9 Descripción de la CI del CUS <Buscar HC>

Nombre: CI_Buscar HC	
Tipo de clase: interfaz	
Atributo	Tipo
txt_nombre	textbox
txt_1er_apellido	textbox
txt_2do_apellido	textbox
txt_CI	textbox
txt_num_HC	textbox
date_fecha_nacimiento	date
cbx_sexo	combobox
cbx_estado	combobox
txt_ubicacion	textbox
date_fecha_entrada	date
date_fecha_salida	date
cbx_motivo_salida	combobox
txt_en_archivo	textbox
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.

Tabla 10 Descripción de la CI del CUS <Crear Reporte>

Nombre: CI_Crear Reporte	
Tipo de clase: interfaz	
Atributo	Tipo
dtp_desde	datetimepicker
dtp_hasta	datetimepicker
chk_motivo	checkbox
chk_fechaE/S	checkbox
chk_estadofisicoHC	checkbox
chk_departamento	checkbox
chk_estadoHC	checkbox
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.

3.4.2 Clases Controladoras

Tabla 11 Descripción de la CC <Autenticar Usuario>

Nombre: CC_VerificarUsuario	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	VerificarUsuario
Descripción:	Función que permite verificar los datos del usuario.
Nombre:	SeleccionarUsuario
Descripción:	Función que permite seleccionar los datos del usuario.

Tabla 12 Descripción de la CC <Gestionar Usuario>

Nombre: CC_GestionarUsuario	
Tipo de clase: controladora	

Atributo	Tipo
Para cada responsabilidad:	
Nombre:	InsertarUsuario
Descripción:	Función que permite insertar un nuevo usuario.
Nombre:	ModificarRol
Descripción:	Función que permite modificar el rol que tiene el usuario
Nombre:	EliminarUsuario
Descripción:	Función que permite eliminar los datos de un usuario.

Tabla 13 Descripción de la CC del repositorio de la CC <Autenticar Usuario>

Nombre: CC_RepositorioAutenticar	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	VerificarUsuario
Descripción:	Función que permite verificar los datos del usuario.
Nombre:	SeleccionarUsuario
Descripción:	Función que permite definir la selección de los datos del usuario.

Tabla 14 Descripción de la CC <Gestionar HC>

Nombre: CC_Gestionar HC	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	BuscarPersona
Descripción:	Función que define la búsqueda de la HC del paciente por los datos personales de este.
Nombre:	ModificarEstadoHC
Descripción:	Función que define la unión de una HC provisional con la HC del paciente.
Nombre:	ModificarUbicacion
Descripción:	Función que define el cambio de la ubicación de las HC.
Nombre:	VerificarUbicacion
Descripción:	Función que define la verificación de la ubicación de las HC.
Nombre:	InsertarHC
Descripción:	Función que define que los datos de las HC sean insertados
Nombre:	InsertarUbicación
Descripción:	Función que define que los datos de la ubicación de la HC sean insertados.

Capítulo 3. Análisis y diseño del sistema

Nombre:	SeleccionarUbicación
Descripción:	Función que define que los datos para seleccionar la ubicación de la HC
Nombre:	BuscarHC
Descripción:	Función que define la búsqueda de las HC.
Nombre:	BuscarEstado
Descripción:	Función que define la búsqueda del estado de las HC
Nombre:	UnionHC
Descripción:	Función que define el cambio del estado de la HC.
Nombre:	InsertarNumHCSecundaria
Descripción:	Función que define que los datos de la HC provisional sean insertados.
Nombre:	RegistroEntrada
Descripción:	Función que define el registro de los datos de entrada de la HC al archivo.
Nombre:	RegistroSalida
Descripción:	Función que define el registro de los datos de salida de la HC al archivo.
Nombre:	BuscarDepartamento
Descripción:	Función que define la búsqueda del departamento que solicitó las HC.
Nombre:	BuscarMotSalida
Descripción:	Función que define la búsqueda del motivo de salida con que salieron las HC del archivo.

Tabla 15 Descripción de la CC del repositorio de la CC <Gestionar HC>

Nombre: CC_Repositorio_Gestionar_HC	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	InsertarHC
Descripción:	Función que permite definir los datos de las HC que van hacer insertados
Nombre:	InsertarNumHC
Descripción:	Función que permite definir que el número de la HC sea insertado.
Nombre:	InsertarUbicacion
Descripción:	Función que permite definir los datos de la ubicación de las HC ha insertar.
Nombre:	InsertarSalida
Descripción:	Función que permite definir los datos de salida de la HC ha insertar.
Nombre:	InsertarEntrada
Descripción:	Función que permite definir los datos de entrada de la HC ha insertar.
Nombre:	ActualizarUnion
Descripción:	Función que permite definir la actualización de los datos de la unión realizada con la HC provisional y la HC del paciente.
Nombre:	ActualizarEstadoHC

Capítulo 3. Análisis y diseño del sistema

Descripción:	Función que permite definir la actualización del estado de la HC.
Nombre:	SeleccionarPersona
Descripción:	Función que permite definir la selección de pacientes.
Nombre:	SeleccionarEstado
Descripción:	Función que permite definir la selección del estado de la HC.
Nombre:	SeleccionarUbicacion
Descripción:	Función que permite definir la selección de la ubicación de la HC.
Nombre:	SeleccionarDepartamento
Descripción:	Función que permite definir la selección del departamento que solicitó la HC.
Nombre:	SeleccionarMotSalida
Descripción:	Función que permite definir la selección del motivo de salida con que salió la HC del archivo.

Tabla 16 Descripción de la CC <Buscar HC>

Nombre: CC_Buscar HC	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	BuscarHCNum
Descripción:	Función que define la búsqueda por el número de la HC.
Nombre:	BuscarPersona
Descripción:	Función que define la búsqueda por los datos del paciente.
Nombre:	BuscarEstadoHC
Descripción:	Función que define la búsqueda por el estado de la HC.
Nombre:	BuscarUbicacion
Descripción:	Función que define la búsqueda por la ubicación de la HC.
Nombre:	BuscarRegistroE/S
Descripción:	Función que define la búsqueda por el registro de entrada/salida en que este la HC.
Nombre:	MostrarTodosDatosHC
Descripción:	Función que define la presentación de todos los datos de la HC.
Nombre:	BuscarMotivo_Salida
Descripción:	Función que define la búsqueda por el motivo de salida con que hayan salido las HC.
Nombre:	BuscarHCDepartamentos
Descripción:	Función que define la búsqueda por los departamentos que hayan solicitado/entregado las HC.
Nombre:	BuscarHC_EstadoFisico
Descripción:	Función que define la búsqueda por el estado físico que tenga la HC.

Tabla 17 Descripción de la CC del repositorio de la CC <Buscar HC>

Nombre: CC_Repositorio_Buscar HC	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	SeleccionarPersona
Descripción:	Función que define la selección del paciente por sus datos.
Nombre:	SeleccionarEstado
Descripción:	Función que define la selección de la HC por el estado.
Nombre:	SeleccionarUbicacion
Descripción:	Función que define la selección de la ubicación de la HC.
Nombre:	SeleccionarRegistroE/S
Descripción:	Función que define la selección del registro de entrada/salida en que este la HC.
Nombre:	SeleccionarDepartamento
Descripción:	Función que define la selección del departamento que solicito la HC.
Nombre:	SeleccionarMotivo_Salida
Descripción:	Función que define la selección del motivo de salida con que hayan salido las HC.

Tabla 18 Descripción de la CC <Crear Reporte>

Nombre: CC_Crear Reporte	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	CrearReporte
Descripción:	Función que define la presentación del reporte creado

Tabla 19 Descripción de la CC del repositorio de la CC <Crear Reporte>

Nombre: CC_Repositorio_Crear Reporte	
Tipo de clase: controladora	
Atributo	Tipo
Para cada responsabilidad:	
Nombre:	SeleccionarEstado
Descripción:	Función que permite definir la selección del estado de la HC.
Nombre:	SeleccionarUbicacion
Descripción:	Función que permite definir la selección de la ubicación de la HC.
Nombre:	SeleccionarDepartamento

Descripción:	Función que permite definir la selección del departamento que solicitó la HC.
Nombre:	SeleccionarMotSalida
Descripción:	Función que permite definir la selección del motivo de salida con que salió la HC del archivo.
Nombre:	SeleccionarHC
Descripción:	Función que permite definir la selección de la HC.
Nombre:	SeleccionarPersona
Descripción:	Función que permite definir la selección del paciente.
Nombre:	SeleccionarRegE/S
Descripción:	Función que permite definir la selección del registro de entrada/salida de la HC del archivo.

3.4.2 Clases entidad

Tabla 20 Descripción de la CE <CE_Usuario>

Nombre: CE_Usuario	
Tipo de clase: entidad	
Atributo	Tipo
CI	boolean
nombre	String
contraseña	String
Para cada responsabilidad:	
Nombre:	Get CI
Descripción:	Función que devuelve el número de identidad del usuario.
Nombre:	Get nombre
Descripción:	Función que devuelve el nombre del usuario.
Nombre:	Get contraseña
Descripción:	Función que devuelve la contraseña del usuario.
Nombre:	Set CI
Descripción:	Función que permite entrar el número de identidad del usuario.
Nombre:	Set nombre
Descripción:	Función que permite entrar el nombre del usuario.
Nombre:	Set contraseña
Descripción:	Función que permite entrar la contraseña del usuario.

Tabla 21 Descripción de la CE <CE_Rol>

Nombre: CE_Rol	
Tipo de clase: entidad	
Atributo	Tipo
Id_Rol	boolean
Nombre_Rol	String

Para cada responsabilidad:	
Nombre:	Get Id_Rol
Descripción:	Función que devuelve el identificador del rol que pueda tener un usuario.
Nombre:	Get Nombre_Rol
Descripción:	Función que devuelve el nombre del rol que pueda tener un usuario.
Nombre:	Set Id_Rol
Descripción:	Función que permite entrar el id del rol del usuario.
Nombre:	Set Nombre_Rol
Descripción:	Función que permite entrar el nombre del rol del usuario.

Tabla 22 Descripción de la CE <CE_Permisos>

Nombre: CE_Permisos	
Tipo de clase: entidad	
Atributo	Tipo
Id_Permiso	Boolean
Nombre_Permiso	String
Para cada responsabilidad:	
Nombre:	Get Id_Permiso
Descripción:	Función que devuelve el identificador del permiso que pueda tener un usuario.
Nombre:	Get Nombre_Permiso
Descripción:	Función que devuelve el nombre del permiso que pueda tener un usuario.
Nombre:	Set Id_permiso
Descripción:	Función que permite entrar el id del permiso del usuario.
Nombre:	Set Nombre_Permiso
Descripción:	Función que permite entrar el nombre del permiso del usuario.

Tabla 23 Descripción de la CE <CE_Datos_Persona>

Nombre: CE_Datos_Persona	
Tipo de clase: entidad	
Atributo	Tipo
id_persona	Integer
nombre	String
primer_apellido	String
segundo_apellido	String
CI	Integer
fecha_nacimiento	Date
nombre_madre	String
nombre_padre	String
centro_trabajo	String
direccion_centro_trabajo	String

fecha_registro	Date
factor_rh	String
telefono	Integer
trabajador_salud	Boolean
id_color_piel	Integer
idsexo	Integer
id_municipio	Integer
Id_provincia	Integer
id_nacionalidad	Integer
Para cada responsabilidad:	
Nombre:	Get id_persona
Descripción:	Función que devuelve el identificador de una persona determinada.
Nombre:	Get nombre
Descripción:	Función que devuelve el nombre de una persona determinada.
Nombre:	Get 1er_Apellido
Descripción:	Función que devuelve el primer apellido de una persona determinada.
Nombre:	Get 2do_Apellido
Descripción:	Función que devuelve el segundo apellido de una persona determinada.
Nombre:	Get CI
Descripción:	Función que devuelve el número del carnet de identidad de una persona determinada.
Nombre:	Get fecha_nacimiento
Descripción:	Función que devuelve la fecha de nacimiento de una persona determinada.
Nombre:	Get nombre_madre
Descripción:	Función que devuelve el nombre de la madre de una persona determinada.
Nombre:	Get nombre_padre
Descripción:	Función que devuelve el nombre del padre de una persona determinada.
Nombre:	Get centro_trabajo
Descripción:	Función que devuelve el centro de trabajo al cual pertenece una persona determinada.
Nombre:	Get direccion_centro_trabajo
Descripción:	Función que devuelve la dirección del centro de trabajo de una persona determinada.
Nombre:	Get fecha_registro
Descripción:	Función que devuelve la fecha de registro de una persona determinada.
Nombre:	Get factor_rh
Descripción:	Función que devuelve el factor (rh) de una persona determinada.
Nombre:	Get telefono
Descripción:	Función que devuelve el teléfono de una persona determinada.
Nombre:	Get trabajador_salud
Descripción:	Función que devuelve si es o no trabajador de salud una persona determinada.
Nombre:	Get id_color_piel

Capítulo 3. Análisis y diseño del sistema

Descripción:	Función que devuelve el identificador del color de piel de una persona determinada.
Nombre:	Get id_sexo
Descripción:	Función que devuelve el identificador del sexo de una persona determinada.
Nombre:	Get id_municipio
Descripción:	Función que devuelve el identificador del municipio al que pertenece una persona determinada.
Nombre:	Get id_provincia
Descripción:	Función que devuelve el identificador de la provincia a la que pertenece una persona determinada.
Nombre:	Get id_nacionalidad
Descripción:	Función que devuelve el identificador de la nacionalidad que tiene una persona determinada.
Nombre:	Set id_persona
Descripción:	Función que permite entrar el identificador de una persona determinada.
Nombre:	Set nombre
Descripción:	Función que permite entrar el nombre de una persona determinada.
Nombre:	Set 1er_Apellido
Descripción:	Función que permite entrar el primer apellido de una persona determinada.
Nombre:	Set 2do_Apellido
Descripción:	Función que permite entrar el segundo apellido de una persona determinada.
Nombre:	Set CI
Descripción:	Función que permite entrar el número del carnet de identidad de una persona determinada.
Nombre:	Set fecha_nacimiento
Descripción:	Función que permite entrar la fecha de nacimiento de una persona determinada.
Nombre:	Set nombre_madre
Descripción:	Permite entrar el nombre de la madre de una persona determinada.
Nombre:	Set nombre_padre
Descripción:	Función que permite entrar el nombre del padre de una persona determinada.
Nombre:	Set centro_trabajo
Descripción:	Función que permite entrar el nombre del centro de trabajo al cual pertenece una persona determinada.
Nombre:	Set direccion_centro_trabajo
Descripción:	Función que permite entrar la dirección del centro de trabajo de una persona determinada.
Nombre:	Set fecha_registro
Descripción:	Función que permite entrar la fecha de registro de una persona determinada.
Nombre:	Set factor_rh
Descripción:	Función que permite entrar el factor (rh) de una persona determinada.
Nombre:	Set telefono
Descripción:	Función que permite entrar el número de teléfono de una persona determinada.

Nombre:	Set trabajador_salud
Descripción:	Función que permite entrar si es o no trabajador de salud una persona determinada.
Nombre:	Set id_color_piel
Descripción:	Función que permite entrar el identificador del color de piel de una persona determinada.
Nombre:	Set id_sexo
Descripción:	Función que permite entrar el identificador del sexo de una persona determinada.
Nombre:	Set id_municipio
Descripción:	Función que permite entrar el identificador del municipio al que pertenece una persona determinada.
Nombre:	Set id_provincia
Descripción:	Función que permite entrar el identificador de la provincia a la que pertenece una persona determinada.
Nombre:	Set id_nacionalidad
Descripción:	Función que permite entrar el identificador de la nacionalidad que tiene una persona determinada.

Tabla 24 Descripción de la CE <CE_HC>

Nombre: CE_HC	
Tipo de clase: entidad	
Atributo	Tipo
id_persona	Integer
num_hc	Integer
fecha_inscripcion	Date
estado_fisico	String
en_archivo	Boolean
fecha_ultima_salida	Date
id_estado	Integer
Para cada responsabilidad:	
Nombre:	Get id_persona
Descripción:	Función que devuelve el identificador de una persona determinada.
Nombre:	Get num_hc
Descripción:	Función que devuelve el número de una historia clínica determinada.
Nombre:	Get fecha_inscripcion
Descripción:	Función que devuelve la fecha de inscripción de una persona determinada.
Nombre:	Get estado_fisico
Descripción:	Función que devuelve el estado físico de una historia clínica determinada.
Nombre:	Get en_archivo
Descripción:	Función que devuelve si se encuentra o no la historia clínica en el archivo.

Nombre:	Get fecha_ultima_salida
Descripción:	Función que devuelve la fecha de la última salida de una historia clínica determinada del archivo.
Nombre:	Get id_estado
Descripción:	Función que devuelve el identificador del estado de una historia clínica determinada.
Nombre:	Set id_persona
Descripción:	Función que permite entrar el identificador de una persona determinada.
Nombre:	Set num_hc
Descripción:	Función que permite entrar el número de una historia clínica determinada.
Nombre:	Set fecha_inscripcion
Descripción:	Función que permite entrar la fecha de inscripción de una persona determinada.
Nombre:	Set estado_fisico
Descripción:	Función que permite entrar el estado físico de una HC determinada.
Nombre:	Set en_archivo
Descripción:	Función que permite entrar si se encuentra o no la HC en el archivo.
Nombre:	Set fecha_ultima_salida
Descripción:	Función que permite entrar la fecha de la última salida de una HC determinada del archivo.
Nombre:	Set id_estado
Descripción:	Función que permite entrar el identificador del estado de una HC determinada.

Tabla 25 Descripción de la CE <CE_hc_estados>

Nombre: CE_EstadoHC	
Tipo de clase: entidad	
Atributo	Tipo
id_estado	Integer
estado	String
Para cada responsabilidad:	
Nombre:	Get id_estado
Descripción:	Función que devuelve el identificador del estado de una HC determinada.
Nombre:	Get estado
Descripción:	Función que devuelve el estado en que se encuentra una HC determinada.
Nombre:	Set id_estado
Descripción:	Función que permite entrar el identificador del estado de una HC determinada.
Nombre:	Set estado
Descripción:	Función que permite entrar el estado en que se encuentra una HC determinada.

Tabla 26 Descripción de la CE <CE_union_HC>

Nombre: CE_union_HC	
Tipo de clase: entidad	
Atributo	Tipo
id_union	Integer
hc_nueva	Integer
fecha_cambio	Date
Para cada responsabilidad:	
Nombre:	Get hc_nueva
Descripción:	Función que devuelve el número de la HC nueva que se quiere unir a la ya existente de un paciente determinado.
Nombre:	Get fecha_cambio
Descripción:	Función que devuelve la fecha en que fue realizada la unión de las HC de una persona determinada.
Nombre:	Get id_union
Descripción:	Función que devuelve el identificador de una HC que ha sido anexada a otra.
Nombre:	Set hc_nueva
Descripción:	Función que permite entrar el número de la HC nueva que se quiere unir a la ya existente de un paciente determinado.
Nombre:	Set fecha_cambio
Descripción:	Función que permite entrar la fecha en que fue realizada la unión de las HC de una persona determinada.
Nombre:	Set id_union
Descripción:	Función que permite entrar el identificador de una HC que ha sido anexada a otra.

Tabla 27 Descripción de la CE <CE_Ubicacion>

Nombre: CE_Ubicacion	
Tipo de clase: entidad	
Atributo	Tipo
piso	Integer
estante	String
no_gaveta	Integer
id_persona	Integer
num_hc	Integer
Para cada responsabilidad:	
Nombre:	Get piso
Descripción:	Función que devuelve el piso en el que se encuentra localizada una HC determinada.
Nombre:	Get estante
Descripción:	Función que devuelve el estante en el que se encuentra localizada una HC determinada.

Nombre:	Get no_gaveta
Descripción:	Función que devuelve el número de la gaveta en la que se encuentra localizada una HC determinada.
Nombre:	Get id_persona
Descripción:	Función que devuelve el identificador de una persona determinada.
Nombre:	Get num_hc
Descripción:	Función que devuelve el número de una HC determinada.
Nombre:	Set piso
Descripción:	Función que permite entrar el piso en el que se encuentra localizada una HC determinada.
Nombre:	Set estante
Descripción:	Función que permite entrar el estante en el que se encuentra localizada una HC determinada.
Nombre:	Set no_gaveta
Descripción:	Función que permite entrar el número de la gaveta en la que se encuentra localizada una HC determinada.
Nombre:	Set id_persona
Descripción:	Función que permite entrar el identificador de una persona determinada.
Nombre:	Set num_hc
Descripción:	Función que permite entrar el número de una historia clínica determinada.

Tabla 24 Descripción de la CE <CE_RegE/S >

Nombre: CE_RegE/S	
Tipo de clase: entidad	
Atributo	Tipo
id_registro	Integer
fecha_hora_entrega	Date
fecha_hora_retorno	Date
entregada_a	String
recibida_de	String
recibida_por	String
Para cada responsabilidad:	
Nombre:	Get id_registro
Descripción:	Función que devuelve el identificador del registro de entrada o salida de una HC determinada.
Nombre:	Get fecha_hora_entrega
Descripción:	Función que devuelve la fecha y la hora en que salió del archivo una HC determinada.
Nombre:	Get fecha_hora_retorno
Descripción:	Función que devuelve la fecha y la hora en que fue devuelta una HC determinada.
Nombre:	Get entregada_a
Descripción:	Función que devuelve el nombre al que fue entregada una HC determinada.

Nombre:	Get recibida_de
Descripción:	Función que devuelve el nombre del que realizó la entrega de una historia clínica determinada.
Nombre:	Get recibida_por
Descripción:	Función que devuelve el nombre del que recibió de vuelta una HC determinada.
Nombre:	Set id_registro
Descripción:	Función que permite entrar el identificador del registro de entrada o salida de una HC determinada.
Nombre:	Set fecha_hora_entrega
Descripción:	Función que permite entrar la fecha y la hora en que salió del archivo una HC determinada.
Nombre:	Set fecha_hora_retorno
Descripción:	Función que permite entrar la fecha y la hora en que fue devuelta una HC determinada.
Nombre:	Set entregada_a
Descripción:	Función que permite entrar el nombre al que fue entregada una HC determinada.
Nombre:	Set recibida_de
Descripción:	Función que permite entrar el nombre del que realizó la entrega de una HC determinada.
Nombre:	Set recibida_por
Descripción:	Función que permite entrar el nombre del que recibió de vuelta una HC determinada.

Tabla 29 Descripción de la CE <CE_Departamento>

Nombre: CE_Departamento	
Tipo de clase: entidad	
Atributo	Tipo
id_departamento	Integer
departamento	String
Para cada responsabilidad:	
Nombre:	Get id_departamento
Descripción:	Función que devuelve el identificador de un departamento determinado.
Nombre:	Get departamento
Descripción:	Función que devuelve el nombre de un departamento determinado.
Nombre:	Set id_departamento
Descripción:	Función que permite entrar el identificador de un departamento determinado.
Nombre:	Set departamento

Descripción:	Función que permite entrar el nombre de un departamento determinado.
--------------	--

Tabla 30 Descripción de la CE <CE_Motivo_Salida>

Nombre: CE_Motivo_Salida	
Tipo de clase: entidad	
Atributo	Tipo
id_motivo_salida	Integer
motivo_salida	String
Para cada responsabilidad:	
Nombre:	Get id_motivo_salida
Descripción:	Función que devuelve el identificador del motivo de salida por el cual fue movida del archivo una HC determinada.
Nombre:	Get motivo_salida
Descripción:	Función que devuelve el motivo de salida de una HC determinada del archivo.
Nombre:	Set id_motivo_salida
Descripción:	Función que permite entrar el identificador del motivo de salida por el cual fue movida del archivo una HC determinada.
Nombre:	Set motivo_salida
Descripción:	Función que permite entrar el motivo de salida de una HC determinada del archivo.

3.5 Prototipo de Interfaz de Usuario

Un prototipo es una versión inicial de un sistema de *software* que se utiliza para demostrar la funcionalidad contenida en los casos de uso; de manera que permita que el usuario verifique que el sistema va a satisfacer sus necesidades a partir de una presentación de la interfaz del producto.

El diseño de una aplicación para dispositivo determina con qué facilidad, rapidez y eficiencia puede un usuario llevar a cabo sus tareas. Para obtener instrucciones detalladas sobre el diseño de la interfaz de la aplicación ver **Anexo IV Interfaz de la Aplicación**.

Una aplicación con una interfaz bien diseñada debe tener, además de un buen diseño gráfico, una buena navegabilidad, usabilidad y distribución de los contenidos.

Persiguiendo este objetivo se han seguido para el diseño de la interfaz de usuario del módulo de archivos del Sistema de Gestión Hospitalaria, los siguientes aspectos:

- ❖ Requerir un mínimo proceso de aprendizaje y permitir su utilización desde el primer momento, por cualquier persona que tenga un mínimo dominio de la computación.
- ❖ Garantizar la legibilidad, el color de los textos debe contrastar con el del fondo, y el tamaño de fuente debe ser lo suficientemente grande como para que el usuario no presente problemas al leer.
- ❖ Requerir de los usuarios un mínimo esfuerzo para alcanzar sus objetivos.
- ❖ Mostrar al usuario solamente aquellas opciones a las que tiene derecho a acceder.
- ❖ Mostrar al usuario un mensaje de confirmación, siempre que vaya a realizar una acción relevante sobre el sistema, que le permita asegurarse de que es correcta la opción seleccionada.

3.6 Conclusiones

En este capítulo se realizó una modelación del sistema en términos del análisis y del diseño de cada caso de uso del sistema y por escenarios de estos, como respuesta a la solución propuesta, identificándose las clases interfaces, de control y de entidad de este flujo de trabajo. Mostrando al final como resultado de todos los diagramas realizados un prototipo de interfaz de usuario.

Conclusiones Generales

- ❖ El sistema modelado que se ha propuesto da solución a la situación problemática que lo originó, y su explotación reportará grandes beneficios en los procesos actuales que desarrolla el módulo de archivos en los hospitales de Cuba.
- ❖ Se cumple con el principal objetivo de este trabajo que es diseñar una aplicación para el módulo de archivos, haciendo posible una mejor obtención de información de los pacientes.
- ❖ Como resultado se tendrá una visión más amplia y específica de los requerimientos a cumplir dentro del sistema para el departamento de archivos.

Recomendaciones

- ❖ Con este trabajo exhortamos a una profundización del tema abordado para así dar pie a un refinamiento con vistas a detectar posibles debilidades en el análisis y diseño realizado.
- ❖ Como idea para una futura mejora puede ser el análisis y diseño de una aplicación Web que permita la solicitud de HC mediante listas de espera.
- ❖ Continuar el seguimiento que ha tenido este trabajo en otras tesis como la del rol de la bases de datos e implementación del módulo de archivos.

Bibliografía

- Característica del sistema de Gestión de Archivos BICOM.*, 2007]. Disponible en:
<http://www.biocom.com/sistema/gestionarchivos/index.html>
- Características del Sistema Care2x Entorno Integrado de Cuidados de la Salud* 2007]. Disponible en:
http://care2x.org/index.php?c2x_lang=es&chglang=1
- Características del sistema Gowin HC.* 2006]. Disponible en:
<http://209.61.190.221/spain/corporate/corporateBaseTemplate.csp?pageID=12220>.
- G., F. E. *Introduccion a Extreme Programing. Ingenieria del SW II.*, 9/12/2002 p.
- HIGHSMITH J , O. K. *Adaptive Software Development : A Collaborative Approach to Managing Complex Systems.* Dorset House., 2000. p.
- Historia Clínica y Registros Médicos Informatizados* 2006]. Disponible en:
www.corpece.org.ec/documentos/articulos/internacionales/historia_clinica_registros_medicos_informatizados.doc
- JACOBSON, I. B., G.; RUMBAUGH, J. *El Proceso Unificado de Desarrollo de Software.* Addison Wesley., 2000. p.
- KRUCHTEN, P. "Architectural Blueprints--The 4+1 View Model of Software Architecture". IEEE Software, Institute of Electrical and Electronics Engineers, Noviembre 1995. pp. 42-50 p.
- MESSERLE, J. *Information systems.* IMIA Yearbook of Medical Informatics 98. . New York:, 1999. 273-275. p.
- STAPLETON, J. *Dsdsm Dynamic System Development Method: The Method in Practice.* Addison Wesley., 1997. p.

Anexo I Modelado del Negocio Propuesto

Actores / Trabajadores del negocio

Tabla 31 Actores del negocio propuesto

Actores del negocio	Justificación
Módulo de Inscripción/Admisión	Personal beneficiado en cuanto al trabajo que le pueda brindar el personal de archivo.
Módulo de Turnos Médicos	Responsable de asignarle turnos al paciente y beneficiado en el trabajo realizado por el personal de archivo.
J' Dpto. de Archivo/HC	Es el usuario interesado en obtener cierta información, y está a cargo del levantamiento estadístico del departamento de archivos.

Tabla 32 Trabajadores del negocio propuesto.

Trabajadores del negocio	Justificación
Personal de Archivo	Personal de trabajo del departamento de Archivo que tiene el objetivo de almacenar, manipular y controlar las historias clínicas dentro de este módulo.

Diagrama de Casos de Uso del Negocio.

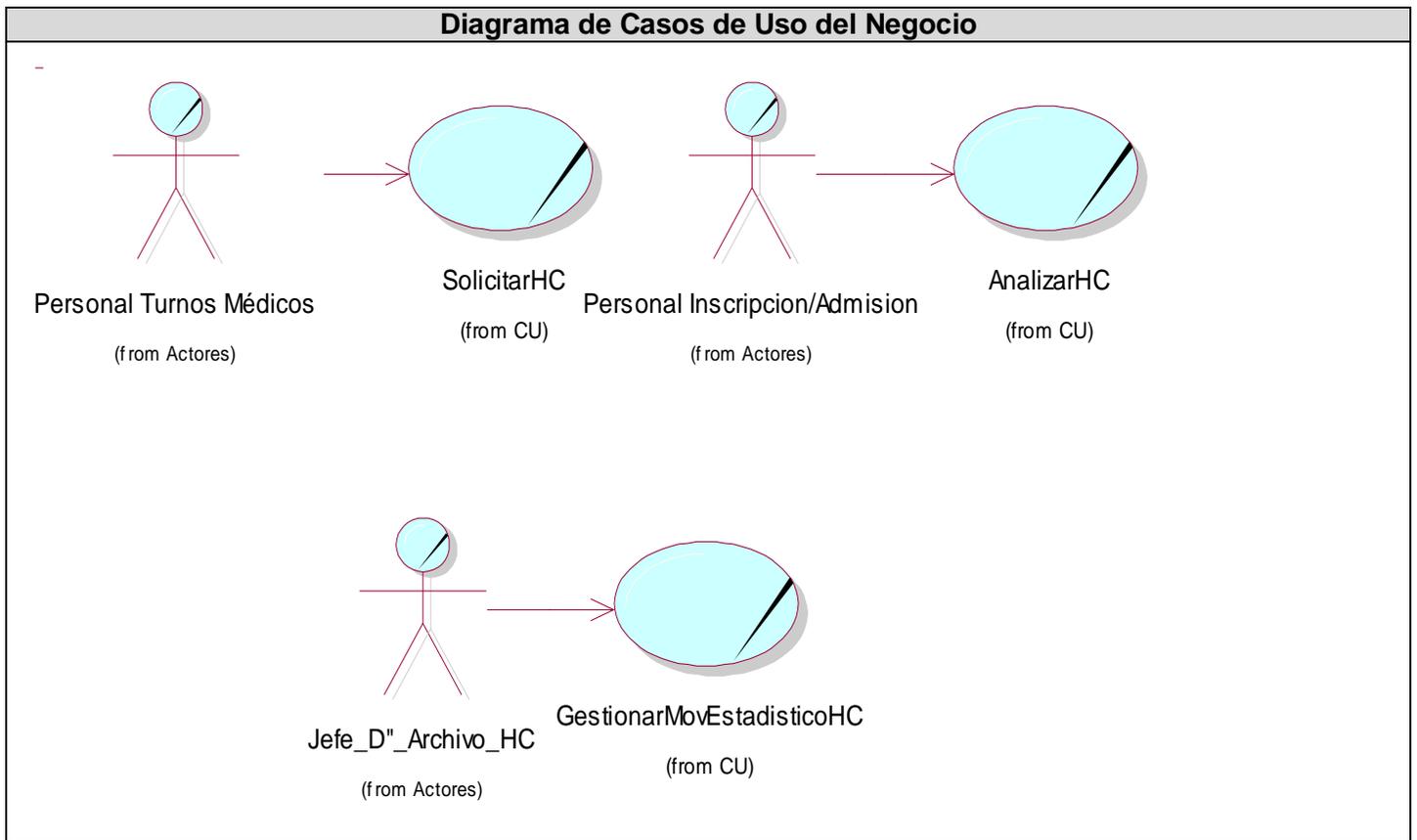


Figura 5 Diagrama de Casos de Uso del Negocio

Diagramas de Actividades

Diagrama de actividad <Analizar HC>

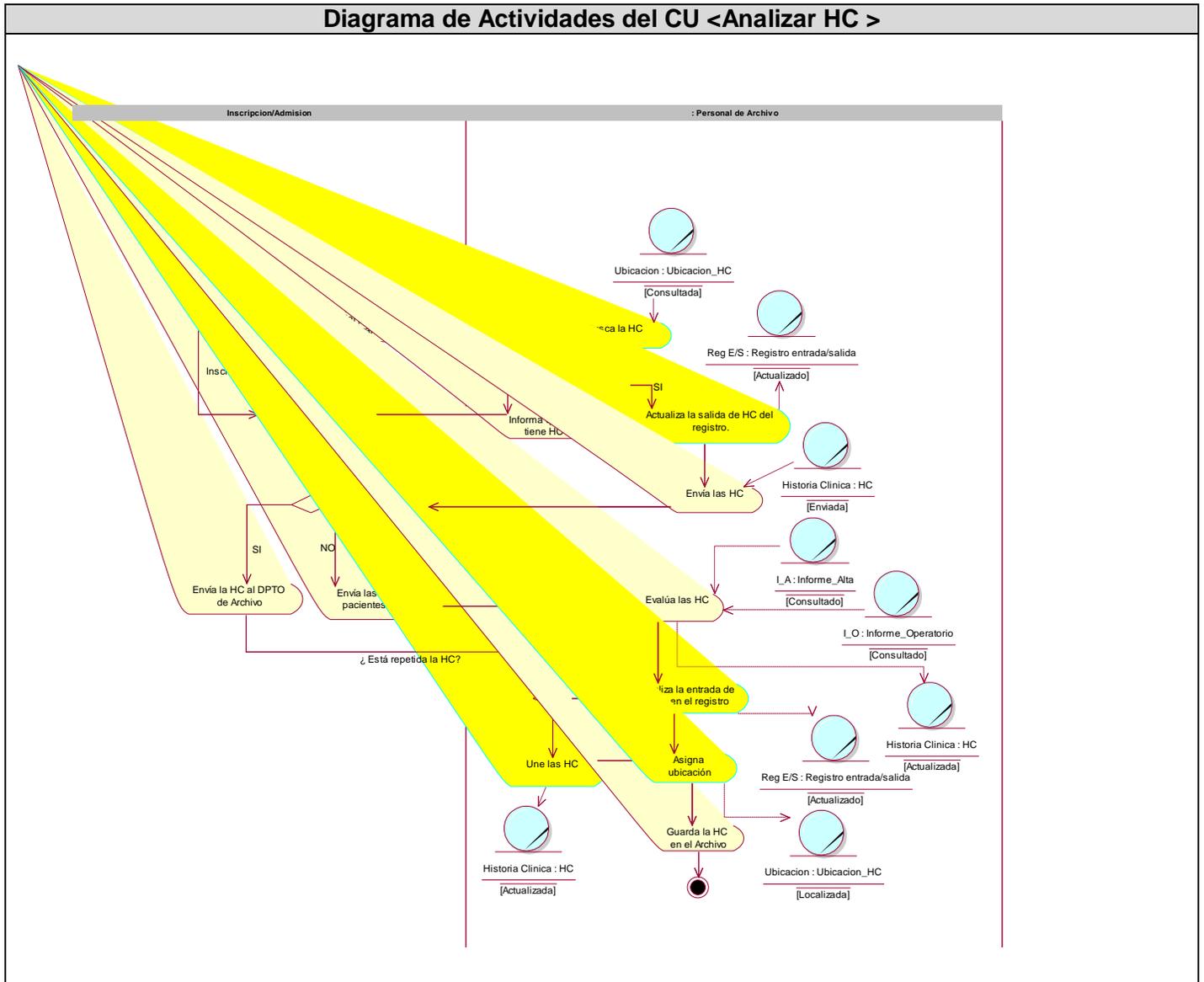


Figura 6 Diagrama de actividad <Analizar HC>

Diagrama de Actividad <Solicitar HC>

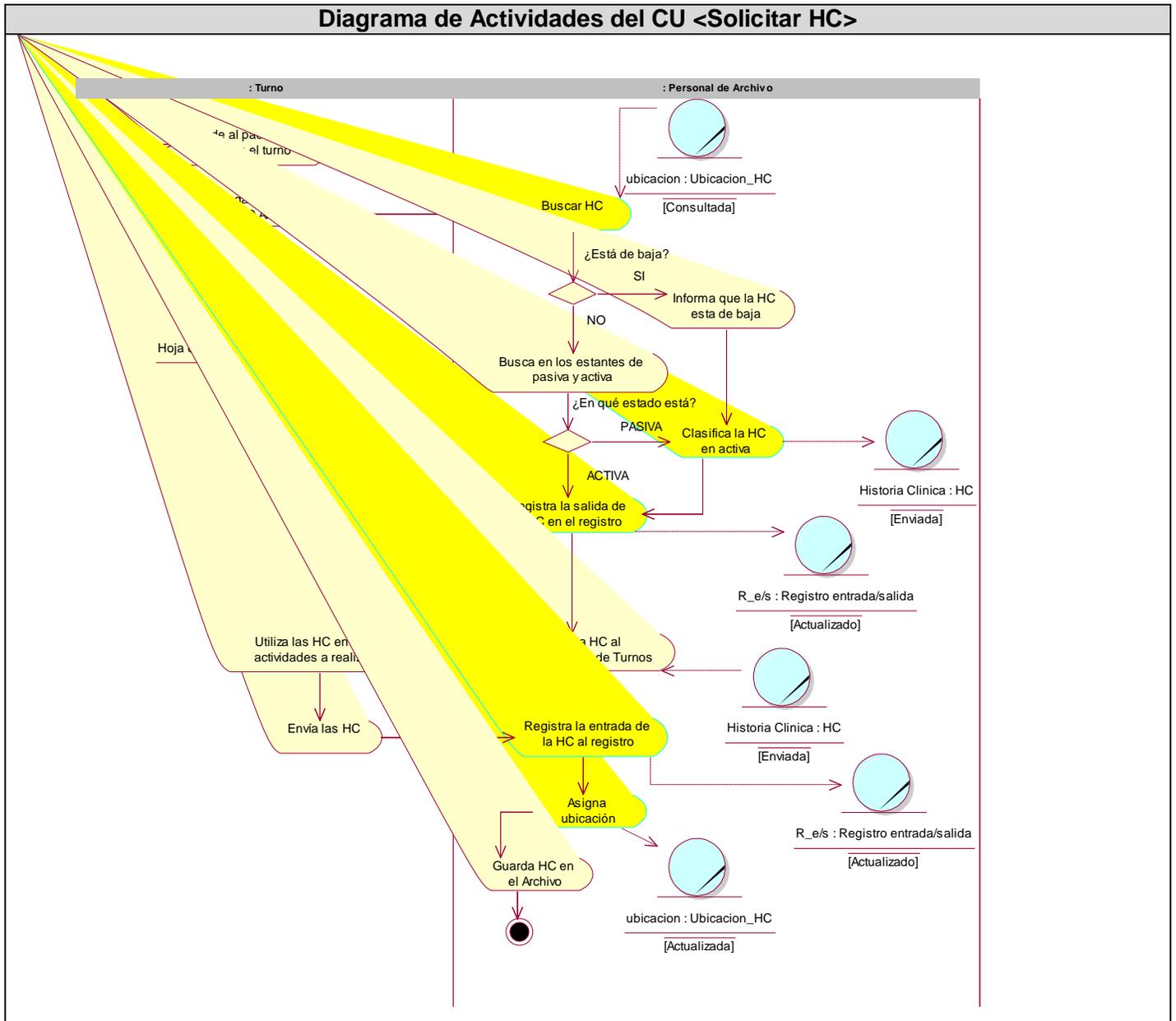


Figura 7 Diagrama de Actividad <Solicitar HC>

Diagrama de actividad <Gestionar Mov Estadísticos HC>

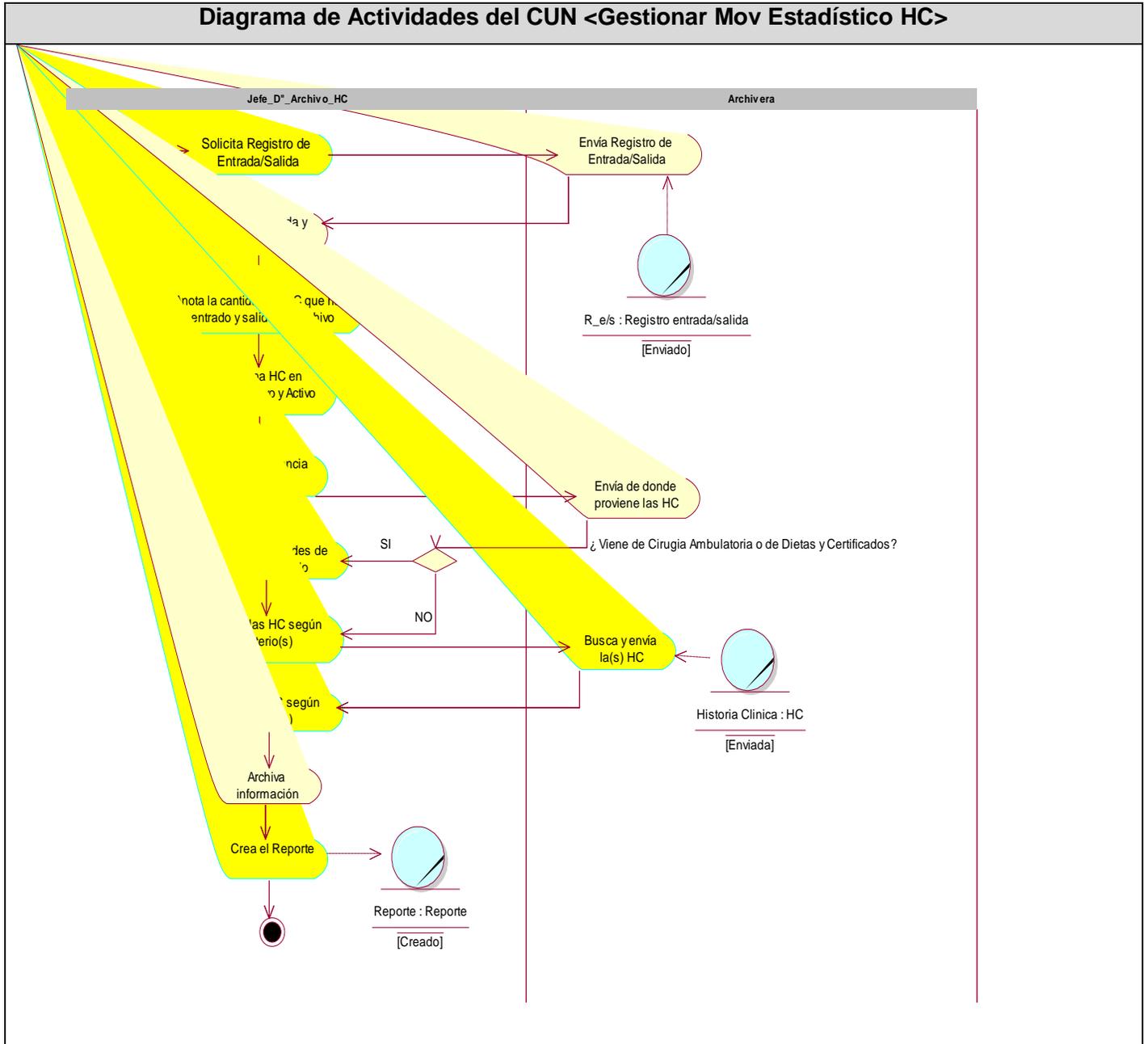


Figura 8 Diagrama de actividad <Gestionar Mov Estadísticos HC>

Modelo de Objetos

Modelos de objetos < Analizar HC>

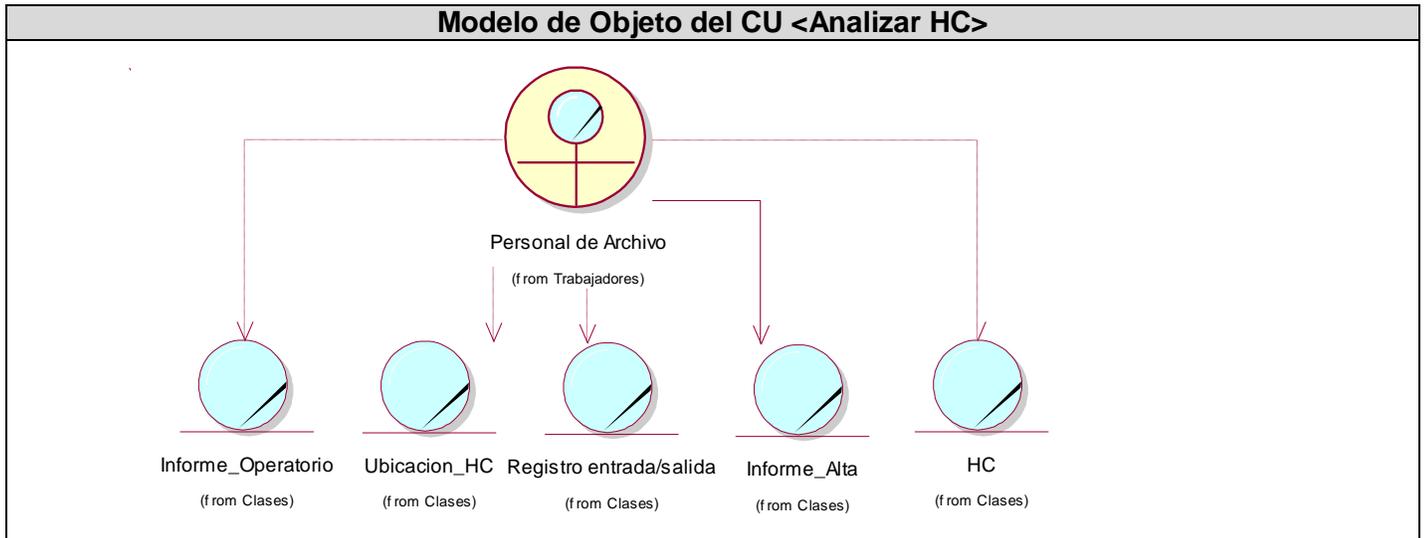


Figura 9 Modelos de objetos <Analizar HC>

Modelo de Objeto < Solicitar HC>

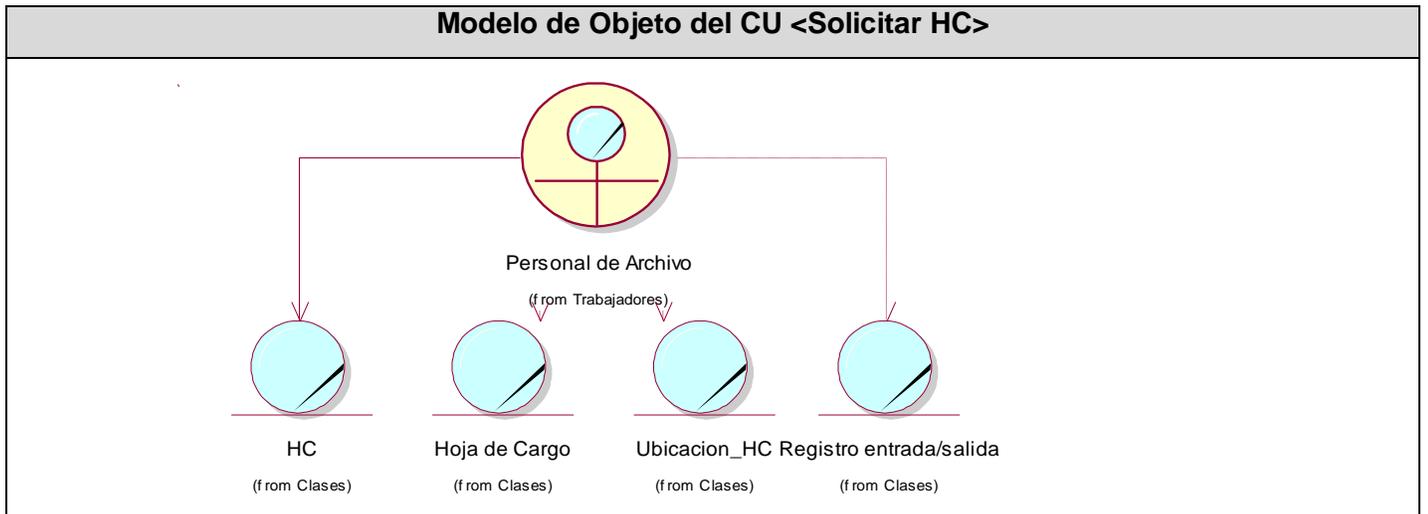


Figura 10 Modelo de Objeto < Solicitar HC>

Modelo de Objeto <Gestionar Mov Estadístico HC>

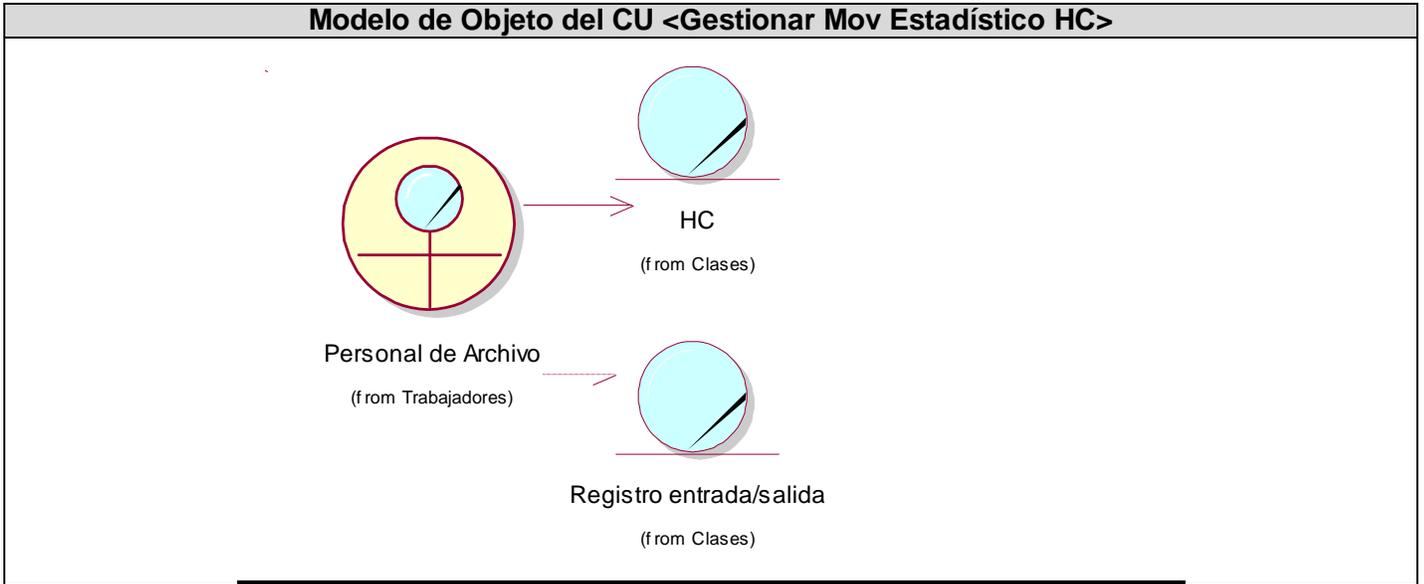


Figura 11 Modelo de Objeto <Gestionar Mov Estadístico HC>

Anexo II Modelado del Sistema

Actores del sistema. Justificación

Tabla 33 Actores del sistema.

Actores del sistema	Justificación
Personal de Archivo	Es la persona que tiene un control de la gestión que se lleva a cabo con las HC dentro del módulo de archivos.
J'DPTO de Archivo	Encargado de interactuar con el sistema de forma tal que tenga un control y conocimiento absoluto del trabajo realizado dentro del Departamento de Archivo creando todos los reportes necesarios del departamento

Casos de uso por ciclos de desarrollo

En el ciclo de desarrollo se propone los casos de uso a realizar, quienes deben cumplir con algunos parámetros como ser guías en los procesos de archivos o registros médicos y su prioridad sea 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:

Tabla 34 CU Arquitectónicamente Significativos

Cód.	Nombre de caso de uso	Paquete	Justificación de la selección.
CU-2	Gestionar HC	CU Arquitectónicamente significativos	CU imprescindible para el funcionamiento del módulo completo.
CU-	Buscar HC	CU Arquitectónicamente significativos	CU necesario para el proceso de búsqueda de las HC

			dentro del módulo de archivos
CU-4	Crear Reporte	CU Arquitectónicamente significativos	CU importante dentro del proceso de control de las HC.

Listado de los casos de uso del sistema.

Tabla 35 Descripción del CUS <Autenticar Usuario>.

CU-1	Autenticar Usuario
Actor:	Usuarios
Descripción:	En este proceso el sistema le pide al usuario: (<i>Usuario y Contraseña de dominio</i>), y por aquí se conoce a que grupo de usuarios pertenece y a que lugares del sistema puede acceder y con que permisos.
Referencia	R1,R1.1,R1.2

Tabla 36 Descripción del CUS <Gestionar Usuario>.

CU-2	Gestionar Usuario
Actor:	Personal de Archivo
Descripción:	Permite insertar un nuevo usuario al sistema, eliminarlo o modificar el rol de este
Referencia	R2, R2.1, R2.2, R2.3

Tabla 37 Descripción del CUS <Gestionar HC>.

CU-3	Gestionar HC
Actor:	Personal de Archivo
Descripción:	Permite clasificar las HC, registrar su E/S, insertar los datos básicos de una HC nueva en el archivo, unir las en caso de varias HC o realizar alguna búsqueda.
Referencia	R2, R2.1, R2.2, R2.3, R2.4, R.2.5, R2.6, R3(include)

Tabla 38 Descripción del CUS <Buscar HC>.

CU-4	Buscar HC
Actor:	Personal de Archivo, J´DPTO de Archivo
Descripción:	Este proceso hace posible la búsqueda de una o varias HC por distintos campos de búsqueda.
Referencia	R-3,

Tabla 39 Descripción del CUS <Crear Reporte>.

CU-5	Crear Reporte
Actor:	J´DPTO de Archivo(inicia)
Descripción:	Caso de uso con la responsabilidad de crear un reporte de acuerdo a un tiempo definido por el usuario haciendo uso de la búsqueda de las HC a partir de los campos de búsqueda que conste el reporte.
Referencia	R4, R3 (include)

Modelo de casos de uso del Sistema.

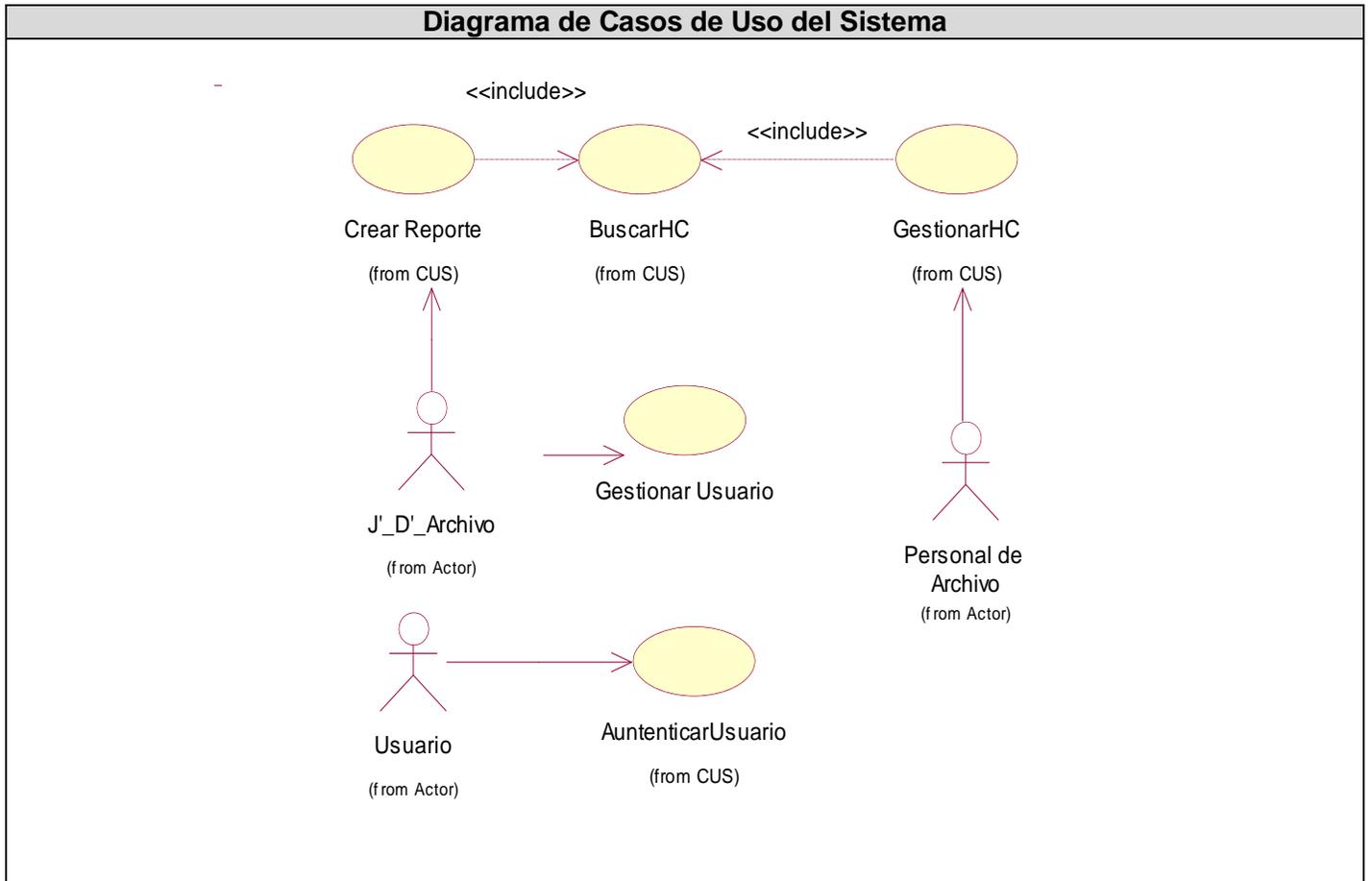


Figura 12 Modelo de casos de uso del Sistema.

Descripción de los Casos de Uso

Tabla 40 CUS <Gestionar HC>

CU-2	Gestionar HC
Actores	Personal de Archivo
Propósito	Permite clasificar, insertar, registrar, unir, y buscar una HC.
Resumen	Es aquí donde se clasifican las HC, se registran su Entrada/Salida, se insertan, organizan las HC que se entran en el archivo, se unen en caso de varias HC y se realiza también búsquedas de estas.
Referencias	R2, R2.1, R2.2, R2.3, R2.4, R.2.5, R2.6, CU-3 Buscar HC(include).
Precondiciones	Usuario del sistema ya autenticado.
Poscondiciones	Se clasifica una HC, se inserta, se registra su Entrada/Salida, se organiza mediante una ubicación a dar, unirlas en caso de varias HC y se realizan búsquedas de estas.
Curso Normal de los Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario del sistema necesita clasificar una HC, eliminarla, registrar su entrada/salida HC, organizarla por orden y unir en caso de que existan varias.	1.1 El sistema ejecuta alguna de las siguientes acciones: a) Si decide clasificar una determinada Historia Clínica, ir a la sección "Clasificar HC". b) Si decide insertar una nueva Historia Clínica, ir a la sección "Insertar HC". c) Si decide unir una Historia Clínica, ir a la sección " Unir

	<p>HC".</p> <p>d) Si decide registrar la entrada/salida de una Historia Clínica, ir a la sección " Registrar la entrada/salida de HC".</p> <p>f) Si se decide buscar una determinada HC ir a la sección "Buscar HC".</p>
Sección "Clasificar HC"	
Acciones del Actor	Respuesta del Sistema
<p>2. El usuario del sistema busca la HC a clasificar por su número de HC Remitirse a la tabla del CU-3 Buscar HC.</p> <p>2.2 El usuario del sistema cambia tanto el estado como la localización de la HC porque se ve afectada en el momento en que tenga un nuevo estado.</p>	<p>2.1 El sistema muestra la interfaz con los datos pertenecientes a esa HC como el nombre, que le permite modificar el nuevo estado asignado a la HC, carnet de identidad, estado y localización</p> <p>2.3 El sistema verifica la localización de la HC.</p>
Curso alternativo	
<p>2.3 En caso de que no se le pueda otorgar esta localización envía un mensaje de error.</p>	
Sección "Insertar HC"	
Acciones del Actor	Respuesta del Sistema
<p>3.</p> <p>3.2 El usuario del sistema inserta los datos a llenar.</p>	<p>3.1 El sistema muestra la interfaz con campos obligatorios para insertar una nueva HC dentro del departamento de Archivos como datos personales del paciente, número de la HC, etc.</p> <p>3.3 el sistema por su parte verifica la entrada los datos.</p>
Curso alternativo	

3.3 Envía un mensaje de error.	
Sección “Unir HC”	
Acciones del Actor	Respuesta del Sistema
4.2 El usuario del sistema inserta los datos nuevos que pide la interfaz como el número de la HC provisional, la dirección y edad actual.	4.1 El sistema muestra la interfaz donde se muestra los campos de su número de HC, dirección, edad y localización dentro del archivo. 4.3 El sistema verifica la veracidad de los datos introducidos.
Curso alternativo	
4.3 El sistema muestra un mensaje de error.	
Sección “Registrar entrada/salida de la HC”	
Acciones del Actor	Respuesta del Sistema
5. El usuario del sistema inserta los datos a registrar en caso de entrada o salida de la HC.	5.1 El sistema valida los datos antes de guardarlos. 5.2 Se verifica que los campos obligatorios estén llenos.
Curso alternativo	
5.1 emite un mensaje de error sino son válidos los datos insertados.	
5.2 Se emite un mensaje donde se le informar al usuario la necesidad de llenar los campos obligatorios.	
Sección “Buscar HC”	
Acciones del Actor	Respuesta del Sistema
6. El usuario selecciona la opción Buscar una HC remitirse a la tabla del CU-3	
Curso Alternativo	

Prioridad	Crítico
------------------	---------

Tabla 415 CUS <Buscar HC>

CU-3	Buscar HC
Actores	Personal de Archivo, J' DPTO de Archivo
Propósito	Buscar las HC que están dentro del Archivo, mostrando su ubicación dentro del mismo.
Resumen	El caso de uso se inicia cuando al personal de Archivo le solicitan una HC y estos proceden a la búsqueda, después de la búsqueda la HC es entregada.
Referencias	R-3
Precondiciones	Tiene que solicitarse una HC al departamento de Archivo.
Poscondiciones	Se entrega satisfactoriamente la HC.
Curso Normal de los Eventos:	
Acciones del Actor	Respuestas del Sistema
<p>1- El usuario del sistema selecciona la opción buscar HC.</p> <p>3- El usuario del sistema determina los campos por donde realizará la búsqueda.</p> <p>4- El personal de archivo introduce los datos correspondientes.</p>	<p>2- El sistema muestra la interfaz de búsqueda de HC, permitiendo la búsqueda por varios campos.</p> <p>5- El sistema verifica los datos y procede con la búsqueda, devuelve la HC y su localización dentro del Archivo.</p>
Curso Alternativo	

5 – Datos introducidos no válidos.	6- El sistema emite un mensaje de error en correspondencia con el error en los datos.
Prioridad	Crítico

Tabla 42 CUS <Crear Reporte>

CUS-4	CrearReporte
Actores	Jefe del departamento de Archivo
Propósito	Realizar los reportes del departamento de Archivo.
Resumen	Este caso de uso se inicia en el momento en que se necesita conocer como ha estado el movimiento de las HC en el departamento, para controlar este movimiento se realizan algunos reportes que posibilitan un amplio conocimiento del mismo.
Referencias	R4, CU-3 Buscar HC (include).
Precondiciones	Se necesita un movimiento de HC en el departamento.
Poscondiciones	Se realizan los reportes necesarios para controlar el flujo de entrada y salida de las HC.
Curso Normal de los Eventos:	
Acciones del Actor	Respuestas del Sistema
1-El jefe del departamento de Archivo selecciona la opción de crear reportes.	2-El sistema le muestra la interfaz de crear los reportes.

<p>3-El jefe del departamento selecciona los reportes que desea crear.</p> <p>5-El jefe del departamento introduce los datos necesarios para realizar las búsquedas de estos para su reporte. Remitirse a la tabla del CU-3 Buscar HC</p>	<p>4-El sistema le pide confirmación de que ese es el reporte a crear.</p> <p>6-Verifica la autenticidad de los datos, y crea el reporte.</p>
Curso Alternativo	
<p>6a- Datos introducidos por el usuario incorrectos</p>	<p>6a.1- El Sistema verifica que el mes y el año entrado no sean mayores que el mes y año actual, y en caso de que sean mayores emite un error.</p>
<p>Prioridad</p>	<p>Crítico</p>

Anexo III Análisis y diseño del sistema.

Diagrama de Clases del Análisis

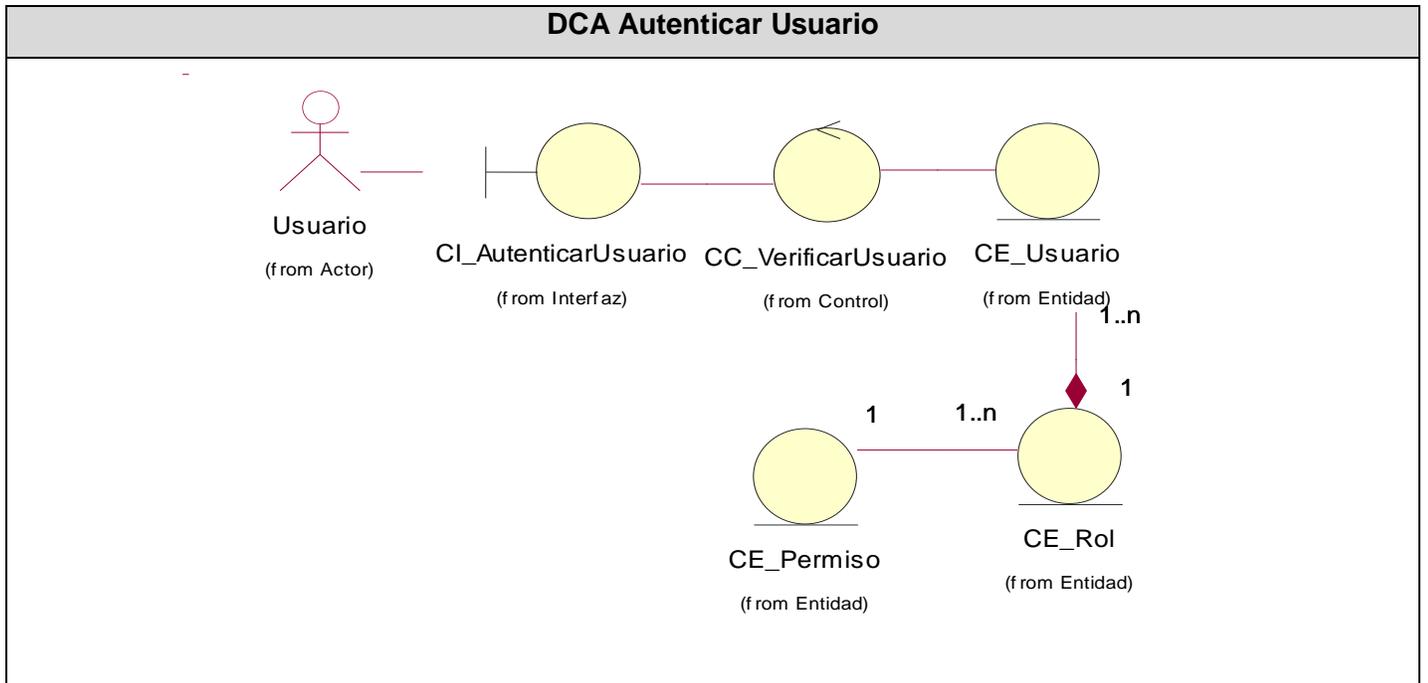


Figura 13 DCA del CUS <Autenticar Usuario>

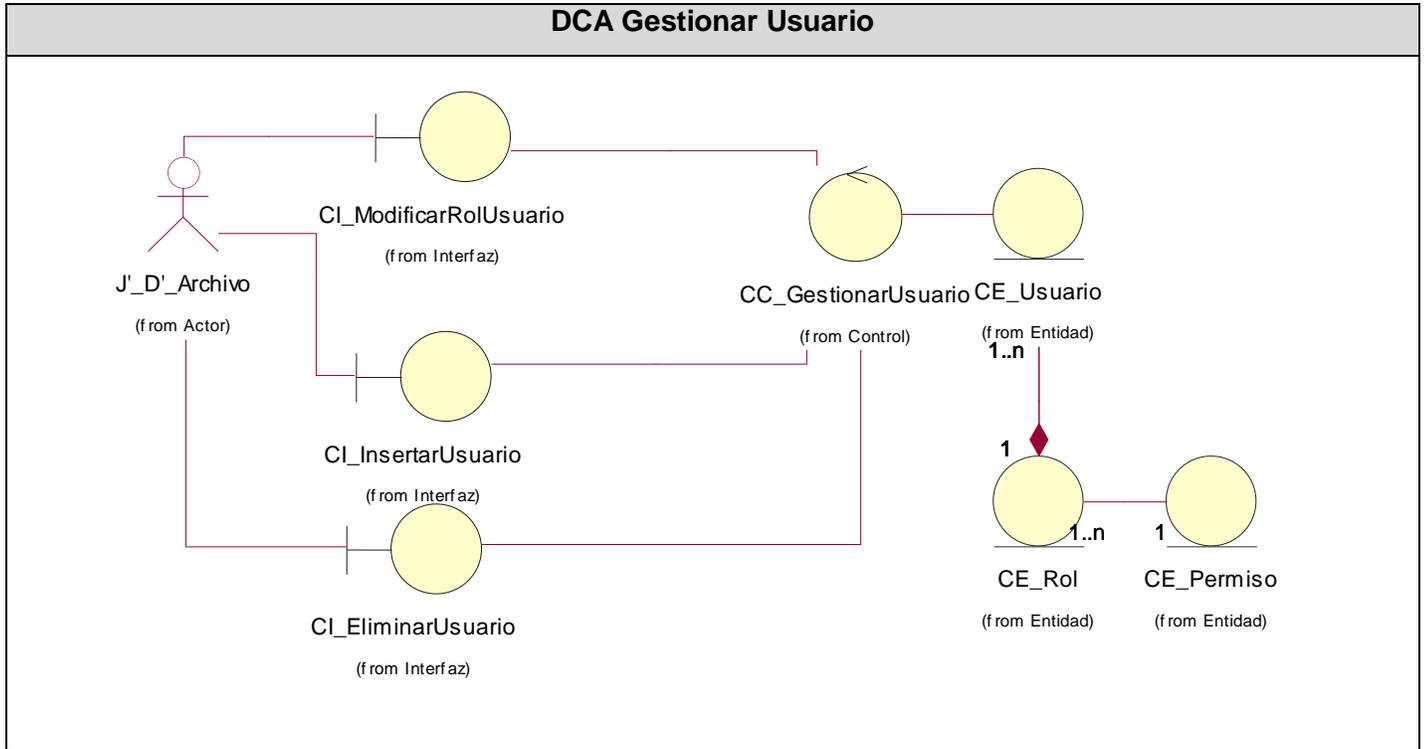


Figura 14 DCA del CUS <Gestionar Usuario>

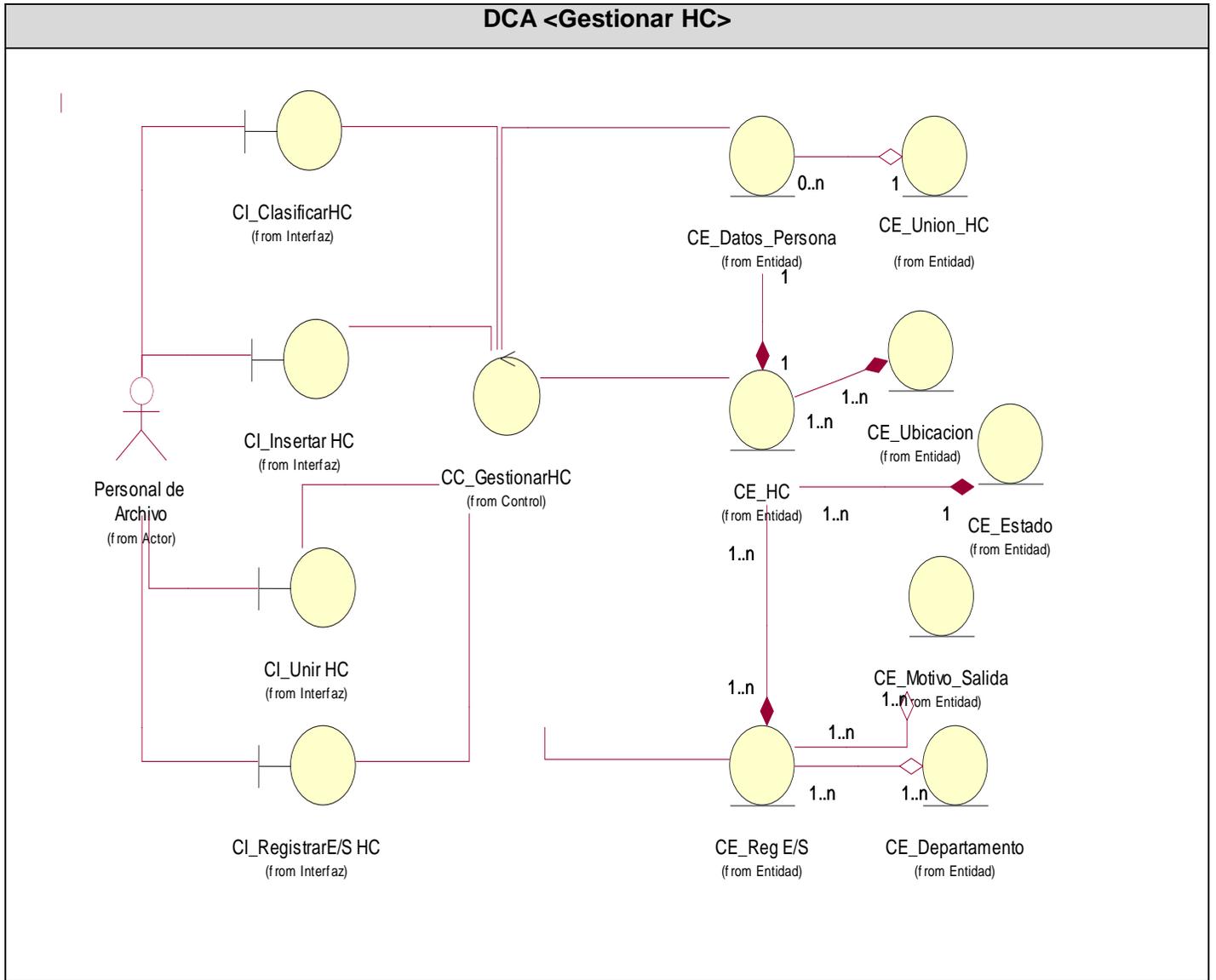


Figura 15 DCA del CUS <Gestionar HC>

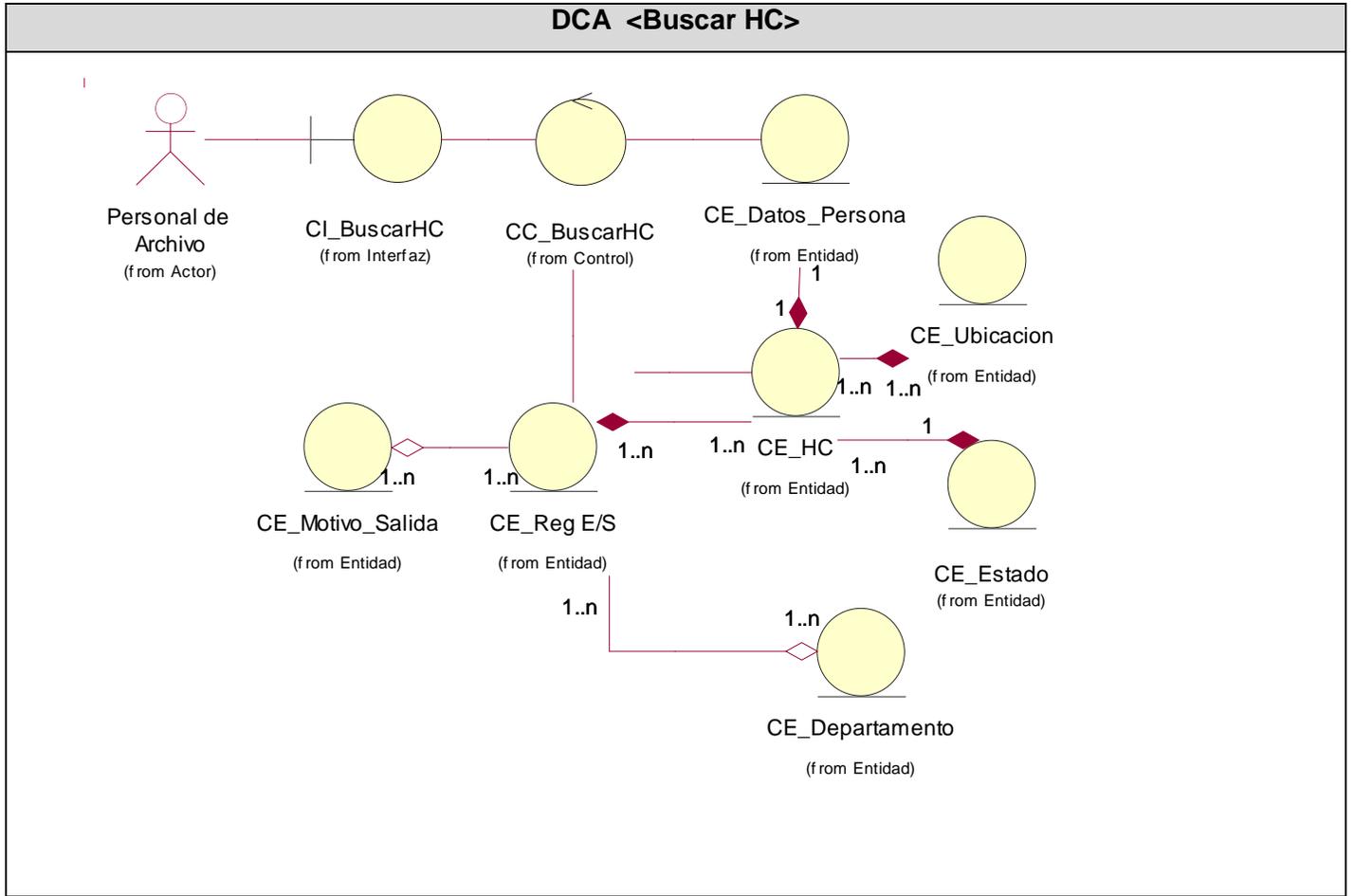


Figura 16 DCA del CUS <Buscar HC>

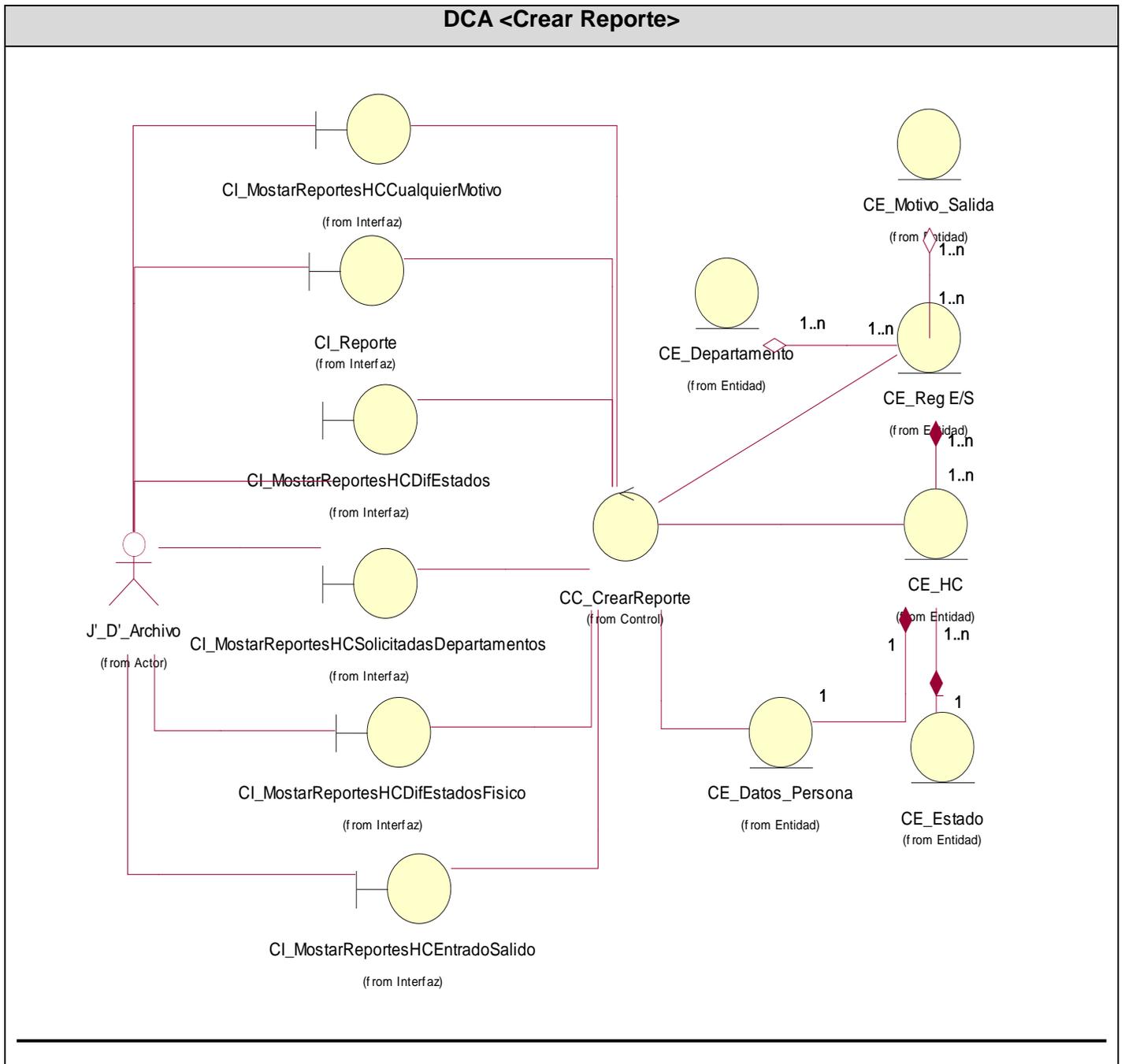


Figura 14 DCA del CUS <Crear Reporte>

Diagramas de Clases del Diseño

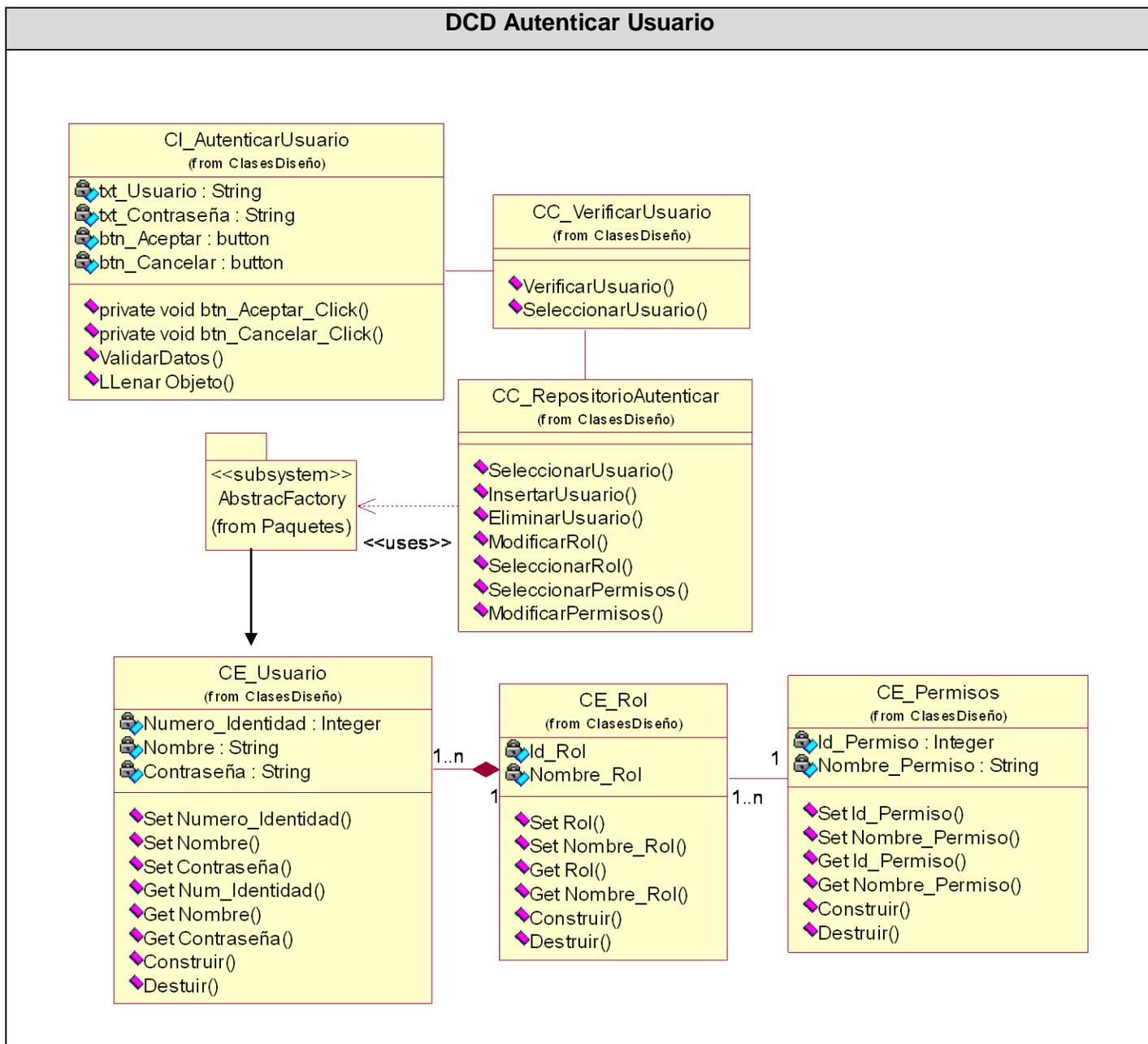


Figura 15 DCD Del CUS <Autenticar Usuario>

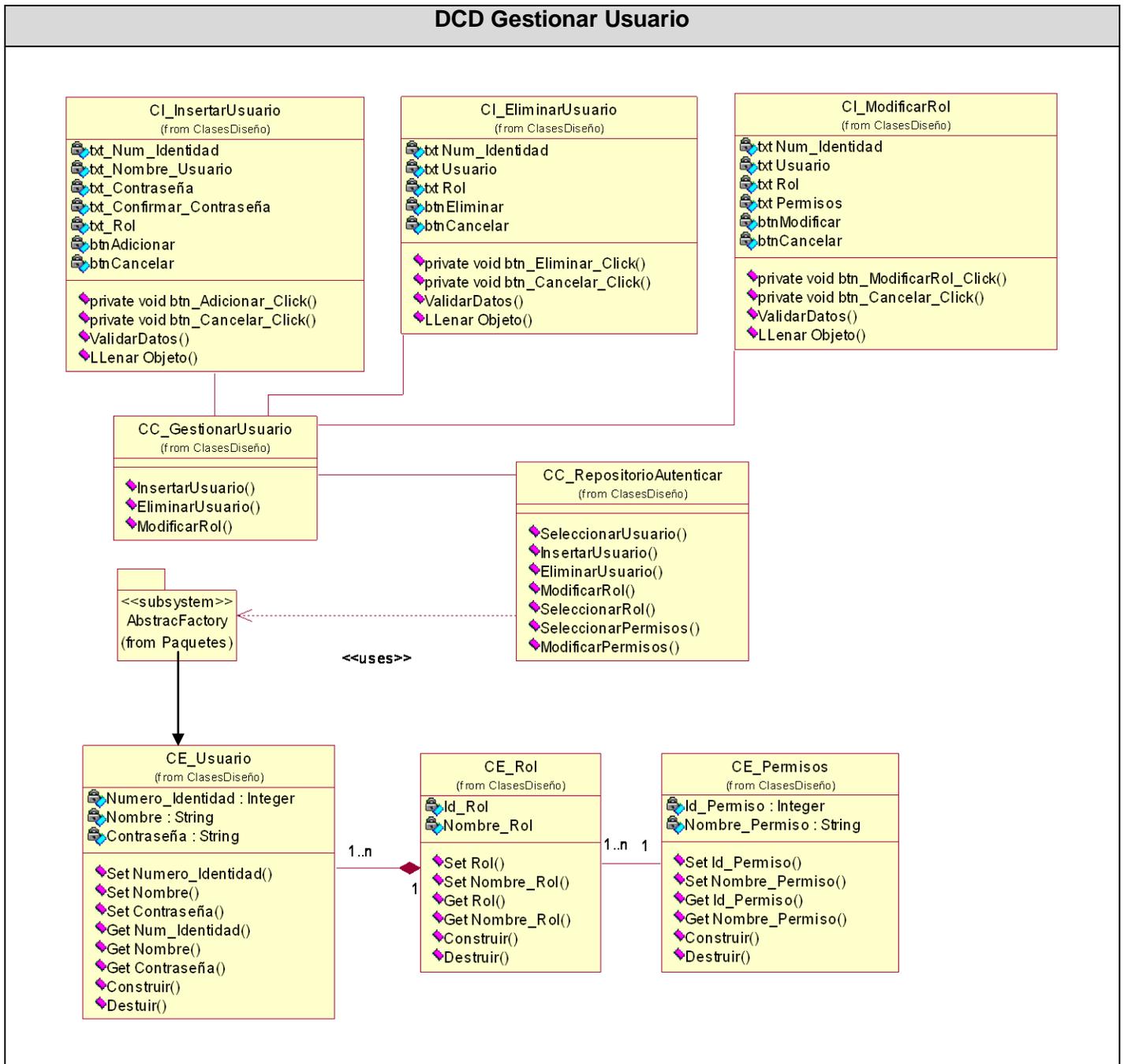


Figura 168 DCD Del CUS <Gestionar Usuario>

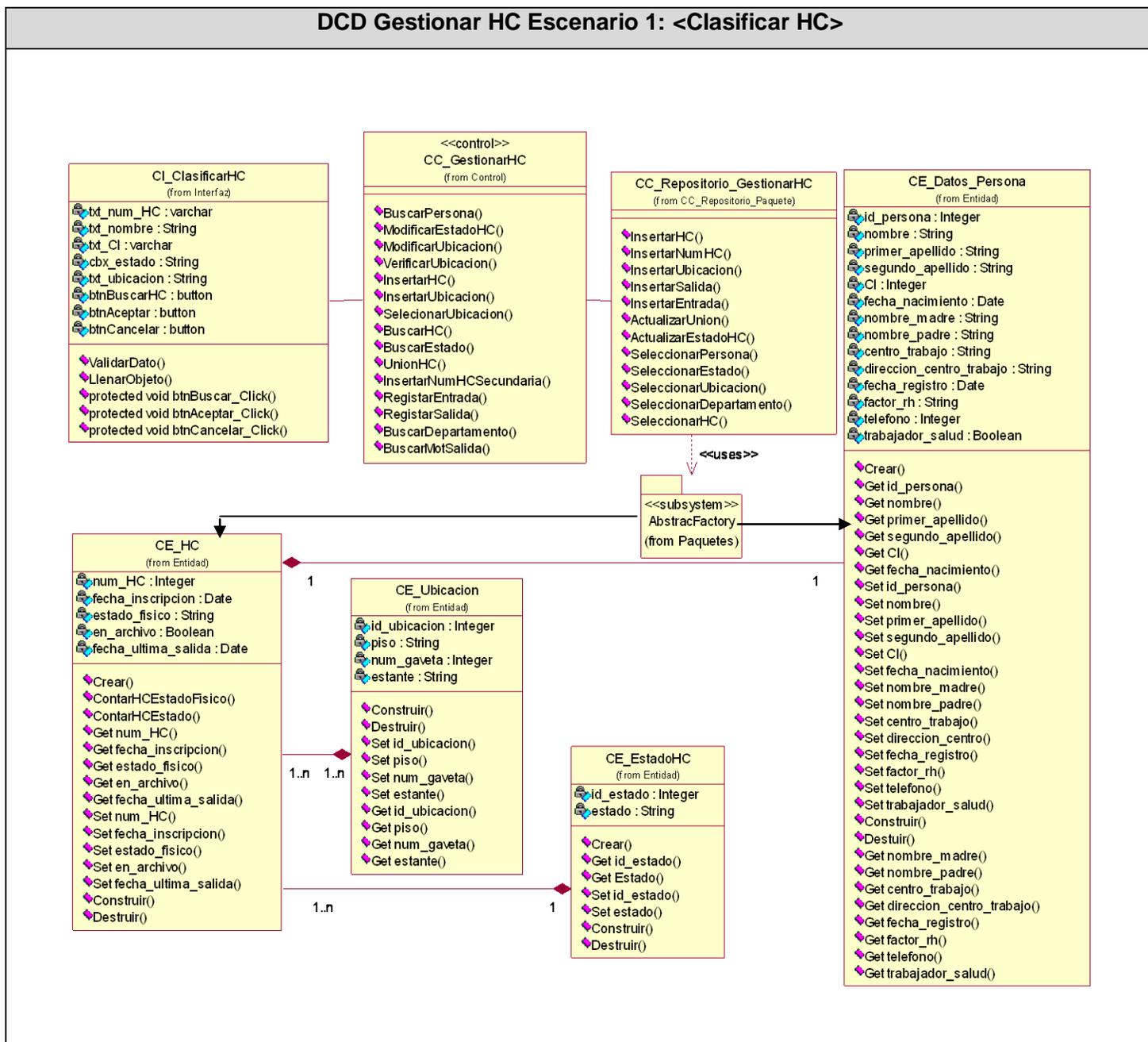


Figura 179 DCD del CUS Gestionar HC Escenario 1: <Clasificar HC>

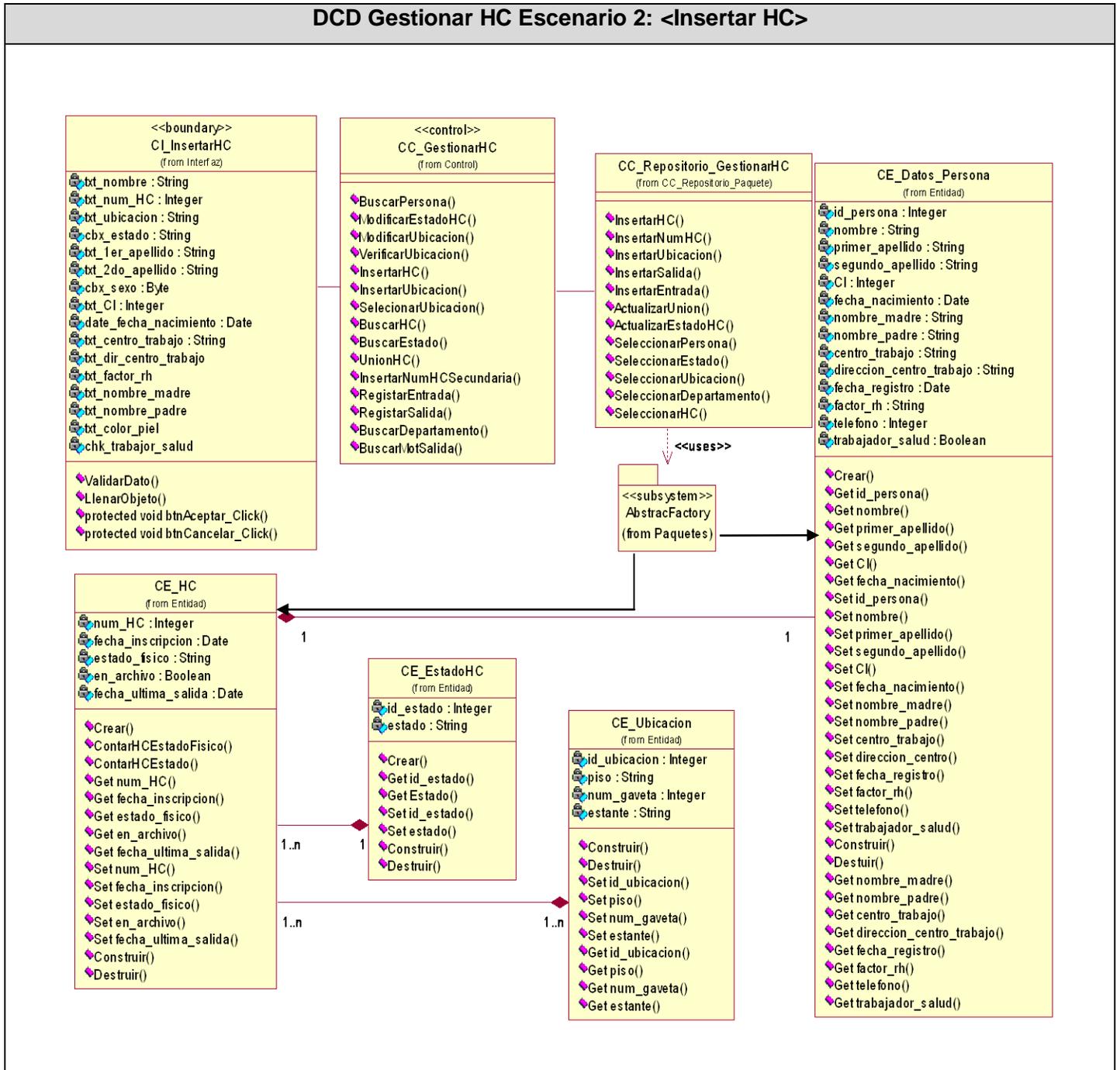


Figura 20 DCD Gestionar HC Escenario 2: <Insertar HC>

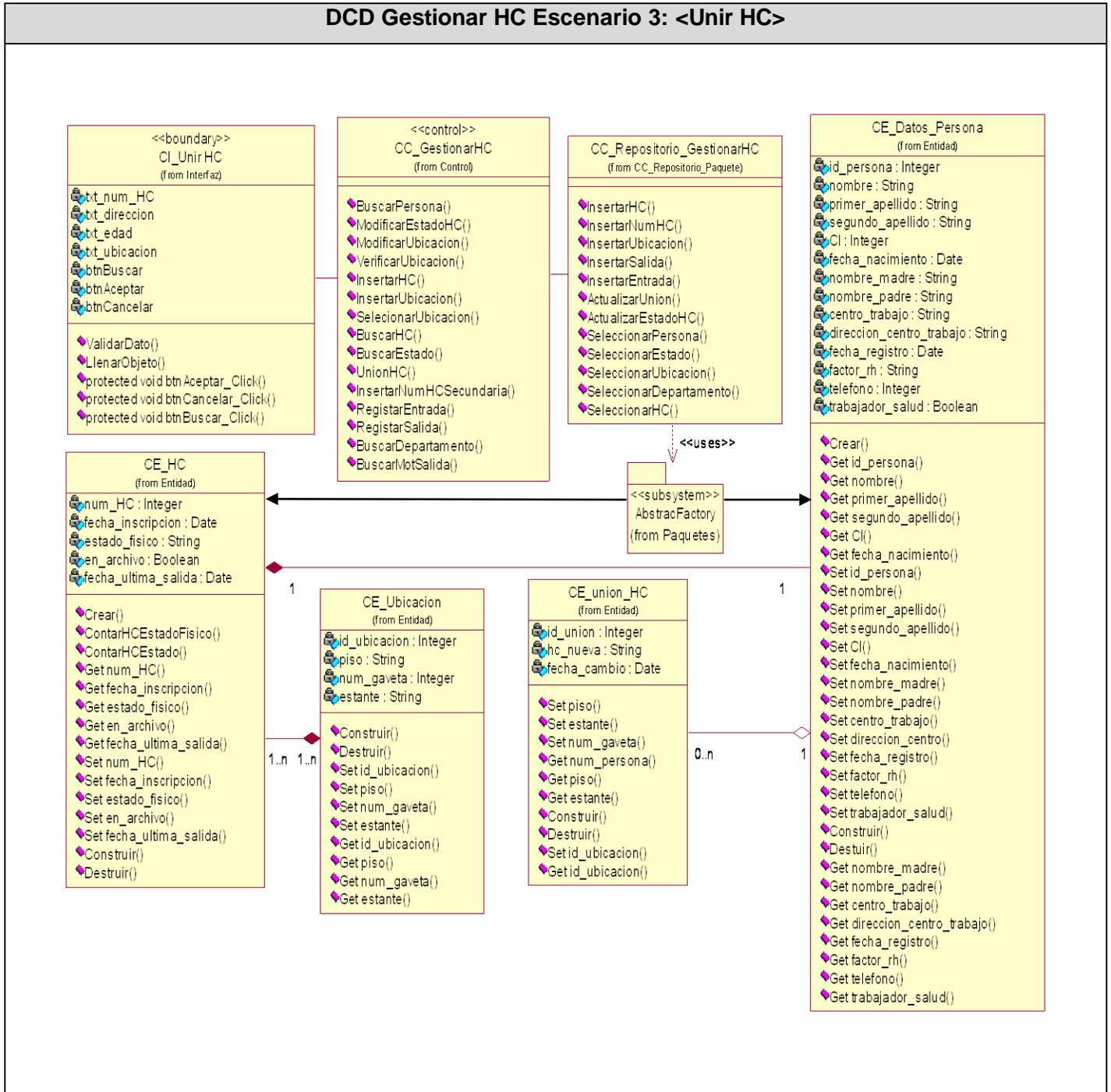


Figura 21 DCD Gestionar HC Escenario 4: <Unir HC>

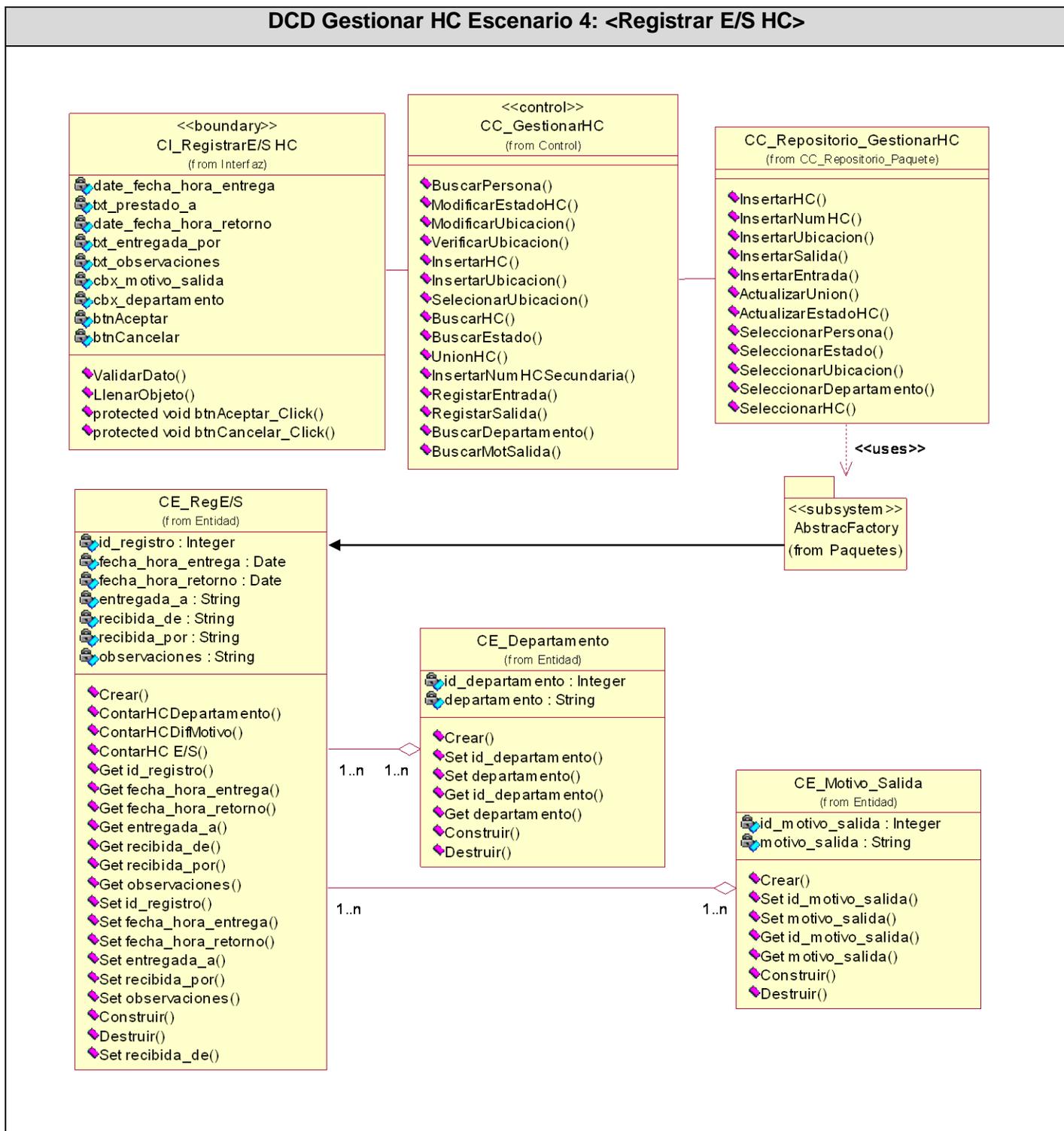


Figura 22 DCD Gestionar HC Escenario 5: <Registrar E/S HC>

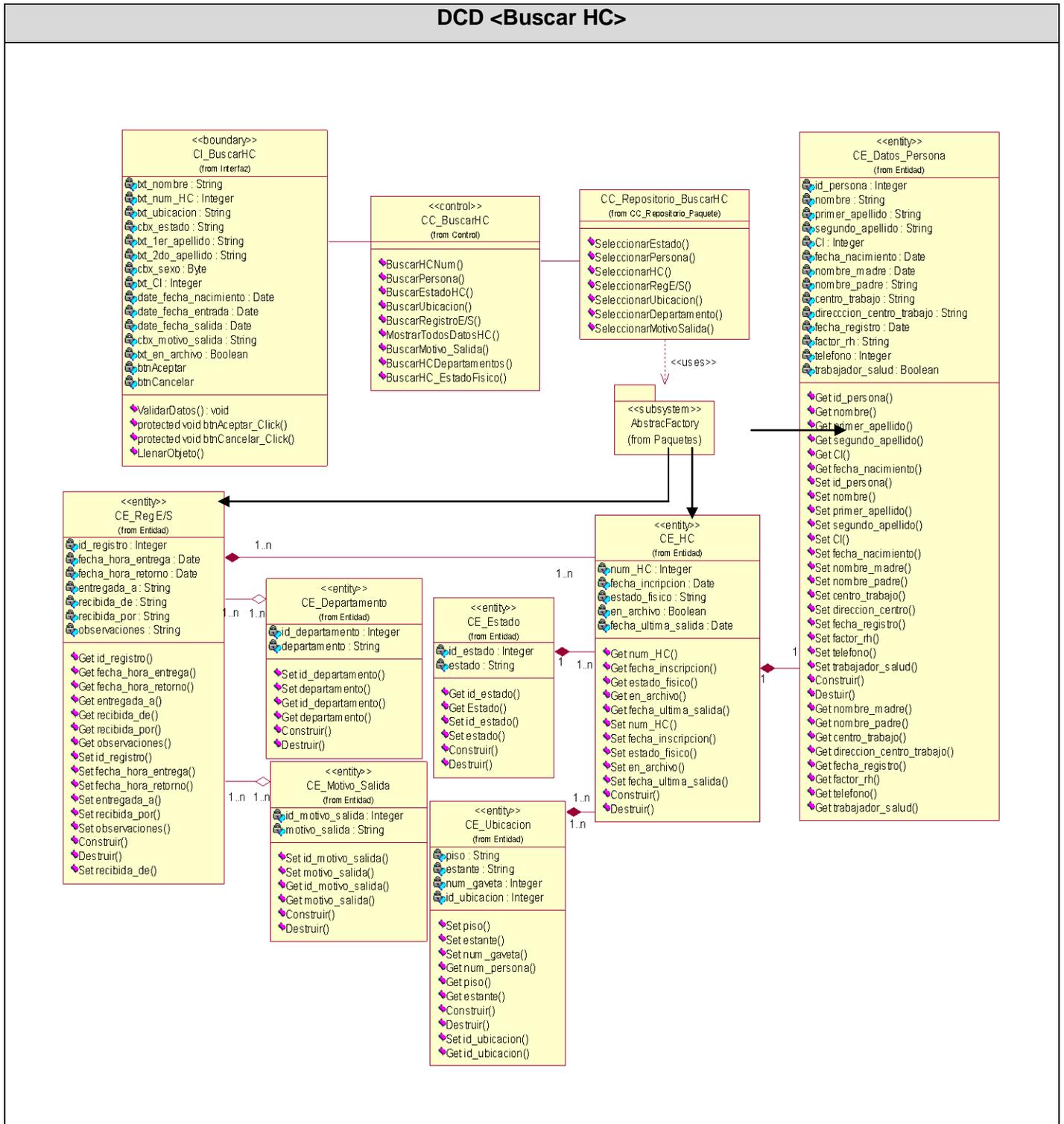


Figura 23 DCD <Buscar HC>

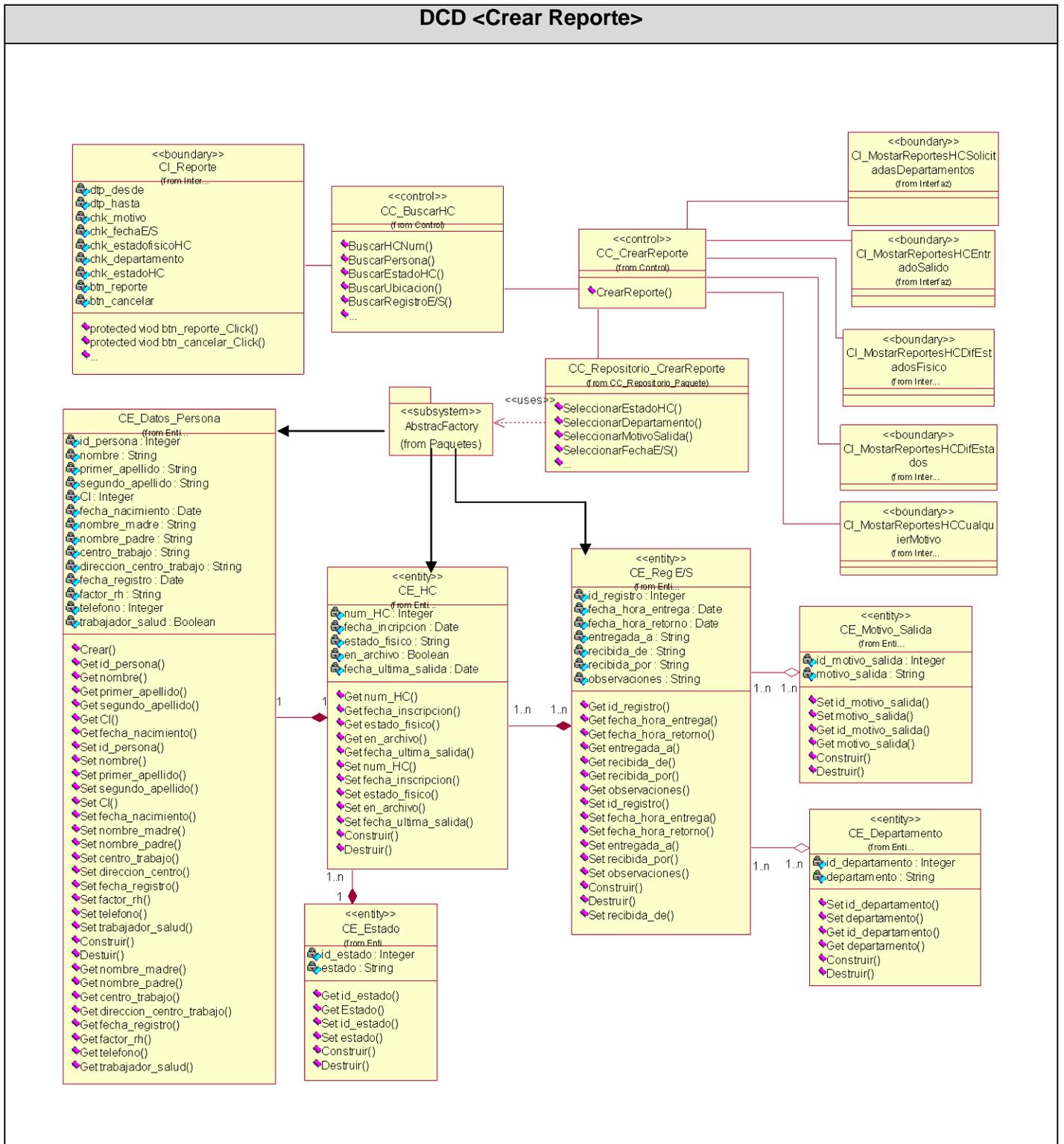


Figura 24 DCD <Crear Reporte>

Diagramas de Interacción del Diseño

Diagramas de Secuencia del Diseño

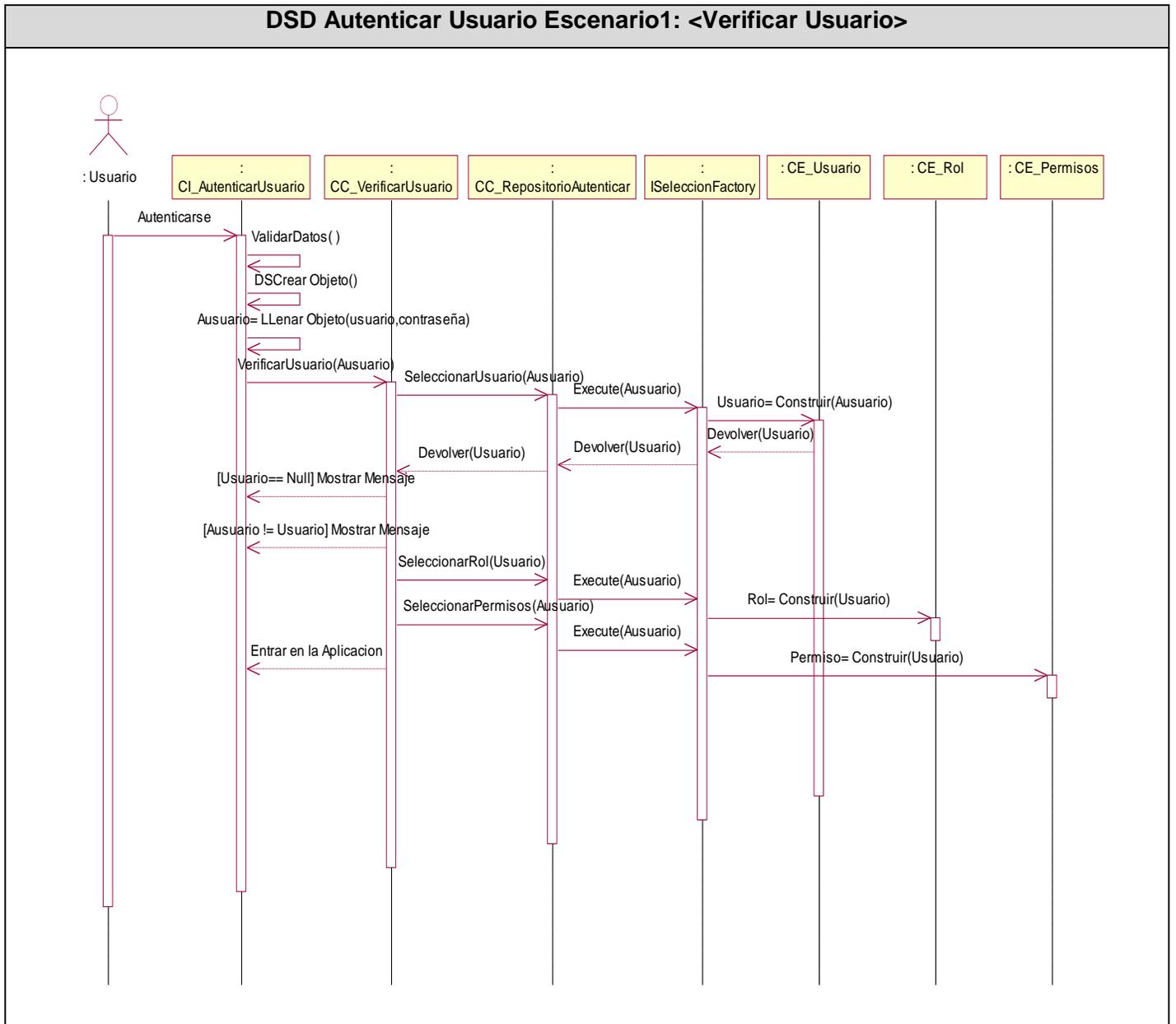


Figura 25 DCD <Verificar Usuario>

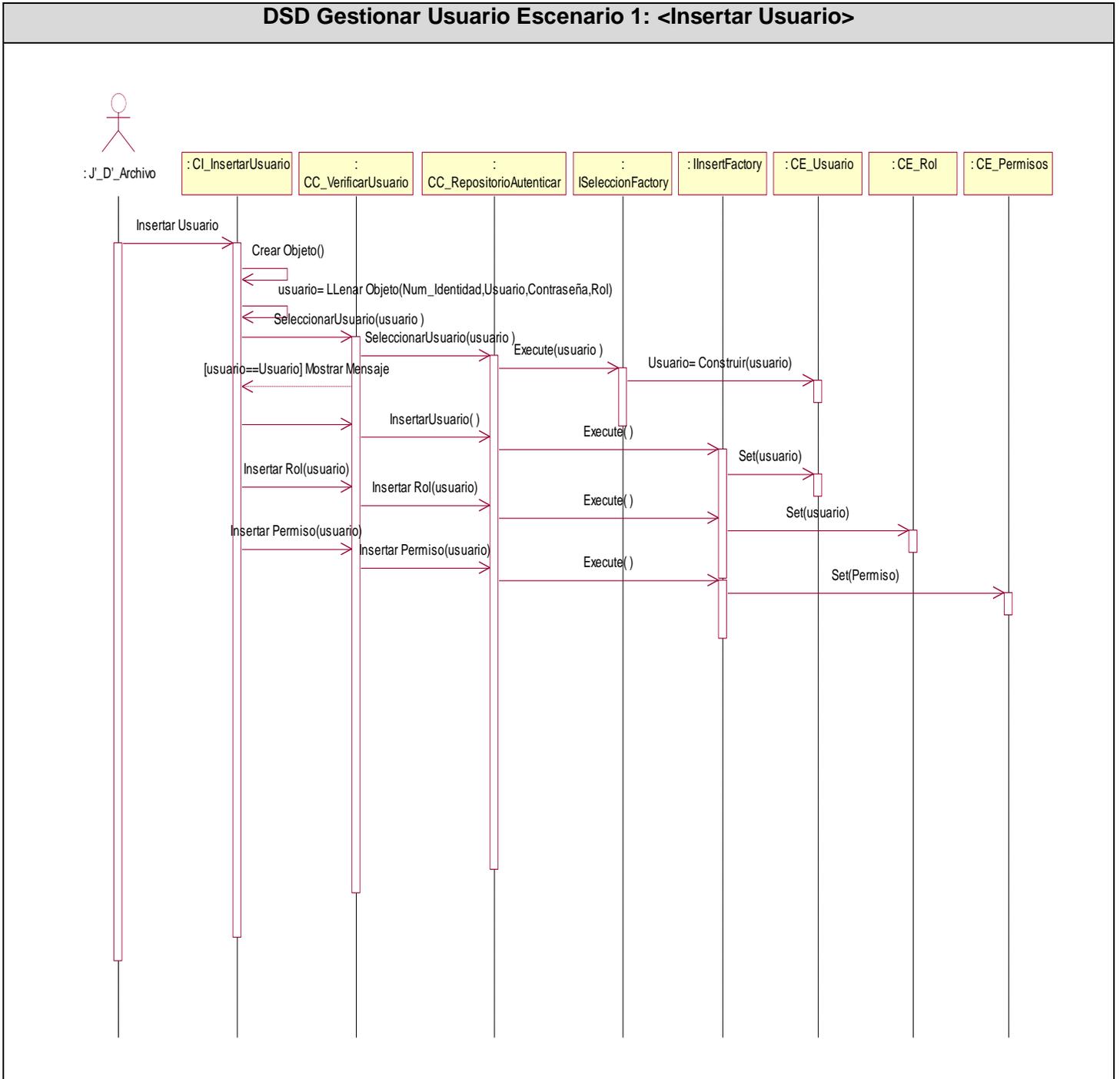
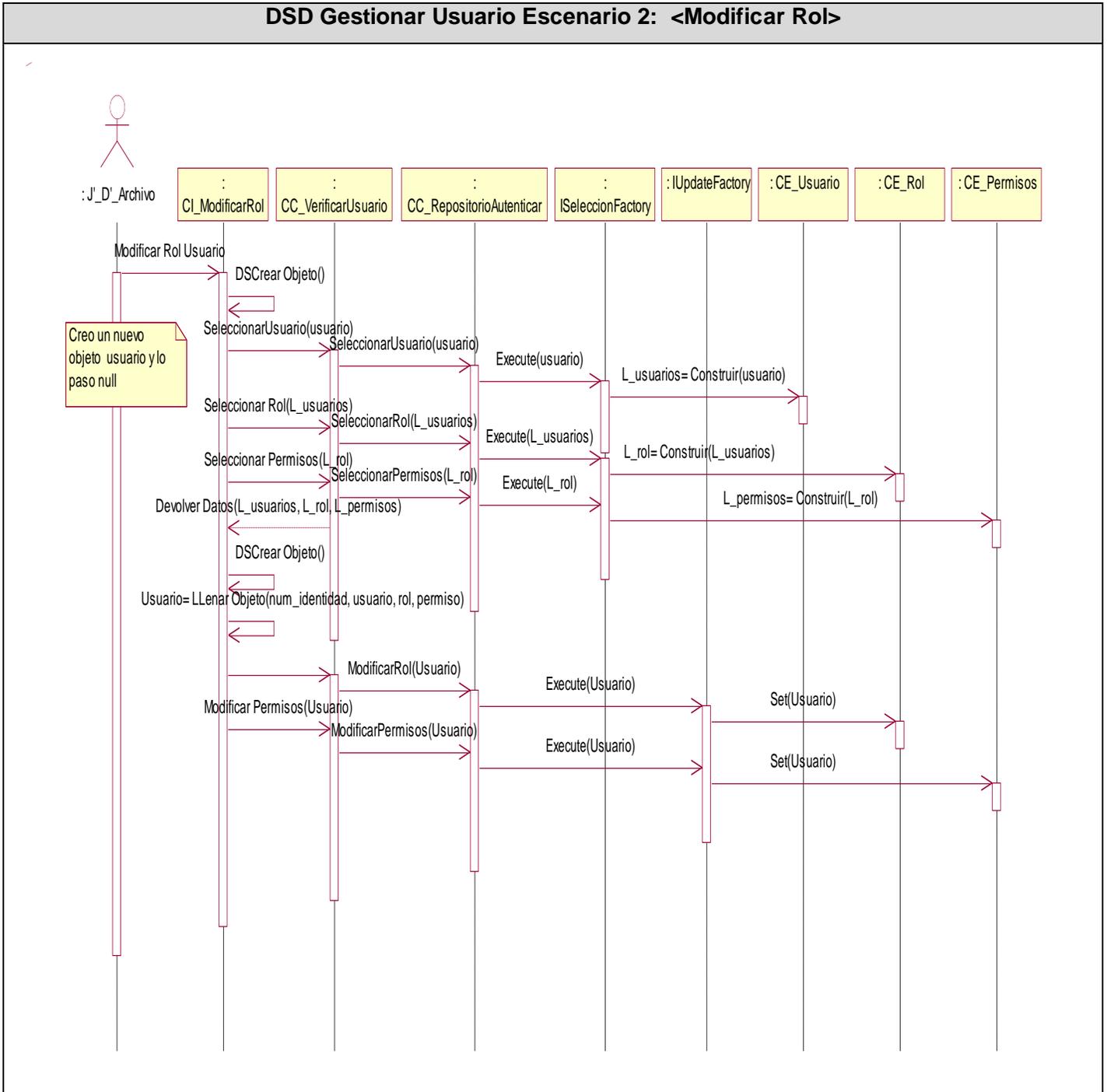
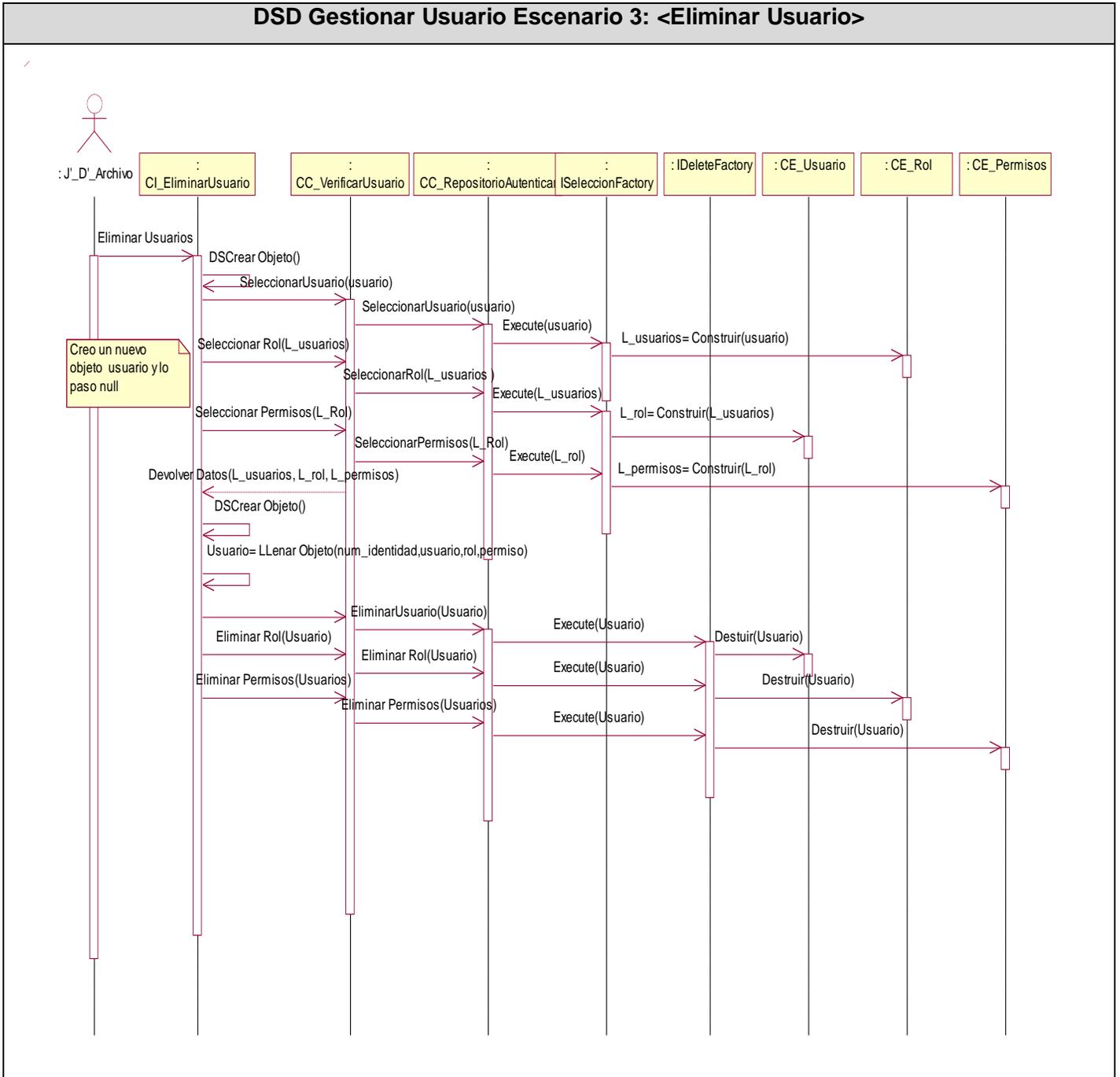


Figura 26 DCD <Insertar Usuario>





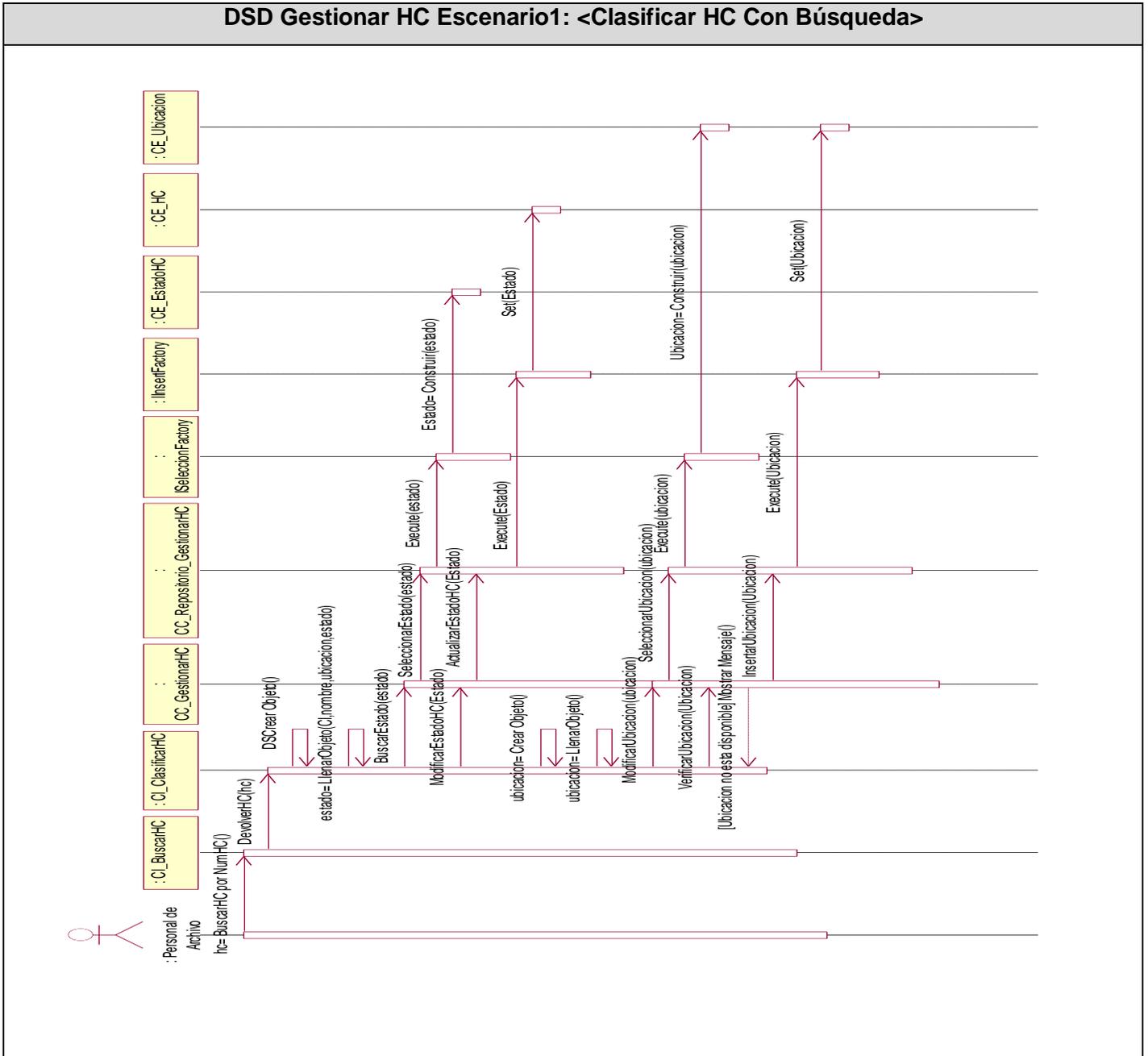


Figura 29 DSD Escenario 1: Clasificar HC con Búsqueda

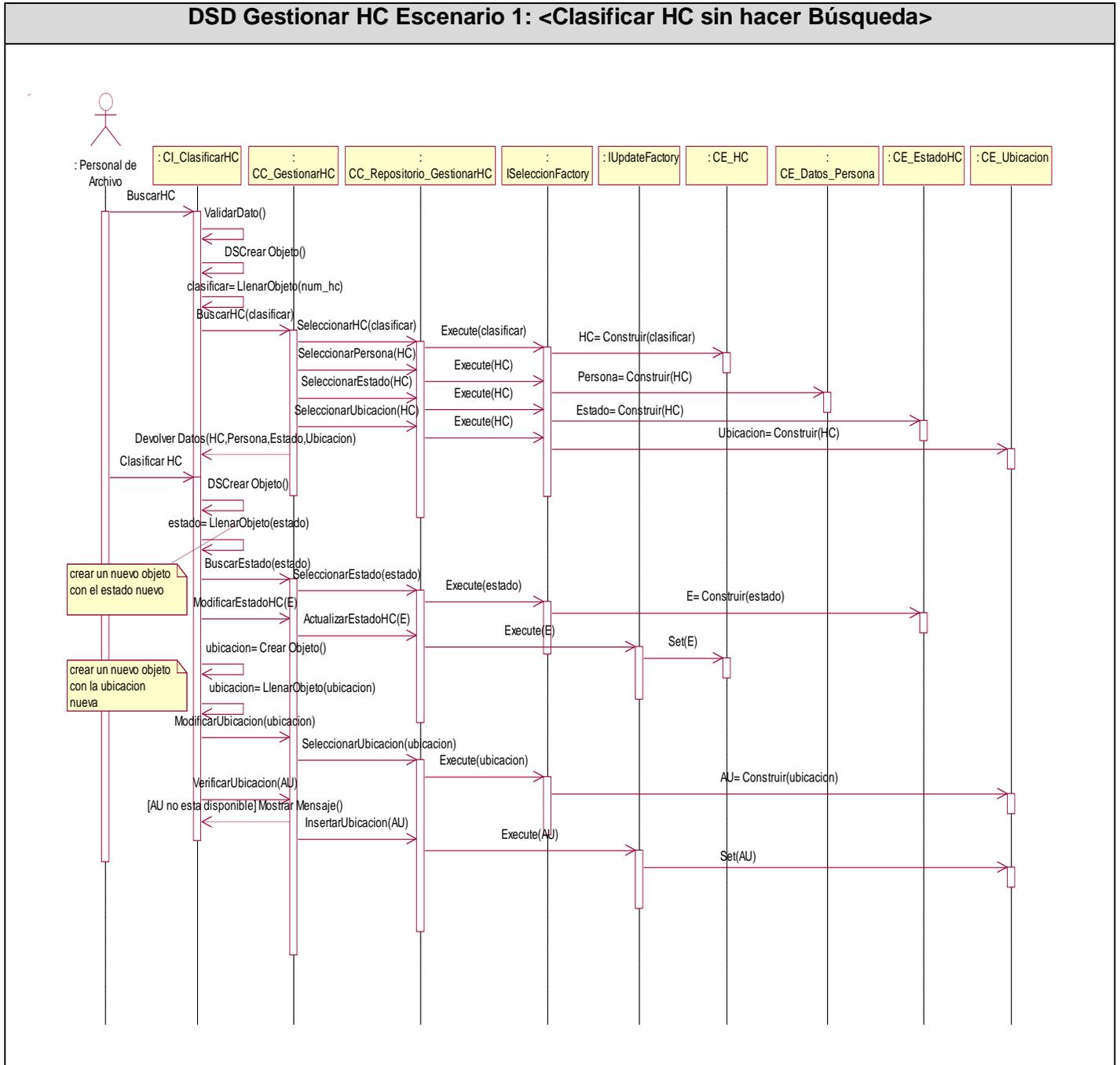


Figura 30 DSD Escenario 1: Clasificar HC sin hacer Búsqueda

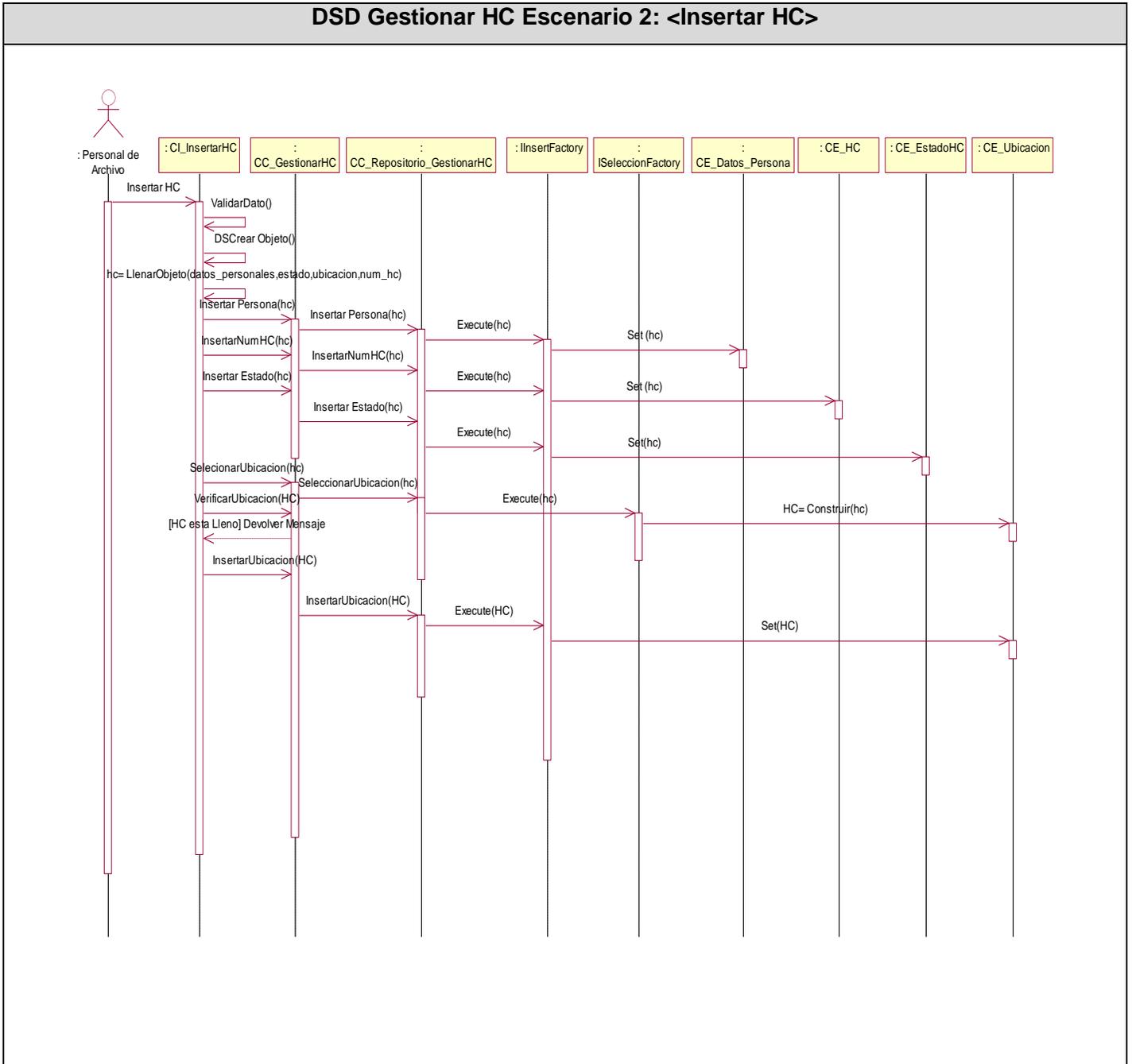


Figura 31 DSD Escenario 2: Insertar HC

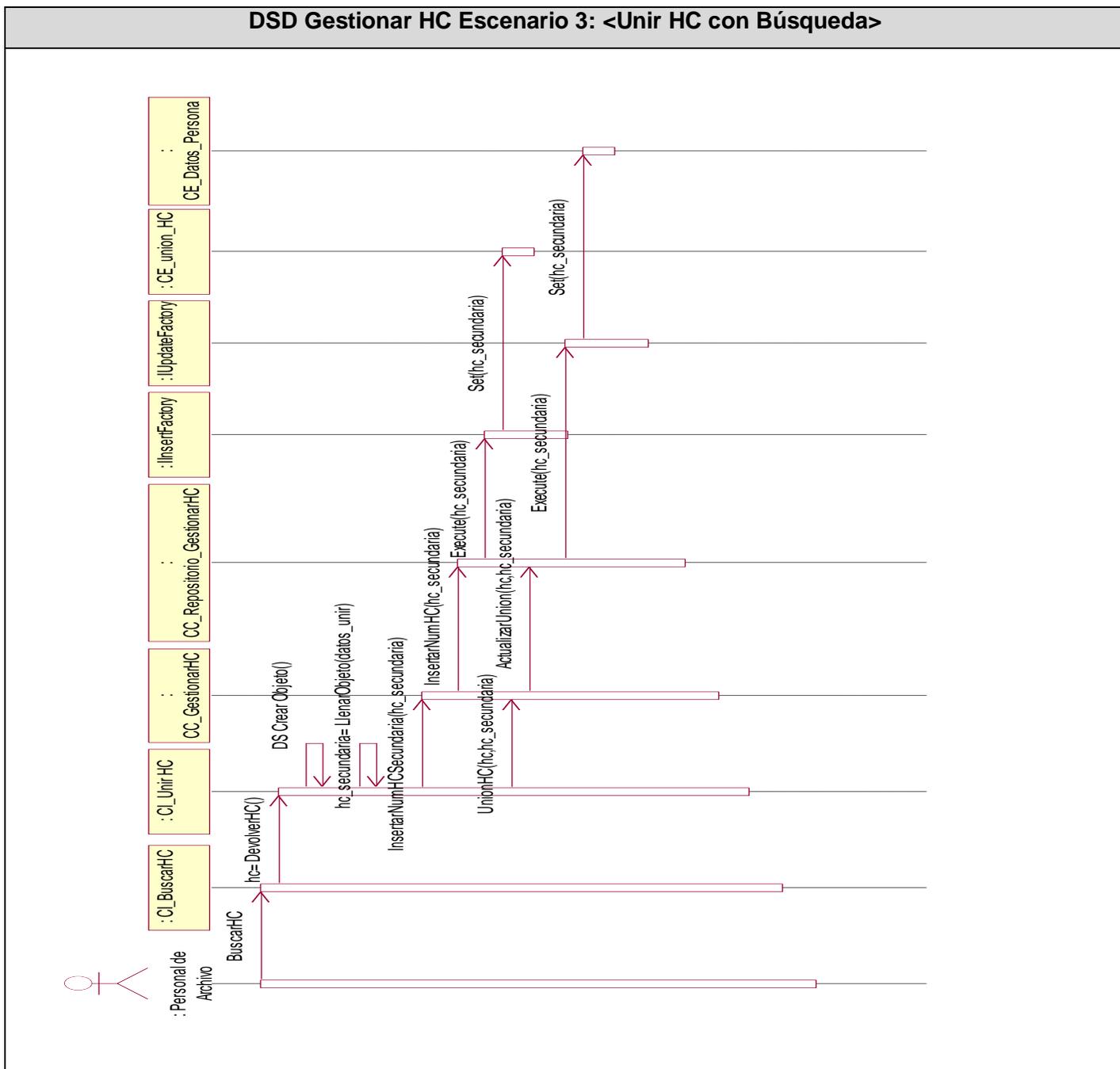


Figura 32 DSD Escenario 3: Unir HC con Búsqueda

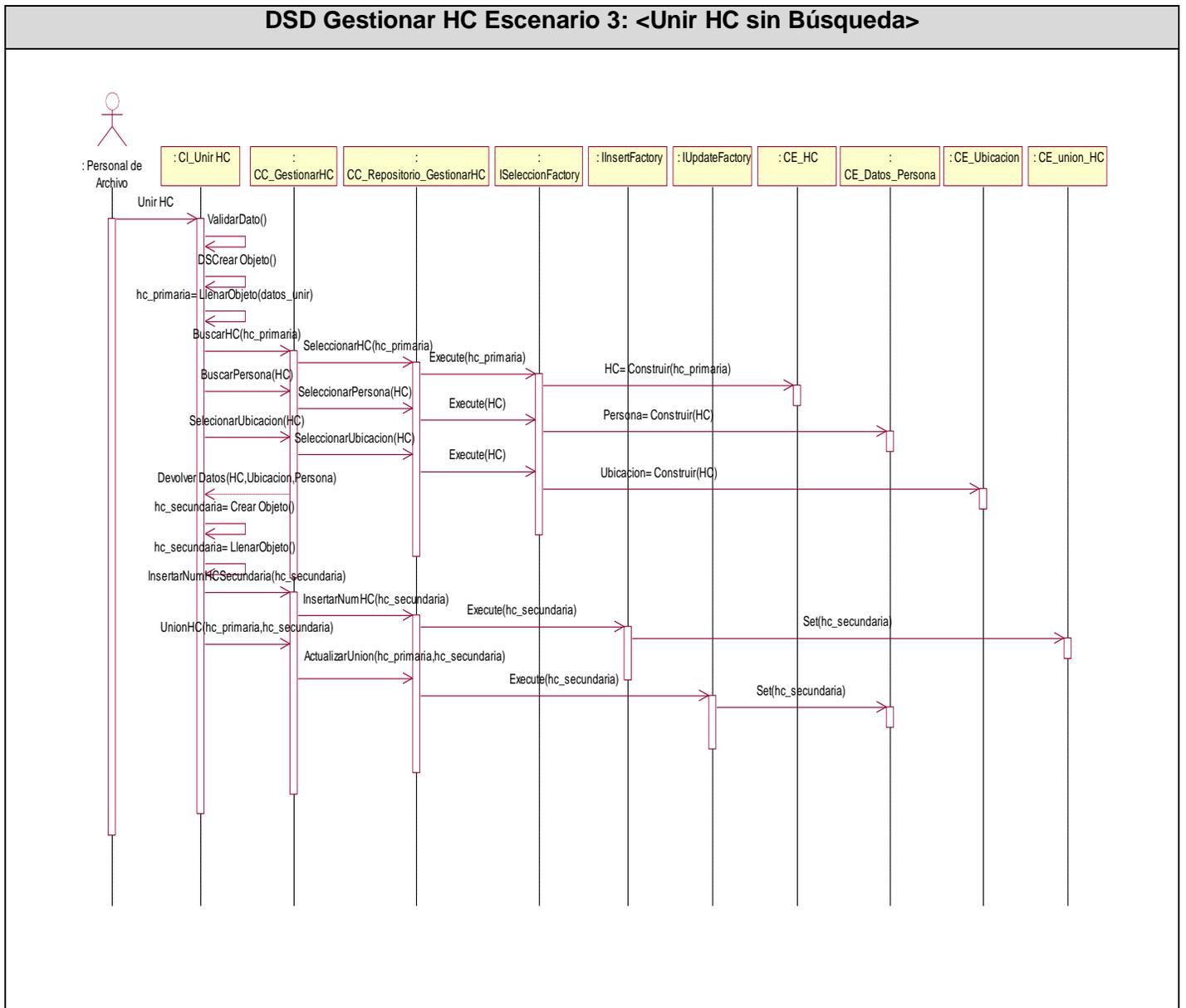


Figura 33 DSD Escenario 3: Unir HC sin Búsqueda.

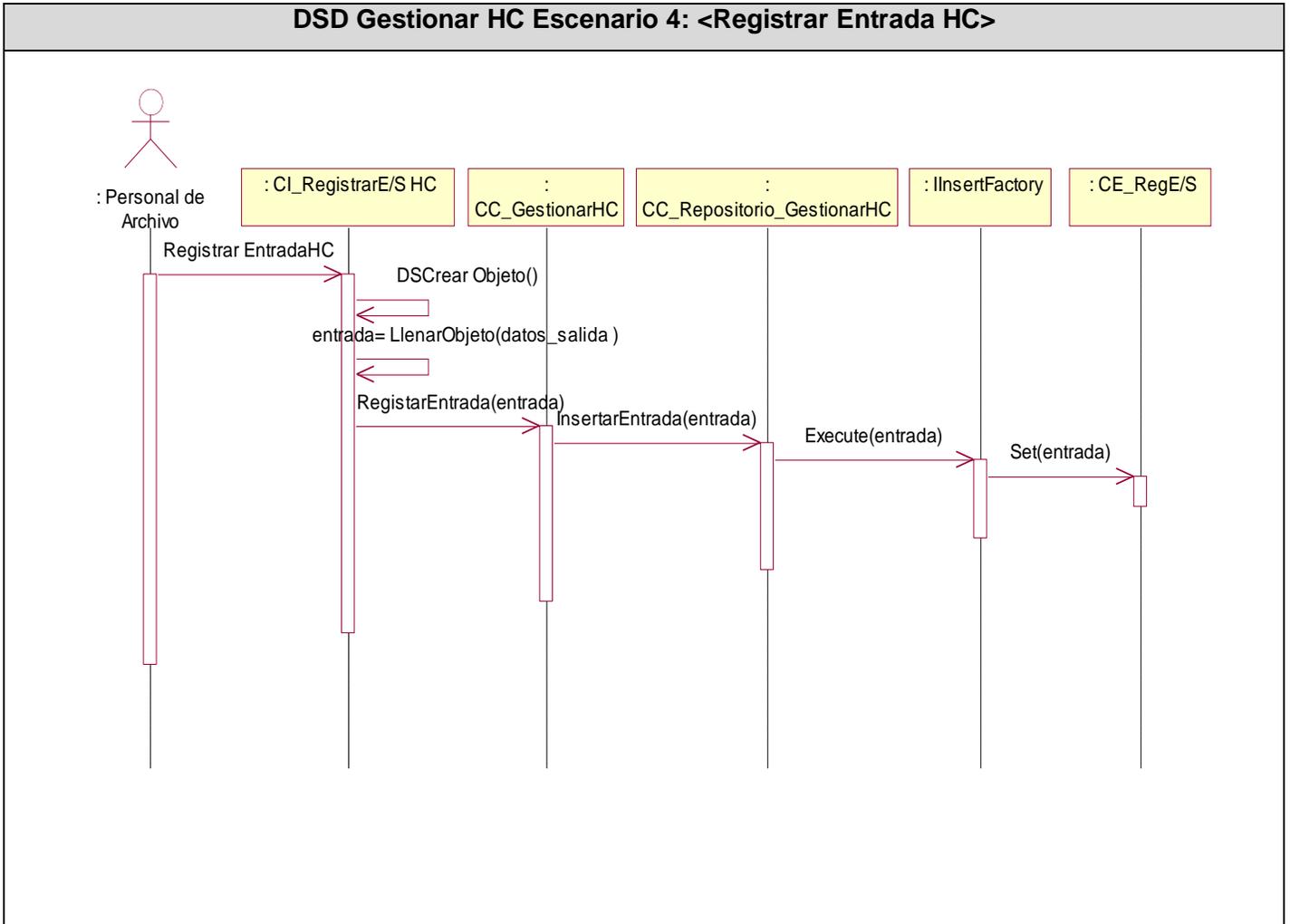


Figura 34 DSD Escenario 4: Registrar Entrada HC

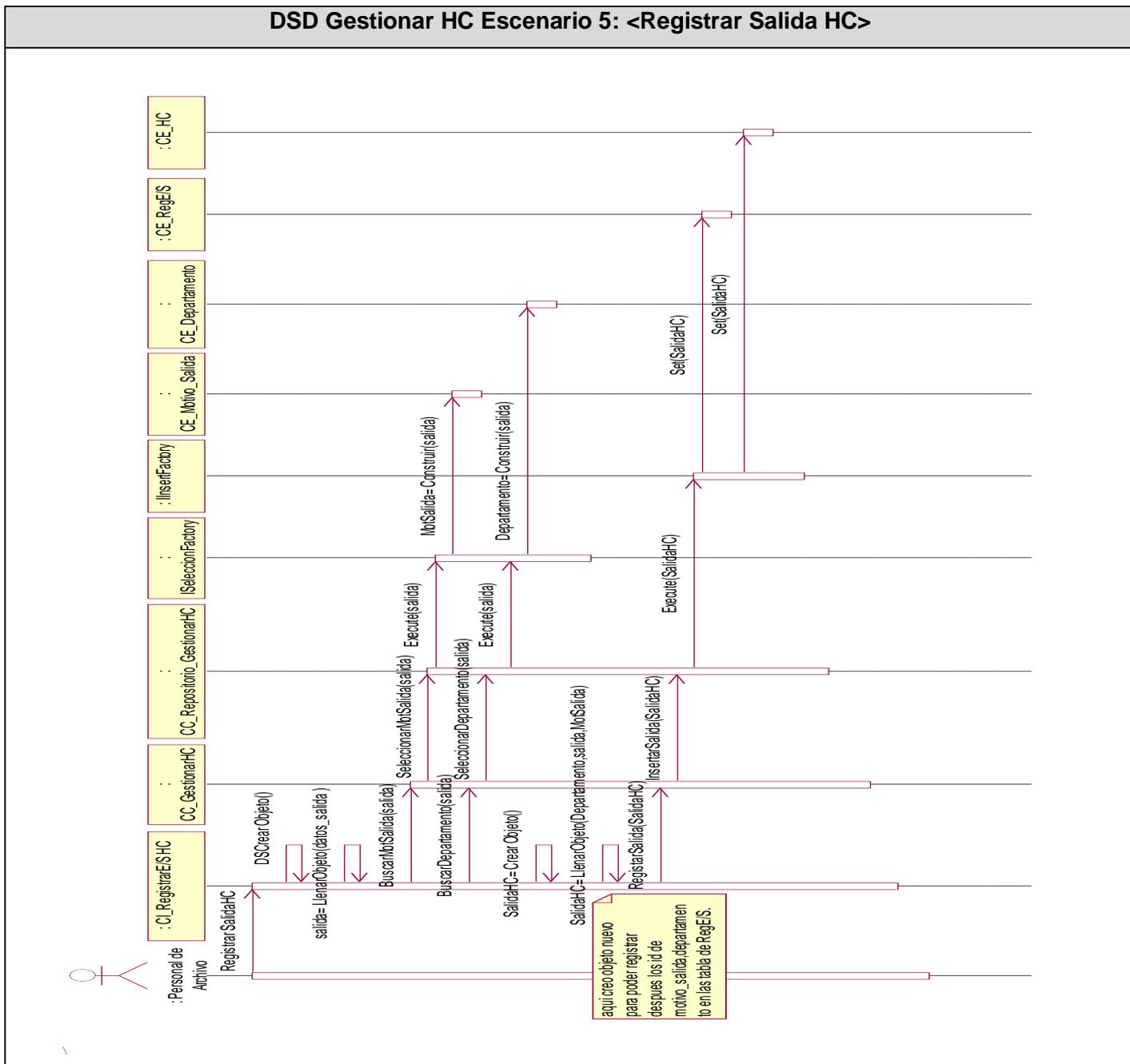


Figura 35 DSD Escenario 4: Registrar Salida HC

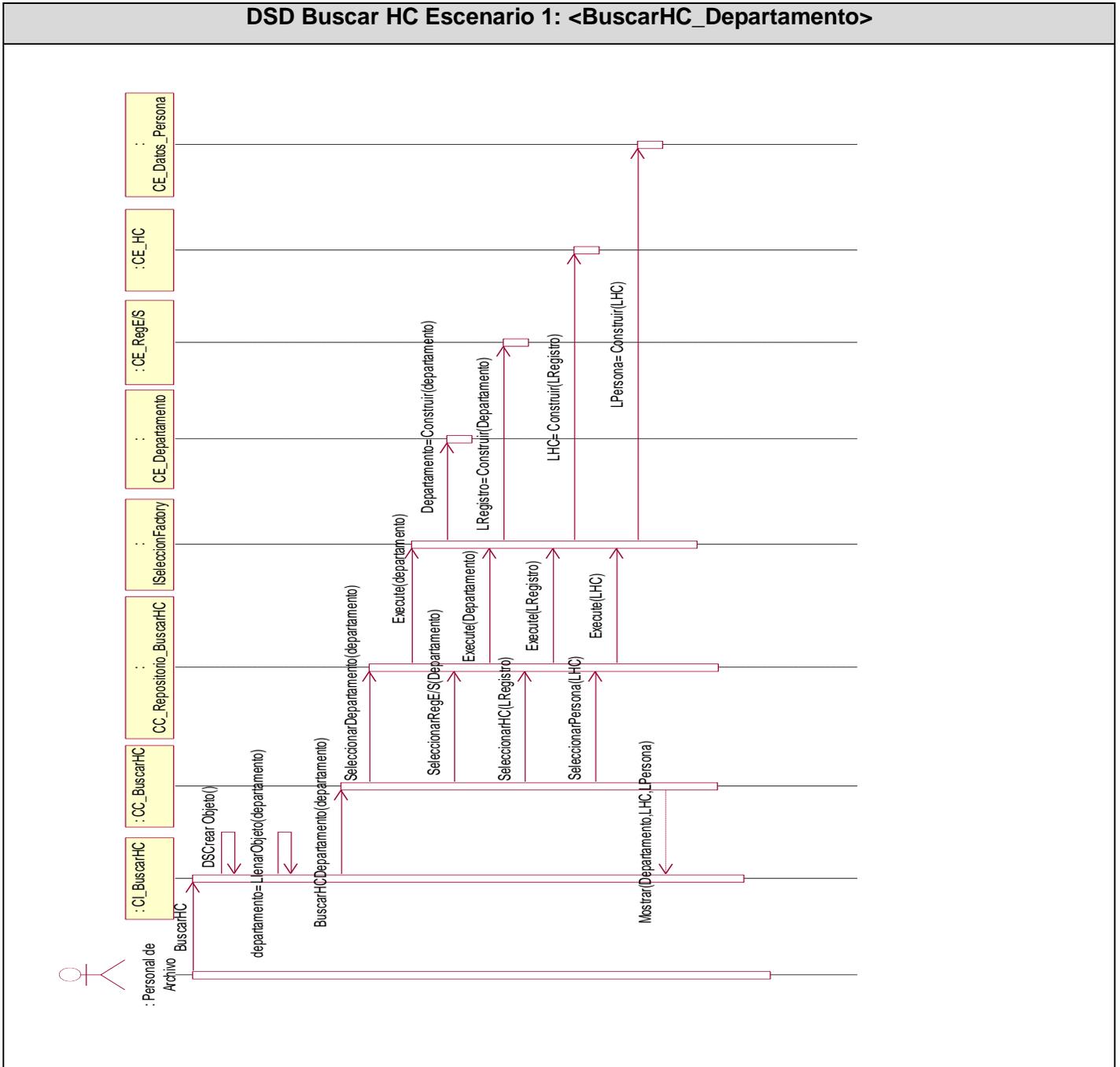


Figura 36 DSD Buscar HC Escenario : BuscarHC_Departamento

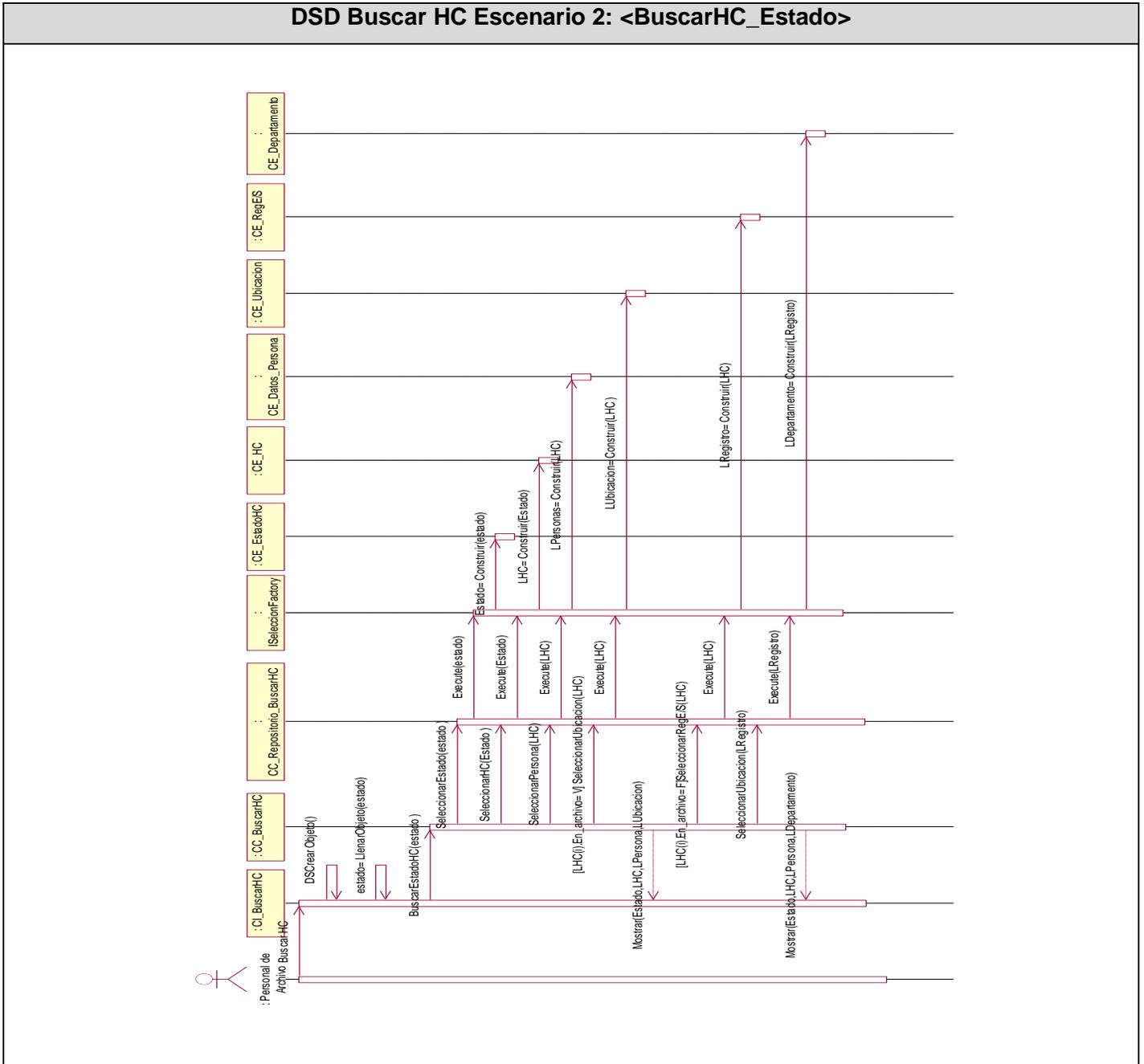


Figura 37 DSD Buscar HC Escenario 2: BuscarHC_Estado

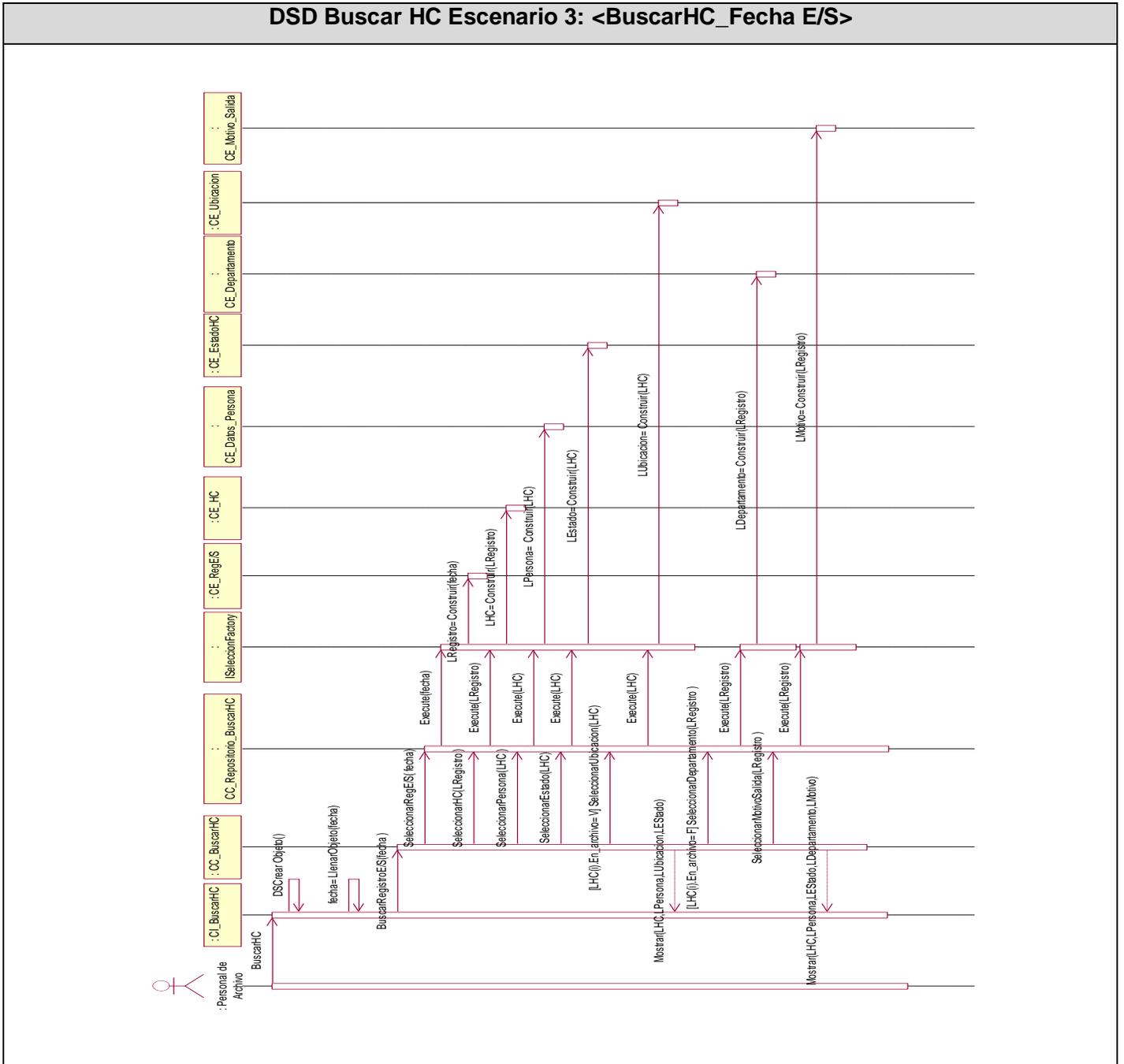


Figura 38 DSD BuscarHC Escenario 3: BuscarHC_FechaE/S

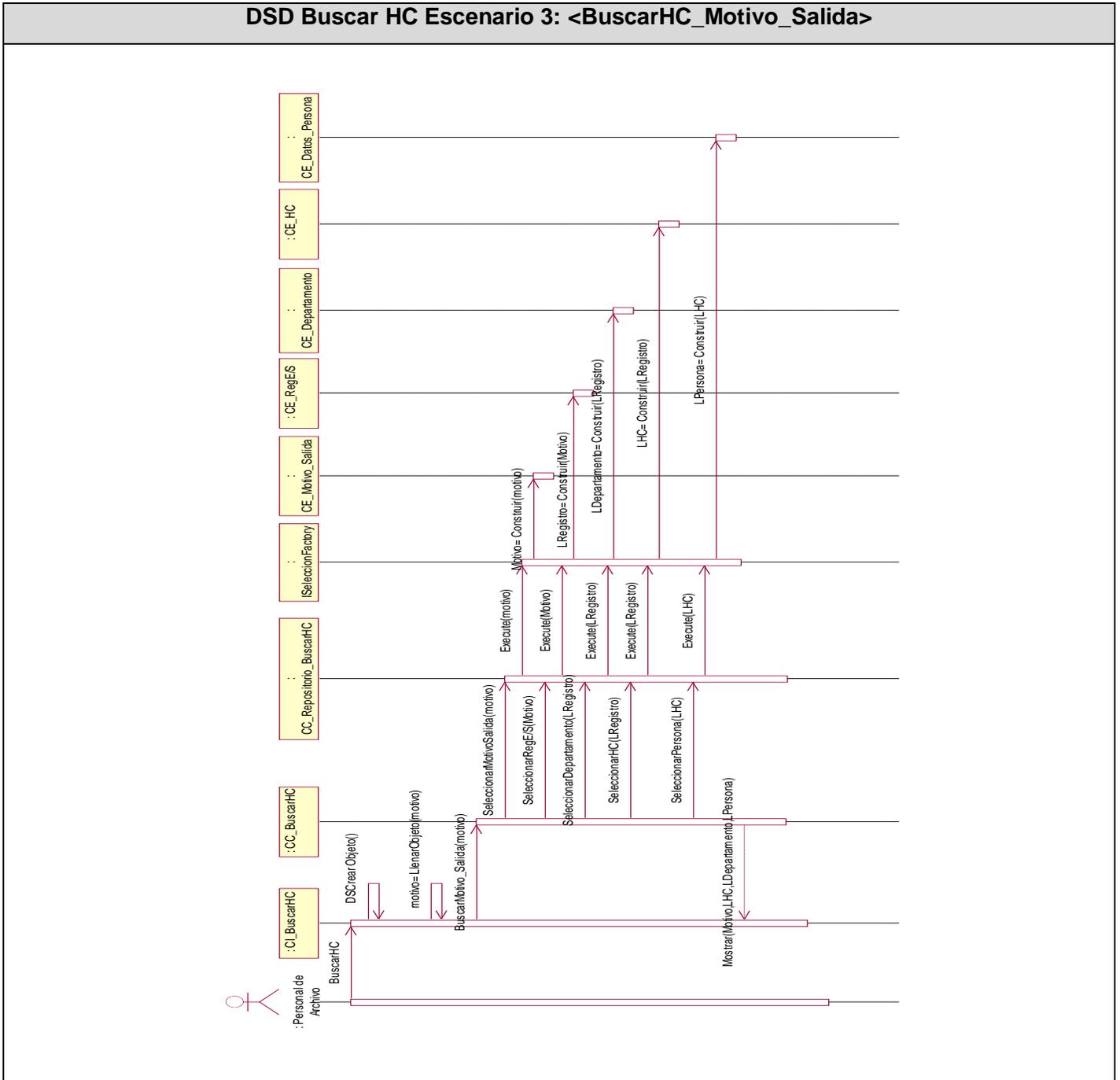


Figura 39 DSD BuscarHC Escenario 3: BuscarHC_Motivo_Salida

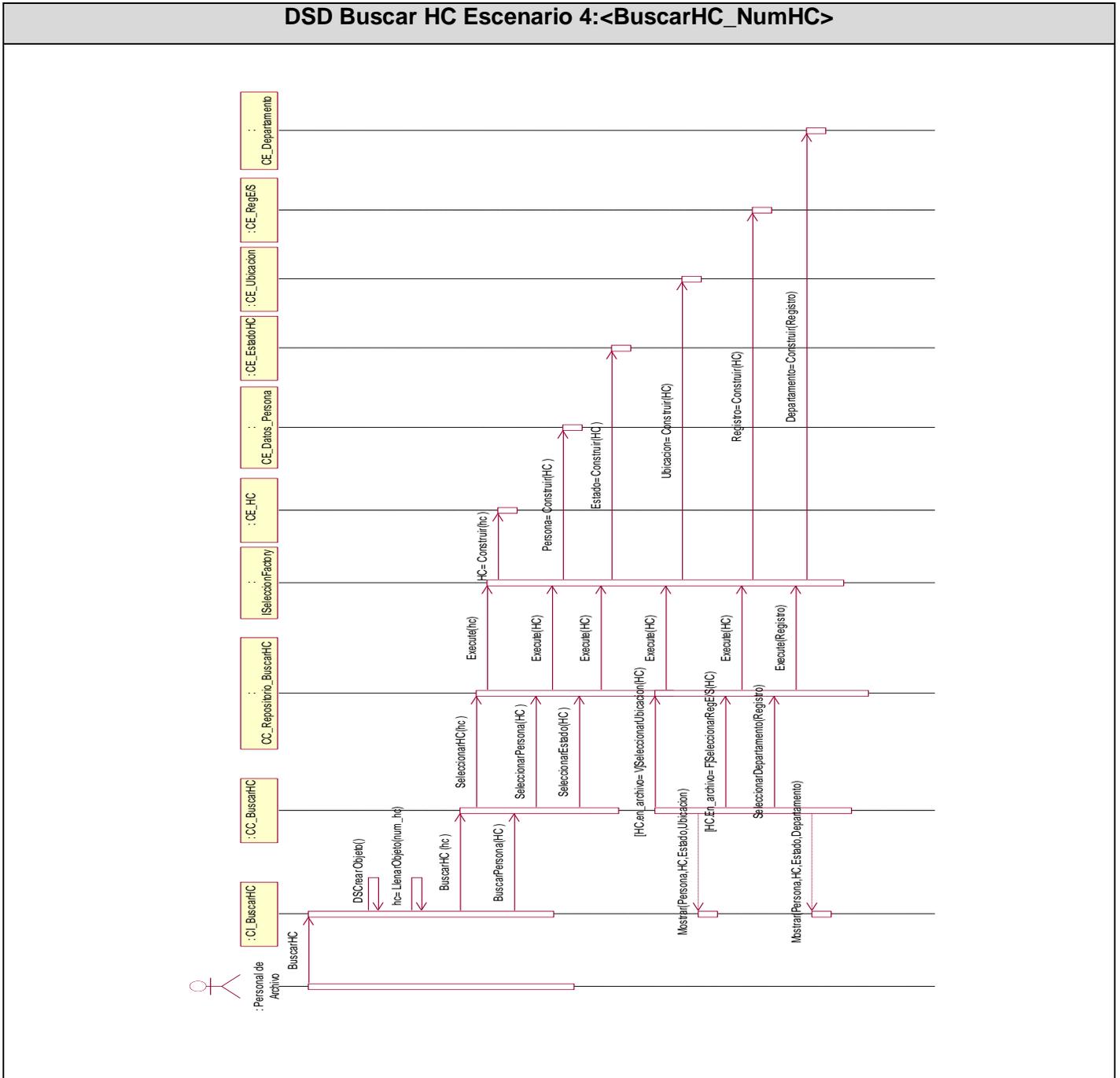


Figura 40 DSD Buscar HC Escenario 4: BuscarHC_NumHC

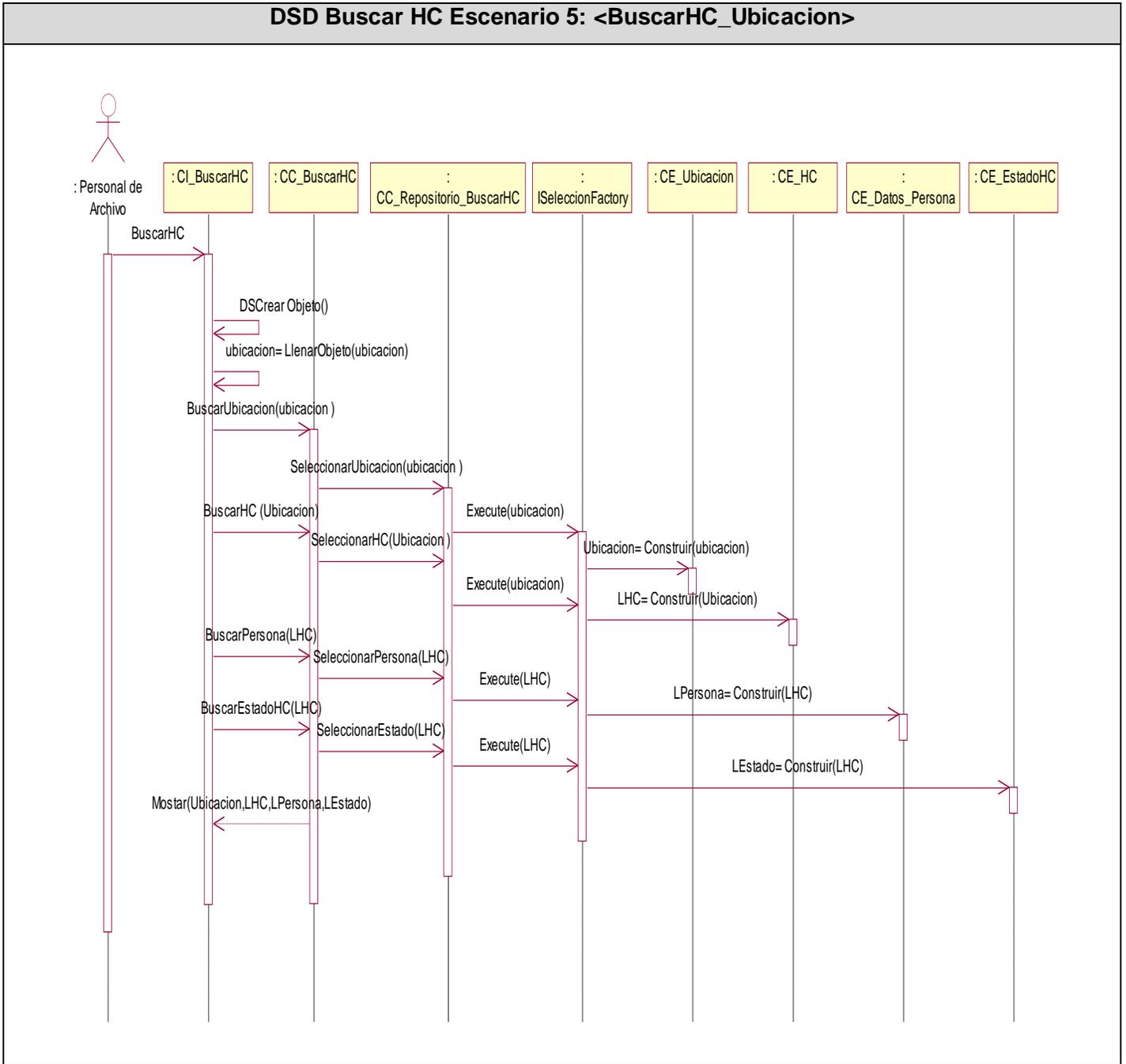


Figura 41 DSD Buscar HC Escenario 5: BuscarHC_Ubicacion

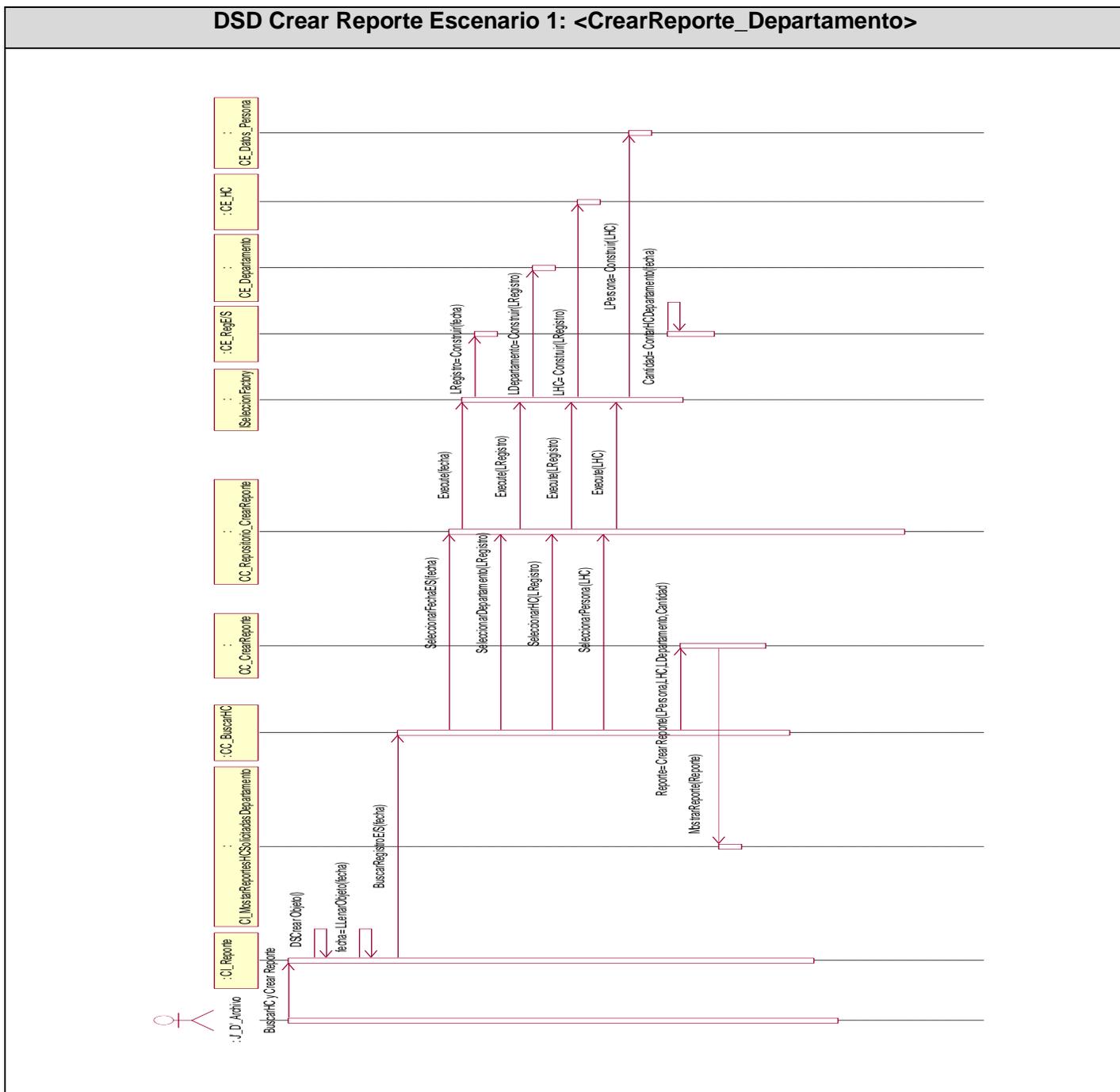


Figura 42 DSD Crear Reporte Escenario 1: CrearReporte_Departamento

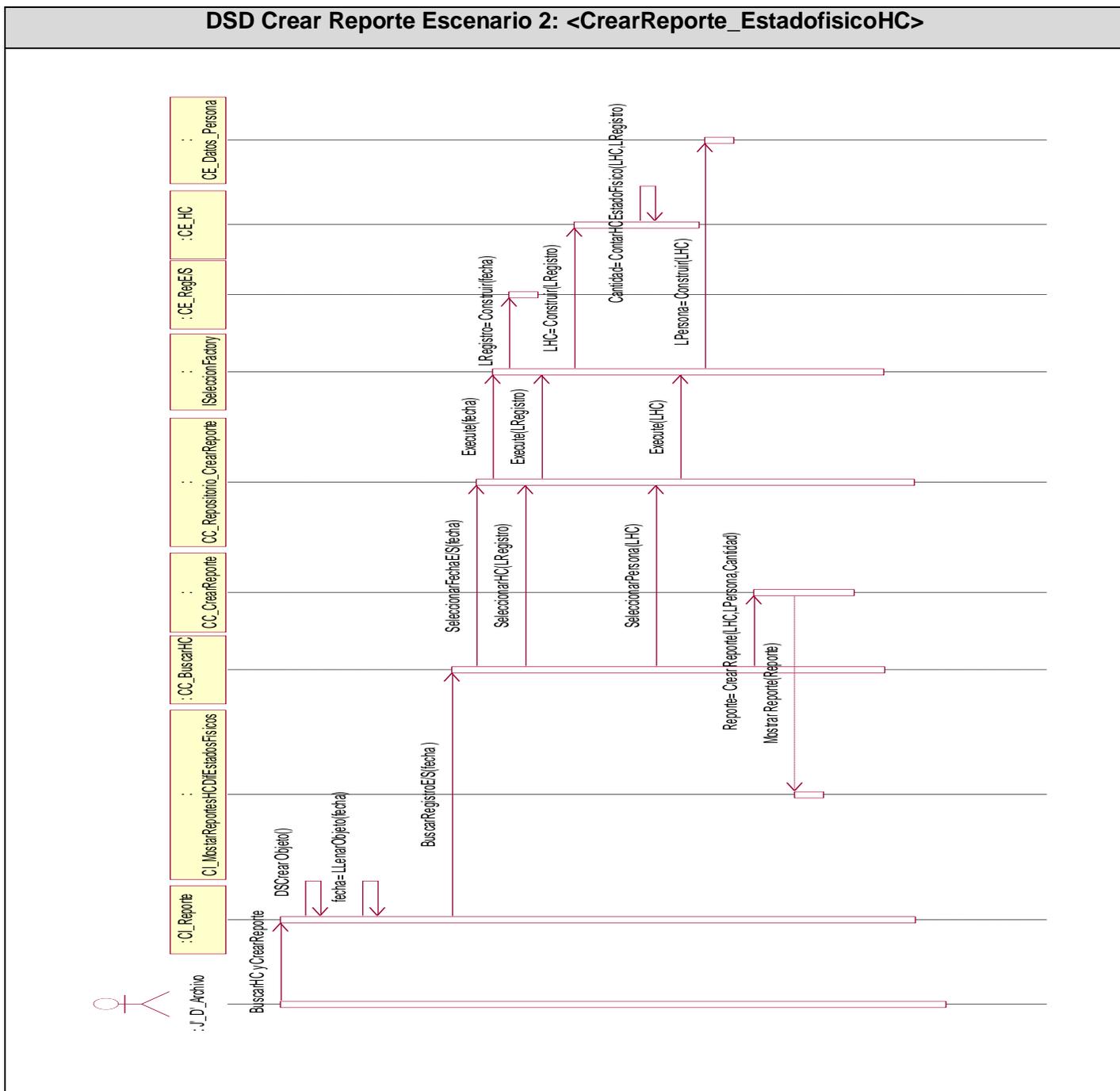


Figura 43 DSD Crear Reporte Escenario 2: CrearReporte_EstadoFisicoHC

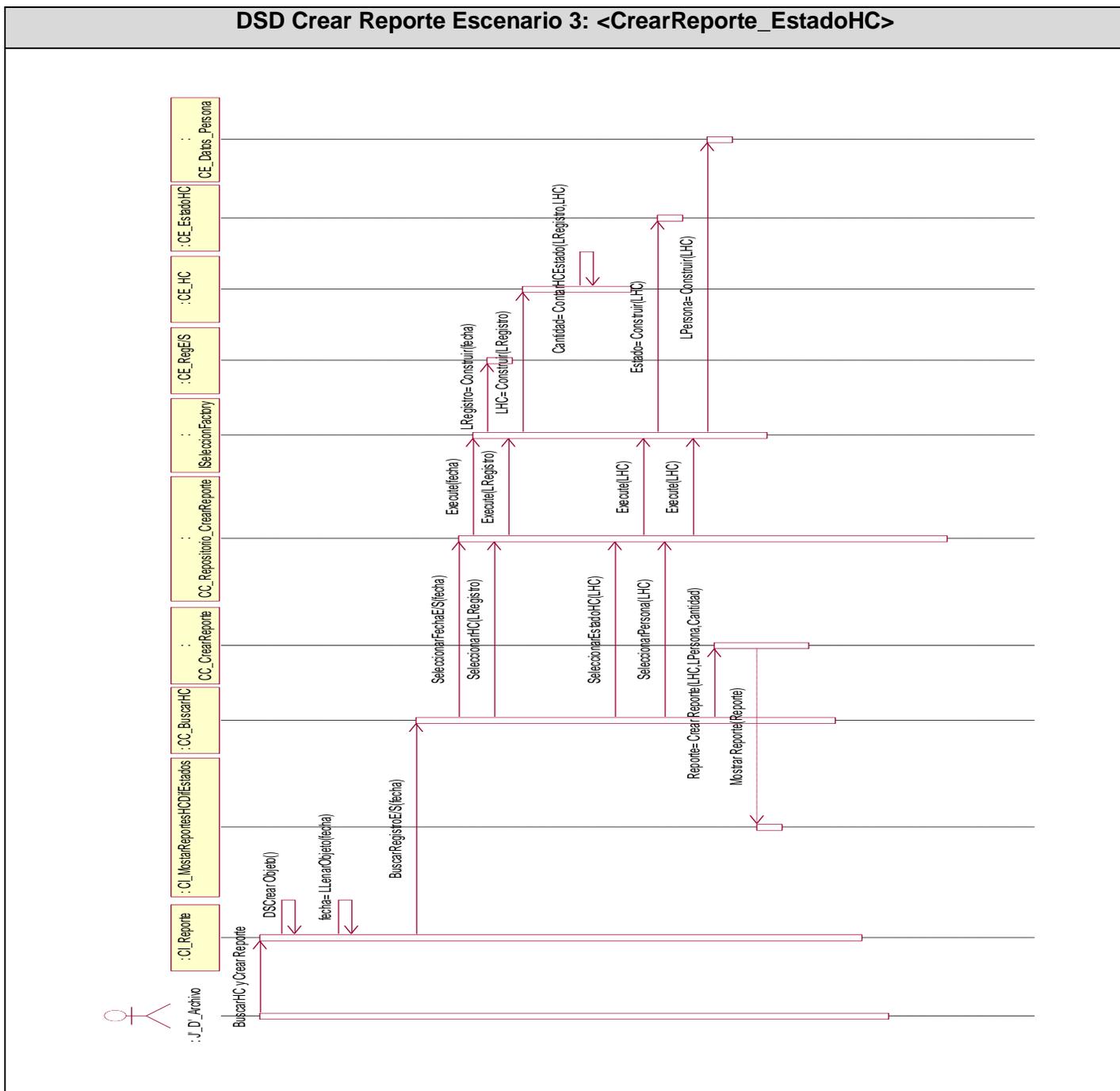


Figura 44 DSD Crear Reporte Escenario 3: CrearReporte_EstadoHC

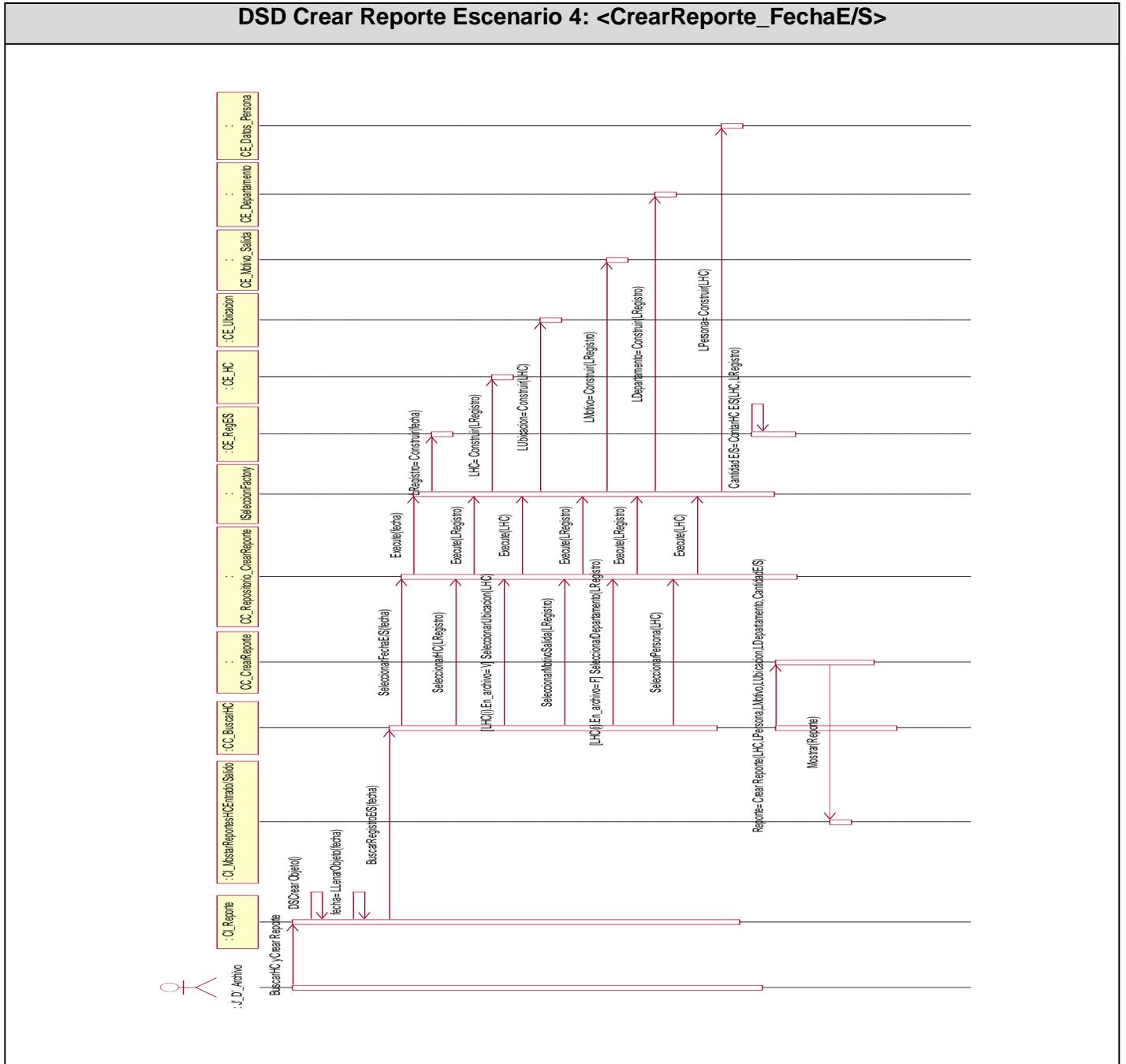


Figura 45 DSD Crear Reporte Escenario 4: CrearReporte_FechaE/S

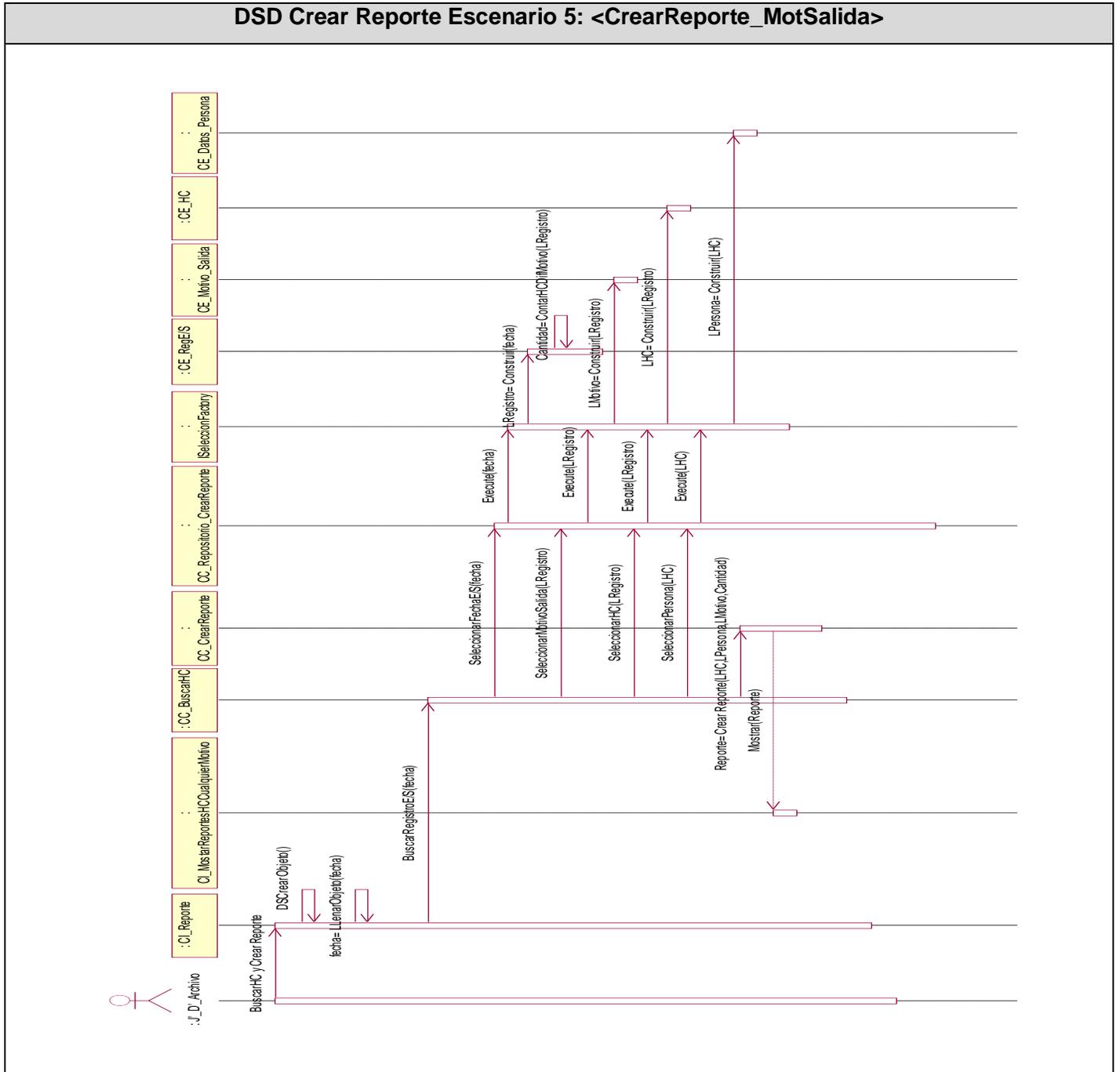


Figura 46 DSD Crear Reporte Escenario 5: CrearReporte_Motivo_salida

Anexo IV Interfaz de la Aplicación



Figura 47 Interfaz Autenticar Usuario

A: Autenticación mediante el nombre de usuario y contraseña.



Figura 48 Interfaz Menú.

B: El usuario puede acceder los campos del Menú (Clasificar HC, Insertar HC, Buscar HC, etc.) según su responsabilidad.

C: Registro de Entrada/ Salida de la HC

D: Acciones (de insertar un nuevo usuario, modificar rol del usuario así como eliminar un usuario) permitidas al usuario de acuerdo a su responsabilidad.

The screenshot shows a software window titled 'frm_Buscar'. At the top, there are three tabs labeled 'Pagina 1', 'Pagina 2', and 'Pagina 3'. Below the tabs, there are several input fields for search criteria: 'Nombre:', 'Apellido1:', 'Apellido2:', and 'CI:'. To the right, there are fields for 'HC:', 'Sexo:' (with a dropdown arrow), and 'Nacimiento:' (with a date '22/05/2007' and a dropdown arrow). Below these fields are two buttons: 'Aceptar' and 'Cancelar'. A large grey rectangular area is visible at the bottom of the window. Callout boxes with letters F, G, and E point to specific elements: F points to the 'Pagina 2' tab, G points to the 'Pagina 3' tab, and E points to the 'Sexo:' dropdown menu.

Figura 49 Interfaz Buscar HC por datos personales

E: Búsqueda de las HC por los datos del paciente.

F: Búsqueda de las HC por la fecha de entrada/salida de la HC del archivo.

G: Búsqueda de la HC por los criterios de Motivo de salida de la HC del archivo o por estado.

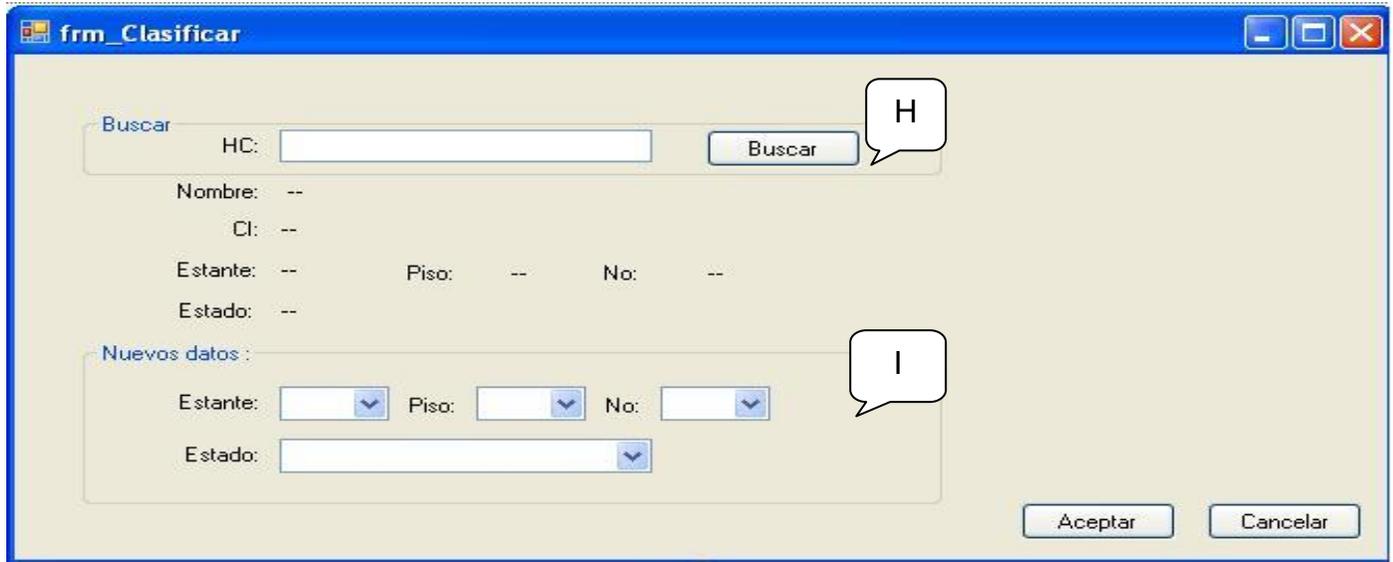


Figura 50 Interfaz Clasificar HC

H: Se hace una búsqueda a la HC que se va actualizar su estado.

I: Nuevos datos asignados a la ubicación de la HC.

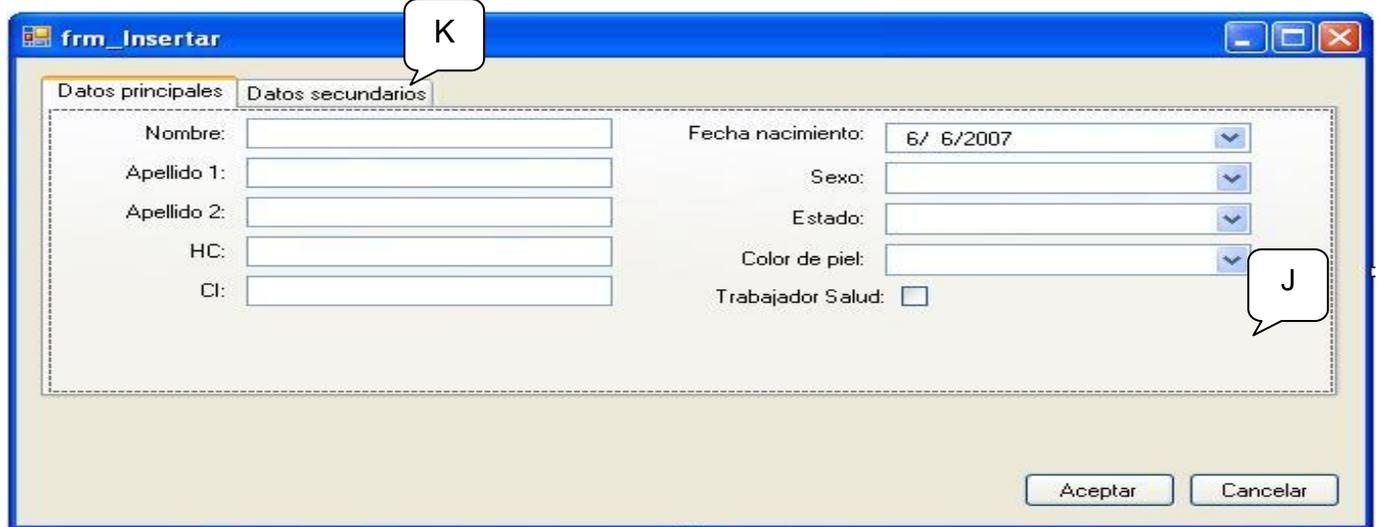


Figura 51 Interfaz Insertar HC con sus datos personales

J: Se inserta los datos personales del paciente.

K: Se Inserta datos del paciente además de otorgarle una ubicación a la HC dentro del archivo.

frm_Unir

HC Primaria

Buscar:

HC: Buscar L

Nombre:

Edad:

Direccion:

Estante: Piso: No:

Cantidad Uniones:

HC Secundaria

HC:

Direccion: Nueva direccion M

Aceptar Cancelar

Figura 52 Interfaz Unir HC

L: Se busca la HC del paciente a quien se le va anexar la HC provisional que se le ha creado y se muestra datos de esta así como su ubicación dentro del archivo y las cantidades de uniones que se le han realizado a la HC.

M: Se actualizan campos en la HC como su dirección particular en caso de algún cambio.

frm_Registrar_Entrada

HC: []

Fecha de entrada: 6/ 6/2007

Condición papel: []

Recibe de: []

Localización: []

Observaciones: []

N

Aceptar Cancelar

Figura 53 Interfaz Registrar entrada de la HC

N: Datos para registrar datos de la entrada de la HC del archivo.

frm_Control_ent_sal

HC: []

Hacia especialidad: []

Prestado a: []

Fecha de salida: 6/ 6/2007

Fecha posible retorno: 6/ 6/2007

Condicion papel: []

Motivo de salida: []

Observaciones: []

Ñ

Aceptar Cancelar

Figura 54 Interfaz Registrar Salida de la HC

Ñ: Datos insertados para registrar la salida de la HC del archivo.

Form2

Intervalo de fechas

desde: 22/05/2007 hasta: 22/05/2007

Por motivo de salida Por solicitud de departamento Por estado físico Por diferentes estados

	#HC	Nombre y Apellidos	#CI	Fecha de salida	Motivo de salida	Localización externa
*						

Aceptar Cancelar

Figura 55 Interfaz Crear Reporte

O: Datos para crear el reporte estadístico del departamento de archivo.

Glosario

DCA – Diagrama de clases de Análisis.
DCD – Diagrama de clases de Diseño.
DSD – Diagrama de Secuencia del Diseño.