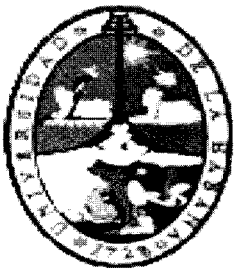
UNIVERSIDAD DE LA HABANA
DEPARTAMENTO DE COMPUTACIÓN



Trabajo de Diploma

Título

Sistema para la informatización de los servicios médicos de la UCI
Módulos de Historia Clínica y Laboratorio.

Autor

Yoemny González Almaguer
Amilkar Yudier Puris Cáceres

Tutor

Ing. Godofredo Ramón Garay Álvarez

Curso 2003-2004

Resumen

Este trabajo muestra los temas fundamentales del estudio e implementación de los módulos de Historia Clínica Personal y Laboratorio Clínico como parte importante del software de servicios médicos de la UCI que desarrollamos. Por la necesidad del proceso de informatizar estos módulos y el costo monetario que la obtención de estos servicios le proporciona a la salud pública cubana nos vimos en la tarea de realizar este trabajo. La automatización de la información médica es una meta que representa enormes ventajas para los profesionales de la salud, los pacientes, y el Estado.

La transformación de las historias clínicas personales de los pacientes de su formato adicional, en copia de papel, a uno electrónico, constituye un eslabón fundamental dentro de esta tarea; sin embargo, la implementación de dichas historias es aún una quimera. Este trabajo propone una vía para lograr un sistema de historias clínicas personales electrónicas en nuestro país. Se brindan las distintas etapas para alcanzar su aplicación de una forma gradual, económica y eficiente, dentro del Sistema Nacional de Salud, lo que le permitiría colocarse en un nivel intermedio de automatización de las historias clínicas personales, con todas las facilidades que implicaría.

Un elemento de trascendental importancia al aplicar el método clínico, en la etapa de postración del diagnóstico presuntivo por parte de cualquier especialista en el campo de la salud, lo constituye sin lugar a dudas la valoración de los resultados de los exámenes complementarios que se les practican a los pacientes. Para la solución de esta problemática realizamos la automatización de este trabajo, automatizando totalmente el trabajo en cualquier laboratorio clínico de nuestro país.

Índice

Introducción

La informática en Cuba cada día gana en desarrollo, se está llevando a todos los niveles de nuestra sociedad y economía. Con este avance se consigue que el desarrollo del trabajo sea superior tanto cuantitativamente como cualitativamente y además le da un valor agregado a cada servicio que se presta. Para nuestro país la asistencia médica presenta máxima prioridad, de ahí la necesidad de la informática médica para lograr un máximo de excelencia en los servicios médicos que se le brinda a la población.

En la actualidad el aporte de la tecnología es fundamental en todas las áreas, pero imprescindible en lo que respecta a la medicina. Consideramos que debe existir una interrelación entre medicina y tecnología. Las ciencias de la salud, y la medicina en particular, son uno de los campos del saber más evolucionados y beneficiados por el uso de las modernas tecnologías de la información, al tiempo que registran un crecimiento exponencial tanto en el número de usuarios, como en el de instituciones y ubicaciones que se han incorporado a la búsqueda de diferentes medios que permitan un mejor nivel de vida.

La informática médica se sitúa en la intersección entre la informática y las diferentes disciplinas en la medicina y los cuidados de salud [1]. Esta nueva ciencia resulta imprescindible para la adquisición no sólo de conocimientos, sino de herramientas que le posibilitan al profesional de la salud a acceder a información, como también a la utilización y creación de software propios del medio en que se desarrollan.

En nuestro caso creamos un software para informatizar los servicios médicos de la Universidad de las Ciencias Informáticas (UCI), dentro de el se encuentran el módulo de Historia Clínica Personal y el módulo Laboratorio Clínico, para la realización de dicha actividad realizamos el análisis, diseño e implementación de ambos modulo.

Una de las problemáticas cruciales y a la vez más interesantes de la joven comunidad de Informática Médica (IM) es el diseño, confección,

implementación y desarrollo de Historias Clínicas Electrónicas (HCE) [2]y Exámenes Complementarios Electrónicos (ECE). De una forma muy rápida podríamos decir que la HCE y un ECE no son más que la historia clínica y los exámenes complementarios convencionales llevados a formato electrónico, con todas las ventajas que este hecho por sí solo implican. Pero en realidad no es tan sencillo como parece; las HCE y los ECE han transitado por diferentes etapas, algunas de las cuales aún en estos momentos de gran desarrollo tecnológico son tan sólo teóricas [3]. En nuestro caso las HCE y los ECE han sido desarrollados teniendo en cuenta las necesidades reales para llevar a cabo la informatización de los servicios médicos de la UCI.

Por tanto, constituye el Objetivo general de este trabajo realizar el análisis, diseño e implementación de una aplicación basada en web para manejar las Historias Clínicas Personales de los pacientes de la UCI y los exámenes complementarios que se realizan en el laboratorio clínico del policlínico de la UCI.

A partir de un análisis del objetivo general se derivan los siguientes objetivos específicos para la aplicación:

- *Facilitar a médicos y especialistas datos referentes a la historia clínica de un paciente de la UCI.*
- *Almacenar las historias clínicas personales de los pacientes de la UCI.*
- *Proporcionar facilidades para la indicación y realización de un examen complementario.*
- *Obtener reportes sobre de los exámenes complementarios realizados.*
- *Realizar breve discusión teórica sobre evaluación del desempeño en aplicaciones web.*

El presente documento se estructura en cuatro capítulos y varios anexos, que incluye todo lo relacionado con el trabajo investigativo realizado, así como el análisis, diseño e implementación de la herramienta que se propone.

Capítulo 1 Fundamentación Teórica

Recoge el análisis de la información existente acerca del tema a tratar y las tendencias actuales que existen en el mundo. También incluye como aspectos de actualidad una descripción del lenguaje de programación a utilizar para la implementación así como del lenguaje de modelación usado.

Capítulo 2 Discusión Teórica

Recoge algunas definiciones para la evaluación del desempeño en aplicaciones web, estrategias a seguir para que su aplicación tenga un buen desempeño y un ejemplo práctico de una aplicación que tiene un excelente desempeño.

Capítulo 3 Características de los subsistemas Módulos de Historia Clínica y Laboratorio

Muestra el problema y la situación problémica, el objeto de automatización, información que se maneja, propuesta del sistema, modelo de negocio, especificación de los requisitos de software y la definición de los casos de uso.

Capítulo 4 Análisis y diseño de los subsistemas Módulos de Historia Clínica y Laboratorio

Muestra la definición del modelo de análisis, diagramas de iteración, diagramas de clases, diseño de la base de datos y sistema de seguridad.

Capítulo 1

Fundamentación Teórica

Introducción

Para la realización de este proyecto es necesario el estudio de un conjunto de tecnologías para de diseño así como para la implementación del código y la representación y almacenamiento de los datos de los lenguajes.

1.1. Software para la Salud. Tendencias actuales

En la actualidad existen una gama de producciones diferenciadas de software para la salud, encaminadas a resolver disímiles problemáticas dentro de esta rama. A nivel internacional están muy desarrollados las aplicaciones referentes a brindar consultas para la utilización de nutrientes en las dietas, estas mayormente en países capitalistas, otras con el objetivo de lograr en los galenos el autoaprendizaje en Errores Innatos del Metabolismo, que permiten a médicos especialistas detectar enfermedades congénitas desconocidas, con el objetivo primordial de alertar sobre la importancia del conocimiento de los Errores Innatos del Metabolismo (EIM), especialmente entre los trabajadores del sector de la salud (médicos, bacteriólogos, nutricionistas, psicólogos). Existen además software para la gestión y control de historias clínicas de los pacientes y software para la evolución de los signos vitales, dado un conjunto de indicadores devenidos de exámenes complementarios previamente aplicados. De estos se conocen varias tendencias como las vinculadas a plataformas Web para aplicaciones que funcionen a nivel de red y otras de forma interactiva que son aplicaciones locales de multimedia mas bien en la parte de la enseñanza y la búsqueda de resultados, como también aplicaciones con uso de bases de datos para la gestión y almacenamiento de datos acerca de pacientes y su evolución y comportamiento de sus signos vitales, aunque predominan las que funcionan a través de la red pues permiten la discusión entre los clientes(galenos) para la toma de decisiones y para el intercambio de experiencias.

Cuba, a pesar de la difícil situación económica que presenta se ha mantenido actualizada en la medida de las condiciones y su desarrollo es reconocido por los especialistas de nivel internacional que tienen oportunidad de conocer nuestro trabajo. Cuba es miembro fundador de la Asociación Mundial de Informática Médica (IMIA), de la cual existe una filial en América Latina (IMIALAT); en nuestro país se han desarrollado tres Congresos Internacionales de Informática Médica

No está nuestro país a la saga en este tipo de producciones de software vinculadas a resolver problemas dentro de la rama de la salud muestra de esto esta en el conjunto de aplicaciones controladas por **CEDISAP** (Centro de Desarrollo Informático para la Salud Pública) creada en 1987, tiene entre sus funciones rectoras el diseño y la implementación de la política y estrategias de desarrollo de la informática en el Sector de la Salud.

Según se refiere por García [2000], un análisis detallado de lo que sucede en la práctica es realizado por Pressman [1998] el que plantea que en el proceso de desarrollo del software ocurre que:

no se utilizan eficazmente las metodologías modernas de desarrollo del software ni las herramientas de Ingeniería del Software asistida por computadora (CASE) que son más importantes que el hardware más moderno para conseguir una buena calidad y productividad,

- No existe una adecuada descripción formal y detallada del ámbito de la información, funciones, rendimiento, interfaces, ligaduras de diseño y criterios de validación por una mala comunicación entre clientes y analistas,
- No se recogen datos sobre el proceso de desarrollo del software,
- No se garantiza la calidad desde el inicio del proyecto, ni se aplica la revisión técnica formal que es un filtro de calidad muy efectivo para encontrar defectos en el software.

Criterios expuestos por varios autores Gibbs [1994], Paulk [1995], Tinnirello [1998] y Álvarez [1999], reconocen en común las deficiencias de que:

los proyectos se entregan a los clientes excesivamente tarde y están sobrepresupuestados,

- los beneficios de los mejores métodos e instrumentos no se pueden obtener en el medio indisciplinado y caótico de desarrollo,
- la calidad y la productividad no sólo se logran aplicando nuevas metodologías y tecnologías para desarrollar y mantener software,
- los procesos de software son improvisados,
- la planificación en tiempo y recursos no se cumple,
- no hay bases objetivas para enjuiciar la calidad del producto y resolver problemas del software,
- las revisiones y pruebas son eliminadas o disminuidas cuando el proyecto se atrasa,
- no se aplican modelos de evaluación de la calidad de los procesos de software.

Según refiere García [2000a], por ser en este nivel primario que se produce el primer contacto de la población con el Sistema Nacional de Salud, y donde deben resolverse más del 80% de los problemas, se hace necesario viabilizar la labor de registro, control y emisión de reportes a diversas instancias con el propósito de beneficiar la calidad de la labor como facultativo así como de la actualización de los informes

Después de todos estos análisis se han logrado en nuestro país herramientas en esta esfera de gran utilidad todas planificadas y diseñadas de la misma forma.

Casos puntuales son APUS con la implementación de módulos para Población, Consultas Medicas, Diagnostico y vigilancia de la situación de Salud, Consultas Medicas, Urgencias; Historia Clínica Personal y Familiar, Urgencias e ITS; **MEDISYS** de **CEDISAP** y **GALEN** de **CENTERSOFT** también con aplicaciones de este tipo.

1.1.1 Uso de Software libre.

La mayoría de las aplicaciones realizadas en nuestro país que pertenecen a los planes de informatización de la salud están regidas por una misma plataforma de ahí que usen software libre.

El término de software libre –lo contrario del software patentado- se aplica a las aplicaciones informáticas que están libremente disponibles bajo un acuerdo de licencia pública de manera que cualquiera puede adaptarlos y mejorarlos. Las quejas más comunes sobre los programas de software patentados de Microsoft son sus costes relativamente elevados y que, cuando fallan, sólo Microsoft puede repararlos. Cada vez más, grandes organizaciones de toda clase optan por soluciones libres para sus necesidades informáticas porque los productos libres son generalmente más baratos, más fiables y más fáciles de reparar cuando fallan. El mercado de las computadoras personales no se está quedando muy atrás.

``Software Libre'' se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto.
- La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Un programa es software libre si los usuarios tienen todas estas libertades. Así pues, deberías tener la libertad de distribuir copias, sea con o sin modificaciones, sea gratis o cobrando una cantidad por la distribución, a cualquiera y a cualquier lugar. El ser libre de hacer esto significa (entre otras cosas) que no tienes que pedir o pagar permisos.

También deberías tener la libertad de hacer modificaciones y utilizarlas de manera privada en tu trabajo u ocio, sin ni siquiera tener que anunciar que dichas modificaciones existen. Si publicas tus cambios, no tienes por qué avisar a nadie en particular, ni de ninguna manera en particular.

La libertad para usar un programa significa la libertad para cualquier persona u organización de usarlo en cualquier tipo de sistema informático, para cualquier clase de trabajo, y sin tener obligación de comunicárselo al desarrollador o a alguna otra entidad específica.

Con este trabajo se pretende además enmarcar la aplicación de los servicios médicos de la UCI dentro de la Red Medica Nacional por lo que se toma la decisión de realizar el modelo de diseño según RUP (Rational Unified Process), la implementación en J2EE, Eclipse como entorno de desarrollo integrado (IDE) y el almacenamiento de datos en MySQL.

1.2 J2EE

La utilización de Servidores de Aplicaciones y la ejecución de aplicaciones basadas en tecnología J2EE impactan en mayor o menor medida en distintas disciplinas dentro de una organización, y analizaremos el impacto sobre aquellas que consideramos algunas de las más relevantes:

- Modelado de Datos.
- Análisis de Requerimientos.
- Arquitectura y Diseño.
- Codificación.
- Testing.
- Entorno y Operación.
- Administración de Proyectos.
- Software Configuration Management.

1.2.1 Arquitectura y Diseño

Durante el proceso de diseño de aplicaciones J2EE, la cantidad de alternativas de solución a un mismo problema es extensa. La elección de la más adecuada

dependerá de varios factores, entre los que cabe destacar las necesidades de performance, la experiencia del grupo de desarrollo de aplicaciones, la disponibilidad de herramientas correctas, las posibilidades de escalabilidad y la magnitud y criticidad de la aplicación a desarrollar.

Es en este sentido que la arquitectura física gana relevancia y deberá acompañar en todo momento la toma de decisiones de diseño: la utilización o no de EJBs (Enterprise Java Beans), la delegación o no de cuestiones como persistencia, seguridad y transaccionabilidad en sus contenedores, la separación de la solución (lógica y eventualmente física) en tres capas, la componentización (packaging) de funcionalidad, entre otras.

1.2.2 Codificación

El estándar J2EE sugiere una arquitectura multicapas para las soluciones basadas en dicha tecnología.

En una arquitectura de estas características aparece la necesidad de contar con distintos roles de desarrollo. Las componentes correspondientes a la primer capa, el Contenedor de Clientes, serán las encargadas de definir las características de interface o "look and feel" de las aplicaciones. Si bien la gran mayoría de los clientes será navegadores Web, es cada vez más común encontrar dispositivos de acceso no tradicionales (teléfonos para acceso vía voz, celulares habilitados para Web, computadores de mano o clientes Java basados en interfaces gráficas de usuario). Será responsabilidad de los Desarrolladores de Estética de Presentación, la implementación de estas componentes. En este rol encontraremos a diseñadores gráficos, artistas Web y especialistas en interfaces.

1.2.3Administración de Proyectos

J2EE define los distintos roles de un grupo de desarrollo que utiliza para esta tecnología: Tool Provider, Enterprise Bean Developer, Deployer, Web Developer, Aplication Assembler, Deployer, etc.

1.3 Java

El lenguaje Java tiene una serie de características que lo hacen destacar. Es el fruto del desarrollo de la empresa Sun Microsystems, que necesitaba un lenguaje de programación para un proyecto interno. Al evaluar los existentes, el equipo de desarrollo decidió crear uno nuevo.

1.3.1 Clases y Objetos

Java está totalmente basado en clases y objetos. Todo en Java (aparte de algunos tipos primitivos) es un objeto. Contrariamente a lenguajes híbridos como C++ o el popular Visual Basic, en Java no se permite programar fuera de un objeto o clase. No hay módulos ni funciones globales.

1.3.2 Multiplataforma

Una característica todavía más distintiva de Java es su capacidad multiplataforma. Lenguajes como C o COBOL se han implementado en múltiples plataformas, pero siempre han necesitado recopilaciones o adaptaciones al pasar de una a otra. En cambio, Java desde el principio ha sido pensado para adaptarse a varios entornos. Esto lo consigue no sólo a nivel de código fuente, sino también a nivel de código compilado. El programa que escribimos se puede compilar en Windows o en Linux, y funciona. Pero, además, hasta el programa compilado puede ejecutarse sin más preparación, en distintas máquinas. Eso lo consigue porque Java se compila y ejecuta, no en un procesador o entorno en particular, sino en lo que se llama una "virtual machine", una máquina virtual. Nuestro programa Java podrá ejecutarse en cualquier sistema operativo que tenga una máquina virtual Java compatible.

1.3.3 Paquetes

Java no es solamente un lenguaje, es una tecnología. Al estar basado en clases y objetos, viene acompañado de un conjunto de éstos, que nos sirven como base para la programación de aplicaciones, de texto, gráficas o que se ejecuten en una página web como "applets". Estas clases se agrupan en paquetes. Gran parte del curso se ocupa en estudiar los distintos paquetes de clases que componen la librería Java estandard. Podemos nombrar los paquetes AWT (Abstract Window Toolkit) para el manejo de gráficos en

cualquier entorno, el paquete Swing con nuevos componentes gráficos, y el JDBC (Java Database Connectivity), para acceso a base de datos.

Estos son paquetes provistos por Sun, pero podemos construir nuestros propios paquetes, o comprar paquetes comerciales de terceras partes. Hay un gran mercado de clases y paquetes ya desarrollados.

1.3.4 Java en el browser

Al ser multiplataforma, Java fue adoptado por Netscape como la tecnología con la que se desarrollaron "applets", pequeñas aplicaciones que corren dentro de una página Web. Cada navegador implementa una máquina virtual Java, que permite ejecutar las applets en el computador del cliente. Esto ha sido implementado por los navegadores de Netscape y de Microsoft, y en el caso del primero, en varias plataformas. Esto ha permitido que las applets se puedan ejecutar tanto en Windows, como en Unix o Linux.

1.3.5 Java en el servidor

Si bien Java alcanzó popularidad en la Web, hoy la corriente principal del desarrollo se concentra en el servidor, donde Sun ha desarrollado una plataforma denominada "Java Enterprise", Java empresarial. El foco se ha puesto en la creación de aplicaciones de negocios, que corren en servidores preparados para albergarlas.

1.3.6 Acceso a base de datos

Uno de los paquetes de clases nombrados, el paquete JDBC, permite acceder a distintas bases de datos, de forma que se encapsulan las diferencias entre ellas. Basta tener un "driver JDBC" para la base que queremos manejar, y podremos acceder a sus datos. De esta forma, Java maneja prácticamente todas las bases de datos relacionales, desde Microsoft SQL Server, Oracle, Sybase, Informix, como Access y MySQL.

1.3.7 Manejo automático de memoria

Durante la ejecución de un programa Java, se crean objetos de distintas clases. En otros lenguajes, la creación de objetos debe estar acompañada de su correspondiente destrucción, a cargo del programador. Por ejemplo, en C++,

es el programador el que crea y destruye programáticamente los objetos. Esto ha sido fuente de múltiples errores, de dos tipos: o un objeto nunca es destruido, ocupando memoria, o un objeto es destruido cuando aún se necesita usarlo. En cambio, en Java, la liberación del objeto es automática. Hay un proceso, denominado "garbage collector" (recolector de basura, podemos traducir), que se encarga de detectar los objetos que no se usan más en nuestra aplicación, y los destruye. El algoritmo que se usa, asegura que sólo se destruye un objeto cuando el resto del programa no lo usa. En Java, el programador puede concentrarse en la lógica de la aplicación, sin tener que vigilar el uso de la memoria, que queda automáticamente manejada por el entorno.

1.3.8 Multitarea

Entre tantas capacidades, Java incorpora desde el principio el manejo de "threads", múltiples hilos de ejecución dentro de nuestro proceso. Con threads, podemos ejecutar varios métodos en paralelo, o en varios procesadores, si están disponibles. En otros lenguajes, este manejo se incorpora mediante extensiones y funciones adicionales. En el lenguaje Java, el manejo de threads ya está incorporado en las facilidades básicas del lenguaje. Está contemplado el acceso sincronizado a objetos y métodos.

1.3.9 Java como tecnología

Java es más que un lenguaje. El conjunto de paquetes disponibles, que siempre está creciendo, ya por aportes nuevos de Sun, ya por la creación de paquetes por otras empresas, permite que Java sea una tecnología totalmente extensible. Hasta se permite conectarse con librerías nativas, con lo que esta extensibilidad prácticamente no tiene límites.

Como vemos, Java tiene muchas facetas. Como objeto de estudio, Java es muy extenso, y, de alguna forma, es un "blanco móvil", ya que permanentemente se le agregan nuevas capacidades.

1.4. Herramientas utilizadas para el desarrollo del sistema

1.4.1. Diseño de la interfaz

Para el diseño de las vistas se utilizaría el *Dreamweaver MX*, del paquete de aplicaciones de Macromedia, conjuntamente con la herramienta Microsoft Word del paquete Office de Windows.

Con Dreamweaver MX se puede diseñar vistas con poco consumo de tiempo solo que el uso de esta herramienta requiere de adiestramiento de ahí que se utilizara el Microsoft Word para la confección de vistas que representan los modelos oficiales usados por los clientes logrando que el usuario encuentre el ambiente familiar.

La herramienta Dreamweaver MX posibilita la creación de una interfaz de usuario integrada, que facilita la interacción dado la potencialidad de sus componentes de trabajo, además permite la creación de sitios dinámicos con la posibilidad de inclusión de otras tecnologías como son, ASP, ASP.NET, PHP y JSP para este caso particular.

1.4.2. Lenguajes y tecnologías

HTML (Hypertext Markup Language)

HTML (*HyperText Markup Language*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces* (*hyperlinks*) que permitan la navegación a otros documentos o fuentes de información relacionadas, y con *inserciones* multimedia (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape).

Java Script

El Java Script es un lenguaje de programación que se utiliza dentro del html. Lo interpreta el navegador y produce alguna acción determinada en la página web

donde está insertado. Java Script es un lenguaje débilmente tipiado, basado en objetos y guiado por eventos, logrando con esto el dinamismo de las páginas que incluyan este tipo de código, útil para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

JSP

JSP, al igual que otros API en Java, es una especificación proporcionada por Sun para que la implementen los vendedores. La especificación JSP se añade a la funcionalidad proporcionada por la especificación de servlet.

Los servlets y Java Server Pages (JSP) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, los CGI (Common Gateway Interface), programas que generan páginas web en el servidor, o los ASP (Active Server Pages), un método específico de Microsoft. Sin embargo, se diferencian de ellos en otras cosas.

Los servlets son respuestas de java a las directivas CGI. Se ejecutan mejor que los CGI, se ejecutan en el servidor e intersectan las solicitudes de navegador, actuando como una especie de capa intermedia entre clientes y aplicación de menor nivel. Los servlets son adecuados para decidir como manejar las solicitudes de clientes e invocar otros objetos de lado servidor pero no son tan adecuados para generar contenido.

Las paginas JSP por otro lado puede ser diseñadas y desarrolladas no tanto como programas sino mas bien como páginas web. Las paginas JSP son ideales para situaciones en las que necesitamos mostrar anotaciones con contenido dinámico integrado. Sin embargo aunque generar HTML es mucho más fácil con JSP que con un servlet, las páginas JSP son menos adecuadas para manejar la lógica de procesamiento.

Las paginas JSP pueden utilizar JavaBeans con un alcance o extensiones de etiquetas especificados para alcanzar una clara separación entre contenido estático y el código Java que produce aplicaciones web dinámicas. Esto

permite que dichas páginas sean creadas y mantenidas por diseñadores con aptitudes de presentación que no necesitan tener contenidos de Java.

Estas características son las que separan la utilidad de las paginas JSP como vistas, de ahí que no se piense en separar el hémelo de las paginas JSP y los servlets sino estos como complementos el uno del otro en el modelo Vista-Controlador.

MySQL

¿Qué es MySQL?

MySQL es el servidor de bases de datos relacionales más popular, desarrollado y proporcionado por MySQL AB. Una de las razones que justifican el rápido crecimiento de popularidad de MySQL, es que se trata de un producto Open Source, y por lo tanto, va de la mano con este movimiento.

Una base de datos es una colección estructurada de datos. La información que puede almacenar una base de datos puede ser tan simple como la de una agenda, un contador, o un libro de visitas, ó tan vasta como la de una tienda en línea, un sistema de noticias, un portal, o la información generada en una red corporativa. Para agregar, acceder, y procesar los datos almacenados en una base de datos, se necesita un sistema de administración de bases de datos, tal como MySQL.

Una base de datos relacional almacena los datos en tablas separadas en lugar de poner todos los datos en un solo lugar. Esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se necesitan consultar datos. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estandardizado para acceder a bases de datos relacionales.

MySQL es Open Source

Open Source significa que la persona que quiera puede usar y modificar MySQL. Cualquiera puede descargar el software de MySQL de Internet y usarlo sin pagar por ello lo que lo hace muy útil para productores de aplicación de que requieran almacenamiento y gestión de datos y no cuenten con

financiamiento de proyecto. Inclusive, cualquiera que lo necesite puede estudiar el código fuente y cambiarlo de acuerdo a sus necesidades. MySQL usa la licencia GPL (Licencia Pública General GNU), para definir qué es lo que se puede y no se puede hacer con el software para diferentes situaciones. Sin embargo, si uno está incómodo con la licencia GPL o tiene la necesidad de incorporar código de MySQL en una aplicación comercial es posible comprar una versión de MySQL con una licencia comercial. Para mayor información, ver la página oficial de MySQL en la cuál se proporciona mayor información acerca de los tipos de licencias.

¿Por qué usar MySQL?

De manera predeterminada MySQL usa conexiones no encriptadas (inseguras) entre el cliente y el servidor, lo que significa que cualquier individuo mal intencionado puede ver, y aún modificar los datos que están siendo transmitidos entre éstos.

A partir de la versión 4.0, MySQL tiene soporte nativo para SSL, mientras tanto, en las versiones anteriores de MySQL se podía usar SSH para crear un canal de comunicación seguro, es decir, al usar SSH podemos establecer una conexión encriptada entre un cliente y un servidor MySQL. SSH es un protocolo que proporciona autenticación, encriptación e integridad de datos para asegurar las comunicaciones en una red, y en principio es una solución bastante efectiva para resolver el problema de la inseguridad de los datos que viajan sobre una red.

El servidor de bases de datos MySQL es muy rápido, seguro, y fácil de usar. Si eso es lo que se está buscando, se le debe dar una oportunidad a MySQL. El servidor MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha estado siendo usado exitosamente en ambientes de producción sumamente exigentes por varios años. Aunque se encuentra en desarrollo constante, el servidor MySQL ofrece hoy un conjunto rico y útil de funciones. Su conectividad, velocidad, y

seguridad hacen de MySQL un servidor bastante apropiado para acceder a bases de datos en Internet.

1.4.3. Lenguaje de modelación

Unified Modeling Languaje (UML) y Patrones

UML es una notación con la cual se construyen sistemas por medio de conceptos orientados a objetos. Esta prescribe un conjunto de notaciones y diagramas estándares, y describe la semántica esencial de lo que estos diagramas y símbolos significan.[LAR99]

Las características más generales de UML son:

- Tecnología de orientación a objetos.
- Viabilidad en la corrección de errores.
- Desarrollo incremental e iterativo.
- Participación del cliente en todas las etapas del proyecto.

1.4.4. El patrón Modelo-Vista-Controlador (MVC)

El patrón Modelo-Vista-Controlador (MVC) descompone una aplicación interactiva en tres grandes bloques:

El ***modelo*** contiene los datos y la funcionalidad de la aplicación. Es independiente de la representación de los datos.

Los objetos de negocio son los objetos del Modelo que implementan la lógica de negocio. Sus funciones fundamentales son:

- Realizar la validación de los datos introducidos por el usuario, tanto sintáctica (numérico, fecha, etc.) como lógica (importe menor que saldo, etc.)
- Ejecutar la petición realizada por el usuario. Para ello podrá valerse de transacciones contra Sistemas Host, consultas a bases de datos, consultas a proveedores de contenidos, etc.
- Generar los objetos que utilizarán las Vistas para mostrar los resultados obtenidos.

  Garantizar la navegación y el flujo de pantallas correcto.

Las ***vistas*** muestran la información al usuario de una cierta forma. Existen todas las que se necesite definir, es la encargada de interaccionar con el usuario y se corresponde con lo que tradicionalmente se conoce como Interfaz de Usuario.

Para este caso las vistas serán del tipo JSP. Las JSP's son las encargadas de generar las páginas HTML que constituyen la visualización de los resultados generados por la petición realizada por el usuario. Para recibir la información que deben mostrar acceden a objetos generados por el Modelo. La recepción de la información se puede hacer de varias maneras, aunque las más habituales son:

- Accediendo a objetos almacenados dentro del Modelo como propiedades.
- El Modelo almacena los resultados como atributos de la petición.
- El Modelo escribe los resultados en la "sesión".

Cada *vista* tiene un ***controlador*** asociado. Los controladores reciben entradas en forma de eventos que responden a mandos realizados por el usuario a través del ratón o del teclado. El control traduce estos eventos a peticiones a la *vista* o al *modelo*.

Cada petición se identifica mediante un parámetro. En base a esta identificación, el Controlador decide qué objeto u objetos de negocio (Modelo) debe ejecutar para resolver la petición.

Tras la ejecución de los objetos de negocio, y en función del resultado devuelto por estos, el Controlador determina qué JSP se usará para visualizar el resultado, generando una redirección que concluirá con la generación del código HTML de la página.

Ventajas

- Múltiples vistas del mismo modelo
- Vistas sincronizadas
- Flexibilidad para cambiar las vistas y los controladores
- La aplicación puede soportar distintos tipos de interfaz de usuario.

- Aumenta en gran medida el nivel de reusabilidad de código. Facilita una evolución continuada de los sistemas, sin puntos de ruptura, ya que un cambio en un sistema afectará a uno o más componentes pero nunca afectará significativamente al "core" de la aplicación.
- Facilidad de desarrollo y acortamiento del "Tiempo de Comercialización" gracias a la paralelización de tareas.

Inconvenientes

- Complejidad creciente.
- Cambios innecesarios. Puede ser que no todas las vistas estén interesadas en los cambios.
- Conexión entre la vista y el controlador. Hay que usar los dos a la vez.
- Si cambia el interfaz del modelo, hay que cambiar todas las vistas y todos los controladores.
- Acceso ineficiente a los datos en la vista. Puede necesitar varias llamadas al modelo para actualizar todos sus datos.
- Tanto la vista como el controlador son específicos de una plataforma.
- Algunas herramientas de diseño de interfaces de usuario incorporan parte del procesamiento de eventos entrada. El controlador deja de ser necesario.

1.4.5. Apache TomCat

Tomcat es un contenedor de Servlets con un entorno JSP. Un contenedor de Servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario. Es sin lugar a dudas el proyecto de software libre más famoso escrito en Java. Creado a partir de código donado por SUN Microsystems a la Apache Software Foundation, ha crecido hasta convertirse en la implementación oficial de referencia de Servlets y JSP sustituyendo a la implementación de SUN original. Esto lo convierte, obviamente, en el contenedor sobre el que todos los demás prueban su adhesión a las especificaciones.

Aunque siempre se le ha acusado de un rendimiento pobre, el caso es que a partir de las nuevas versiones de la serie 4.x, creadas para cumplir las especificaciones 2.3 de Servlets y 1.2 de JSP, Tomcat se ha reescrito entero lo que mejora considerablemente su rendimiento. Es más, sin ir más lejos, Tomcat es el contenedor Web que más se está utilizando en servidores de aplicaciones comerciales, donde gigantes como IBM con su todopoderoso WebSphere han elegido a Tomcat como contenedor Web. Otra de las muestras de lo estandarizado del uso de Tomcat en su utilización como servidor de Servlets y JSP por parte de los entornos de desarrollo, tanto libres como comerciales. Ejemplos de esto son Borland JBuilder o Eclipse que integran un servidor Tomcat para poder desarrollar nuestras páginas JSP y Servlets.

¿Por qué se necesita Tomcat para ejecutar Java en Apache?

El funcionamiento principal de Apache desde su creación fue la de *aceptar y responder* requisiciones de Páginas en Internet, y como fue mencionado en Servidores de Páginas y "Java Application Servers" , estas requisiciones correspondían a documentos estáticos (puro HTML), es por esto que cuando se requiere ejecutar algún tipo de contenido dinámico (programas) como "Java", es necesario coordinar los esfuerzos de Apache con otro ambiente, en el caso de "Java" es precisamente "Tomcat" quien ofrece facilidades para ejecutar los dos componentes más utilizados en ambientes "Java": *"JSP ("Java Server Pages")" y "Servlets"*.

1.4.6. Apache Axis

Axis es una implementación Open-Source de un "SOAP Engine"; a través de este componente es posible llevar acabo una comunicación mediante WebServices una de las variaciones más prometedoras del lenguaje xml.

En el caso de "Axis", éste se encuentra diseñado para ser ejecutado dentro de un "Java Application Server" o bien un "Servlet Engine" como Tomcat 4.x; aunque hoy en día ya existen otros "SOAP Engines" que pueden ser ejecutados de manera independiente de un "Java Application Server" o "Servlet Engine", al residir un "SOAP Engine" dentro estos, se permite que métodos

residentes puedan ser publicados como "Web-Services" y de esta manera ser accesibles de otras plataformas/lenguajes que también soporten "SOAP"

Para utilizar Axis dentro de (Tomcat 4.x) es necesario diseñar un WAR ("Web-Archive") donde residirán las Clases/Métodos que serán ejecutados como "Web-Services" así como *Axis*; para simplificar la instalación de *Axis* se asume que se utilizará la configuración "Default" de (Tomcat 4.x)

1.4.7. Eclipse

Es un entorno de desarrollo integrado IDE (Integrated Development Environments) Open Source (CPL) cedido por IBM. Este entorno de desarrollo integrado ofrece, el control del editor de código, del compilador y del depurador desde una única interfaz de usuario. Su misión consiste en evitar tareas repetitivas, facilitar la escritura de código correcto, disminuir el tiempo de depuración e incrementar la productividad del desarrollador. Estas tareas pueden realizarse de muchas maneras distintas: mediante la inclusión de asistentes para las tareas más habituales y mecánicas, de editores que completen automáticamente el código y señalen los errores sintácticos, de gestores de archivos fuente, etc. Eclipse no es un IDE más a añadir a la lista, el objetivo de IBM ha sido crear una plataforma de desarrollo modular que cualquier herramienta de desarrollo pueda usar con cualquier lenguaje de programación.

Además Eclipse es una plataforma universal para integrar herramientas de desarrollo, basada en plug-ins. Plataforma universal, pues emplea una estructura abierta de plug-ins que permite expandir las capacidades de la plataforma base hasta el infinito; pudiendo ser añadidos automáticamente al entorno de desarrollo, lo que lo convierte en uno muy adecuado para el desarrollo de software.

1.4.8. DBDesigner

Esta herramienta es destacable por su sencillez. Además de modelar sobre MySQL, dispone de la capacidad de generar documentación e incluso pantallas de administración sobre PHP.

Con DBDesigner una vez que tenga diseñada la base de datos usted puede exportar su diseño en un .sql o se puede conectar con un servidor de base de datos y construirla allí, además puede importar base de datos existentes escritas en .sql, y puede guardar sus datos en formato xml nativo.

Para mi gusto esta herramienta está más pensada para trabajar con modelos físicos (como MySQL) que con modelo conceptuales de datos. Es sencilla, bastante estable (aunque la interfaz tiene algunos errores), rápida y además es GRATIS.

Conclusiones

Teniendo en cuenta la necesidad de la implantación de una aplicación para la automatización de los servicios médicos de la UCI y la potencialidad del uso de la tecnología que propicia J2EE como una tecnología portable ya probada por sus seguidores dados los años de implementación, crecimiento de las potencialidades de las aplicaciones modeladas según los conceptos de Modelo-Vista-Controlador como poderosa tecnología de desarrollo, con el objetivo de la parametrizabilidad e integración de aplicaciones, se perfila como positiva y necesaria la idea de combinar estos conceptos con el fin de lograr una aplicación con la posibilidad de brindar una mayor satisfacción de los usuarios que interactúan con dicho sistema, garantizando la prestación de servicios con alta velocidad y confiabilidad de la personalización de los ambientes. A nivel empresarial, esta fusión proporciona la facilidad de obtener información actualizada en tiempo real y totalmente dinámica y cambiante, además de la posibilidad de comunicación y unificación de la información gestionada a favor de los pacientes, útil para la óptima toma de decisiones, dada la posibilidad de tener habilitada la aplicación todo el día y todos días del año, permitiendo así acceder a la información que se necesita de acuerdo con su disponibilidad y contando con la ausencia de una aplicación de este tipo hasta el momento.

Capítulo 2

Discusión Teórica

Introducción

Con la introducción de nuevas aplicaciones y de plataformas del desarrollo, las infraestructuras de la Web están llegando a ser cada vez más complejas. Las herramientas de prueba de la Web le ayudarán a los programadores determinar el funcionamiento del servidor bajo cargas anticipadas y observar cual será su desempeño ante pruebas de rendimiento, pero encontrar la herramienta justa para satisfacer los requisitos de prueba de una empresa puede ser un desafío.

El propósito de esta sección es explicar qué se debe hacer para llevar a cabo pruebas de escalabilidad, pruebas de rendimiento y optimización en un entorno J2EE (Java 2 Enterprise Edition) y como seleccionar el software adecuado para realizar estas pruebas.

2.1 Definiciones

2.1.1 Tiempo de respuesta (Response time) - El tiempo que pasa entre la petición inicial y la descarga completa de la respuesta (es decir, el desplegado por entero de la página web).

2.1.2 Carga (Load) - Una medida del uso del sistema. Se dice que un servidor tiene "carga elevada" cuando la aplicación que soporta experimenta un fuerte tráfico.

2.1.3 Escalabilidad (Scalability) – Una aplicación escalable tiene un tiempo de respuesta que aumenta linealmente cuando aumenta la carga. Dicha aplicación es capaz de procesar cada vez más volumen si se añaden recursos adicionales de hardware de forma lineal (es decir, no exponencial).

2.1.4 Herramientas de automatización de pruebas (Automation testing tools) – Herramientas (Silk de Segue Software, WebLoad, etc.) que se usan para simular un usuario haciendo peticiones de páginas o ejecutando flujo de trabajo preprogramado en un sitio.

2.1.5 Herramientas de pruebas de carga (Load testing tools) – La mayoría de herramientas de automatización de pruebas (por ejemplo, WebLoad) pueden usarse como software de pruebas de carga. Estas herramientas simulan cualquier cantidad de usuarios usando un sitio y proveen datos importantes como, por ejemplo, tiempos medios de respuesta.

2.1.6 Perfilador (Profiler) - Un perfilador es un programa que examina una aplicación mientras ésta se ejecuta. Provee útil información de ejecución como el tiempo invertido en determinados bloques de código, el grado de utilización de la memoria y del heap, el número de instancias de cada clase que hay en memoria, etc.

2.2 Un Proceso para la prueba de rendimiento.

2.2.1 Pruebas funcionales. La mayoría de aplicaciones comienzan el proceso de prueba completando, antes que nada, las pruebas funcionales. Es decir, asegurándose que funcionan todos los casos de uso y flujo de trabajo de la aplicación.

2.2.2 Pruebas de carga y escalabilidad. El proceso de probar la carga y escalabilidad tiene dos aspectos:

a. Probar el tiempo de respuesta cuando se aumenta el tamaño de la base de datos.
b. Probar el tiempo de respuesta cuando se aumenta el número de usuarios concurrentes.

2.2.3 Interpretación de los resultados. Después de medir el tiempo de respuesta con diferentes cargas y tamaños de la base de datos, se pueden hacer interpretaciones basándose en el tiempo medio de

respuesta obtenido en las pruebas y el uso de recursos del servidor durante las mismas.

2.2.4 Optimización. Después de identificar anomalías en el paso anterior, se interpretan los resultados y se localiza el problema.

2.3 Pruebas de carga y escalabilidad.

El propósito de las pruebas de carga y escalabilidad es asegurar que la aplicación tendrá un buen tiempo de respuesta durante los picos de uso. También se puede evaluar cómo la aplicación se comportará a lo largo del tiempo (conforme el sitio Web contenga cada vez más información en la base de datos).

Para ejecutar las pruebas de rendimiento, deberá simular el uso del servidor con diferentes cargas. Como regla general, una simulación de carga baja (es de uno a cinco usuarios concurrentes), carga media (de 10 a 50 usuarios concurrentes), carga alta (100 usuarios concurrentes) y carga extrema (1000 o más usuarios concurrentes). Nótese que esos números son arbitrarios y dependen de las necesidades del negocio. Además, simular 10 usuarios concurrentes con software de pruebas de carga no es representativo de 10 personas, ya que cada "robot" en la prueba de carga puede esperar sólo milisegundos antes de acceder de nuevo al servidor. Por lo tanto, usar un probador de carga para simular 10 usuarios es probablemente representativo de los comportamientos de navegación en el web de 30 o 40 personas.

Una vez ha probado estos tres niveles de carga, puede comparar tiempos medios de respuesta para ver si su sistema es escalable, es decir, si el tiempo de respuesta aumenta linealmente.

2.4 Interpretando los resultados.

La parte divertida de este proceso es interpretar los resultados de las pruebas de carga. Examinemos algunas de las diferentes posibilidades:

1. **El tiempo de respuesta aumenta demasiado cuando la base de datos se llena mucho**
   El tiempo de respuesta no debería aumentar demasiado si se pasa de una base de datos con 100 filas en sus tablas a una con 50,000 filas. La tecnología de indexado de bases de datos hace que hallar una fila en una tabla tarde unos cuantos milisegundos, incluso si hay cientos de miles de filas. Por lo tanto, si el tiempo de respuesta aumenta mucho después de pasar de una base de datos de tamaño moderado a una de tamaño extremo, entonces probablemente aún no se han indexado las columnas apropiadas.
2. **El tiempo de respuesta aumenta exponencialmente cuando aumenta la carga**
   Si el sistema se vuelve inutilizable conforme se aumenta el número de usuarios concurrentes, entonces el sistema no es escalable. Interpretar estos resultados es difícil pues el problema podría ser causado por el hardware, la configuración de despliegue, la arquitectura, etc. Asegúrese de que observa los recursos del servidor durante las pruebas:
   a. Observe los requerimientos de memoria.
   b. Observe el uso de CPU. Si la CPU se usa demasiado, se necesita un procesador más rápido o más procesadores. Si la CPU se usa poco, entonces probablemente el problema esté en la entrada/salida. Compruebe las conexiones de bases de datos, el número de threads (hilos) que se ejecutan y la configuración de la red en las máquinas de prueba.

Si el problema no se resuelve después de comprobar la configuración, de verificar que la lentitud no se debe a un cuello de botella del hardware y de revisar rápidamente la arquitectura buscando código para optimizar, es el momento de ejecutar un perfilador de código.

2.5 Optimización.

Se necesita optimizar la base de datos, la arquitectura, la configuración y el hardware. Como se mencionó en el apartado anterior, el motivo más común de que la aplicación no sea escalable es que no se haya afinado la base de datos.

Después de optimizar la base de datos y optimizar la configuración de hardware, el siguiente paso es optimizar el código y esto se hace con un perfilador.

Un perfilador es un programa que analiza la aplicación mientras ésta se ejecuta. Un perfilador provee información a la que no se podría acceder de otra manera, como, por ejemplo:

1. **Cuantos objetos de cada clase hay en memoria y el comportamiento del recolector de basura** (garbage collector).
    - Esta información puede ayudar a identificar clases que deberían estar en un "pool".
    - También puede ayudar a afinar el "heap" de Java.
2. **Cuanto tiempo la aplicación pasa en determinadas clases.** Esta es la función más importante. El perfilador indicará qué clases son los cuellos de botella.

La optimización de la arquitectura es sumamente específica para cada proyecto, pero a continuación se incluyen algunos consejos:

1. **Asegúrese de que se han minimizado las llamadas a la red, especialmente las llamadas a la base de datos.**
    - Es mejor tener una base de datos grande que muchas pequeñas.
    - Confirme que ejbStore no almacena nada para operaciones de sólo lectura.
    - Use objetos de detalle para obtener el estado de los beans de entidad.

2. **Asegúrese de aprovecharse de la "cache" cuando sea posible.**
   Su servidor de aplicaciones probablemente permite colocar los beans de entidad en la "cache" de memoria. Asegúrese de que se aprovecha de ello, ya que reducirá dramáticamente las llamadas a la base de datos y acelerará el acceso a los datos.
3. **Asegúrese de que está usando beans de sesión como una "fachada" de los beans de entidad.**
   Puede encapsular el flujo de trabajo de un caso de uso completo en una llamada de red a un único método de un bean de sesión (y en una única transacción).

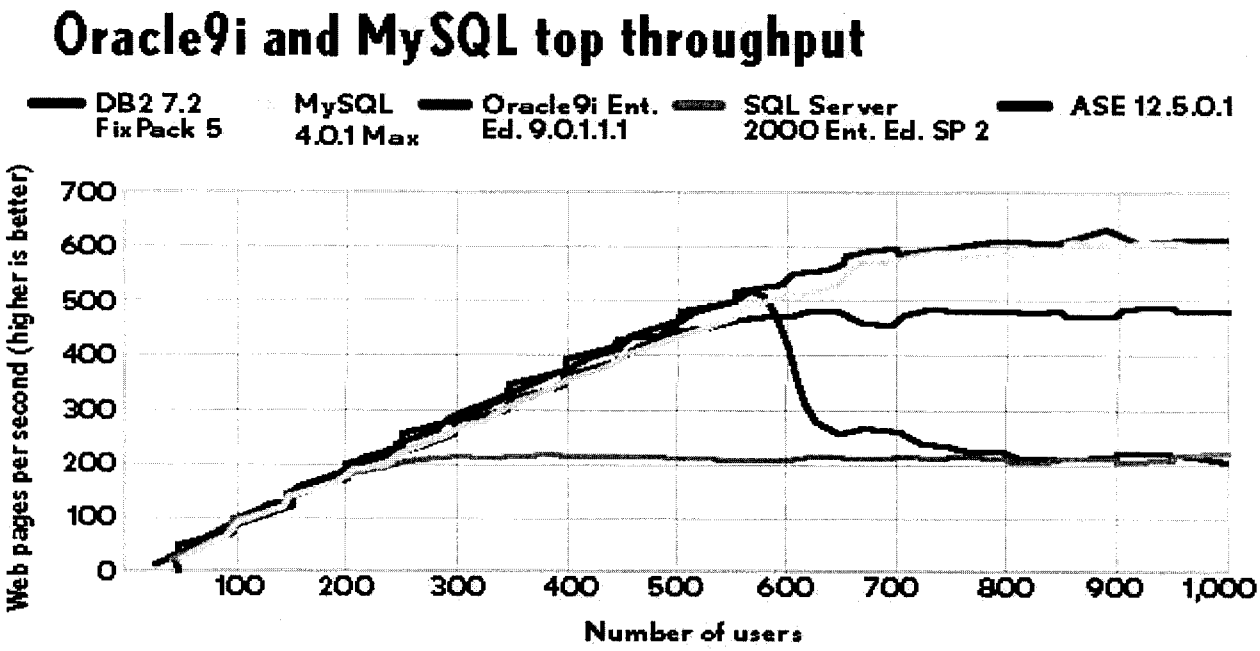
Realizar pruebas de rendimiento y optimizar una aplicación puede ser una tarea muy desafiante. Afortunadamente hay herramientas en el mercado que pueden simplificar el proceso. Usando esas herramientas y siguiendo los pasos sencillos que se han explicado, usted debería ser capaz de seguir la pista a los cuellos de botella de un sistema de forma efectiva.

Algunas de estas herramientas son Empirix's e-Test Suite, Mercury LoadRunner, Pushtotest, TPC Benchmark W (TPC-W), Compuware's, Optimize-It entre otros.

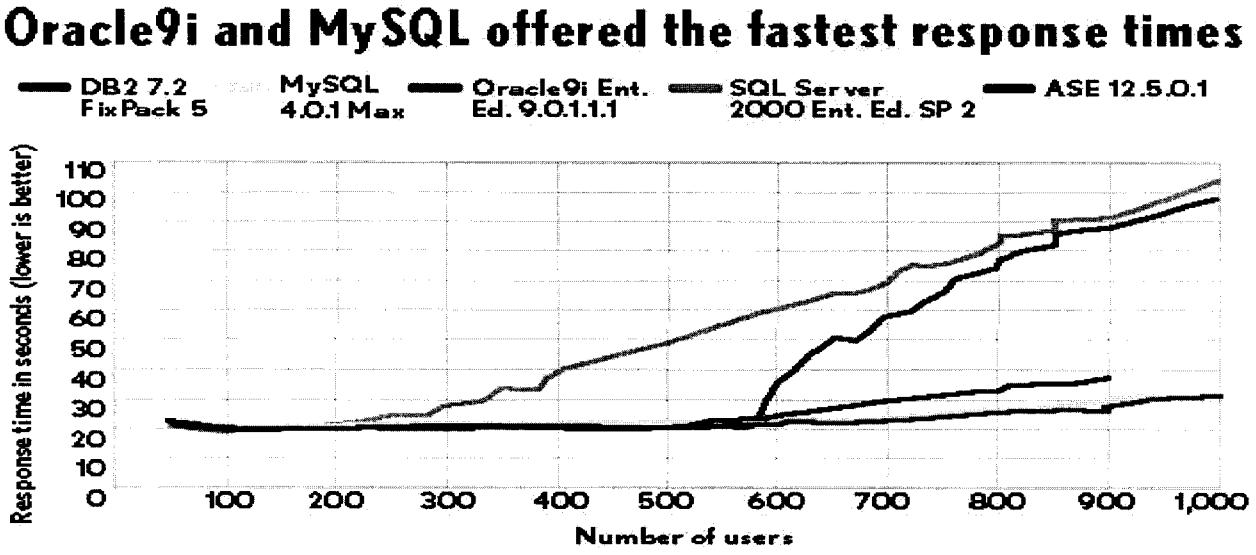
2.6 Empirix's e-Test Suite:

Empirix's e-Test Suite ofrece una herramienta comprensiva para probar aplicaciones Web con un ambiente Windows. La última versión agrega algunas capacidades útiles para realizar pruebas de test a Web Services. Muchos usuarios plantean que lo que necesitaría Empirix es un wizard para hacer el sistema un poco más intuitivo. Con más de 3,000 clientes satisfechos y una historia impresionante de innovación, Empirix está redefiniendo las normas para Web y esta trabajando en las pruebas de aplicaciones de voz. Empirix's e-Test Suite cuesta $20,000 para simular 50 usuarios virtuales.

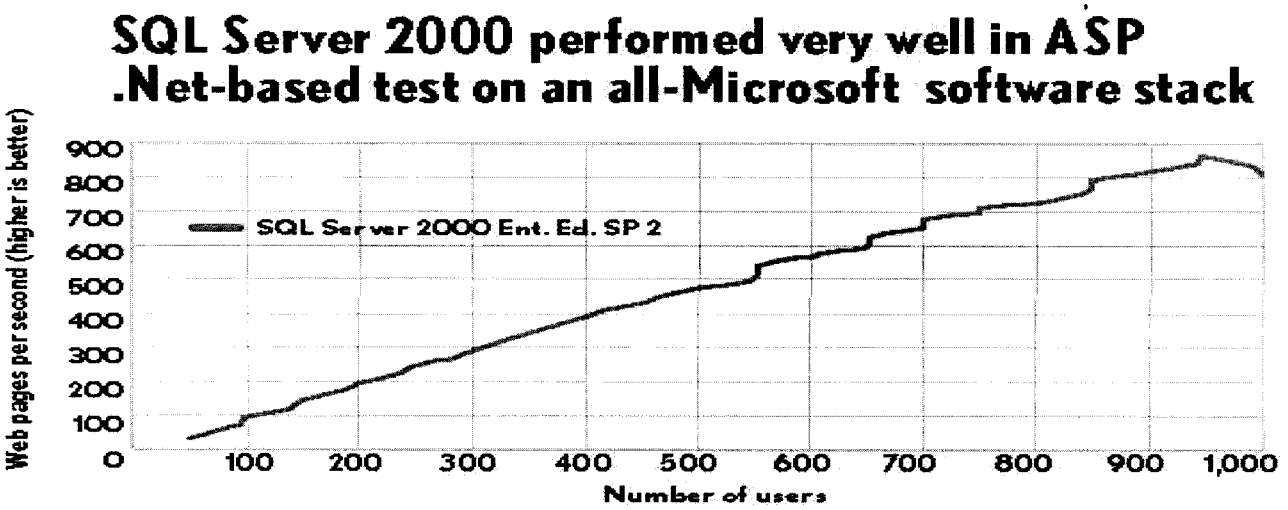
Algunas pruebas realizadas con Empirix.



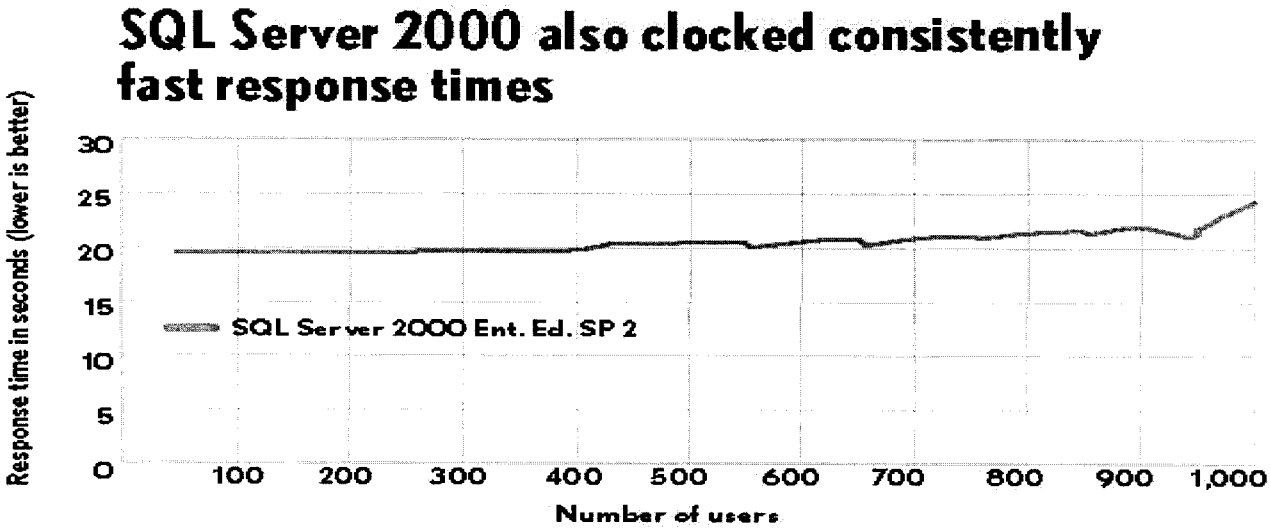
Throughput is in returned Web pages per second from the application server. Number of users is number of concurrent Web clients driving the load. Response time is the time to complete the six bookstore user action sequences, weighted by frequency of each sequence in the mix. All tests were conducted on an HP NetServer LT 6000r with four 700MHz Xeon CPUs, 2GB of RAM, a Gigabit Ethernet Intel Corp. Pro/1000 F Server Adapter and 24 9.1GB Ultra3 SCSI hard drives used for database storage.



Throughput is in returned Web pages per second from the application server. Number of users is number of concurrent Web clients driving the load. Response time is the time to complete the six bookstore user action sequences, weighted by frequency of each sequence in the mix. All tests were conducted on an HP NetServer LT 6000r with four 700MHz Xeon CPUs, 2 GB of RAM, a Gigabit Ethernet Intel Corp. Pro/1000 F Server Adapter and 24 9.1GB Ultra3 SCSI hard drives used for database storage.



Throughput is in returned Web pages per second from the application server. Number of users is number of concurrent Web clients driving the load. Response time is the time to complete the six bookstore user action sequences, weighted by frequency of each sequence in the mix. All tests were conducted on an HP NetServer LT 6000r with four 700MHz Xeon CPUs, 2 GB of RAM, a Gigabit Ethernet Intel Pro/1000 F Server Adapter and 24 9.1GB Ultra3 SCSI hard drives used for database storage.



Throughput is in returned Web pages per second from the application server. Number of users is number of concurrent Web clients driving the load. Response time is the time to complete the six bookstore user action sequences, weighted by frequency of each sequence in the mix. All tests were conducted on an HP NetServer LT 6000r with four 700MHz Xeon CPUs, 2 GB of RAM, a Gigabit Ethernet Intel Pro/1000 F Server Adapter and 24 9.1GB Ultra3 SCSI hard drives used for database storage.

2.7 Mercury LoadRunner:

LoadRunner, de Mercury Interactive, es utilizada para predecir el comportamiento del sistema y su rendimiento bajo condiciones extremas con una pesada carga de usuarios, simulando la actividad de usuarios reales.

Cada uno de estos usuarios utiliza las aplicaciones con transacciones reales, mientras LoadRunner mide los tiempos de respuesta, los retrasos en las redes y el rendimiento, tanto de los servidores como de las aplicaciones, para ayudar a aislar los cuellos de botella y ajustar al máximo el rendimiento dentro de la infraestructura de las aplicaciones. LoadRunner proporciona una monitorización a medida, integrada y en tiempo real, que captura y muestra los datos de rendimiento desde varios niveles del sistema durante la prueba de carga. Estas capacidades permiten a los clientes minimizar sus ciclos de realización de pruebas, optimizar el rendimiento y reducir los costes globales. En fin LoadRunner es complicado, caro y muy bueno.

2.8 TPC Benchmark W (TPC-W).

El TPC Benchmark W (TPC-W), está dedicado a las transacciones web. El volumen de trabajo es ejecutado en un ambiente de comercio Internet controlado que simula las actividades de un negocio orientado a las acciones transaccionales de un servidor web. Los ejercicios de volúmenes de trabajo asociados a un extenso sistema de componentes en cada ambiente, están caracterizados por:

* Múltiples sesiones de navegación "on-line".
* Generación dinámica de páginas con acceso y actualización de las bases de datos.
* Objetos web consistentes.
* Ejecución simultánea de múltiples tipos de transacciones que abarcan un amplio espectro de complejidades.
* Bases de datos que constan de diferentes tablas, con una amplia variedad de tamaños, atributos y relaciones entre ellas.

Este también es muy bueno pero caro.

Ejemplo resultados de TPC-W

Top Ten TPC-W Results by Price/Performance

Item Count 10,000

Rank	Company	System	WIPS	$/WIPS	System Availability	Database	Operating System	HTTP Server	Date Submitted
1	DELL	Dell PowerEdge 6400/500 with PowerApp.Web 120	7,783	24.50 US $	01/28/02	Microsoft SQL Server 2000 Standard Ed.	Microsoft Windows 2000 Advanced Server	Microsoft Internet Information Server 5.0	01/28/02
2	DELL	Dell PowerEdge 6400/500 with PWebL20	6,622	25.70 US $	12/19/01	Microsoft SQL Server 2000 Standard Ed.	Microsoft Windows 2000 Advanced Server	Microsoft Internet Information Server 5.0	12/19/01
3	DELL	PowerEdge 6650 1.8GHz w/ PowerEdge 1650 1.4GHz	10,445	27.60 US $	08/22/02	Microsoft SQL Server Enterprise Edition SP2	Microsoft Windows 2000 Advanced Server	Microsoft Internet Information Server 5.0	08/22/02
4	IBM	IBM eServer xSeries 350 with IBM eServer 330	7,073	31.77 US $	12/17/01	Microsoft SQL Server 2000 Standard Ed.	Microsoft Windows 2000 Advanced Server	Microsoft Internet Information Server 5.0	12/17/01

WIPS – Web Interactions Per Second durante sesiones de compras
80% de interacciones de navegación (browse)
20% de interacciones de pedido
Coste/rendimiento ($/WIPS) = Coste Hard+Coste Soft+Coste Mant./WIPS

2.9 PushToTest:

PushToTest presenta TestMaker, una utilidad de código libre para construir agentes encargados de medir la escalabilidad, actuación y funcionalidad de los servicios web. PushToTest personaliza TestMaker a las necesidades específicas del sistema, dirige la escalabilidad y la actuación de la prueba. Este único acercamiento comercial entrega las soluciones baratas.
TestMaker es una aplicación 100% realizada en Java y corre dondequiera que este instalado Java, incluso Windows, Linux, Solaris, y Macintosh OS X. solamente requiere Java 1.4.1 ó una versión mayor. Además existe el **TestMaker 4.0.9** gratis en Internet.

2.10 Probando y optimizando THESERVERSIDE.COM

TheServerSide.com experimentó numerosos problemas de escalabilidad antes de su lanzamiento. Usando los pasos mencionados anteriormente, se resolvieron todos los problemas, por lo que TheServerSide.com es uno de los portales basados en Java más rápidos que hay.

El primer paso para probar TheServerSide.com fue llenar la base de datos con datos de prueba. Después de llenarla con una cantidad moderada y con una cantidad extrema (añadiendo 16,000 mensajes y 40,000 usuarios a la base de datos), se encontraron un problema serio. El tiempo de respuesta de las páginas de nivel máximo saltó desde los 2 segundos a 12 segundos para un único usuario.

Si conocer los pasos necesarios se cometieron los errores más comunes: Doblaron la velocidad de la CPU y la memoria de la máquina. Esto sólo redujo el tiempo de respuesta a 8 segundos y, por lo tanto, no era la causa del cuello de botella.

El problema que tenían indicaba que algo fallaba en la base de datos. Después de comprobar cómo la base de datos procesaba las consultas, descubrieron

que las columnas de clave primaria (y otras) no se indexaban correctamente. Esto significa que la base de datos tenía que hacer búsquedas lineales.

Una vez optimizada la base de datos, comenzamos a ejecutar pruebas de carga apropiadas. Usaron WebLoad, una potente herramienta para pruebas de cargas. La copia de evaluación permite probar con 12 usuarios concurrentes (probablemente representativos de 30 o 40 personas reales). Después de ejecutar las pruebas al máximo (12 usuarios concurrentes), encontraron que el sitio Web era muy poco escalable. El tiempo de respuesta saltó de los 3 o 4 segundos de un único usuario a 15 o 20 segundos por página bajo cargas más pesadas. Obviamente, estos números no eran lo suficientemente buenos.

Habiendo optimizado la base de datos y actualizado el hardware, pasaron a examinar la arquitectura. Se hicieron pequeñas modificaciones, pero aún no podían encontrar la causa del problema. Cuando empezaron a usar un perfilador, todo cambió. Después de ejecutar remotamente Optimize-It (tenía una ventana en la máquina local mostraba las estadísticas del servidor ejecutándose en su proveedor de servicios de Internet), se descubrió la causa del problema. 30% del tiempo de CPU se perdía en comunicaciones de socket con la base de datos. Optimize-It permitió seguir la pista y ver qué objetos y métodos iniciaban estas llamadas. Así se identificó el problema de diseño que hacía que se consultaba un conteo en la base de datos cada vez que querían mostrar un mensaje en TheServerSide.com. Después de arreglar ese problema tonto, el número de llamadas a la base de datos usadas para mostrar una página bajó de 15 a 1, y, de repente, el tiempo de respuesta disminuyó a cerca de un segundo. Esto era exactamente lo que se quería.

Conclusiones:

Recomendamos que cuando se termine la fase de implementación del sistema completó se le realicen las pruebas expuestas anteriormente siguiendo los pasos necesarios para poder encontrar posibles errores. De los software existentes y entre los mencionados debería de utilizarse el TestMaker de PushToTest debido que con el se pueden realzar estas pruebas fácilmente y se encuentran versiones gratis en Internet.

Capítulo 3

Características del sistema

Introducción

A través de este capítulo se describe el objeto de estudio, el entorno de trabajo en que se desarrolla la aplicación, se realiza la propuesta del sistema y se analizan los requerimientos funcionales y no funcionales Se realiza el modelado del negocio y se realiza la definición de los casos de uso, de los actores que intervienen en ellos y se muestra el diagrama resultante de casos de uso.

3.1 Problema y situación problémica

Módulo Laboratorio Clínico

En la UCI existe un centro médico el cual brinda sus servicios a todas las personas que trabajan o estudian en ella, dentro del centro médico esta presente el laboratorio clínico donde se le realizan los exámenes complementarios a los pacientes, que les son indicados por los médicos en una consulta, además se brindan diferentes partes mensuales a estadística. Todo esto se realiza de la siguiente manera: el médico cuando esta consultando a un paciente y necesita indicarle un examen complementario este escribe la indicación en un papel, el paciente entonces debe presentarse en laboratorio clínico con dicha indicación donde el técnico le recoge una muestra la cual va a ser analizada y posteriormente a partir de ella se obtiene si dicho examen es positivo o no, este resultado lo puede ir a recoger el paciente o el médico que se lo indico para que en la siguiente consulta el médico pueda realizar buen juicio sobre la patología del paciente.

Además el personal que labora en el laboratorio clínico debe rendir muchos reportes diarios y mensuales a estadística y todo este trabajo es realizado de forma manual manejando una gran cantidad de datos referentes a los exámenes complementarios.

La ausencia de un sistema automatizado para responder a todos los servicios que brinda el laboratorio clínico afecta al personal del laboratorio y los médicos que necesitan indicarle un examen complementario a los pacientes. Con el impacto de realizar todas las acciones expresadas anteriormente, una solución exitosa pudiera ser contribuir al funcionamiento eficiente de varios servicios y elevar la satisfacción del personal. Acelerar el proceso de informatización integral de los servicios médicos.

Módulo Historia Clínica Personal

Unos de los problemas que afectan estos servicios son las transacciones con el documento oficial de Historia Clínica Personal de cada paciente ya que se almacenan en formato de papel, el cual se puede deteriorar con el transcurso del tiempo. También se hace costoso la actualización y creación de las mismas. Además la lectura de estas puede resultar engorrosa por haber sido actualizadas por diferentes médicos y muchas veces la extensión de los resúmenes médicos trae consigo la necesidad de adjuntar papeles para culminar el examen.

3.2 Objeto de automatización

Módulo Laboratorio Clínico

Se desea automatizar todos estos procesos expresados anteriormente donde el médico va a poder indicarle un examen complementario (EC) a un paciente y esta indicación será recepcionada de manera automática en el laboratorio clínico donde el paciente solo deberá presentarse con la identificación necesaria para realizarle el examen. Una vez que el resultado sea introducido en el sistema por el técnico, el médico podrá acceder a este en cualquier momento, por otra parte todos los reportes serán realizados por el sistema facilitándole el trabajo al personal del laboratorio clínico.

Módulo Historia Clínica Personal

Como parte del proceso de automatización de los servicios médicos en la UCI es de máxima necesidad contar dentro de este con un sistema automatizado de las Historias Clínicas Personales, que le permita al personal médicos

almacenar información digital de todo lo referente a un paciente interno de la UCI, teniendo así un historial médico de todos las personas que transitaron por el centro. Con este sistema se les facilitara el trabajo en tiempo y costo a los servicios médicos del país, proporcionándoles un acceso rápido, seguro, centralizado y con opciones de desarrollo.

3.3 Información que se maneja

Módulo Laboratorio Clínico.

La información que se manipula serán todos los datos del paciente al que se le indica el EC, los datos del médico que indica este EC, el tipo de examen que se le va a realizar a dicho paciente, el tipo de solicitud que puede ser urgente o electivo, la fecha en la que es indicado el EC, la procedencia de la solicitud, es decir si un EC es solicitado durante una consulta MGI al procedencia será el nombre del consultorio donde se realizo dicha consulta, si esta consulta es externa que seria si es realizada cuando el médico esta haciendo terreno por la residencia de la UCI, la fecha de atención del paciente en el laboratorio clínico y los datos del técnico que lo atendió, el número de la muestra del EC, la fecha en la que se inserto en el sistema el resultado del EC, los datos del técnico que lo insertó, el resultado del EC y una breve descripción del resultado obtenido que es escrita por el técnico que realizó.

Referente a los tipos de exámenes complementarios que se le pueden indicar a los pacientes se debe conocer el código de este, su nombre, el área donde se realiza este EC, si el resultado obtenido viene dado por un intervalo de positividad, si es así se necesita dicho intervalo y si este EC esta disponible en este momento.

Módulo Historia Clínica Personal

La información que se maneja dentro de la Historia Clínica Personal de cada paciente en el sistema de salud pública cubano se encuentran los datos referentes a:

- Nombre de paciente

- Primer apellido
- Segundo apellido
- Carné de identidad
- Grupo sanguíneo
- Factor RH
- Sexo
- Raza
- Alergia
- Grupo dispensalarial
- Grado de escolaridad
- Domicilios: aquí se archivan toda la información del lugar donde con vive la persona, la fecha en que se dio alta en ese domicilio y en caso de baja la fecha de esta y al consultorio médico al que pertenecía, así con cada uno de los domicilios por el que pase.
- Antecedentes Patológicos Familiares: aquí se archiva toda la información del diagnostico presentado por los familiares del paciente y el parentesco que tiene con el mismo.
- Antecedentes Patológicos Personales: se guardan todos los diagnósticos presentados por el paciente, la fecha del diagnostico y en caso de operación que es otro antecedente el tipo de la operación y la fecha en que esta se llevo acabo.
- Factores de Riesgo para mayores de quince años: esta información es para el paciente en su etapa de vida después de los quince años de edad, aquí se almacenas datos referentes a fecha en que se le hizo el chequeo, el índice de masa corporal, si fuma o no, la tensión arterial máxima, mínima ,si toma bebidas alcohólicas, si tiene incremento de lípidos sanguíneos, si es sedentario o no, si se le realizo la prueba psicológica, un examen de mamas, un examen bucal, un tacto rectal, los riesgos precondicional del paciente, si fue controlado o no y el método que se usó.
- Menores de un año: esta información es para la etapa del paciente hasta el primer año de vida, almacenado la fecha del día del examen, la talla, el peso, el C.C, el C.T y el V.N.

- Esquema de Vacunación: en este caso se almacena la información referente a la fecha en que su puso la vacuna, la descripción de esta vacuna y el tipo, esto es para un numero fijo y conocido de vacunas.
- Consulta: se adjunta a la historia clínica personal todos los datos referentes a una consulta como médico que la realizo, la fecha, la impresión diagnostica, el motivo de la consulta, la historia de la enfermedad actual, conducta a seguir, reconsulta, datos del examen físico realizado por aparatos, los tratamientos y los exámenes complementarios mandados.

3.4 Propuesta del sistema.

Módulo Laboratorio Clínico.

Se propone un sistema que podrá almacenar todos los datos referentes a los EC y brindara la facilidad a los médicos de indicar EC y recibir los resultados de estos de manera automática, además le brindara al personal del laboratorio clínico diversas facilidades para la realización de los EC y los para la realización de los análisis de estos en la obtención de reportes estadísticos.

Módulo Historia Clínica Personal

Este sistema propone el almacenamiento de las historias clínicas de los pacientes internos de la UCI, la modificación, creación de esta y de todos los datos que esta recoge, así como mostrarle al sistema médico que lo solicite, los datos de cada paciente.

3.5 Modelo de negocio.

Módulo Laboratorio Clínico

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

De ahí, que en el campo del software también resulte útil la creación de modelos que organicen y presenten los detalles importantes de problemas

reales que se vinculan con el sistema informático a construir. Estos modelos deben cumplir una serie de propiedades, entre ellas la de ser coherentes y relacionados. Uno de los modelos útiles previo al desarrollo de un software es el modelo del negocio.

El modelado del negocio es una técnica para comprender los procesos del negocio de la organización. Los propósitos que se persiguen al realizarse el modelado del negocio, son:

- Entender la estructura y la dinámica de la organización.
- Entender los problemas actuales e identificar mejoras potenciales.
- Asegurarse de que los clientes, usuarios finales y desarrolladores tienen una idea común de la organización.
- Derivar los requerimientos del sistema a partir del modelo de negocio que se obtenga.

Módulo Historia Clínica Personal

El trabajo con las historias clínicas personales en los servicios médicos de la UCI actualmente se realiza de la siguiente forma: cuando llega un nuevo paciente al consultorio el médico le llena una nueva historia clínica personal y en esta todos los datos precisos, después se pasa a archivar estos papeles en un lugar determinado, esto lo hace la enfermera del consultorio. En caso de que el paciente tenga historia clínica asignada la enfermera busca el historial y en ese momento se realiza la consulta, adjuntando en la historia clínica los datos de la consulta.

Este proceso algo engorroso, por las transacciones diarias que se realizan dentro de cada historia clínica personal solicitado por el médico, además de la poca capacidad de archivo que presentan los consultorios médicos. Para esto damos como pronta solución, la informatización de este proceso de gestión de la historia clínica.

3.6 Clientes y trabajadores del negocio.

Después de todas las entrevistas y estudios necesarios realizados (haciendo una simplificación del negocio) se obtiene un listado de clientes y trabajadores del sistema como se muestra a continuación:

Módulo Laboratorio Clínico

Actores del negocio	Justificación
Médico	El médico puede ser un médico MGI, un médico de estomatología, un médico de ITS los que pueden indicar un EC desde una consulta.
Estadística	Recibe los reportes del laboratorio clínico.
Paciente	Se le indica y se le toma la una muestra para la realización del examen complementario

Trabajadores del negocio	Justificación
Técnico	Es el encargado de realizar los EC en el laboratorio clínico e enviarle los reportes a la subdirección y estadística.

Módulo Historia Clínica Personal

Actores del negocio	Justificación
paciente	En este caso hago referencia a todo el personal interno de la UCI. Que llegan a la

	consulta por motivos médicos y le solicitan una consulta a la enfermera.

Trabajadores del negocio	**Justificación**
Personal médico	En este grupo encontramos a los médicos de los consultorios, los médicos MGI del policlínico y los especialistas que vienen al centro en días específicos. Es el personal que hace todas las gestiones sobre la historia clínica personal de cada paciente.
Enfermeras	En el grupo se encuentran las enfermeras de los consultorios médicos y de la consulta MGI del policlínico. Son las que buscan la historia clínica personal de un paciente determinado en el lugar donde se archivan estas en caso de que no este coge una nueva y se la lleva al medio para llenarla.

3.7 Casos de Uso del Negocio:

Módulo Laboratorio Clínico

1. Indicar un EC: Un Personal Médico le indica un examen complementario a un paciente durante una consulta.
2. Realizar Examen complementario: Un técnico del laboratorio clínico le realiza el EC a un paciente.
3. Obtener reportes: El técnico del laboratorio le brinda los reportes diarios ó mensuales a estadística.

Módulo Historia Clínica Personal

1. Solicitar consulta: le permite al paciente solicitar una consulta en su consultorio de atención, lo cual conlleva una búsqueda de su historia clínica, en caso de que la tenga se revisa, y si no se le llena una nueva

con sus datos específicos. Después de la consulta se le modifica, pues el médico le introduce una hoja de consulta nueva.

El Diagrama de casos de uso del negocio del Módulo Laboratorio Clínico puede verse en el anexo 1 y del Módulo Historias Clínicas Personales en el anexo 2.

3.8 Especificación de los casos de uso del negocio.

Este proceso incluye una fase de modelado del negocio, que describe los procesos del negocio de la organización bajo estudio de manera que se puedan construir, de forma sencilla y directa, versiones iniciales de los modelos conceptuales y de casos de uso. Cada proceso del negocio se describe haciendo uso de un diagrama de actividades UML con calles (*swimlanes*). Posteriormente, se identifican los casos de uso del sistema a partir de las actividades y los *conceptos* (clases del dominio) a partir de los datos (objetos de información que fluyen entre las actividades) [4].

La especificación de los casos de uso del negocio con sus diagramas de actividades correspondientes al Módulo Laboratorio Clínico puede verse en el anexo 3 y al Módulo Historias Clínicas Personales en el anexo 4.

3.9 Diagrama de clases del modelo de objeto del negocio.

El diagrama de clases, como artefacto que se construye para describir el modelo de objetos del negocio, muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos. Aunque se puede construir un único diagrama, se recomienda confeccionarlo para cada caso de uso de negocio para una mejor claridad.

Como todo diagrama de clases, se pueden representar, además de la asociación, los distintos tipos de relaciones entre las entidades de negocio (agregación, composición y generalización / especialización), la cardinalidad y navegabilidad de las relaciones, pero para efectos de su utilización posterior es suficiente con mostrar la relación entre los trabajadores y estos con las entidades.

El diagrama de clases del modelo de objeto de negocio del Módulo Laboratorio Clínico puede verse en el anexo 5 y del Módulo Historias Clínicas Personales en el anexo 6.

3.10 Especificación de los requisitos de software del sistema.

El análisis de requisitos permite al ingeniero de sistemas especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software. El análisis de requisitos permite al ingeniero del software (a menudo llamado analista en esta función) refinar la definición del software y construir los modelos de los dominios de datos, funcional y de comportamiento que van a ser tratados por el software. El análisis de requisitos proporciona modelos al diseñador del software que pueden traducirse en el diseño de datos, arquitectónico y de interfaz. Finalmente, la especificación de requisitos proporciona al diseñador y al cliente los medios para valorar la calidad una vez que se ha construido el software.

3.11 Requerimientos funcionales.

Los requerimientos funcionales no son más que la determinación clara y concisa de qué debe ser capaz de hacer el sistema, éstas se corresponden con opciones que ejecutará el software, operaciones realizadas de forma oculta o condiciones extremas a determinar por el sistema.

Módulo Laboratorio Clínico

1. Tener guardados todos los EC realizados en el laboratorio Clínico del policlínico clínico de la UCI.
2. Tener todos los tipos de exámenes disponibles en el laboratorio.
3. Hacer la solicitud para indicarle un EC a un paciente.
4. Agregar un EC disponible en el laboratorio.
5. Registrar el número de la muestra.
6. Modificar los datos de un examen disponible en el laboratorio.

7. Agregar el resultado de un EC realizado.
8. Mostrar al médico el resultado de un EC realizado.
9. El sistema debe proporcionar un listado de los EC realizados en un día.
10. El sistema debe proporcionar un listado de los EC por un médico.
11. El sistema debe proporcionar un listado con un análisis de positividad de cada médico.

Módulo Historia Clínica Personal.

1. Mostrar la historia clínica personal de cada estudiante, trabajador y cualquier persona que tenga que ver con este centro y dentro de esta todos los datos que se le adjuntan.
2. Crear nuevas historias clínica personal y todos los datos que la componen.
3. Modificar datos en las historias clínicas personales y solo algunos de los datos que esta presenta.

3.12 Requerimientos no funcionales

3.12.1 Seguridad

El sistema se acogerá a las condiciones de seguridad del Sistema Integral de los servicios médicos.

Realizar el tratamiento de errores para todas las acciones que se realizan en el sistema.

3.12.2 Interfaces con otros sistemas

El sistema debe ser capaz de estar integrado a los demás módulos de los servicios médicos.

3.12.3 Requisitos suplementarios

El sistema deberá restringir el acceso a las distintas funcionalidades de acuerdo con los permisos definidos para cada usuario o grupo.

3.12.4. Requisitos de performance

Tiempo de respuesta rápido. El sistema debe tener un tiempo de respuesta rápido ante cualquier solicitud del usuario.

3.13 Definición de los actores del sistema.

Módulo Laboratorio Clínico

Actores	Justificación
Médicos	Es una de las personas que interactúa con el sistema. Solicita la indicación de un EC y necesita le acceder al resultado de un EC (puede ser un Médico MGI, un estomatólogo, un médico ITS ó un médico del cuerpo de guardia).
Técnico Laboratorio	Es la persona que realiza los exámenes y lleva el control de estos en el laboratorio.
Estadística	Es quien obtiene todos los reportes que se generan en el laboratorio.

Módulo Historia Clínica Personal.

Actores	Justificación
Personal médico autorizado	Este grupo encierra todos los médicos, especialistas y enfermeras autorizados a revisar los datos de las historias clínicas.
Médicos MGI	En este grupo estas los médicos de los consultorios de cada área de atención, así como los médicos de la consulta MGI del policlínico.

3.14 Listado de casos de uso

Módulo Laboratorio Clínico

CU-1	Indicar examen complementario.

Actores	Médico
Descripción: El médico durante la consulta pueden indicar un examen complementario que necesita realizarle al paciente.	
Referencias	Requerimientos funcionales: 3.

CU-2	Obtener los resultados de exámenes complementarios.
Actores	Médico
Descripción: El médico durante la consulta necesita saber cual fue el resultado de un examen que le fue indicado al paciente en consultas anteriores.	
Referencias	Requerimientos funcionales: 8.

CU-3	Obtener reportes.
Actores	Estadística
Descripción: La persona que trabaja en estadística obtiene diariamente los reportes que realiza el laboratorio clínico.	
Referencias	Requerimientos funcionales: 9, 10,11.

CU-4	Definir número de muestra.
Actores	Técnico
Descripción: El técnico del laboratorio clínico debe registrar cual es el número de la muestra que le fue tomada al paciente para la realización del EC.	

Referencias	Requerimientos funcionales: 5.

CU-5	Realizar Examen complementario.
Actores	Técnico
Descripción: El técnico del laboratorio inserta en el sistema el resultado y la descripción del EC realizado.	
Referencias	Requerimientos funcionales: 1,7.

CU-6	Adicionar Examen complementario disponible.
Actores	Técnico
Descripción: El técnico del laboratorio adiciona un examen disponible, este se va poder realizar en el laboratorio.	
Referencias	Requerimientos funcionales: 2, 3

CU-7	Modificar Examen complementario disponible.
Actores	Técnico
Descripción: El técnico del laboratorio modifica un examen disponible, este se va poder realizar o no en el laboratorio.	
Referencias	Requerimientos funcionales: 4.

Módulo Historia Clínica Personal.

CU-1	Gestionar Historia clínica personal
Actores	Médicos MGI
Descripción: Algunos de los médicos MGI le solicita al sistema, gestionar la historia clínica asociada a determinado paciente. Eso incluye una modificación o una creación de esta.	
Referencias	Requerimientos funcionales: 2, 3.

CU-2	Mostrar Historia clínica personal
Actores	Médicos MGI y Personal médico autorizado
Descripción: Algunos de los médicos MGI o del personal médico autorizado a revisar estos datos le solicita al sistema, que le muestre la historia clínica asociada a determinado paciente. Eso incluye una búsqueda de los datos de esta para visualizarlos.	
Referencias	Requerimientos funcionales: 1.

CU-3	Gestionar Domicilio
Actores	Médicos MGI
Descripción: Algunos de los médicos MGI le solicita al sistema, gestionar los domicilios de una historia clínica personal asociada a determinado paciente. Eso incluye una modificación o una creación de estos.	
Referencias	Requerimientos funcionales: 2, 3.

CU-4	Gestionar Antecedentes patológicos familiares
Actores	Médicos MGI

Descripción: Algunos de los médicos MGI le solicita al sistema, gestionar los antecedentes patológicos familiares de una historia clínica personal asociada a determinado paciente. Eso incluye una modificación o una creación de estos.	
Referencias	Requerimientos funcionales: 2, 3.

CU-5	Gestionar Antecedentes patológicos personales
Actores	Médicos MGI
Descripción: Algunos de los médicos MGI le solicita al sistema, gestionar los antecedentes patológicos personales de una historia clínica personal asociada a determinado paciente. Eso incluye una modificación o una creación de estos datos.	
Referencias	Requerimientos funcionales: 2, 3.

CU-6	Gestionar Factores de riesgo para mayores de quince años
Actores	Médicos MGI
Descripción: Algunos de los médicos MGI le solicita al sistema, gestionar los factores de riesgo para mayores de quince años de una historia clínica personal asociada a determinado paciente. Eso incluye una modificación o una creación de estos datos.	
Referencias	Requerimientos funcionales: 2, 3.

CU-7	Gestionar Menor de un año
Actores	Sistemas médicos
Descripción: Algunos de los médicos MGI le solicita al sistema, gestionar los datos para menor de un año de una historia clínica personal asociada a determinado paciente. Eso incluye una modificación o una creación de estos datos.	
Referencias	Requerimientos funcionales: 2, 3.

CU-8	Gestionar Esquema de vacunación
Actores	Sistemas médicos
Descripción: Algunos de los médicos MGI le solicita al sistema, gestionar los datos para el esquema de vacunación de una historia clínica personal asociada a determinado paciente. Eso incluye una modificación o una creación de estos datos.	
Referencias	Requerimientos funcionales: 2, 3.

3.15 Diagrama de casos de uso del sistema.

Un diagrama de Casos de Uso muestra las distintas operaciones que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

Los diagramas de casos de uso del sistema del Módulo Laboratorio Clínico pueden verse en el anexo 7 y del Módulo Historias Clínicas Personales en el anexo 8.

3.16 Casos de uso por ciclo.

Los casos de uso por ciclo del Módulo Laboratorio Clínico pueden verse en el anexo 9 y del Módulo Historias Clínicas Personales en el anexo 10.

3.17 Casos de uso expandidos.

Se expandieron los casos de uso del primer ciclo de desarrollo tal y como se muestra en el anexo 11 los del Módulo Laboratorio Clínico y en el anexo 12 los del Módulo Historias Clínicas Personales.

Capítulo 4

Análisis y diseño del sistema

4.1 Análisis

4.1.1 Modelo conceptual de clases de análisis.

El modelo conceptual se realizó utilizando el diagrama de clases de la natación UML de forma simplificada. Se utilizan las clases preliminares, las asociaciones preliminares entre estas, la multiplicidad o cardinalidad para cada asociación y los nombres para las clases y las asociaciones.

El modelo conceptual de clases del análisis se muestra en el anexo 13 para el Módulo Laboratorio Clínico y en el anexo 14 para el Módulo Historias Clínicas Personales.

4.2 Diseño

4.2.1 Diagramas de Interacción.

Los diagramas de interacción no son más que una descripción del modo en el que cada operación detectada en los diagramas de secuencia lleva a cabo sus responsabilidades y modifica el estado del sistema. En UML los diagramas de interacción pueden representarse a través de los Diagramas de Colaboración y/o de los Diagramas de Secuencia.

El tipo de diagrama seleccionado para construir los diagramas de interacción fue el de Secuencia, debido a que muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente.

Los diagramas de interacción definidos se muestran en el anexo 15 para el Módulo Laboratorio Clínico y en el anexo 16 para el Módulo Historias Clínicas Personales.

4.2.2 Diagrama de diseño Web del sistema.

El diagrama del diseño Web del sistema (diagrama de clases para diseño orientado a objetos) se obtiene como resultado del refinamiento del modelo conceptual y se basa fundamentalmente en los diagramas de interacción.
El diagrama de clases del sistema se puede ver en el anexo 17 para el Módulo Laboratorio Clínico y en el anexo 18 para el Módulo Historias Clínicas Personales. Está dividido en varios diagramas para que sea más fácil su comprensión.

4.2.3 Descripción de las clases.

Para facilitar el análisis, se clasifican las clases en:

- **Clase de control.**

  - Representan coordinación, secuencia, transacciones, y control de otros objetos.
  - Se usan mucho para encapsular el control de un caso de uso concreto.
  - También derivaciones y cálculos complejos.
  - Manejan y coordinan las acciones y los flujos de control principales, y delegan trabajo a otros objetos (de interfaz y de entidad).

  Los detalles de las clases de control se pueden ver en el anexo 19 para el Módulo Laboratorio Clínico y en el anexo 20 para el Módulo Historias Clínicas Personales.

- **Clase de entidad.**

  - Modelan información que posee una larga vida.

- Modelan la información y el comportamiento asociado a algún fenómeno o concepto, como una persona, un objeto del mundo real, o un suceso del mundo real.
- Derivan normalmente de una clase de entidad del negocio.
- Puede tener comportamiento complejo.
- Aísla los cambios en la información que representa.

Los detalles de las clases de entidad se pueden ver en el anexo 21 para el Módulo Laboratorio Clínico y en el anexo 22 para el Módulo Historias Clínicas Personales.

Clase de interfaz.

- Modelan la interacción entre el sistema y sus actores (usuarios y sistemas externos).
- Reciben y presentan información y peticiones de y hacia los actores.
- Reúnen los requisitos en los límites del sistema.
- Suelen ser abstracciones de ventanas, formularios, interfaces de impresoras, censores, terminales...
- Describan lo que se obtiene con la interacción, no el proceso físico de cómo se ejecuta.
- Una clase de interfaz por cada actor, y viceversa.

Los detalles de las clases de interfaz se pueden ver en el anexo 23 para el Módulo Laboratorio Clínico y en el anexo 24 para el Módulo Historias Clínicas Personales.

4.2.4 Diseño de la BD del Sistema.

4.2.4.1 Diagrama Entidad Relación de la BD.

El diseño de la base de datos del sistema puede verse en el anexo 25 para el Módulo Laboratorio Clínico y en el anexo 26 para el Módulo Historias Clínicas Personales.

4.2.4.2 Descripción de las tablas.

El gestor de la base de datos a utilizar es estructurado por tanto es necesario llevar las clases a tablas, se partió para esto de definir las clases persistentes. Las clases simples se representaron como tablas y en las relaciones de 1.. n se añadió una llave extranjera al extremo m, que se corresponde con la llave de la clase del extremo 1. En el caso de las relaciones n..m se crea una nueva tabla que tiene como llave las clases que conforman la relación.

En el Anexo 27 para el Módulo Laboratorio Clínico y en el anexo 28 para el Módulo Historias Clínicas Personales.

4.3 Seguridad del sistema.

Los datos que manipulará el sistema son de alta confiabilidad, es decir, se debe prever cómo lograr que dicha información se almacene y se trasmita de forma eficiente y segura.

Toda transacción segura por la red debe contemplar los aspectos de autenticidad, integridad y confidencialidad. Son varios los sistemas y tecnologías que se han desarrollado para intentar implementar estos aspectos en las transacciones electrónicas, siendo sin duda SSL el más conocido y usado en la actualidad. SSL permite la confidencialidad y la autentificación en las transacciones por Internet, siendo usado principalmente en aquellas transacciones en la que se intercambian datos sensibles, como números de tarjetas de crédito o contraseñas de acceso a sistemas privados.

SSL es una de las formas base para la implementación de soluciones PKI (Infraestructuras de Claves Públicas). Secure Socket Layer es un sistema de protocolos de carácter general diseñado en 1994 por la empresa Netscape Communications Corporation y está basado en la aplicación conjunta de criptografía simétrica, criptografía asimétrica (de llaves públicas), certificados digitales y firmas digitales para conseguir un canal o medio seguro de comunicación a través de Internet.

De los sistemas criptográficos simétricos, motor principal de la encriptación de datos transferidos en la comunicación, se aprovecha la rapidez de operación, mientras que los sistemas asimétricos se usan para el intercambio seguro de las claves simétricas, consiguiendo con ello resolver el problema de la confidencialidad en la transmisión de datos.

Desde el punto de vista de su implementación en los modelos de referencia OSI y TCP/IP, SSL se introduce como una especie de nivel o capa adicional, situada entre la capa de Aplicación y la capa de Transporte, sustituyendo los sockets del sistema operativo, lo que hace que sea independiente de la aplicación que lo utilice, y se implementa generalmente en el puerto 443. (NOTA: Los puertos son las interfaces que hay entre las aplicaciones y los protocolos TCP/IP del sistema operativo).

SSL proporciona servicios de seguridad a la pila de protocolos, encriptando los datos salientes de la capa de Aplicación antes de que estos sean segmentados en la capa de Transporte y encapsulados y enviados por las capas inferiores. Es más, también puede aplicar algoritmos de compresión a los datos a enviar y fragmentar los bloques de tamaño mayor a 2^{14} bytes, volviéndolos a reensamblar en el receptor.

Por todas estas razones toda la información que se envié por la red, en el sistema, viajará encriptada a través del protocolo de transmisión de datos SSL.

Conclusiones.

Después de haber analizado los temas anteriormente expuestos se llegó a la conclusión siguiente: el uso inteligente de las herramientas seleccionadas sin dudas proporcionará el éxito de los objetivos propuestos. El dominio del contenido relacionado con el tema de las plataformas seleccionadas, garantizará en gran medida la calidad en los resultados, resumido esto en complacer a cabalidad las necesidades del cliente así como las de los usuarios que trabajarán con el sistema.

Conclusiones

Con la realización del proyecto se arribaron a las siguientes conclusiones:

- Ambos módulos fueron adicionados al sitio web de los servicios médicos de la UCI.
- Con lo realizado se cumple con el objetivo propuesto: realizar el análisis y diseño de los módulos Laboratorio Clínico e Historias clínicas personales, con aplicación concreta en los servicios médicos de la UCI.
- Se adquirieron habilidades de programación en Java mediante la implementación de Servlets y páginas JSP, a la hora de trabajar con el gestor de bases de datos MySql, en el trabajo con el servidor de aplicaciones TomCat, así como también se ampliaron los conocimientos de Rup y UML.
- Se realizó un analisis sobre evaluación del desempeño en aplicaciones web con el objetivo de realizar la evaluación del sistema una vez terminado.

Recomendaciones

Se recomienda:

- Terminar cada uno de los módulos que componen el sistema médico de la UCI.
- Refinar los colores y estilos de cada una de nuestras interfaces con el usuario, para mejorar su apariencia.
- Continuar con la investigación para aumentar las funcionalidades del sistema, con el objetivo de obtener nuevas mejoras en futuras versiones del mismo.
- Promover la designación de un administrador para que mantenga la actualización y mantenimiento del Sitio.
- Iniciar los trabajos de seguridad informática del Sitio encaminados a construir herramientas específicas sobre la Web para administrar mejor el mismo, así como la base de datos que lo conforma.
- Realizar módulos para lograr el intercambio y procesamiento de la información con herramientas como el XML.
- Continuar desarrollando las potencialidades, capacidades y atractivos del Sitio.
- Realizar el estudio de Firmas Digitales para la incorporación al Sitio.
- Desarrollar con potencialidad una ayuda para el mejor trabajo en el Sistema.
- Desarrollar un manual de usuario para el previo estudio del Sistema antes de antes del comienzo del trabajo con el mismo.

Referencias Bibliográficas

[1] Bemmel JH Van. Medical Informatics. Art or Science? Meth Inform Me 1996; 35: 157-172.

[2] Waegemann CP. When will complete medical record systems exist? J. of Health Management Technology 1996 Feb v17 n2 p10 (1).

[3] Toward An Electronic Patient Record Newsletter. What Is An Electronic Patient Record? 1995 Medical Records Institute (Internet).

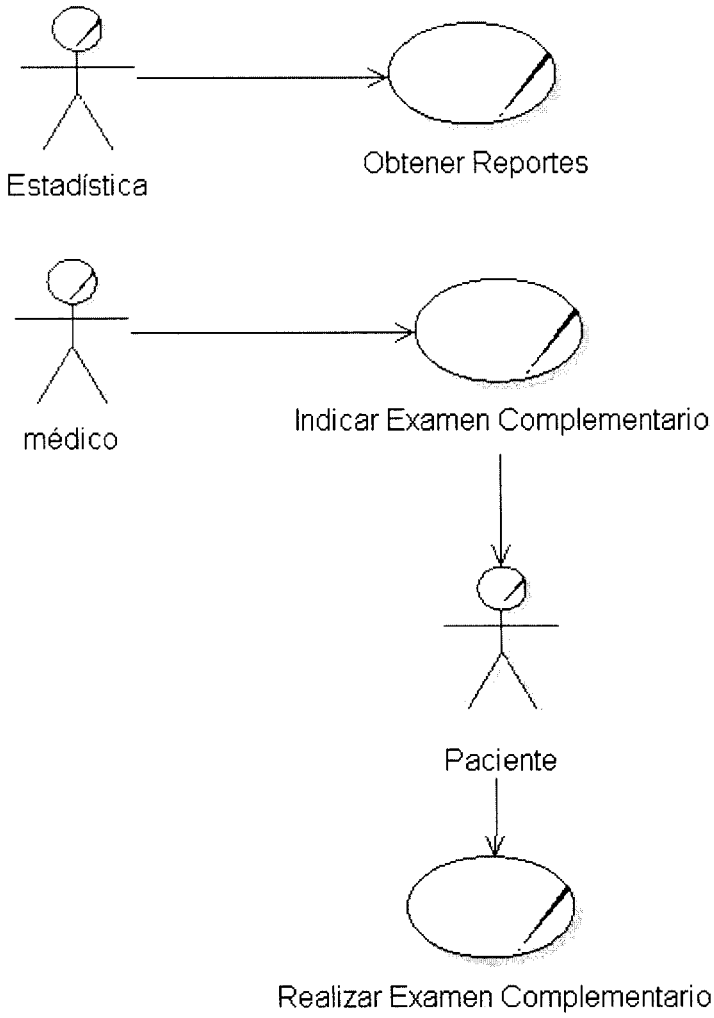
[4] Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modelling Language User Guide. Addison-
Wesley (1999)

Bibliografía

1. Pushtotest - Free open-source software test automation solutions
   http://www.pushtotest.com/ (15/06/04)
2. Guía de Axis
   http://www.osmosislatina.com/axis/index.htm (24/05/04)
3. Sun Microsystems
   http://www.sun.com/ (22/06/04)
4. The Java Tutorial
   http://java.sun.com/docs/books/tutorial/ (15/04/04)
5. SQL Tutorial
   http://www.w3schools.com/sql/default.asp (18/04/04)
6. Introducción al UML
   http://www.yoprogramo.com/articulo4.php (24/05/04)
7. Java Servlet Technology
   http://java.sun.com/products/servlet/ (22/06/04)
8. A Tutorial on Java Servlets and Java Server Pages (JSP)
   http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/ (22/06/04)
9. E-Test Suite Upgrade Tackles Web Services
   http://www.eweek.com/article2/0,1759,1607658,00.asp (22/06/04)
10. News Article: Empirix e-Test Suite 6.0
http://www.empirix.com/Empirix/Corporate/News+and+Info/News+Articles/Web+Application/empirix+e-test+suite+6.0+(crn+product+review).html (22/06/04)
11. Products - Mercury LoadRunner for Automated Load Testing
    http://www.mercury.com/us/products/performance-center/loadrunner/
(22/04/06)
12. TPC-W
    http://www.tpc.org/tpcw/default.asp (22/06/04)
13. C. Larman, UML y patrones, introducción al análisis y diseño orientado a objetos, Segunda edición, Prentice-Hall, 2002.
14. Glosario de términos
    http://www.smdata.com/glosario.htm (24/06/04)

Anexos

Anexo 1. Diagrama de casos de uso del negocio (MLC).

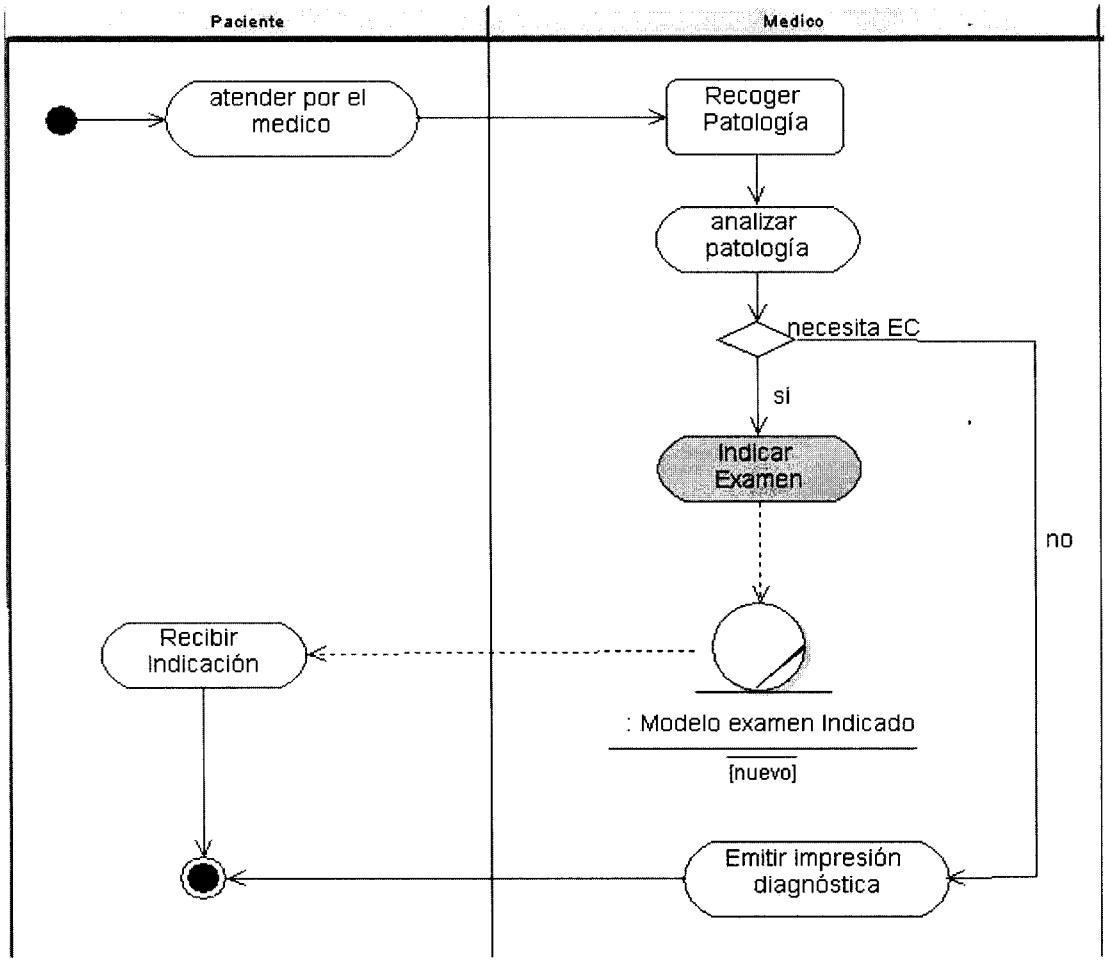


Anexo 2. Diagrama de casos de uso del negocio (MHCP).

Anexo 3. Especificación de los casos de uso del negocio (MLC).

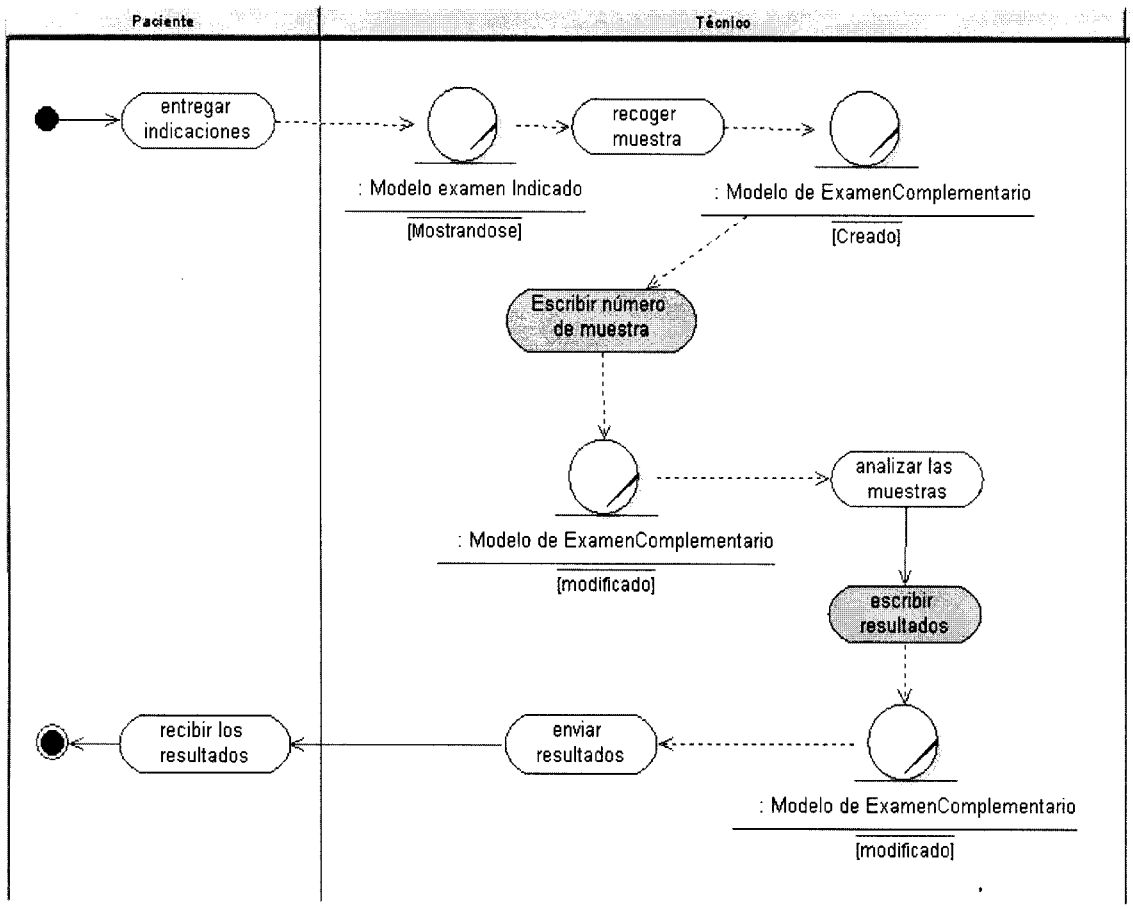
<table>
<tr><td>Nombre del caso de uso del negocio:</td><td colspan="2">Indicar un examen complementario.</td></tr>
<tr><td>Actores del negocio:</td><td colspan="2">Paciente y Médico.</td></tr>
<tr><td>Propósito:</td><td colspan="2">El objetivo de este proceso es indicarle a un paciente la realización de un examen complementario.</td></tr>
<tr><td>Casos de uso asociados:</td><td colspan="2">-</td></tr>
<tr><td colspan="3">Resumen:
El proceso comienza cuando un paciente llega a una consulta de un médico MGI, de un estomatólogo, ó un médico ITS, necesita indicarle la realización de un examen complementario teniendo en cuenta la patología del paciente para poder emitir su impresión diagnostica.</td></tr>
<tr><td colspan="3">Flujo de trabajo</td></tr>
<tr><td>Acción del actor</td><td colspan="2">Respuesta del negocio</td></tr>
<tr><td>1. El paciente comienza a ser atendido por el médico.
4. El paciente recibe la indicación del examen.</td><td colspan="2">2. El médico le recoge la patología del paciente.
3. Si para la emisión de una impresión diagnostica el médico necesita tener el resultado de un examen complementario, este le escribe la indicación del complementario para que el paciente se dirija al laboratorio clínico y se le realice dicho examen
3.1 Si el médico no necesita la realización de un examen complementario este emite su impresión diagnostica.</td></tr>
<tr><td>Prioridad:</td><td colspan="2">Para que se le pueda realizar un EC a un paciente es necesario que este haya sido indicado anteriormente.</td></tr>
</table>

Mejoras:	La automatización de este proceso le permitirá al médico indicarle un examen a un paciente a través de una computadora y esta solicitud será recogida automáticamente en el laboratorio.
Cursos alternos: médico necesariamente no tiene porque verse en la necesidad de indicarle un EC para obtener una impresión diagnostica.	



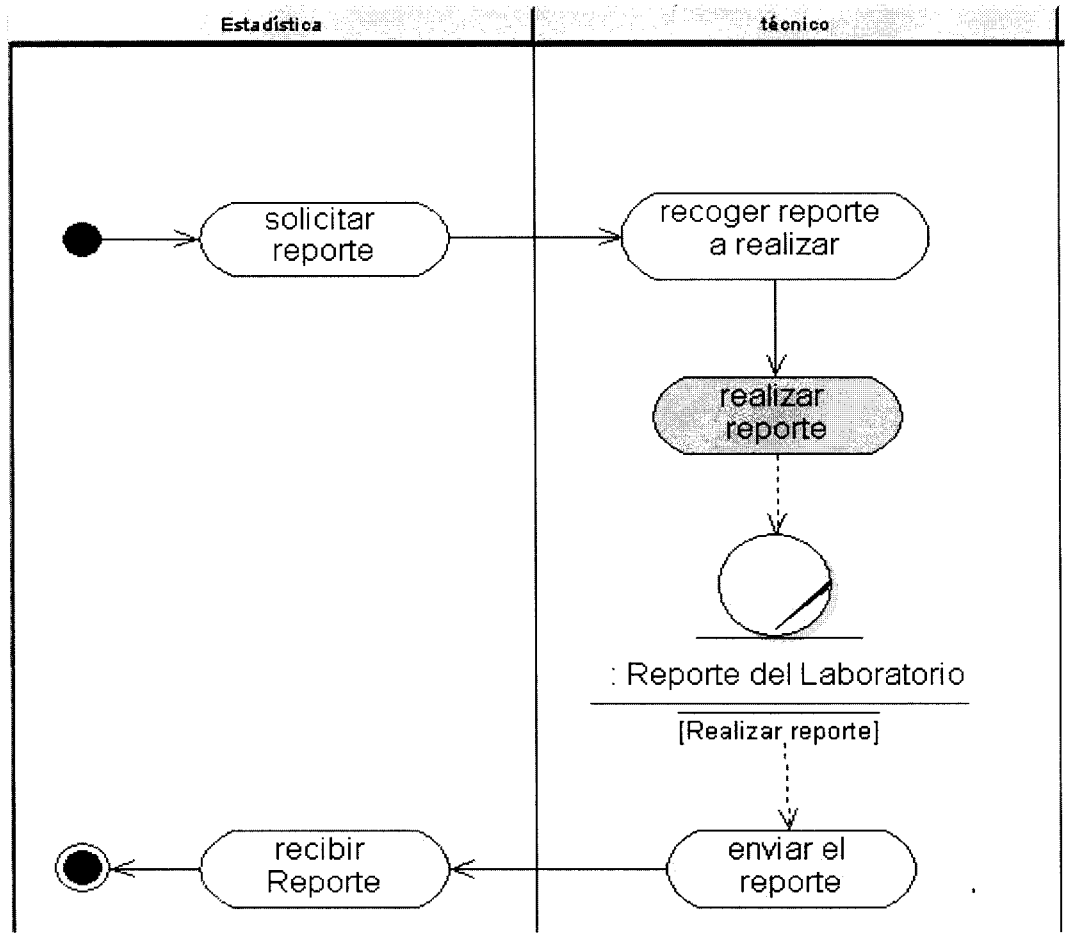
Nombre del caso de uso del negocio:	Realizar un examen complementario.

Actores del negocio:	Paciente
Propósito:	El objetivo de este proceso es llevar a cabo la realización de un EC por parte del técnico de laboratorio
Casos de uso asociados:	-
Resumen: El proceso comienza cuando un paciente llega al laboratorio clínico con la indicación de uno o varios EC, el técnico le recoge una muestra para ser analizada y a partir de este análisis saber si el EC es positivo o negativo.	
Flujo de trabajo	
Acción del actor	**Respuesta del negocio**
1. El paciente entrega la indicación de EC. 6. El paciente va a recibir los resultados para que posteriormente se los pueda mostrar al médico en la próxima consulta	2. El técnico le recoge una muestra en un frasco. 3. Escribe el número de la muestras en la indicación. 4. Las muestras son analizadas 5. A partir del análisis realizado a las muestras el técnico escribe el resultado en la indicación.
Prioridad:	-
Mejoras:	La automatización de este proceso le permitirá al técnico introducir el número de la muestra para un EC en el sistema, y cuando vaya a registrar el resultado del EC busca el examen para la muestra analizada y registra el resultado de este en el sistema.
Cursos alternos:	



Nombre del caso de uso del negocio:	Obtener reportes.
Actores del negocio:	Estadística
Propósito:	El objetivo de este proceso es realizar distintos reportes que tiene que debe de entregar a estadística.
Casos de uso asociados:	-

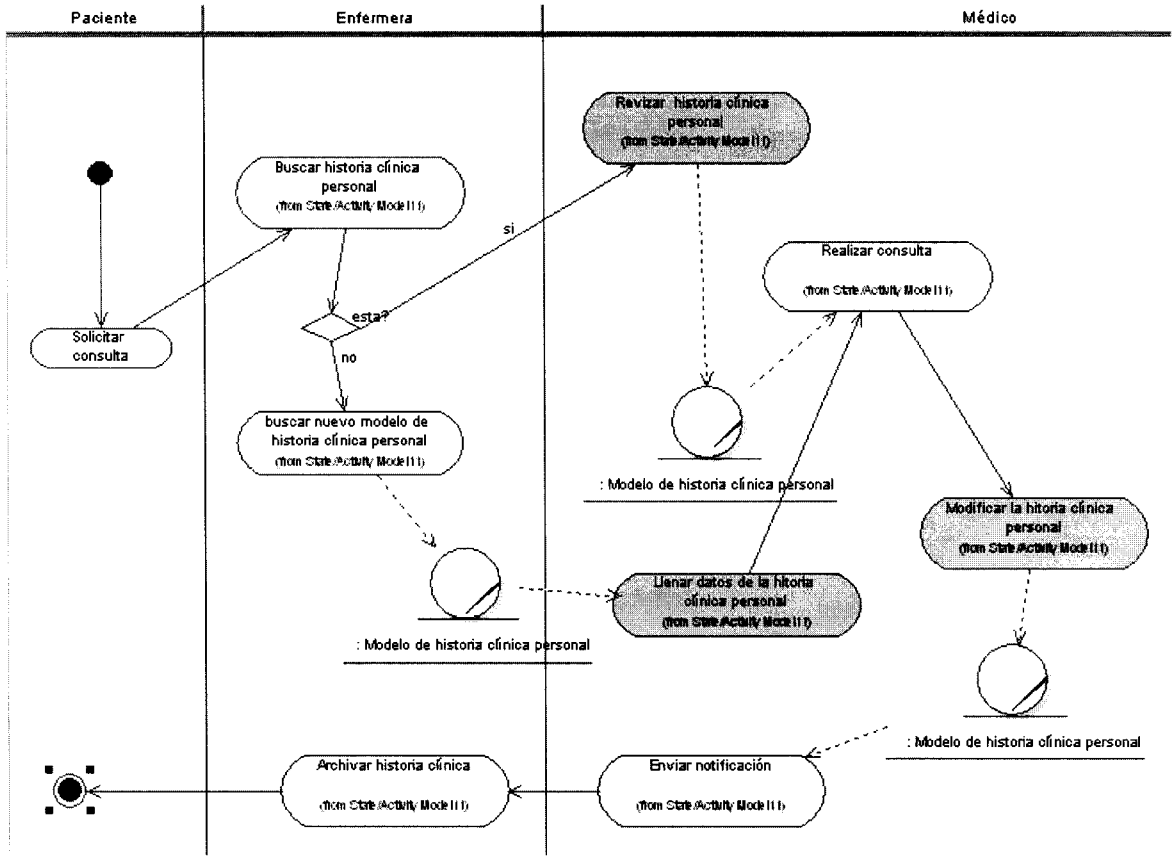
Resumen:	
El proceso consiste en la realización de un reporte cuando es solicitado al laboratorio clínico.	
Flujo de trabajo	
Acción del actor	**Respuesta del negocio**
1. Estadística realiza la solicitud de un informe. 5. Estadística recibe el reporte.	2. El técnico obtiene los datos de la solicitud del reporte. 3. Realiza el reporte teniendo en cuenta la información que necesita estadística. 4. El reporte es enviado en papel ó por correo electrónico.
Prioridad:	-
Mejoras:	La automatización de este proceso le permitirá al técnico no tener que realizar los reportes que necesita estadística, el sistema por si solo será capaz de brindarle todos estos reportes.
Cursos alternos:	



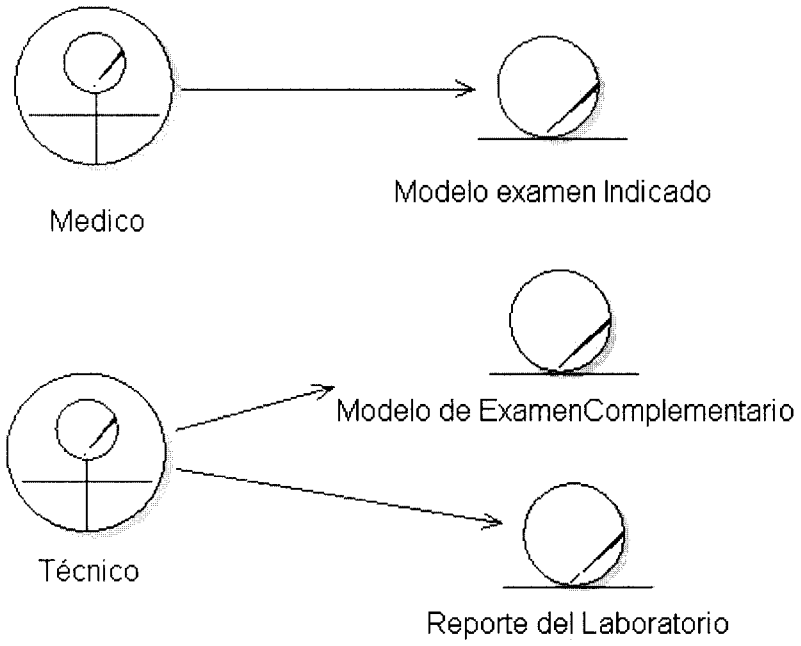
Anexo 4. Especificación de los casos de uso del negocio (MHCP).

Nombre del caso de uso del negocio:	Solicitar consulta
Actores del negocio:	Paciente
Propósito:	El objetivo de este proceso es posibilitarle al paciente una consulta con el médico.
Casos de uso asociados:	-
Resumen: El proceso comienza cuando un paciente se dirige al consultorio médico de su área de atención o a la consulta MGI del policlínico en otros casos y solicita una consulta a la enfermera del consultorio.	

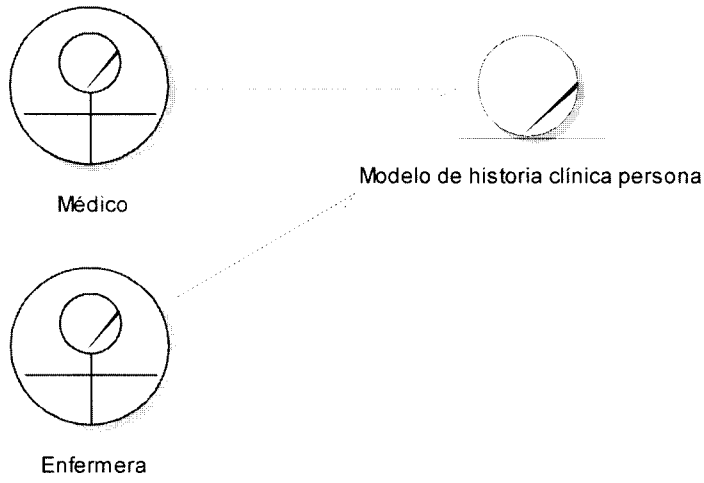
Flujo de trabajo	
Acción del actor	**Respuesta del negocio**
1. El paciente se dirige al consultorio y solicitar una consulta. 2. El paciente entra en la consulta.	1. La enfermera le hace espera hasta el momento de la consulta, le busca su historia clínica personal en un estante donde se almacenan estas en su área de atención. 2. Se le entrega la historia clínica personal del paciente en caso de que la tenga, para revisar los datos que esta tiene, sino se le entrega un modelo en blanco para que el médico introduzca los datos del paciente.
Prioridad:	
Mejoras:	Automatizar este proceso de archivar, buscar y realizarle modificaciones, además de hacer centralizado este proceso.
Cursos alternos:	



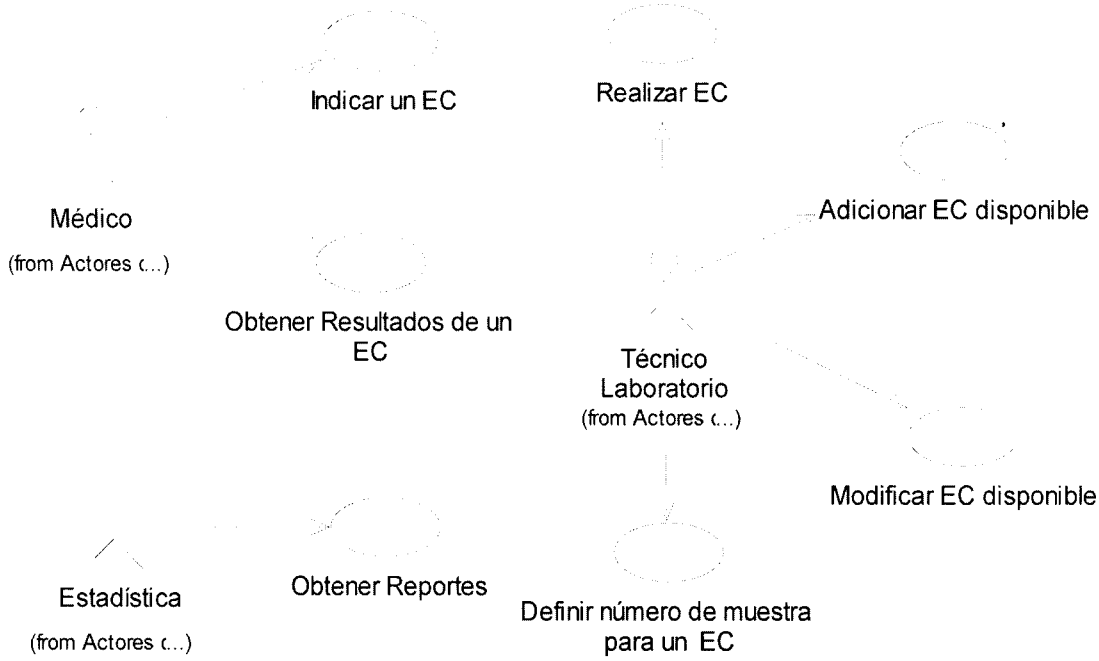
Anexo 5. Diagrama de clases del modelo de objeto del negocio (MLC).



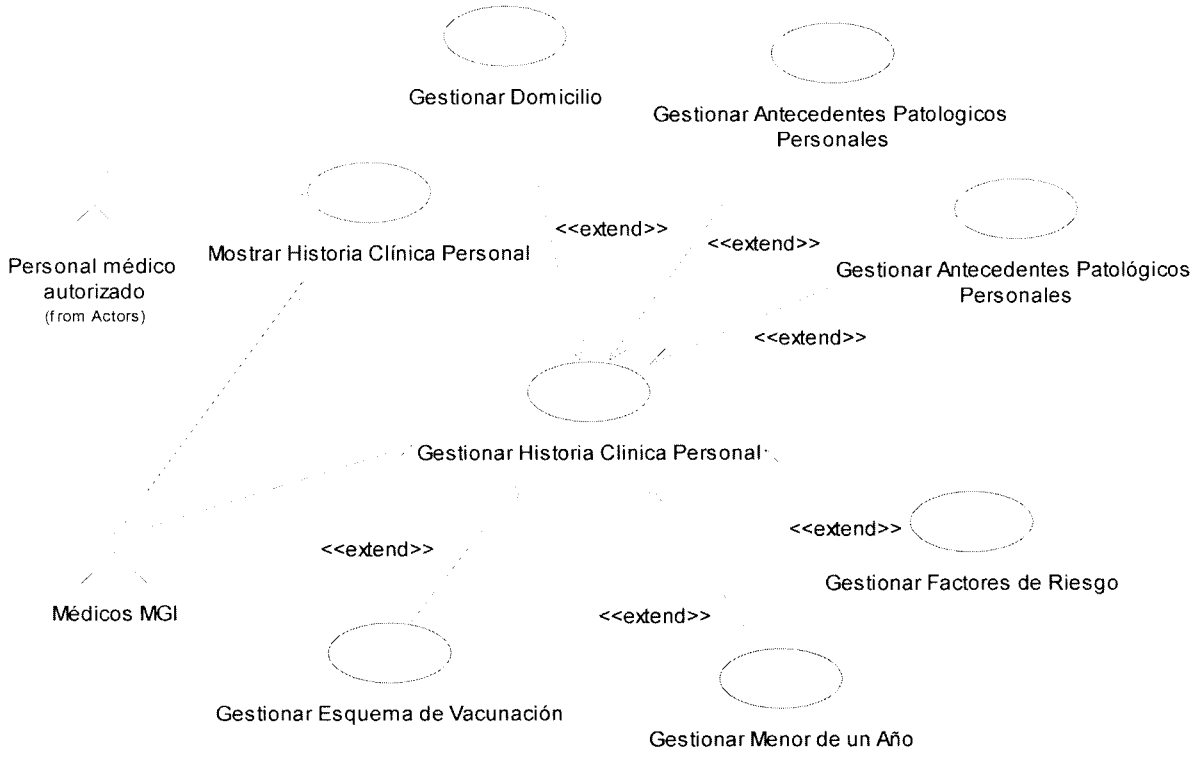
Anexo 6. Diagrama de clases del modelo de objeto del negocio (MHCP).



Anexo 7. Diagrama de casos de uso (MLC).



Anexo 8. Diagrama de casos de uso (MHCP).



Anexo 9. Casos de uso por ciclo (MLC).

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
1	• Indicar examen complementario • Obtener los resultados de exámenes complementarios. • Definir número de la muestra.		Este caso de uso le brinda la posibilidad al médico de indicar un EC. Este caso de uso es necesario porque le permite al médico obtener el resultado de un EC que le ha sido al paciente que esta atendiendo. Este caso de uso es necesario porque le permite técnico obtener el número de la muestra que se le ha obtenido al

	• Realizar Examen complementario • Gestionar Examen complementario disponible.		paciente. Este caso de uso es necesario porque el técnico registra en el sistema el resultado del EC. Este caso de uso es necesario porque le permite al técnico adicionar o modificar un nuevo EC que el laboratorio a partir de esteva a tener la posibilidad de realizar.
2	• Obtener reportes.		Este caso de uso es necesario porque le permite a estadística obtener los reportes que se generan en el laboratorio.

Anexo 10. Casos de uso por ciclo (MHCP).

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
1	• Gestionar Historia clínica personal • Gestionar Domicilios • Gestionar Antecedente patológicos familiares onar antecedentes		Este caso de uso es necesario para la gestión de los otros casos de uso, pues es la primera cara del sistema para los casos de modificar y crear una nueva historia clínica personal. Este caso de uso es necesario para tener un registro de todos los domicilios por los que paso dicho paciente. Este caso de uso es necesario para tener un registro de todos los antecedentes patológicos familiares para tener un diagnostico de posibles enfermedades del paciente.

	patológicos personales.		Este caso de uso es necesario para tener un registro de todos los antecedentes patológicos personales anteriores para posibles padecimientos patológicos del paciente.
2	• Gestionar Esquema de vacunación. • Gestionar Factores de riesgo para mayores de quince años. • Mostrar Historia Clínica Personal		Este caso de uso es necesario para que el personal tenga una descripción de todas las vacunas que el paciente se puso. Este caso de uso es necesario porque le permite al personal médico de la UCI tener un registro de todos los factores de riesgo de cada paciente desde el momento que su edad sobrepasa los quince años. Este caso de uso es importante para el sistema ya que aquí se va a visualizar todos los datos de la historia clínica personal asociada a determinado paciente.

Anexo 11. Casos de uso expandidos (MLC).

CU-1	Indicar examen complementario.	
Propósito	Indicarle a un paciente la realización de un examen complementario.	
Actores	Médico.	
Resumen: El médico durante la realización de una consulta selecciona la opción de indicar examen complementario donde indica que examen es el que se le necesita realizar al paciente y si este es urgente ó electivo.		
Referencias	RF-3	
Acción del actor		**Respuesta del sistema**

1) El médico solicita indicar un EC.	2) El sistema envía al médico a la página de indicaron del EC. 3) El sistema en esta página le muestra los exámenes disponibles en ese momento.
4) El usuario llena los datos del formulario: nombre del EC y si es urgente o electivo.	5) El sistema verifica los datos son validos. 6) En caso afirmativo el sistema guarda en la base de datos la indicación del EC y le da la posibilidad al médico de indicarle otro EC. 7) Si este no desea indicar otro EC el sistema le brinda la posibilidad de seguir realizando la consulta.
Flujo alternativo	
Acción del actor	**Respuesta del sistema**
	5) El sistema le envía un mensaje de error por haber introducido un dato no valido y le muestra el formulario de indicaron del EC para que vuelva a introducir el EC a indicar y si este es urgente o electivo.
Puntos de extensión.	

CU-2	Obtener los resultados de exámenes complementarios.
Propósito	Saber si el EC que le fue indicado a un paciente dio positivo.
Actores	Médico.

Resumen: El médico durante la realización de una consulta selecciona la opción de ver resultado de un EC.	
Referencias	RF-8
Acción del actor	**Respuesta del sistema**
1) El médico solicita ver resultado de un examen.	2) El sistema busca todos los EC que le han sido realizados a este paciente y se los muestra al médico con su respectivo resultado. 3) El sistema le brinda la posibilidad al médico que pueda seguir realizando la consulta.
Flujo alternativo	
Acción del actor	**Respuesta del sistema**
Puntos de extensión.	

CU-3	Obtener reportes.
Propósito	Que la persona de estadística obtenga todos los reportes que son generados en el laboratorio clínico.
Actores	Estadística.
Resumen: La persona de estadística diariamente va acceder a los reportes acerca de los datos que se manejan en el laboratorio clínico que en este caso serian los EC realizados.	

Referencias	RF-9,10,11
Acción del actor	**Respuesta del sistema**
1) La persona de estadística selecciona el reporte que desea obtener.	2) El sistema realiza la consulta adecuada y le muestra los datos referentes al reporte. 3) El sistema le brinda la posibilidad a la persona de estadística la posibilidad de obtener otro reporte.
Flujo alternativo	
Acción del actor	**Respuesta del sistema**
Puntos de extensión.	

CU-4	Realizar Examen complementario.
Propósito	Que el resultado de un EC que ha sido realizado en el laboratorio sea registrado en el sistema.
Actores	Técnico.
Resumen: El técnico del laboratorio una vez que ha sido realizado el EC en el laboratorio clínico registra este resultado en el sistema.	
Referencias	RF-1,7
Acción del actor	**Respuesta del sistema**
1) El técnico selecciona la opción de	2) El sistema le brinda un formulario donde el

realizar examen.	técnico debe introducir la fecha de recepción del EC y el número de muestra de este EC.
3) El técnico debe llenar el formulario.	4) El sistema busca si existe algún examen con esos datos para registrarle el resultado en caso afirmativo le muestra al técnico un formulario donde este va a introducir el resultado del EC y una descripción de la realización del mismo. 5) El sistema le brinda la posibilidad de realizar cualquier otra actividad del laboratorio clínico.
Flujo alternativo	
Acción del actor	**Respuesta del sistema**
	6) El sistema si no encuentra ningún EC con los datos seleccionados por el técnico muestra un mensaje de error y le brinda la posibilidad de que entre los datos de nuevo.
Puntos de extensión.	

CU-5	Registrar el número de la muestra.
Propósito	Que cuando un paciente sea atendido en el laboratorio clínico y el técnico obtenga la muestra del EC la registre en el sistema.

Actores	Técnico.
Resumen: El técnico del laboratorio una vez que ha sido obtenido la muestra del paciente registra el numero que tiene dicha muestra en el sistema.	
Referencias	RF-5

Acción del actor	**Respuesta del sistema**
1) El técnico selecciona la opción de atender paciente.	2) El sistema le brinda un formulario donde el técnico debe introducir el número de solapin del paciente si es interno y si no lo es debe introducir el CI si es una persona externa del centro y selecciona buscar exámenes indicados. 3) Si existe algún examen indicado a este paciente entonces el técnico obtiene la muestra la numera y este numero lo introduce como un dato referente al EC. 4) El sistema le brinda la posibilidad de realizar otra acción del laboratorio clínico.
Flujo alternativo	
Acción del actor	**Respuesta del sistema**

	5) El sistema si no encuentra ningún EC indicado para los datos del paciente introducido por muestra un mensaje de error y le brinda la posibilidad de que entre los datos de nuevo pensando que hayan entrado datos equivocados.
Puntos de extensión.	

CU-6	Adicionar Examen Complementario disponible.	
Propósito	Que cuando el laboratorio clínico se vea con la capacidad de realización de nuevos EC los datos referentes a este sean introducidos el sistema.	
Actores	Técnico.	
Resumen: El técnico del laboratorio llena el formulario para adicionar un nuevo EC disponible.		
Referencias	RF-2,4	
Acción del actor		**Respuesta del sistema**
1) El técnico selecciona la opción de adicionar un nuevo EC disponible.		2) El sistema le brinda un formulario donde el técnico debe introducir el código, el nombre del EC, el área del laboratorio donde este se realiza, el intervalo de positividad y si este esta disponible en este momento. 3) Si los datos introducidos son validos entonces se inserta el EC disponible en la base de datos.

Flujo alternativo	
Acción del actor	**Respuesta del sistema**
	4) Si los datos no son validos el sistema muestra un mensaje de error y le permite al técnico realizar la acción nuevamente.
Puntos de extensión.	

CU-7	Modificar Examen Complementario disponible.	
Propósito	El técnico puede buscar los EC que están en la base de datos y modificar los que sean necesarios.	
Actores	Técnico.	
Resumen: El técnico del laboratorio llena el formulario para modificar un EC existente.		
Referencias	RF-5	
Acción del actor		**Respuesta del sistema**
1) El técnico selecciona la opción de mostrar EC disponibles.		2) El sistema le muestra los EC que aparecen en la Base de datos.
3) El técnico selecciona el EC que desea modificar.		4) El sistema brinda un formulario con los datos del EC que ha sido seleccionado el

	paciente, se pueden modificar los datos referente al EC y el le brinda la opción de modificar.
5) El técnico selecciona modificar.	6) Los datos son actualizados en la base de datos y le permite al técnico modificar otro EC.
Flujo alternativo	
Acción del actor	**Respuesta del sistema**
	4) Si los datos no son validos el sistema muestra un mensaje de error y le permite al técnico realizar la acción nuevamente.
Puntos de extensión.	

Anexo 12. Casos de uso expandidos (MHCP).

CU-1	**Gestionar Historia clínica personal**
Propósito	Realizar modificaciones o crear una historia clínica personal por cada paciente.
Actores	Médicos MGI.
Resumen: Médicos MGI le hacen una petición de cada una de las acciones que brinda el sistema.	
Referencias	RF- 2, 3
Acción del actor	**Respuesta del sistema**

1) Petición de modificar una historia clínica personal	1) El sistema chaquea si el paciente tiene historia clínica personal asociada, verificando los datos almacenados. 2) En caso afirmativo, el sistema muestra los datos con la opción de modificación de cada uno de ellos. 3) En caso contrario el sistema le da la opción de crearle una nueva.
2) Petición de crear una historia clínica personal.	1) El sistema chaquea si el paciente tiene historia clínica asociada verificando los datos almacenados. 2) En caso negativo el sistema muestra la caja de diálogo para que el actor introduzca los datos referentes al paciente. 3) En caso contrario el sistema envía un mensaje de ya tiene historia clínica personal asociada.
Flujo alternativo	
Puntos de extensión.	

CU-2	**Mostrar Historia clínica personal**
Propósito	Realizar la muestra de la historia clínica personal del paciente solicitado.
Actores	Médicos MGI y Personal médico autorizado.
Resumen: Cualquier médico MGI o algún personal médico autorizado hacen una petición de la acción que se brindan.	
Referencias	RF-1

Acción del actor	**Respuesta del sistema**
1) Petición de mostrar una historia clínica personal de un paciente determinado	1) El sistema chaquea si el paciente tiene historia clínica personal asociada, verificando los datos almacenados. 2) En caso afirmativo, el sistema le envía una muestra de estos datos. 3) En caso contrario el sistema le da la opción de crearle una nueva.
Flujo alternativo	
Puntos de extensión.	

CU-3	**Gestionar Domicilios**
Propósito	Realizar la creación o modificación de los datos de cada domicilio por el que trancito el paciente.
Actores	Médicos MGI.
Resumen: Cualquier médico MGI hace una petición de cada una de las acciones que se brindan.	
Referencias	RF-2, 3

Acción del actor	**Respuesta del sistema**
1) Petición de crear domicilios	1) El sistema muestra la caja de diálogo para que el actor introduzca los datos referentes a los domicilios anteriores del paciente.
2) Petición de modificar un domicilio.	1) El sistema chaquea si el paciente tiene datos almacenados referentes a los domicilios.

	2) En caso afirmativo el sistema muestra una caja de dialogo con los valores anteriores del domicilio y con opción de modificaron. 3) En caso contrario el sistema envía un mensaje de que no hay datos referentes a los domicilios.
Flujo alternativo.	
Puntos de extensión.	

CU-4	**Gestionar Antecedentes patológicos familiares**
Propósito	Realizar modificaciones o crear datos de cualquier antecedente patológico familiar del paciente.
Actores	Médicos MGI.
Resumen: Cualquier médico MGI hace una petición de cada una de las acciones que se brindan.	
Referencias	RF- 2, 3

Acción del actor	**Respuesta del sistema**
1) Petición de crear antecedentes patológicos familiares.	1) El sistema muestra la caja de diálogo para que el actor introduzca los datos referentes a los antecedentes patológicos familiares del paciente.
2) Petición de modificar un antecedente patológico familiar.	1) El sistema chaquea si el paciente tiene datos almacenados referentes a los antecedentes patológicos familiares. 2) En caso afirmativo el sistema muestra una

	caja de dialogo con los valores anteriores de un antecedente patológico familiar y con opción de modificaron. 3) En caso contrario el sistema envía un mensaje de que no hay datos referentes a los antecedentes patológicos familiares.
Flujo alternativo	
Puntos de extensión.	

CU-5	**Gestionar Antecedentes patológicos personales**
Propósito	Realizar modificaciones o crear datos de cualquier antecedente patológico personal del paciente.
Actores	Sistemas médicos.
Resumen: Cualquier médico MGI hace una petición de cada una de las acciones que se brindan.	
Referencias	RF- 2, 3

Acción del actor	**Respuesta del sistema**
1) Petición de crear antecedentes patológicos personales.	1) El sistema muestra la caja de diálogo para que el actor introduzca los datos referentes a los antecedentes patológicos personales del paciente.
2) Petición de modificar un antecedente patológico personal.	1) El sistema chaquea si el paciente tiene datos almacenados referentes a los antecedentes patológicos personales. 2) En caso afirmativo el sistema muestra una

	caja de dialogo con los valores anteriores de un antecedente patológico personal y con opción de modificaron. 3) En caso contrario el sistema envía un mensaje de que no hay datos referentes a los antecedentes patológicos personales.
Flujo alternativo	
Puntos de extensión.	

CU-6	**Gestionar Factores de riesgo para mayores de quince años**
Propósito	Realizar modificaciones o crear datos de cualquier factor de riesgo para mayor de quince años del paciente.
Actores	Médicos MGI.
Resumen: Cualquier médico MGI hace una petición de cada una de las acciones que se brindan	
Referencias	RF-2, 3

Acción del actor	**Respuesta del sistema**
1) Petición de crear factores de riesgo para mayores de quince años.	1) El sistema muestra la caja de diálogo para que el actor introduzca los datos referentes a los factores de riesgo para mayores de quince años del paciente.
2) Petición de modificar un factor de riesgo para mayores de quince años.	1) El sistema chaquea si el paciente tiene datos almacenados referentes a los factores de riesgo para mayores de quince años. 2) En caso afirmativo el sistema muestra una

	caja de dialogo con los valores anteriores de un factor de riesgo para mayores de quince años y con opción de modificaron. 3) En caso contrario el sistema envía un mensaje de que no hay datos referentes a los factores de riesgo para mayores de quince años.
Flujo alternativo	
Puntos de extensión.	

CU-7	**Gestionar Menor de un año**
Propósito	Realizar modificaciones o crear datos para menor de un año del paciente.
Actores	Médicos MGI.
Resumen: Cualquier médico MGI hacen una petición de cada una de las acciones que se brindan	
Referencias	RF-2, 3

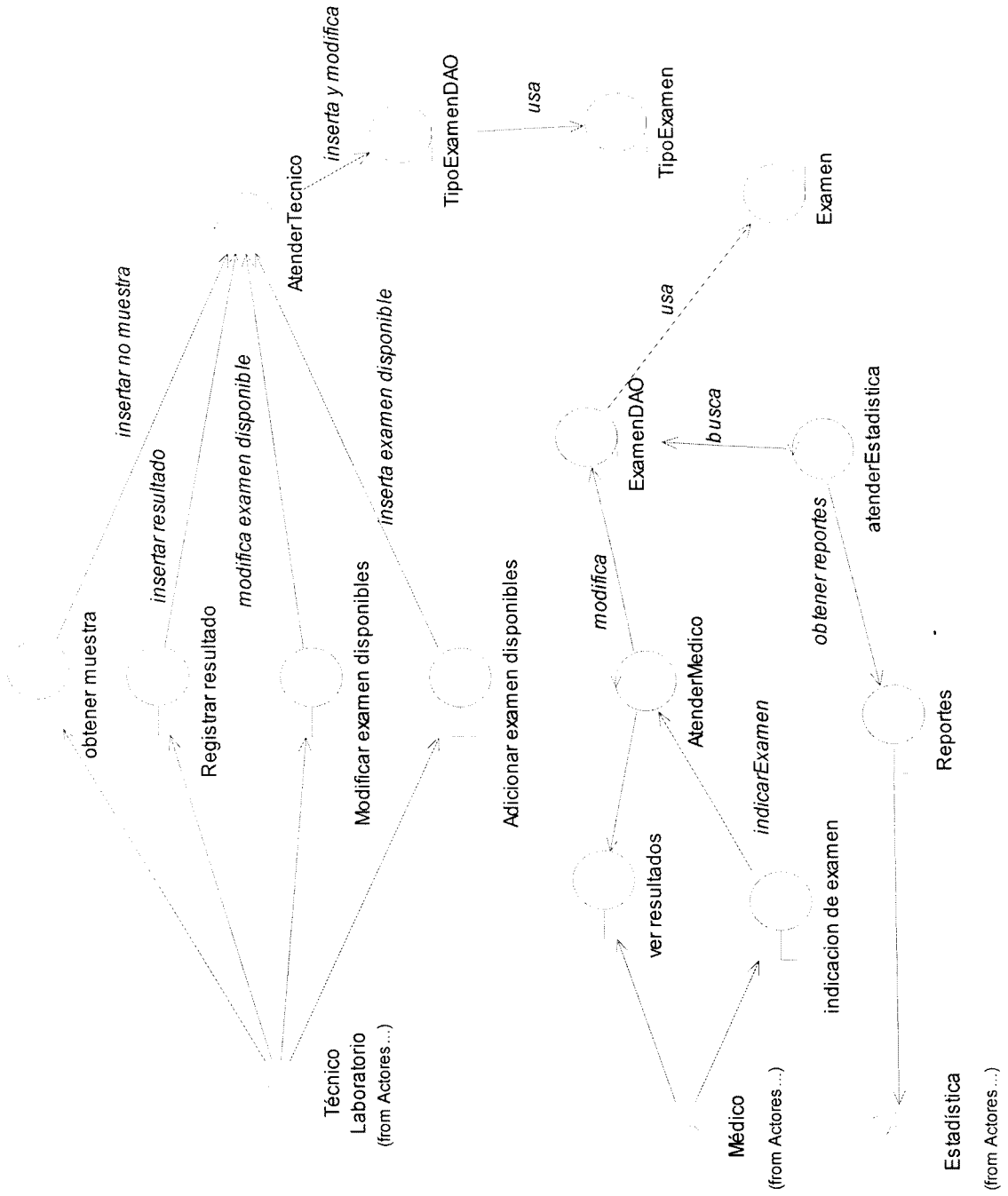
Acción del actor	**Respuesta del sistema**
1) Petición de crear datos para menor de un año	1) El sistema muestra la caja de diálogo para que el actor introduzca los datos para menor de un año del paciente.
2) Petición de modificar un dato para menor de un año	1) El sistema chaquea si el paciente tiene datos almacenados para menor de un año. 2) En caso afirmativo el sistema muestra una caja de dialogo con los valores anteriores de

	un datos de menor de un año y con opción de modificaron. 3) En caso contrario el sistema envía un mensaje de que no hay datos referentes para menor de un año.
Flujo alternativo	
Puntos de extensión.	

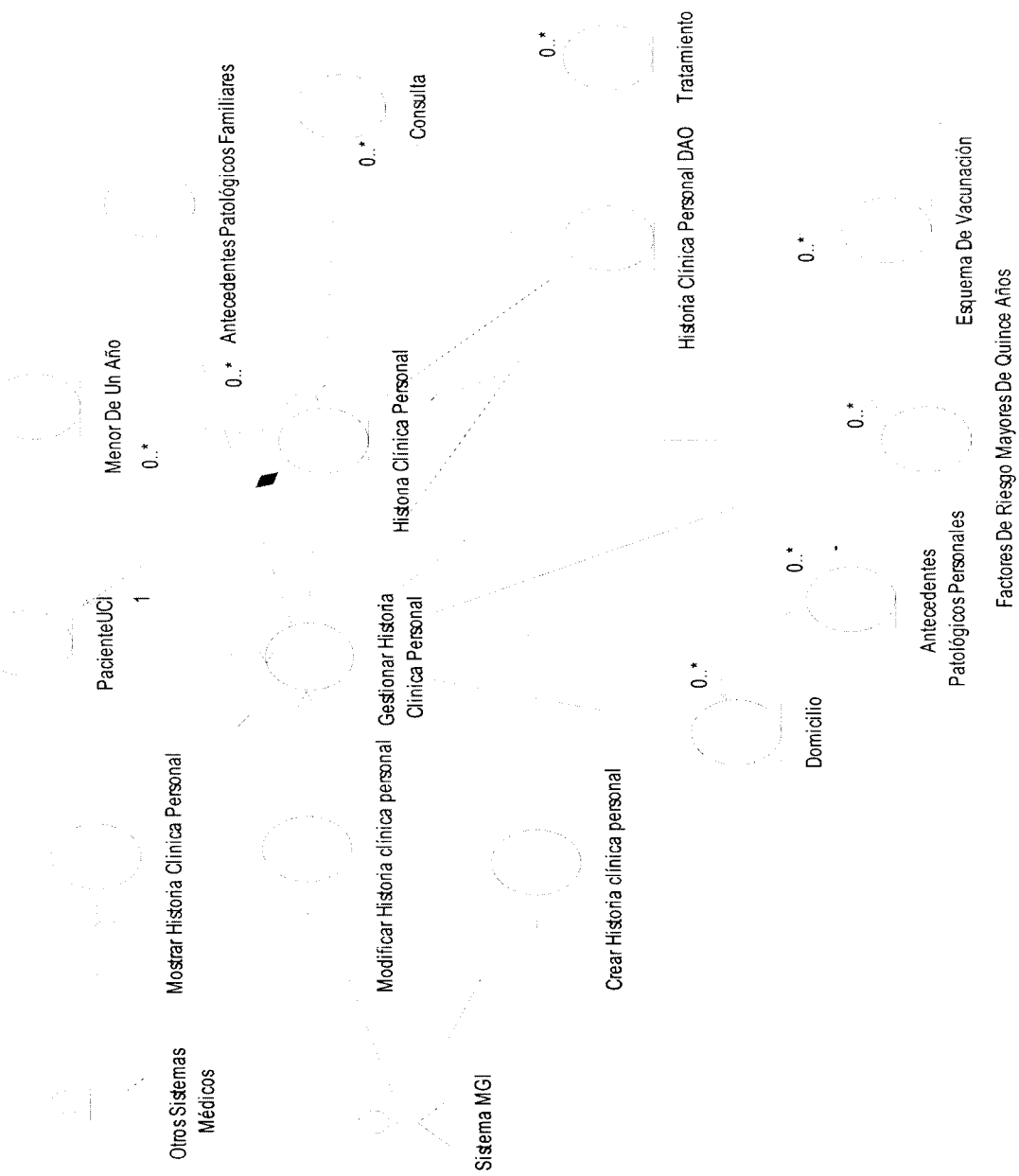
CU-8	**Gestionar Esquema de vacunación**	
Propósito	Realizar modificaciones o crear datos del esquema de vacunación de cada paciente.	
Actores	Médicos MGI.	
Resumen: Cualquier médico MGI hacen una petición de cada una de las acciones que se brindan		
Referencias	RF- 2, 3	
Acción del actor		**Respuesta del sistema**
1) Petición de crear el esquema de vacunación		1) El sistema muestra la caja de diálogo para que el actor introduzca los datos referentes al esquema de vacunación del paciente.
2) Petición de modificar el esquema de vacunación.		1) El sistema chaquea si el paciente tiene datos almacenados referentes del esquema de vacunación 2) En caso afirmativo el sistema muestra una caja de dialogo con los valores anteriores del esquema de vacunación y con opción de

	modificaron. 3) En caso contrario el sistema envía un mensaje de que no hay datos referentes del esquema de vacunación.
Flujo alternativo	
Puntos de extensión.	

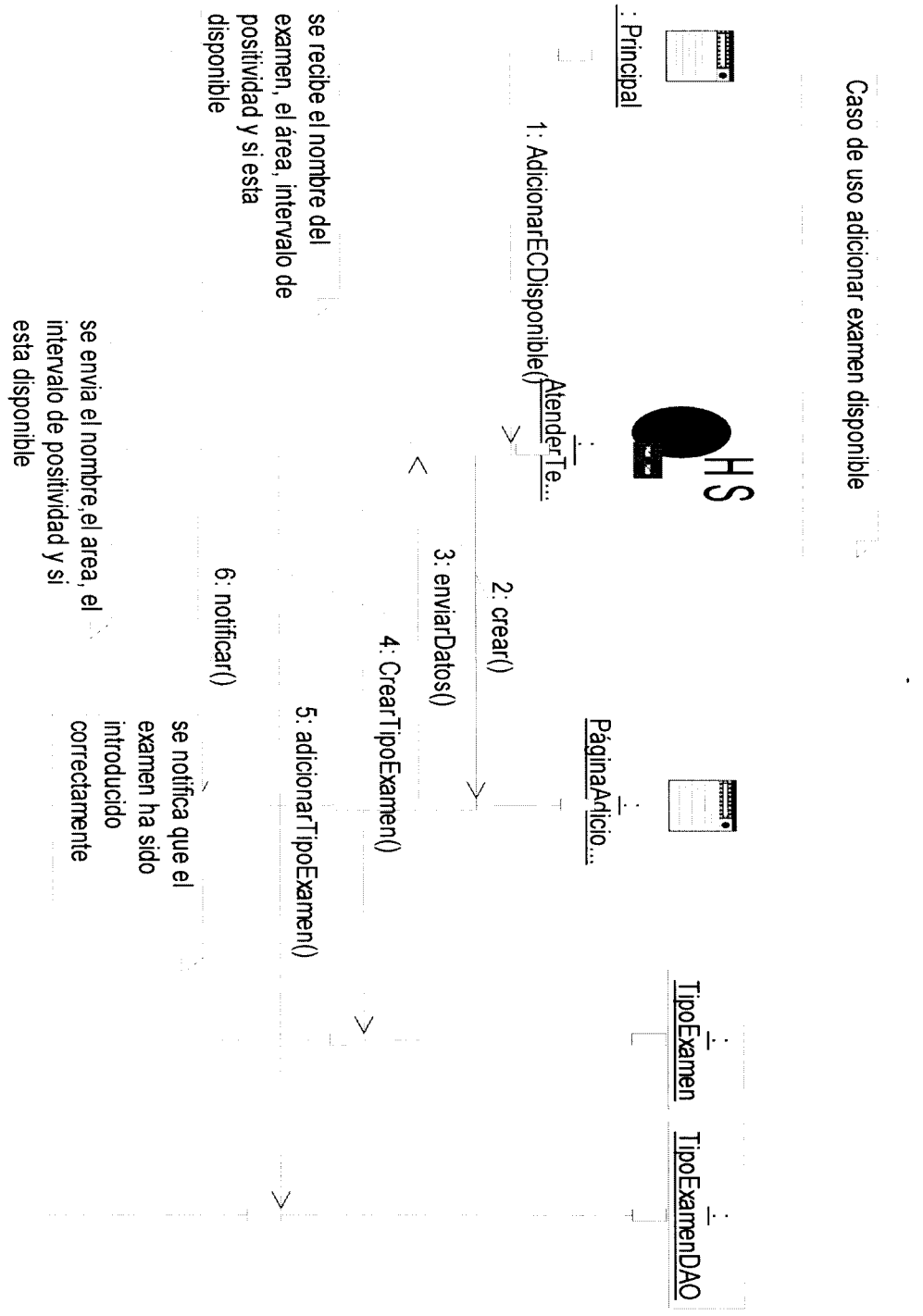
Anexo 13. Diseño clases del análisis. (MLC).



Anexo 14. Diseño clases del análisis. (MHCP).



Anexo 15. Diagramas de Interacción. (MLC).



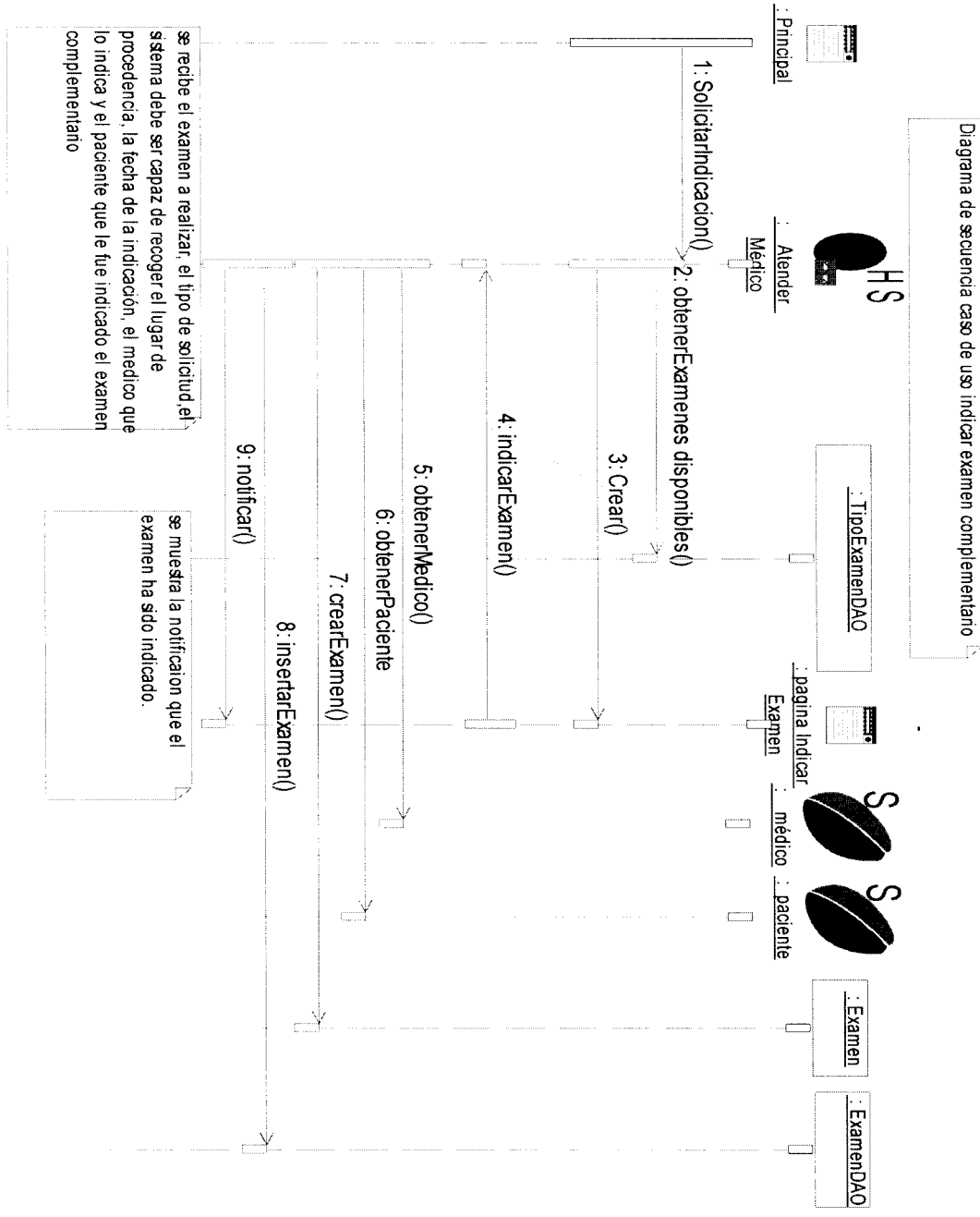
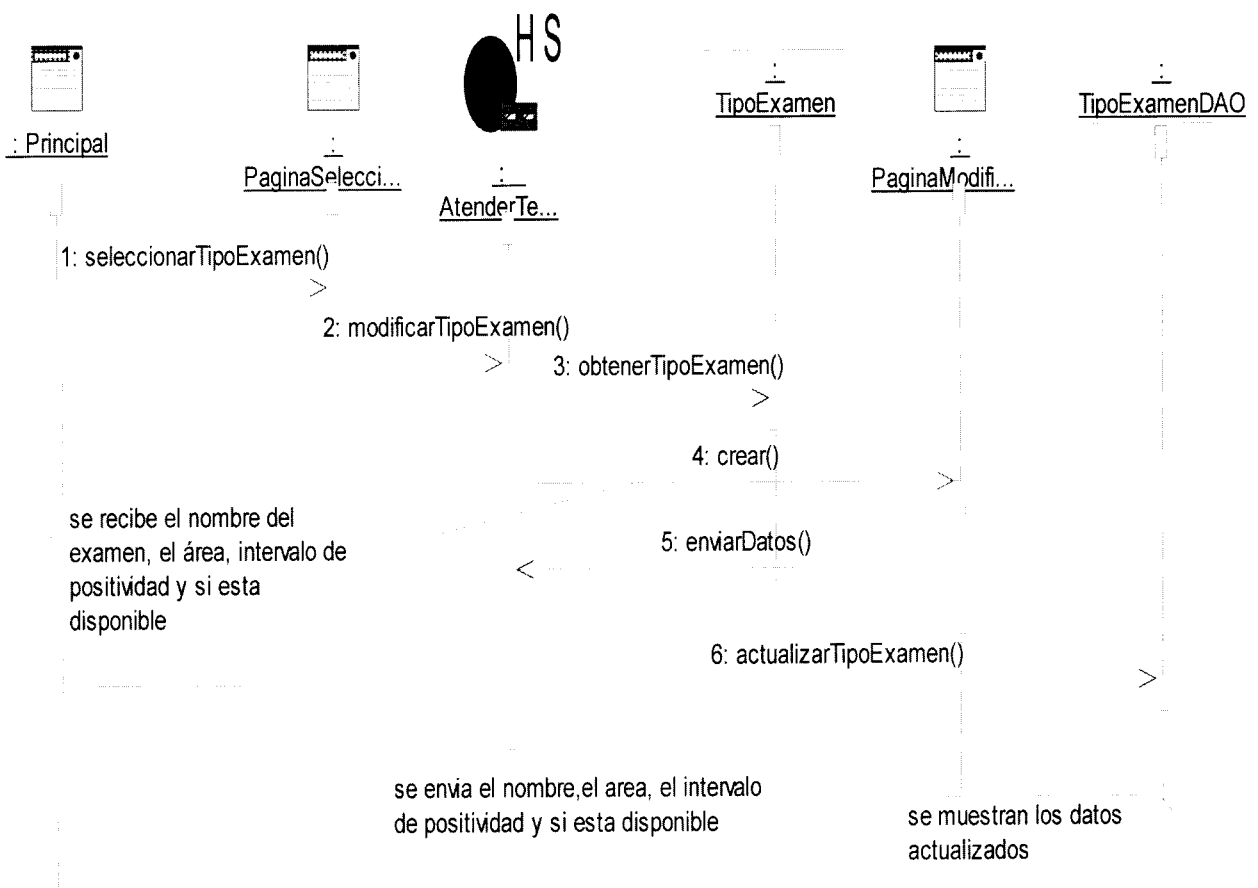


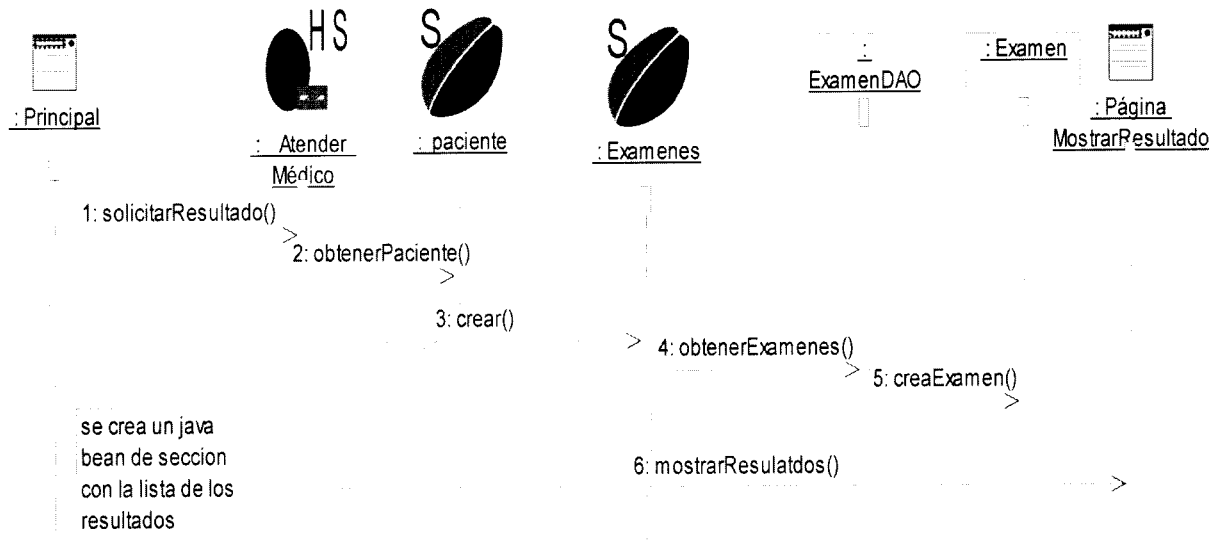
Diagrama de secuencia caso de uso indicar examen complementario
: Principal
: Atender Médico
: TipoExamenDAO
: pagina Indicar Examen
: médico
: paciente
: Examen
: ExamenDAO
1: SolicitarIndicacion()
2: obtenerExamenes disponibles()
3: Crear()
4: indicarExamen()
5: obtenerMedico()
6: obtenerPaciente
7: crearExamen()
8: insertarExamen()
9: notificar()
se recibe el examen a realizar, el tipo de solicitud,el sistema debe ser capaz de recoger el lugar de procedencia, la fecha de la indicación, el medico que lo indica y el paciente que le fue indicado el examen complementario
se muestra la notificaion que el examen ha sido indicado.

Caso de uso modificar examen disponible

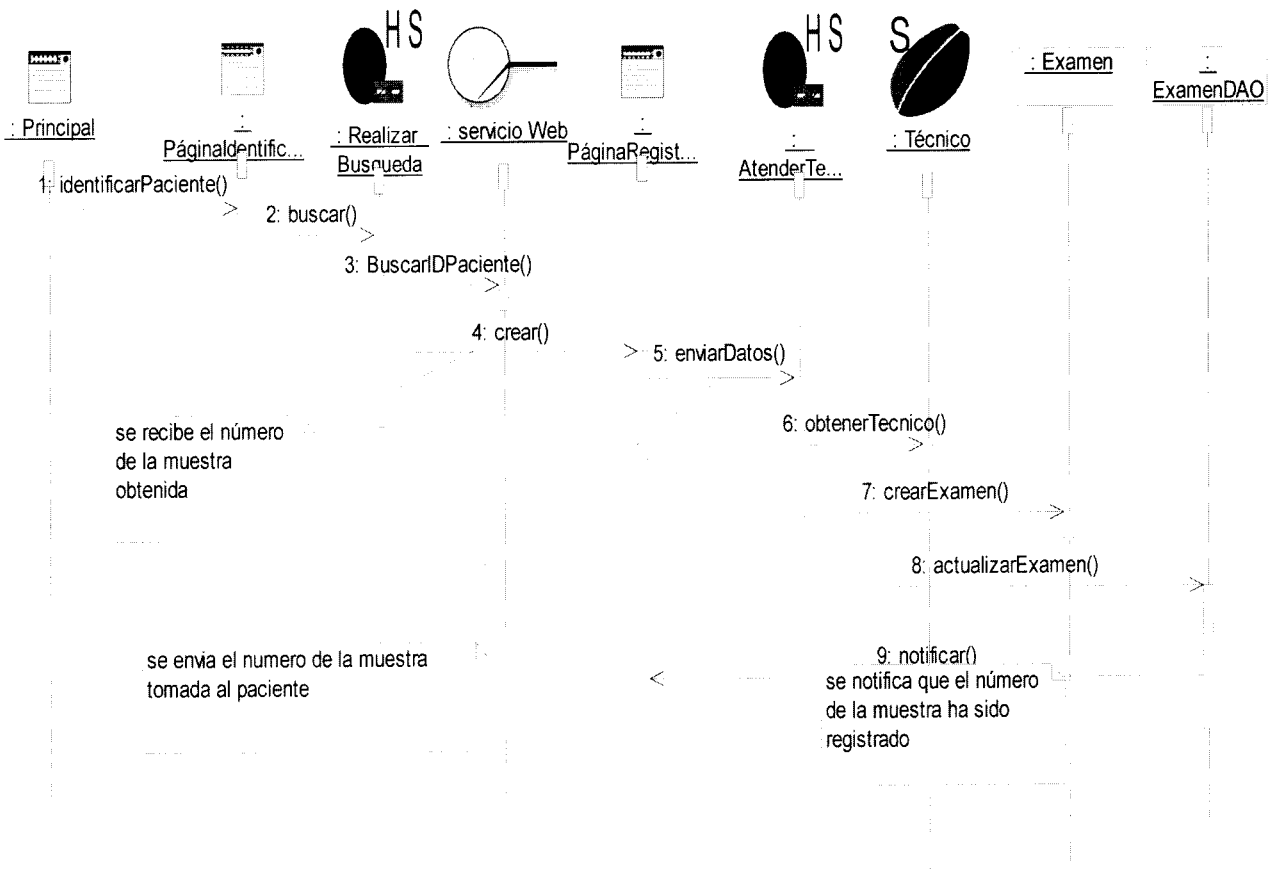


caso de uso obtener reporte
: Principal
HS
atenderEs...
PaginaSelecci...
ExamenDAO
PaginaMostra...
1: seleccionarReporte()
2: crear()
3: enviarDatos()
se recibe los parametros
que se necesitan para
cada reporte
4: obtenerResultados()
5: mostrarResultados()
se envían los datos del
reporte que se desea obtener

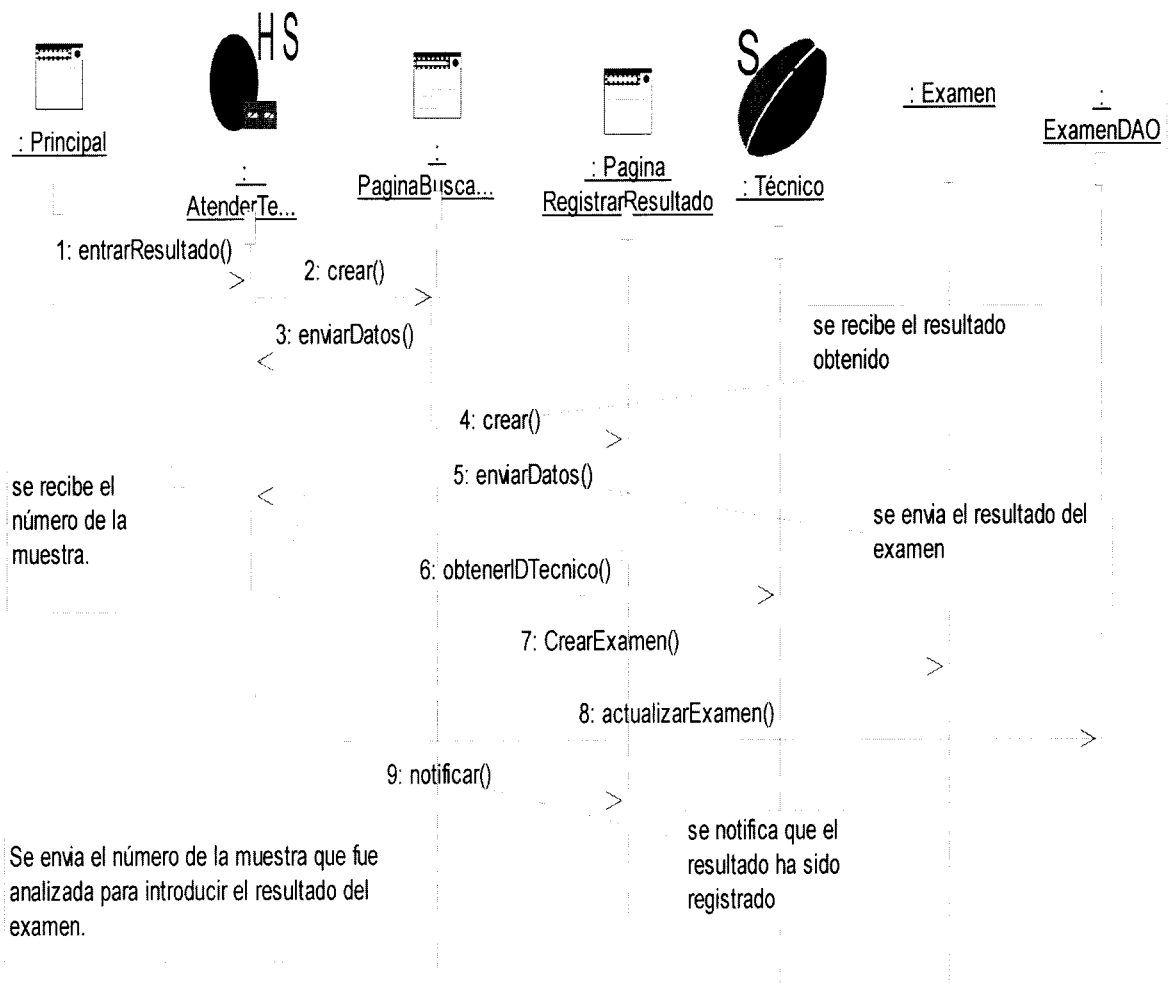
Caso de uso obtener resultado de examen complementario



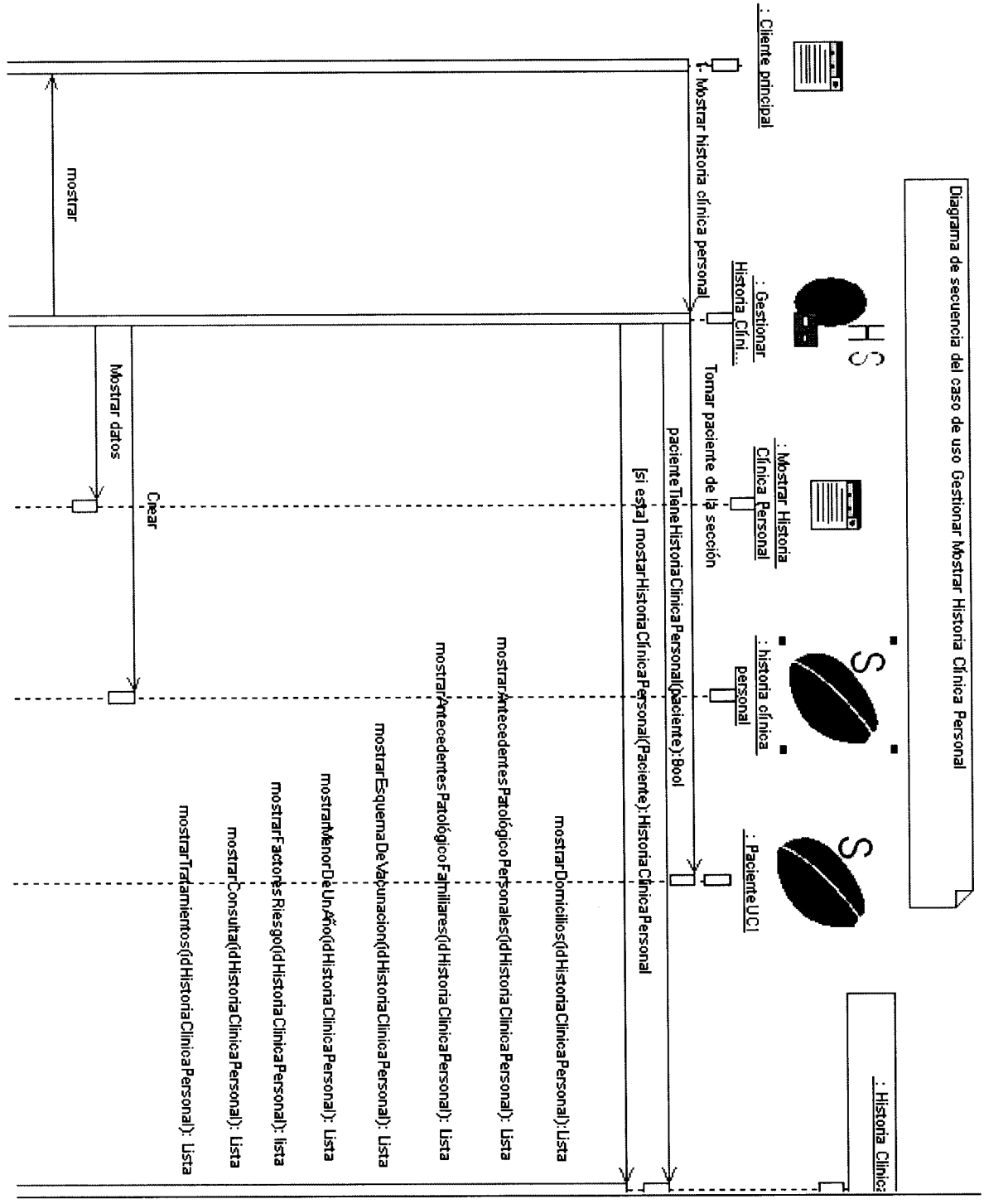
Caso de uso registrar muestra

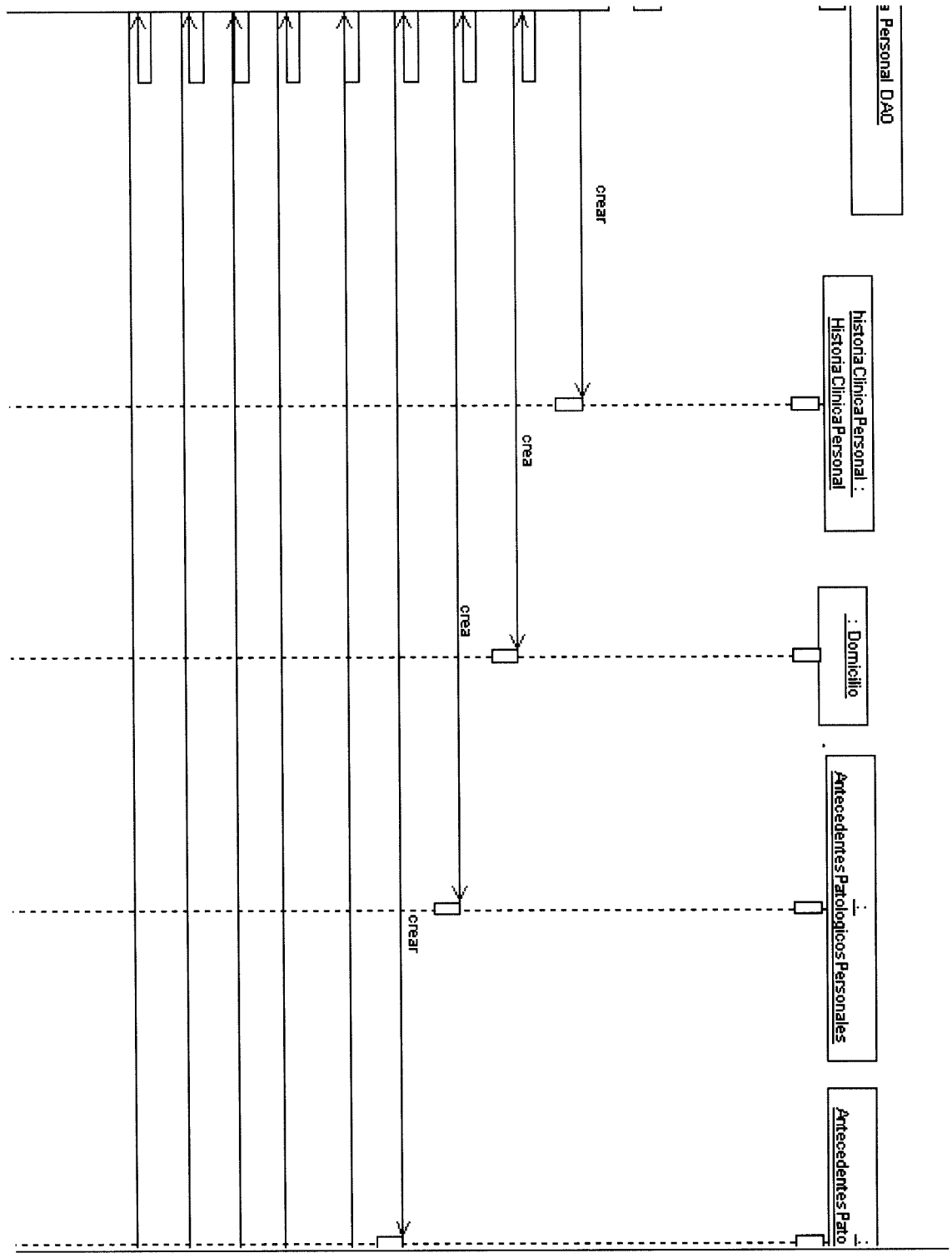


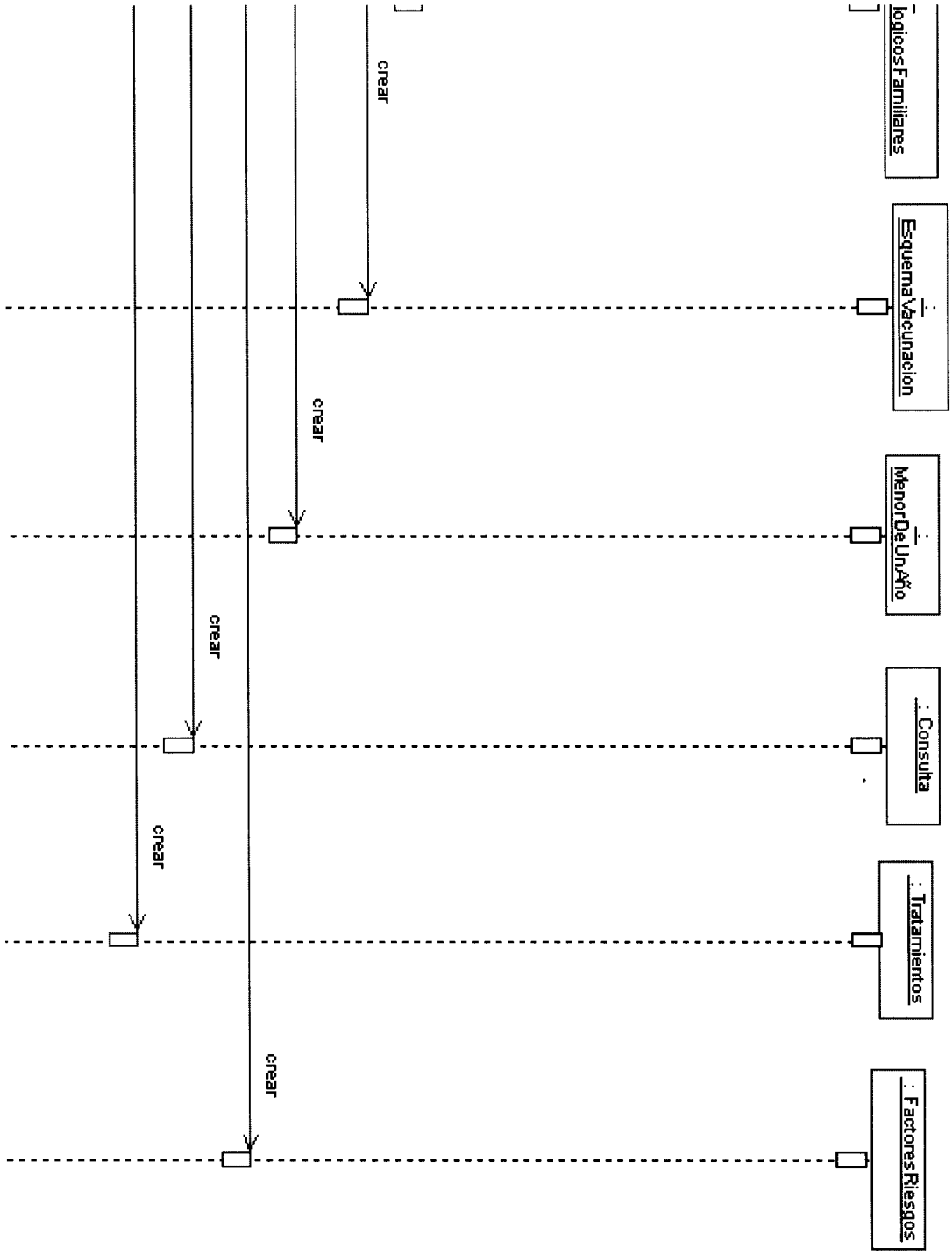
Caso de uso Registrar resultado



Anexo 16. Diagramas de Interacción. (MHCP).




Personal DAO
historiaClinicaPersonal : HistoriaClinicaPersonal
: Domicilio
AntecedentesPatologicosPersonales
AntecedentesPato
crear
crea
crea
crear


logicosFamiliares
EsquemaVacunacion
MenorDeUnAño
: Consulta
: Tratamientos
: FactoresRiesgos
crear
crear
crear
crear
crear

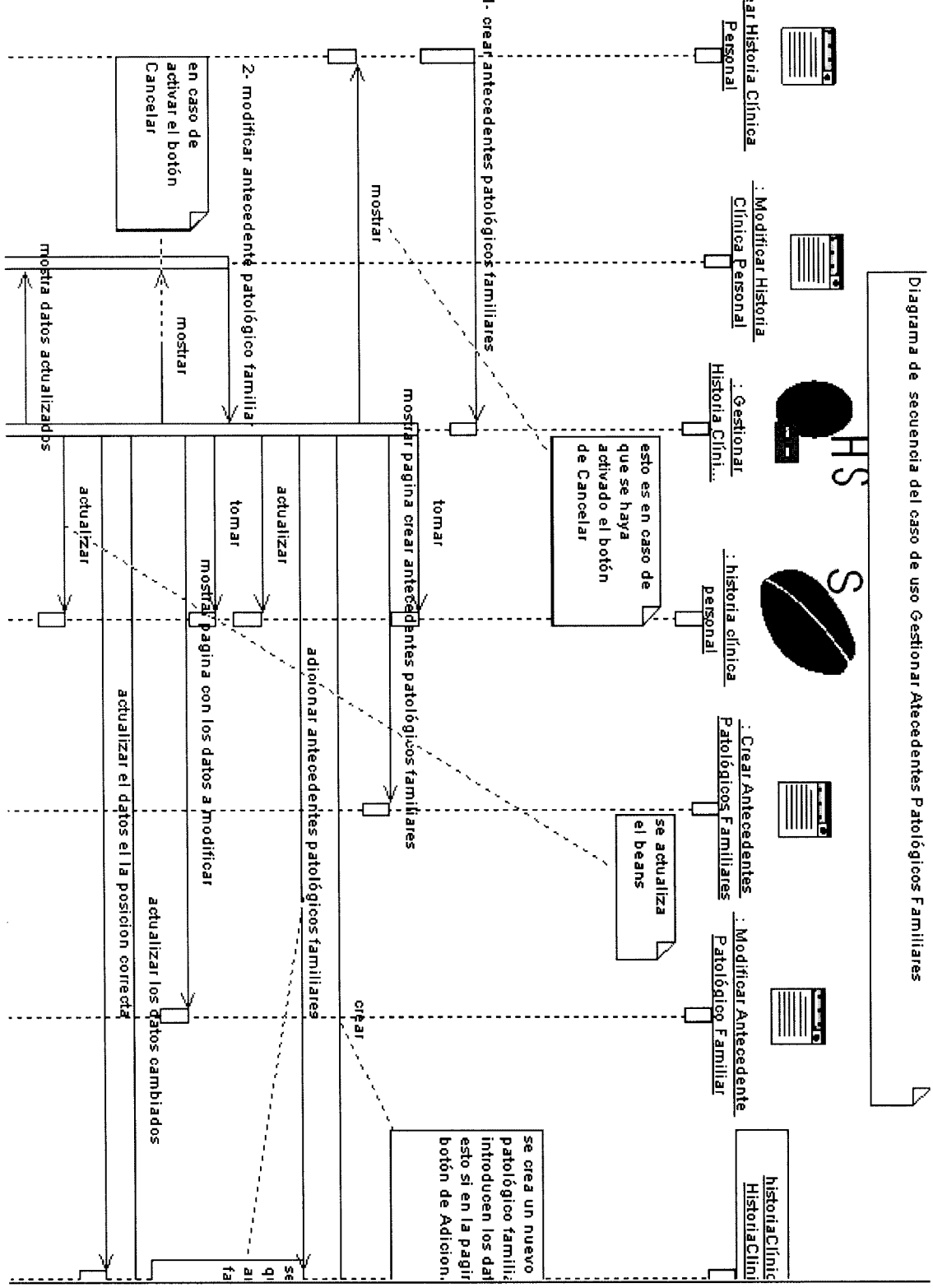
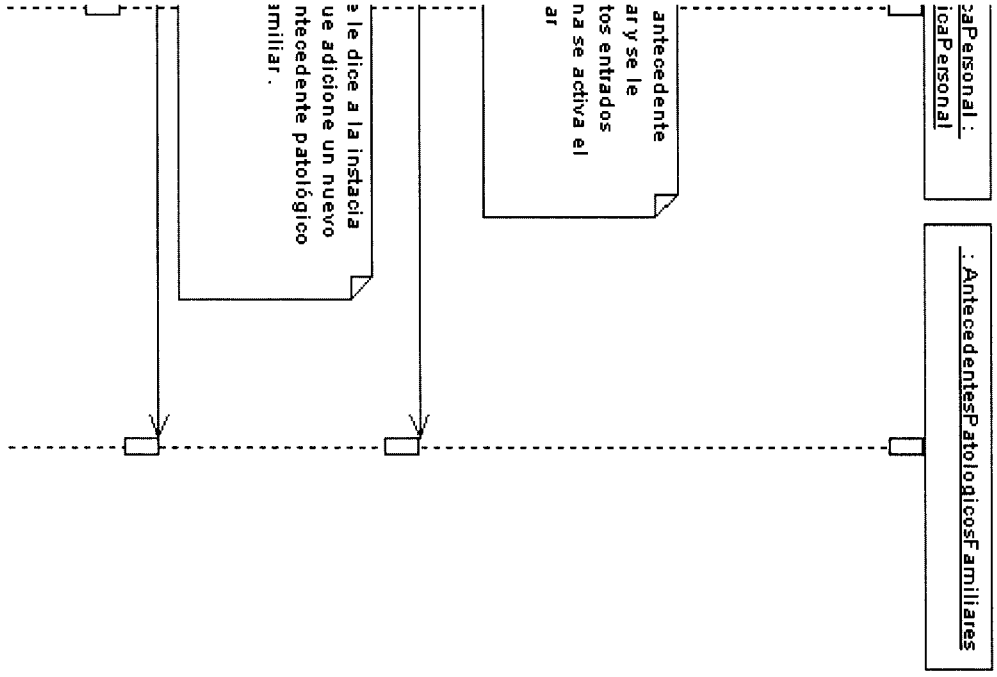


Diagrama de secuencia del caso de uso Gestionar Atecedentes Patológicos Familiares
: Crear Historia Clínica Personal
: Modificar Historia Clínica Personal
: Gestionar Historia Clíni...
: historia clínica personal
: Crear Antecedentes Patológicos Familiares
: Modificar Antecedente Patológico Familiar
historiaClínic HistoriaClini
1- crear antecedentes patológicos familiares
esto es en caso de que se haya activado el botón de Cancelar
se actualiza el beans
se crea un nuevo patológico familia introducen los dat esto si en la pagir botón de Adicion.
mostrar
mostrar pagina crear antecedentes patológicos familiares
tomar
crear
2- modificar antecedente patológico familia
en caso de activar el botón Cancelar
mostrar
actualizar
tomar
adicionar antecedentes patológicos familiares
mostrar pagina con los datos a modificar
actualizar
actualizar los datos cambiados
actualizar el datos el la posicion correcta
mostra datos actualizados


caPersonal :
icaPersonal
: AntecedentesPatologicosFamiliares
antecedente
ar y se le
tos entrados
na se activa el
ar
e le dice a la instacia
ue adicione un nuevo
ntecedente patológico
miliar .

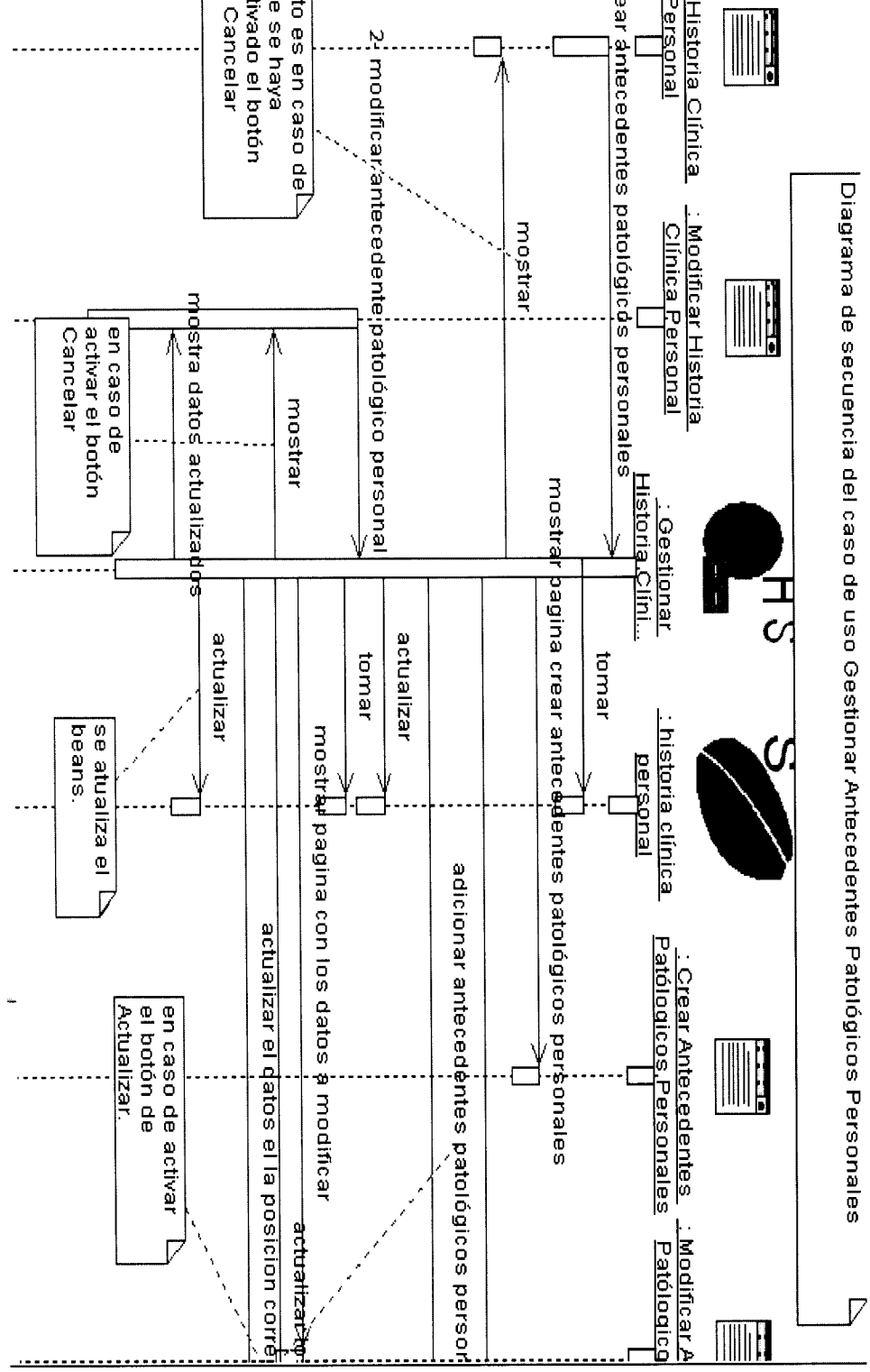
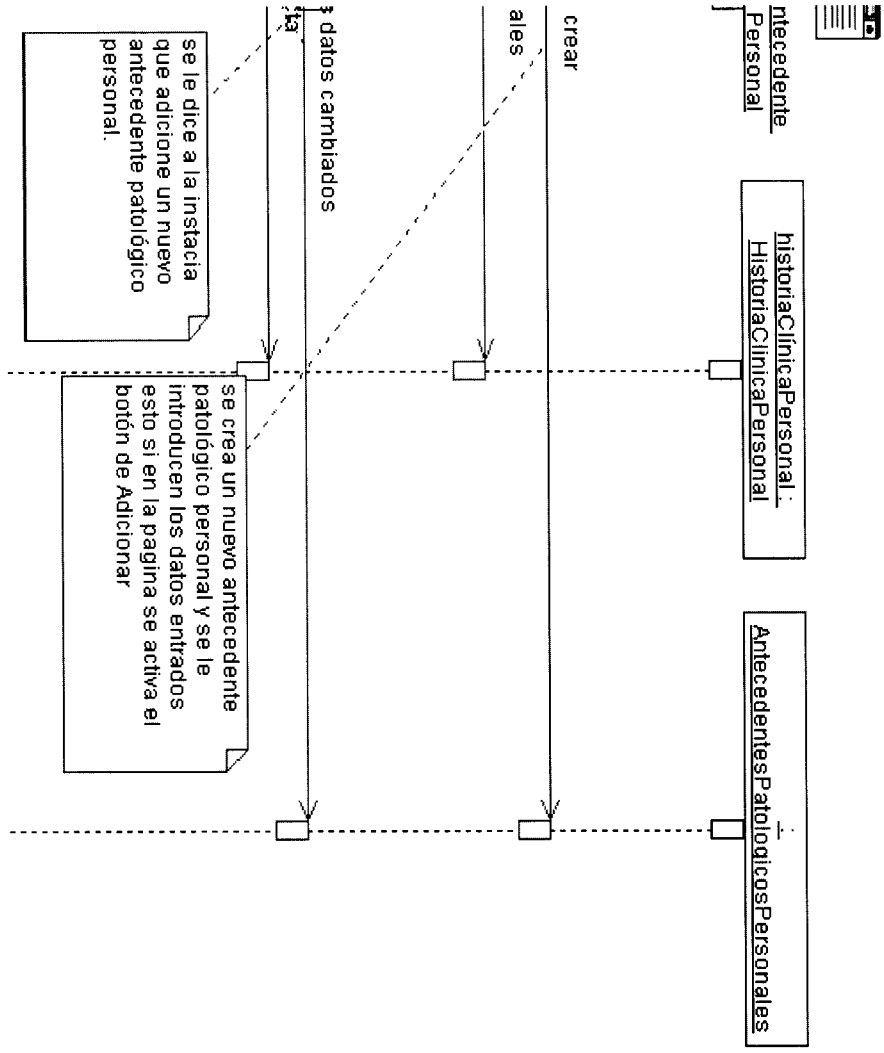


Diagrama de secuencia del caso de uso Gestionar Antecedentes Patológicos Personales
: Crear Historia Clínica Personal
: Modificar Historia Clínica Personal
: Gestionar Historia Clíni...
: historia clínica personal
: Crear Antecedentes Patólogicos Personales
: Modificar A Patólogico
1- crear antecedentes patológicos personales
mostrar
mostrar pagina crear antecedentes patológicos personales
tomar
esto es en caso de que se haya activado el botón de Cancelar
2- modificar antecedente patológico personal
mostrar
actualizar
tomar
adicionar antecedentes patológicos person
en caso de activar el botón Cancelar
mostrar pagina con los datos a modificar
actualizar
mostra datos actualizados
actualizar el datos el la posicion corre
actualizar
se atualiza el beans.
en caso de activar el botón de Actualizar.


ntecedente
Personal
historiaClínicaPersonal :
HistoriaClínicaPersonal
AntecedentesPatologicosPersonales
crear
ales
datos cambiados
se le dice a la instacia
que adicione un nuevo
antecedente patológico
personal.
se crea un nuevo antecedente
patológico personal y se le
introducen los datos entrados
esto si en la pagina se activa el
botón de Adicionar

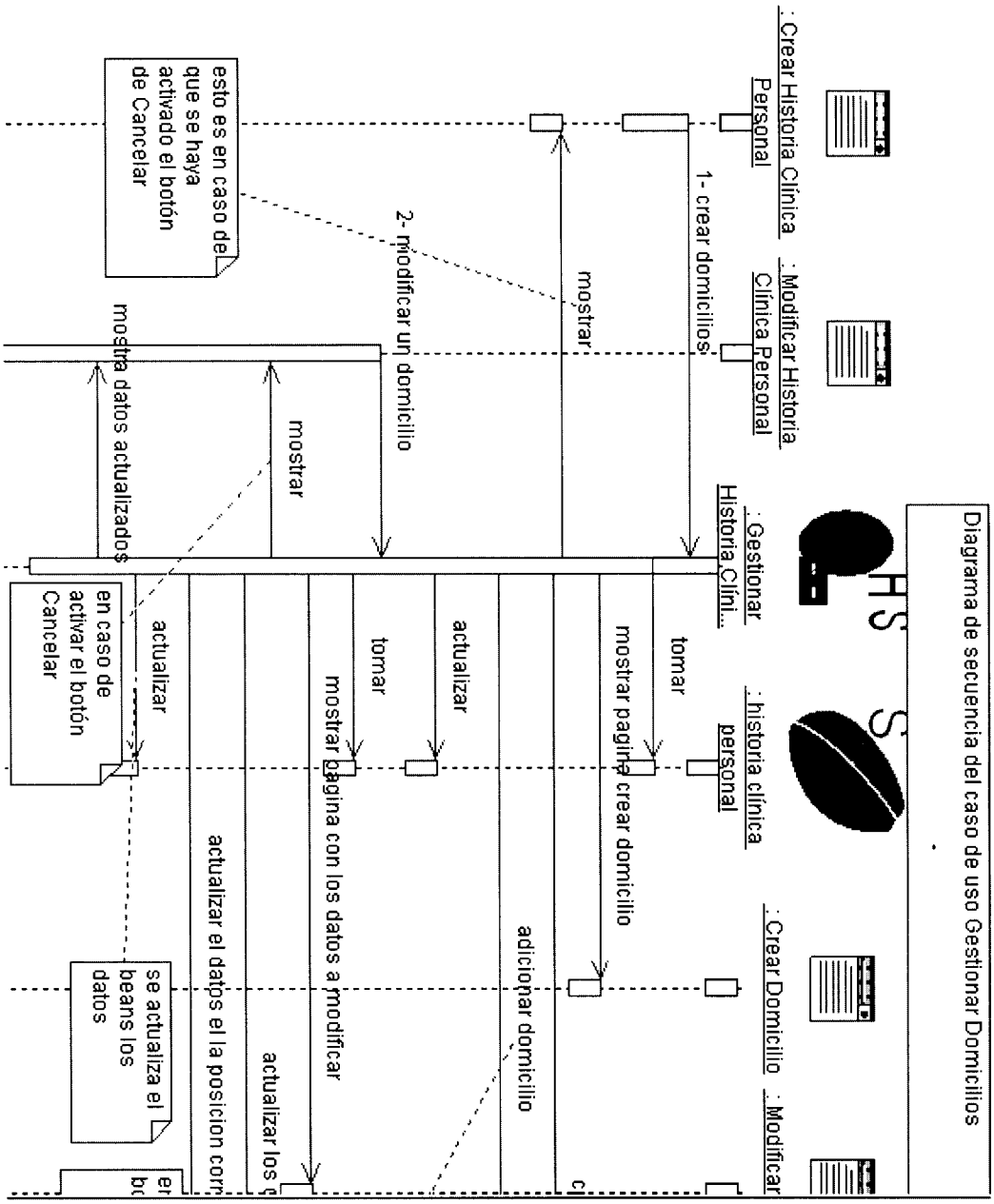
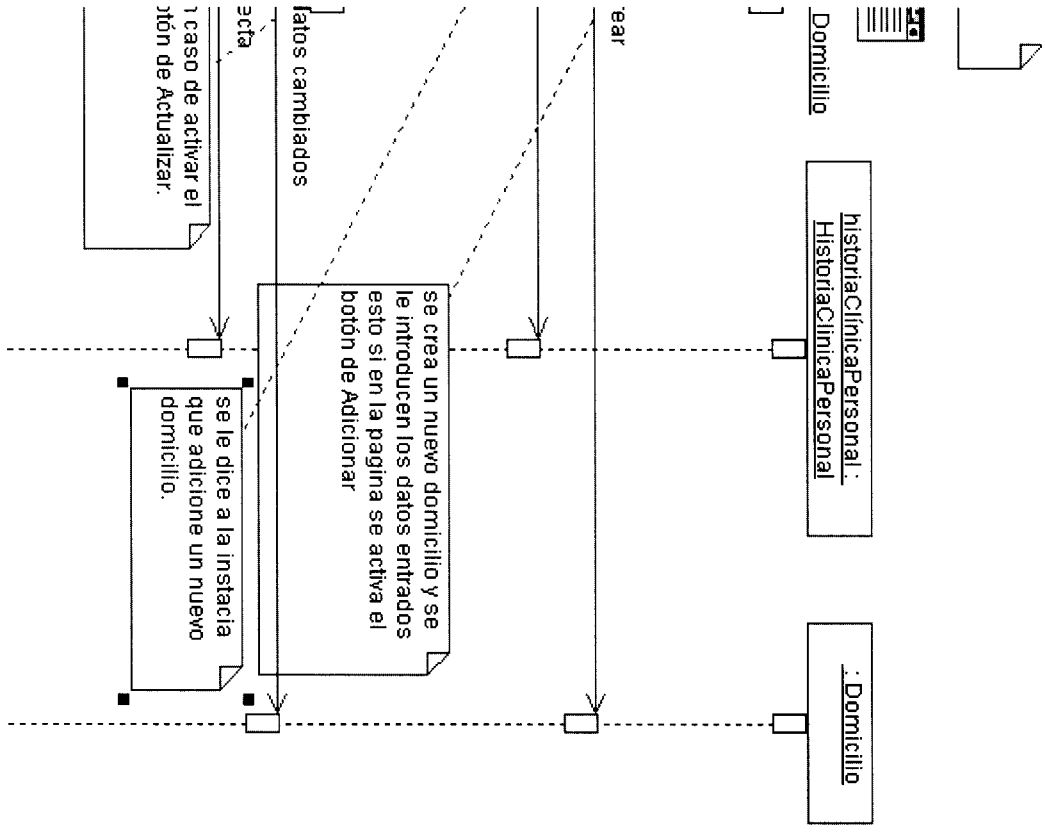


Diagrama de secuencia del caso de uso Gestionar Domicilios
: Crear Historia Clínica Personal
: Modificar Historia Clínica Personal
: Gestionar Historia Clíni...
: historia clínica personal
: Crear Domicilio
: Modificar
1- crear domicilios
mostrar
2- modificar un domicilio
mostrar
mostrar datos actualizados
esto es en caso de que se haya activado el botón de Cancelar
tomar
mostrar pagina crear domicilio
adicionar domicilio
actualizar
tomar
mostrar pagina con los datos a modificar
actualizar los
actualizar el datos el la posicion corr
actualizar
en caso de activar el botón Cancelar
se actualiza el beans los datos


Domicilio
historiaClínicaPersonal :
HistoriaClinicaPersonal
: Domicilio
ear
atos cambiados
se crea un nuevo domicilio y se
le introducen los datos entrados
esto si en la pagina se activa el
botón de Adicionar
se le dice a la instacia
que adicione un nuevo
domicilio.
1 caso de activar el
ión de Actualizar.

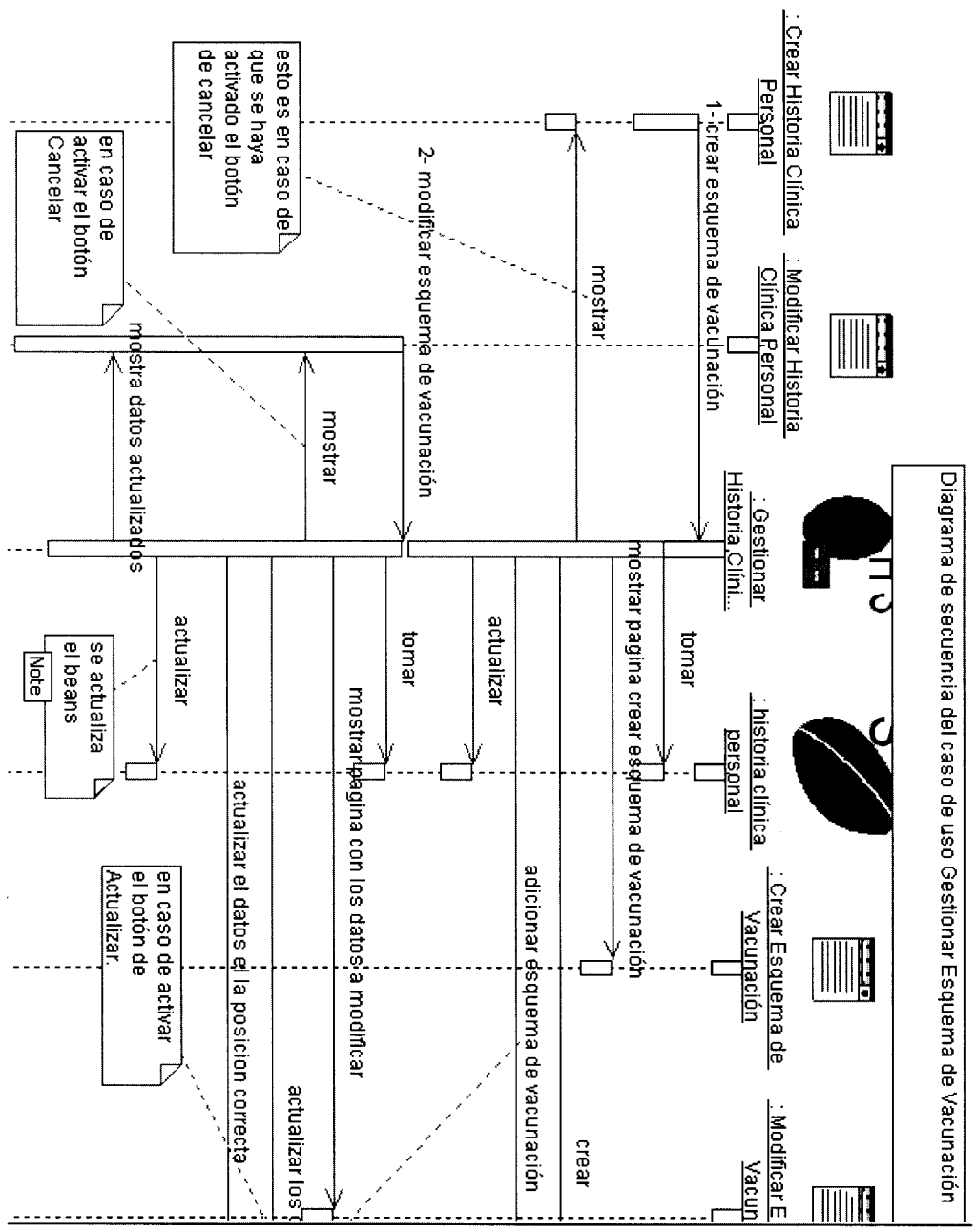
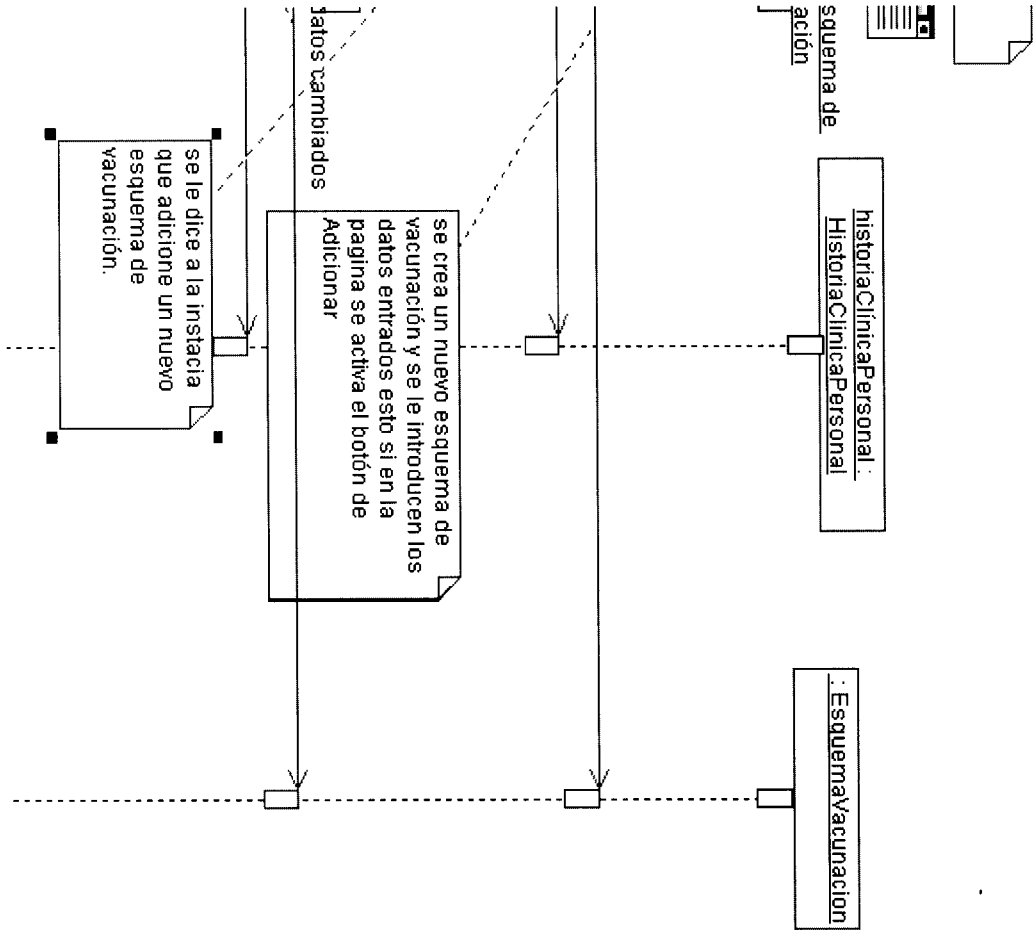


Diagrama de secuencia del caso de uso Gestionar Esquema de Vacunación
: Crear Historia Clínica Personal
: Modificar Historia Clínica Personal
: Gestionar Historia Clini...
: historia clínica personal
: Crear Esquema de Vacunación
: Modificar E Vacun
1- crear esquema de vacunación
mostrar
mostrar pagina crear esquema de vacunación
tomar
crear
adicionar esquema de vacunación
actualizar
2- modificar esquema de vacunación
mostrar
tomar
mostrar pagina con los datos a modificar
actualizar los
actualizar el datos en la posicion correcta
actualizar
mostra datos actualizados
esto es en caso de que se haya activado el botón de cancelar
en caso de activar el botón Cancelar
se actualiza el beans
Note
en caso de activar el botón de Actualizar.


esquema de
ación
historiaClínicaPersonal :
HistoriaClínicaPersonal
: EsquemaVacunacion
datos cambiados
se crea un nuevo esquema de
vacunación y se le introducen los
datos entrados esto si en la
pagina se activa el botón de
Adicionar
se le dice a la instacia
que adicione un nuevo
esquema de
vacunación.

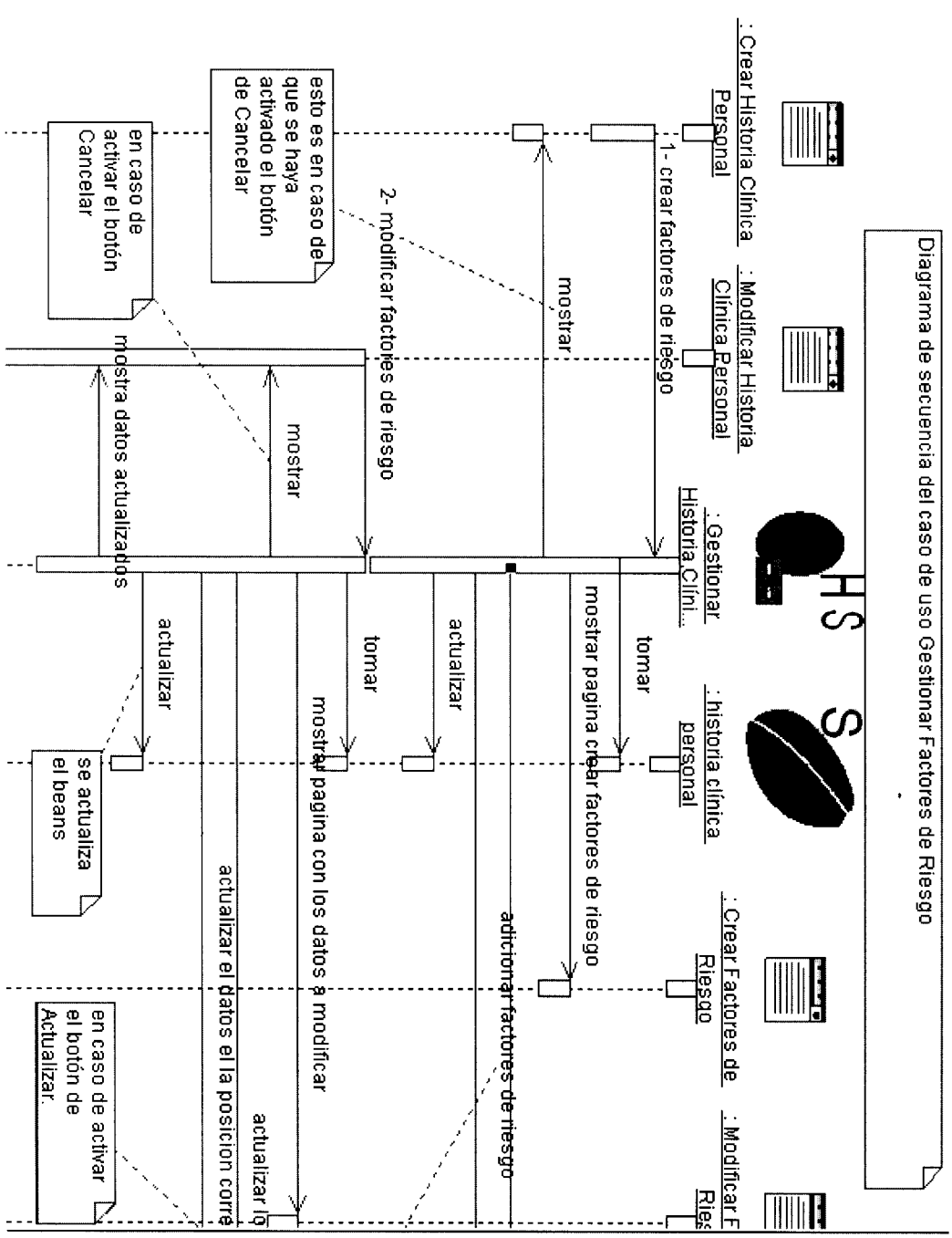
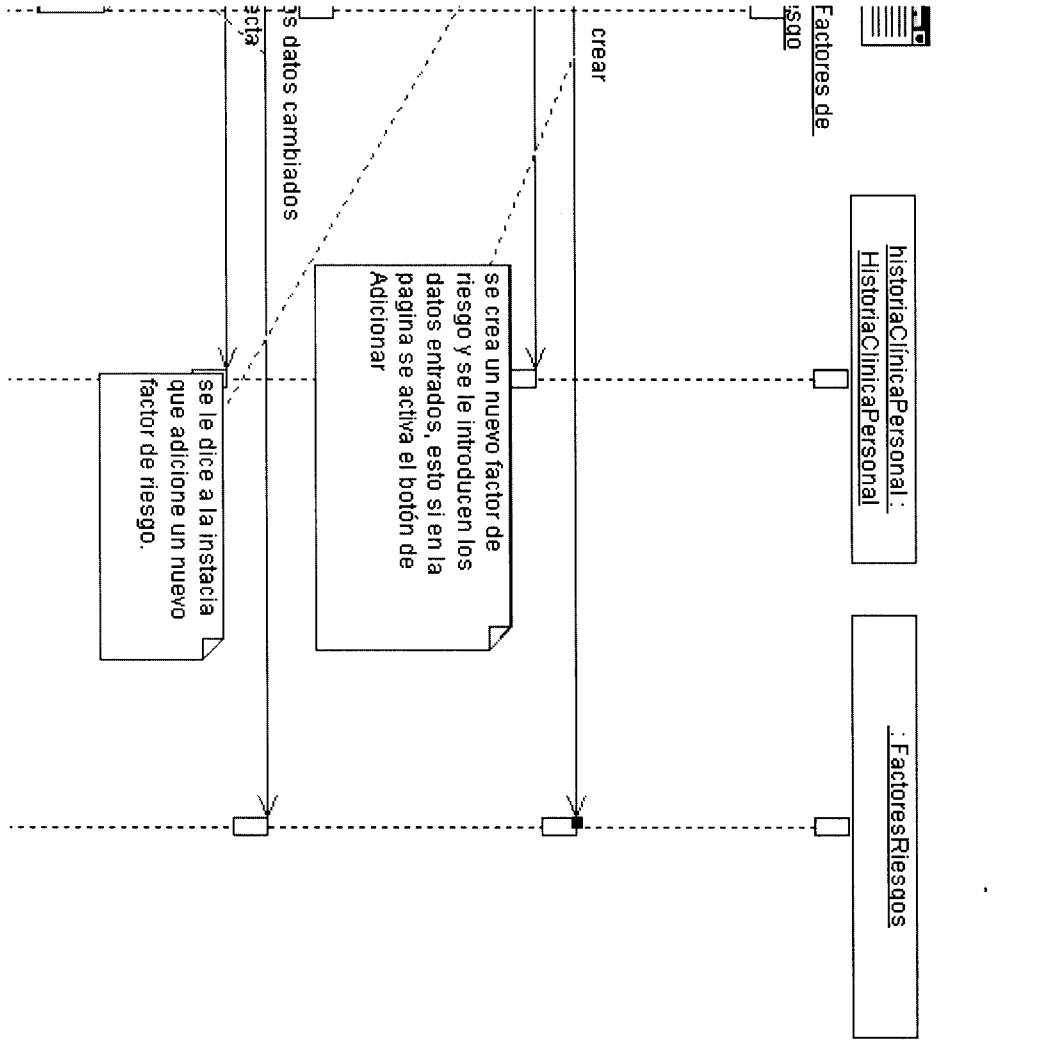
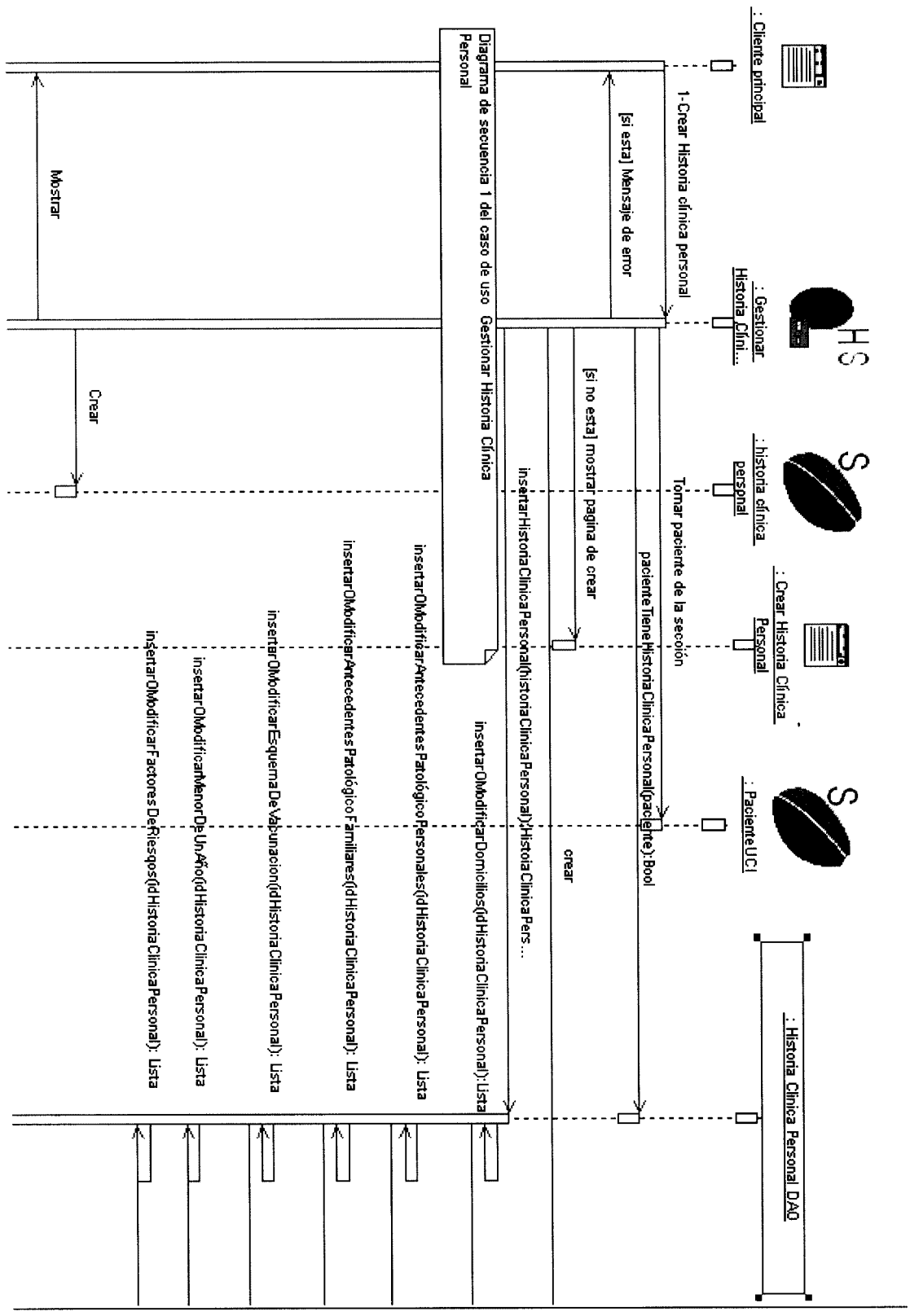
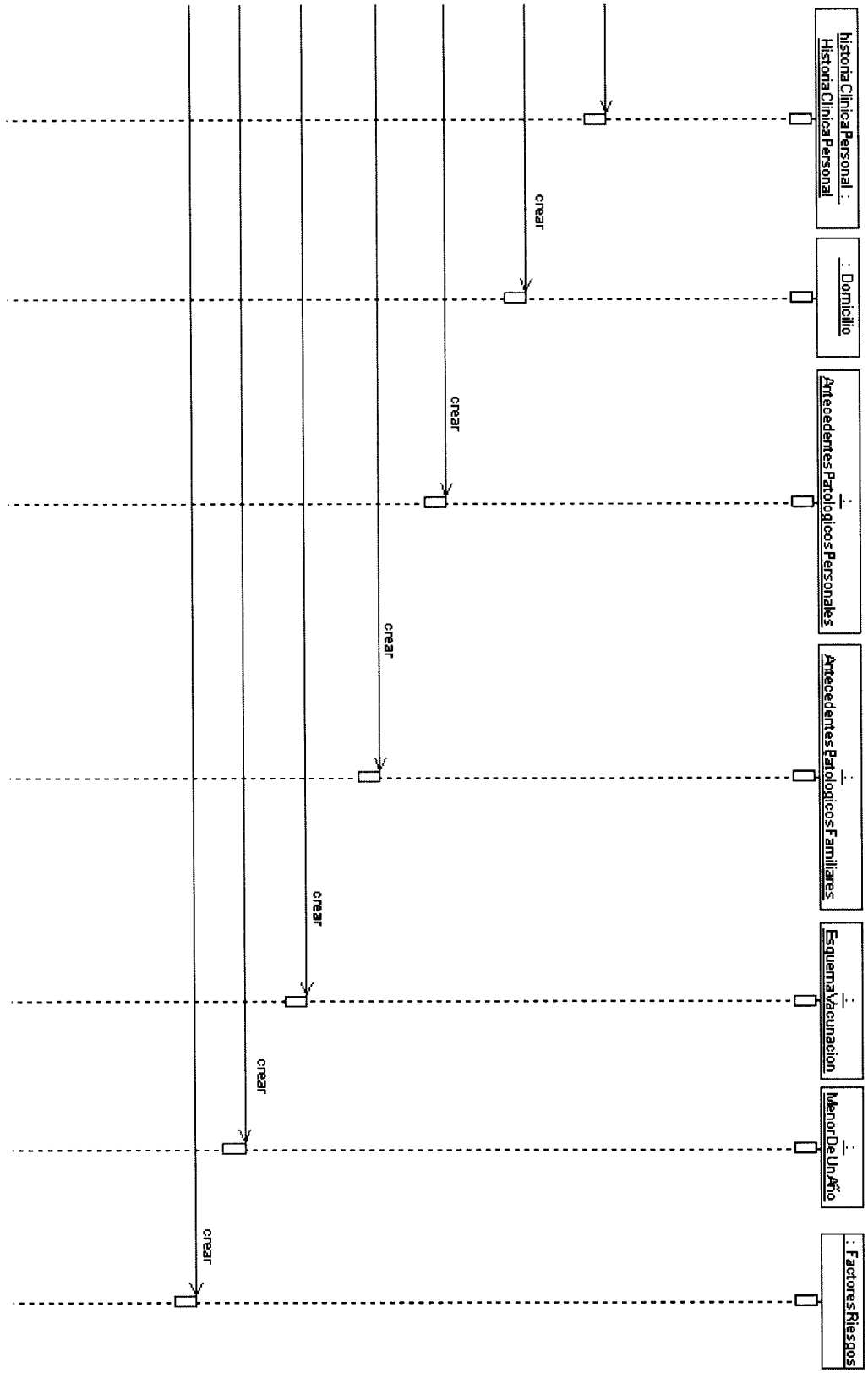
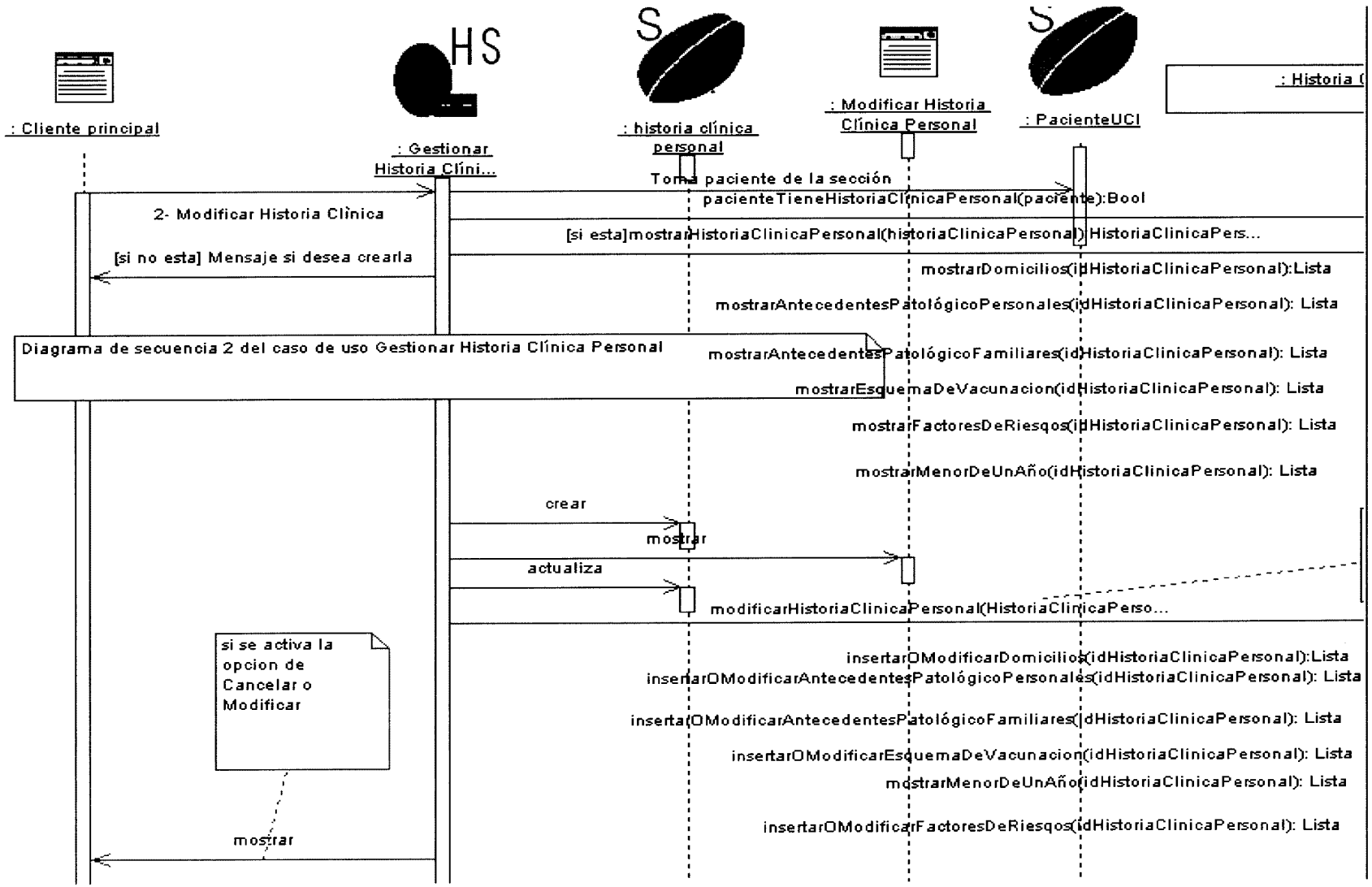


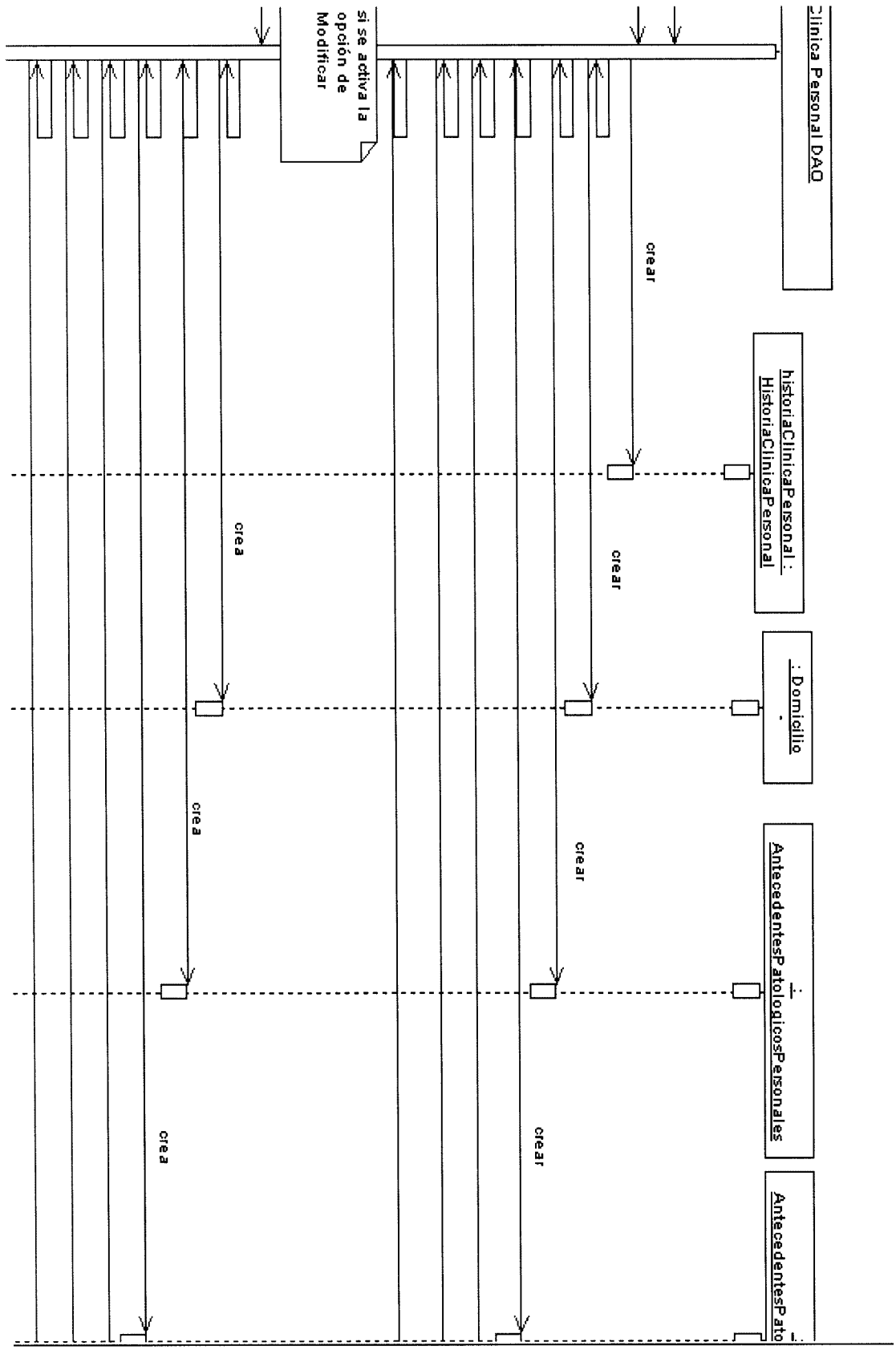
Diagrama de secuencia del caso de uso Gestionar Factores de Riesgo
: Crear Historia Clínica Personal
: Modificar Historia Clínica Personal
: Gestionar Historia Clíni...
: historia clínica personal
: Crear Factores de Riesgo
: Modificar F Ries
1- crear factores de riesgo
mostrar
mostrar pagina crear factores de riesgo
tomar
adicionar factores de riesgo
esto es en caso del que se haya activado el botón de Cancelar
2- modificar factores de riesgo
en caso de activar el botón Cancelar
mostrar
mostra datos actualizados
actualizar
tomar
actualizar
mostrar pagina con los datos a modificar
actualizar el datos el la posicion corre
actualizar lo
se actualiza el beans
en caso de activar el botón de Actualizar.

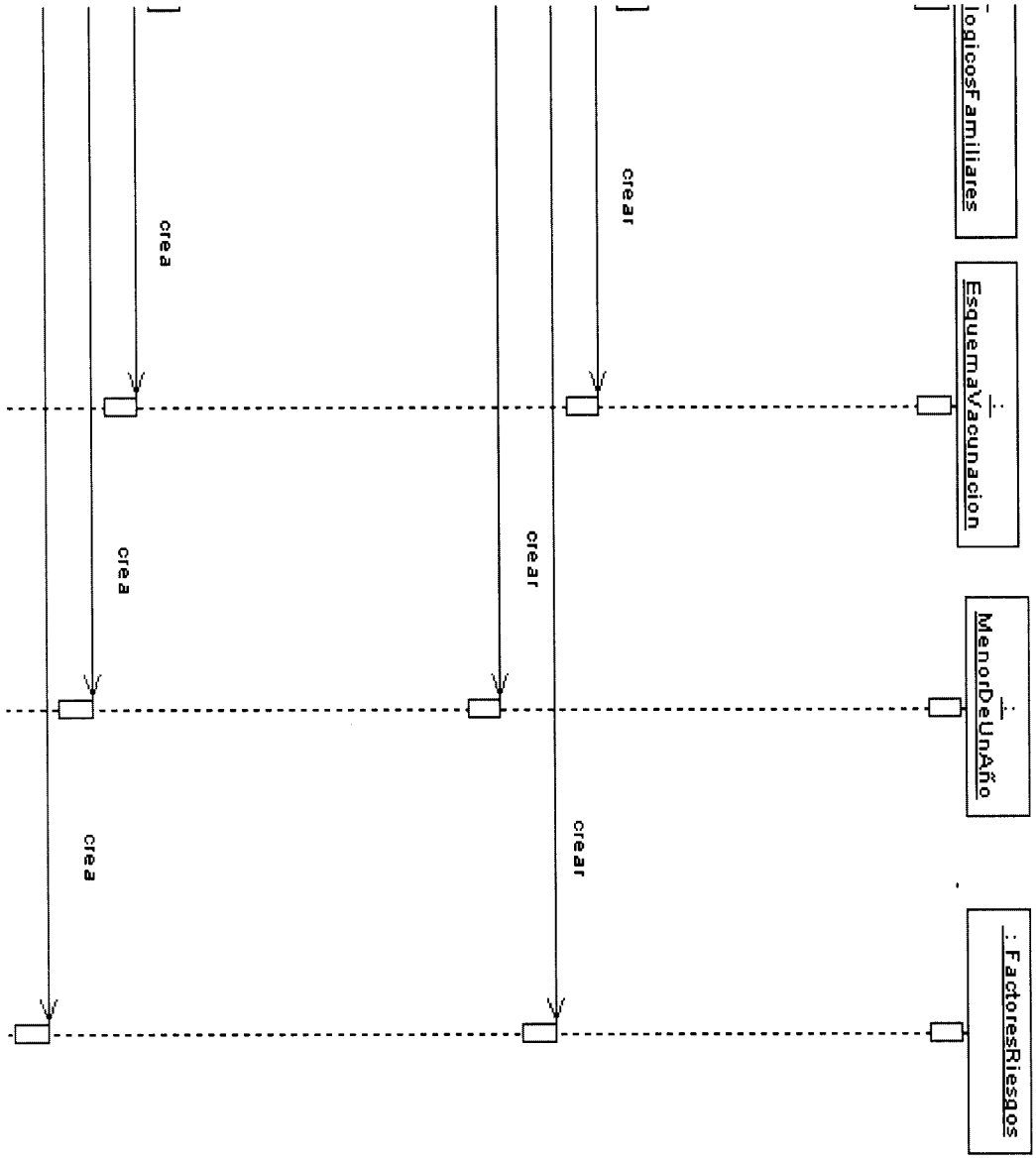

Factores de
isgo
historiaClínicaPersonal :
HistoriaClínicaPersonal
: FactoresRiesgos
crear
se crea un nuevo factor de
riesgo y se le introducen los
datos entrados, esto si en la
pagina se activa el botón de
Adicionar
ts datos cambiados
ecta
se le dice a la instacia
que adicione un nuevo
factor de riesgo.


: Cliente principal
: Gestionar Historia Clíni...
: historia clínica personal
: Crear Historia Clínica Personal
: PacienteUCI
: Historia Clínica Personal DAO
1-Crear Historia clínica personal
[si esta] Mensaje de error
Tomar paciente de la secoión
pacienteTieneHistoriaClinicaPersonal(paciente):Bool
[si no esta] mostrar pagina de crear
crear
insertarHistoriaClinicaPersonal(historiaClinicaPersonal):HistoiaClinicaPers...
insertarOModificarDomicilios(idHistoriaClinicaPersonal):Lista
insertarOModificarAntecedentesPatológicoPersonales(idHistoriaClinicaPersonal): Lista
insertarOModificarAntecedentesPatológicoFamiliares(idHistoriaClinicaPersonal): Lista
insertarOModificarEsquemaDeVabunacion(idHistoriaClinicaPersonal): Lista
insertarOModificarMenorDeUnAño(idHistoriaClinicaPersonal): Lista
insertarOModificarFactoresDeRiesgos(idHistoriaClinicaPersonal): Lista
Mostrar
Crear
Diagrama de secuencia 1 del caso de uso Gestionar Historia Clínica Personal

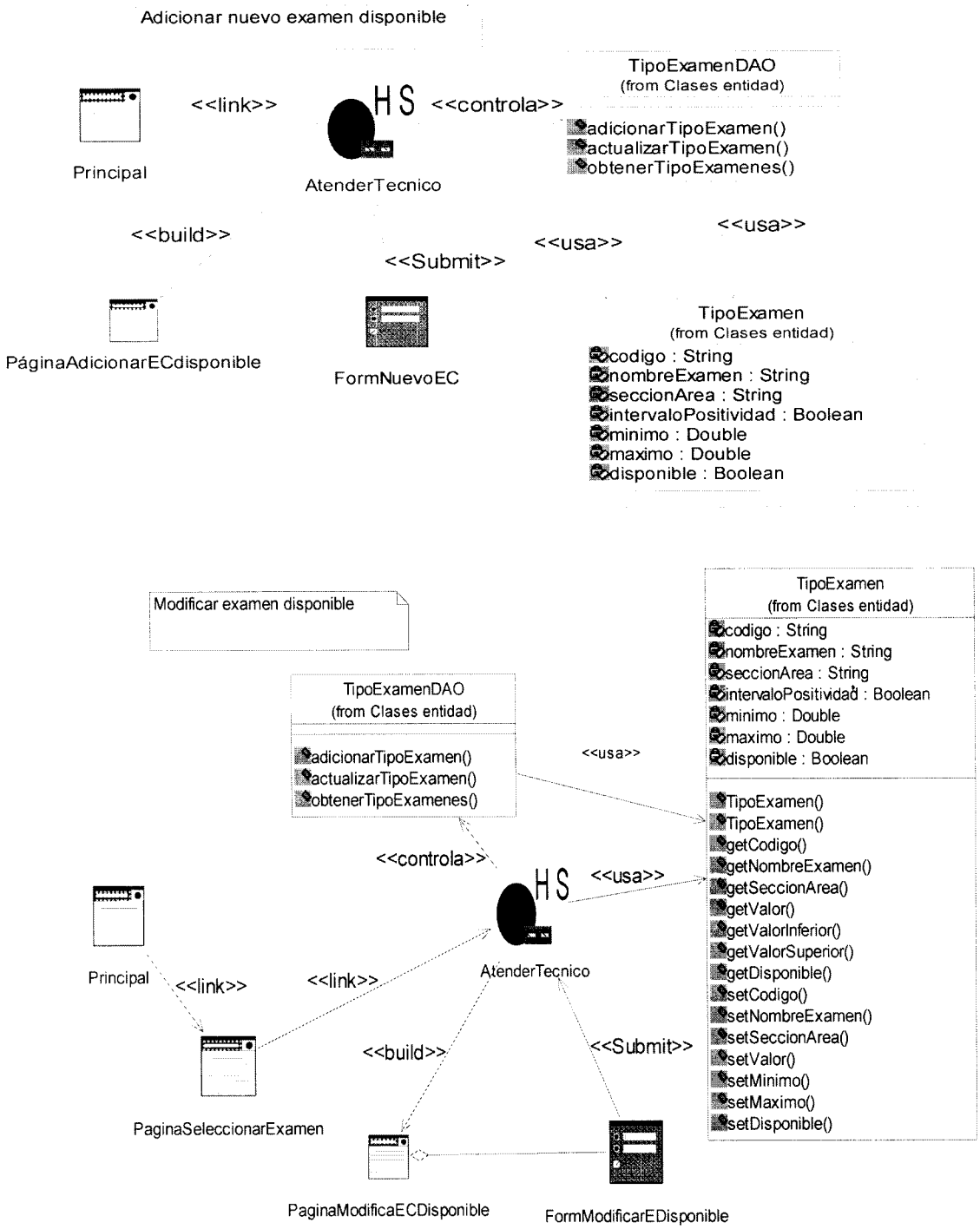

historiaClinicaPersonal : HistoriaClinicaPersonal
: Domicilio
AntecedentesPatologicosPersonales : AntecedentesPatologicosPersonales
AntecedentesPatologicosFamiliares : AntecedentesPatologicosFamiliares
EsquemaVacunacion : EsquemaVacunacion
MenorDeUnAño : MenorDeUnAño
: FactoresRiesgos
crear
crear
crear
crear
crear
crear

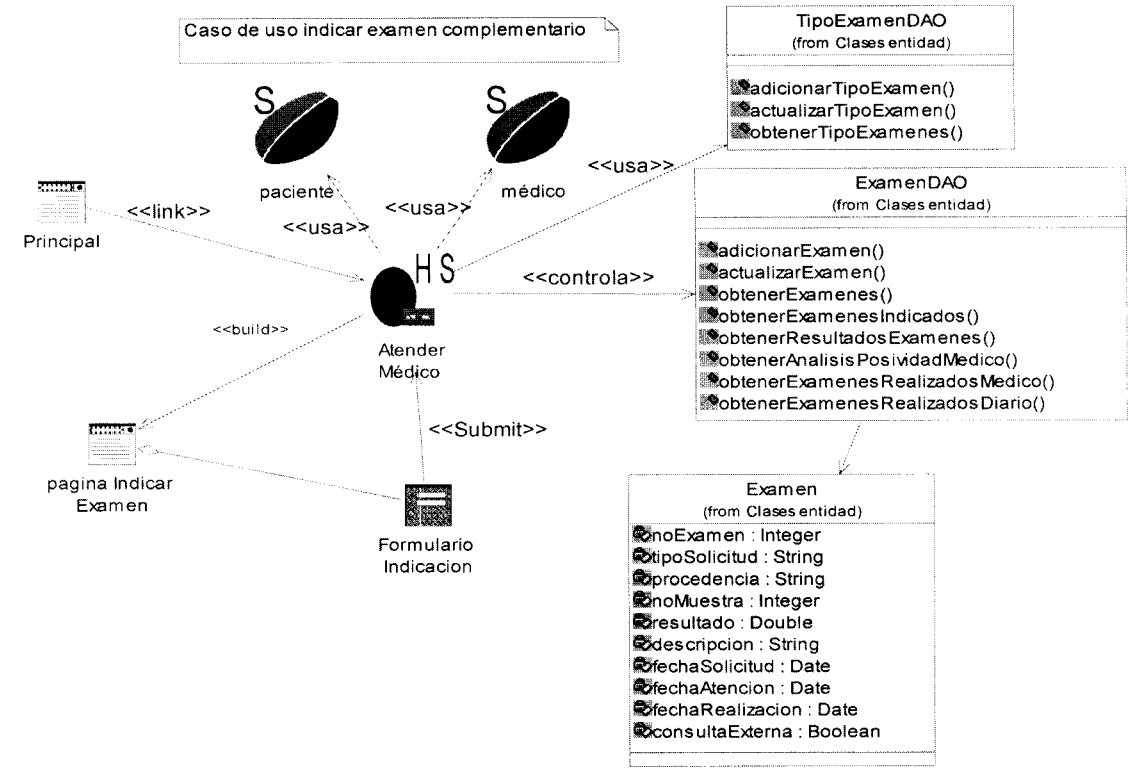

: Cliente principal
: Gestionar Historia Clíni...
: historia clínica personal
: Modificar Historia Clínica Personal
: PacienteUCI
: Historia (
Toma paciente de la sección
pacienteTieneHistoriaClinicaPersonal(paciente):Bool
2- Modificar Historia Clìnica
[si esta]mostrarHistoriaClinicaPersonal(historiaClinicaPersonal):HistoriaClinicaPers...
[si no esta] Mensaje si desea crearla
mostrarDomicilios(idHistoriaClinicaPersonal):Lista
mostrarAntecedentesPatológicoPersonales(idHistoriaClinicaPersonal): Lista
Diagrama de secuencia 2 del caso de uso Gestionar Historia Clínica Personal
mostrarAntecedentesPatológicoFamiliares(idHistoriaClinicaPersonal): Lista
mostrarEsquemaDeVacunacion(idHistoriaClinicaPersonal): Lista
mostrarFactoresDeRiesgos(idHistoriaClinicaPersonal): Lista
mostrarMenorDeUnAño(idHistoriaClinicaPersonal): Lista
crear
mostrar
actualiza
modificarHistoriaClinicaPersonal(HistoriaClinicaPerso...
si se activa la opcion de Cancelar o Modificar
insertarOModificarDomicilios(idHistoriaClinicaPersonal):Lista
insertarOModificarAntecedentesPatológicoPersonales(idHistoriaClinicaPersonal): Lista
insertarOModificarAntecedentesPatológicoFamiliares(idHistoriaClinicaPersonal): Lista
insertarOModificarEsquemaDeVacunacion(idHistoriaClinicaPersonal): Lista
mostrarMenorDeUnAño(idHistoriaClinicaPersonal): Lista
insertarOModificarFactoresDeRiesgos(idHistoriaClinicaPersonal): Lista
mostrar

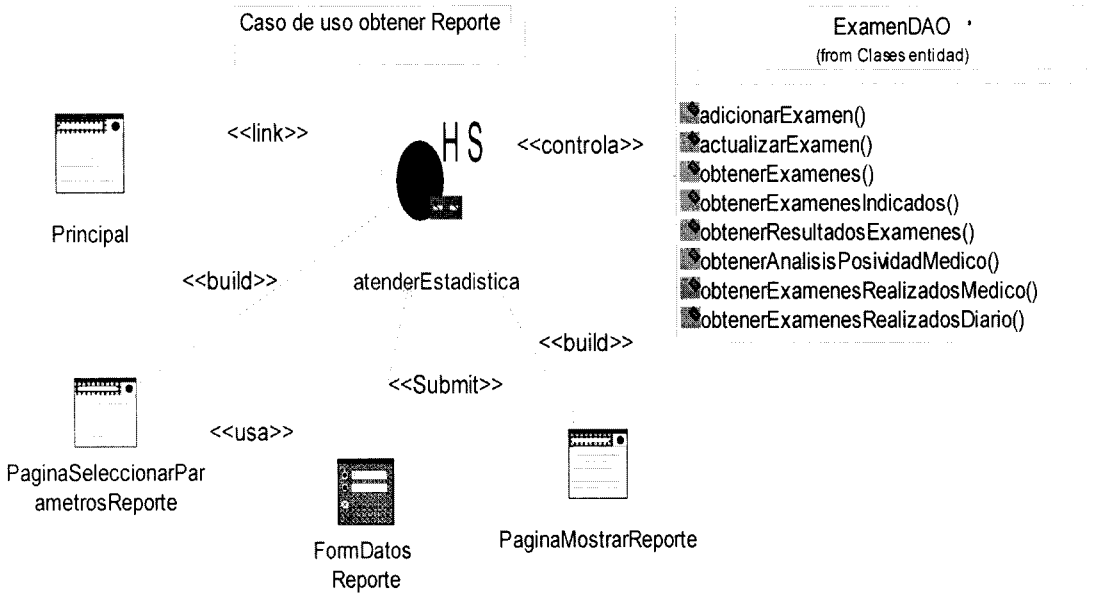

Clinica Personal DAO
si se activa la opción de Modificar
crear
historiaClinicaPersonal : HistoriaClinicaPersonal
crear
crea
: Domicilio
crear
crea
AntecedentesPatologicosPersonales
crear
crea
AntecedentesPato

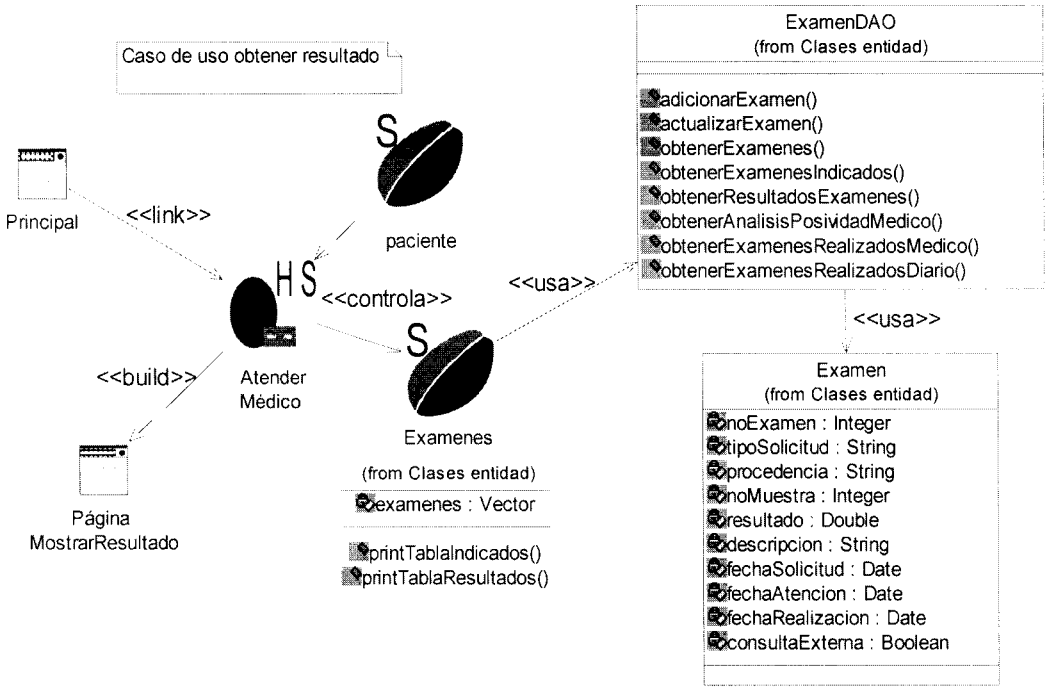

logicosFamiliares
EsquemaVacunacion
MenorDeUnAño
: FactoresRiesgos
crear
crea
crear
crea
crear
crea

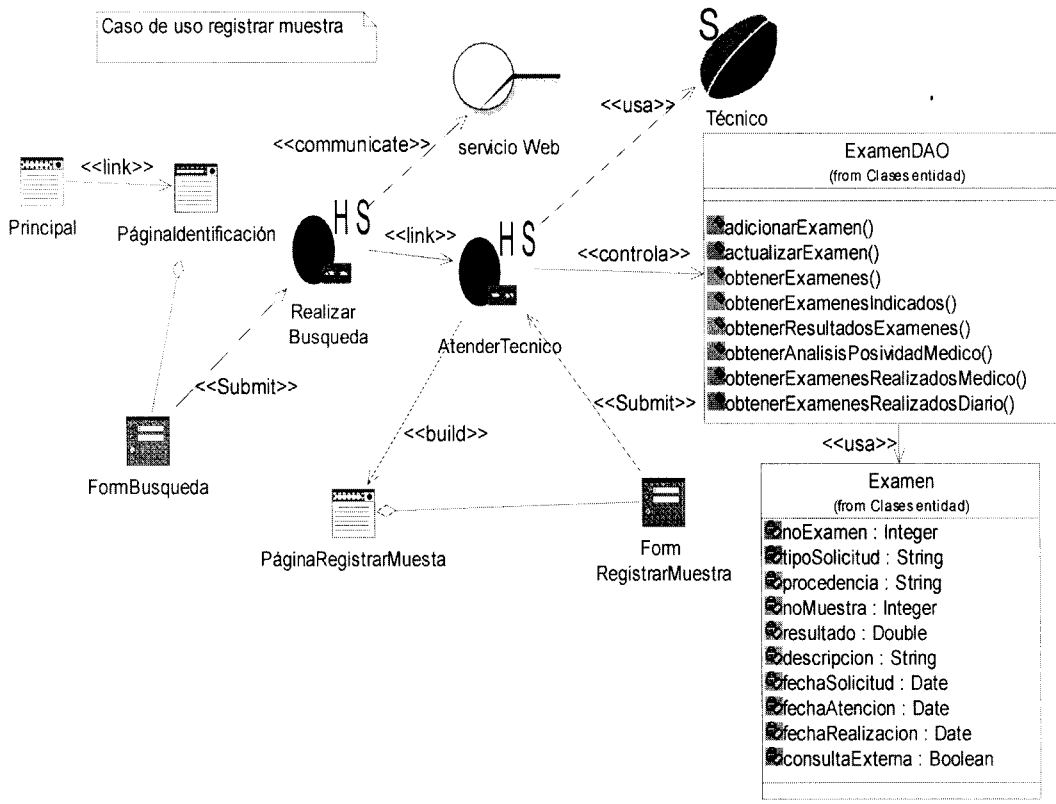
Anexo 17. Diagrama de diseño Web del sistema. (MLC).

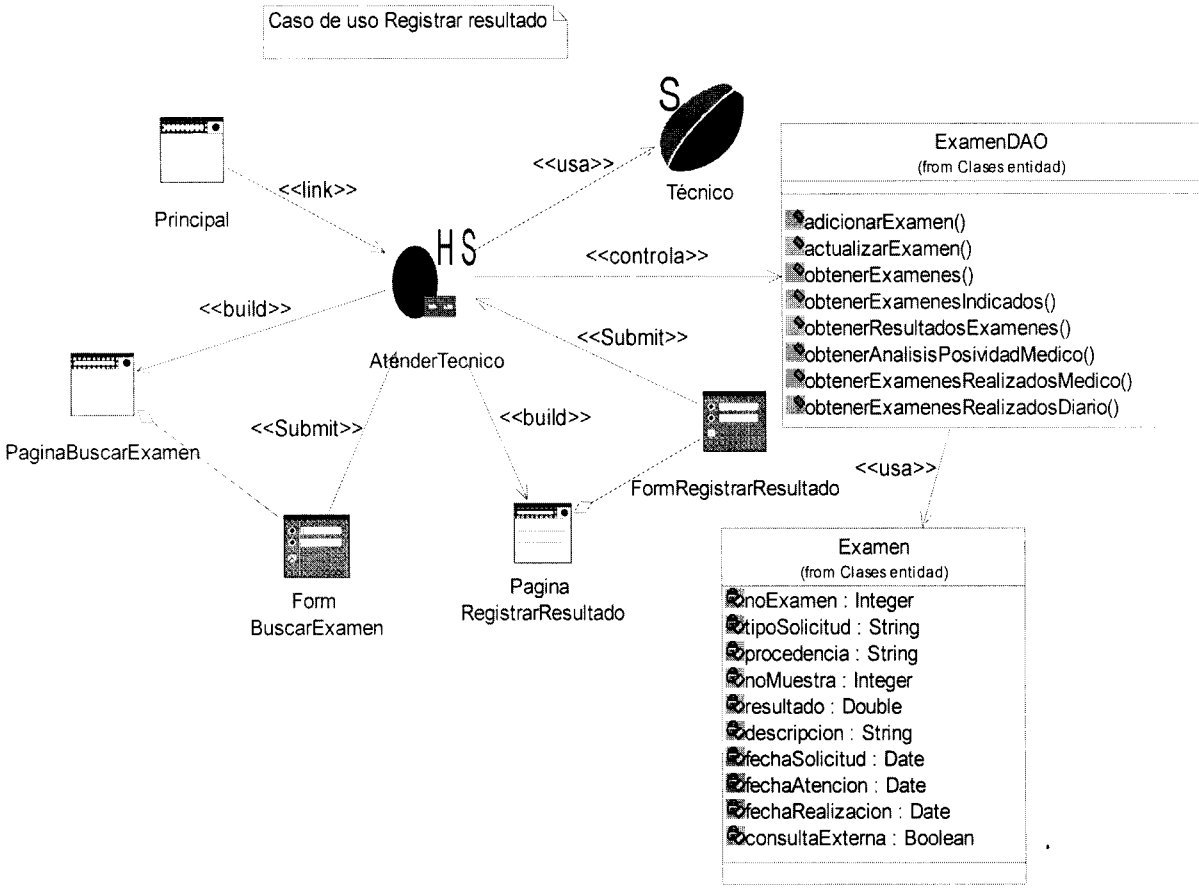



Caso de uso indicar examen complementario
TipoExamenDAO
(from Clases entidad)
adicionarTipoExamen()
actualizarTipoExamen()
obtenerTipoExamenes()
paciente
médico
<<usa>>
<<usa>>
<<usa>>
<<link>>
Principal
ExamenDAO
(from Clases entidad)
adicionarExamen()
actualizarExamen()
obtenerExamenes()
obtenerExamenesIndicados()
obtenerResultadosExamenes()
obtenerAnalisisPosividadMedico()
obtenerExamenesRealizadosMedico()
obtenerExamenesRealizadosDiario()
<<controla>>
<<build>>
Atender
Médico
<<Submit>>
pagina Indicar
Examen
Formulario
Indicacion
Examen
(from Clases entidad)
noExamen : Integer
tipoSolicitud : String
procedencia : String
noMuestra : Integer
resultado : Double
descripcion : String
fechaSolicitud : Date
fechaAtencion : Date
fechaRealizacion : Date
consultaExterna : Boolean


Caso de uso obtener Reporte
ExamenDAO
(from Clases entidad)
adicionarExamen()
actualizarExamen()
obtenerExamenes()
obtenerExamenesIndicados()
obtenerResultadosExamenes()
obtenerAnalisisPosividadMedico()
obtenerExamenesRealizadosMedico()
obtenerExamenesRealizadosDiario()
<<link>>
<<controla>>
Principal
<<build>>
atenderEstadistica
<<build>>
<<Submit>>
<<usa>>
PaginaSeleccionarPar
ametrosReporte
FormDatos
Reporte
PaginaMostrarReporte


Caso de uso obtener resultado
Principal
<<link>>
paciente
HS
<<controla>>
<<usa>>
<<build>>
Atender
Médico
Página
MostrarResultado
Examenes
(from Clases entidad)
examenes : Vector
printTablaIndicados()
printTablaResultados()
ExamenDAO
(from Clases entidad)
adicionarExamen()
actualizarExamen()
obtenerExamenes()
obtenerExamenesIndicados()
obtenerResultadosExamenes()
obtenerAnalisisPosividadMedico()
obtenerExamenesRealizadosMedico()
obtenerExamenesRealizadosDiario()
<<usa>>
Examen
(from Clases entidad)
noExamen : Integer
tipoSolicitud : String
procedencia : String
noMuestra : Integer
resultado : Double
descripcion : String
fechaSolicitud : Date
fechaAtencion : Date
fechaRealizacion : Date
consultaExterna : Boolean


Caso de uso registrar muestra
<<communicate>>
servicio Web
<<usa>>
Técnico
Principal
<<link>>
PáginaIdentificación
HS
<<link>>
HS
<<controla>>
Realizar
Busqueda
AtenderTecnico
<<Submit>>
<<Submit>>
<<build>>
FormBusqueda
PáginaRegistrarMuesta
Form
RegistrarMuestra
ExamenDAO
(from Clases entidad)
adicionarExamen()
actualizarExamen()
obtenerExamenes()
obtenerExamenesIndicados()
obtenerResultadosExamenes()
obtenerAnalisisPosividadMedico()
obtenerExamenesRealizadosMedico()
obtenerExamenesRealizadosDiario()
<<usa>>
Examen
(from Clases entidad)
noExamen : Integer
tipoSolicitud : String
procedencia : String
noMuestra : Integer
resultado : Double
descripcion : String
fechaSolicitud : Date
fechaAtencion : Date
fechaRealizacion : Date
consultaExterna : Boolean


Caso de uso Registrar resultado
Principal
<<link>>
<<usa>>
Técnico
ExamenDAO
(from Clases entidad)
adicionarExamen()
actualizarExamen()
obtenerExamenes()
obtenerExamenesIndicados()
obtenerResultadosExamenes()
obtenerAnalisisPosividadMedico()
obtenerExamenesRealizadosMedico()
obtenerExamenesRealizadosDiario()
<<controla>>
<<build>>
<<Submit>>
AtenderTecnico
PaginaBuscarExamen
<<Submit>>
<<build>>
FormRegistrarResultado
<<usa>>
Form
BuscarExamen
Pagina
RegistrarResultado
Examen
(from Clases entidad)
noExamen : Integer
tipoSolicitud : String
procedencia : String
noMuestra : Integer
resultado : Double
descripcion : String
fechaSolicitud : Date
fechaAtencion : Date
fechaRealizacion : Date
consultaExterna : Boolean

Anexo 18. Diagrama de diseño Web del sistema. (MHCP).

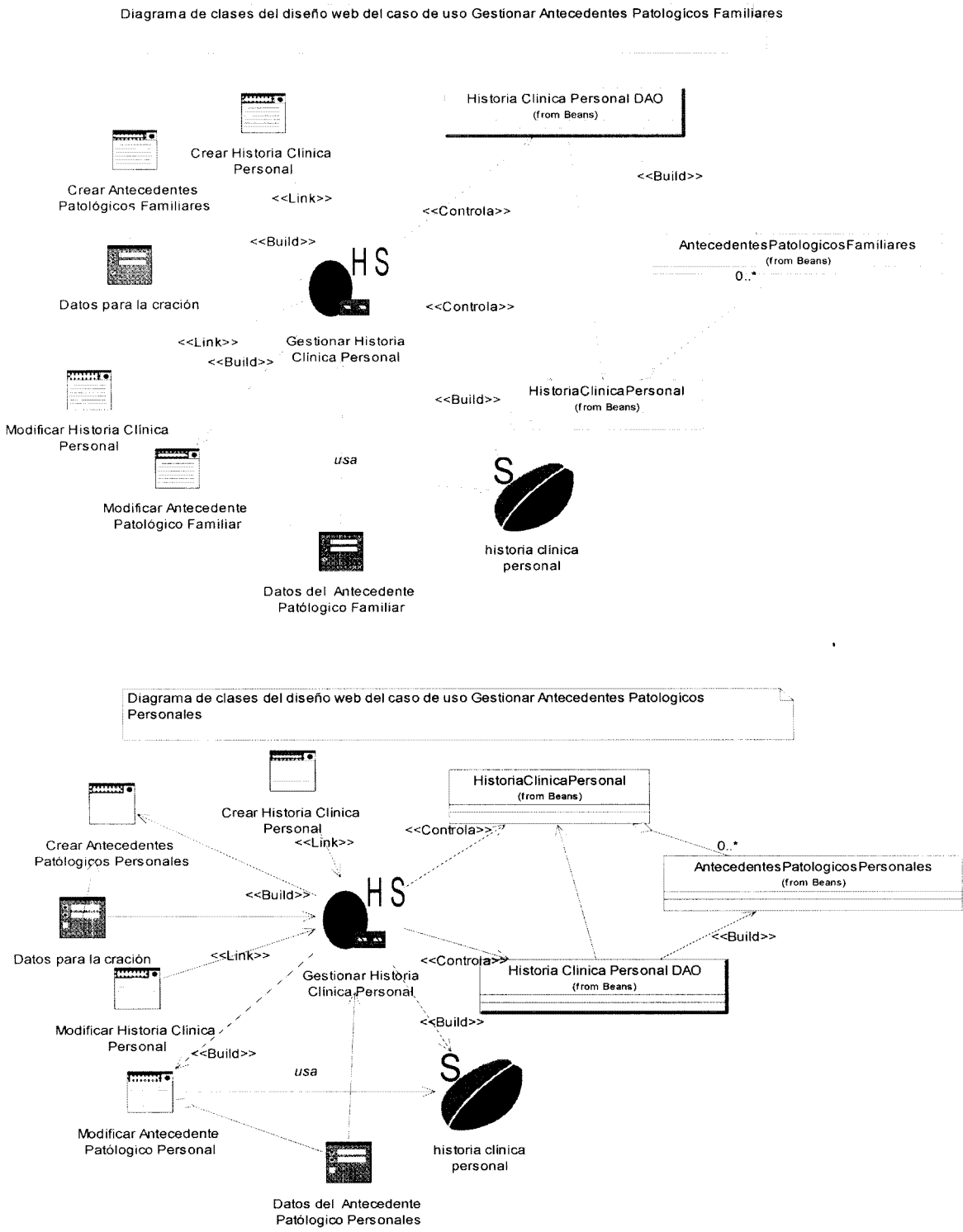


Diagrama de clases del diseño web del caso de uso Gestionar Domicilios

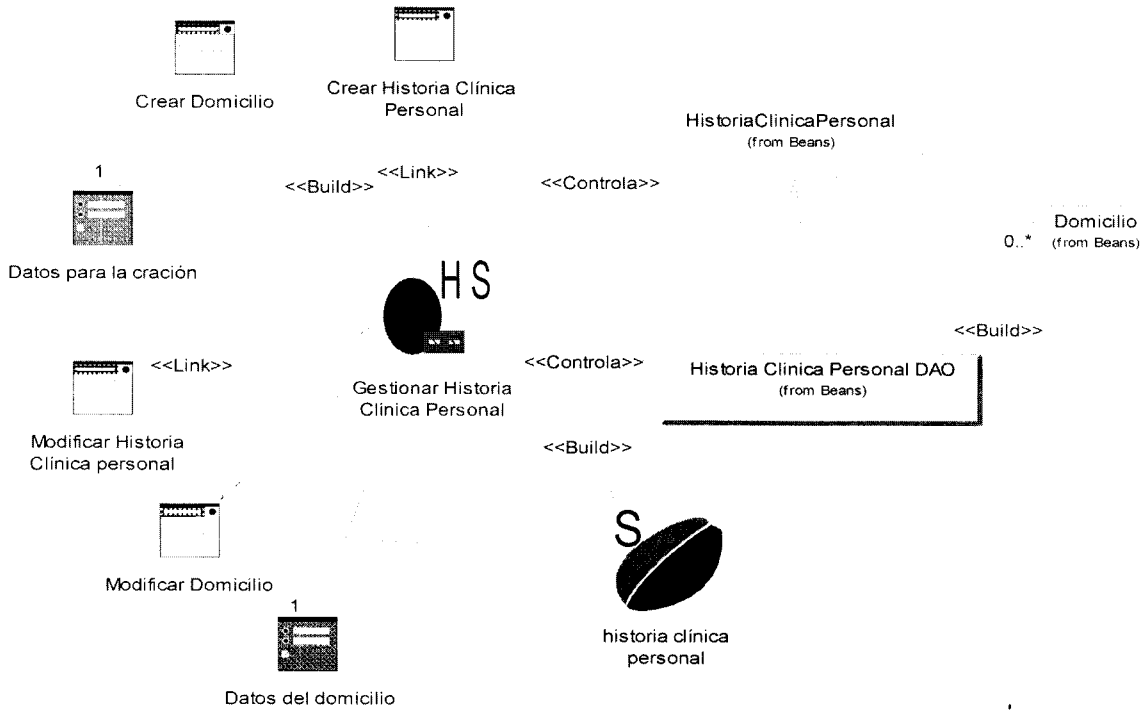
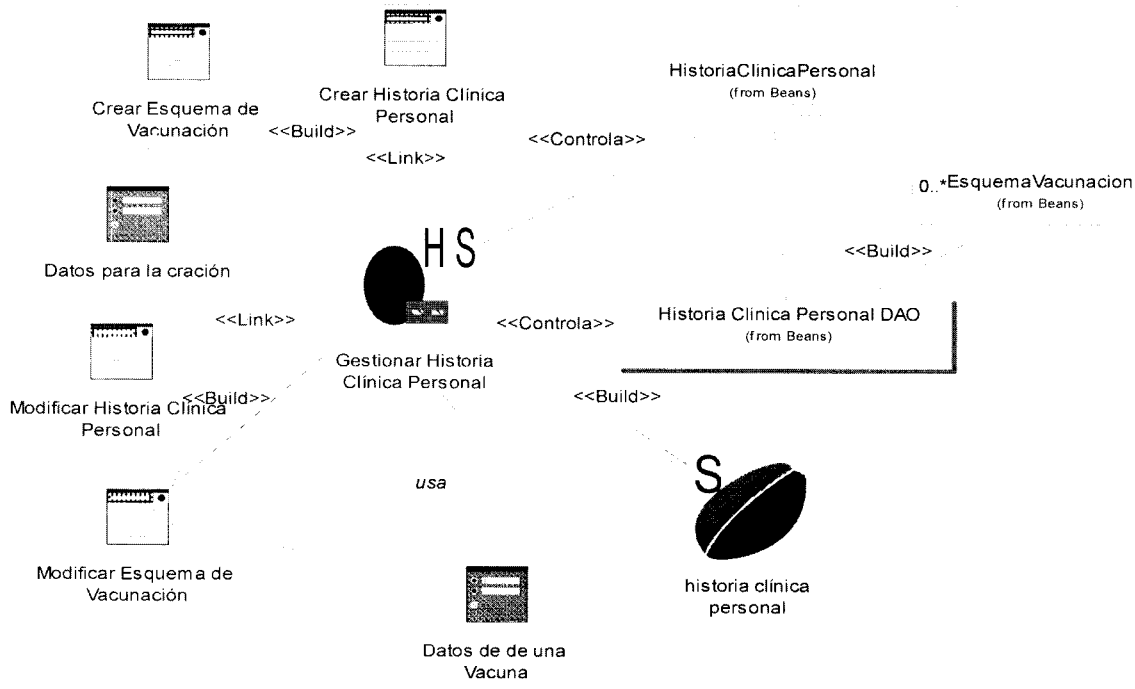


Diagrama de clases del diseño web del caso de uso Gestionar Esquema de Vacunación



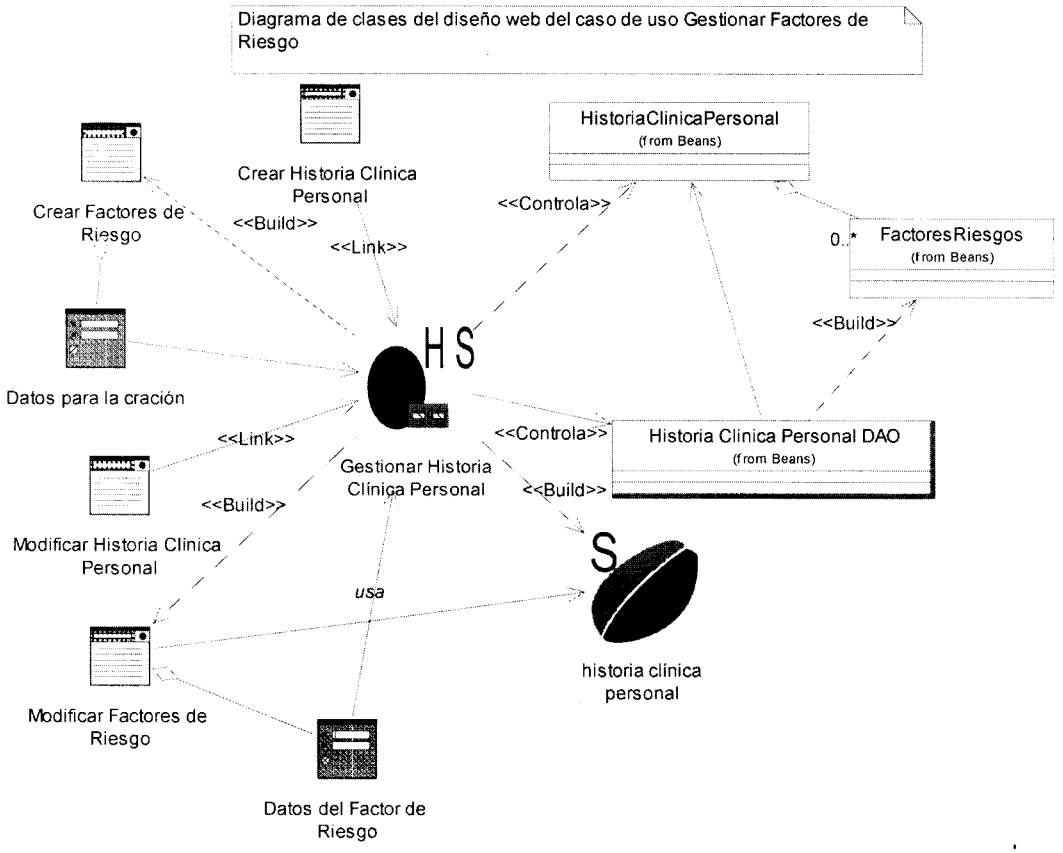


Diagrama de clases del diseño web del caso de uso Gestionar Factores de Riesgo
HistoriaClinicaPersonal
(from Beans)
Crear Historia Clínica Personal
Crear Factores de Riesgo
<<Build>>
<<Link>>
<<Controla>>
0..*
FactoresRiesgos
(from Beans)
<<Build>>
Datos para la cración
HS
<<Link>>
<<Controla>>
Historia Clinica Personal DAO
(from Beans)
Gestionar Historia Clínica Personal
<<Build>>
<<Build>>
Modificar Historia Clínica Personal
S
usa
historia clínica personal
Modificar Factores de Riesgo
Datos del Factor de Riesgo

Diagrama de clases del diseño web del caso de uso Gestionar Menor de un Año

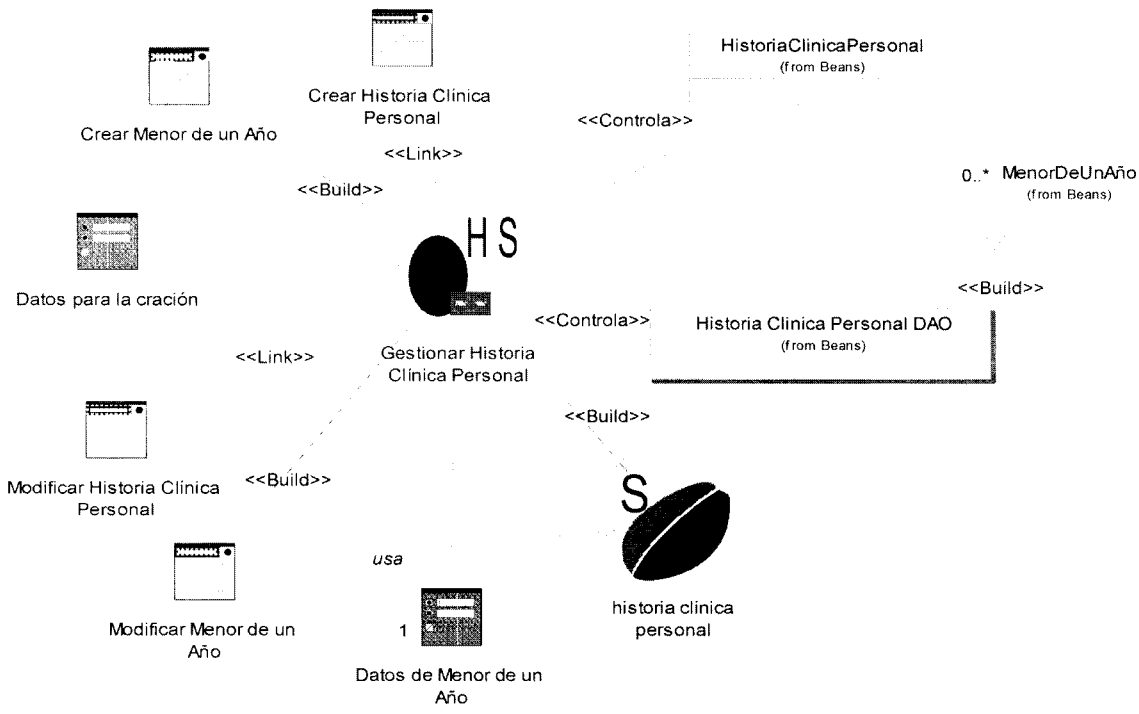
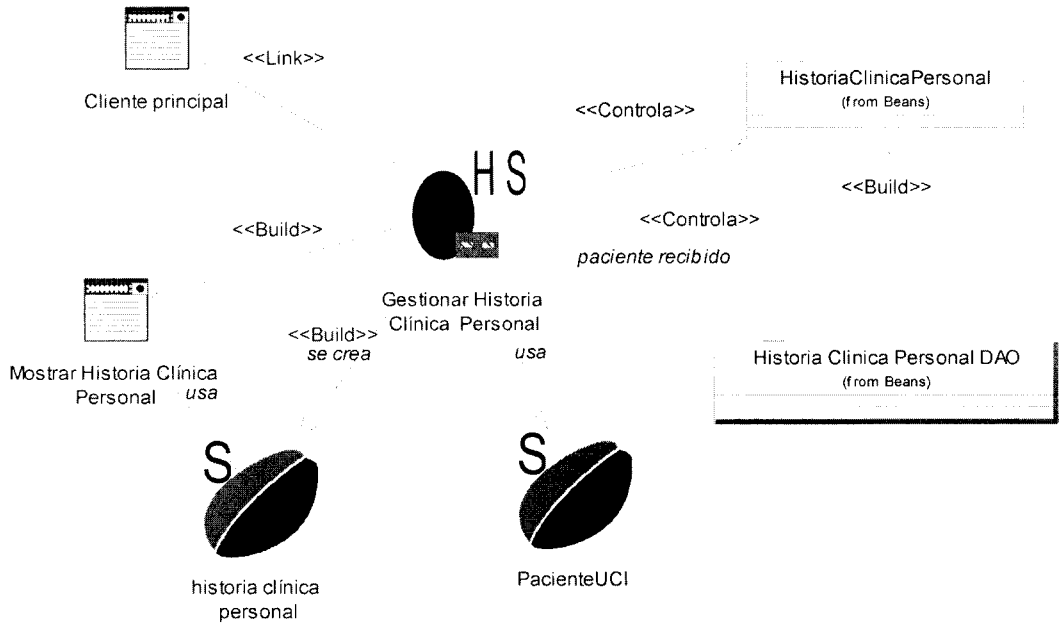


Diagrama de clases del diseño web del caso de uso Mostrar Historia Clínica Personal



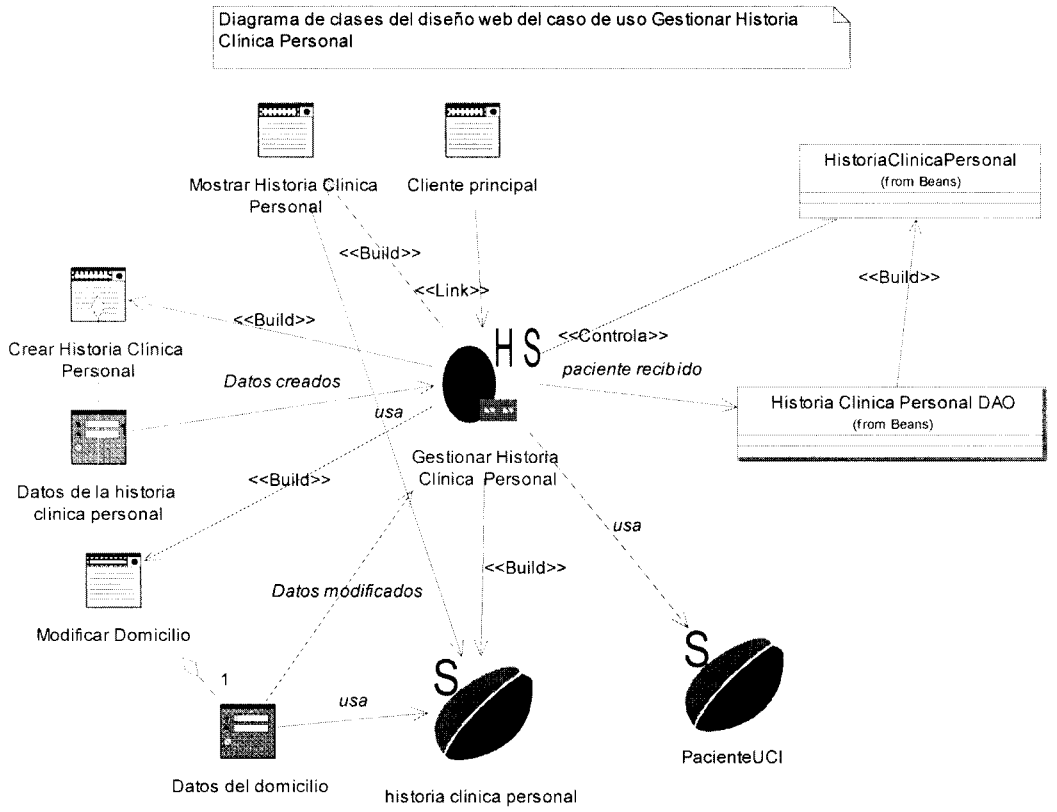


Diagrama de clases del diseño web del caso de uso Gestionar Historia Clínica Personal
Mostrar Historia Clinica Personal
Cliente principal
HistoriaClinicaPersonal
(from Beans)
<<Build>>
<<Link>>
<<Build>>
Crear Historia Clinica Personal
<<Build>>
<<Controla>>
paciente recibido
Datos creados
usa
Historia Clinica Personal DAO
(from Beans)
Gestionar Historia Clinica Personal
Datos de la historia clinica personal
<<Build>>
usa
<<Build>>
Datos modificados
Modificar Domicilio
1
usa
PacienteUCI
Datos del domicilio
historia clinica personal

Anexo 19. Clases de Control (MLC).

Nombre: Atender Médico	
Tipo de clase : controladora	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	IndicarExamen()
Descripción:	Le permite al médico indicarle un examen a un paciente.
Nombre:	ObtenerResultados()
Descripción:	Le permite al médico ver los resultados de los exámenes que han sido indicados por el a un paciente.

Nombre: Atender Técnico	
Tipo de clase : controladora	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	AdicionarTipoExamen()
Descripción:	Le permite al insertar un nuevo tipo de examen en el sistema.
Nombre:	ActualizarTipoExamen()
Descripción:	Le permite al técnico modificar un tipo de examen existente.
Nombre:	ActualizarExamen()
Descripción:	Le permite al técnico modificar un examen, puede definir el número de muestra que tiene este examen y escribir el resultado del examen con una descripción.

Nombre: Atender Estadística	
Tipo de clase : controladora	
Atributo	**Tipo**
Para cada responsabilidad:	

Nombre:	ObtenerReportes()
Descripción:	Le permite al personal de estadística visualizar los reportes que se generan en el laboratorio clínico.

Nombre: Realizar Búsqueda	
Tipo de clase : controladora	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	BuscarPaciente()
Descripción:	Le permite al técnico buscar los datos del paciente.

Anexo 20. Clases de Control. (MHCP).

Nombre: gestionarHistoriaClinicaPersonal	
Tipo de clase : controladora	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	Mostrar historia clínica personal.
Descripción:	Le permite al sistema que solicito el servicio de mostrarle los datos de la historia clínica personal de un paciente en una página dentro de esta todos los datos referentes a domicilio, antecedentes patológicos familiares y personales, consultas realizadas, tratamiento de cada una de esta, los factores de riesgo, el esquema de vacunación y los datos para menor de un año, si existen, en caso de que el paciente tenga una historia clínica asociada, si no le muestra la opción de crearle una. La búsqueda en la base de datos la realiza la clase HistoriaClinicaPersonalDAO controlado también por esta clase. Los datos recopilados los introduce en sección por el beans de sección.

Nombre:	Modificar historia clínica personal.
Descripción:	Le permite al sistema que lo solicito mostrarle todos los datos de la historia clínica personal de un paciente con la acción de modificación a cada uno de ellos, en caso de que el paciente tenga historia clínica personal asociada. Después que se de la orden de modificación este toma todos los datos y los envía a modificarlos en la base de datos, con la clase HistoriaClinicaPersonalDAO.
Nombre:	Modificar domicilio
Descripción:	Le muestra al usuario una página con los datos del domicilio que se quiere modificar, si se activa la opción de modificar se recogen los datos y se introducen al la posición especifica en la lista del beans de HistoriaClinicaPersonal y se introduce este en la sección. Si se activa Cancelar se muestra la página de modificar historia clínica.
Nombre:	Modificar antecedentes patológicos personales.
Descripción:	Le muestra al usuario una página con los datos del antecedente patológico personal que se quiere modificar, si se activa la opción de modificar se recogen los datos y se introducen al la posición especifica en la lista del beans de HistoriaClinicaPersonal y se introduce este en la sección. Si se activa Cancelar se muestra la página de modificar historia clínica.
Nombre:	Modificar factores de riesgo.
Descripción:	Le muestra al usuario una página con los datos del factor de riego que se quiere modificar, si se activa la opción de modificar se recogen los datos y se introducen al la posición especifica en la lista del beans de HistoriaClinicaPersonal y se introduce este en la sección. Si se activa Cancelar se muestra la página de modificar historia clínica.
Nombre:	Modificar antecedentes patológicos familiares.
Descripción:	Le muestra al usuario una página con los datos del antecedente patológico familiar que se quiere modificar, si

	se activa la opción de modificar se recogen los datos y se introducen al la posición especifica en la lista del beans de HistoriaClinicaPersonal y se introduce este en la sección. Si se activa Cancelar se muestra la página de modificar historia clínica.
Nombre:	Modificar esquema de vacunación.
Descripción:	Le muestra al usuario una página con los datos de l vacuna que se quiere modificar que se quiere modificar, si se activa la opción de modificar se recogen los datos y se introducen al la posición especifica en la lista del beans de HistoriaClinicaPersonal y se introduce este en la sección. Si se activa Cancelar se muestra la página de modificar historia clínica.
Nombre:	Crear domicilio.
Descripción:	Le muestra al usuario una página para entrar datos de un domicilio, controlando de que estos datos estén correctos después procede a introducirlos en el bean de sección que es de tipo HistoriaClinicaPersonal, esto en caso de la página mandarle la orden de agregar, en caso contrario nos muestra la página que solicito el servicio.
Nombre:	Crear Antecedentes Patológicos Familiares.
Descripción:	Le muestra al usuario una página para entrar datos de un antecedente patológicos familiares, controlando de que estos datos estén correctos después procede a introducirlos en el bean de sección que es de tipo HistoriaClinicaPersonal, esto en caso de la página mandarle la orden de agregar, en caso contrario nos muestra la página que solicito el servicio.
Nombre:	Crear Factores de Riesgo.
Descripción	Le muestra al usuario una página para entrar datos de un factor de riesgo, controlando de que estos datos estén correctos después procede a introducirlos en el beans de sección que es de tipo HistoriaClinicaPersonal, esto en

	caso de la página mandarle la orden de agregar, en caso contrario nos muestra la página que solicito el servicio.
Nombre:	Crear Esquema de Vacunación.
Descripción	Le muestra al usuario una página para entrar datos de un vacuna, controlando de que estos datos estén correctos después procede a introducirlos en el bean de sección que es de tipo HistoriaClinicaPersonal, esto en caso de la página mandarle la orden de agregar, en caso contrario nos muestra la página que solicito el servicio.
Nombre:	Crear Menor de un Año
Descripción:	Le muestra al usuario una página para entrar datos de un menor de un año, controlando de que estos datos estén correctos después procede a introducirlos en el bean de sección que es de tipo HistoriaClinicaPersonal, esto en caso de la página mandarle la orden de agregar, en caso contrario nos muestra la página que solicito el servicio.
Nombre:	Crear Antecedentes Patológicos Personales
Descripción:	Le muestra al usuario una página para entrar datos de un antecedente patológico personal, controlando de que estos datos estén correctos después procede a introducirlos en el bean de sección que es de tipo HistoriaClinicaPersonal, esto en caso de la página mandarle la orden de agregar, en caso contrario nos muestra la página que solicito el servicio.
Nombre:	Crear historia clínica personal.
Descripción:	Le permite al sistema que lo solicito mostrarle una página con posibilidades de entrada de datos. Primero hace un chequeo para ver si el paciente tiene historia clínica personal asociada usando la clase HistoriaClinicaPersonalDAO, en caso de que tenga se le envía un mensaje que el usuario ya tiene historia clínica asociada. En caso contrario muestra la página de creación.

Anexo 21. Clases de Entidad. (MLC).

Nombre: Examen	
Tipo de clase : entidad	
Atributo	**Tipo**
Paciente	Paciente
Médico	PersonalMédico
tipoExamen	TipoExamen
noExamen	Integer
Procedencia	String
fechaSolicitud	Date
tipoSolicitud	String
consultaExterna	Boolean
fechaAtencion	Date
tecnicoAtiende	PersonalMédico
fechaRealizacion	Date
tecnicoRealiza	PersonalMédico
noMuestra	Integer
Resultado	Double
Descripcion	String
Para cada responsabilidad:	
Nombre:	Examen(paramPaciente, paramTipoexamen, paramMédico, paramNoExamen, paramProcedencia, paramFechaSolicitud, paramTipoSolicitud, paramConsultaExterna, paramFechaAtencion, paramTecnicoAtiende, paramFechaRealizacion, paramTecnicoRealiza, paramNoMuestra, paramResultado, paramDescripcion)
Descripción:	Crea el objeto.
Nombre:	getPaciente():Paciente
Descripción:	Devuelve el objeto paciente.
Nombre:	setPaciente(paramPaciente)
Descripción:	Se coloca el objeto Paciente.
Nombre:	getNombrePaciente():String

Descripción:	Devuelve el nombre del paciente al que pertenece este examen.
Nombre:	getApellidos():String
Descripción:	Devuelve el apellido del paciente.
Nombre:	getDireccion():String
Descripción:	Devuelve la dirección del paciente.
Nombre:	getSexo():String
Descripción:	Devuelve el sexo del paciente.
Nombre:	getEdad():Integer
Descripción:	Devuelve la edad del paciente.
Nombre:	getTipoExamen()TipoExamen
Descripción:	Devuelve el tipo de examen que se le realiza.
Nombre:	getNombreExamen()String
Descripción:	Devuelve el nombre del examen que se realiza.
Nombre:	getNombreMédico():String
Descripción:	Devuelve el nombre del médico que indica el examen.
Nombre:	getEspecialidad():String
Descripción:	Devuelve la especialidad del médico que indica el examen.
Nombre:	getArea():String
Descripción:	Devuelve área del médico que indica el examen.
Nombre:	getNoExamen():Integer
Descripción:	Devuelve el número del examen realizado al paciente.
Nombre:	setNoExamen(noExamen)
Descripción:	Coloca el número del examen realizado al paciente.
Nombre:	getFechaSolicitud()Date
Descripción:	Devuelve la fecha en que fue indicado el examen por el médico.
Nombre:	setFechaSolicitud(fecha)
Descripción:	Se escribe la fecha en que fue indicado el examen por el médico.
Nombre:	getProcedencia()String
Descripción:	Devuelve el lugar de procedencia de la indicación.
Nombre:	setProcedencia(procedencia)
Descripción:	Se escribe el lugar de procedencia de la indicación.
Nombre:	getConsultaExterna()Boolean

Descripción:	Devuelve si el examen fue indicado en una consulta externa.
Nombre:	setConsultaExterna(consulta)
Descripción:	Se escribe si el examen fue indicado desde una consulta externa.
Nombre:	getFechaAtencion():Date
Descripción:	Devuelve la fecha de recepción de la muestra.
Nombre:	setFechaAtencion(fecha)
Descripción:	Se registra la fecha de la recepción de la muestra.
Nombre:	getNombreTecnicoAtiende()String
Descripción:	Devuelve el nombre del tecnico que realizó la recepción de la muestra.
Nombre:	setTecnicoAtiende(tecnico)
Descripción:	Se registra el tecnico que realizó la recepción de la muestra
Nombre:	getFechaRealizacion()Date
Descripción:	Devuelve la fecha en que fue analizada la muestra.
Nombre:	setFechaRealizacion(fecha)
Descripción:	Se escribe la fecha en que fue analizada la muestra del examen.
Nombre:	getNombreTecnicoRealiza()
Descripción:	Devuelve el nombre del tecnico que analiza la muestra y registra el resultado obtenido.
Nombre:	setTecnicoRealiza(tecnico)
Descripción:	Se registra el tecnico que realiza el analisis de la muestra.
Nombre:	getNoMuestra()Integer
Descripción:	Devuelve el número de muestra del examen.
Nombre:	setNoMuestra(nomuestra)
Descripción:	Se escribe el número de muestra que tiene el examen.
Nombre:	getResultado():Double
Descripción:	Devuelve el resultado obtenido.
Nombre:	setResultado(resultado)
Descripción:	Se escribe el resultado obtenido.
Nombre:	getDescripcion()String
Descripción:	Devuelve la descripcion del examen.
Nombre:	setDescripcion(descripcion)

Descripción:	Se escribe la descripcion del examen.
Nombre:	esPositivo()Boolean
Descripción:	Devuelve un valor positivo si el resultado obtenido en el analisis de la muestra se encuentre en el intervalo de positividad del examen.

Nombre: ExamenDAO	
Tipo de clase : entidad	
Atributo	**Tipo**
Driver	String
url	String
Username	String
Password	String
Para cada responsabilidad:	
Nombre:	adicionarExamen(examen)
Descripción:	Inserta un examen en base de datos, si se realizó la inserción sin problema devuelve true.
Nombre:	actualizarExamen(examen)
Descripción:	Actualiza un examen en la base de datos, si se realizó la actualización sin problema devuelve true.
Nombre:	obtenerExamenes():Vector
Descripción:	Devuelve una lista de todos los examenes.
Nombre:	obtenerExamenesIndicados(idPaciente):Vector
Descripción:	Devuelve los examenes indicados a un paciente.
Nombre:	ObtenerResultadoExamen(idPaciente):Vector
Descripción:	Devuelve los resultados de los examenes realizados a un paciente
Nombre:	obtenerAnalisisPosividadMédico():String
Descripción:	Imprime una tabla con una lista de los médicos que han indicado exámenes, la cantidad de examenes indicados, cantidad de exámenes realizados y positividad en la indicación del examen.
Nombre:	obtenerExamenesRealizadosMédico():String
Descripción:	Imprime una tabla con todos los exámenes realizados por un médico.

Nombre:	obtenerExamenesRealizadosDiario():String
Descripción:	Imprime una Tabla con la cantidad exámenes realizados un día específico separados por el tipo de examen que es.

Nombre: TipoExamen	
Tipo de clase : entidad	
Atributo	**Tipo**
Codigo	Integer
NombreExamen	String
seccionArea	String
IntervaloPositividad	Boolean
Minimo	Double
Maximo	Double
Disponible	Boolean
Para cada responsabilidad:	
Nombre:	TipoExamen(codigo, nombreExamen, seccionArea, intervalo, minimo, maximo, disponible)
Descripción:	Construye el objeto tipo de examen.
Nombre:	getCodigo()Integer
Descripción:	Devuelve el codigo del examen.
Nombre:	setCodigo(codigo)
Descripción:	Se escribe el codigo del examen.
Nombre:	getNombreExamen():String
Descripción:	Devuelve el nombre del examen.
Nombre:	setNombreExamen(nombre)
Descripción:	Se escribe el nombre del examen.
Nombre:	getSeccionArea():String
Descripción:	Devuelve el área donde se realiza el examen.
Nombre:	setSeccionArea(area)
Descripción:	Se escribe el área donde se realiza el examen.
Nombre:	getIntervaloPositividad():Boolean
Descripción:	Devuelve si el examen tiene intervalo de positividad.

Nombre:	setIntervaloPositividad(intervalo)
Descripción:	Se escribe si el examen tiene un intervalo de positividad.
Nombre:	getMinimo():Double
Descripción:	Devuelve el valor mínimo del intervalo.
Nombre:	setMinimo(valor)
Descripción:	Se escribe el valor mínimo del intervalo de positividad.
Nombre:	getMaximo():Double
Descripción:	Devuelve el valor máximo del intervalo.
Nombre:	setMaximo(valor)
Descripción:	Se escribe el valor máximo del intervalo de positividad.
Nombre:	getDisponible():Boolean
Descripción:	Devuelve si el examen esta se puede realizar en este momento en el laboratorio.
Nombre:	setDiponible(disponible)
Descripción:	Se escribe si el examen se puede realizar.

Nombre: TipoExamenDAO	
Tipo de clase : entidad	
Atributo	**Tipo**
Driver	String
url	String
Username	String
Password	String
Para cada responsabilidad:	
Nombre:	AdicionarTipoExamen(tipoexamen)
Descripción:	Inserta un tipo de examen en la base de datos.
Nombre:	ActualizarTipoExamen(tipoexamen)
Descripción:	Actualiza un tipo de examen en la base de datos.
Nombre:	MostrarTiposExamen(tipoexamen):Integer
Descripción:	Muestra una lista con todos los tipos de examen existente en el laboratorio

Nombre: Examenes	
Tipo de clase : entidad	
Atributo	**Tipo**
Lista	Vector
Para cada responsabilidad:	
Nombre:	Examenes()
Descripción:	Crea un objeto Examenes que tiene una lista de los examenes de un paciente.
Nombre:	MostrarExamenesIndicados(idpaciente):String
Descripción:	Muestra los examenes indicados a un paciente.
Nombre:	MostrarResultadosExamenes()
Descripción:	Muestra los resultados de los examenes realizados a un paciente.

Nombre: Paciente	
Tipo de clase : entidad	
Atributo	**Tipo**
IdPaciente	String
Nombre	String
Apellidos	String
Sexo	String
Direccion	String
Para cada responsabilidad:	
Nombre:	Paciente(idPaciente, nombre, apellidos, sexo, direccion)
Descripción:	Crea el objeto paciente.
Nota :	Tiene también todos los get y set para cada atributo...

Nombre: PersonalMédico	
Tipo de clase : entidad	
Atributo	**Tipo**
Idpersona	String
Nombre	String
Especialidad	String

Area	String
Para cada responsabilidad:	
Nombre:	PersonalMédico(idpersona, nombre, especialidad, area)
Descripción:	Se crea un objeto de tipo personal médico.
Nota:	Los get y set de todos los atributos.....

Anexo 22. Clases de Entidad. (MHCP).

Nombre: HistoriaClinicaPersonal	
Tipo de clase : entidad	
Atributo	**Tipo**
idHistoriaClinicaPersonal	String
paciente	Paciente
listaConsulta	Vector
listaDomicilio	Vector
listaMenorDeUnAño	Vector
listaAntecedentesPatológicosFamiliares	Vector
listaAntecedentesPatológicosPersonales	Vector
listaEsquemaVacunacion	Vector
listaFactoresRiesgo	Vector
grupoSanguineo	String
factorRH	String
alergia	String
grupoDispensalarial	String
Para cada responsabilidad:	
Nombre:	getIdHistoriaClinicaPersonal():String
Descripción:	Retorna el id de la historia clínica personal
Nombre:	HistoriaClinicaPersonal()
Descripción:	Es uno de los constructores de la clase, dentro de el se crean cada una de las listas.
Nombre:	HistoriaClinicaPersonal(param : Paciente)
Descripción:	A este constructor se le pasa el paciente y dentro se crean cada una de las listas.

Nombre:	printlnTableMostrarAntecedentesPatológicosPersonales(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los antecedentes patológicos personales con sus valores que están en la listaAntecedentesPatológicosPersonales.
Nombre:	printlnTableModificarAntecedentesPatológicosPersonales(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los antecedentes patológicos personales con sus valores que están en la listaAntecedentesPatológicosPersonales y con los link a la página de modificar antecedentes patológicos personales.
Nombre:	printlnTableMostrarAntecedentesPatológicosFamiliares(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los antecedentes patológicos familiares con sus valores que están el la listaAntecedentesPatológicosFamiliares.
Nombre:	printlnTableModificarAntecedentesPatológicosFamiliares(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los antecedentes patológicos familiares con sus valores que están en la listaAntecedentesPatológicosPersonales y con los links a la página de modificar antecedentes patológicos familiares.
Nombre:	printlnTableMostrarDomicilios(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los domicilios y sus valores que están en la listaDomicilio.
Nombre:	printlnTableModificarDomicilios(): String

Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los domicilios y sus valores que están en la listaDomicilio y con los links a la página de modificar domicilio.
Nombre:	printlnTable1MostrarFactoresRiesgo(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los factores de riesgo y sus valores que están en la listaFactoresRiesgos, solo la primera parte de los datos.
Nombre:	printlnTable1ModificarFactoresRiesgo(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los factores de riesgo y sus valores que están en la listaFactoresRiesgos, solo la primera parte de los datos y con los links a la página de modificar factores de riesgo.
Nombre:	printlnTable2MostrarFactoresRiesgo(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los factores de riesgo y sus valores que están en la listaFactoresRiesgos, solo la parte restante de los datos.
Nombre:	printlnTable2ModificarFactoresRiesgo(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de los factores de riesgo y sus valores que están en la listaFactoresRiesgos, solo la parte restante de los datos y con los links a la página de modificar factores de riesgo.
Nombre:	printlnTableMostrarEsquemaVacunacion(): String
Descripción:	Esta función devuelve una string con los tags HTML para la

	impresión en la página de la tabla del esquema de vacunación y sus valores que están en la listaEsquemaVacunacion,
Nombre:	printlnTableModificarEsquemaVacunacion(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla del esquema de vacunación y sus valores que están en la listaEsquemaVacunacion y con los links a la página de modificar esquema de vacunación,
Nombre:	printlnTableMostrarConsultas(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la tabla de las consultas y sus valores que están en la listaConsulta.
Nombre:	printlnTableMostrarTratamientos(): String
Descripción:	Esta función devuelve una string con los tags HTML para la impresión en la página de la de los tratamientos por consulta y sus valores.
Nombre:	getDomicilioConsultorio(param:int): String
Descripción:	Esta función devuelve el consultorio del domicilio que esta en la posición param de la listaDomicilios. Nota: Se implementó un get por todos los atributos da la clase Domicilio, lo que cambia solamente es el tipo del valor que devuelve la función, por el tipo del atributo a que se hace referencia.
Nombre:	getAntecedentesPatológicosFamiliaresParentesco(param:int):String
Descripción:	Esta función devuelve el parentesco del antecedente patológico familiar en la posición param de la listaAntecedentesPatológicosFamiliares.

	Nota: Se implementó un get por todos los atributos da la clase AntecedentesPatológicosFamiliares, lo que cambia solamente es el tipo del valor que devuelve la función, por el tipo del atributo a que se hace referencia.
Nombre:	getAntecedentesPatológicosPersonalesDiagnostico(param:int): String
Descripción:	Esta función devuelve el diagnostico del antecedente patológico personal en la posición param de la listaAntecedentesPatológicosPersonales. Nota: Se implementó un get por todos los atributos da la clase AntecedentesPatológicosPersonales, lo que cambia solamente es el tipo del valor que devuelve la función, por el tipo del atributo a que se hace referencia.
Nombre:	getEsquemaVacunaciónfecha(param:int):String
Descripción:	Esta función devuelve la fecha en que el paciente se puso la vacuna en la posición param de la listaEsquemaVacunacion. Nota: Se implementó un get por todos los atributos da la clase EsquemaVacunacion, lo que cambia solamente es el tipo del valor que devuelve la función, por el tipo del atributo a que se hace referencia.
Nombre:	getFactoresRiesgosTensionArterialMinima(param:int):String
Descripción:	Esta función devuelve la tensión arterial máxima del factor de riego en la posición param de la listaFactoresRiesgo. Nota: Se implementó un get por todos los atributos da la clase FactoresRiesgo, lo que cambia solamente es el tipo del valor que devuelve la función, por el tipo del atributo a que se hace referencia.

Nombre:	getMenorDeUnAñoTalla(param:int):Float
Descripción:	Esta función devuelve la talla del menor de un año en la posición param de la listaMenorDeUnAño. Nota: Se implementó un get por todos los atributos da la clase MenorDeUnAño, lo que cambia solamente es el tipo del valor que devuelve la función, por el tipo del atributo a que se hace referencia.
Nombre:	getAntecedentesPatológicosPersonales(): Vector
Descripción:	Esta función devuelve un vector que es una lista de objetos, en este caso es la listaAntecedentesPatológicosPersonales. Nota: Se implementó un método get por cada atributo de esta clase, lo que cambia es el tipo del valor que devuelve la función, que esta dada por el tipo del atributo al que se referencia.
Nombre:	setAntecedentesPatológicosFamiliares(param:Vector):void
Descripción:	Esta función sobre escribe el valor de la listaAntecedentesPatológicosFamiliares por el valor de param. Nota: Se implementó un método set por cada atributo de esta clase, solo cambia la lista que se sobre escribe, el nombre de la función da una visión de la lista a la que hace referencia.

Nombre: Domicilio	
Tipo de clase : entidad	
Atributo	**Tipo**
idDomicilio	int
consultorio	String
areaSalud	String
fechaNaciemiento	String
direccionDomicilio	String

fechaAlta	String
fechaBaja	String
Para cada responsabilidad:	
Nombre:	getDireccionDomicilio():String
Descripción:	Devuelve el valor del atributo direccionDomicilio.
Nombre:	setDireccionDomicilio(String param): void
Descripción:	Sobre escribe el valor del atributo direccionDomicilio por el valor de param.
Nombre:	Domicilio()
Descripción:	Es el constructor de la clase.
Nota:	Se implementó un get y un set por cada uno de los atributos de esta clase.

Nombre: AntecedentesPatológicosFamiliares	
Tipo de clase : entidad	
Atributo	**Tipo**
idAtecedentesPatológicosFamiliares	int
parentesco	String
diagnostico	String
Para cada responsabilidad:	
Nombre:	AntecedentesPatológicosFamiliares ()
Descripción:	Es el constructor de la clase.
Nombre:	getParentesco():String
Descripción:	Devuelve el valor del atributo parentesco.
Nombre:	void setParentesco(String param)
Descripción:	Sobre escribe el valor del atributo parentesco por el valor de param.
Nota:	Se implementó un get y un set por cada uno de los atributos de esta clase.

Nombre: AntecedentesPatológicosPersonales	
Tipo de clase : entidad	
Atributo	**Tipo**
idAtecedentesPatológicosPersonales	int
añoDiagnostico	String
diagnostico	String
operacionTipo	String
añoOperacion	String
secuelas	Boolean
Para cada responsabilidad:	
Nombre:	AntecedentesPatológicosPersonales()
Descripción:	Es el constructor de la clase.
Nombre:	getDiagnostico():String
Descripción:	Devuelve el valor del atributo diagnostico.
Nombre:	void setDiagnostico(param: String)
Descripción:	Sobre escribe el valor del atributo diagnostico por el valor de param.
Nota:	Se implementó un get y un set por cada uno de los atributos de esta clase.

Nombre: FactoresRiesgo	
Tipo de clase : entidad	
Atributo	**Tipo**
idFactoresRiesgo	int
fecha	String
masaCorporal	String
tabaquismo	String
tensionArterialMinima	String
tensionArterialMaxima	String
ingestionDeBebidasAlcohol	Boolean
incrementoLipidosSang	Boolean

sedentarismo	Boolean
pruebaCitologica	Boolean
examenMamas	Boolean
examenBucal	Boolean
tactoRectal	Boolean
tactoRectal	Boolean
riesgoPrecondicional	String
controlado	Boolean
metodo	String

Para cada responsabilidad:	
Nombre:	FactoresRiesgo()
Descripción:	Es el constructor de la clase.
Nombre:	getTensionArterialMinima():String
Descripción:	Devuelve el valor del atributo tensionArterialMinima.
Nombre:	void getTensionArterialMinima(param:String): void
Descripción:	Sobre escribe el valor del atributo tensionArterialMinima por el valor de param.
Nota:	Se implementó un get y un set por cada uno de los atributos de esta clase.

Nombre: MenorDeUnAño	
Tipo de clase : entidad	
Atributo	**Tipo**
idMenorDeUnAño	int
talla	Float
C_C	Float
C_T	Float
V_N	Float
peso	Float
fecha	String
Para cada responsabilidad:	

Nombre:	MenorDeUnAño()
Descripción:	Es el constructor de la clase.
Nombre:	getTalla():Float
Descripción:	Devuelve el valor del atributo talla.
Nombre:	void setTalla(Float param)
Descripción:	Sobre escribe el valor del atributo talla por el valor de param.
Nota:	Se implementó un get y un set por cada uno de los atributos de esta clase.

Nombre: EsquemaVacunación	
Tipo de clase : entidad	
Atributo	**Tipo**
idEsquemaVacunación	int
fecha	String
Para cada responsabilidad:	
Nombre:	EsquemaVacunación()
Descripción:	Es el constructor de la clase.
Nombre:	getFecha():String
Descripción:	Devuelve el valor del atributo fecha.
Nombre:	void setFecha(String param)
Descripción:	Sobre escribe el valor del atributo talla por el valor de param.
Nombre:	getEdad(fechaNacimiento:String):String
Descripción:	Esta función devuelve la edad que tenia el paciente cuando se puso la vacuna, se calcula con la fecha en que se la puso y la fachaNacimiento.
Nota:	Se implementó un get y un set por cada uno de los atributos de esta clase.

Nombre: HistoriaClinicaPersonalDAO	
Tipo de clase : entidad	
Atributo	**Tipo**
driver	int
url	String
password	String
userName	String
cantidadFilasResultado	int
Para cada responsabilidad:	
Nombre:	pacienteTieneHistoriaClínicaPAciante(paciente:PacienteUCI): Boolean
Descripción:	Es función busca en la base de datos la historia clínica asociada a este paciente, en caso de que tenga devuelve true en caso contrario devuelve falso.
Nombre:	mostrarHistoriaClínicaPersonal(paciente:Paciente):Historia ClínicaPersonal
Descripción:	Esta función devuelve la historia clínica personal asociada al paciente dado, llamando a las funciones mostrar de cada una de las listas que componen a esta.
Nombre:	mostrarHistoriaClínicaPersonal(historiaClinicaPersonal: HistoriaClínicaPersonal):Boolean
Descripción:	Esta función devuelve el valor de si se pudo modificar o no, llamando para la modificación a cada una de las funciones insertarOModificar de cada una de las listas que la componen, menos la listaConsultas. Estos datos de modifican en la base de datos.
Nombre:	insertarHistoriaClínicaPersonal(historiaClinicaPersonal: HistoriaClínicaPersonal): HistoriaClínicaPersonal
Descripción:	Esta función devuelve la misma historia clínica personal pero con el valor del atributo idHistoriaClínicaPersonal, llamando para la inserción a cada una de las funciones insertarOModificar de cada una de las listas que la

	componen, menos listaConsultas. Estos datos se insertan en la base de datos.
Nombre:	mostrarDomicilios(idHistoriaClínicaPersonal:String):Vector
Descripción:	Esta función busca todos los domicilios asociados a esta historia clínica personal en la base de datos, devolviendo una lista de estos datos. En caso que no halla devuelve la lista vacía.
Nombre:	mostrarAntecedentesPatológicosFamiliares(idHistoriaClínicaPersonal:String):Vector
Descripción:	Esta función busca todos los antecedentes patológicos familiares asociados a esta historia clínica personal en la base de datos, devolviendo una lista con estos datos. En caso que no halla devuelve la lista vacía.
Nombre:	mostrarAntecedentesPatológicosPersonales(idHistoriaClínicaPersonal:String):Vector
Descripción:	Esta función busca todos los antecedentes patológicos personales asociados a esta historia clínica personal en la base de datos, devolviendo una lista con estos datos. En caso que no halla devuelve la lista vacía.
Nombre:	mostrarFactoresRiesgo(idHistoriaClínicaPersonal:String):Vector
Descripción:	Esta función busca todos los factores de riego para mayores de quince años asociados a esta historia clínica personal en la base de datos, devolviendo una lista con estos datos. En caso que no halla devuelve la lista vacía.
Nombre:	mostrarEsquemaVacunación(idHistoriaClínicaPersonal:String):Vector
Descripción:	Esta función busca todas las vacunas asociados a esta historia clínica personal en la base de datos, devolviendo una lista con estos datos. En caso que no halla devuelve la lista vacía.
Nombre:	mostrarMenorDeUnAño(idHistoriaClínicaPersonal:String):Vector

Descripción:	Esta función busca todos los datos de menor de un año asociados a esta historia clínica personal en la base de datos, devolviendo una lista con estos datos. En caso que no halla devuelve la lista vacía.
Nombre:	mostrarConsultas(idHistoriaClínicaPersonal:String):Vector
Descripción:	Esta función busca todas las consultas asociados a esta historia clínica personal en la base de datos, devolviendo una lista con estos datos. En caso que no halla devuelve la lista vacía. Para cada consulta se busca la lista de tratamientos que esta tiene.
Nombre:	mostrarTratamientos(idHistoriaClínicaPersonal:String):Vect or
Descripción:	Esta función busca todos los tratamientos asociados a una consulta de una historia clínica personal en la base de datos, devolviendo una lista con estos datos. En caso que no halla devuelve la lista vacía. Para cada consulta se busca la lista de tratamientos que esta tiene.
Nombre:	insertarOModificarDomicilios(listaDomicilios:Vector):Vector
Descripción:	Esta función recorre toda la lista para ver cual tiene que insertar y cual modificar, en caso de las inserciones le pone el id con que se insertó al elemento. Devuelve la misma lista modificada.
Nombre:	insertarOModificarAntesedentesPatológicosPersonales(lista AntecedentesPatológicosPersonales:Vector):Vector
Descripción:	Esta función recorre toda la lista para ver cual tiene que insertar y cual modificar, en caso de las inserciones le pone el id con que se insertó al elemento. Devuelve la misma lista modificada.
Nombre:	insertarOModificarAntesedentesPatológicosFamiliares(lista AntecedentesPatológicosFamiliares:Vector):Vector
Descripción:	Esta función recorre toda la lista para ver cual tiene que insertar y cual modificar, en caso de las inserciones le pone el id con que se insertó al elemento. Devuelve la misma

	lista modificada.
Nombre:	insertarOModificarFactoresRiesgo(listaFactoresRiesgo:Vector):Vector
Descripción:	Esta función recorre toda la lista para ver cual tiene que insertar y cual modificar, en caso de las inserciones le pone el id con que se insertó al elemento. Devuelve la misma lista modificada.
Nombre:	insertarOModificarMenorDeUnAño(listaMenorDeUnAño:Vector):Vector
Descripción:	Esta función recorre toda la lista para ver cual tiene que insertar y cual modificar, en caso de las inserciones le pone el id con que se insertó al elemento. Devuelve la misma lista modificada.
Nombre:	insertarOModificarEsquemaVacunación(listaEsquemaVacunación:Vector):Vector
Descripción:	Esta función recorre toda la lista para ver cual tiene que insertar y cual modificar, en caso de las inserciones le pone el id con que se insertó al elemento. Devuelve la misma lista modificada.
Nota:	Todas esta funciones son de tipo static, al igual que los atributos

Anexo 23. Clase de Interfaz. (MLC).

Nombre: PaginaAdicionarECDisponible	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para adicionar un tipo de examen.

Nombre: PaginaSeleccionarExamen	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de selección de examen.

Nombre: PaginaModificarECDisponible	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de Modificar un examen disponible.

Nombre: PaginaSeleccionarReporte	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de selección del reporte que se desea.

Nombre: PaginaMostrarReporte	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de mostrar reporte.

Nombre: PaginaMostrarResultado	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de mostrar resultados.

Nombre: PaginaIdentificarPaciente	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de identificación del paciente.

Nombre: PaginaRegistrarMuestra	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de registrar la muestra.

Nombre: PaginaBuscarExamen	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de buscar examen.

Nombre: PaginaRegistrarResultado	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página de registrar el resultado.

Anexo 24. Clase de Interfaz. (MHCP).

Nombre: Mostrar Historia Clínica Personal	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para mostrar los datos de la historia clínica personal.

Nombre: Modificar Historia Clínica Personal	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para modificar los datos de la historia clínica personal.

Nombre: Crear Historia Clínica Personal	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para introducir los datos de la historia clínica personal.

Nombre: Modificar Domicilio	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para modificar los datos de un domicilio

Nombre: Crear Domicilio	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para introducir los datos de un domicilio

Nombre: Modificar Antecedentes Patológicos Personales	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para modificar los datos de un antecedente patológico personal.

Nombre: Crear Antecedentes Patológicos Personales	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para introducir los datos de un antecedente patológico personal.

Nombre: Modificar Antecedentes Patológicos Familiares	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para modificar los datos de un antecedente patológico familiares.

Nombre: Crear Antecedentes Patológicos Familiares	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para insertar los datos de un antecedente patológico familiares.

Nombre: Modificar Factores de Riesgo	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para modificar los datos de un factor de riesgo

Nombre: Crear Factores de Riesgo	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para insertar los datos de un factor de riesgo

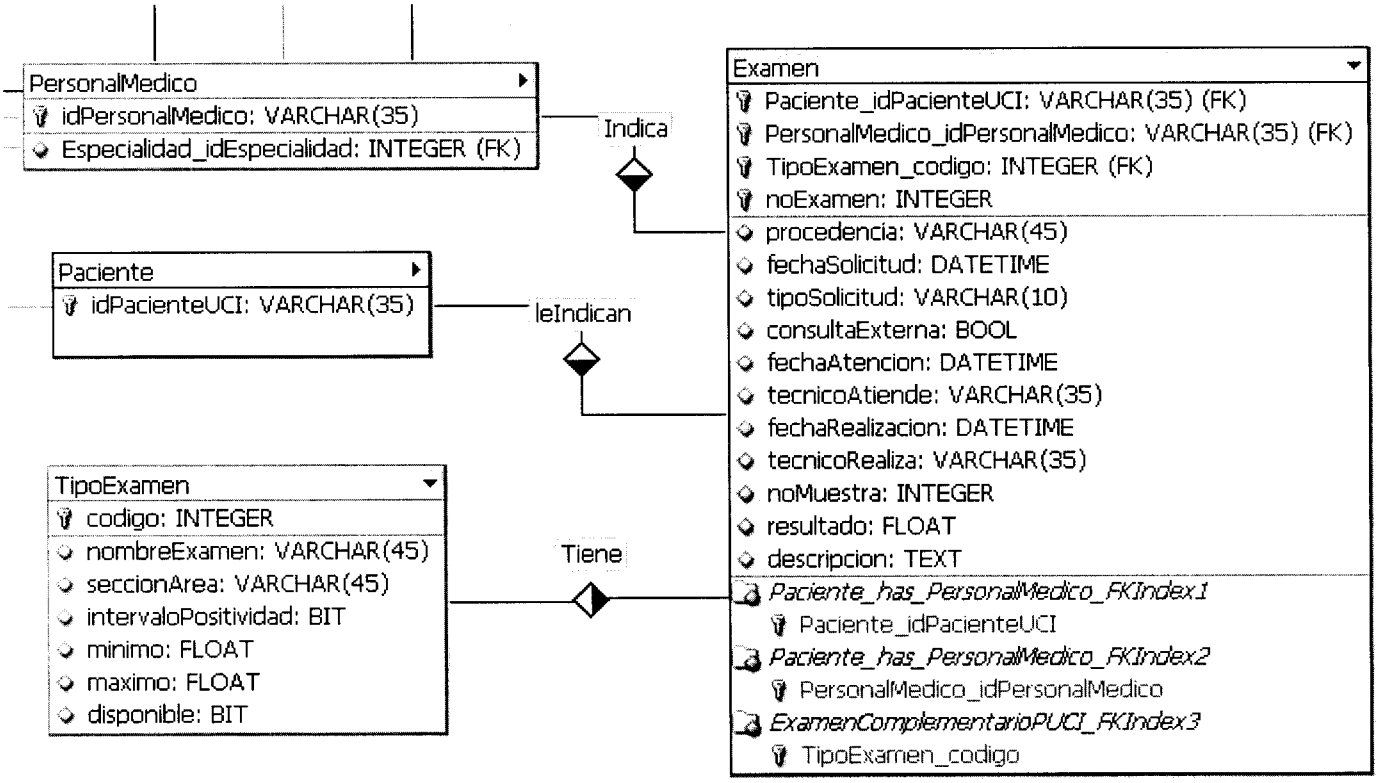
Nombre: Modificar Menor de un Año	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para modificar los datos de un menor de un año.

Nombre: Crear Menor de un Año	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para insertar los datos de un menor de un año

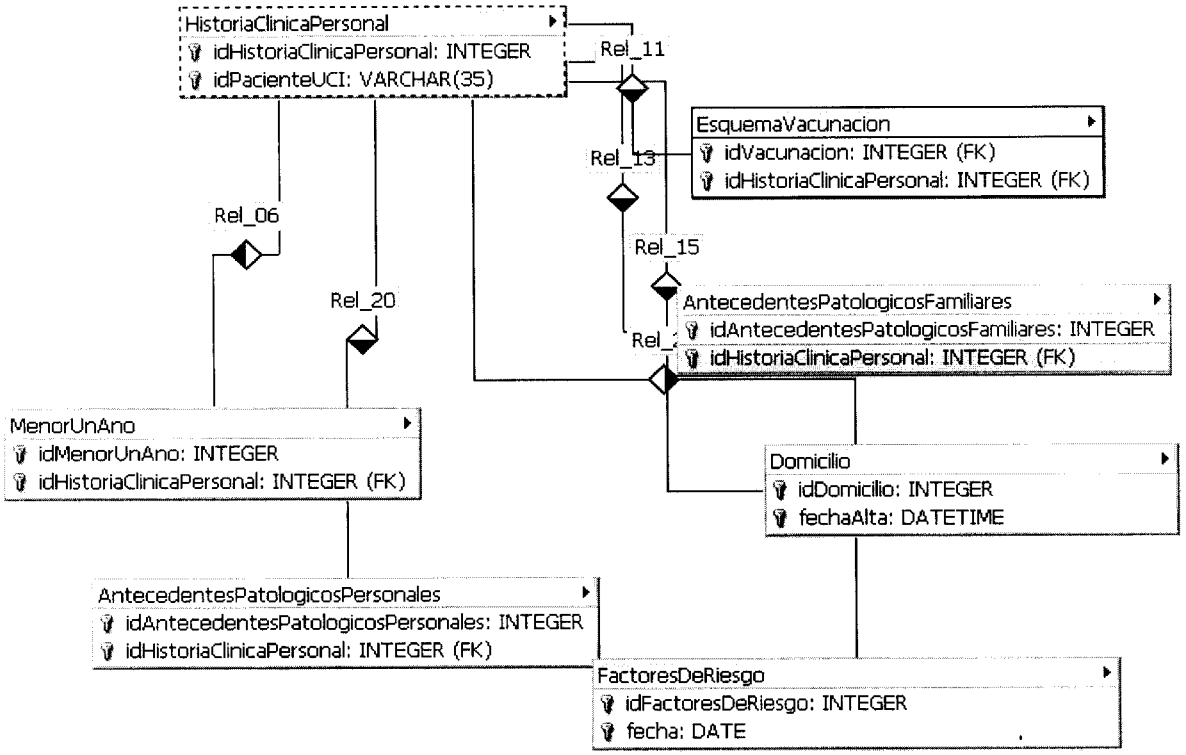
Nombre: Modificar Esquema de Vacunación	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para modificar los datos de una vacuna.

Nombre: Crear Esquema de Vacunación	
Tipo de clase : interfaz	
Atributo	**Tipo**
Para cada responsabilidad:	
Nombre:	CrearPágina()
Descripción:	Crear la página para insertar los datos de las vacuna que se ha puesto el paciente.

Anexo 25. Diagrama Entidad Relación. (MLC).



Anexo 26. Diagrama Entidad Relación. (MHCP).



Anexo 27. Descripción de las tablas. (MLC).

Nombre: Examen		
Descripción: Esta tabla contiene el listado de los exámenes complementarios.		
Atributo	**Tipo**	**Descripción**
idPaciente	VARCHAR(35)	Identificador del paciente que se le realiza el examen.
IdPersonalMédico	VARCHAR(35)	Identificador del médico que indica el examen complementario.
Codigo	INTEGER	Codigo del examen que se realiza.
noExamen	INTEGER	El número del examen realizado al paciente
procedencia	VARCHAR(45)	El nombre de la consulta donde se indica el examen complementario.
tipoSolicitud	VARCHAR(10)	El tipo de solicitud.
consultaExterna	BOOL	Si se indicó el examen desde una consulta externa.
fechaAtencion	DATETIME	La fecha en que se obtuvo la muestra del examen.
tecnicoAtiende	VARCHAR(35)	Identificador del técnico que recoge la muestra.
fechaRealizacion	DATETIME	La fecha en que se registra el resultado del examen.
tecnicoRealiza	VARCHAR(35)	Identificador del técnico que registra el resultado del examen.

noMuestra	INTEGER	El número de la muestra recogida del examen.
Resultado	FLOAT	El resultado obtenido
descripcion	TEXT	Una descripcion del resultado obtenido.

Nombre: TipoExamen		
Descripción: Esta tabla contiene el listado de los tipos de exámenes complementarios que se realizan.		
Atributo	**Tipo**	**Descripción**
Codigo	INTEGER	El código del examen.
nombreExamen	VARCHAR(45)	El nombre del examen.
seccionArea	VARCHAR(45)	La sección ó el area donde se realiza el examen.
Intervalo	BOOL	Si el resultado depende de una escala.
Minimo	FLOAT	El valor mínimo para saber si el resultado es positivo.
Maximo	FLOAT	El valor máximo para saber si el resultado es positivo.
Disponible	BOOL	Si el examen se puede realzar.

Nombre: PersonalMédico		
Descripción: Esta tabla contiene el listado de las personas que laboran en el centro médico.		
Atributo	**Tipo**	**Descripción**

idPersonalMédico	VARCHAR(35)	El identificador de la persona que trabaja en el centro médico.
especialidad	INTEGER	La especialidad de la persona.
Area	VARCHAR(10)	El área donde esta labora.

Nombre: Paciente		
Descripción: Esta tabla se utiliza para simular la relación con la tabla persona UCI.		
Atributo	**Tipo**	**Descripción**
idPaciente	VARCHAR(35)	El identificador del paciente.

Anexo 28. Descripción de las tablas. (MHCP).

Nombre: historiaclinicapersonal		
Descripción: Esta tabla contiene el listado de los datos de la historia clínica personal.		
Atributo	**Tipo**	**Descripción**
grupoSanguineo	Varchar(5)	Es el grupo sanguíneo al que pertenece el paciente.
alergia	Varchar(255)	A lo que le hace alergia el paciente.
grupoDispensalarial	Varchar(255)	Una descripción de cuanto dinero ingresa al hogar del paciente.
factorRH	Varchar(10)	Es el factor RH

Nombre: antecedentespatologicosfamiliares		
Descripción: Esta tabla contiene el listado de los datos de los antecedentes patológicos familiares asociados a una historia clínica personal de un paciente.		
Atributo	**Tipo**	**Descripción**
diagnostico	Varchar(255)	La enfermedad que presenta algún familiar que puede ser posibles padecimientos del paciente.
parentesco	Varchar(20)	A lo que le hace alergia el paciente.

Nombre: antecedentespatologicospersonales		
Descripción: Esta tabla contiene el listado de los datos de los antecedentes patológicos personales asociados a una historia clínica personal de un paciente.		
Atributo	**Tipo**	**Descripción**
diagnostico	Varchar(255)	Diagnostico del paciente.
anoDiagnostico	date	Año en el que se le diagnostico el antecedente patológico personal.
operacionTipo	Varchar(255)	El tipo de operación que se le realizó al paciente
anoOperacion	date	Año de la operación.
secuelas	Tinyint(1)	Si le dejo trastornos patológicos.

Nombre: domicilio		
Descripción: Esta tabla contiene el listado de los datos de los domicilios asociados a una historia clínica personal de un paciente.		
Atributo	**Tipo**	**Descripción**
fechaAlta	date	Fecha en que ingresó al domicilio
fechaBaja	date	Fecha en que se dio baja del domicilio
areaSalud	Varchar(255)	El área de salud que lo atendía
consultorio	Varchar(10)	El consultorio donde se trataba.
direccionDomicilio	Varchar(255)	La dirección de donde vivía.

Nombre: factoresderiesgo		
Descripción: Esta tabla contiene el listado de los datos de los factores de riego para mayores de quince años asociados a una historia clínica personal de un paciente.		
Atributo	**Tipo**	**Descripción**
fecha	date	Fecha en que se le hizo el chequeo.
masaCorporal	date	El índice de masa corporal presentado por el paciente.
tensionArterialMinima	Varchar(4)	El valor máximo de la tensión arterial.
tensionArterialMaxima	Varchar(4)	El valor mínimo de la tensión arterial.
tabaquismo	Tinyint(1)	Si fuma o no.

ingestionBebidasAlcohol	Tinyint(1)	Si ingiere bebidas alcohólicas o no.
incrementoLipidosSang	Tinyint(1)	Si tiene incrementos de los lípidos sanguíneos o no.
sedentarismo	Tinyint(1)	Si es sedentario o no.
pruebapsitologica	Tinyint(1)	Si se le realizó la prueba psicológica o no.
examenMamas	Tinyint(1)	Si se le realizó un examen de mamas o no.
examenBucal	Tinyint(1)	Si se le realizó un examen bucal o no.
tactoRectal	Tinyint(1)	Si se le realizó un tacto rectal o no.
riesgoPrecondicional	Varchar(255)	Una breve descripción del riesgo precondicional que presenta.
controlado	Tinyint(1)	Si fue controlado o no.
metodo	Varchar(255)	El método que se le realizó para controlarlo.

Nombre: menorunano

Descripción: Esta tabla contiene el listado de los datos para menores de un año asociados a una historia clínica personal de un paciente.

Atributo	**Tipo**	**Descripción**
fecha	date	Fecha del chaqueo.
talla	float	La talla del paciente.
peso	float	El peso del paciente.

c_C	float	Circunferencia Cefálica
c_T	float	Circunferencia Toráxico
v_N	float	Valor Nutritivo

Nombre: esquemavacunacion		
Descripción: Esta tabla contiene el listado de los datos de las vacunas asociadas a una historia clínica personal de un paciente.		
Atributo	**Tipo**	**Descripción**
fecha	date	La fecha en que se puso la vacuna.

Glosario de términos

ASCII
(American Standard Code for Information Interchange)
Código americano estándar para intercambio de información.
Estándar que define cómo representar dígitos, letras, signos y signos de puntuación en computadoras (por ejemplo, la A mayúscula corresponde al código número 65). Aunque existen otros estándares, el ASCII es el más popular.

Base de Datos (BD)

Conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerado una colección de datos variables en el tiempo.

BROWSER
(Navegador) Programa usado para accesar diferentes tipos de recursos en Internet. Los más famosos hoy en día son los browser de WWW (Netscape; Internet Explorer, Mosaic, Opera, etc.) y suelen trabajar con una arquitectura cliente/servidor.

CGI (Common Gateway Interface)
CGI, es un interface para que programas externos (pasarelas) puedan rodar bajo un servidor de información. Actualmente, los servidores de información soportados son servidores HTTP (hypertext Transfer Protocol). Las pasarelas pueden usarse para muchos propósitos, algunos de ellos:
Manejo de formas y cuestionarios Conversión de las man pages del sistema a páginas html y presentación del resultado por parte del cliente WWW Interface con bases de datos WAIS y Archie, y presentación de los resultados en formato html por parte de clientes WWW Mensajería electrónica (comunicación con los administradores WWW)

GNU
La Fundación para el Software Libre (FSF - Free Software Foundation) está

dedicada a eliminar las restricciones de uso, copia, modificación y distribución del software. Promueve el desarrollo y uso del software libre en todas las áreas de la computación. Específicamente, la Fundación pone a disposición de todo el mundo un completo e integrado sistema de software llamado GNU. La mayor parte de este sistema está ya siendo utilizado y distribuido.

HIPERTEXTO

(HTLM) Texto que incluye links (enlaces) a otros documentos. Se lo puede definir como un sistema que vincula mediante enlaces activables elementos de información. En el concepto de WWW, el término hipertexto se funde con el de hipermedia, ya que no sólo se puede acceder a texto, sino también a imágenes, video y sonido.

HTTP (Hypertext Transfer Protocol)

HTTP es un protocolo con la ligereza y velocidad necesaria para distribuir y manejar sistemas de información hipermedia. Es un protocolo genérico orientado al objeto, que puede ser usado para muchas tareas como servidor de nombres y sistemas distribuidos orientados al objeto, por extensión de los comandos, o métodos usados. Una característica de HTTP es la independencia en la visualización y representación de los datos, permitiendo a los sistemas ser construidos independientemente del desarrollo de nuevos avances en la representación de los datos HTTP ha sido usado por los servidores World Wide Web desde su inicio en 1.990.

INTERNET:

Es la red de redes. Nacida como experimento del ministerio de defensa americano, conoce su difusión más amplia en el ámbito científico-universitario. Embrión de las 'superautopistas de la información'. Para convertirse en ellas faltan mayores infraestructuras y anchos de banda. Desde el punto de vista técnico, Internet es un gran conjunto de redes de ordenadores interconectadas (la mayor red mundial).

Intranet

Es una adaptación de las mismas tecnologías que existen en Internet, para que sean utilizadas dentro de la red internan de una empresa u organización de forma tal que sus miembros puedan intercambiar información de todo tipo, utilizando el Web como interfaz común.

JAVA

Java es un lenguaje orientado a objetos y desarrollado por Sun Microsystems. Comparte similitudes con C, C++ y Objetive C. Basándose en otros lenguajes orientados al objeto, Java recoge lo mejor de todos ellos y elimina sus puntos más conflictivos. El principal objetivo de JAVA fue hacer un lenguaje que fuera capaz de ser ejecutado de una forma segura a través de Internet (aunque el código fuera escrito de forma maliciosa). Java no es un lenguaje para ser usado solo en el WWW, pero su despegue y utilización se debe al World Wide Web. Hoy día casi todos los browser interpretan código Java

LINUX

Linux es una implementación independiente con "espíritu" POSIX (especificación para sistemas operativos). Tiene extensiones System V y BSD, y ha sido escrito completamente a base de aportaciones. Linux no tiene codigo propietario. Linux está distribuido libremente bajo "GNU Public License".

MACINTOSH:

Serie de ordenadores de Apple Computer. Posee un sistema operativo basado en ventanas. El entorno es intuitivo, eliminando el teclado de los comandos del sistema. Prácticamente todo puede hacerse a través de menús desplegables y de ratón. A todos los objetos se le asigna una representación gráfica (iconos).

MICROSOFT WINDOWS

Sistema operativo gráfico de Microsoft basado en ventanas. Es el más popular en entornos PC. Permite el acceso a Internet mediante TCP/IP y Winsockets.

MOSAIC

Cliente WWW desarrollado en NCSA para las siguientes plataformas: Mosaic para X: Usa X11/Motif. Fué el primer cliente para WEB. Soporta http 1.0. Disponible mediante FTP anonymous en ftp.ncsa.uiuc.edu en el directorio Mosaic. Mosaic para MS-Windows: Precisa las librerías de 32 bit (win32). Disponible vía FTP anonymous en ftp.ncsa.uiuc.edu en el directorio PC/Windows/Mosaic.

NETSCAPE

Cliente WWW desarrollado por Netscape Communications Corp. Descarga y visualiza las imágenes en forma incremental, permitiendo, mientras, leer el texto (también descargado de forma incremental). Es probablemente el mejor cliente WWW. Soporta acceso directo a news, sin pasarelas, y muchas de las extensiones de HTML.

SGBD

Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios , desde diferentes puntos de vista y a la vez.

Software

Palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

TCP/IP

"Transmission Control Protocol/Internet Protocol" Es el conjunto de protocolos que definen a Internet. Originalmente diseñado para el sistema operativo UNIX, hoy en día existe software TCP/IP disponible para la mayoría de los sistemas operativos. Para poder utilizar la Internet, su computador debe tener software TCP/IP.

UNIX

Unix es un sistema operativo multiusuario y multitarea. Como características más importantes:
Redireccionamiento de Entradas/Salidas. Sistema jerárquico de ficheros. Estructura de árbol invertido (File System). Interface simple e interactivo con el usuario. Alta portabilidad al estar escrito en C. Es casi independiente del hardware Creación de utilidades fácilmente. Los componentes básicos del Unix.

WWW

(World Wide Web)

World Wide Web

Servidor de información, desarrollado en el CERN (Laboratorio Europeo de Física de Partículas), buscando construir un sistema distribuido hipermedia e hipertexto. También llamado WEB y W3. World Wide Web {Amplia o Extensa Red Mundial}. En Internet, se refiere a la holgada red de distintos tipos de documentos, conectados mediante enlaces de hipertexto, los cuales se encuentran embebidos dentro de los mismos documentos. Para más información ver Internet y ¿Que es Internet?