

Universidad de las Ciencias Informáticas

Facultad 7



**Título: Sistema Automatizado para la Gestión de
Información en Rehabilitación.**

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

Autor: Néstor Llanes Guerra

Tutor: Ing. Yusniel Ávila Malagón

Ciudad de la Habana, junio del 2007

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de esta tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 23 días del mes de junio del año 2007.

Néstor Llanes Guerra

Ing. Yusniel Ávila Malagón

Firma del Autor

Firma del Tutor

*La finalidad de la ciencia no es brindar la sabiduría perpetua,
sino poner límites a la perpetuación del error.*

Bertold Brech

AGRADECIMIENTOS

A la Revolución Cubana por darme la posibilidad de formarme como profesional de la informática.

A todos los que me han acompañado en este largo camino de la vida y la educación, dándome fuerzas para salir adelante y realizar mis sueños.

Agradecimientos especiales a los compañeros de proyecto y en especial a Daniel Miranda y Alejandro Martínez.

A mi tutor, por su apoyo.

A mis compañeros de estos 5 largos años de estudio y sacrificio, por brindarme su amistad desinteresada en cada momento y su apoyo, en especial a Karen, fue genial haber compartido estos años con ustedes.

A toda mi familia, por su apoyo en todo momento, especialmente a mi hermano Leandro, mi hermana Elizabeth, mi madre Esther y mi padre Néstor.

Un agradecimiento especial a Eliurkis Díaz por su gran ayuda.

Y un agradecimiento especial al colectivo de profesores de mi facultad por hacer posible mi formación como profesional.

A todos, gracias.

DEDICATORIA

A quienes me forjaron, educaron y siempre estarán guiándome donde quiera que estén, mi madre y mi padre.

A mis hermanos, que siempre han estado dándome su apoyo incondicional.

A mis amigos que sin ellos todo este sueño no se hubiera hecho realidad.

A toda mi familia en general, gracias.

Néstor Llanes Guerra

RESUMEN

El objetivo propuesto en el trabajo, es implementar la aplicación Web “Sistema Automatizado para la Gestión de Información en Rehabilitación” mediante el uso de herramientas de software libre, para emplearla en la gestión de la información asociada a las tareas del flujo médico en las Salas de rehabilitación del sector de la salud. Actualmente, en su mayoría, se hace de forma manual.

Para darle solución a estos problemas se decidió desarrollar una aplicación Web, basada en tecnología PHP5 y con gestor de base de datos MySQL. Para el desarrollo del sistema se utilizó la metodología RUP, junto con el Lenguaje Unificado de Modelado (UML). Para documentar el desarrollo del software se decidió utilizar la herramienta Rational Rose 2003, es una herramienta es muy completa y ofrece amplias potencialidades. Como lenguajes de programación se utilizó PHP, y una arquitectura provechosa (MVC).

El sistema propuesto contribuirá a que toda la información sea accesible, obteniendo además, una reducción del tiempo en las búsquedas de información. Por otra parte, permitirá mejorar las condiciones de trabajo del personal, evitándoles el agotamiento y demora que produce el procesamiento manual de la información al contribuir positivamente en el almacenamiento y control de ésta.

TABLA DE CONTENIDOS

AGRADECIMIENTOS	I
DEDICATORIA.....	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Medicina Física y Rehabilitación	4
1.2 Servicio de Rehabilitación Integral	4
1.3 Descripción actual de como se lleva a cabo el negocio.....	5
1.4 Análisis crítico de la ejecución de los procesos	5
1.5 Sistemas automatizados existentes	7
1.6 Propuesta de solución.....	8
CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.....	10
2.1 Internet.....	10
2.2 Aplicaciones Web.....	10
2.3 Modelo Cliente Servidor.....	12
2.4 Servidor Web Apache.....	12
2.5 Lenguajes de programación Web.....	14
2.5.1 PHP (Hypertext Preprocessor).....	14
2.5.2 Java Script.....	17
2.6 Framework	18
2.7 Patrones de diseño.....	18
2.7.1 Modelo Vista controlador.....	19
2.7.2 Active Record.....	22
2.8 Sistemas Gestores de Bases de Datos (SGBD).....	22
2.8.1 MySQL.....	23
2.9 Proceso de desarrollo.....	24
2.9.1 Lenguaje Unificado de Modelado (UML).....	25
2.10 Herramientas utilizadas.....	26
2.10.1 Diseño Gráfico.....	26
2.10.2 Diseño de la interfaz.....	26
2.10.3 Zend Studio.....	26
2.10.4 EMS SQL Manager for MySql.....	27
2.10.5 Case Studio.....	27
2.10.6 Rational Rose.....	27
CAPÍTULO 3: CARACTERÍSTICAS DEL SISTEMA.....	29
3.1 Especificación de Requisitos.....	29
3.2 Modelo de negocio propuesto	29
3.2.1 Actores del negocio.....	29
3.2.2 Trabajadores del negocio.....	30
3.2.3 Diagrama de casos de uso del negocio.....	31
3.2.4 Diagramas de actividades	35

3.2.5 Diagrama de clases del modelo de objetos.....	37
3.3 Modelo del sistema	39
3.3.1 Definición de los requisitos funcionales.....	39
3.3.2 Definición de los requisitos no funcionales.....	40
3.3.3 Actores del Sistema.	42
3.3.4 Diagrama de Casos de Uso del Sistema a automatizar.....	44
3.3.5 Descripción del Caso de Uso del Sistema.	45
CAPÍTULO 4: ANALISIS Y DISEÑO DEL SISTEMA.....	50
4.1 Análisis.....	50
4.2 Diagrama de Clases de Análisis.	50
4.3 Diseño.	58
4.3.1 Diagramas de Interacción.	58
4.3.2 Diagrama de clases del diseño.	59
4.3.4 Descripción de las Clases.....	71
4.4 Diseño de la base de datos.....	82
4.4.1 Modelo de datos.....	82
4.4.2 Descripción de las Tablas.	84
4.5 Principios de diseño.	90
4.6 Tratamiento de errores.....	90
4.7 Seguridad.....	90
4.8 Interfaz de usuario.....	90
4.9 Formato de salida de los reportes.....	91
4.10 Ayuda.	92
CAPÍTULO 5: IMPLEMENTACIÓN.....	93
5.1 Implementación.....	93
5.2 Diagramas de Implementación.....	93
5.3 Diagrama de despliegue.	94
5.4 Diagrama de componentes.	95
CONCLUSIONES.....	99
RECOMENDACIONES	100
REFERENCIAS BIBLIOGRÁFICAS.....	101
BIBLIOGRAFÍA	102
ANEXO 1 MODELO DEL SISTEMA	103
ANEXO 2 MODELO DE DISEÑO	116
GLOSARIO.....	131

INTRODUCCIÓN

El desarrollo de la informática, ha contribuido al mejoramiento del manejo de la información. Existen hoy posibilidades insospechadas en este ámbito. Cuba no ha quedado al margen de este desarrollo, por lo que el Ministerio de Salud Pública se propuso como principal tarea, informatizar los servicios que brindan los centros hospitalarios del país. Aprovechando sus ventajas, automatizar los procesos de gestión y ofrecer una atención cualitativamente superior a la población cubana.

La necesidad de los servicios de fisioterapia ha conllevado a la apertura de clínicas a lo largo de toda la geografía nacional. Estas, tienen la necesidad y el deber de atender óptimamente al paciente, lo que depende en gran medida de la buena gestión de la información en las mismas. Para realizarla es necesario contar con herramientas que nos faciliten el trabajo, la dirección, el control de los datos y la supervisión de las actividades de los empleados y pacientes. Para lo cual, las Nuevas Tecnologías nos aportan un alto grado de apoyo y facilitan, la implantación de Sistemas de Información que permiten el control de las clínicas de manera global e individualizada.

La gestión de la información en las áreas de fisioterapia y rehabilitación ha sido de trascendental importancia a lo largo de la historia, ya que de esta manera otros pueden conocer el funcionamiento y eficacia de estas especialidades. La necesidad de un sistema capaz de gestionar esta información automatizadamente, ha sido la motivación principal de este trabajo.

En su mayoría, las salas destinadas a la rehabilitación en Cuba, manejan la información de forma manual, lo que trae consigo atrasos a la hora de actualizar la misma, pérdidas, demoras en la prestación de los servicios y acumulación de grandes volúmenes de información, todo esto es parte de la **situación problemática**.

En el centro de rehabilitación del CIMEQ existe un sistema que gestiona esta información, pero está hecho con herramientas obsoletas. Es manejado por una sola persona y no tienen acceso a la información, las personas que la necesitan en un momento determinado.

De lo antes planteado podemos definir el siguiente **problema**: ¿Cómo gestionar el manejo y procesamiento del flujo de la información en las salas de rehabilitación del sector de la salud?

Por lo que el **objeto de estudio** son los procesos que abarcan el control y manejo del flujo de la información en las áreas de rehabilitación.

Donde el **campo de acción** comprende los procesos automatizados para el manejo y procesamiento de la información en las áreas de rehabilitación.

Se ha propuesto como **objetivo general** implementar una aplicación Web “Sistema Automatizado para la Gestión de Información en Rehabilitación” mediante el uso de herramientas de software libre, para emplearla en la gestión de la información asociada a las tareas del flujo médico en las Salas de rehabilitación del sector de la salud.

Como **idea a defender** una aplicación Web, basada en un sistema gestor de bases de datos potente y eficaz, y un intérprete rápido eficaz como PHP5; la implementación y puesta en marcha del sistema, mejorará el control y manejo del flujo de la información en las salas de rehabilitación.

Tareas de la Investigación:

- ✓ Analizar como se maneja la información en los centros de rehabilitación de todo el país.
- ✓ Evaluar con otros sistemas usados en estas áreas.
- ✓ Desarrollar un estudio detallado de las arquitecturas, seleccionando la idónea para nuestro sistema.
- ✓ Implementar un Sistema de Gestión de la Información para las Áreas de Rehabilitación en el MINSAP.

Para obtener toda la información necesaria que nos permita llegar al resultado esperado utilizaremos los siguientes métodos teóricos y empíricos:

- ✓ La entrevista: para la recopilación de toda la información necesaria para el diseño de la aplicación, dígase flujo actual de los eventos en el proceso de rehabilitación, datos de entrada, datos de salida, características de los procesos, etc.
- ✓ Se usa el método histórico: para saber si ya existen otros sistemas informáticos de este tipo implementados y en uso, en el mundo, y de existir, conocer sobre su comportamiento y como han interactuado con estos los fisiatras.
- ✓ El método de análisis: para poder entender y modelar el proceso de rehabilitación para estudiar las relaciones y cualidades del objeto de estudio.

Con el funcionamiento de la aplicación se espera que se logre un incremento significativo de la capacidad organizativa de los centros de rehabilitación del país, el aumento de la calidad en la atención médica a los pacientes.

El presente trabajo, estructurado en 5 capítulos, distribuidos de la siguiente forma:

Capítulo 1. Fundamentación Teórica: descripción de los procesos en los centros de Rehabilitación, problemas que motivan la investigación, sistemas existentes vinculados al campo de acción, se plantea el problema y se propone la solución.

Capítulo 2. Tendencias y Tecnologías Actuales a Considerar: descripción de tendencias y tecnologías seleccionadas, para el desarrollo de la propuesta de solución.

Capítulo 3. Características del Sistema: definición de los procesos, actores, trabajadores, casos de uso del negocio, diagramas de clases del modelo de objetos del negocio; requisitos funcionales y no funcionales; actores y casos de uso del sistema.

Capítulo 4. Análisis y Diseño del Sistema: descripción del diseño a través de diagramas de clases análisis y diseño de la aplicación. A través de estereotipos Web, que describen la relación entre las páginas. Se obtiene el diagrama de la base de datos y se definen, además, los principios de diseño.

Capítulo 5. Implementación: descripción de los modelos de implementación, los diagramas de despliegue y de componentes. Se describe como se implementan en términos de componentes y su organización y dependencia entre nodos físicos, en los que funcionará la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo, se describe como se llevan a cabo los principales procesos en los centros de rehabilitación integral y los principales problemas que motivaron esta investigación.

Se explica la forma en que se manipula la información hoy en día, en los centro de rehabilitación. Se identifican y analizan las necesidades de los fisiatras, describiéndose los aspectos que serán objeto de automatización, se analizan varios sistemas que pudieran constituir una variante de solución, justificándose su no conveniencia. Se plantea la propuesta de solución y se fundamentan los objetivos propuestos.

1.1 Medicina Física y Rehabilitación

La Medicina Física y Rehabilitación es la especialidad médica que desarrolla un conjunto de medidas de promoción de salud, prevención, diagnóstico, investigación y tratamiento, para obtener la máxima reincorporación posible a la comunidad de los pacientes con enfermedades o secuelas invalidantes, congénitas o adquiridas.

La especialidad utiliza esencialmente como terapéutica los agentes físicos, naturales y artificiales, métodos de reeducación funcional, métodos de terapia ocupacional, los métodos de la ortopedia técnica y las ayudas ortésicas. Se encarga de implementar, dirigir y controlar el desarrollo del Proceso de Rehabilitación.[1]

1.2 Servicio de Rehabilitación Integral

El Servicio de Rehabilitación Integral es una estrategia de nuestro Sistema de Salud para lograr la atención adecuada y la reincorporación social del discapacitado, así como garantizar una mejor calidad de vida para nuestra población.

Su objetivo primordial es disminuir el impacto de la discapacidad por medio de la ampliación de coberturas y la integración de estas personas. Este enfoque abarca la disponibilidad de los recursos de

rehabilitación, desde la comunidad, para aprovechar y potenciar los recursos humanos y materiales de esta.[2]

1.3 Descripción actual de como se lleva a cabo el negocio.

El análisis del flujo de procesos permite reconocer como funciona realmente el negocio para producir uno o varios resultados. El resultado puede ser un producto, un servicio, una información o combinación de ellos. Analizar el flujo de los procesos permite revelar problemas potenciales tales como: la circulación doble de la información, los pasos innecesarios, la duplicación del trabajo entre otros.

En las salas de rehabilitación, donde se garantizan los servicios y recursos necesarios para atender a los pacientes que reciben asistencia médica.

El paciente llega al servicio de Rehabilitación donde es atendido por la recepcionista que es la encargada de identificar si el paciente viene por tratamiento o consulta, si es por tratamiento el paciente es remitido hacia el área donde se aplica dicho tratamiento, si es por consulta ella se encarga de planificarle una consulta con el médico.

Llegado el día de la consulta este le llena la hoja de cargo del mismo, donde recoge una serie de datos entres los que se encuentran: impresión diagnóstica, problemas encontrados y le indica cuales son los tratamientos que se debe realizar. Luego es remitido con la secretaria que será la que llevará a cabo el proceso de planificación del horario del tratamiento al paciente, de acuerdo a la disponibilidad que tengan los técnicos en su horario. Para darle seguimiento al tratamiento, es tomada la asistencia junto a una serie de datos así como si hay mejoría.

1.4 Análisis crítico de la ejecución de los procesos

Aunque el negocio actual de cómo se llevan a cabo los procesos ha permitido satisfacer las necesidades básicas, cabe destacar aspectos deficientes que se han detectado y que han generado la situación problemática que se trata de resolver con el presente trabajo.

Una vez que los pacientes han pasado por el proceso de planificación de consulta, se les envía a los médicos un reporte de los pacientes que van a tener consulta ese día.

El día de esta, los pacientes son vistos por un médico que es el encargado de evaluar la situación del paciente, de acuerdo a los resultados le manda una serie de tratamientos que serán los que el paciente deberá aplicarse. La situación del los pacientes es determina por las informaciones siguientes:

- ✓ **Diagnóstico** (Artropatía, traumatismo, amputados, parálisis, polineuropatías, I.M.A, afecciones respiratorias, esclerosis múltiple, lesiones inflamatorias, etc.)
- ✓ **Tipo de diagnóstico** (presuntivo, informado.)
- ✓ **Tipo de discapacidad**, de acuerdo a la naturaleza del cambio en la estructura corporal (estructura del sistema nervioso, estructura del ojo, el oído y estructuras relacionadas, estructuras involucradas en la voz y el habla, estructuras de los sistemas cardiovascular, inmunológico y respiratorio, estructuras relacionadas con los sistemas digestivos, metabólico y endocrino, estructuras relacionadas con el sistema genitourinario y el sistema reproductor, estructuras relacionadas con el movimiento, piel y estructuras relacionadas)
- ✓ **Lugar de tratamiento**(Hospital, domicilio, ambulatorio)
- ✓ **Referencia**(Médico de familia, Hospital, Especialidades, otras)
- ✓ **Tipo de deficiencia**, de acuerdo a las funciones corporales, extensión o magnitud (funciones mentales, funciones sensoriales y dolor, funciones de la voz y el habla, funciones de los sistemas cardiovascular, hematológico, inmunológico y respiratorio, funciones de los sistemas digestivos, metabólico y endocrino, funciones genitourinarias y reproductoras, funciones neuromusculoesqueléticas y relacionadas con el movimiento, funciones de la piel y estructuras relacionadas)

Los **tratamientos** aplicados a los pacientes son:

- ✓ Infrarrojo, Hivamat, Láserterapia, Fototer, Parafina, Crioterapia, Peloidoterapia, Magnetoterapia, Corriente Exitomotriz, Corriente Analgésica, Iontoforesis, Ultrasonido, Hidroterapia, Alta Frecuencia, Tracción Vertebral, Terapia Combinada, Mecanoterapia, Kinesioterapia, Masaje Occidental, Fonoforesis, Estimulación Temprana, Calor Húmedo, MNT, Actividades de la Vida Diaria (AVD), Coordinación, Relajación, Funcionales, Logopedia y Foniatría, Podología, Consejería, Nutricional, Psicología.

Una vez terminado el proceso de consulta la recepcionista pasa a planificar los tratamientos que se les indicaron a los pacientes, de allí se les envía a los técnicos los tratamientos que deberá aplicar.

Los técnicos se encargan de aplicarles los tratamientos a los pacientes, los pacientes solo serán dados de baja del tratamiento si se ausentan más de tres días o si termina su periodo de tratamientos satisfactoriamente, es decir si con los tratamientos que se le aplicaron se pudo mejorar.

Reglas del negocio

Se identificaron las siguientes reglas que debe seguir la aplicación que se desarrolle, a fin de de respetar y garantizar las restricciones que existen en el negocio:

- ✓ La secretaria es la encargada de manipular toda la información de los pacientes, desde que se registran hasta que se les termina el tratamiento.
- ✓ El paciente es consultado por un fisiatra del departamento, donde le prescribe los tratamientos que deberán llevarse a cabo.
- ✓ Toda la información que se gestiona, es de forma manual.
- ✓ No existe un mecanismo para centralizar la información.

1.5 Sistemas automatizados existentes

Cada sistema de rehabilitación que se ha creado tienen una tendencia dentro de cada centro donde se aplique, es decir, estos sistemas deben cumplir objetivos generales las cuales son:

- ✓ Proporcionar a las clínicas un Sistema de Información homogéneo, abierto y tecnológicamente avanzado.
- ✓ Proporcionar una mayor calidad asistencial a los pacientes de las clínicas.
- ✓ Facilitar el acceso a la información por parte de todos los estamentos implicados en la gestión de las clínicas: personal, asesorías y pacientes.
- ✓ Facilitar la adecuación de las clínicas a diversos procedimientos, fruto de la adecuación de las mismas a sistemas de calidad.

Dentro de estos objetivos generales se encontrarían los específicos que estos serían atendiendo a cada una de las necesidades o requisitos que deba cumplir dicha aplicación dentro de la clínica.

Ej. Lograr la autenticación de usuarios, es decir, permisos necesarios para acceder a la aplicación y realizar los cambios que se necesiten hacer.

Existen varios softwares que han sido diseñados con la finalidad de gestionar todo lo relacionado con clínicas y salas de rehabilitación. ALCFISO es uno de ellos, este sistema que tiene potentes características como son, gestionar de forma cómoda y eficaz sus pacientes e historiales clínicas, administra sencillamente las citas diarias de una forma totalmente visual, la facturación de las sesiones de los pacientes tanto si vienen de forma individual como a través de mutua, crea folletos con los ejercicios que el fisioterapeuta prescriba al paciente, incorpora módulo de copias de seguridad y soporte vía Web.[3]

La aplicación FisioSalus esta compuesto por tres módulos fundamentales los cuales gestionan toda la información de estos centros, agenda para las actividades, historia clínica para almacenar y consultar los datos clínicos y emitir informes, administración, para gestionar la facturación y cobro a pacientes.[4]

En estos momentos existe un sistema para rehabilitación en nuestro país el cual se encarga de procesar toda la información en este centro, este es manipulado por la secretaria

Características:

- ✓ Desarrollado en FoxBase.
- ✓ Interfaz alfanumérica.
- ✓ Almacenamiento en ficheros.

Desventajas:

- ✓ Implica, gasto de tiempo en el procesamiento de la información. Pues la información de todo el centro es procesada por una sola persona, lo que implica un doble trabajo.
- ✓ Escasa documentación.
- ✓ No está implementado para su uso en otras clínicas.

1.6 Propuesta de solución.

Teniendo en cuenta que la información que se genera en estos centros es manipulada de forma manual y no dispone de algún medio o herramienta que le permita realizar estas tareas de formas mas sencilla, se propone la elaboración de un sistema informático que brinde soporte a los procesos de las salas de rehabilitación antes descritos, siendo estos procesos sus principales funcionalidades.

Además de otras funcionalidades que deben de ser incluidos también, como son la administración del sistema, con la cual se debe permitir gestionar toda la información con la que se trabaja (estructura

organizativa, permisos de usuario, roles del sistema, módulos de la aplicación, información sobre los usuarios, etc.), y que brinden la mayor flexibilidad posible, con vistas a su adaptación a cualquier tipo de situación que pueda presentarse. La agenda con la cual se van a planificar los tratamientos y el resto de las actividades del centro.

Toda la información que es consultada en estos centros se encuentra en una base de dato central. Incluir herramientas que permitan navegar por la estructura de las tablas de la aplicación y poder sacar hasta la última gota de información, sin que se tenga que depender de las búsquedas predefinidas. Seguridad total de los datos ante accidentes tales como averías, cortes de luz o desconexiones.

Conclusiones

En este capítulo, se abordan conceptos básicos del proceso de gestión de la información en los centros de rehabilitación y los principales problemas que motivaron la investigación. Se identifican y analizan las necesidades de los fisiatras describiéndose los aspectos que serán objeto de automatización

Se hace un estudio detallado, de otros sistemas que sirven para gestionar la información en estos centros y pudieran constituir una variante de solución, fundamentando su inconveniencia. De esta manera queda demostrada la necesidad de cambio y se plantea una propuesta de solución.

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR.

En el presente capítulo se hace un análisis de las tecnologías y metodologías que pudieran ser útiles en el desarrollo de la propuesta de solución. Teniendo en cuenta las necesidades vistas y las características del entorno donde se aplicará la solución propuesta, se realizó un estudio sobre las posibles herramientas a utilizar.

Se determinan los servidores Web, lenguajes de programación para la Web, patrones de diseño, Sistemas Gestores de Bases de Datos mayormente utilizados a escala internacional, y las distintas metodologías de desarrollo de software. Teniendo en cuenta que deben garantizar el cumplimiento de los intereses de los usuarios.

2.1 Internet.

Se le pudiera decir como todos le conocemos “La Red de Redes” o “La Autopista de la Información”, más bien es el conjunto de redes interconectadas entre si, que permiten la comunicación entre los mas de 30 millones de usuarios que existen en todo el mundo que acceden a la red de redes, que comparten información unas con otras por medio de páginas o sitios. Para que todos estos usuarios puedan coexistir y comunicarse efectivamente entre sí, debe existir un camino físico que una sus computadoras (líneas telefónicas, conmutadas, redes digitales, enlaces satelitales, microondas, fibra óptica, cable coaxial, etc.) [5]

Internet funciona con la estrategia “Cliente/Servidor” lo que significa que en la Red hay ordenadores Servidores que dan una información concreta en el momento que se solicite, y por otro lado están los ordenadores que piden dicha información, los llamados Clientes.

2.2 Aplicaciones Web.

Las aplicaciones Web son una especialización de las aplicaciones cliente-servidor, o sea, su arquitectura general es la de un sistema cliente/servidor, donde tanto el cliente (el navegador) como el servidor (el servidor Web), y el protocolo mediante el que se comunican (el HTTP: HyperText Transfer Protocol) son estándar, y no han de ser creados por el desarrollador.

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR

Las Aplicaciones Web utilizan en la parte del cliente las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. Si no existe lógica de negocios en el servidor, el sistema no puede ser considerado una aplicación Web, en ese caso se considera como un sitio Web. En esencia una aplicación Web usa un sitio Web como entrada a una aplicación típica.

Principales ventajas de las aplicaciones Web.

- ✓ Se puede migrar de sistema operativo o cambiar el Hardware libremente sin afectar el funcionamiento de las aplicaciones de servidor, o sea, son multiplataforma.
- ✓ No se requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones. Solo un computador con un buen navegador Web.
- ✓ Se facilita el trabajo a distancia. Se puede trabajar desde cualquier PC o computador portátil con conexión a Internet.
- ✓ Al funcionar en un navegador, se requiere un conocimiento básico de informática para utilizar una aplicación Web.
- ✓ Las actualizaciones de software son distribuidas automáticamente y de forma transparente al usuario.

El desarrollo de aplicaciones Web está siendo utilizado en muchas organizaciones, ésta situación va ir creciendo indefinidamente. Es por ello que día a día se requieran más programadores capacitados para desarrollos basados en el World Wide Web.

No obstante a la serie de ventajas que presenta tiene además algunas desventajas, las cuales son:

- ✓ La seguridad es mucho más difícil de implementar.
- ✓ Los elementos estándar HTML tienen algunas limitaciones. Dependiendo del navegador que se use, pueden aparecer diferentes para varios usuarios. Esto puede causar alguna confusión si los usuarios se mueven de una estación de trabajo a otra diferente.
- ✓ El desempeño puede ser bajo en un servidor poco potente debido a que se están enviando los datos y el diseño de la pantalla cada vez que se pide un documento HTML.
- ✓ Se necesita de una conexión permanente y rápida.

✓ La interactividad no se produce en tiempo real, en las aplicaciones Web cada acción del usuario conlleva un tiempo de espera algunas veces excesivo hasta que se obtiene la reacción del sistema.[6]

2.3 Modelo Cliente Servidor.

La arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes:

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

Ventajas de la arquitectura cliente-servidor:

- ✓ El servidor no necesita potencia de procesamiento, parte del proceso se reparte con los clientes.
- ✓ Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.[7]

2.4 Servidor Web Apache.

Un servidor de páginas Web es un programa que permite acceder a páginas Web alojadas en un ordenador. Hoy en día Apache es el servidor Web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad, surgió en abril de 1996 y ya en julio del 2002 era utilizado por el 57% de los sitios Web de Internet.

Tiene capacidad para servir páginas tanto de contenido estático, para lo que nos serviría sencillamente un viejo ordenador 486, como de contenido dinámico a través de otras herramientas soportadas que facilitan la actualización de los contenidos mediante bases de datos, ficheros u otras fuentes de información, es muy potente y altamente configurable.

Los servidores Web suministran páginas Web a los navegadores que lo solicitan. En términos más técnicos, los servidores Web soportan el Protocolo de Transferencia de Hipertexto como HTTP (HyperText Transfer Protocol), el estándar de Internet para comunicaciones Web. Usando HTTP, un servidor Web envía páginas Web en HTML y Common Gateway Interface (CGI), así como otros tipos de scripts a los navegadores o browsers cuando éstos los requieren. Cuando un usuario hace clic sobre un enlace a una página Web, se envía una solicitud al servidor Web para localizar los datos nombrados por ese enlace. El servidor Web recibe esta solicitud y suministra los datos que le han sido solicitados o bien devuelve un mensaje de error.

El servidor Apache es un software que esta estructurado en módulos, es decir, está dividido en muchas porciones de código que hacen referencia a diferentes aspectos o funcionalidades del servidor Web. Esta modularidad es intencionada ya que la configuración de cada módulo se hace mediante la configuración de las directivas que están contenidas dentro del módulo. Los módulos del Apache se pueden clasificar en tres categorías:

- 3 Módulos Base: Módulo con las funciones básicas del Apache.
- 4 Módulos Multiproceso: Son los responsables de la unión con los puertos de la máquina, aceptando las peticiones y enviando a los hijos a atender a las peticiones.
- 5 Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor.

Las funcionalidades más elementales se encuentran en el módulo base, siendo necesario un módulo multiproceso para manejar las peticiones. Se han diseñado varios módulos multiprocesos para cada uno de los sistemas operativos sobre los que se ejecuta el Apache, optimizando el rendimiento y rapidez del código. [8]

2.5 Lenguajes de programación Web.

Existen muchos lenguajes para el desarrollo de aplicaciones Web. Estos lenguajes se dividen en lenguajes de lado servidor (son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él) y lenguajes de lado cliente (son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre-tratamiento, como son el HTML, el Java y el JavaScript, los cuales son simplemente incluidos en el código HTML)

Por otra parte, dentro de los lenguajes de lado servidor más utilizados para el desarrollo de aplicaciones Web tenemos: el PHP, el ASP y el PERL; también se encuentra el JSP.

2.5.1 PHP (Hypertext Preprocessor).

El PHP, es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. El PHP originalmente diseñado en Perl, seguidos por la escritura de un grupo de CGI binarios escritos en el lenguaje C por el programador Danés-Canadiense Rasmus Lerdorf en el año 1994 para mostrar su currículum vitae y guardar ciertos datos, como la cantidad de tráfico que su página Web recibía. En los siguientes tres años, se fue convirtiendo en lo que se conoce como PHP/FI 2.0.

Esta forma de programar llegó a muchos usuarios, pero el lenguaje no tomó el peso actual hasta que dos programadores israelíes de Technion, Zeev Suraski y Andi Gutmans reescribieron el analizador gramatical en el año 1997, y crearon la base del PHP 3, cambiando el nombre del lenguaje a la forma actual. Para 1999, Suraski y Gutmans reescribieron el código de PHP, produciendo lo que hoy se conoce como Zend Engine o motor Zend. En mayo del 2000, PHP 4 fue lanzado bajo el poder del motor Zend Engine 1.0. El 13 de julio de 2004, PHP 5 fue lanzado, utilizando el motor Zend Engine II.

En la versión más reciente de PHP, se incluye el novedoso PDO (PHP Data Objects) y mejoras utilizando las ventajas que provee el nuevo Zend Engine 2. Según estudios, más de un millón de servidores tienen esta capacidad implementada y los números continúan creciendo.

Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre las que se pueden mencionar InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP

también ofrece la integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en pdf (Portable Document Format) hasta analizar código XML (eXtensible Markup Language) y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+.

Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas. Es muy rápido y su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado. Es multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server) de forma que el código que se haya creado para una de ellas no tiene porqué modificarse al pasar a la otra.

Su sintaxis está inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja, de modo que si se está familiarizado con esta sintaxis, le resultará muy fácil aprender PHP. Su librería estándar es realmente amplia, lo que permite reducir los llamados "costes ocultos", uno de los principales defectos de ASP (Active Server Pages).

PHP tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos. Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día, a diferencia con el código de otros lenguajes.

Debido a su amplia distribución PHP esta perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

El funcionamiento del PHP se puede describir a través de los pasos siguientes:

- ✓ Escribir en las páginas HTML pero con el código PHP dentro.
- ✓ Guardar la página en el servidor Web.

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR

- ✓ Un navegador solicita una página al servidor.
- ✓ El servidor interpreta el código PHP.
- ✓ El servidor envía el resultado del conjunto de código HTML y el resultado del código PHP que también es HTML.

En ningún caso se envía código PHP al navegador, por lo que todas las operaciones realizadas son transparentes al usuario, el código PHP es ejecutado en el servidor y el resultado es enviado al navegador. El resultado es normalmente una página HTML. Por lo que al usuario le parecerá que está visitando una página HTML que cualquier navegador puede interpretar.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. Además se encuentra libre en el mercado y se puede acceder a él por medio de Internet.

Después de seis años, y después que la comunidad ha revisado el paquete de legados que ha dejado el PHP, se han realizado cambios estructurales en el lenguaje para ofrecer innovación en el nuevo PHP 5 y solucionar muchos de los problemas encontrados en PHP 4.

Afortunadamente, lo nuevo de PHP 5 mejora muchas áreas en el lenguaje y su ejecución, como por ejemplo:

- ✓ Programación orientada a objetos (OOP).
- ✓ MySQL.
- ✓ XML.
- ✓ Integración nativa con el Zend Engine.

Los diseñadores de PHP5 han realizado un cambio radical en el tratamiento de las variables objeto: en PHP5 todas las variables que nombran objetos son en realidad referencias. No hay que usar el operador '&' ni en las asignaciones, ni en el paso de parámetros que son objetos, ahorrándose con ello gran cantidad de potenciales errores.

La principal novedad en las clases de PHP5 es la inclusión de modificadores de control de acceso para implementar la encapsulación --piedra angular en la programación orientada a objetos de la que adolecía PHP4--.

PHP5 introduce tres palabras clave (public, private y protected) que sustituyen a var en la definición de variables miembro --atributos-- de la clase, y que preceden a la definición de funciones miembro --métodos-- .

Otros lenguajes como Perl (Practical Extraction and Report Language), ASP (Active Server Pages) y JSP (Java Server Pages) tienen características similares al PHP aunque poseen rasgos que los marcan y por ello los distingue, entre ellos podemos encontrar:

- ✓ Características multiplataformas: Menos el ASP, que es solamente soportado por la plataforma Windows, los demás lenguajes están soportados en múltiples plataformas.
- ✓ Velocidad de ejecución: la velocidad es mayor en PHP, seguidos por PERL y JSP.
- ✓ Disponibilidad de recursos: actualmente los más utilizados en la Internet son el PHP y el JSP, siendo más utilizado en la publicación de artículos y códigos de ejemplos. PHP tiene una de las comunidades más grandes en Internet, al igual que la de Java.
- ✓ Familiaridad con el lenguaje: En la universidad los lenguajes más utilizados por los programadores es el ASP y el PHP.

De acuerdo a estas comparaciones, el PHP resulta mucho más favorecido, por tanto pensamos que es el adecuado para implementar la propuesta de sistema de este trabajo, particularmente PHP5. [9]

2.5.2 Java Script.

JavaScript es un lenguaje de programación interpretado del lado del cliente porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Está diseñado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador Web. Este se integra directamente en páginas HTML y la ventaja que presenta sobre el HTML es que permite crear páginas Web más dinámicas, lo que las hace más atractivas para el usuario. [10]

2.6 Framework

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

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

2.7 Patrones de diseño.

Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. Los patrones capturan la experiencia existente y probada para promover buenas prácticas.

Los Patrones de Diseño (Design Patterns) son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Estos se dividen en tres grandes categorías:

- ✓ Patrones Creacionales: solucionan problemas de creación de instancias. Nos ayudan a encapsular y abstraer dicha creación.
- ✓ Patrones Estructurales: solucionan problemas de composición (agregación) de clases y objetos.
- ✓ Patrones de Comportamiento: soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan

Un patrón de diseño es:

- ✓ una solución estándar para un problema común de programación
- ✓ una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios

- ✓ un proyecto o estructura de implementación que logra una finalidad determinada
- ✓ un lenguaje de programación de alto nivel
- ✓ una manera más práctica de describir ciertos aspectos de la organización de un programa
- ✓ conexiones entre componentes de programas
- ✓ la forma de un diagrama de objeto o de un modelo de objeto

Los patrones de diseño han contribuido a dar flexibilidad y extensibilidad a nuestros diseños. Pero en adición, han demostrado ser una forma muy útil (exitosa) de reutilizar diseño, ya que ellos no sólo nombran, abstraen e identifican aspectos claves de estructuras comunes de diseño, sino que generalmente son descritos en una forma específica documental, haciendo su comprensión y aplicación fácil para el conjunto de desarrolladores.

Podemos decir que los beneficios que un patrón produce pueden ser medidos en varios sentidos:

- ✓ Contribuyen a reutilizar diseño, identificando aspectos claves de la estructura de un diseño que puede ser aplicado en una gran cantidad de situaciones. La importancia de la reutilización del diseño no es despreciable, ya que ésta nos provee de numerosas ventajas: reduce los esfuerzos de desarrollo y mantenimiento, mejora la seguridad, eficiencia y consistencia de nuestros diseños, y nos proporciona un considerable ahorro en la inversión.
- ✓ Mejoran (aumentan, elevan) la flexibilidad, modularidad y extensibilidad, factores internos e íntimamente relacionados con la calidad percibida por el usuario.

Incrementan nuestro vocabulario de diseño, ayudándonos a diseñar desde un mayor nivel de abstracción.

2.7.1 Modelo Vista controlador.

En ocasiones se lo define más bien como un patrón de diseño o como práctica recurrente.

Un propósito común en numerosos sistemas es el de tomar datos de un almacenamiento y mostrarlos al usuario. Luego que el usuario introduce modificaciones, las mismas se reflejan en el almacenamiento. Dado que el flujo de información ocurre entre el almacenamiento y la interfaz, una tentación común, un impulso espontáneo (hoy se llamaría un anti-patrón) es unir ambas piezas para reducir la cantidad de código y optimizar la performance. Sin embargo, esta idea es antagónica al hecho de que la interfaz suele cambiar, o acostumbra depender de distintas clases de dispositivos (clientes ricos, browsers, PDAs); la programación de interfaces de HTML, además, requiere habilidades

muy distintas de la programación de lógica de negocios. Otro problema es que las aplicaciones tienden a incorporar lógica de negocios que van más allá de la transmisión de datos.

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

-Modelo. El modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

-Vista. Maneja la visualización de la información.

-Controlador. Interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- ✓ El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)
- ✓ El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- ✓ El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- ✓ El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los

CAPÍTULO 2: TENDENCIAS Y TECNOLOGÍAS ACTUALES A CONSIDERAR

cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.

Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.

✓ La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. La separación entre vista y controlador puede ser secundaria en aplicaciones de clientes ricos y, de hecho, muchos frameworks de interfaz implementan ambos roles en un solo objeto. En aplicaciones de Web, por otra parte, la separación entre la vista (el browser) y el controlador (los componentes del lado del servidor que manejan los requerimientos de HTTP) está mucho más taxativamente definida.

Entre las ventajas del estilo señaladas están:

✓ Soporte de vistas múltiples. Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación de Web pueden utilizar el mismo modelo de objetos, mostrado de maneras diferentes.

✓ Adaptación al cambio. Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares o PDAs. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo. Este patrón sentó las bases para especializaciones ulteriores, tales como Page Controller y Front Controller.

Entre las desventajas, se han señalado:

✓ Complejidad. El patrón introduce nuevos niveles de indirección y por lo tanto aumenta ligeramente la complejidad de la solución. También se profundiza la orientación a eventos del código de la interfaz de usuario, que puede llegar a ser difícil de depurar. En rigor, la configuración basada en

eventos de dicha interfaz corresponde a un estilo particular (arquitectura basada en eventos) que aquí se examina por separado.

✓ Costo de actualizaciones frecuentes. Desacoplar el modelo de la vista no significa que los desarrolladores del modelo puedan ignorar la naturaleza de las vistas.

Si el modelo experimenta cambios frecuentes, por ejemplo, podrían desbordar las vistas con una lluvia de requerimientos de actualización. Hace pocos años sucedía que algunas vistas, tales como las pantallas gráficas, involucraban más tiempo para plasmar el dibujo que el que demandaban los nuevos requerimientos de actualización. [11]

2.7.2 Active Record.

Patrón de software utilizado en aplicaciones robustas, que permite trabajar los registros de una tabla en una base de datos como instancias de una clase, por ejemplo Clientes ó Productos en los cuales podemos aplicar métodos Buscar, Guardar y Borrar sin necesidad de utilizar sentencias SQL. [12]

2.8 Sistemas Gestores de Bases de Datos (SGBD).

Los Sistemas Gestores de Bases de Datos (SGBD) pueden definirse como un paquete generalizado de software, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. [13]

Los SGBD deben proporcionar a los usuarios la capacidad de almacenar datos, acceder a ello y actualizarlos, de forma rápida, estructurada y ocultándole la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento),

Existen distintos objetivos que deben cumplir los SGBD:

- ✓ Independencia de los datos y los programas de aplicación.
- ✓ Minimización de la redundancia.
- ✓ Integración y sincronización de las bases de datos.
- ✓ Integridad de los datos.

- ✓ Seguridad y protección de los datos.
- ✓ Facilidad de manipulación de la información.
- ✓ Control centralizado.

La información es representada a través de tuplas, las cuales describen el fenómeno, proceso o ente de la realidad objetiva que se está analizando y se representan a través de tablas.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Interbase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

Todos los sistemas mencionados anteriormente facilitan el trabajo con la base de datos y tienen características que los diferencian, por ejemplo:

- ✓ **Oracle:** requiere de una licencia para poderlo utilizar, es decir, es necesario pagar para poder utilizarlo.
- ✓ **Microsoft SQL Server:** no es multiplataforma, solo puede ser utilizado con el sistema operativo Windows que está patrocinado por la compañía Microsoft, no es gratuito.
- ✓ **MySQL:** soporta un subconjunto de SQL92, además de ser multiplataforma y gratuito.
- ✓ **PostgreSQL:** soporta un subconjunto de SQL92.

2.8.1 MySQL.

MySQL es uno de los sistemas gestores de bases de datos más populares desarrollados bajo la filosofía de código abierto.

Entre las características disponibles de las últimas versiones se puede destacar:

- ✓ Disponibilidad en gran cantidad de plataformas y sistemas.
- ✓ Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones.
- ✓ Transacciones y claves foráneas.
- ✓ Conectividad segura.
- ✓ Replicación.
- ✓ Búsqueda e indexación de campos de textos.

Tiene la ventaja de ser un SGBD robusto, que puede almacenar gran cantidad de datos, es rápido, seguro, estable, gratuito y soporta múltiples lenguajes de programación, con lo que puede conectarse a una base de datos de este tipo a través de cualquiera de ellos.

Una de las desventajas de MySQL es que es gratis para aplicaciones de código abierto, de lo contrario hay que pagar licencia comercial. [14]

2.9 Proceso de desarrollo.

Cada día la producción de software busca adecuarse más a las necesidades del usuario, esto trae como consecuencia que aumente en tamaño y complejidad. Para lograr la productividad del software se necesita un proceso que integre las múltiples facetas del desarrollo del mismo. Se hace necesario definir la metodología de ingeniería del software que guiará el proceso de automatización, se ha escogido el Proceso Unificado de Desarrollo de Software (RUP).

El Proceso Unificado de Rational, (Rational Unified Process, de ahí las siglas RUP), fue publicado en 1998 como resultado de varios años de experiencia.

RUP es un proceso de desarrollo de software, o sea, conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos.

Es un proceso basado en componentes, que utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. No obstante, los verdaderos aspectos definitorios del Proceso Unificado se resumen en que está dirigido por casos de uso, este avanza a través de una serie de flujos de trabajo, que parten de los casos de uso; centrado en la arquitectura y es iterativo e incremental.

Está acompañado de una herramienta muy buena que soporta cada uno de los procesos que necesitamos: Rational Rose Enterprise Edition 2003.

Además cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

2.9.1 Lenguaje Unificado de Modelado (UML).

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

Sus creadores pretendieron con este lenguaje, unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas en un acercamiento estándar.

El UML permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas que estén involucradas en el proceso de desarrollo de los sistemas, esto se lleva a cabo mediante un conjunto de símbolos y diagramas.

El UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y proporciona un estándar que permite al analista de sistemas generar un anteproyecto de varias facetas que sean comprensibles para los clientes, desarrolladores y todos aquellos que estén involucrados en el proceso de desarrollo. Un modelo UML indica que es lo que supuestamente hará el sistema pero no como lo hará.

- ✓ De forma general las principales características son:
- ✓ Lenguaje unificado para la modelación de sistemas.
- ✓ Tecnología orientada a objetos.
- ✓ El cliente participa en todas las etapas del proyecto.
- ✓ Corrección de errores viables en todas las etapas.
- ✓ Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

Existen varias herramientas CASE (Computer-Aided Systems Engineering), que dan asistencia a analistas, ingenieros de software y desarrolladores durante el ciclo de vida de desarrollo de un software, pero es Rational Rose líder en el modelado del desarrollo de los proyectos y es ésta precisamente la que se utiliza en la modelación de este proyecto. La herramienta fue desarrollada por los creadores de UML, utilizando la notación estándar en la arquitectura de software. Esta herramienta integra todos los elementos que propone la metodología RUP para cubrir el ciclo de vida de un proyecto. [16]

2.10 Herramientas utilizadas.

2.10.1 Diseño Gráfico.

Macromedia Fireworks herramienta muy fácil de utilizar para crear cualquier tipo de gráficos, por su integración con Dreamweaver hace más cómodo el diseño de la aplicación, por lo que hacer complicados gráficos para la Web se convierte en una tarea muy fácil. [17]

Adobe Photoshop durante cualquier tipo de desarrollo, el usuario despliega sus conocimientos con total libertad haciendo uso de una infinidad de recursos. De principio a fin. [18]

2.10.2 Diseño de la interfaz.

Macromedia Dreamweaver MX es uno de los editores de desarrollo Web más utilizado a nivel profesional para la creación de sitios Web. Su amplio abanico de herramientas permite crear desde la más simple página Web personal hasta el sitio Web más completo y complejo para una gran empresa y utilizar casi todos los recursos de la Web.

Este editor de HTML profesional para el diseño, codificación y desarrollo de páginas, sitios y aplicaciones Web; permite la edición visual, o sea, crear páginas rápidamente sin escribir una línea de código, así como también la codificación manual. Dreamweaver ayuda además a construir aplicaciones Web dinámicas apoyadas en bases de datos.

Puede ser personalizado. Se pueden crear objetos y comandos propios, modificar los accesos directos de teclado, e incluso escribir código JavaScript para extender las capacidades del Dreamweaver con nuevos comportamientos. Soporta varias tecnologías del servidor para la construcción de aplicaciones Web, tales como: Macromedia ColdFusion, Microsoft ASP, Microsoft ASP.NET, Sun JavaServer Pages (JSP) y PHP. [19]

2.10.3 Zend Studio.

Zend Studio es uno de los ambientes de desarrollo integrado o Integrated Development Environment (IDE) disponible para desarrolladores profesionales que agrupa todos los componentes de desarrollo necesarios para ciclo de desarrollo de aplicaciones PHP. A través de un comprensivo conjunto de herramientas de edición, depurado, análisis, optimización y bases de datos, Zend Studio acelera los ciclos de desarrollo y simplifica los proyectos complejos. [20]

2.10.4 EMS SQL Manager for MySql.

Herramienta muy intuitiva y de gran alcance para la administración y el desarrollo del servidor de la base de datos de MySQL. Trabaja con cualquier versión de MySQL a partir del 3. 23 y ayudas de las características más recientes de MySQL, los procedimientos almacenados y las funciones, llaves extranjeras de InnoDB, funciones usuario-definibles, tablas transacción-seguras, nuevo protocolo de la autenticación de MySQL 5, MySQL 4, extensiones del privilegio del usuario x, charsets y ayuda de las colaciones para las bases de datos y las tablas.

Ofrece el un montón de herramientas de gran alcance para los usuarios experimentados tales como diseñador visual de la base de datos, constructor visual de la queries. Tiene una nueva interfaz gráfica avanzada. Su interfaz gráfico (GUI) le facilita la creación o edición de todos los objetos de la base de datos de una manera simple y directa, maneja a usuarios y administra privilegios del usuario, corrige metadata de las sentencias del SQL, del extracto y de la impresión, crea informes de la estructura de la base de datos en formato del HTML, y otros servicios que hagan su trabajo con el servidor de MySQL tan fácil como puede ser. [21]

2.10.5 Case Studio.

Herramienta profesional para el diseño de bases de datos, que permite crear visualmente los diagramas de relación de entidad (ERD) y los diagramas de flujos de datos (DFD), para varios sistemas de la base de datos como son DB2, MS SQL, MS acceso, Sybase ASE, interBase, MySQL, PostgreSQL y más. Su principal características, es su potente sistema de ingeniería inversa (DDL), que permite identificar y estructurar bases de datos ya existentes para poder trabajar con ellas sin problemas. El programa dispone de parches de idioma para un gran número de países. [22]

2.10.6 Rational Rose.

Rational Rose es una herramienta case desarrollado por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto. Permite comprender la aplicación y capturar y documentar el comportamiento del software. Brinda la posibilidad de reusar el diseño y el código. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue) y permite que los arquitectos y diseñadores practiquen el desarrollo orientado al modelado utilizando

UML, permitiéndoles producir modelos independientes a la, arquitectura de la plataforma del software y necesidades del negocio. [23]

Conclusiones

En este capítulo se hace un análisis detallado de las tecnologías y metodologías que son utilizados para el sistema propuesto. Se utilizarán herramientas de software libre para su desarrollo, con el objetivo de estandarizar el diseño de los módulos y facilitar el mantenimiento de los mismos.

El sistema usa la metodología RUP que junto con el Lenguaje Unificado de Modelado (UML) constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas. Para documentar el desarrollo del software se decidió utilizar la herramienta Rational Rose 2003 ya que esta herramienta es muy completa y ofrece amplias potencialidades.

Como lenguajes de programación se utilizará PHP por la razón de ser un lenguaje de programación utilizado con éxito en el desarrollo de millones de aplicaciones Web, además de poseer las características y ventajas mencionadas con anterioridad, el uso del Framework CodeIgniter nos traerá ventajas en tiempo de desarrollo, por estar diseñado con una arquitectura provechosa (MVC), y como todo framework se pueden reutilizar las clases de una forma más eficiente.

CAPÍTULO 3: CARACTERÍSTICAS DEL SISTEMA.

Antes de comenzar a desarrollar un sistema, es necesario comprender la organización y los procesos que tienen lugar en ella. Para lograr una mejor comprensión del problema a resolver y el común entendimiento entre clientes y desarrolladores; se realiza la modelación del negocio.

Este posibilita obtener una visión más clara del proceso, por ello en este capítulo se exponen las políticas y condiciones que deben cumplirse. Se describen los actores y trabajadores del negocio y el modelo de objetos. Además se reflejan los casos de uso del sistema, y su descripción.

3.1 Especificación de Requisitos.

Fase de especificación de requisitos es en la cual se debe conseguir el catálogo de requisitos del sistema que englobe: la definición de los objetivos del sistema, los requisitos de almacenamiento de información, la descripción de los actores del sistema, los requisitos funcionales, descritos a través de los casos de uso, los requisitos de interacción, en lo que se recogerá el sistema de navegación de la aplicación, la interacción con el usuario y los requisitos no funcionales.

3.2 Modelo de negocio propuesto

El primer paso del modelado del negocio consiste en capturar y definir los procesos de negocio de la organización bajo estudio. En el capítulo1 se hizo una descripción general de los procesos identificados en el negocio actual, así como un análisis crítico de la ejecución de los mismos. Teniendo en cuenta las deficiencias detectadas y bajo un análisis profundo de las fuentes de problemas potenciales se ha elaborado una propuesta de negocio que mantiene invariable el flujo de los procesos pero incurre en cambios de la ejecución de los mismos.

3.2.1 Actores del negocio

Un actor del negocio es cualquier individuo, grupo, organización, máquina o sistema de información externo que interactúa con el negocio. El término actor significa el rol que algo o alguien juega cuando interactúa con el negocio para beneficiarse de sus resultados. De acuerdo con esta idea un actor del negocio representa un tipo particular de usuario del negocio más que un usuario físico, ya que varios usuarios físicos pueden realizar el mismo papel en relación al negocio, o sea, ser instancias de un mismo actor.

Actores del Negocio:

Actores del Negocio	Descripción
Paciente	El paciente es el principal beneficiado con los resultados del negocio. El más interesado en recibir un buen servicio de salud, con lo cual se beneficia.

3.2.2 Trabajadores del negocio

Un trabajador define el comportamiento y las responsabilidades de un individuo que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio.

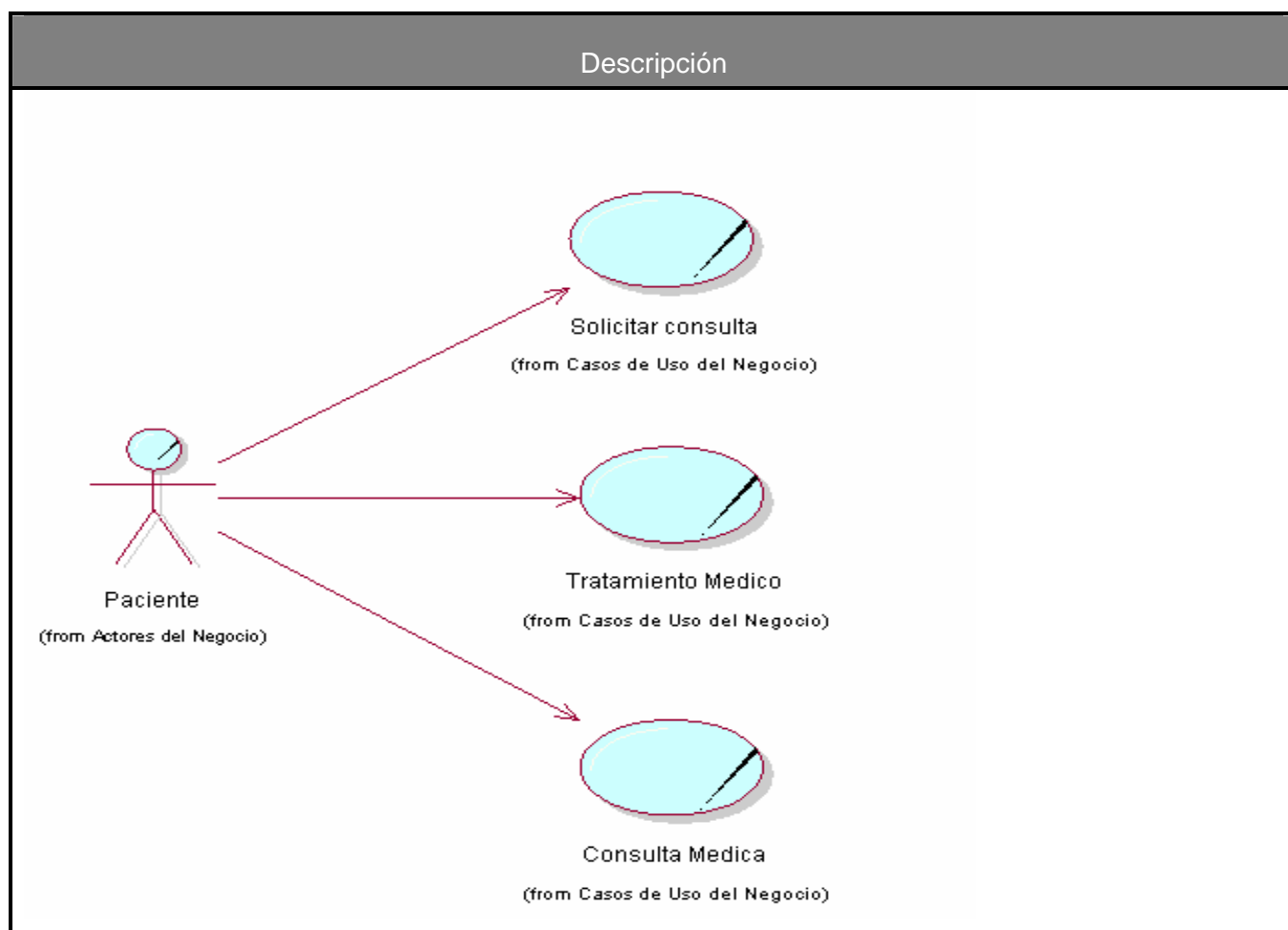
Trabajadores del negocio:

Trabajadores del Negocio	Descripción
Recepcionista	Es la encargada de registrar al paciente, comprobar si existe la Historia Clínica del paciente en el Hospital y si no tiene Historia Clínica es la encargada de tomarle los datos para confeccionarle una. Se encargará de planificarle al paciente los turnos y los tratamientos. Es la encargada de informar la cantidad de tratamientos por consultas que se elaboraron.
Médico Fisioterapeuta	Es el encargado de atender al paciente, diagnosticarle la enfermedad y ponerle un tratamiento en dependencia de la enfermedad diagnosticada. Es el único que puede darle de alta a un paciente. Planifica las acciones de salud de cada paciente.
Técnicos	Es el encargado de aplicarle al paciente el tratamiento orientado por el fisioterapeuta, y tiene la obligación de informarle a este sobre la evolución del mismo.

3.2.3 Diagrama de casos de uso del negocio.

El diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y su interacción con los actores del negocio.

Diagrama de Casos de Usos del Negocio.



Especificación de los Casos de Uso.

Caso de uso “Solicitar Consulta”.

Caso de Uso:	Solicitar Consulta
Actores:	Paciente
Trabajadores:	Recepcionista
Resumen:	El caso de uso se inicia cuando el paciente llega a la Sala de Rehabilitación solicitando atención médica, este es atendido por la recepcionista, la cual le toma los datos necesarios, y le crea un turno médico para que sea visto por un médico especialista (Fisioterapeuta).
Precondiciones:	
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1. El paciente se presenta en la Sala de Rehabilitación y solicita una consulta.	1.1 La recepcionista revisa en la Base de Datos si tiene una Historia Clínica.
	1.2 Pide al paciente el número de la Historia Clínica.
2. Le da el número de la Historia Clínica a la enfermera.	2.1 Busca la Historia Clínica, revisa los datos y habilita la Historia Clínica para que el especialista que lo va a atender tenga acceso a todos los datos del Paciente.
	3. 1 La enfermera le planifica una consulta para que lo vea el médico especialista y le comunica el día de la consulta.
3. El paciente recibe un turno para la consulta y se retira.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	1.1 Si la Enfermera no encuentra historia clínica crea una nueva.
Poscondiciones	

Caso de uso “Consulta médica”.

Caso de Uso:	Consulta Médica
Actores:	Paciente
Trabajadores:	Médico Especialista.
Resumen:	El caso de uso se inicia cuando el paciente entra a la Sala donde va a ser atendido por el especialista, este revisará los datos de la Historia Clínica del mismo y le hará previamente un diagnóstico para evaluar la situación del paciente y mandarle los tratamientos.
Precondiciones:	El paciente debe tener cita con el especialista.
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1. El paciente se presenta en la Sala del especialista.	1.1 El especialista realiza previamente un diagnóstico al paciente para evaluar su situación.
	1.2 El especialista le orienta al paciente los tratamientos médicos que va a recibir por los técnicos y lo remite con la recepcionista.
2. El paciente se dirige a la recepcionista.	2.1 La recepcionista busca un horario disponible para el tratamiento y se lo da a conocer al paciente.
3. El paciente escoge horario para recibir tratamiento.	3.1 La recepcionista planifica el tratamiento y se lo comunica al paciente.
4. Recibe el horario y se retira.	
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Poscondiciones	El especialista debe tener el listado de las citas de ese día.

Caso de uso “Tratamientos Médicos”.

Caso de Uso:	Tratamientos Médicos
Actores:	Paciente
Trabajadores:	Técnicos de Fisioterapia.
Resumen:	El caso de uso se inicia cuando el paciente entra a la Sala donde va a ser atendido por los técnicos que le aplicarán el/ o los tratamientos médicos, indicados por el especialista que lo atendió. Estos le informarán al especialista sobre la evolución del paciente en la Sala. Se evaluará el progreso del paciente en los tratamientos, así como se le dará de alta al mismo al finalizar los tratamientos médicos orientados.
Precondiciones:	El paciente debe tener un horario para los tratamientos.
Flujo Normal de Eventos	
Sección	
Acción del Actor	Respuesta del Negocio
1. El paciente se presenta en la Sala para recibir el tratamiento.	1.1 La recepcionista le pide el número de la Historia Clínica.
2. Le entrega el número de la Historia Clínica.	2.1 Busca la_Historia Clínica.
	2.2 Le entrega la_Historia Clínica al técnico en rehabilitación.
	2.3 El técnico en rehabilitación la recibe y llama al paciente para el tratamiento.
3. El paciente entra a la Sala de tratamiento.	3.1 El técnico le aplica el tratamiento.
4. Se retira de la Sala.	4.1 El técnico recoge los datos del tratamiento aplicado.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
Poscondiciones	El técnico debe tener el listado de las los pacientes que se atenderán ese día.

3.2.4 Diagramas de actividades

Un diagrama de actividad demuestra la serie de actividades que deben ser realizadas en un proceso del negocio, así como las distintas rutas que pueden irse desencadenando. Este es dividido en canales, donde cada canal representa el actor que está llevando a cabo la actividad y muestra cómo se utilizan las entidades del negocio.

Diagrama “Solicitar Consulta”.

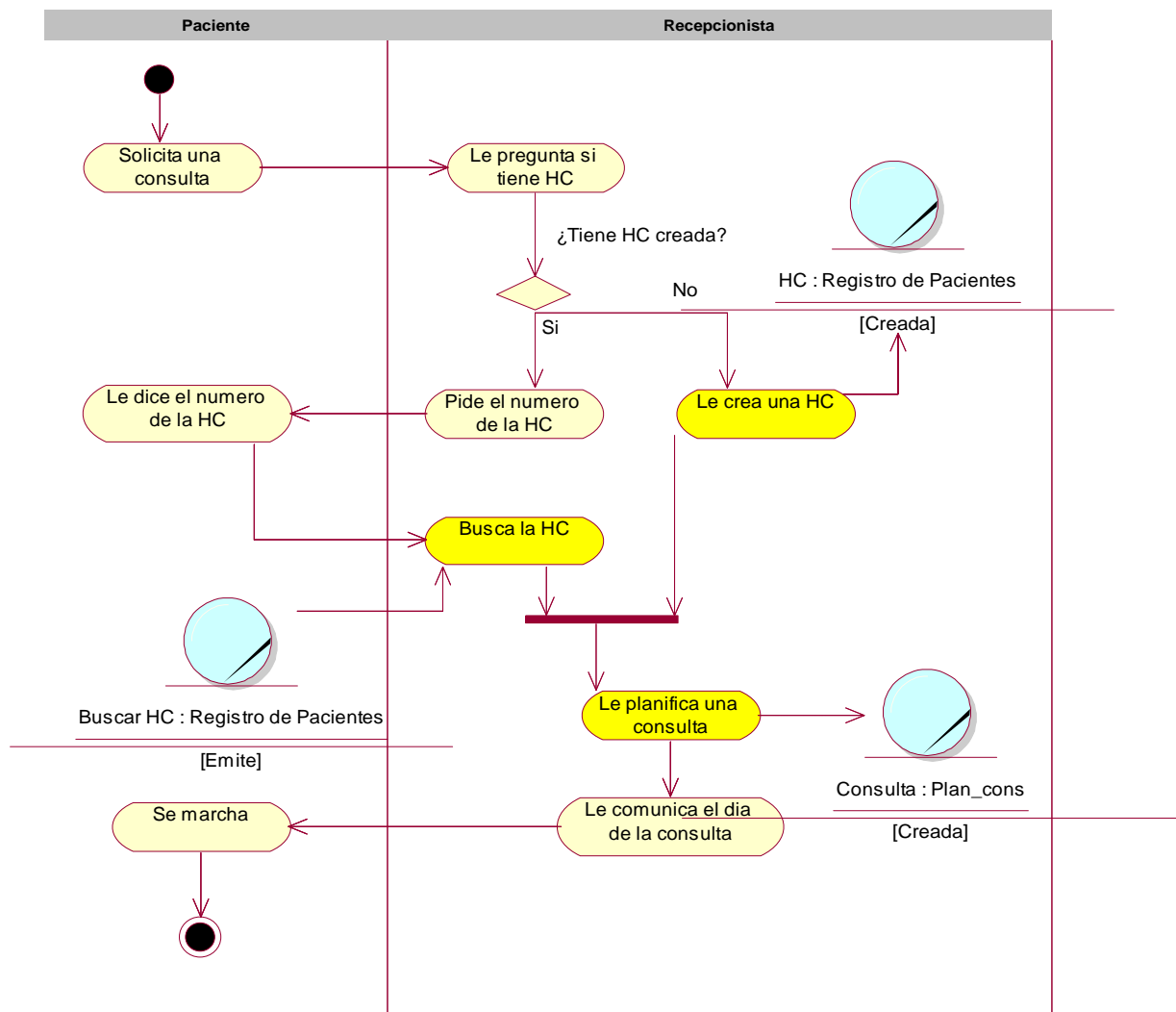


Diagrama “Consulta Médica”.

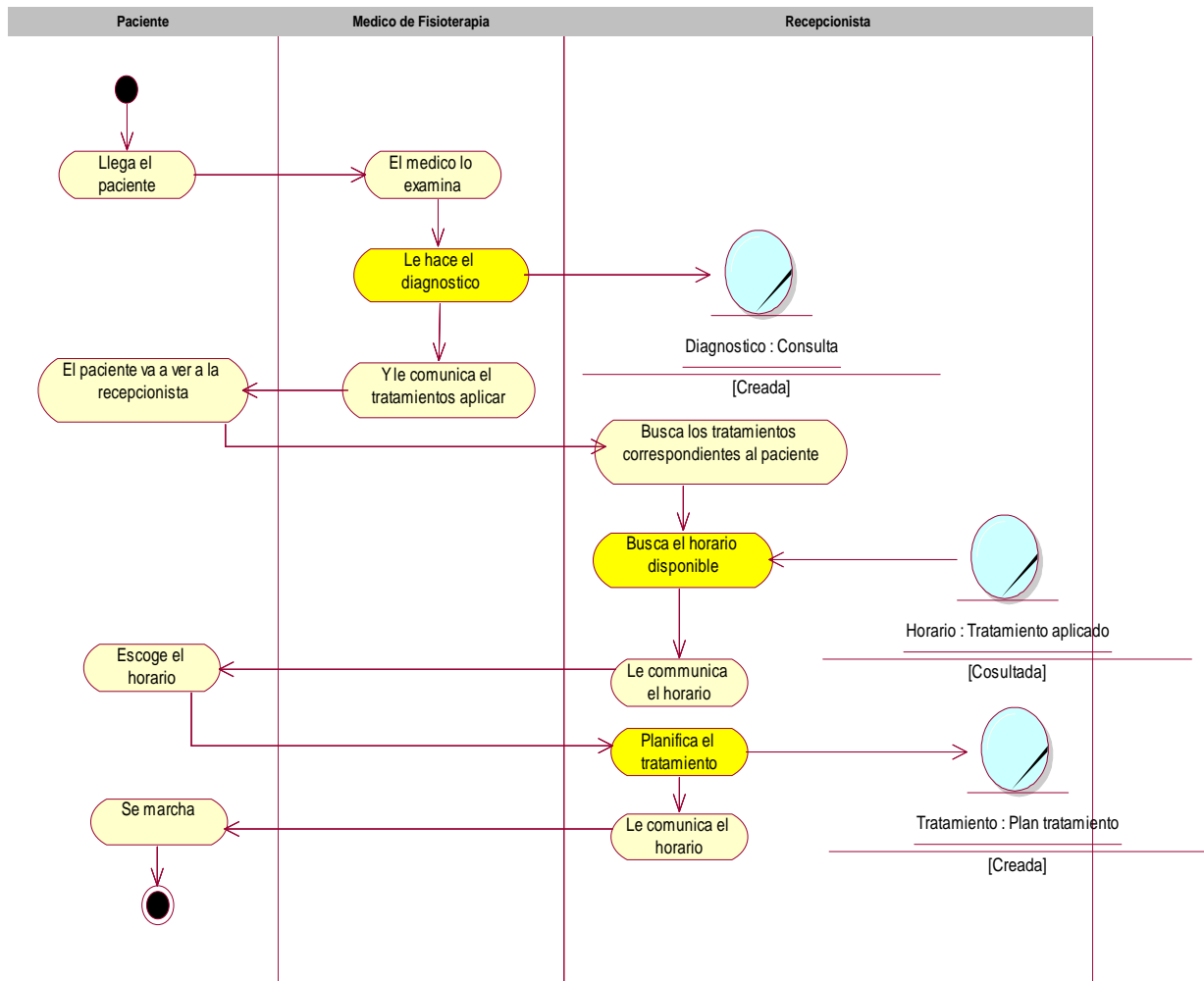
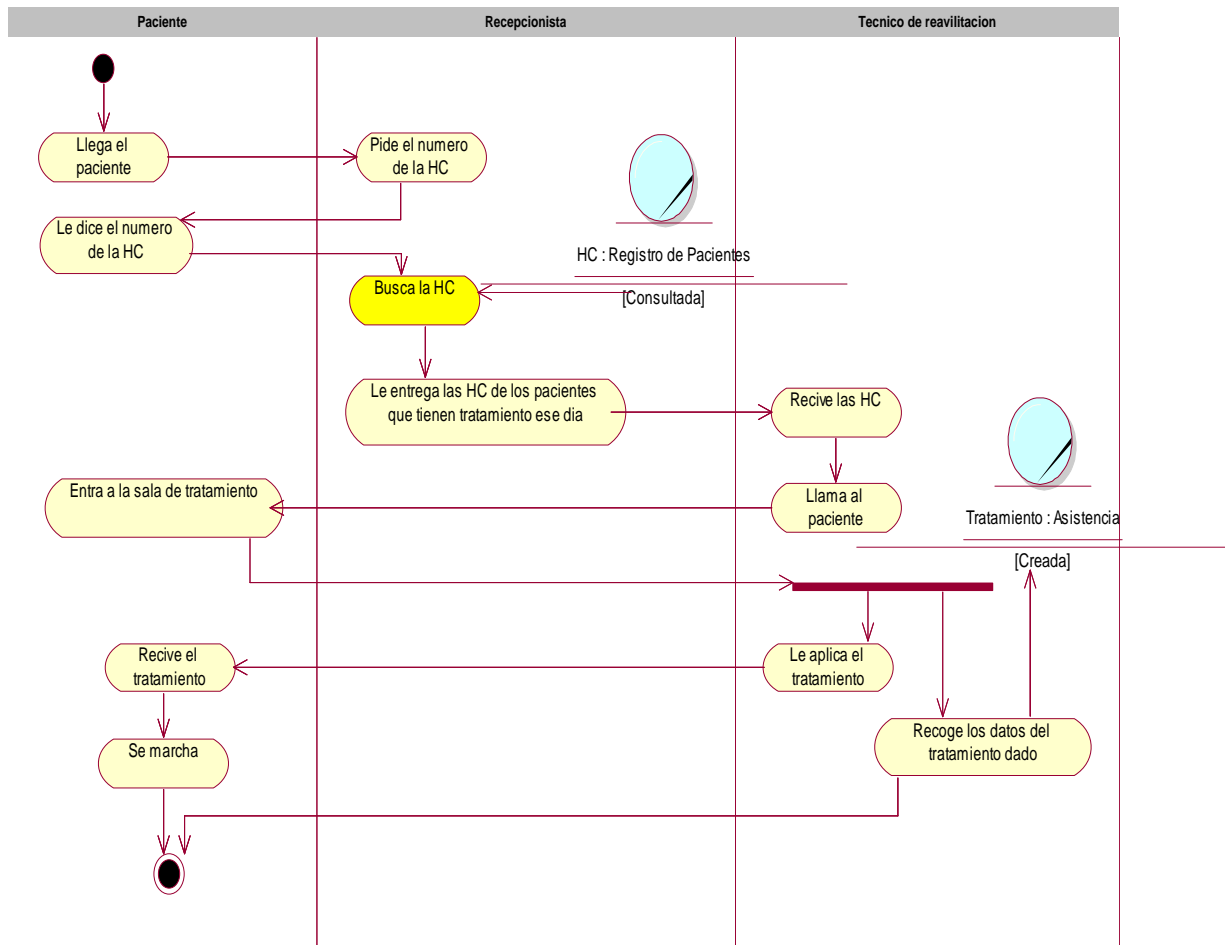


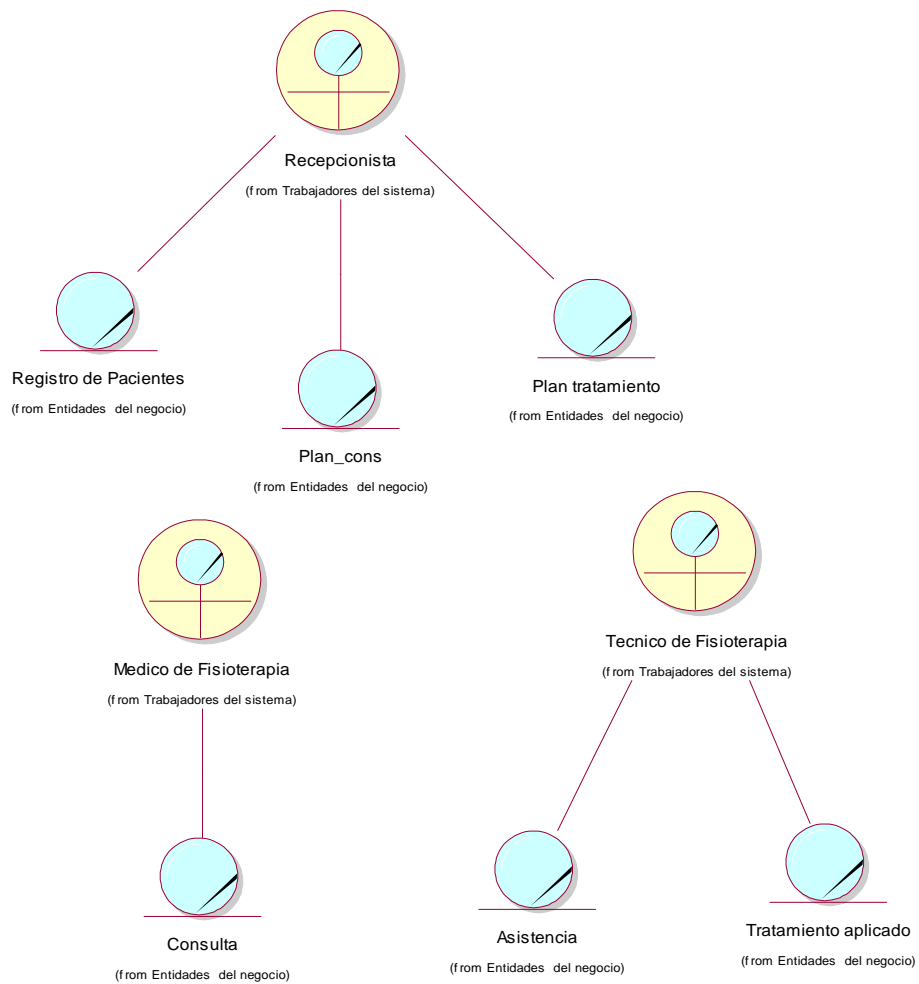
Diagrama “Tratamientos Médicos”.



3.2.5 Diagrama de clases del modelo de objetos

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo.

Modelo de Objetos



3.3 Modelo del sistema

3.3.1 Definición de los requisitos funcionales

Los requerimientos funcionales son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema y que están fuertemente ligados a las opciones del programa.

Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

R.1 Autenticar Usuario

R.2 Gestionar Personal médico

R.2.1 Registrar Personal médico

R 2.2 Modificar Personal médico

R.3 Buscar paciente

R.4 Planificar Consultas

R.4 Planificar Tratamientos

R.5 Generar reportes

R.6 Gestionar Pacientes

R 6.1 Registrar Paciente

R 6.2 Modificar Paciente

R.7 Gestionar codificadores

R 7.1 Actualizar en los Codificadores

R 7.2 Agregar en los Codificadores

R 7.3 Eliminar en los Codificadores

R.8 Realizar consulta.

R.9 Asistencia al tratamiento.

R.10 Gestionar Usuario

R.10.1 Registrar Usuario

R 10.2 Modificar Usuario

R 10.3 Eliminar Usuario

3.3.2 Definición de los requisitos no funcionales.

Los requerimientos no funcionales son características que describen alguna forma o restricción para la realización de algún requerimiento (funcionalidad) o conjunto de ellas e inclusive todos los requerimientos. Se consideran los atributos del sistema, propiedades que debe tener el producto.

A continuación se muestran los requerimientos no funcionales:

✓ **Apariencia o interfaz externa**

La interfaz no contiene muchas imágenes para no demorar las respuestas al usuario. El diseño de la interfaz es sencillo y claro de usar con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones. Es formal, serio y con una navegación sugerente, todo esto teniendo en cuenta el fin con el que se desarrolla la aplicación.

✓ **Usabilidad**

El sistema ha sido diseñado de tal forma que permite a los usuarios tener control en todo momento del sitio Web, evitando para eso la sobrecarga de información y para cada proceso una muestra a través de mensaje de las acciones hechas por los usuarios. Los colores usados permiten el acceso a usuarios con problemas de distinción de color, se reduce el tiempo de latencia, además, requiriendo un mínimo proceso de aprendizaje.

✓ **Rendimiento**

La disponibilidad de trabajo en red contra el servidor es constante. Se garantiza que la respuesta a solicitudes de los usuarios del sistema sea en un período de tiempo breve para evitar la acumulación de trabajo por parte de los responsables. El sistema deberá de ser lo más estable y confiable posible.

✓ **Soporte**

Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra. El sistema es de fácil instalación

✓ **Ayuda y documentación en línea**

El sistema brinda a los usuarios una buena ayuda en línea de modo que si el usuario presenta algún problema pueda acudir al mismo, así como una documentación apropiada para el mejor trabajo con el mismo.

✓ **Software**

Para el funcionamiento del sistema en el servidor es necesario el Sistema Operativo Windows o Linux, Windows 98 o superior, y para Linux alguna distribución del Debían u otro que tenga el usuario al alcance.

✓ **Hardware**

Se necesitan como requerimientos mínimos una PC con procesador Pentium II o superior, una memoria RAM de 128 o superior disco duro preferiblemente de 10 GB o más, con tarjeta de red no importando la velocidad.

✓ **Portabilidad**

No es necesario especificar ningún sistema operativo para trabajar con la aplicación, ya que este podrá ser accesible desde cualquiera. El producto corre sobre una plataforma Web, codificada en “PHP5” y sus sistemas de bases de datos en My SQL.

✓ **Seguridad**

El sistema se encarga de controlar los diferentes niveles de acceso y funcionalidad de usuarios al sitio, de identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema, este tendrá acceso a las opciones que le son referidas según su rol en el sistema. Garantiza que la información sea vista únicamente por quien tiene derecho a verla. Se hacen validaciones de la información tanto en el cliente como en el servidor, no obstante los usuarios acceden de manera rápida y operativa al sistema sin que los requerimientos de seguridad se conviertan en un retardo para ellos.

✓ **Confidencialidad**

Toda la información está protegida del acceso no autorizado, los administradores de sistema son los únicos que podrán transformar la información, los operadores solo podrán ver la información.

✓ **Disponibilidad**

Se garantiza a los usuarios del sistema el acceso a la información solicitada en todo momento (si tiene permiso para ello).

✓ **Políticos-Culturales**

Toda modificación al funcionamiento establecido en los requerimientos será realizada por la Dirección de Salud Pública conjuntamente con el personal de los centros de rehabilitación.

✓ **Restricciones en el diseño y la implementación**

Es una aplicación Web desarrollada con la tecnología para creación de páginas Web dinámicas PHP5 y con gestor de base de datos en My SQL.

✓ **Legales**

El sistema se basa en un estándar que se rige por normas internacionales y cumple con las normas y leyes establecidas en nuestro país.

✓ **Confiabilidad**

La herramienta de implementación a utilizar tiene soporte para recuperación ante fallos y errores.

✓ **Restricciones**

Se utiliza UML para lograr una mejor documentación del sistema y como herramienta de apoyo Rational Rose. Se utiliza como lenguaje de programación PHP5 y el gestor de base de datos My SQL.

3.3.3 Actores del Sistema.

Para cumplir con los objetivos propuestos al inicio del este trabajo. Teniendo en cuenta todos los requerimientos planteados, se definen varios roles para diferenciar el nivel de acceso al sistema, Administrador, Recepcionista, Técnicos.

Los actores del sistema pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema, y pueden intercambiar información con él o ser recipientes pasivos de información. En este caso los actores que interactúan con el sistema se definen a continuación en la tabla.

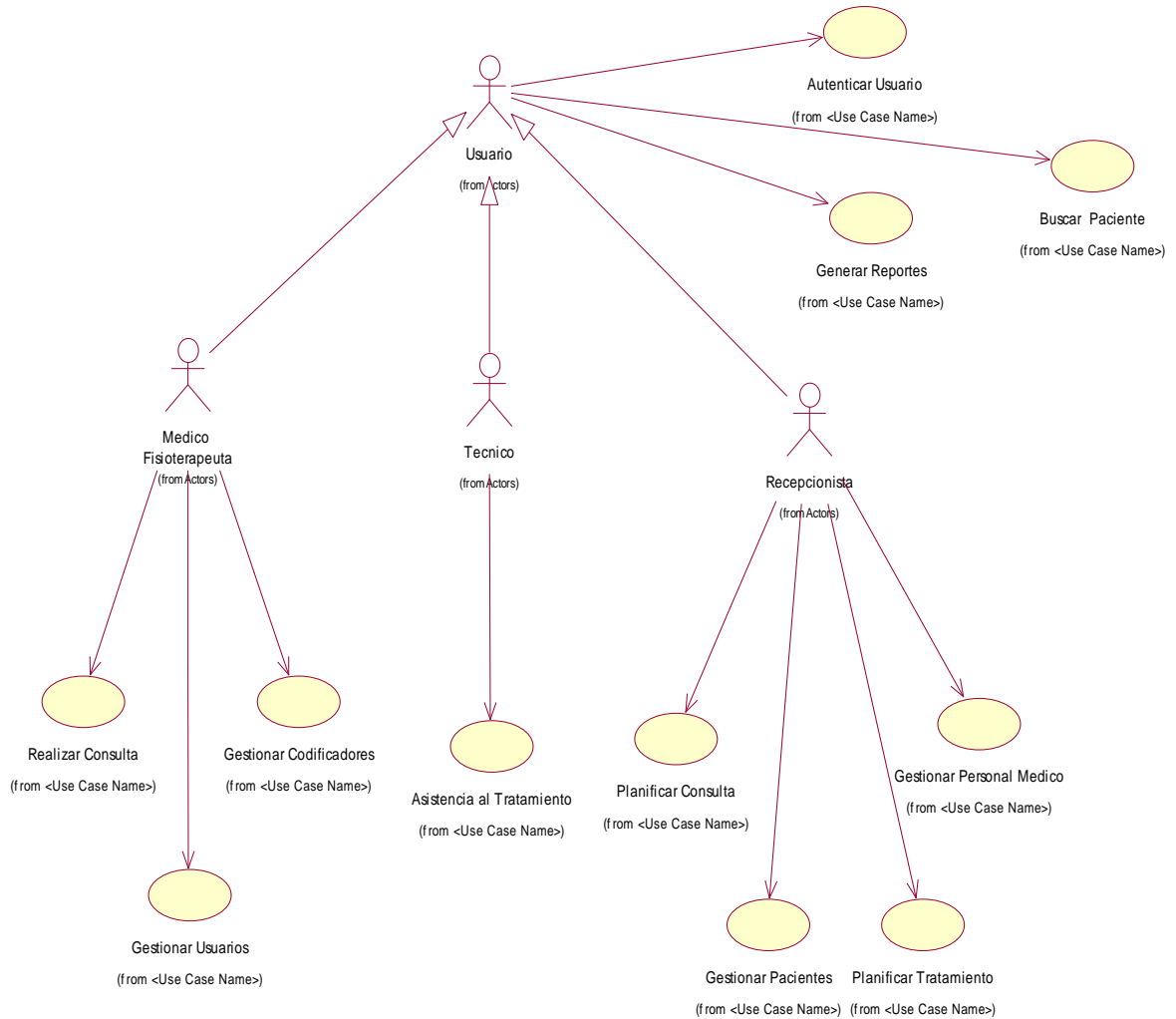
Actores del sistema.

Actores del Sistema	Descripción
Usuarios	Va a ser todo aquel personal que va a tener algún rol en el sistema, es decir los que van a poder trabajar con la aplicación.
Administrador	Es el encargado de realizar las configuraciones del sistema, la actualización de los codificadores (agregar nuevos, actualizar los existentes, eliminarlos etc.), además de configurar los niveles de accesos de los demás usuarios del sistema y trabaja con todas las funcionalidad del sistema.

CAPÍTULO 3: CARACTERÍSTICAS DEL SISTEMA

Recepcionista	Tiene acceso para registrar y actualizar los datos de los pacientes en sistema, planifican las consultas y los tratamientos de los pacientes y es el encargado de enviar reporte estadístico con datos de registro de pacientes.
Técnico	Son los encargados de registrar en el sistema las asistencias de los pacientes al tratamiento, así como emitir reportes referentes a los tratamientos aplicados.

3.3.4 Diagrama de Casos de Uso del Sistema a automatizar.



3.3.5 Descripción del Caso de Uso del Sistema.

“Autenticar usuario”.

Caso de uso:	Autenticar Usuario
Actores:	Usuario (que puede ser Recepcionista, Técnico, Médico Fisioterapeuta)
Descripción:	Permite que solamente puedan acceder a la aplicación todos aquellos que trabajaran directamente con la misma, dándole acceso a operar en el, según los vínculos a los que tienen permiso.
Referencias:	RF1

“Gestionar Personal Médico”.

Caso de uso:	Gestionar Personal médico
Actores:	Recepcionista
Descripción:	Aquí solo tiene acceso la recepcionista, esta tiene la posibilidad de agregar y/o modificar un personal médico de la base de datos.
Referencias:	R.2, R.2.1, R.2.2

“Buscar Paciente”.

Caso de uso:	Buscar paciente
Actores:	Recepcionista, Técnico, Médico Fisioterapeuta
Descripción:	Cuando se solicita buscar alguna información referente al paciente. Se define el criterio de búsqueda y luego se muestra los datos.
Referencias:	RF2

“Planificar Consultas”.

Caso de uso:	Planificar Consultas
Actores:	Recepcionista
Descripción:	Cuando la recepcionista solicita planificar una consulta, se muestra los días disponibles y se selecciona el día que el paciente realizara la consulta, realizando satisfactoriamente la planificación del paciente.

Referencias:	RF3

“Planificar Tratamientos”.

Caso de uso:	Planificar Tratamientos
Actores:	Recepcionista
Descripción:	El sistema mostrara una interfaz con los días disponibles y se selecciona el día con la hora que el paciente se realizará el tratamiento.
Referencias:	RF4

“Generar Reportes”.

Caso de uso:	Generar reportes
Actores:	Recepcionista, Técnico, Médico Fisioterapeuta
Descripción:	El sistema da la posibilidad de escoger el reporte que va generar, y le mostrando los datos a recoger, dándole la posibilidad de guardarlos como pdf.
Referencias:	RF5

“Gestionar Pacientes”.

Caso de uso:	Gestionar Pacientes
Actores:	Recepcionista
Descripción:	Aquí solo tiene acceso la recepcionista, el sistema da la posibilidad de agregar y/o modificar un paciente a la base de datos, verificando posteriormente que se haya realizado correctamente la acción seleccionada.
Referencias:	R.6, R.6.1, R.6.2

“Gestionar Codificadores”.

Caso de uso:	Gestionar Codificadores
Actores:	Médico Fisioterapeuta
Descripción:	Aquí solo tiene acceso el Médico Fisioterapeuta, el sistema da la posibilidad agregar, modificar u eliminar en los codificadores existentes en la base de datos, verificando posteriormente que se haya realizado correctamente la acción seleccionada.
Referencias:	RF7

“Realizar Consulta”.

Caso de uso:	Realizar Consulta
Actores:	Médico Fisioterapeuta
Descripción:	Aquí solo tiene acceso el Médico Fisioterapeuta, el sistema le muestra los datos que se deben llenar para realizar la consulta.
Referencias:	RF8

“Asistencia al Tratamiento”.

Caso de uso:	Asistencia al tratamiento
Actores:	Técnico
Descripción:	Una ve que el paciente se presenta en la sala a realizarse el tratamiento, el técnico solicita la hoja de asistencia del paciente y actualiza los datos correspondientes a ese día en el tratamiento.
Referencias:	RF10

“Gestionar usuario”.

Caso de uso:	Gestionar Usuario
Actores:	Médico Fisioterapeuta
Descripción:	Permite acceder a este vínculo solamente el médico Fisioterapeuta, el sistema brinda la posibilidad de efectuar las operaciones de agregar,

	<p>modificar y/o eliminar un usuario.</p> <p>Garantizar que los datos de cada Usuario se registren.</p>
Referencias:	R.11 , R.11.1, R.11.2, R.11.3

Casos de Uso por ciclo.

Cód	Nombre del CU	Cic lo	Justificación de la selección.
CU-1	Autenticar	1	Se decide desarrollar en el primer ciclo debido a la importancia que tiene el manejo de la seguridad del sistema con el objetivo de evitar la pérdida o alteración de información por personal no autorizado al mismo.
CU-2	Gestionar Personal médico	1	Se decide desarrollar en el primer ciclo debido a que este caso de uso es necesario para el desarrollo de otros casos de usos, sin el personal médico no ocurriría ninguna acción en el negocio.
CU-3	Buscar paciente	1	Se decide desarrollar en el primer ciclo ya que está relacionado con otros casos de usos, los cuales no se podría realizar sin este implementado.
CU-4	Planificar Consultas	1	Se decide desarrollar en el primer ciclo por ser una de las opciones que dan inicio al negocio del sistema.
CU-5	Planificar Tratamientos	1	Se decide desarrollar en el primer ciclo por ser una actividad fundamental a la hora de atender a los pacientes.
CU-6	Generar reportes	1	Se decide desarrollar en el primer ciclo por ser una funcionalidad de gran importancia a la hora de evaluar el desempeño de la clínica y sus profesionales.
CU-7	Gestionar Pacientes	1	Se decide desarrollar en el primer ciclo ya que es el caso de uso que da inicio al negocio del problema.
CU-8	Gestionar codificadores	1	Se decide desarrollar en el primer ciclo ya que en la aplicación existe una gran cantidad de codificadores.

CU-9	Realizar consulta.	1	Se decide desarrollar en el primer ciclo ya que otros casos de usos dependen de este para poderse realizar, adema de constituir una prioridad para el sistema.
CU-10	Asistencia al tratamiento.	1	Se decide desarrollar en el primer ciclo ya que es uno de los casos de uso que constituyen una prioridad inmediata en la aplicación.
CU-11	Gestionar Usuario	1	Se decide desarrollar en el primer ciclo ya que es el caso de uno de los casos de usos que asegura la seguridad de la aplicación.

Conclusiones

En este capítulo, se realizó el análisis del negocio, se generaron los artefactos que propone RUP en los flujos de trabajo: modelamiento del negocio y levantamiento de requisitos. Se describen cada uno de los casos de usos y se le dan prioridad a aquellos que serán implementados en el primer ciclo de desarrollo.

Se analizaron las características del sistema, se especificaron los requerimientos funcionales y no funcionales del sistema, obteniéndose los Casos de Uso y las cualidades del sistema. Además, se describen los casos de uso para una mejor comprensión de los mismos.

CAPÍTULO 4: ANALISIS Y DISEÑO DEL SISTEMA.

Tras la definición y descripción, en el anterior capítulo, de las funcionalidades deseadas y necesarias del sistema propuesto; se hace necesario definir cómo se desarrollará el mismo.

Este capítulo, tiene como objetivo plantear la concepción general del diseño, del sistema propuesto y cómo se implementa. Se presentan los diagramas de clases Web, que detallan la interacción de las distintas páginas; se estructura la información que se desea persista, a través del diseño de la base de datos. También son descritos, los estándares de diseño y programación seguidos.

4.1 Análisis.

El flujo de trabajo de análisis de sistema tiene esencialmente tres diferencias con respecto a la mayoría de las propuestas. Estos son: los diferentes medios de almacenamiento de información, la importancia de la interfaz y los aspectos navegacionales. Por ello, habrá que incluir el estudio de estos aspectos en el análisis para después pasarlos e incluirlos en el diseño.

4.2 Diagrama de Clases de Análisis.

Los diagramas de clases presentan la estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc.). Son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). El diagrama de clases de más alto nivel (main class diagram), será lógicamente un dibujo de los paquetes que componen el sistema. A su vez cada paquete tendrá un main class diagram que muestra las clases del paquete

Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, su cardinalidad (cuantos objetos intervienen en la relación) y su opcionalidad (cuando un objeto es opcional el que intervenga en una relación). La descripción de clases complejas se puede documentar con diagramas de estados

Diagramas de Clases “Autenticar Usuario”.

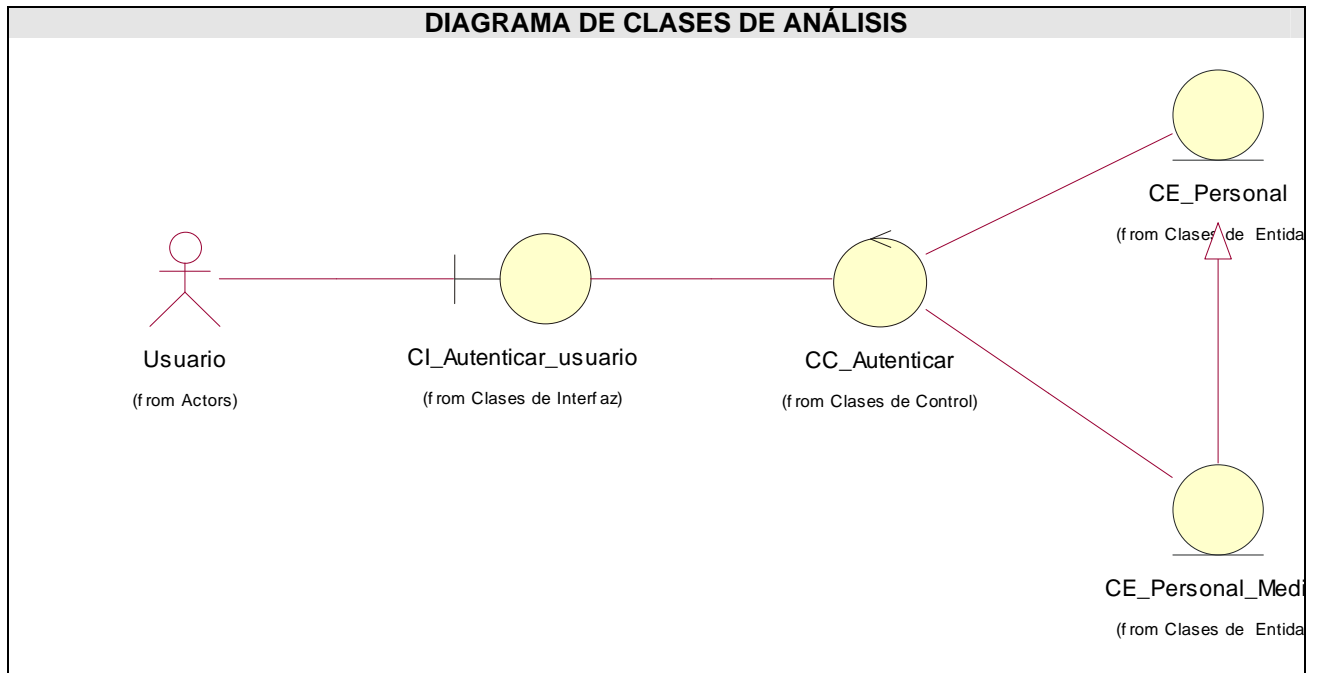


Diagrama de Clases “Asistencia al Tratamiento”.

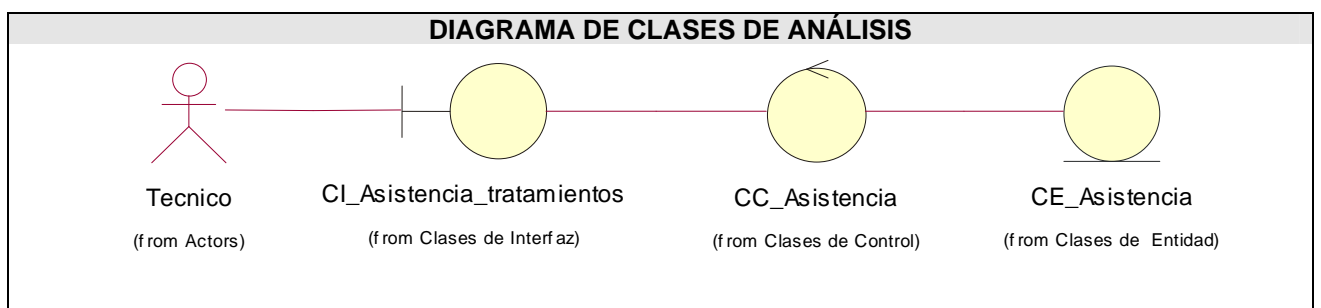


Diagrama de Clases “Buscar Paciente”.

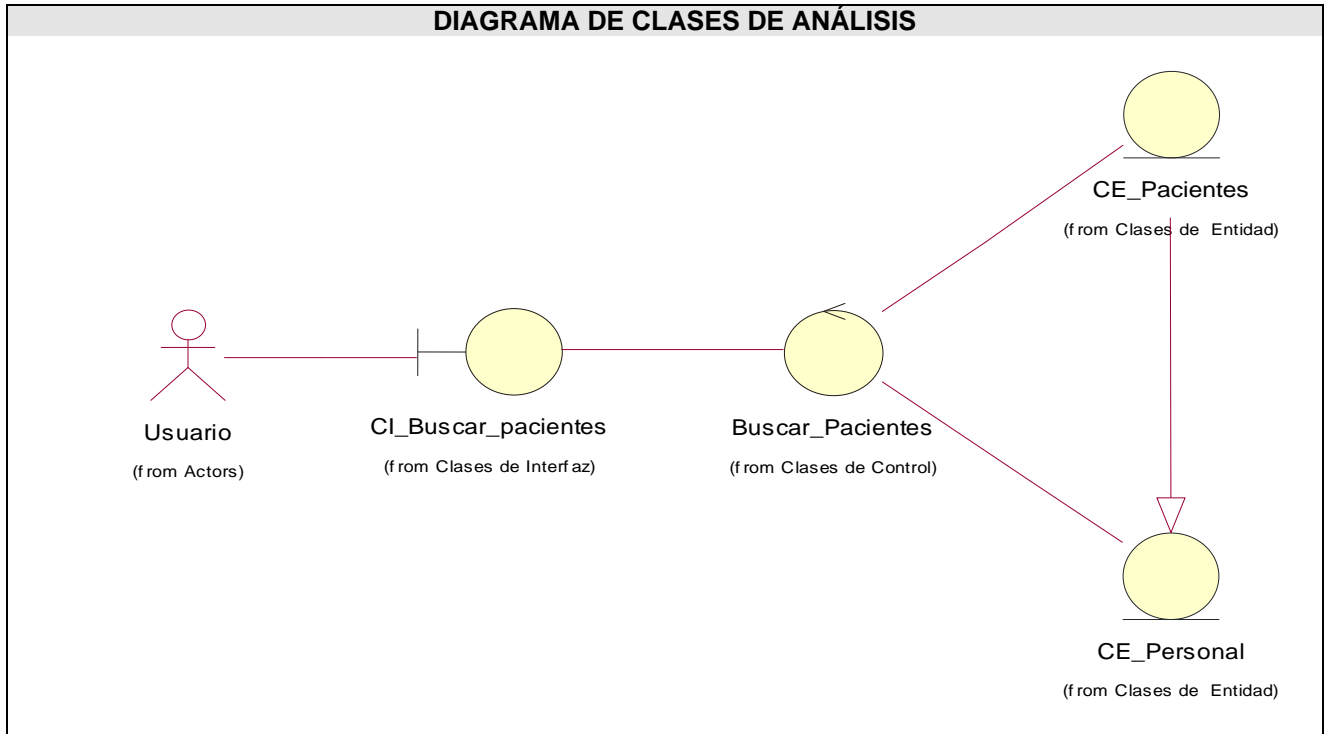


Diagrama de Clases “Generar Reportes”.

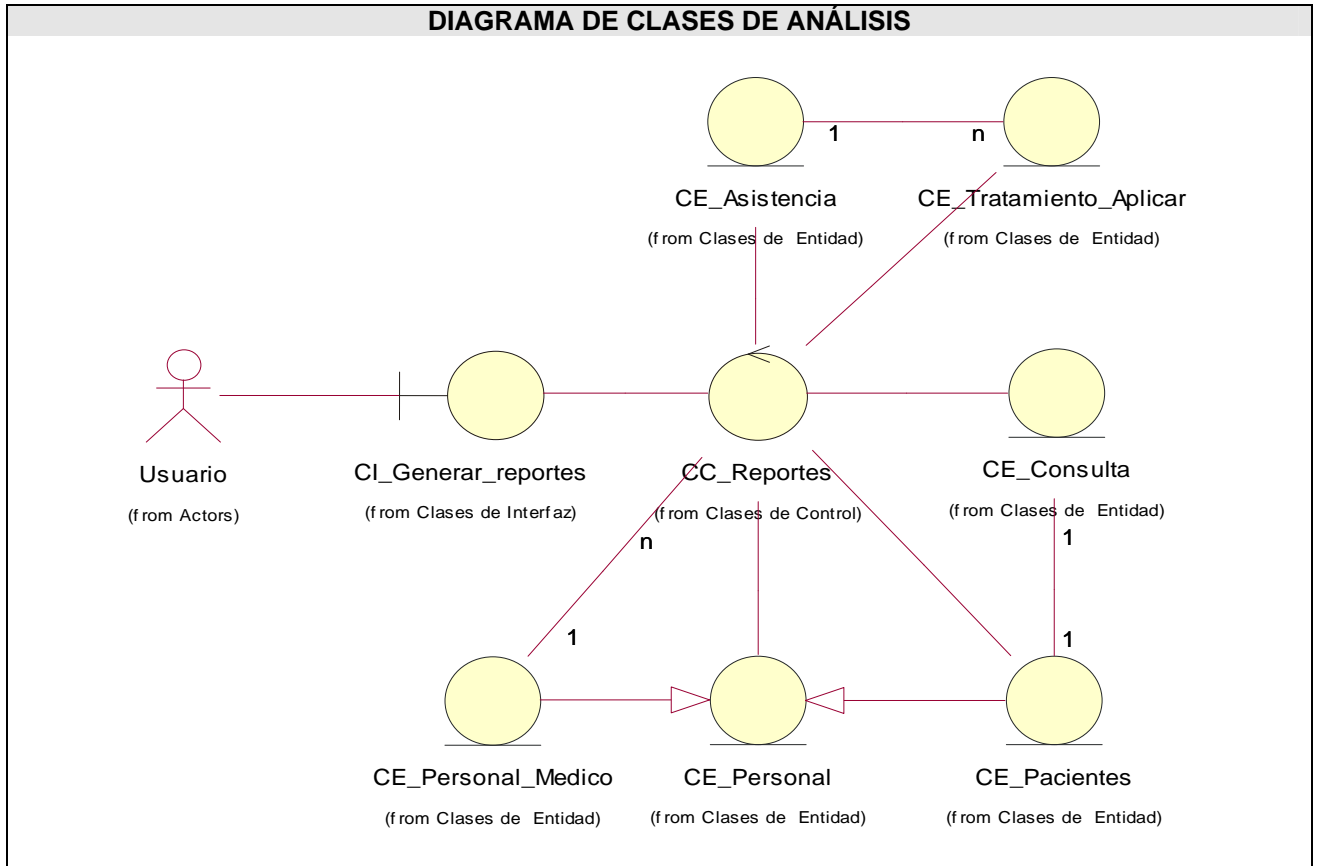


Diagrama de Clases “Gestionar Codificadores”

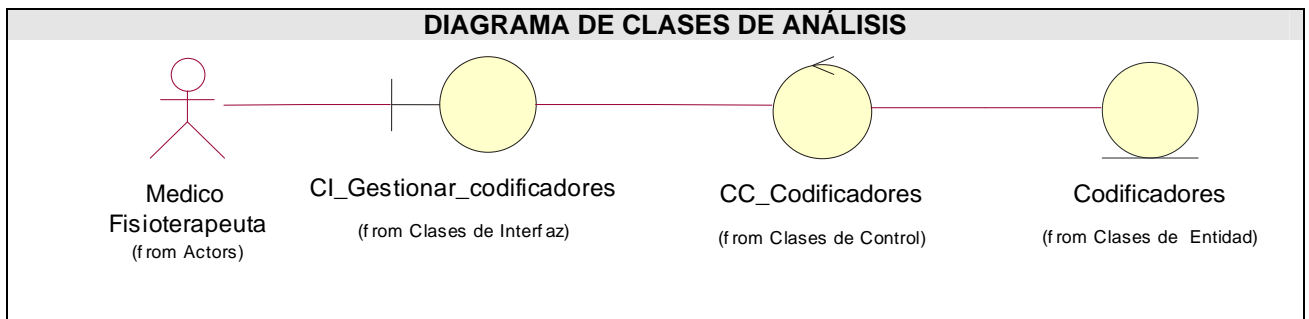


Diagrama de Clases “Gestionar Pacientes”.

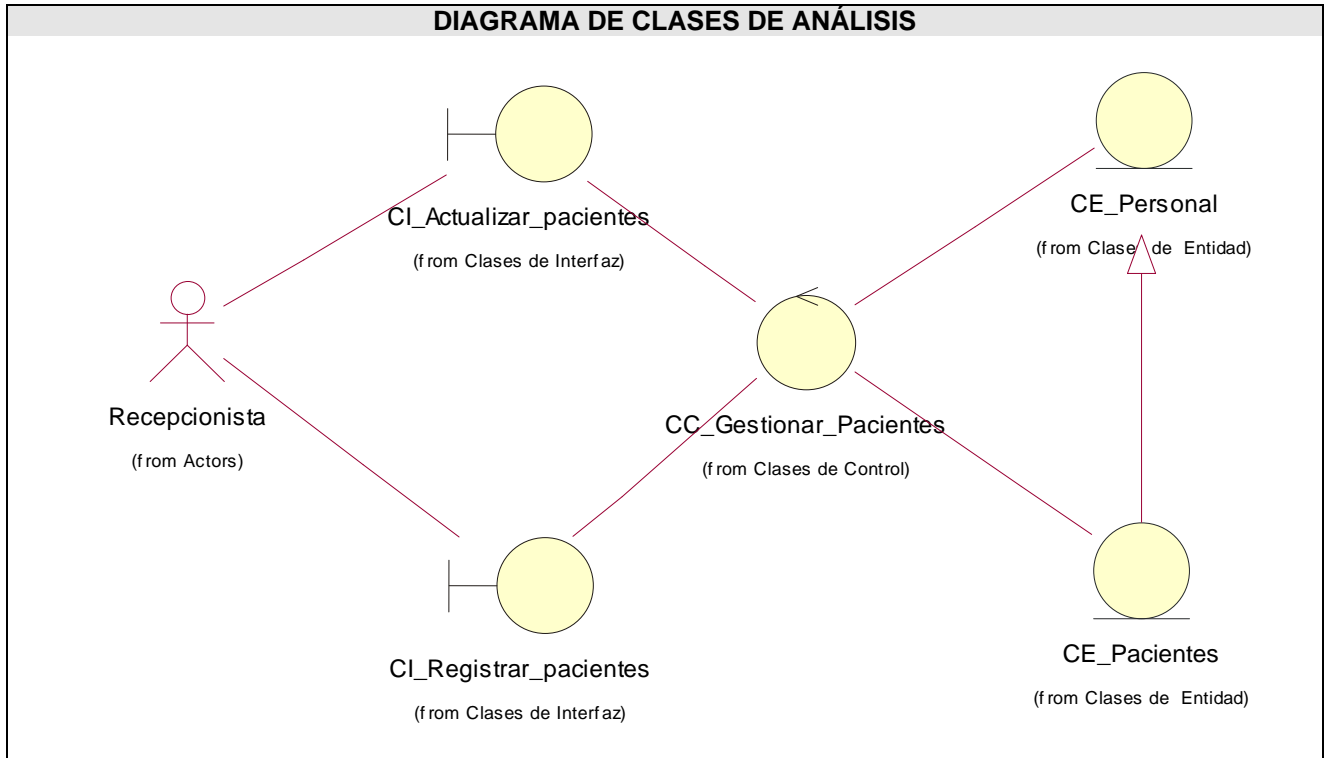


Diagrama de Clases “Gestionar Personal Médico”.

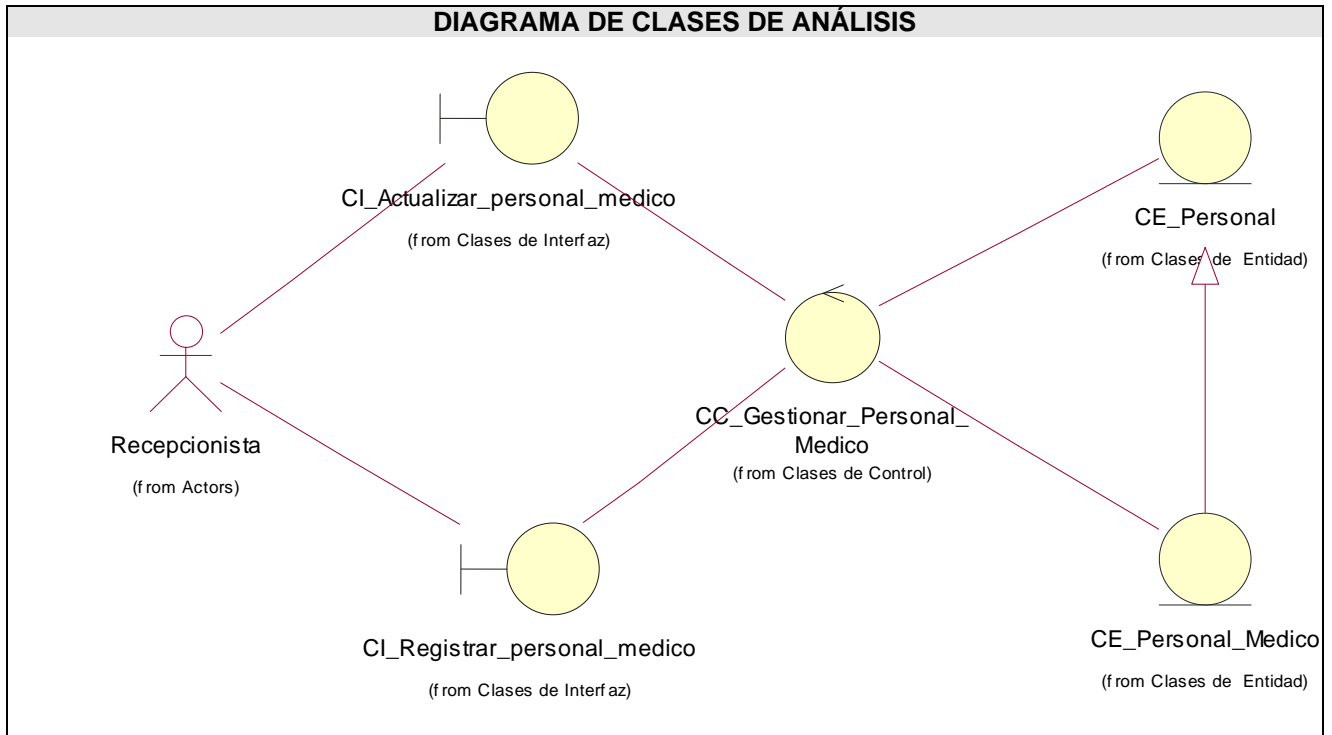


Diagrama de Clases “Gestionar Usuarios”.

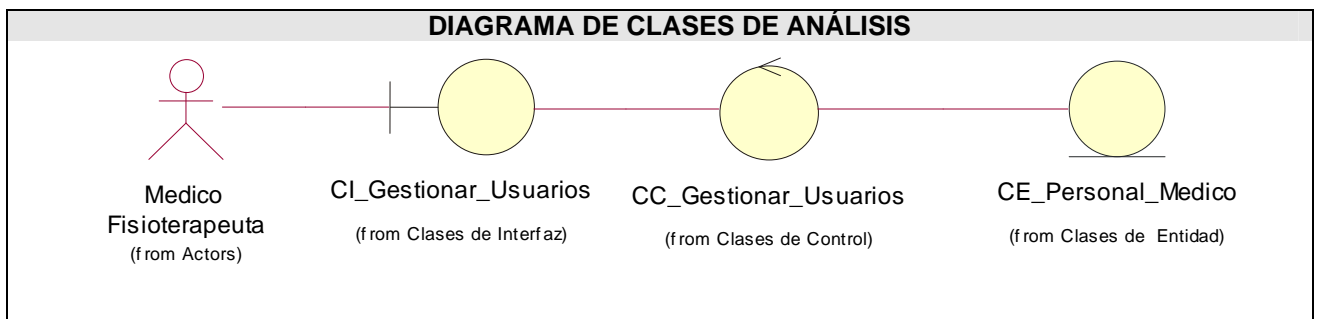


Diagrama de Clases “Planificar Consulta”.

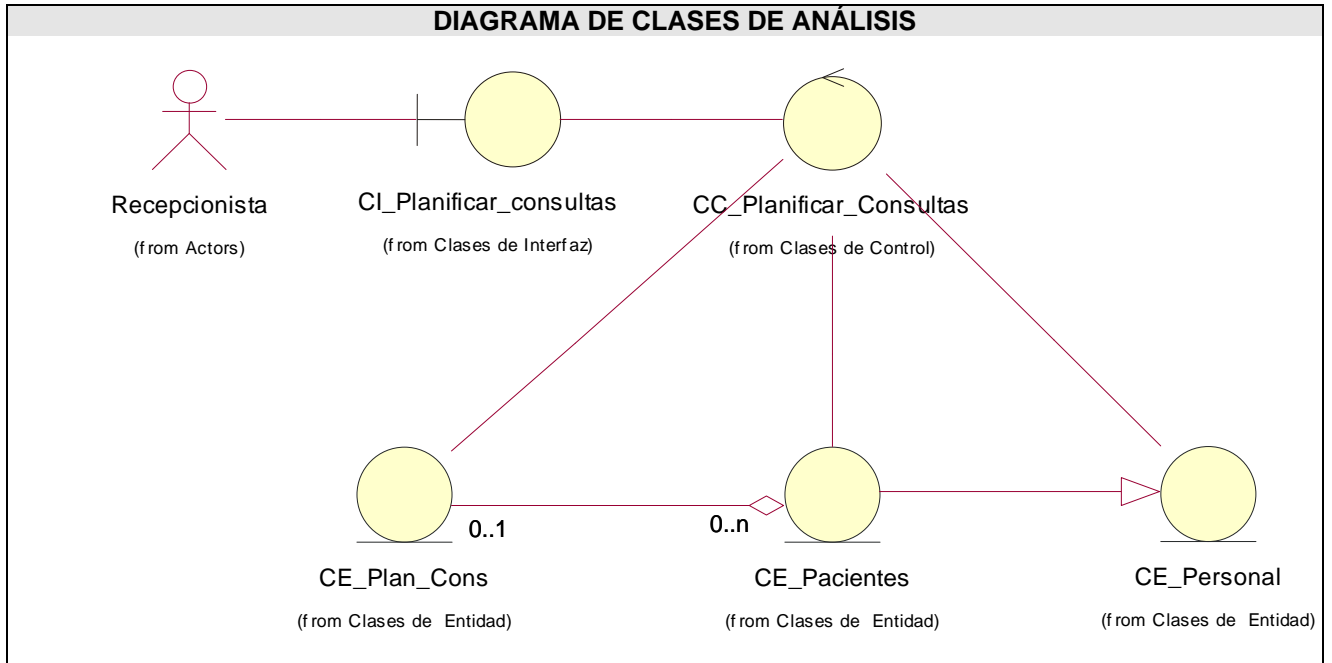


Diagrama de Clases “Planificar tratamientos”.

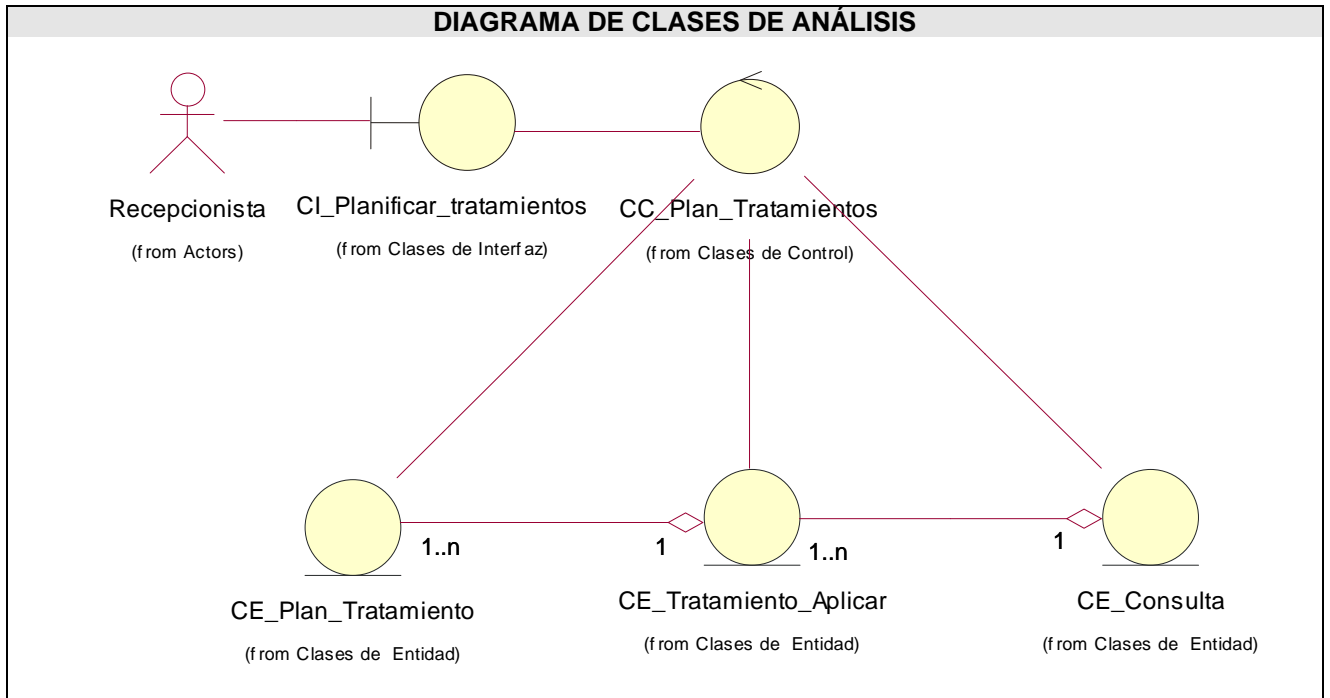
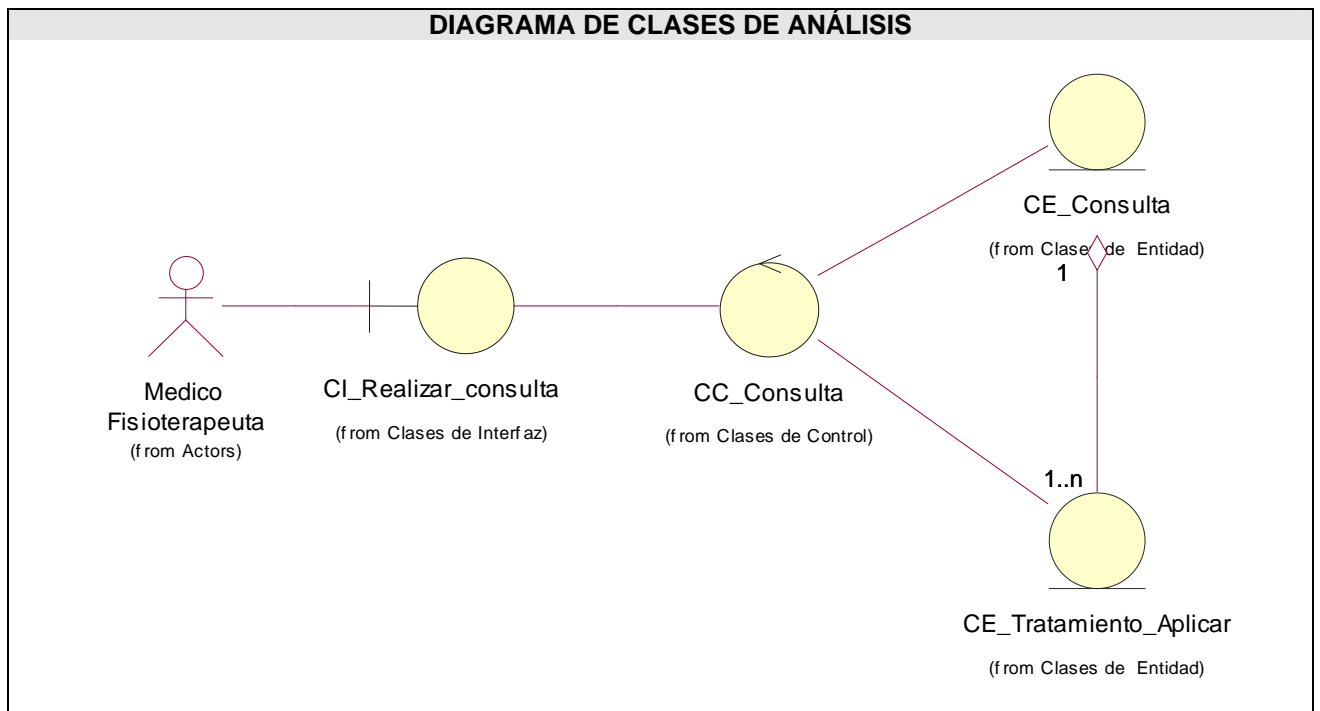


Diagrama de Clases “Realizar Consulta”.



4.3 Diseño.

En este flujo de trabajo se parte del modelo de clases de análisis conseguido en el flujo anterior, así como del análisis hecho a los casos de uso y de la agrupación en paquetes de análisis. A partir de estos productos se realiza el diseño de la arquitectura del sistema, hacer un diseño de los casos de uso, conseguir un modelo de clases de diseño y realizar una división del sistema en subsistemas. Pero además, hay que recoger y diseñar los aspectos de navegación y de interfaz de usuario.

4.3.1 Diagramas de Interacción.

Un diagrama de interacción es el nombre colectivo que se le da a los diagramas de secuencia y los diagramas de colaboración. Ambos diagramas son isomorfos, es decir, se puede convertir de uno a otro sin pérdida de información. Un diagrama de secuencia es un diagrama de interacción que resalta la ordenación temporal de los mensajes. Un diagrama de secuencia presenta un conjunto de objetos y

los mensajes enviados y recibidos por ellos., Los objetos suelen ser instancias con nombre o anónimas de clases, pero también pueden representarse instancias de otros elementos, tales como colaboraciones, componentes y nodos. Un diagrama de colaboración es un diagrama de interacción que resalta la organización estructural de los objetos que envían y reciben mensajes. Un diagrama de colaboración muestra un conjunto de objetos, enlaces entre estos objetos y mensajes enviados y recibidos por estos objetos. Los objetos normalmente son instancias con nombre o anónimas de clases, pero también pueden representar instancias de otros elementos, como colaboraciones, componentes y nodos. (Ver Anexo 3).

4.3.2 Diagrama de clases del diseño.

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia, todo el código que irá creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente. En el caso de las aplicaciones Web, el diagrama de clases representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase, es muy importante pues estos son los artefactos que se necesitan modelar para que el desarrollador los implemente y obtener así el producto final con la calidad requerida.

Al tratar de utilizar el diagrama de clases tradicional para modelar aplicaciones Web surgen varios problemas, por lo cual los especialistas del Rational plantearon la creación de una extensión al modelo de análisis y diseño que permitiera representar el nivel de abstracción adecuado y la relación con los restantes artefactos de UML.

El diagrama de clases Web, fue definido, a partir de los diferentes casos de uso del sistema y empleando las extensiones de UML para Web, a continuación se muestran los diagramas de clases para los distintos paquetes.

Diagrama de Clase de Diseño “Autenticar Usuario”.

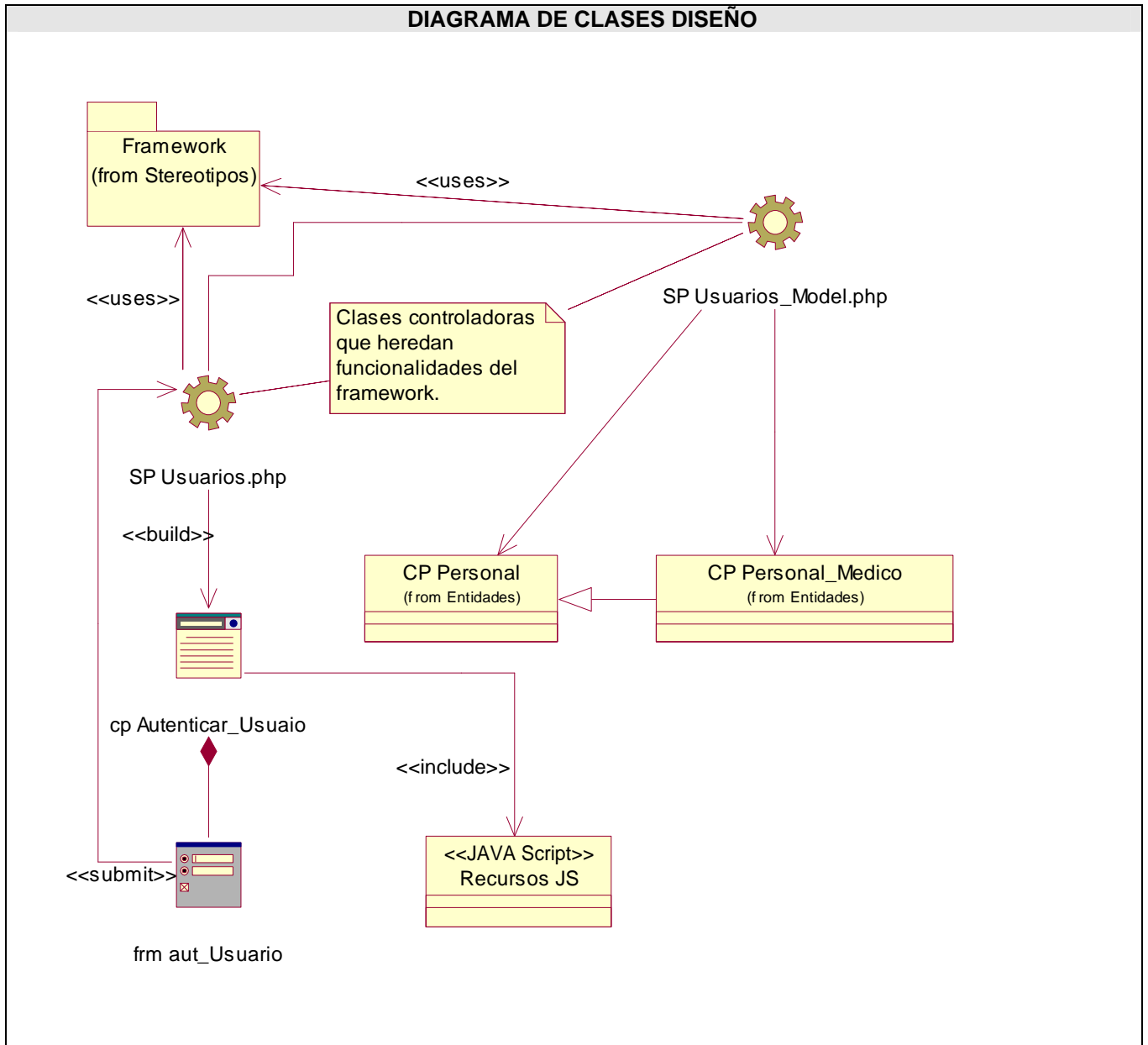


Diagrama de Clases de Diseño “Asistencia al Tratamiento”.

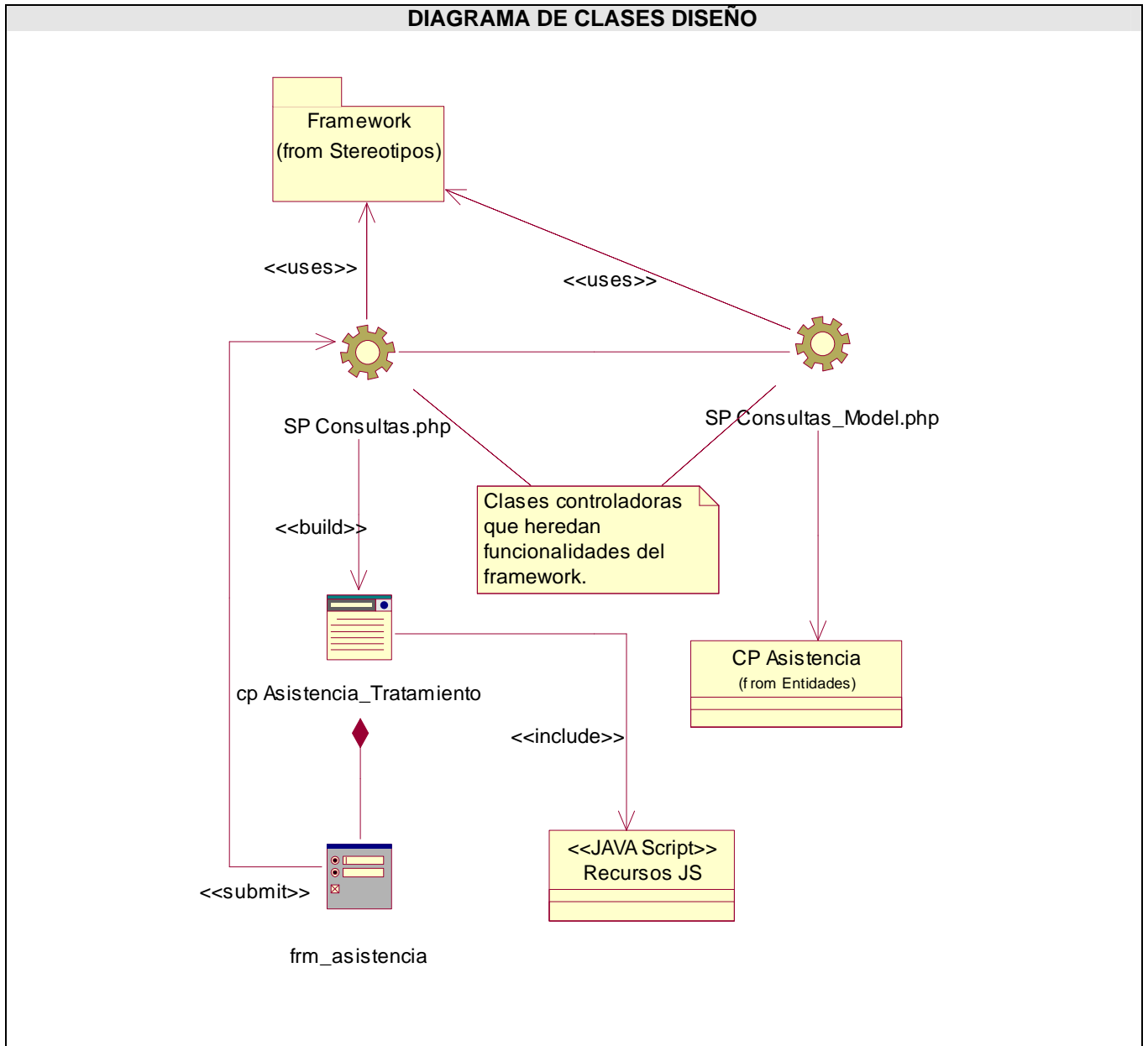


Diagrama de Clases de Diseño “Buscar Pacientes”.

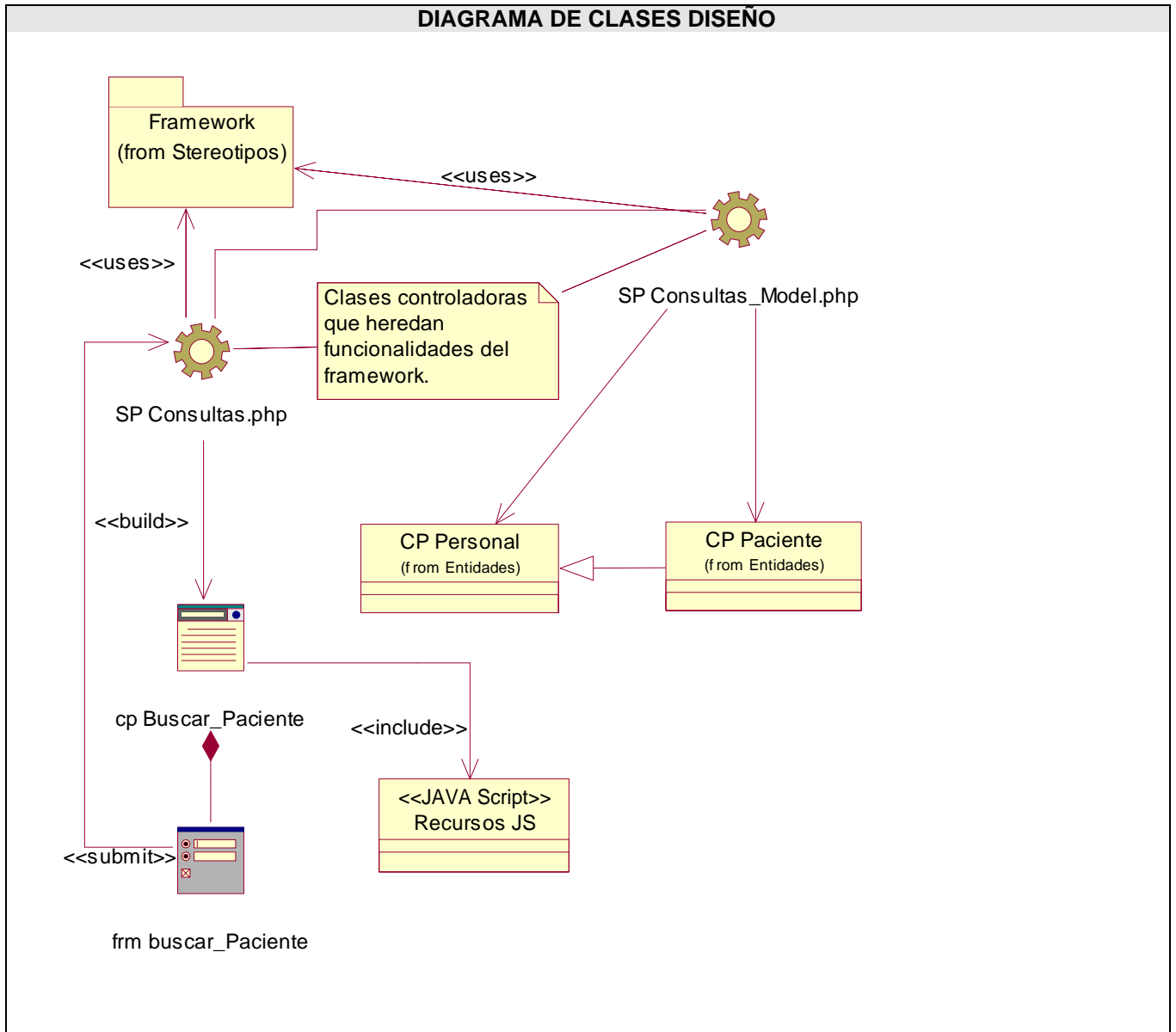


Diagrama de Clases de Diseño “Generar Reportes”.

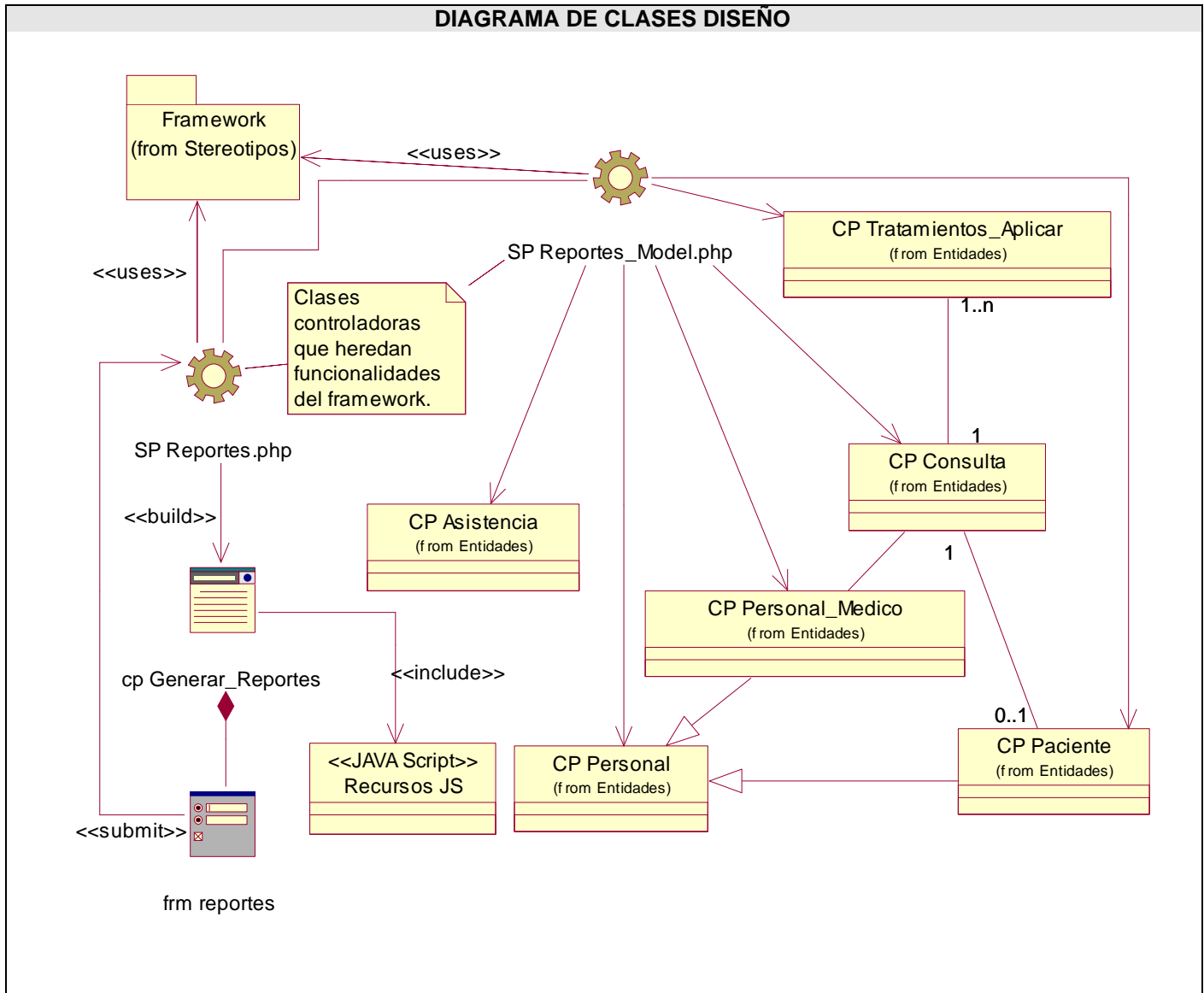


Diagrama de Clases de Diseño “Gestionar Pacientes”.

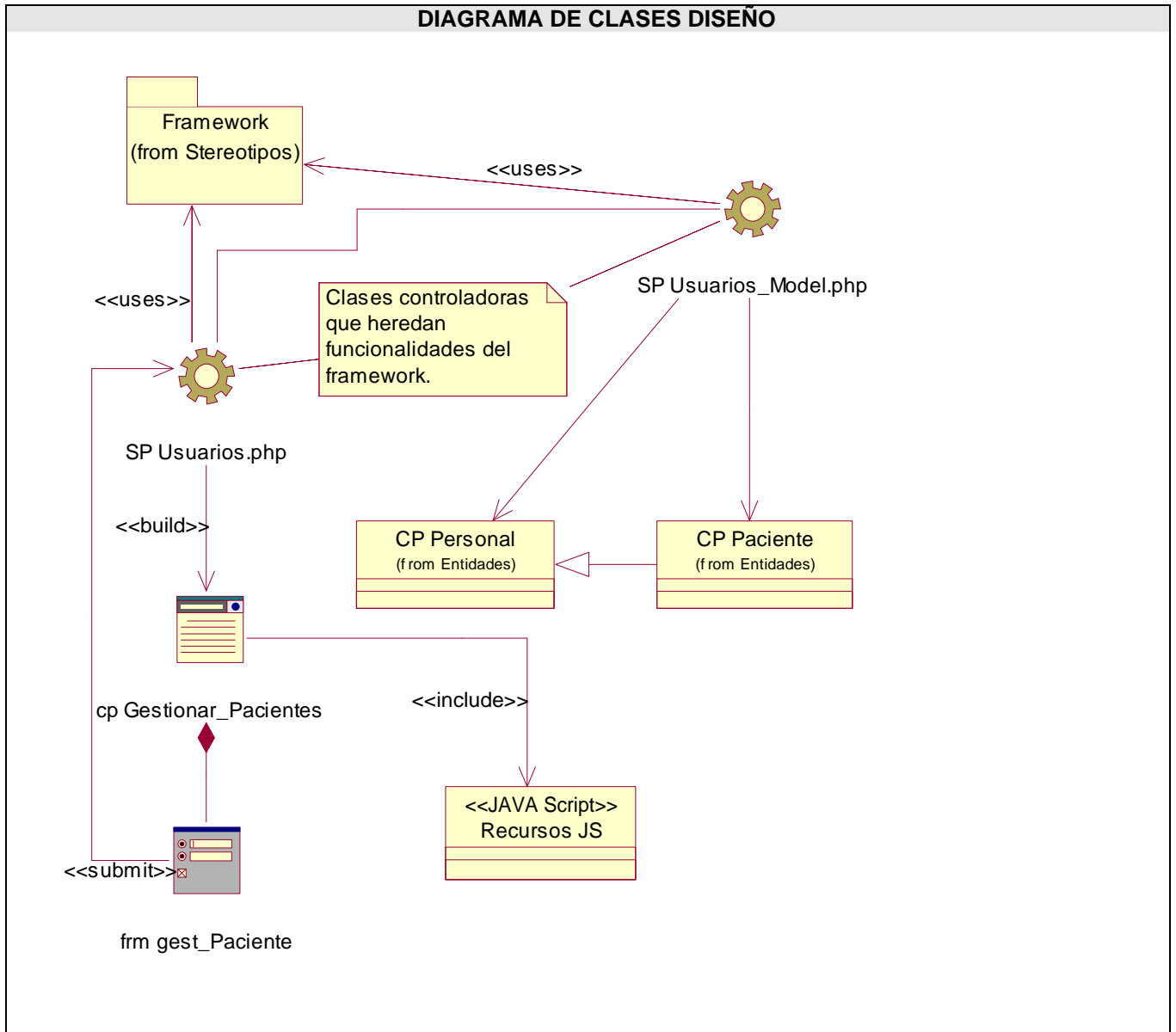


Diagrama de Clases de Diseño “Gestionar Personal Médico”.

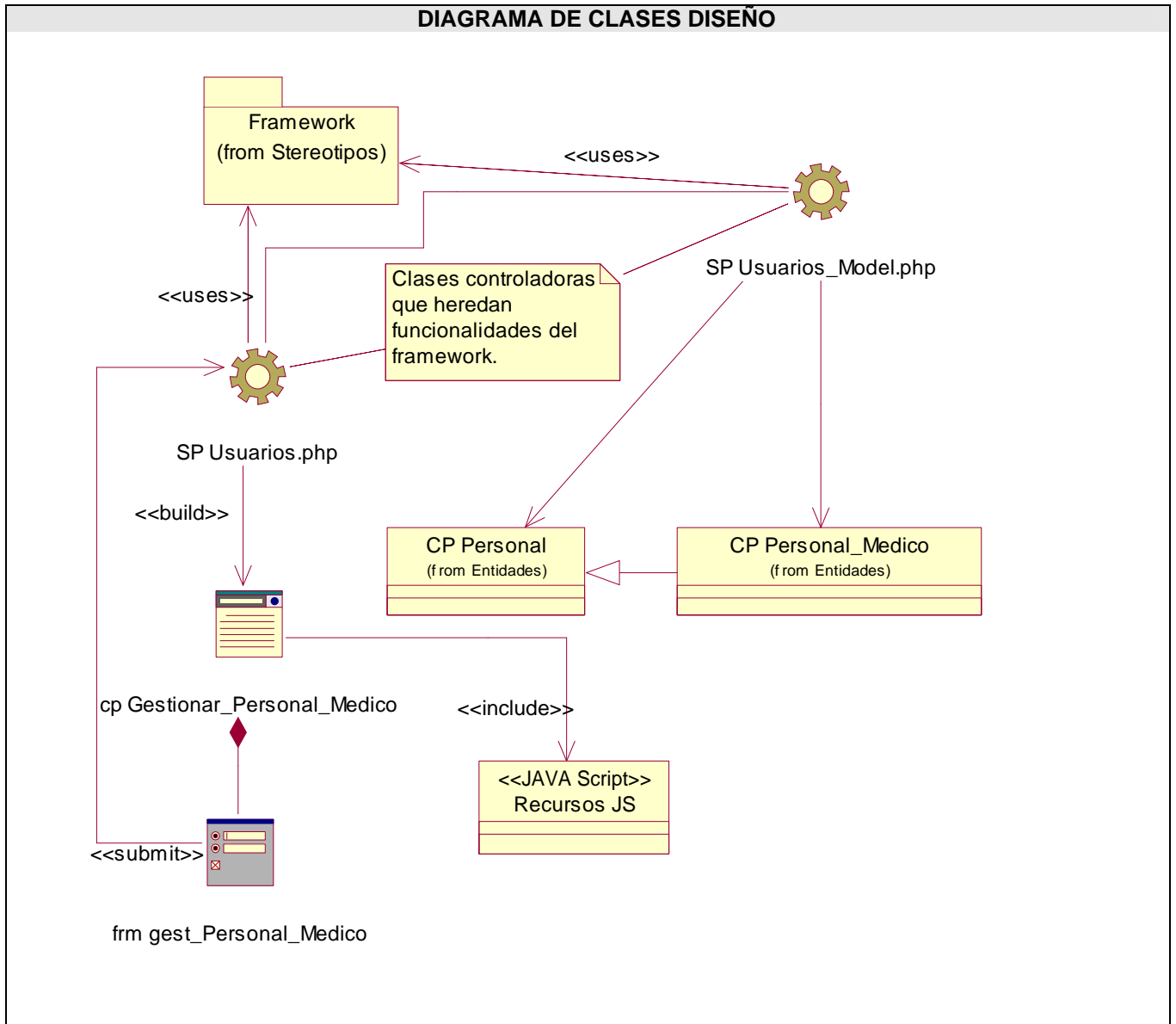


Diagrama de Clases de Diseño “Gestionar Usuarios”.

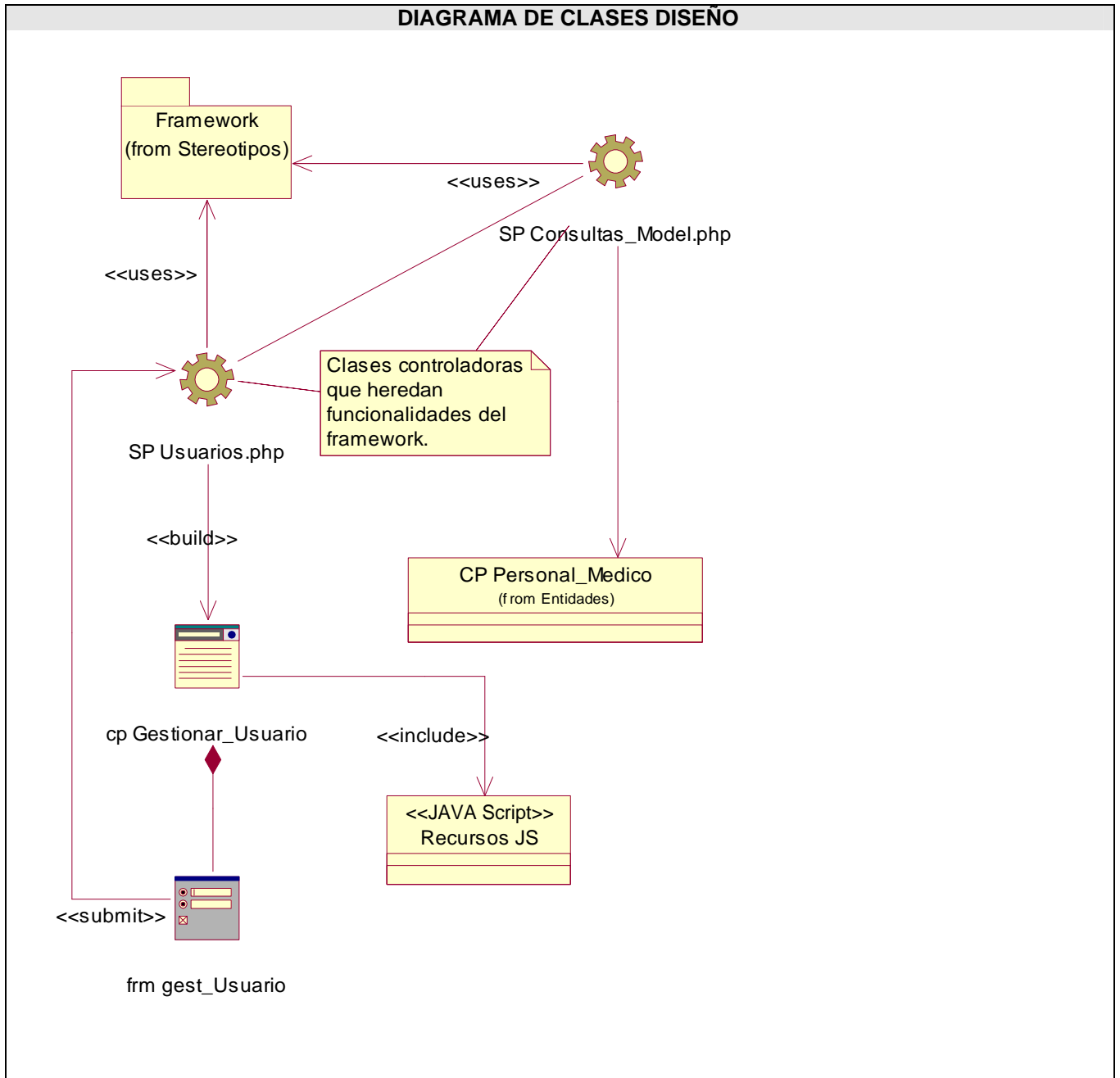


Diagrama de Clases de Diseño “Realizar consulta”.

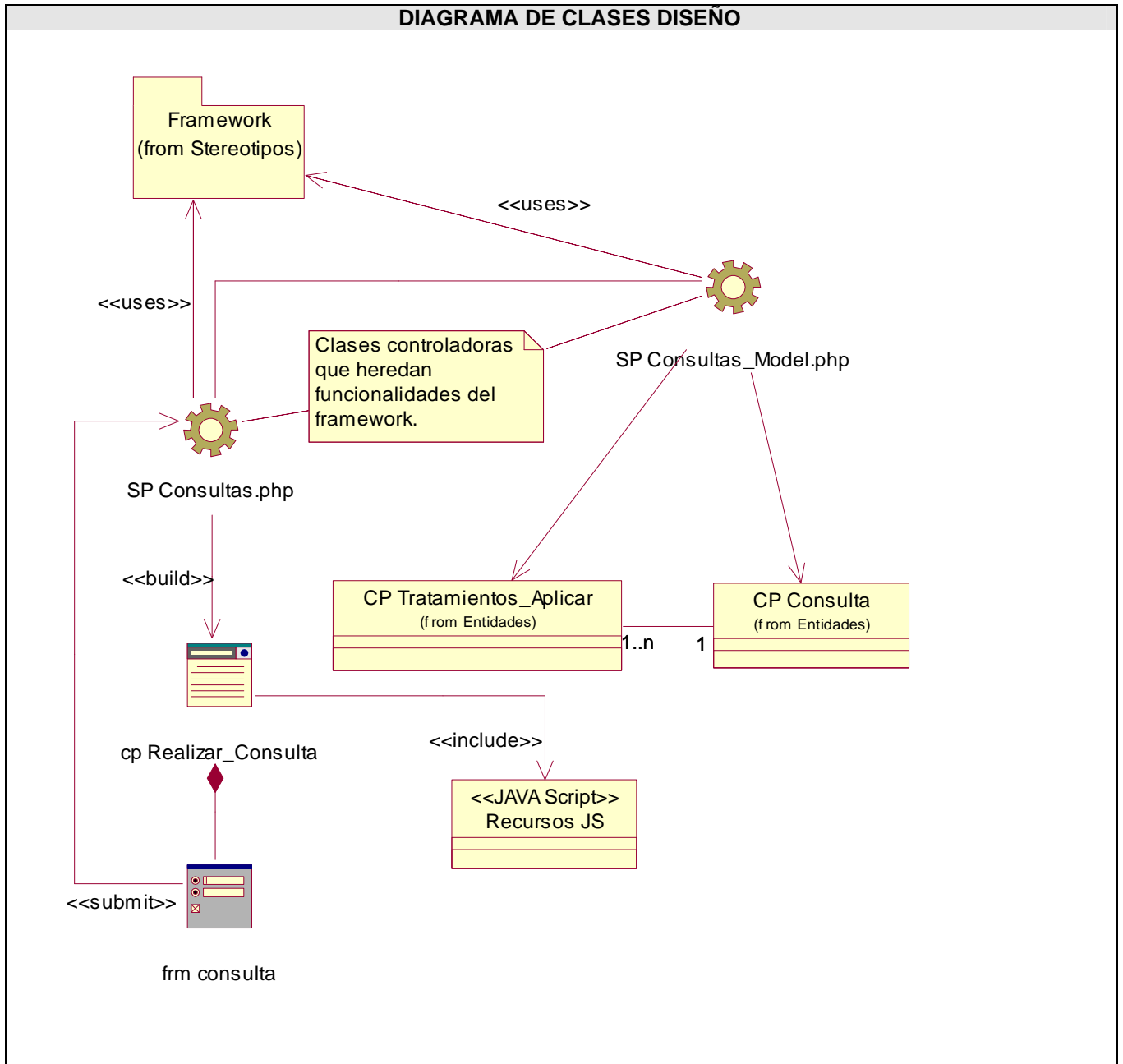


Diagrama de Clases de Diseño “Gestionar Codificadores”.

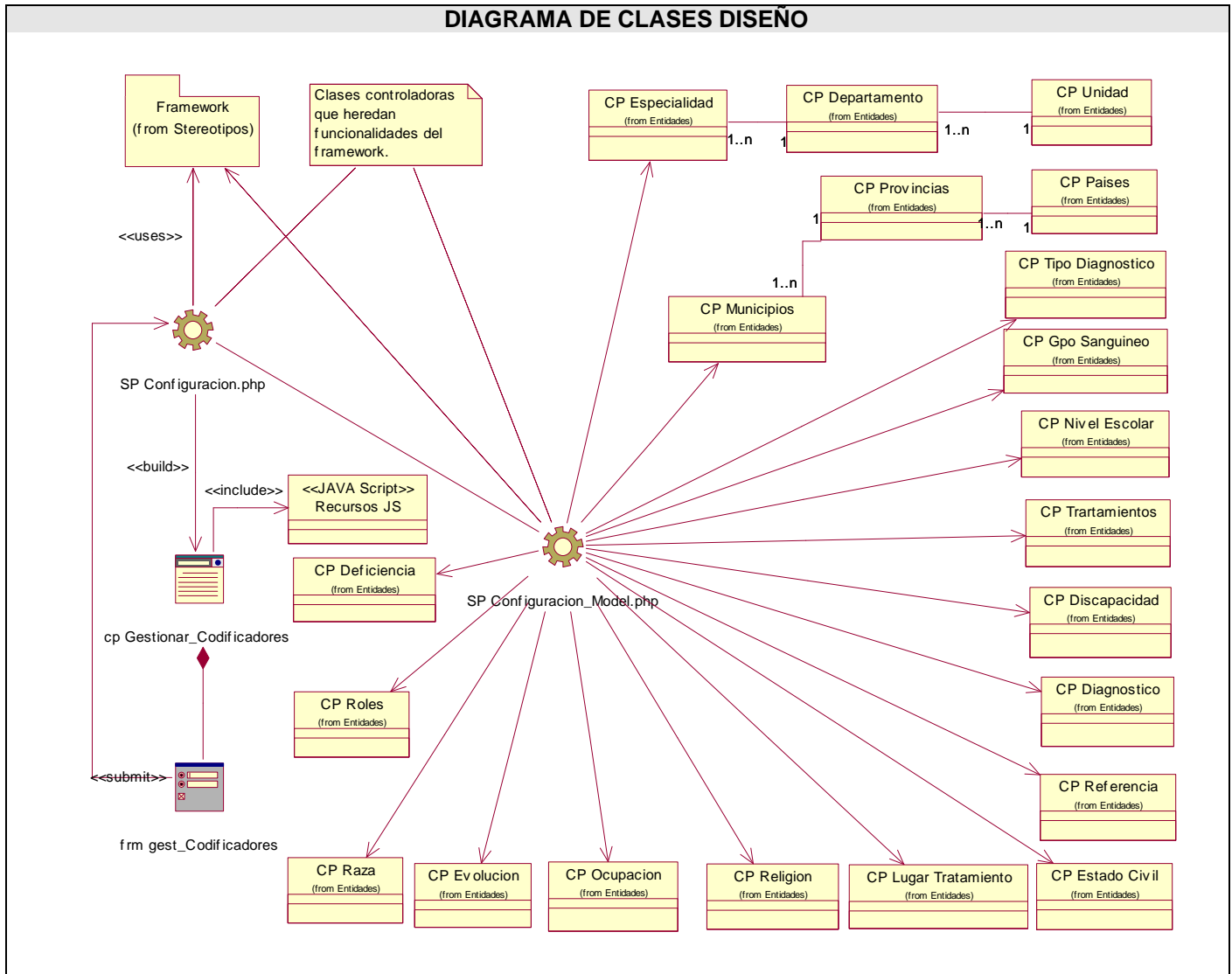


Diagrama de Clases de Diseño “Planificar Consulta”.

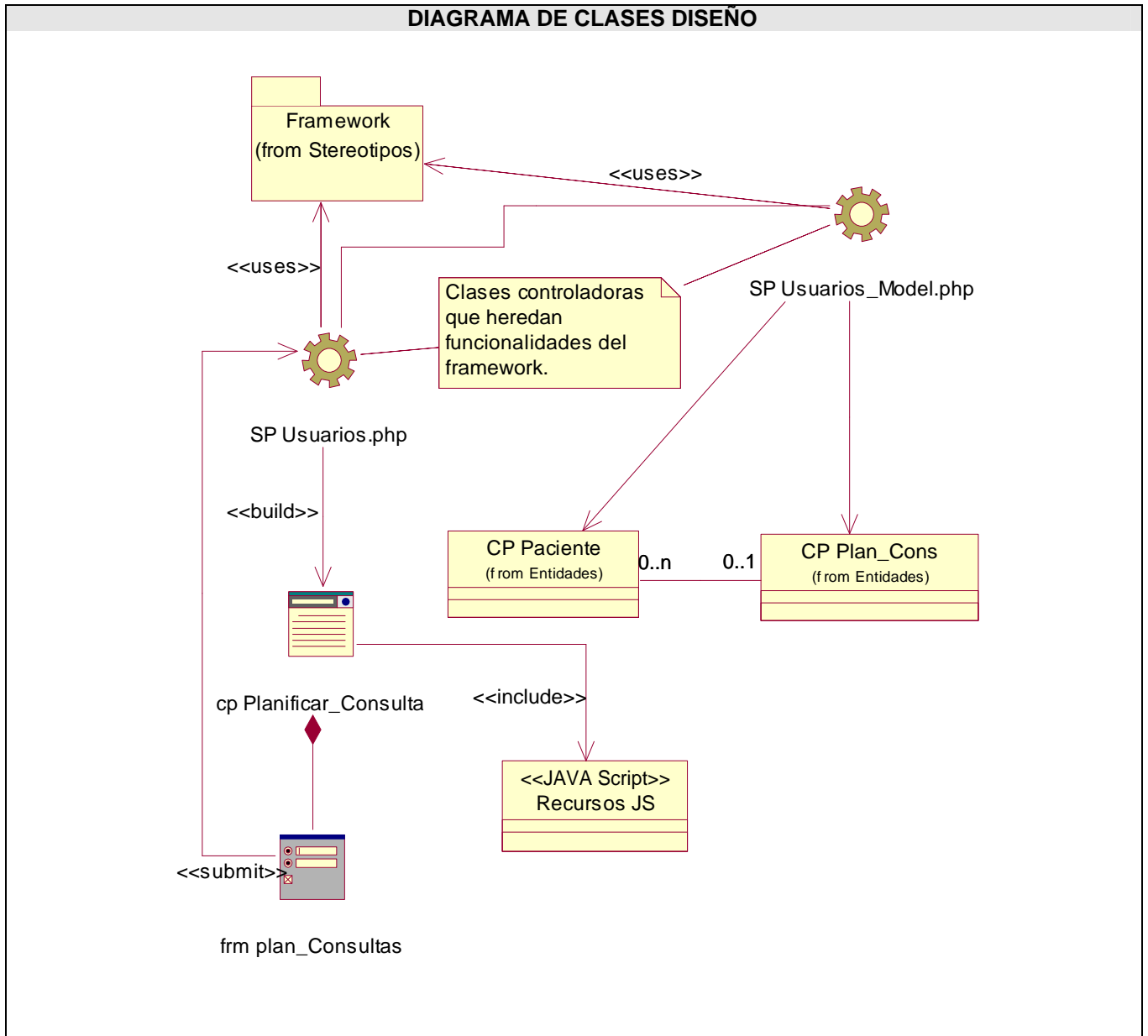
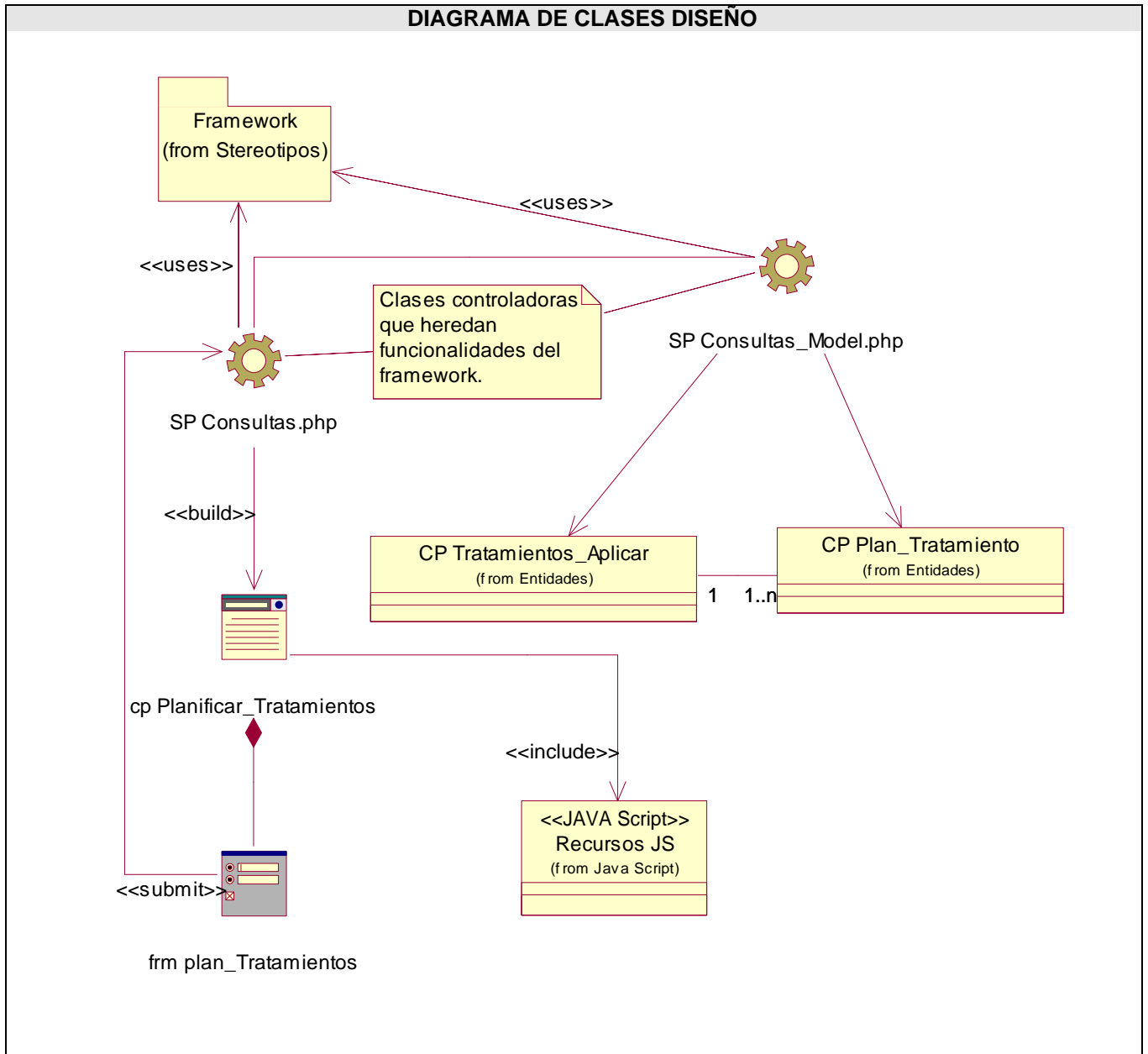


Diagrama de Clases de Diseño “Planificar Tratamientos”.



4.3.4 Descripción de las Clases.

Nombre: Usuarios	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	Login(\$user, \$pass)
Descripción:	Es la encargada de comprobar si el usuario y contraseñas suministradas son las correctas.
Nombre:	Regist_Personal_Medico(\$array_datos)
Descripción:	Para registrar al personal médico en el sistema.
Nombre:	Regist_paciente(\$array_datos)
Descripción:	Para registrar los pacientes en el sistema.
Nombre:	Plan_Consulta(\$array_datos)
Descripción:	Para planificarle las consultas a los pacientes.
Nombre:	Listar()
Descripción:	Devuelve el nombre de usuario y el rol existente en la BD.
Nombre:	Llena combos()
Descripción:	Encargada de llenar los combos necesarios para registrar alguna persona en la aplicación.
Nombre:	IsUser(\$user)
Descripción:	Comprueba que el usuario exista en la BD.
Nombre:	IsPass(\$user, \$pass)
Descripción:	Comprueba que la contraseña exista para el usuario que se entró.
Nombre:	Tiene_acceso()
Descripción:	Selecciona el tipo de usuario (rol) que tiene en la BD.
Nombre:	Salir(\$_SESSION['USER'])
Descripción:	Para cerrar la sesión del usuario que estaba conectado.

Nombre: Configuración	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	list_tablas()
Descripción:	Se utiliza que me lista todas las tablas de la BD.
Nombre:	listado()
Descripción:	Se utiliza para mostrar todos los campos que tenga una tabla determinada, se le pasa el nombre de una tabla y esta me devuelve un arreglo de los campos que esta tenga, con las opciones de actualizar y eliminar.
Nombre:	Insert(\$tabla, \$arra_datos)
Descripción:	Inserta datos en una tabla.
Nombre:	Delete(\$tabla, \$arra_datos)
Descripción:	Elimina los datos de una tabla.
Nombre:	Update(\$tabla, \$arra_datos)
Descripción:	Actualiza los datos de una tabla.

Nombre: Consultas	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	asistencia()
Descripción:	Se utiliza a la hora de darle los tratamientos a los pacientes, se registran datos como el día, hora, mejorías etc.
Nombre:	planificar()
Descripción:	Es utilizada para planificarles las fechas de los tratamientos a los pacientes, es decir el día y hora en que este debe venir atenderse.
Nombre:	buscar_paciente()
Descripción:	Para hacer búsqueda de pacientes según el criterio suministrado por el usuario.
Nombre:	show()
Descripción:	Muestra un calendario con todos los tratamientos que ya se encuentran planificados en la BD.
Nombre:	list_diag()
Descripción:	Lista los diagnósticos que se encuentran en la BD
Nombre:	list_trat()
Descripción:	Lista los tratamientos que se encuentran en la BD

Nombre:	list_def()
Descripción:	Lista las deficiencias que se encuentran en la BD
Nombre:	list_evol()
Descripción:	Lista las posibles evoluciones que se encuentran en la BD.
Nombre:	list_doc()
Descripción:	Lista los médicos que se encuentran en la BD
Nombre:	Get_evente_day()
Descripción:	Devuelve un arreglo con los tratamientos planificados para un mes específico.
Nombre:	eventos_día()
Descripción:	Devuelve un arreglo con los días y la información relacionada.
Nombre:	Chechevent()
Descripción:	Chequea dado un día los tratamientos que se deben aplicar.
Nombre:	Checkeventos()
Descripción:	Chequean los pacientes que deben asistir dado un día y el tratamiento a aplicar.

Nombre: Autenticar_Usuaio	
Tipo de clase: Client Page	
Caso de Uso: Autenticar Usuario	
Descripción General: Permite el acceso de los usuarios al sistema, comprueba si el usuario que está solicitando el acceso se encuentra registrado en la Base de Datos y si su cuenta de usuario está habilitada, este entra al sistema con los privilegios que tiene según el tipo de usuario, guardándose algunos datos del mismo. Si no se cumple alguna de estas condiciones se reporta un error de acceso.	
PARÁMETROS DE ENTRADA	
Descripción	Tipo
login	varchar
password	varchar

Nombre: Asistencia_Tratamiento	
Tipo de clase: Client Page	
Caso de Uso: Asistencia Tratamiento	
Descripción General: Permite verificar la asistencia del paciente a los tratamientos, registrando una serie de datos del mismo del día que se realizó el tratamiento.	

PARÁMETROS DE ENTRADA	
Descripción	Tipo
# HC	Integer
Tratamiento	Varchar
Fecha	DateTime
Asistencia	bool
Evolucion	varchar
Observaciones	varchar

Nombre: Buscar_Paciente	
Tipo de clase: Client Page	
Caso de Uso: Buscar Paciente	
Descripción General: permite al usuario buscar alguna información referente al los usuarios, comprueba si los datos referentes a la búsqueda se encuentran en la base de datos, si es así los muestra, de lo contrario reporta un error.	
PARÁMETROS DE ENTRADA	
Descripción	Tipo
Nombre	varchar
Carnet Identidad	Integer
# de Historia Clinica	Integer

Nombre: Realizar_Consulta	
Tipo de clase: Client Page	
Caso de Uso: Realizar Consulta	
Descripción General: permite al usuario realizar una consulta, registrando una serie de datos del paciente, y especificando los tratamientos que el mismo se realizara.	
PARÁMETROS DE ENTRADA	
Descripción	Tipo
Historia Clínica	Integer
Discapacidad	varchar
Lugar de Tratamiento	varchar
Doctor Atiende	varchar
Motivos Consulta	varchar
Deficiencias	varchar

Diagnóstico	varchar
Tipo Diagnóstico	varchar
Micro Historia	varchar
Tratamientos	varchar
Cantidad de Sesiones	Integer

Nombre: Planificar_Tratamientos	
Tipo de clase: Client Page	
Caso de Uso: Planificar Tratamientos	
Descripción General: permite al usuario Planificarle los Tratamientos a los pacientes, registrando una serie de datos del paciente, además de dar la opción de ver cuales son las fechas que están disponibles para que usuario pueda saber cuales son las fechas en que puede planificarle el tratamiento al paciente.	
PARÁMETROS DE ENTRADA	
Descripción	Tipo
Historia Clínica	Integer
Tratamiento	varchar
Sesiones	Integer
Hora	Time
Fechas Disponibles	Date

Nombre: Planificar_Consulta	
Tipo de clase: Client Page	
Caso de Uso: Planificar Consulta	
Descripción General: permite al usuario Planificar una consulta al paciente, registrando una serie de datos del mismo, especificando varios campos que deberán ser obligatorios para la realización de la consulta. (*)	
PARÁMETROS DE ENTRADA	
Descripción	Tipo
Nombre	varchar
2do Nombre	varchar
1er Apellido	varchar
2do Apellido	varchar

Fecha Nacimiento	Date
Carnet Identidad	Integer
Fecha de Consulta	Date
Dirección Particular	varchar
Especialidad Remite	varchar
Foto	imagen
Otra HC	varchar
Sexo	bool
Edad	Integer
Médico	Varchar

Nombre: Gestionar_Codificadores
Tipo de clase: Client Page
Caso de Uso: Gestionar_Codificadores
Descripción General: Permite al usuario escoger un tipo de codificador, listando todos records del mismo, dando la posibilidad de gestionar los mismos, dígame gestionar las acciones de adicionar, modificar y eliminar.

Nombre: Personal	
Tipo de clase: entidad	
Atributo	Tipo
Id_personal	Integer
Nombre1	Varchar
Nombre2	Varchar
Appellido1	Varchar
Appellido2	Varchar
Carnet_Identidad	Integer
Fecha_Nac	Date
Edad	Integer
Sexo	Varchar
Direccion_particular	Varchar
Fecha_nacimiento	Date
Fecha_ingreso	Date
Registrado_por	Varchar
Id_raza	Integer
Id_gpo_sang	Integer
Id_ocip	Integer
Id_escolar	Integer
Id_civil	Integer
Id_religion	Integer

Nombre: Personal_médico	
Tipo de clase: entidad.	
Atributo	Tipo
Id_pmed	Integer
User_name	Varchar
Password	Varchar
Ano_experiencia	Integer
Id_espec	Integer
Id_rol	Integer
Id_personal	Integer

Nombre: Pacientes	
Tipo de clase (interfaz, controladora, entidad--)	
Atributo	Tipo
Hc	Integer
Hc_old	Varchar
Id_plan_cons	Integer
Id_egreso	Integer
Id_refer	Integer
Id_personal	Integer

Nombre: Consulta	
Tipo de clase: entidad.	
Atributo	Tipo
Id_consulta	Integer
microhistoria	Varchar
Motico_consulta	Varchar
Num_secc	Integer
Id_pmed	Integer
Hc	Integer
Id_diag	Integer
Id_t_diag	Integer
Id_disc	Integer
Id_ltto	Integer
Id_def	Integer

Nombre: Asistencia	
Tipo de clase: entidad.	
Atributo	Tipo
Id_fech_trat	Integer

Observaciones	Varchar
Id_plan_trat	Integer

Nombre: Deficiencia	
Tipo de clase: entidad.	
Atributo	Tipo
Id_def	Integer
N_def	Varchar

Nombre: Departamento	
Tipo de clase: entidad.	
Atributo	Tipo
Id_dpto	Integer
N_dpto	Varchar
Id_unidad	Integer

Nombre: Diagnóstico	
Tipo de clase: entidad.	
Atributo	Tipo
Id_diag	Integer
N_diag	Varchar

Nombre: Discapacidad	
Tipo de clase: entidad.	
Atributo	Tipo
Id_disc	Integer
N_disc	Varchar

Nombre: Egreso	
Tipo de clase: entidad.	
Atributo	Tipo
Id_egreso	Integer
Fecha	Date
Rehabilitado	Varchar

Nombre: Especialidad	
Tipo de clase: entidad.	
Atributo	Tipo
Id_espec	Integer
N_espec	Varchar
Id_dpto	Integer

Nombre: Estado Civil	
Tipo de clase: entidad.	
Atributo	Tipo
Id_civil	Integer
N_civil	Varchar

Nombre: Evolución	
Tipo de clase: entidad.	
Atributo	Tipo
Id_evol	Integer
N_evol	Varchar
Id_fech_trat	Integer

Nombre: Gpo Sanguíneo	
Tipo de clase: entidad.	
Atributo	Tipo
Id_gposang	Integer
N_gposang	Varchar

Nombre: Lugar Tratamiento	
Tipo de clase: entidad.	
Atributo	Tipo
Id_ltto	Integer
N_ltto	Varchar

Nombre: Municipios	
Tipo de clase: entidad.	
Atributo	Tipo
Id_mun	Integer
N_mun	Varchar
Id_prov	Integer

Nombre: Nivel Escolar	
Tipo de clase: entidad.	
Atributo	Tipo
Id_escolar	Integer
N_escolar	Varchar

Nombre: Ocupación	
Tipo de clase: entidad.	
Atributo	Tipo
Id_ocup	Integer
N_ocup	Varchar

Nombre: Paises	
Tipo de clase: entidad.	
Atributo	Tipo
Id_pais	Integer
N_pais	Varchar

Nombre: Plan_Cons	
Tipo de clase: entidad.	
Atributo	Tipo
Id_plan_cons	Integer
medico	Varchar
Horario	Date

Nombre: Plan Tratamiento	
Tipo de clase: entidad.	
Atributo	Tipo
Id_plan_trat	Integer
Fecha	Date
Hora	Date
Id_tto_apl	Integer

Nombre: Provincias	
Tipo de clase: entidad.	
Atributo	Tipo
Id_prov	Integer
N_prov	Varchar
Id_pais	Integer

Nombre: Raza.	
Tipo de clase: entidad.	
Atributo	Tipo
Id_raza	Integer
Name_raza	Varchar

Nombre: Referencia	
Tipo de clase: entidad.	
Atributo	Tipo
Id_refer	Integer
N_refer	Varchar

Nombre: Religión	
Tipo de clase: entidad.	
Atributo	Tipo
Id_religion	Integer
N_religion	Varchar

Nombre: Roles	
Tipo de clase: entidad.	
Atributo	Tipo
Id_rol	Integer
N_rol	Varchar

Nombre: Tipo Diagnóstico	
Tipo de clase: entidad.	
Atributo	Tipo
Id_t_diag	Integer
N_t_diag	Varchar

Nombre: Tratamientos Aplicar	
Tipo de clase: entidad.	
Atributo	Tipo
Id_tto_apl	Integer
Num_secc	Integer
Hora	Date
Id_trat	Integer

Nombre: Tratamientos	
Tipo de clase: entidad.	
Atributo	Tipo
Id_trat	Integer
N_trat	Varchar
Duracion	Date

Nombre: Unidad	
Tipo de clase: entidad.	
Atributo	Tipo
Id_unidadt	Integer
N_unidad	Varchar

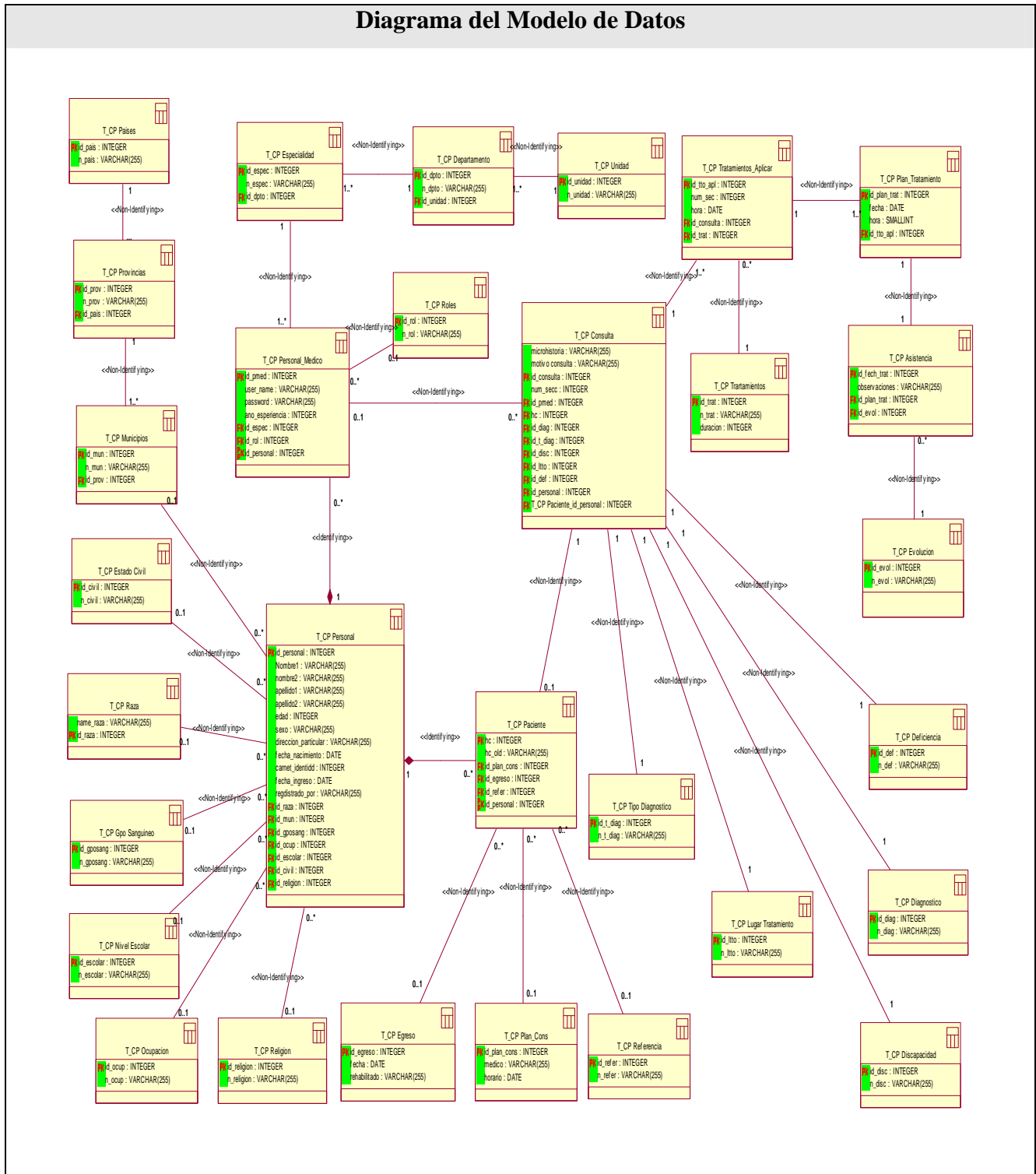
4.4 Diseño de la base de datos.

La base de datos es el sistema utilizado para el almacenamiento de datos y acceso controlado a los datos almacenados. En este epígrafe se muestra el diseño de la base de datos del sistema propuesto a través del diagrama de clases persistentes y el esquema de la base de datos generados a partir de este, el modelo de datos.

4.4.1 Modelo de datos.

El modelo de los datos describe la representación lógica y física de datos persistentes en el sistema. A continuación se muestra el modelo de datos.

Diagrama del Modelo de Datos



4.4.2 Descripción de las Tablas.

Nombre: Personal		
Descripción: Recoge los datos generales de cualquier persona que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_personal	Integer	Identificador para los datos de esta tabla.
Nombre1	Varchar	Responde al nombre.
Nombre2	Varchar	Responde al segundo nombre.
Apellido1	Varchar	Responde al primer apellido.
Apellido2	Varchar	Responde al segundo apellido.
Carnet_Identidad	Integer	Numero del carné de identidad.
Fecha_Nac	Date	Fecha de nacimiento de la persona.
Edad	Integer	Edad de la persona.
Sexo	Varchar	Sexo de la persona.
Direccion_particular	Varchar	Dirección particular de la persona.
Fecha_ingreso	Date	Fecha de ingreso a la BD.
Registrado_por	Varchar	Persona que registro el record en la BD.
Id_raza	Integer	Identificador de la tabla raza.
Id_gpo_sang	Integer	Identificador de la tabla grupo sanguíneo.
Id_ocup	Integer	Identificador de la tabla ocupación.
Id_escolar	Integer	Identificador de la tabla nivel escolar.
Id_civil	Integer	Identificador de la tabla estado civil.
Id_religion	Integer	Identificador de la tabla religión.

Nombre: Personal Médico		
Descripción: Recoge los datos generales de cualquier persona que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_pmed	Integer	Identificador para los datos de esta tabla.
User_name	Varchar	Nombre de usuario para interactuar con la aplicación(es único).
Password	Varchar	Contraseña para el usuario que interactúa con la aplicación.
Ano_experiencia	Integer	Responde a los años de experiencia del médico.
Id_espec	Integer	Identificador de la tabla especialidad.
Id_rol	Integer	Identificador de la tabla roles.
Id_personal	Integer	Identificador de la tabla personal.

Nombre: Pacientes		
Descripción: Recoge los datos específicos de los pacientes registrados en la aplicación.		
Atributo	Tipo	Descripción
Hc	Integer	Número de historia clínica del paciente en la sala de rehabilitación.
Hc_old	Varchar	Un número de historia clínica del paciente en otro centro de salud.
Id_plan_cons	Integer	Identificador de la tabla plan_cons.
Id_egreso	Integer	Identificador de la tabla egreso.
Id_refer	Integer	Identificador de la tabla referencia.
Id_personal	Integer	Identificador de la tabla personal.

Nombre: Consulta		
Descripción: Recoge los datos generales de cualquier persona que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_consulta	Integer	Identificador para los datos de esta tabla.
microhistoria	Varchar	Responde a una microhistoria del paciente.
Motico_consulta	Varchar	Responde a los motivos de consulta.
Num_secc	Integer	Número de secciones de un tratamiento.
Id_pmed	Integer	Identificador de la tabla persona_médico.
Hc	Integer	Identificador de la tabla pacientes).
Id_diag	Integer	Identificador de la tabla diagnósticos.
Id_t_diag	Integer	Identificador de la tabla tipo de diagnósticos.
Id_disc	Integer	Identificador de la tabla discapacidades.
Id_ltto	Integer	Identificador de la tabla lugar del tratamiento.
Id_def	Integer	Identificador de la tabla deficiencias.

Nombre: Asistencia		
Descripción: Recoge los datos generados a la hora que el técnico le aplica los tratamientos al paciente.		
Atributo	Tipo	Descripción
Id_fech_trat	Integer	Identificador para los datos de esta tabla.
Observaciones	Varchar	Observaciones hechas por el técnico.
Id_plan_trat	Integer	Identificador de la tabla planificación de tratamiento.
Id_evol	Integer	Identificador de la tabla evolución.

Nombre: Deficiencia		
Descripción: Recoge las diferentes deficiencias que pueden aparecer.		
Atributo	Tipo	Descripción
Id_def	Integer	Identificador para los datos de esta tabla.
N_def	Varchar	Describe el nombre de la deficiencia

Nombre: Departamento		
Descripción: Recoge los datos generales del departamento.		
Atributo	Tipo	Descripción
Id_dpto	Integer	Identificador para los datos de esta tabla.
N_dpto	Varchar	Describe el nombre del departamento.
Id_unidad	Integer	Identificador de la tabla unidad.

Nombre: Diagnostico		
Descripción: Se recogen los posibles diagnósticos que se puedan hacer.		
Atributo	Tipo	Descripción
Id_diag	Integer	Identificador para los datos de esta tabla.
N_diag	Varchar	Describe el nombre del diagnóstico.

Nombre: Discapacidad		
Descripción: Se recoge las diferentes discapacidades con las que un paciente puede llegar.		
Atributo	Tipo	Descripción
Id_disc	Integer	Identificador para los datos de esta tabla.
N_disc	Varchar	Describe el nombre de la discapacidad.

Nombre: Egreso		
Descripción: Recoge los datos generales del egreso del paciente.		
Atributo	Tipo	Descripción
Id_egreso	Integer	Identificador para los datos de esta tabla.
Fecha	Date	Fecha de egreso
Rehabilitado	Varchar	Motivos de egreso...

Nombre: Especialidad		
Descripción: Recoge los datos referente a las especialidades.		
Atributo	Tipo	Descripción
Id_espec	Integer	Identificador para los datos de esta tabla.
N_espec	Varchar	Descripción de la especialidad.
Id_dpto	Integer	Identificador de la tabla departamento.

Nombre: Estado Civil		
Descripción: Recoge los datos generales de cualquier persona que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_civil	Integer	Identificador para los datos de esta tabla.
N_civil	Varchar	

Nombre: Evolución		
Descripción: Recoge los datos de la evolución del paciente.		
Atributo	Tipo	Descripción
Id_evol	Integer	Identificador para los datos de esta tabla.
N_evol	Varchar	Descripción de la evolución del paciente.

Nombre: Gpo Sanguineo		
Descripción: Recoge datos del grupo sanguíneo de las personas que se registren en la aplicación.		
Atributo	Tipo	Descripción
Id_gposang	Integer	Identificador para los datos de esta tabla.
N_gposang	Varchar	Descripción del grupo sanguíneo.

Nombre: Lugar Tratamiento		
Descripción: Recoge los datos del lugar donde se aplicará el tratamiento.		
Atributo	Tipo	Descripción
Id_ltto	Integer	Identificador para los datos de esta tabla.
N_ltto	Varchar	Descripción del lugar de tratamiento.

Nombre: Municipios		
Descripción: Recoge los datos de cualquier municipio que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_mun	Integer	Identificador para los datos de esta tabla.
N_mun	Varchar	Nombre del municipio.
Id_prov	Integer	Identificador de la tabla provincias.

Nombre: Nivel Escolar		
Descripción: Recoge los datos del nivel escolar de la persona que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_escolar	Integer	Identificador para los datos de esta tabla.
N_escolar	Varchar	Descripción del nivel escolar.

Nombre: Ocupacion		
Descripción: Recoge los datos de ocupación de la persona.		
Atributo	Tipo	Descripción
Id_ocup	Integer	Identificador para los datos de esta tabla.
N_ocup	Varchar	Dscripción de la ocupación.

Nombre: Países		
Descripción: Recoge los datos de cualquier país que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_pais	Integer	Identificador para los datos de esta tabla.
N_pais	Varchar	Nombre de los países.

Nombre: Plan_Cons		
Descripción: Recoge los datos de la planificación de las consultas.		
Atributo	Tipo	Descripción
Id_plan_cons	Integer	Identificador para los datos de esta tabla.
medico	Varchar	El medico que lo va atender en la consulta.
Horario	Date	Fecha en la cual deberá asistir a la consulta.

Nombre: Plan Tratamiento		
Descripción: Recoge los datos de la planificación de tratamientos a los pacientes.		
Atributo	Tipo	Descripción
Id_plan_trat	Integer	Identificador para los datos de esta tabla.
Fecha	Date	Recoge las fechas de tratamientos.
Hora	Date	Recoge la hora de los tratamientos.
Id_tto_apl	Integer	Identificador de la tabla tratamiento aplicar.

Nombre: Provincias		
Descripción: Recoge los datos de cualquier provincia que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_prov	Integer	Identificador para los datos de esta tabla.
N_prov	Varchar	Nombre de la provincia.
Id_pais	Integer	Identificador de países.

Nombre: Raza		
Descripción: Recoge los datos generales de cualquier raza que se registre en la aplicación.		
Atributo	Tipo	Descripción
Id_raza	Integer	Identificador para los datos de esta tabla.
Name_raza	Varchar	Nombre de la raza.

Nombre: Referencia		
Descripción: Recoge los datos relacionados con las referencias que se encuentran en la aplicación.		
Atributo	Tipo	Descripción
Id_refer	Integer	Identificador para los datos de esta tabla.
N_refer	Varchar	Nombre de la referencia.

Nombre: Religion		
Descripción: Recoge los datos relacionados con las religiones en la aplicación.		
Atributo	Tipo	Descripción
Id_religion	Integer	Identificador para los datos de esta tabla.
N_religion	Varchar	Nombre de las religiones.

Nombre: Roles		
Descripción: Recoge los datos relacionados con los tipos de roles en la aplicación.		
Atributo	Tipo	Descripción
Id_rol	Integer	Identificador para los datos de esta tabla.
N_rol	Varchar	Nombre de los diferentes roles.

Nombre: Tipo Diagnostico		
Descripción: Recoge los datos de los diferentes diagnósticos introducidos en la aplicación.		
Atributo	Tipo	Descripción
Id_t_diag	Integer	Identificador para los datos de esta tabla.
N_t_diag	Varchar	Nombre de los diagnósticos.

Nombre: Tratamientos Aplicar		
Descripción: Recoge los datos generados en la consulta del paciente(es decir los tratamientos que el paciente deberá realizarse).		
Atributo	Tipo	Descripción
Id_tto_apl	Integer	Identificador para los datos de esta tabla.
Num_secc	Integer	Números de secciones de los tratamientos.
Hora	Date	Fecha en que aplicarán los tratamientos.
Id_trat	Integer	Identificador de la tabla tratamientos.

Nombre: Tratamientos		
Descripción: Recoge los nombres de los tratamientos que se realizan en el centro de rehabilitación.		
Atributo	Tipo	Descripción
Id_trat	Integer	Identificador para los datos de esta tabla.
N_trat	Varchar	Nombre de los tratamientos.
Duracion	Date	Tiempo que dura el tratamiento.

Nombre: Unidad		
Descripción: Recoge el nombre del centro de rehabilitación registrado en la aplicación.		
Atributo	Tipo	Descripción
Id_unidad	Integer	Identificador para los datos de esta tabla.
N_unidad	Varchar	Nombre del centro de rehabilitación.

4.5 Principios de diseño.

El diseño de la interfaz de una aplicación, el formato de los reportes, la concepción de la ayuda y el tratamiento de excepciones tiene gran influencia en el éxito o fracaso de una aplicación. A continuación se describen los principios de diseño seguidos para el desarrollo del sistema en cuestión.

4.6 Tratamiento de errores.

En el sistema propuesto se evitan, minimizan y tratan los posibles errores, con el fin de garantizar la integridad y confiabilidad de la información que en este se registra y muestra. Los errores se tratan en una página especial que incluye el fichero de configuración general, y está preparada para recoger el número del error y presentar la pantalla con el error que le corresponde a ese código.

Los mensajes de error que emite el sistema se muestran en un lenguaje de fácil comprensión para los usuarios. Cuando se introduce información en un formulario y faltan datos, sale un cuadro de alerta indicando el campo o dato que falta. Similar ocurre cuando se introduce información errónea en un campo numérico o e-mail.

4.7 Seguridad.

Para lograr garantizar la seguridad e integridad de la información que se encuentra en la base de datos del sistema, se crean usuarios que tienen determinados permisos para la gestión de esta información. Los niveles de acceso están dados por los tipos de usuarios o roles definido según las características de los usuarios que interactúan con el sistema.

Esta es la forma en que se garantiza la seguridad y protección de la información que se maneja a través del sistema mediante el acceso al mismo de diferentes niveles de usuario.

4.8 Interfaz de usuario.

Para el diseño de las interfaz de usuario se tuvieron en cuenta una serie de aspectos básicos que se pueden definir como: “el conjunto pasos que seguirá el usuario, durante todo el tiempo que se relacione con el programa, detallando lo que verá y escuchará en cada momento, y las acciones que realizará, así como las respuestas que el sistema dará”.Según la calidad de la interfaz un sistema se puede llegar a convertir en un éxito o un fracaso.

Para el diseño de la interfaz se tuvieron en cuenta una serie de aspectos como son: reducir la carga a la memoria, atajos a usuarios expertos, obtener información de retroalimentación, diseño de diálogos que conducen a una conclusión, previsión de errores y manejo de errores simples, lograr deshacer acciones fácilmente, que el usuario sintiera la sensación de control.

Las aplicaciones Web tienen como características fundamentales que proporcionan un ambiente amigable, conducen a un aprendizaje más natural, establecen un “sentimiento” (sobre todo en la uniformidad del ambiente) al usuario que enriquece su experiencia en el uso de la aplicación.

Además se tuvo en cuenta:

- Utilizar una misma tipografía, forma y estilo en todas las páginas.
- La facilidad del usuario de poder navegar desde cualquier punto a otro dentro de la aplicación.
- La simplicidad y consistencia, favoreciendo la usabilidad de la aplicación.

Navegación simple en todas las páginas de la aplicación, de forma tal que siempre sea accesible por el usuario.

Estabilidad y uniformidad del diseño, para así poder ubicar al usuario dentro del mismo y hacerlo sentir parte de él.

Se utilizó una hoja de estilos para guardar la configuración del diseño para todas las páginas, para los botones y las líneas se utilizaron estos estilos, eliminando así el número de imágenes que demoren la presentación de la página.

Los formularios de entradas ocupan el centro superior y las entradas organizadas por importancia. Se incluye una breve explicación del objetivo del formulario, y alguna especificación con respecto a las entradas.

Se realizan múltiples operaciones en cada página, de forma que el usuario no tenga que moverse tanto dentro de la aplicación, para completar una operación.

4.9 Formato de salida de los reportes.

Generar reportes que permitan un control de la información que fluye en las salas de Rehabilitación es una de las principales funcionalidades del sistema propuesto. Estos se obtienen, en dependencia de las necesidades del usuario.

Los informes, resumen los resultados en las salas de rehabilitación, se han concebido en ventanas diferentes a la aplicación, utilizando letra legible y colores claros, de fondo, para no recargar la página y lograr calidad y nitidez en la impresión de la información.

Cada reporte e informe tiene un encabezado que le identifica y describe brevemente, luego se muestra la información obtenida de manera legible y organizada en un formato PDF.

4.10 Ayuda.

La ayuda está accesible como parte del menú en todas las páginas de la aplicación, y con el fin de que el usuario vea solo la información que necesita en ese momento, cada página muestra como realizar solo aquellas operaciones que se estén realizando en el momento, además se aportan los conceptos que se manejan en la aplicación, para que el usuario se familiarice con algunas entradas.

La ayuda constará en gran parte de la explicación funcional del sistema aunque abarcará algunos temas teóricos para mayor comprensión. Esto tiene el objetivo de que el usuario no solo tenga la explicación funcional del sistema sino que también pueda entender en que consiste el mismo y tenga mayor información en caso de decidir posteriormente en su mantenimiento.

Conclusiones.

En el presente capítulo, se desarrollaron los diagramas de clases del análisis y el diseño de la aplicación. Además se obtuvo el diagrama de la base de datos del sistema. Se describieron los principios de diseño seguidos: los temas de estándares de la interfaz, concepción del tratamiento de errores de la aplicación y el sistema de ayuda que tendrá.

CAPÍTULO 5: IMPLEMENTACIÓN

Tras haber creado una propuesta del modelo de implementación, en el capítulo anterior. Se hace necesario definir como se implementarán los elementos del modelo de diseño.

Este capítulo tiene el objetivo de describir cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo se organizan de acuerdo a los nodos específicos, del modelo de despliegue.

Los diagramas de despliegue y componentes conforman el modelo de implementación al describir los componentes, su organización y dependencia entre nodos físicos, en los que funcionará la aplicación.

5.1 Implementación.

En este flujo de trabajo, la tarea fundamental es conseguir un programa ejecutable que implemente a los modelos conseguidos en diseño. En el flujo de implementación se deben implementar la arquitectura, las clases y los subsistemas, integrándolos para conseguir el sistema final. Una tarea esencial a tener en cuenta en este flujo es decidir en qué entorno habrá que realizar la implementación.

Algunas veces este aspecto viene definido en la especificación de requisitos pero en otros casos no se sabe en qué lenguaje se trabajará hasta que no se ha diseñado el sistema. Una vez definido el entorno y con las especificidades del diseño, la tarea de la implementación se convierte prácticamente en traducir a código lo que se ha diseñado. Así por ejemplo, del diseño básico se extrae información sobre la arquitectura del sistema, los sistemas de almacenamiento a usar y la estructura de la base de datos que se le asociará.

5.2 Diagramas de Implementación

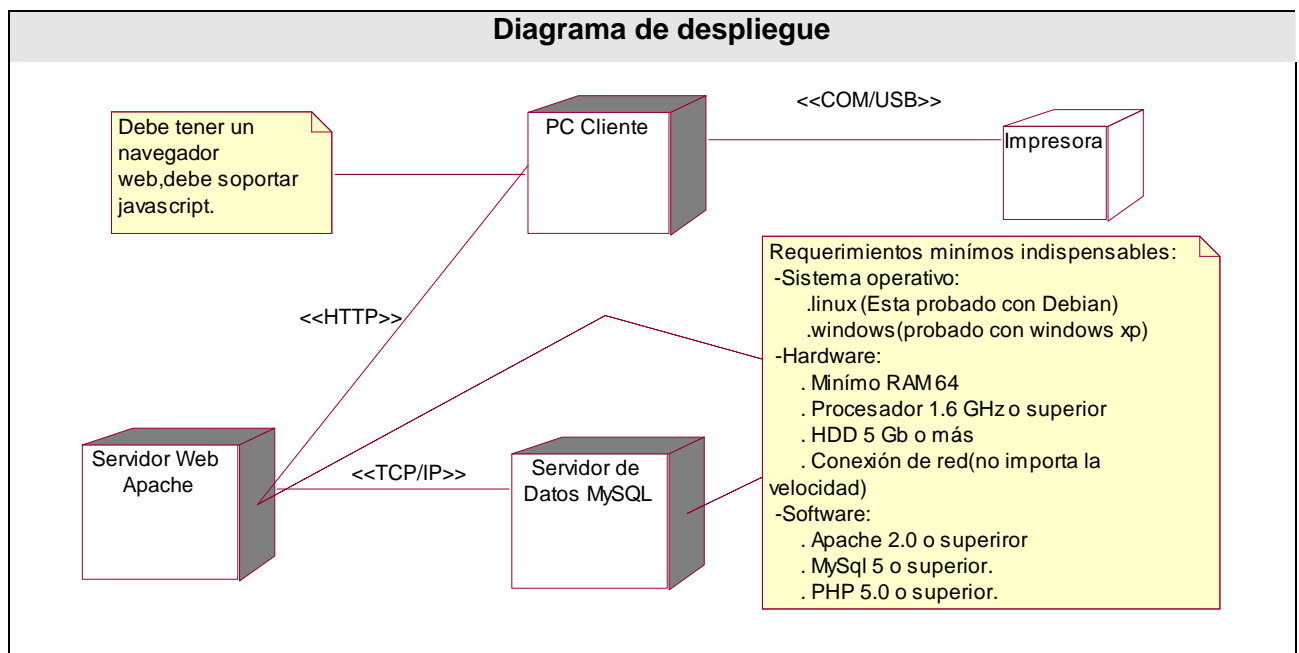
Un diagrama de implementación muestra las dependencias entre las partes de código del sistema (diagrama de componentes) o la estructura del sistema en ejecución (diagrama de despliegue): los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema, mientras que los diagramas de despliegue se utilizan para modelar la vista de despliegue estática.

5.3 Diagrama de despliegue.

El diagrama de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Es una colección de nodos y arcos; donde cada nodo representa un recurso de cómputo, normalmente un procesador o un dispositivo de hardware similar.

Muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un nodo es un objeto físico en tiempo de ejecución, es decir una máquina que se compone habitualmente de, por lo menos, memoria y capacidad de procesamiento, a su vez puede estar formada por otros componentes.

El diagrama de despliegue muestra la topología del hardware sobre el que se ejecuta el sistema.



5.4 Diagrama de componentes.

Un diagrama de componentes muestra un conjunto de componentes y sus relaciones. Estos diagramas se utilizan para describir la vista de implementación estática de un sistema. Los diagramas de componentes se relacionan con los diagramas de clases, ya que un componente normalmente se corresponde con una o más clases, interfaces o colaboraciones.

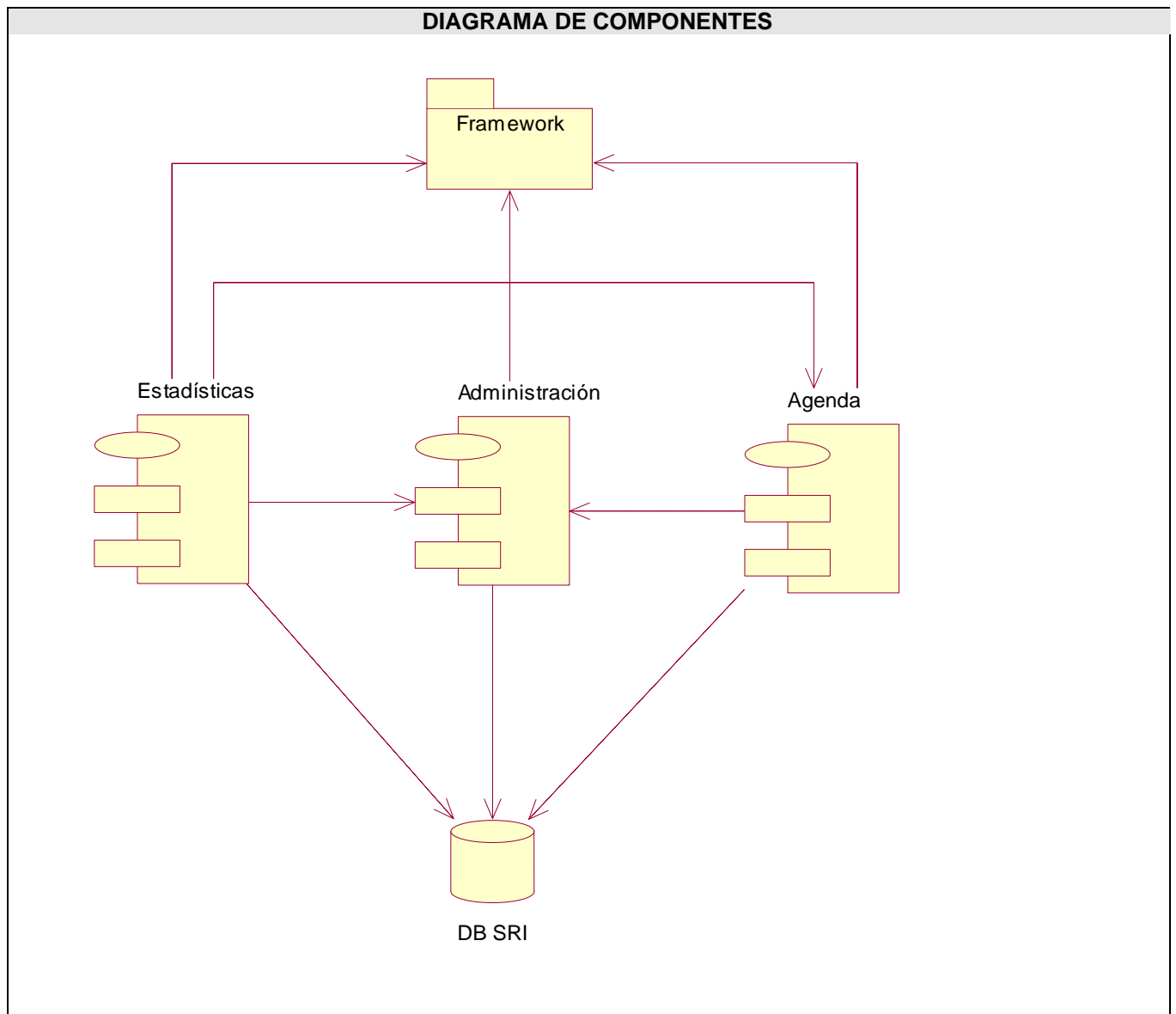


Diagrama de Componentes “Administración”.

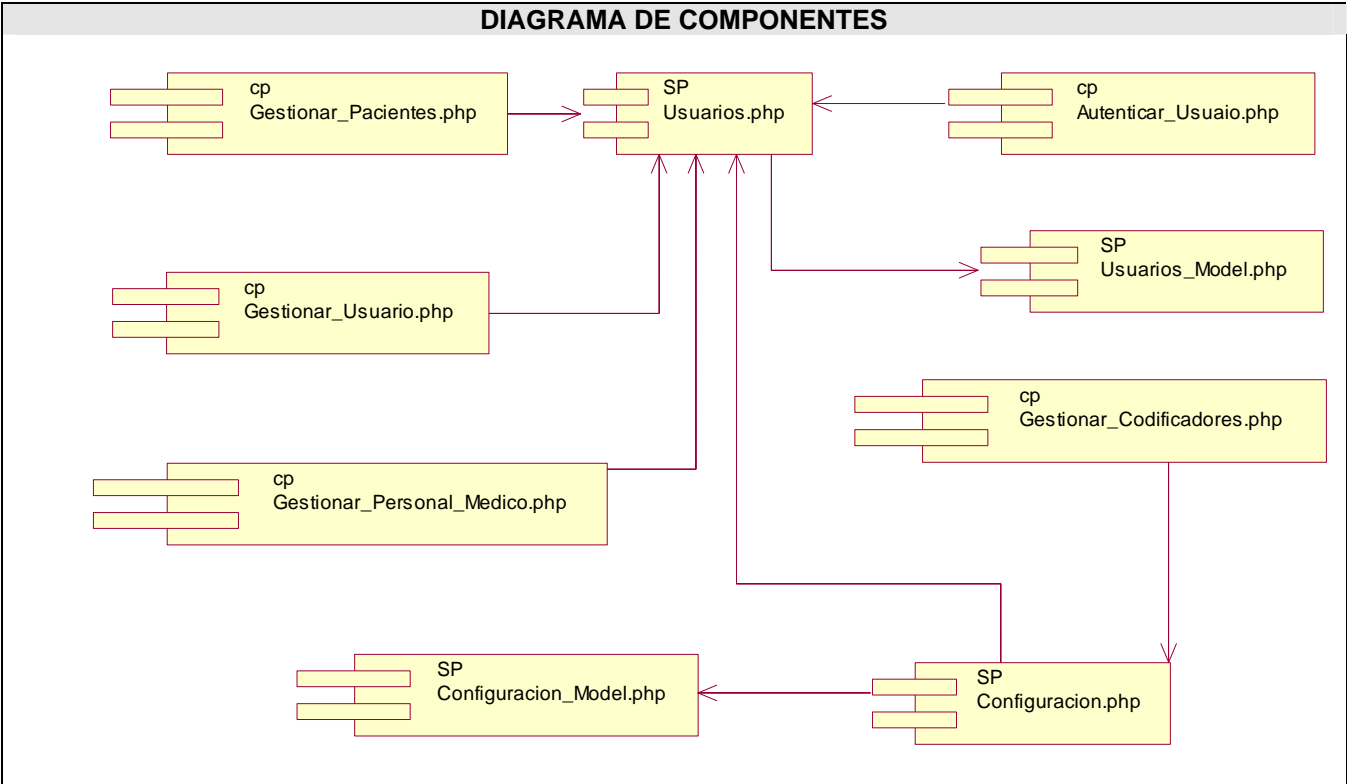


Diagrama de Componentes “Agenda”.

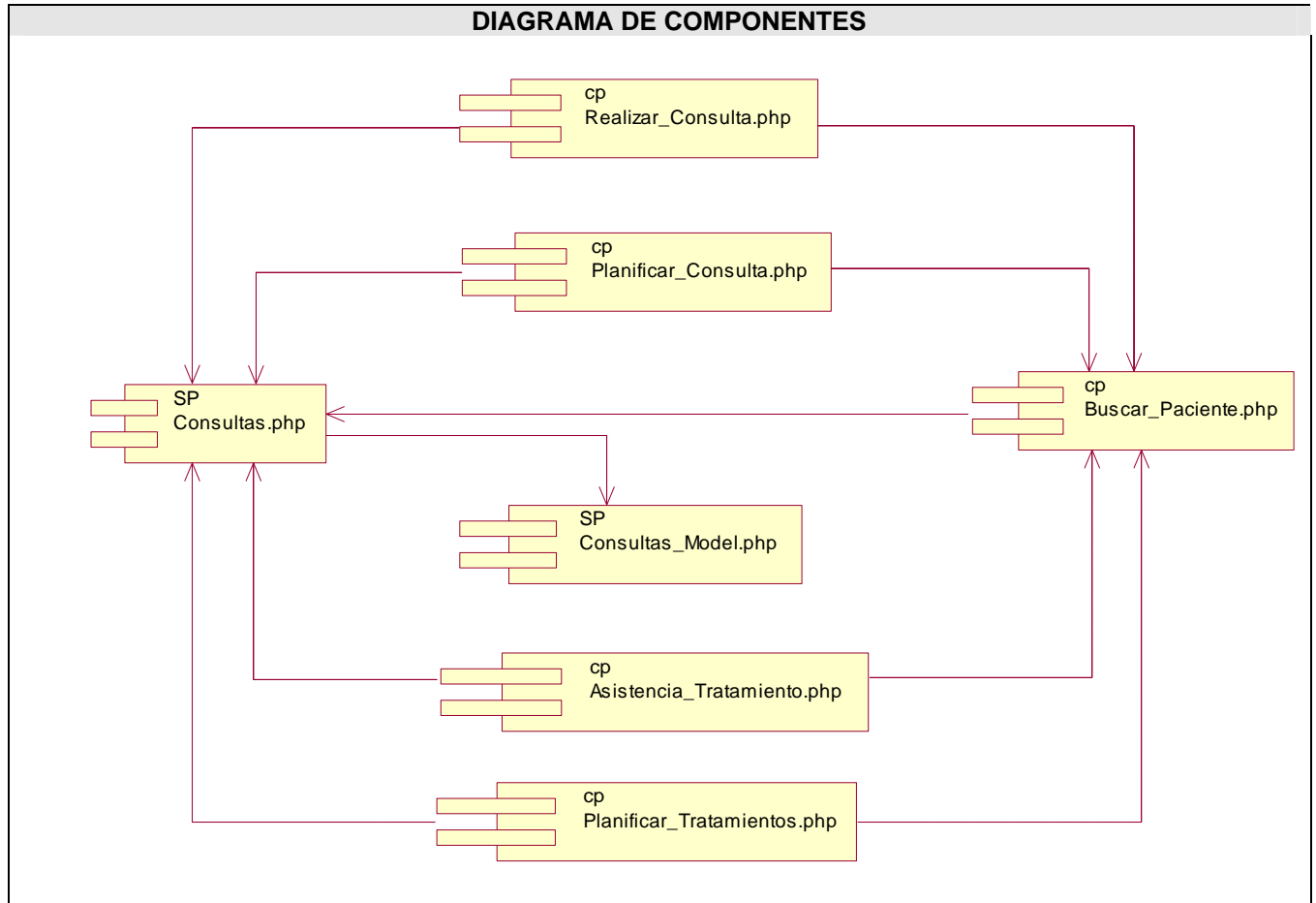
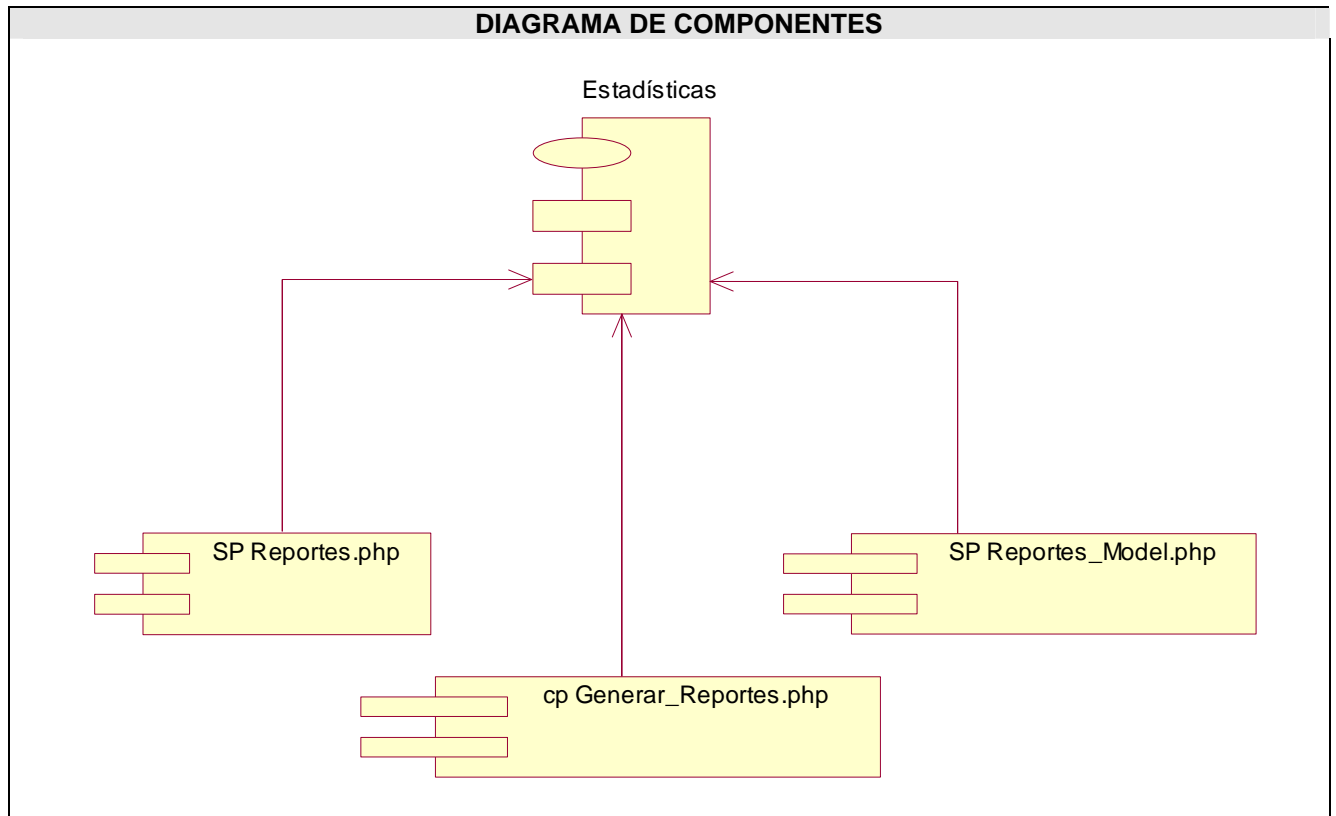


Diagrama de Componentes “Estadísticas”.**Conclusiones**

En este capítulo se plasman los resultados de la fase de implementación del sistema, y se generan diagramas de implementación a través de la utilización del lenguaje de modelado UML. Además se obtuvieron los diagramas de componentes para describir como va a estar organizado el código en la aplicación y los de despliegue para mostrar la topología del hardware sobre la cual se ejecuta el sistema.

CONCLUSIONES

Durante el desarrollo del trabajo se le dió cumplimiento a los objetivos planteados en el diseño, además se lograron los resultados propuestos:

- ✓ Se desarrolló una solución para automatizar la Gestión de la Información en los Centros de Rehabilitación de la salud, que da soporte a los procesos relacionados con el flujo médico desarrollado en estos centro; reafirmando así la utilidad y validez del empleo de las tecnologías informáticas para apoyar las labores en cualquier esfera.
- ✓ Se analizó el manejo de la información en los centros de rehabilitación de todo el país.
- ✓ Se evaluaron sistemas existentes en esta área para implementar una aplicación que cumpliera con los principales requisitos.
- ✓ Se realizó un estudio detallado de las arquitecturas, seleccionando la idónea para el sistema, demostrando la eficacia de desarrollo de los lenguajes, tecnologías y metodología propuestos en las diferentes etapas de desarrollo de la aplicación.

RECOMENDACIONES

Como recomendaciones del trabajo:

- ✓ Lograr una integración con los sistemas ya implementados en INFOMED, así como con el componente de seguridad SAAA, implementado por la empresa SOFTEL.
- ✓ Desarrollar otros módulos, para gestionar el control y disponibilidad de los medios de las salas que conforman el centro de rehabilitación.
- ✓ Utilizar la Tecnología AJAX (Asíncrono JavaScript y XML) en la aplicación, para lograr una apariencia más amigable.

REFERENCIAS BIBLIOGRÁFICAS

- [1] D. S. H. Tápanes, 2007. disponible en: <http://www.sld.cu/sitios/rehabilitacion/temas.php?idv=4581>.
- [2] D. S. H. Tápanes, 2007. Disponible en: <http://www.sld.cu/sitios/rehabilitacion/temas.php?idv=4582>.
- [3] ALCFisio©. Disponible en: <http://www.alcprogramas.com/alcfisio.htm>.
- [4] QSOFT. Disponible en: <http://www.fisiosalus.com/>.
- [5] Internet. Disponible en: <http://es.wikipedia.org/wiki/Internet>.
- [6] Aplicaciones Web. Disponible en: http://es.wikimedia.org/wiki/Aplicaci%C3%B3n_web.
- [7] Cliente Servidor. Disponible en: <http://es.wikimedia.org/wiki/Cliente-servidor>.
- [8] Servidor Apache. Disponible en: http://es.wikimedia.org/wiki/Servidor_HTTP_Apache.
- [9] Lenguaje PHP. Disponible en: <http://es.wikimedia.org/wiki/PHP>.
- [10] Lenguaje Java Script. Disponible en: <http://es.wikipedia.org/wiki/Javascript>.
- [11] Patrones de arquitectura. Disponible en: http://es.wikipedia.org/wiki/Modelo_Vista_Controlador.
- [12] Patrones de diseño. Disponible en: http://es.wikipedia.org/wiki/Patr%C3%B3n_ActiveRecord.
- [13] Un vistazo a los SGBD. Disponible en:
http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbd.php.
- [14] MySql. Disponible en: <http://es.wikipedia.org/wiki/MySQL>.
- [15] Porque MySql. Disponible en: <http://www.software-shop.com/Productos/MySQL/mysql.html>.
- [16] Jacobson, I.; Booch, G. y Rumbaugh, J. *El Proceso Unificado de Desarrollo de software*. Addison-Wesley.
- [17] Un vistazo al Fireworks. Disponible en: <http://www.portalprogramas.com/323-descargar-Macromedia-Fireworks.html>.
- [18] Photoshop. Disponible en: <http://www.desarrolloweb.com/articulos/1905.php>.
- [19] Dreamweaver. Disponible en: <http://www.desarrolloweb.com/articulos/332.php>.
- [20] Zend Studio. Disponible en: <http://www.downtwarez.com/variado/271-zend-studio-enterprise-edition-5-5-0-a.html?highlight=3d+studio>.
- [21] EMS SQL Manager. Disponible en:
<http://www.archivospc.com/programas/categorias/SQL.php?page=2>.
- [22] Cases Studio. Disponible en: <http://www.fileheaven.com/descargar/case-studio-2/26647.htm>.
- [23] Rational Rose. Disponible en: <http://ateam.lsi.upc.es/~ese/web/documents/lab/0304Q2/lessons/lese-2/LESE-2%20-%20Introduccion%20a%20Rational%20Rose.ppt?PHPSESSID=0ade5aba4d6b566cb794d3b7def7cd08>

BIBLIOGRAFÍA

- Clases de Ingeniería del Software I, curso 2006-2007, UCI.
- Introducción a php. Disponible en: <www.ciberteca.net/webmaster/php> [Fecha de consulta 24 marzo 2007].
- Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. Primera Edición por Prentice Hall, Hispanoamericana.
- Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda Edición por Prentice Hall.
- MARRERO, D. Modelado de aplicaciones Web con UML. En: Conferencia de Ingeniería de Software, Diciembre 2002, ISPJAE (CEIS).
- Matos, Rosa María. Introducción al trabajo con Base de Datos. Asignatura de Sistemas de Gestión de Base de Datos.
- Peralta, Mario. Estimación del esfuerzo basada en casos de uso. Centro de Ingeniería del Software e Ingeniería del Conocimiento, Buenos Aires, Argentina.
- Quatrani, Terry. Visual Modeling with Rational Rose 2000 and UML, Publisher Addison Wesley, Second Edition October 19, 1999
- Tutorial de MySQL. Disponible en: <<http://es.tldp.org/mysql-es/web/navegable/tutorial/tutorial.html>> [Fecha de consulta 26 marzo 2007].

ANEXO 1 MODELO DEL SISTEMA

Expansión de los Casos de Usos.

“Autenticar usuario”.

Caso de Uso:	Autenticar Usuario	
Actores:	Usuario (que puede ser Recepcionista, Técnico, Médico Fisioterapeuta)	
Resumen:	El proceso se inicia cuando el usuario introduce su nombre de usuario y contraseña, si se encuentra en la base de datos puede entrar al sistema, de lo contrario se muestra un mensaje de error para que el usuario verifique los datos entrados.	
Precondiciones:	El usuario debe estar registrado antes de autenticarse.	
Referencias	R1	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Sistema	
1. Entra usuario y contraseña.	1.1 Verificar BD	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	1.1 Se encuentra en la BD: verifica el rol que cumple, y entra al sistema. 1.2 No se encuentra en la BD: se muestra un mensaje para que verifique los datos entrados.	
Poscondiciones	El usuario ha sido autenticado	

“Gestionar Usuarios”.

Caso de Uso:	Gestionar Usuarios
Actores:	Medico Fisioterapeuta
Resumen:	<p>El proceso se inicia cuando un administrador del sistema solicita agregar, modificar o eliminar un usuario para el sistema en la base de datos.</p> <p>1. Si se solicita agregar un usuario, el sistema muestra una interfaz para entrar los datos referentes al usuario, luego el sistema valida y agrega el mismo, mostrando posteriormente un mensaje indicando que se ha agregado correctamente.</p> <p>2. Si se solicita modificar un usuario, el sistema muestra un listado de usuarios, se selecciona el usuario a modificar, y el sistema muestra los datos para modificar, a continuación el sistema valida los datos y actualiza la base de datos, mostrando un mensaje indicando que el usuario se modificó correctamente.</p> <p>3. Si se solicita eliminar un usuario, el sistema muestra un listado de usuarios, se selecciona el usuario a eliminar, y a continuación el sistema actualiza la base de datos, mostrando un mensaje indicando que el usuario se ha eliminado correctamente.</p> <p>Finaliza una vez que se realiza una de las operaciones seleccionadas.</p>
Precondiciones:	El usuario al que se le realizará la acción debe estar registrado en la base datos como personal médico.
Referencias	R.11 , R.11.1, R.11.2, R.11.3
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona una opción.	1.1 Agregar , eliminar , modificar

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>1.1 Si desea agregar: el sistema muestra una interfaz con los datos.</p> <p>Si desea modificar: el sistema muestra un listado de usuarios, se selecciona el usuario a modificar y se muestran los datos.</p> <p>Si desea eliminar: el sistema muestra un listado de usuarios, se selecciona el usuario a eliminar, y el sistema actualiza la base de datos.</p>
Poscondiciones	<p>Para las siguientes acciones:</p> <p>Agregar: Se crea un nuevo usuario.</p> <p>Modificar: Se modifica un usuario.</p> <p>Eliminar: Se elimina un usuario.</p>

Caso de uso “Generar reportes”

Caso de Uso:	Generar Reportes
Actores:	Recepcionista, Técnico, Médico Fisioterapeuta
Resumen:	El proceso se inicia cuando un usuario del sistema solicita generar un reporte, luego el sistema le da la posibilidad de escoger el reporte que va generar, y le muestra una interfaz con los datos a recoger, dándole la posibilidad de guardarlos como pdf. Finaliza una vez que se realiza la operación seleccionada.
Precondiciones:	El usuario debe haber seleccionado la opción de generar reportes.
Referencias	R.5

Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. Entrar usuario y contraseña	1.1 Verificar si se encuentra en la BD.
	2. El sistema muestra los reportes posibles
3. El usuario selecciona un tipo de reporte.	3.1 Genera el reporte.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	1.1 No se encuentra en la BD: se muestra un mensaje para que verifique los datos entrados.
Poscondiciones	El sistema genera los reportes

“Gestionar pacientes”.

Caso de Uso:	Gestionar Paciente
Actores:	Recepcionista
Resumen:	<p>El proceso se inicia cuando la recepcionista solicita agregar, modificar un paciente de la base de datos.</p> <p>1. Si se solicita agregar un paciente, el sistema muestra una interfaz para entrar los datos referentes a los pacientes, luego el sistema valida y agrega la misma, mostrando posteriormente un mensaje indicando que se ha agregado correctamente.</p> <p>2. Si se solicita modificar un paciente, el sistema muestra un listado de los pacientes, se selecciona el paciente a modificar, y el sistema muestra los datos para modificar, a continuación el sistema valida los datos y actualiza la base de datos, mostrando un mensaje indicando que el paciente se modificó correctamente.</p> <p>Finaliza una vez que se realiza una de las operaciones seleccionadas.</p>
Precondiciones:	Que el usuario haya seleccionado una de las opciones.

Referencias	R.6, R.6.1, R.6.2
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. Entrar usuario y contraseña	1.1 Verificar BD
2. El usuario selecciona una opción.	2.1 Agregar , modificar
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>1.1 Se encuentra en la BD: entra al sistema.</p> <p>No se encuentra en la BD: se muestra un mensaje para que verifique los datos entrados.</p> <p>2.1 Si desea agregar: el sistema muestra una interfaz con los datos que necesarios para esta operación.</p> <p>2.2 Si desea modificar: el sistema muestra un listado de los pacientes, donde se selecciona el paciente al que se le quiere modificar los datos y se procede con la operación.</p>
Poscondiciones	<p>Para las siguientes acciones:</p> <p>Agregar: Se crea un nuevo paciente.</p> <p>Modificar: Se modifican los datos de un paciente existente.</p>

“Buscar paciente”.

Caso de Uso:	Buscar Pacientes.	
Actores:	Recepcionista, Médico Fisioterapeuta, Técnico	
Resumen:	El proceso se inicia cuando un usuario del sistema solicita buscar alguna información. El sistema muestra una interfaz para entrar los datos referentes a la búsqueda, luego el sistema muestra los datos Finaliza una vez que se ha realizado satisfactoriamente la búsqueda.	
Precondiciones:	El usuario debe haber solicitado la búsqueda.	
Referencias	R.2	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Sistema	
1. Entrar usuario y contraseña	1.1 Verificar BD	
2. El usuario solicita la búsqueda.	2.1 El sistema localiza la información en la base de datos	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	1.1 Se encuentra en la BD: entra al sistema. No se encuentra en la BD: se muestra un mensaje para que verifique los datos entrados. 2.1 Si la respuesta resulta satisfactoria se le informa al usuario los datos solicitados. Si resulta insatisfactoria el sistema muestra un mensaje de error.	
Poscondiciones	La existencia de la información solicitada por el usuario.	

“Planificar tratamiento”.

Caso de Uso:	Planificar Tratamientos.	
Actores:	Recepcionista.	
Resumen:	<p>El proceso se inicia cuando la recepcionista solicita planificar un tratamiento, el sistema le muestra una interfaz con los días disponibles y esta selecciona el día con la hora que el paciente se realizará el tratamiento.</p> <p>Finaliza una vez que se ha realizado satisfactoriamente la planificación.</p>	
Precondiciones:	El usuario debe haber seleccionado la opción. De planificar tratamientos.	
Referencias	R.4	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Sistema	
1. Entrar usuario y contraseña	1.1 Verificar BD	
2. El usuario selecciona opción.	2.1 Planificar tratamientos	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	<p>1.1 Se encuentra en la BD: entra al sistema.</p> <p>No se encuentra en la BD: se muestra un mensaje para que verifique los datos entrados.</p> <p>2.1 Si la respuesta resulta satisfactoria se le informa al usuario los datos solicitados, y se procede a planificar los tratamientos.</p> <p>Si resulta insatisfactoria el sistema muestra un mensaje de error.</p>	
Poscondiciones	Se planifica el día del tratamiento.	

“Planificar consultas”.

Caso de Uso:	Planificar Consultas.
Actores:	Recepcionista.
Resumen:	<p>El proceso se inicia cuando la recepcionista solicita planificar una consulta, el sistema le muestra una interfaz con los días disponibles y esta selecciona el día que el paciente realizara la consulta.</p> <p>Finaliza una vez que se ha realizado satisfactoriamente la planificación.</p>
Precondiciones:	El usuario debe haber seleccionado la opción.
Referencias	R.3
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. Entrar usuario y contraseña	1.1 Verificar BD
2. El usuario selecciona la opción.	2.1 Planificar consulta.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>1.1 Se encuentra en la BD: entra al sistema.</p> <p>No se encuentra en la BD: se muestra un mensaje para que verifique los datos entrados.</p> <p>2.1 Si la respuesta resulta satisfactoria se le informa al usuario los datos solicitados, y se procede con la planificación de la consulta.</p> <p>Si resulta insatisfactoria el sistema muestra un mensaje de error.</p>
Poscondiciones	Se planifica el día de la Consulta.

“Realizar consulta”.

Caso de Uso:	Realizar Consultas
Actores:	Médico Fisioterapeuta.
Resumen:	<p>El proceso se inicia cuando el Médico solicita realizar una consulta, el sistema le muestra una interfaz con los datos que se deben llenar para esta operación.</p> <p>Finaliza una vez que se ha realizado satisfactoriamente la consulta.</p>
Precondiciones:	El usuario debe haber seleccionado la opción.
Referencias	R.8
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. Entrar usuario y contraseña	1.1 Verificar BD
2. El usuario selecciona la opción.	2.1 Realizar consulta.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>1.1 Se encuentra en la BD: entra al sistema.</p> <p>No se encuentra en la BD: se muestra un mensaje para que verifique los datos entrados.</p> <p>2.1 Si la respuesta resulta satisfactoria se le informa al usuario los datos solicitados, y realiza la consulta.</p> <p>Si resulta insatisfactoria el sistema muestra un mensaje de error.</p>
Poscondiciones	Se realiza la consulta.

“Gestionar Codificadores”.

Caso de Uso:	Gestionar Codificadores	
Actores:	Médico Fisioterapeuta.	
Resumen:	<p>El proceso se inicia cuando el administrador del sistema solicita agregar, modificar u eliminar en los codificadores existentes en la base de datos.</p> <ol style="list-style-type: none"> 1. Si se solicita agregar, el sistema muestra una interfaz para entrar los datos referentes al codificador en cuestión, luego el sistema valida y agrega la misma. 2. Si se solicita modificar, el sistema muestra una interfaz con un listado de los datos posibles a modificar, a continuación el sistema valida los datos y actualiza la base de datos, mostrando un mensaje indicando que se modificó correctamente el dato. 3. Si se solicita eliminar, el sistema muestra una interfaz con un listado de los datos posibles a eliminar, a continuación el sistema válida que la operación sea válida y elimina el dato seleccionado de la base de datos, mostrando un mensaje que se actualizó correctamente. <p>Finaliza una vez que se realiza una de las operaciones seleccionadas.</p>	
Precondiciones:	El usuario debe haber seleccionado una de las opciones.	
Referencias	R.7	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Sistema	
1. Entrar usuario y contraseña	1.1 Verificar BD	
2. El usuario selecciona la opción.	2.1 Realizar consulta.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	1.1 Se encuentra en la BD: entra al sistema.	
	No se encuentra en la BD: se muestra un	

	<p>mensaje para que verifique los datos entrados.</p> <p>2.1 Si la respuesta resulta satisfactoria se le informa al usuario los datos solicitados, y realiza la consulta.</p> <p>Si resulta insatisfactoria el sistema muestra un mensaje de error.</p>
Poscondiciones	Se realiza la consulta.

“Asistencia al tratamiento”.

Caso de Uso:	Asistencia al tratamiento.	
Actores:	Técnico fisiatra.	
Resumen:	<p>El proceso se inicia cuando el paciente acude al la sala de rehabilitación para aplicarse los tratamientos planificados, el técnico solicita la hoja de asistencia del paciente y actualiza los datos correspondientes a ese día en el tratamiento.</p> <p>Finaliza una vez que se realiza la operación seleccionada.</p>	
Precondiciones:	El usuario debe haber seleccionado la opción.	
Referencias	R.10	
Flujo Normal de Eventos		
Sección “”		
Acción del Actor	Respuesta del Sistema	
1. Entrar usuario y contraseña	1.1 Verificar BD	
2. El usuario selecciona la opción.	2.1 Asistencia al tratamiento.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	1.1 Se encuentra en la BD: entra al sistema.	
	No se encuentra en la BD: se muestra un mensaje para que verifique los datos	

	<p>entrados.</p> <p>2.1 Si la respuesta resulta satisfactoria se le informa al usuario los datos solicitados, y realiza la operación de seleccionada.</p> <p>Si resulta insatisfactoria el sistema muestra un mensaje de error.</p>
Poscondiciones	Se realiza la consulta.

“Gestionar Personal Médico”.

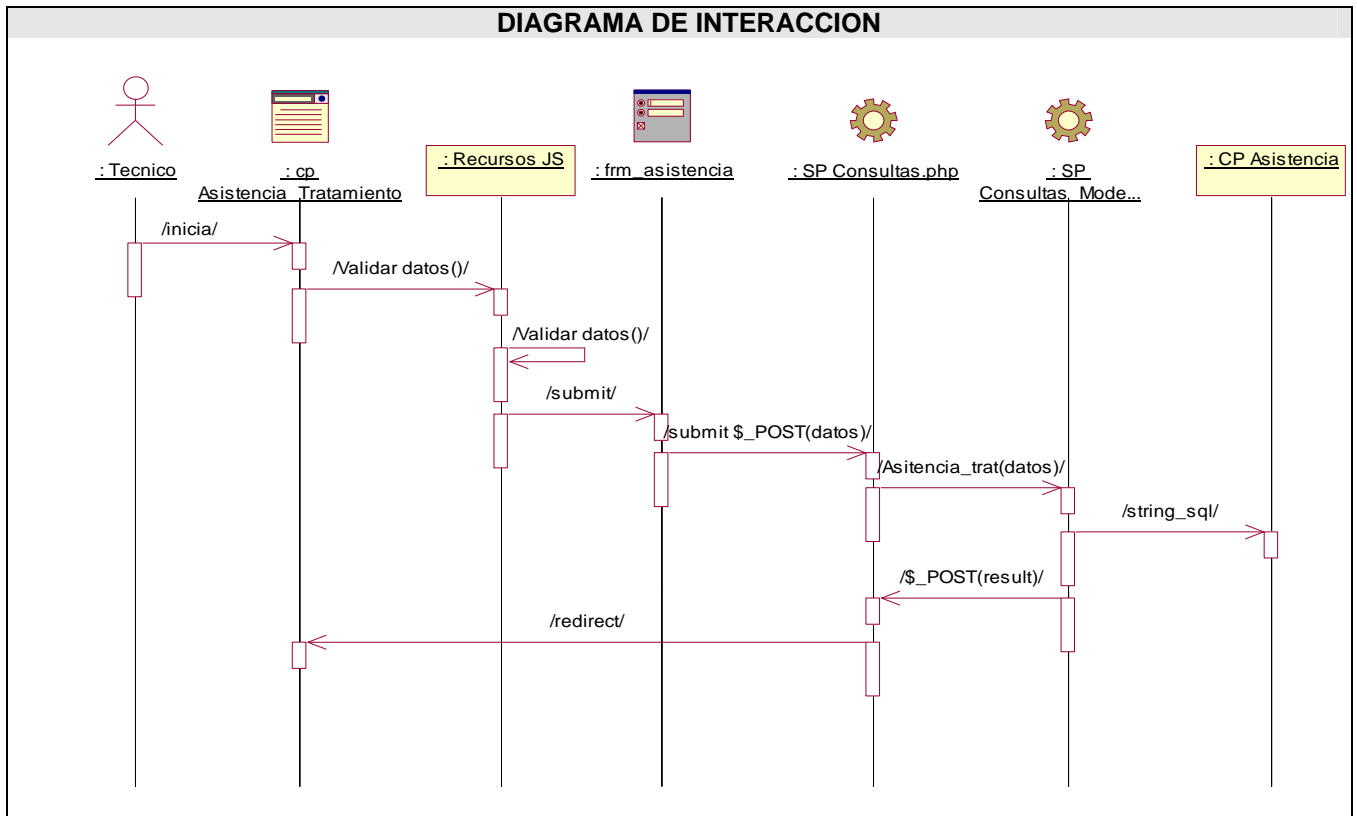
Caso de Uso:	Gestionar Personal Médico
Actores:	Recepcionista
Resumen:	<p>El proceso se inicia cuando la recepcionista solicita agregar, modificar un personal médico de la base de datos.</p> <p>1. Si se solicita agregar un personal médico, el sistema muestra una interfaz para entrar los datos referentes al personal médico, luego el sistema valida y agrega los datos, mostrando posteriormente un mensaje indicando que se ha agregado correctamente.</p> <p>2. Si se solicita modificar un personal médico, el sistema muestra un listado del personal médico existente en la bases datos, se selecciona al personal médico que será modificado, y el sistema muestra los datos para modificar, a continuación el sistema valida los datos y actualiza la base de datos, mostrando un mensaje indicando que el personal médico se modificó correctamente.</p> <p>Finaliza una vez que se realiza una de las operaciones seleccionadas.</p>
Precondiciones:	Que el usuario haya seleccionado una de las opciones.
Referencias	R.2, R.2.1, R.2.2
Flujo Normal de Eventos	
Sección “”	
Acción del Actor	Respuesta del Sistema
1. Entrar usuario y contraseña	1.1 Verificar BD

2. El usuario selecciona una opción.	2.1 Agregar , modificar
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	<p>1.1 Se encuentra en la BD: entra al sistema.</p> <p>No se encuentra en la BD: se muestra un mensaje para que verifique los datos entrados.</p> <p>2.1 Si desea agregar: el sistema muestra una interfaz con los datos que necesarios para esta operación.</p> <p>2.2 Si desea modificar: el sistema muestra un listado del personal médico, donde se selecciona el personal médico al que se le quiere modificar los datos y se procede con la operación.</p>
Poscondiciones	<p>Para las siguientes acciones: Agregar: Se crea un nuevo personal médico. Modificar: Se modifican los datos del personal médico existente.</p>

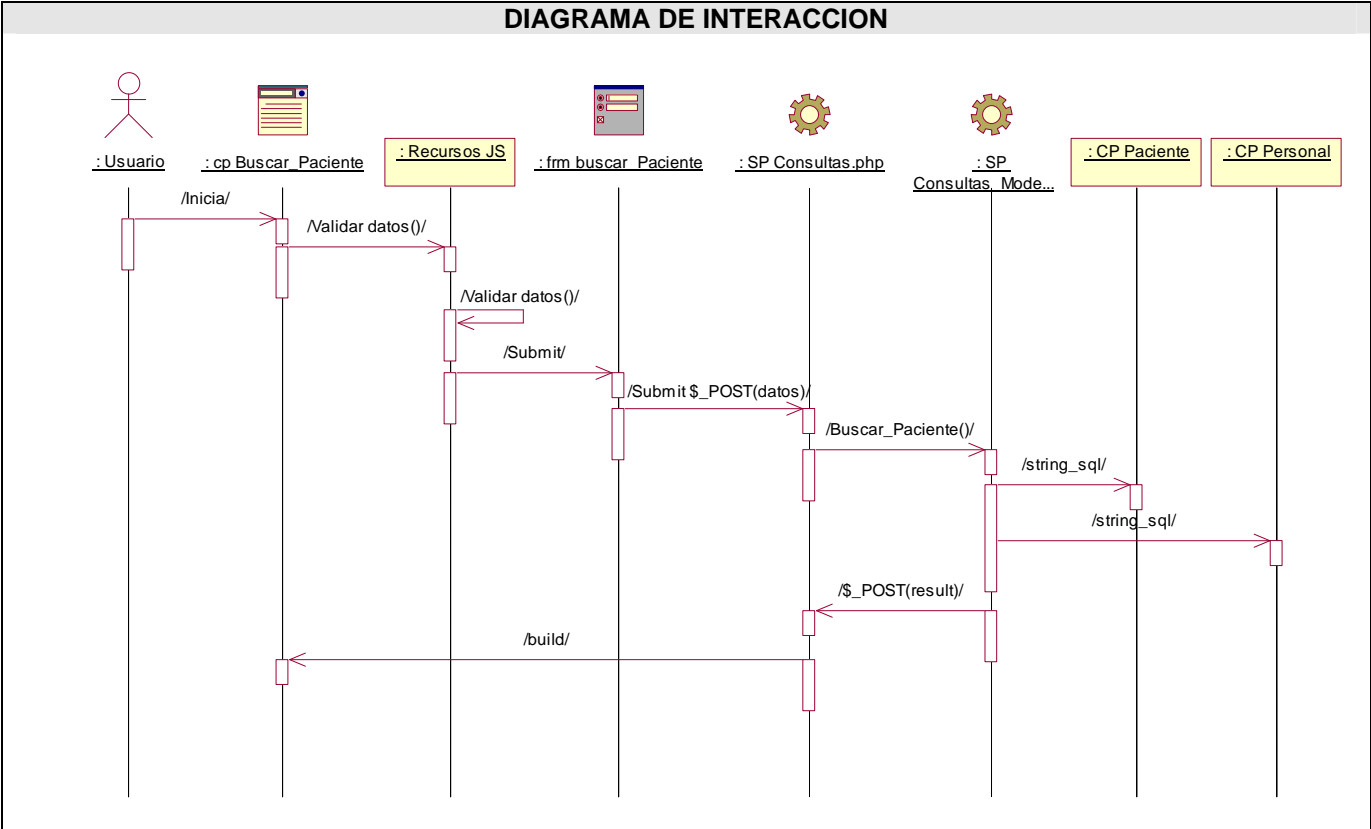
ANEXO 2 MODELO DE DISEÑO

Diagramas de Interacción.

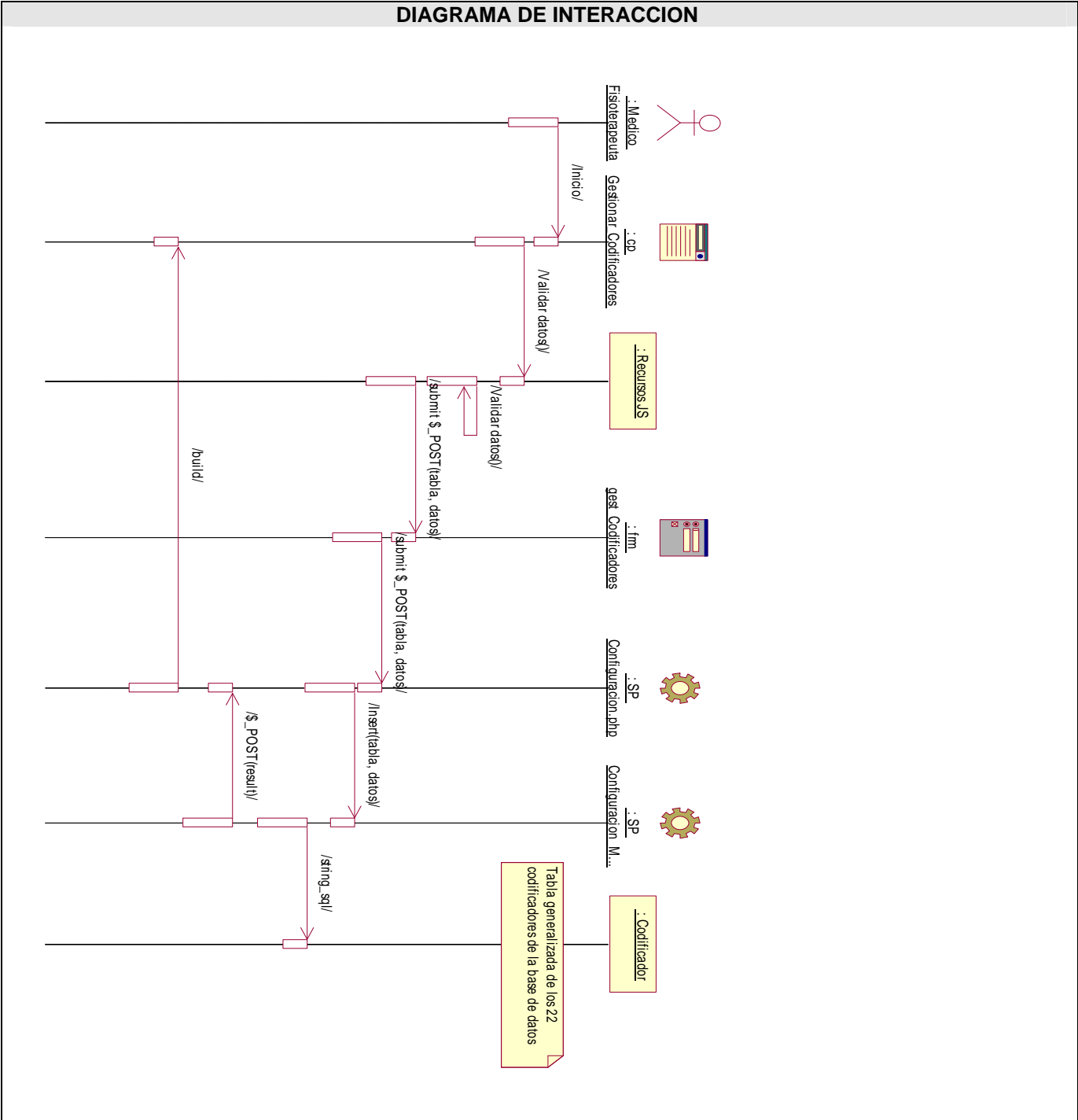
“Asistencia al Tratamiento”.



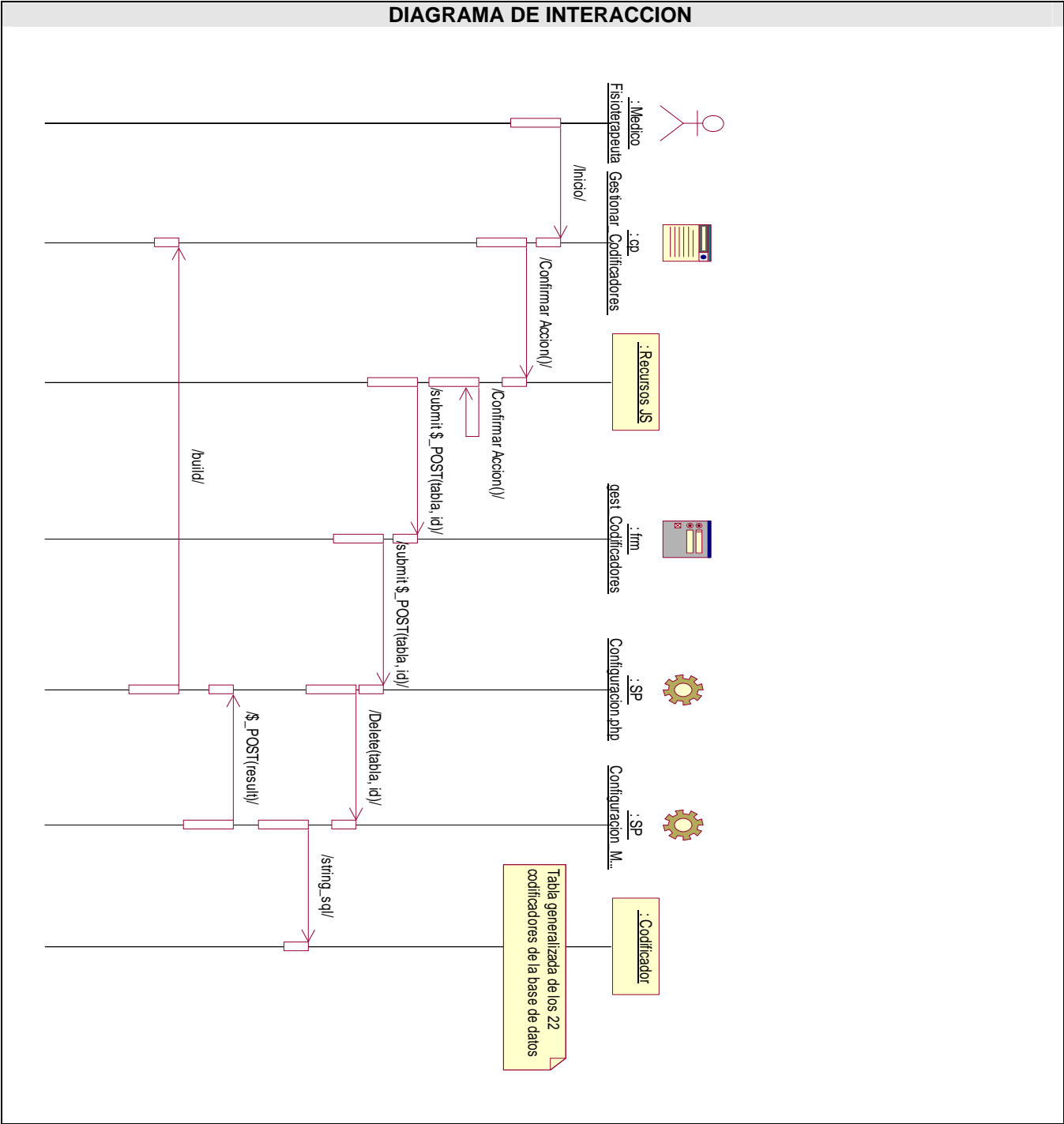
“Buscar pacientes”.



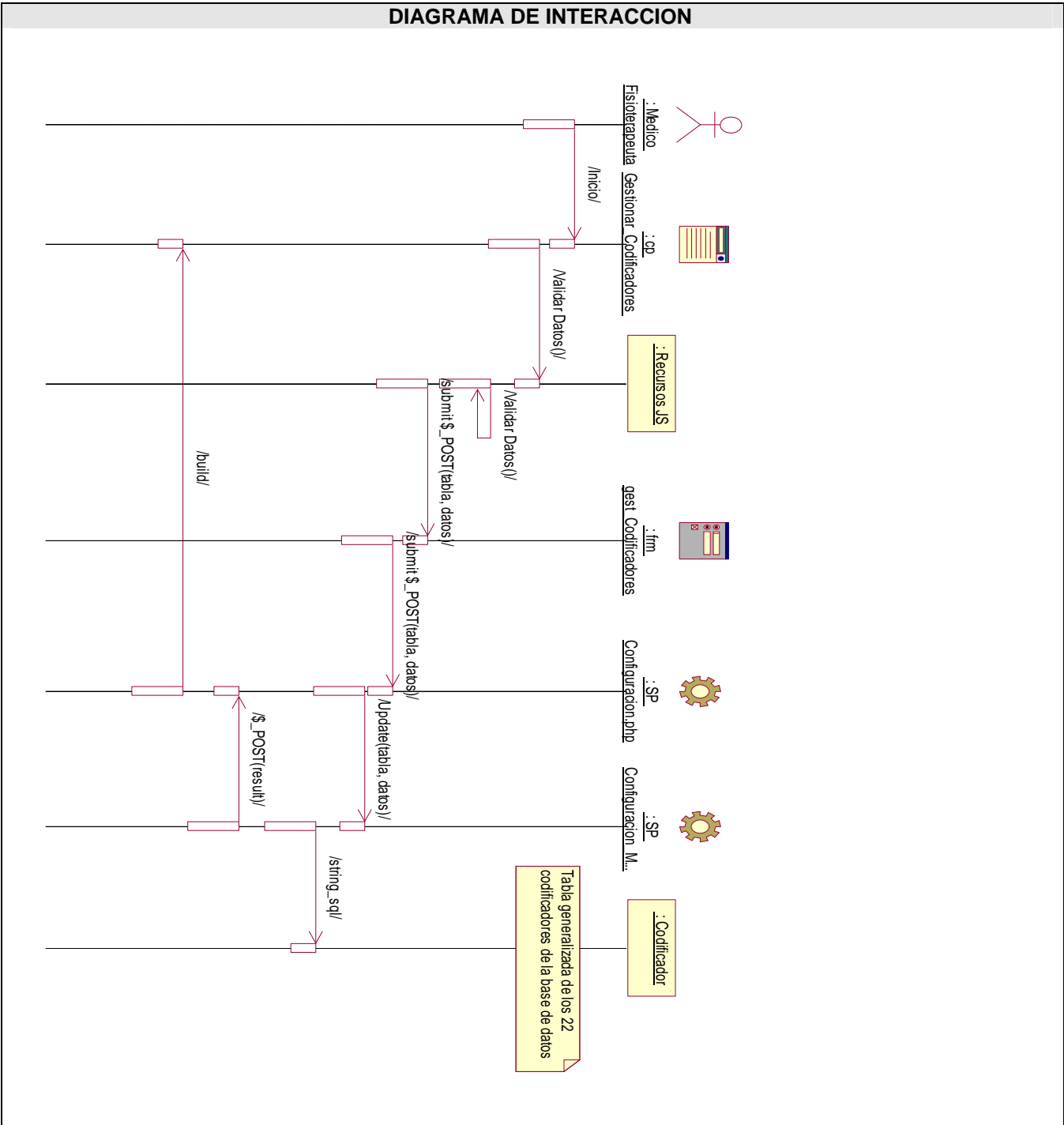
“Adicionar en los Codificadores”.



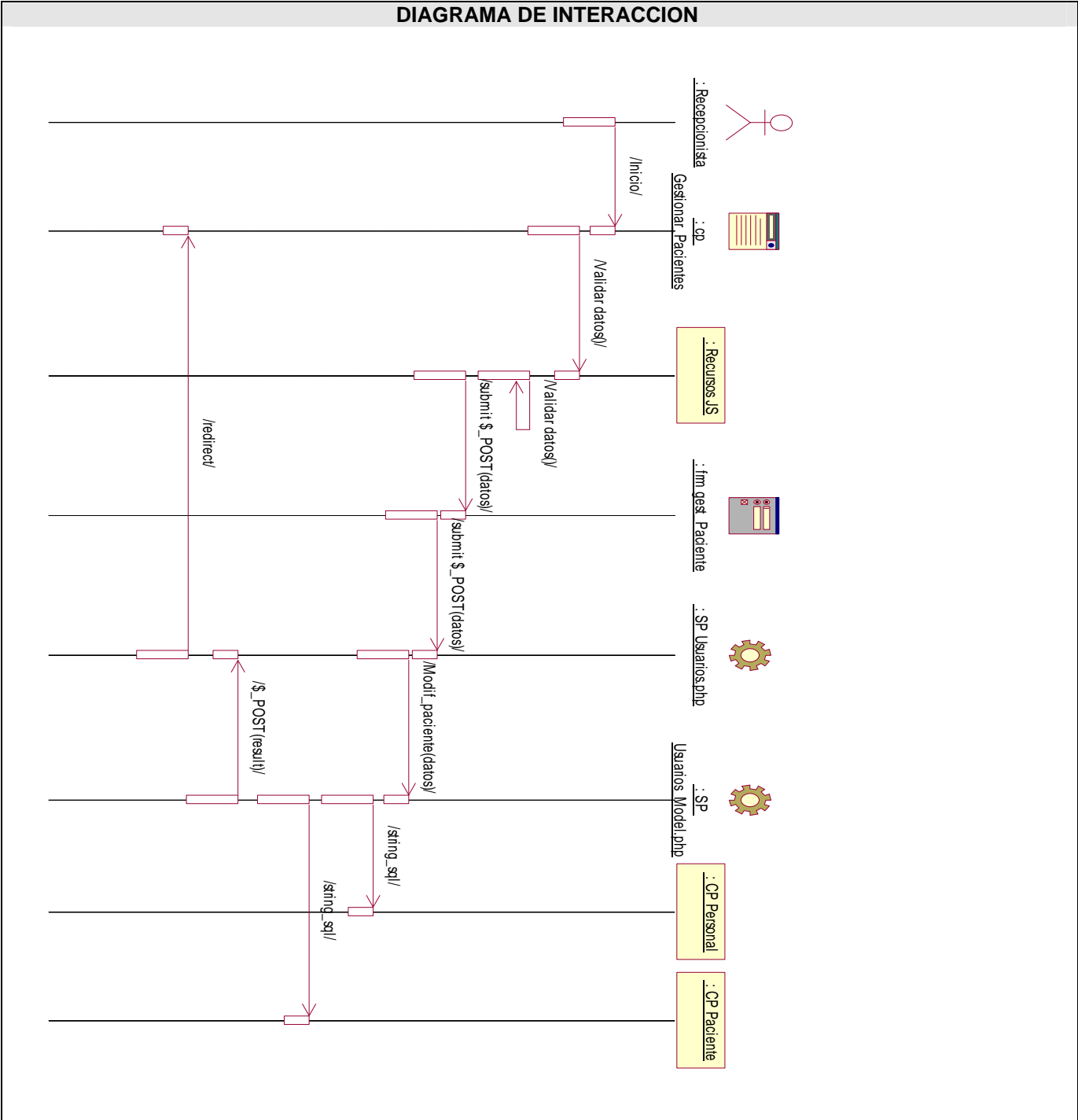
“Eliminar en los Codificadores”.



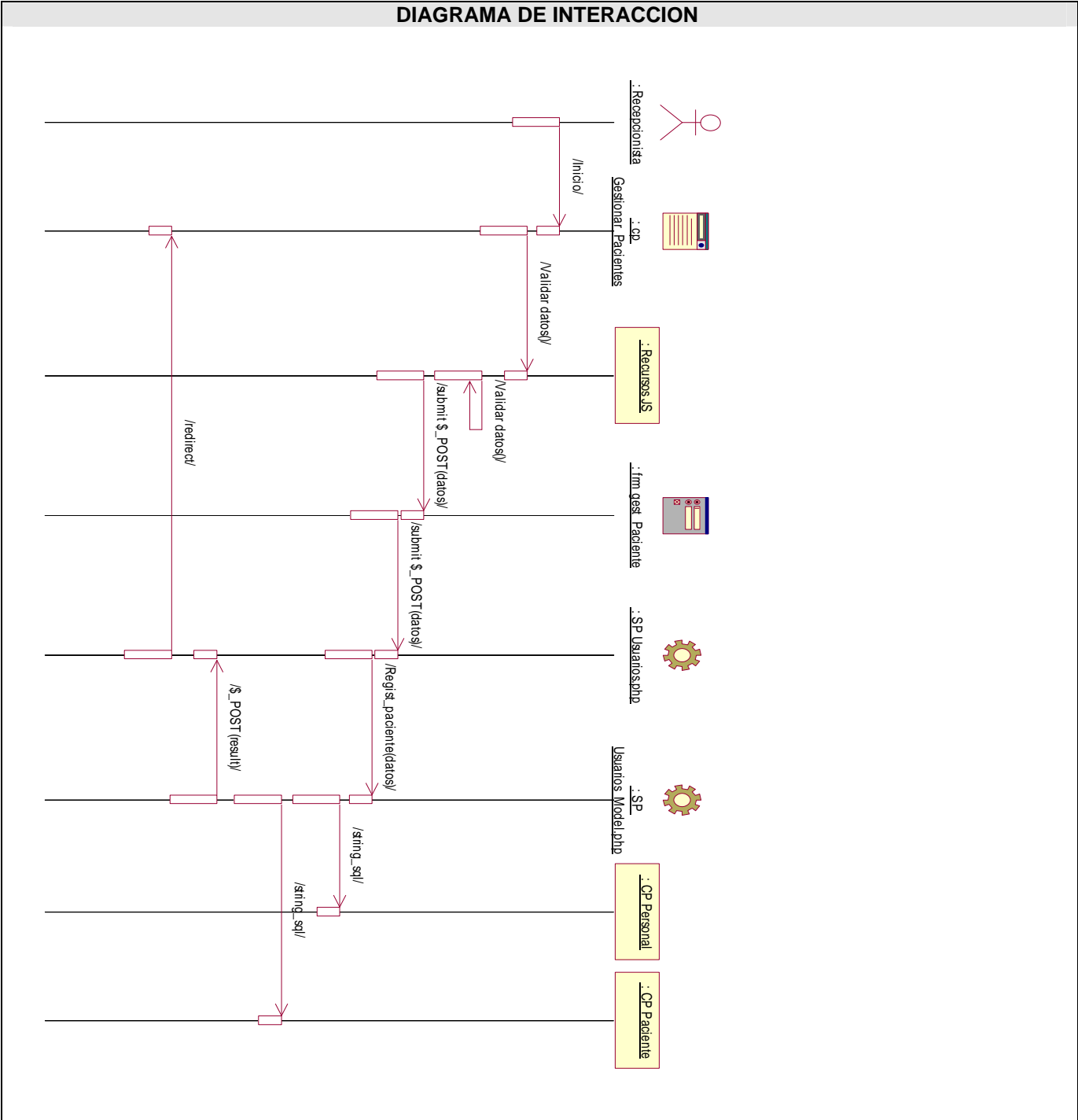
“Modificar en los Codificadores”.



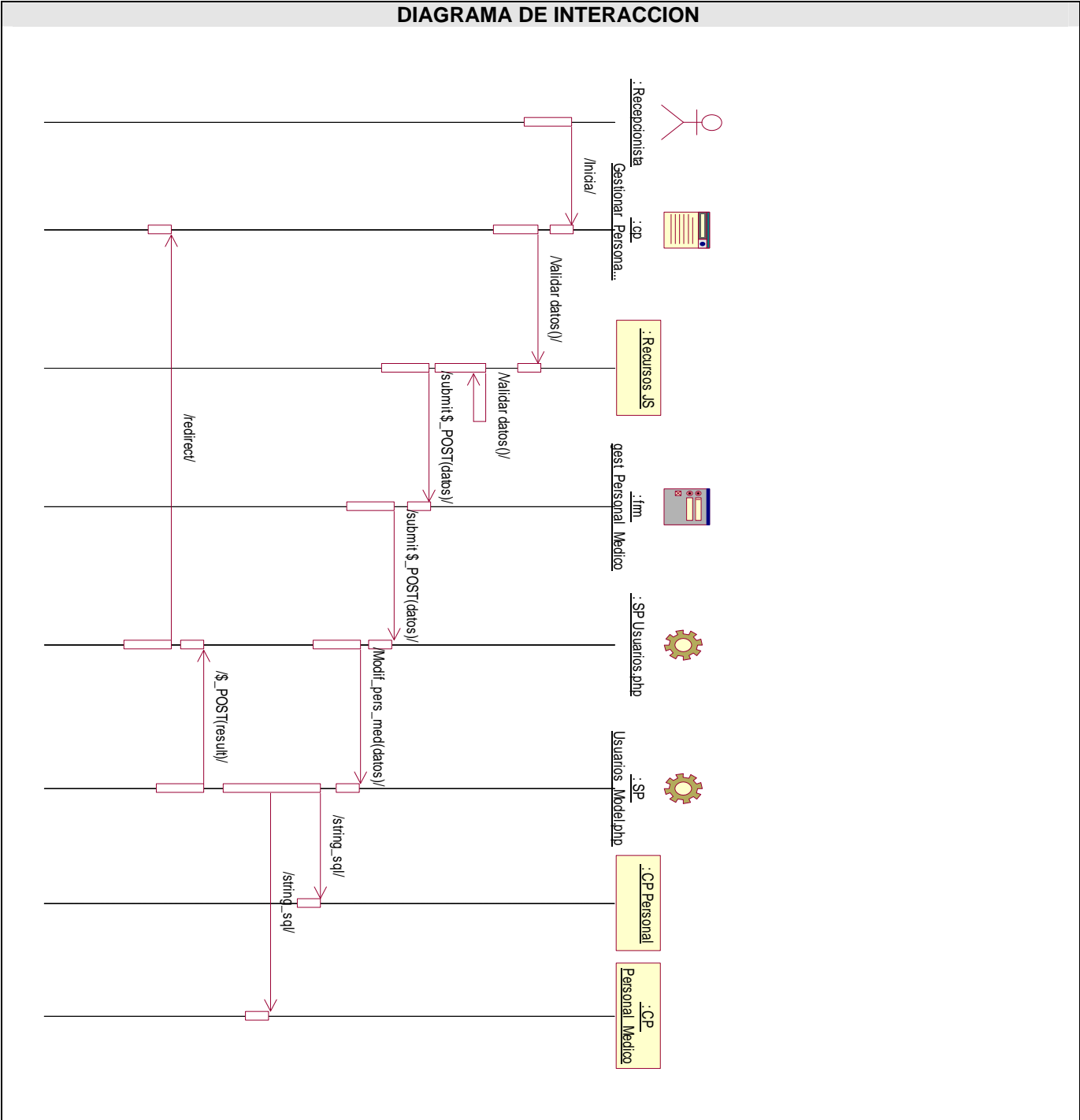
“Modificar Pacientes”.



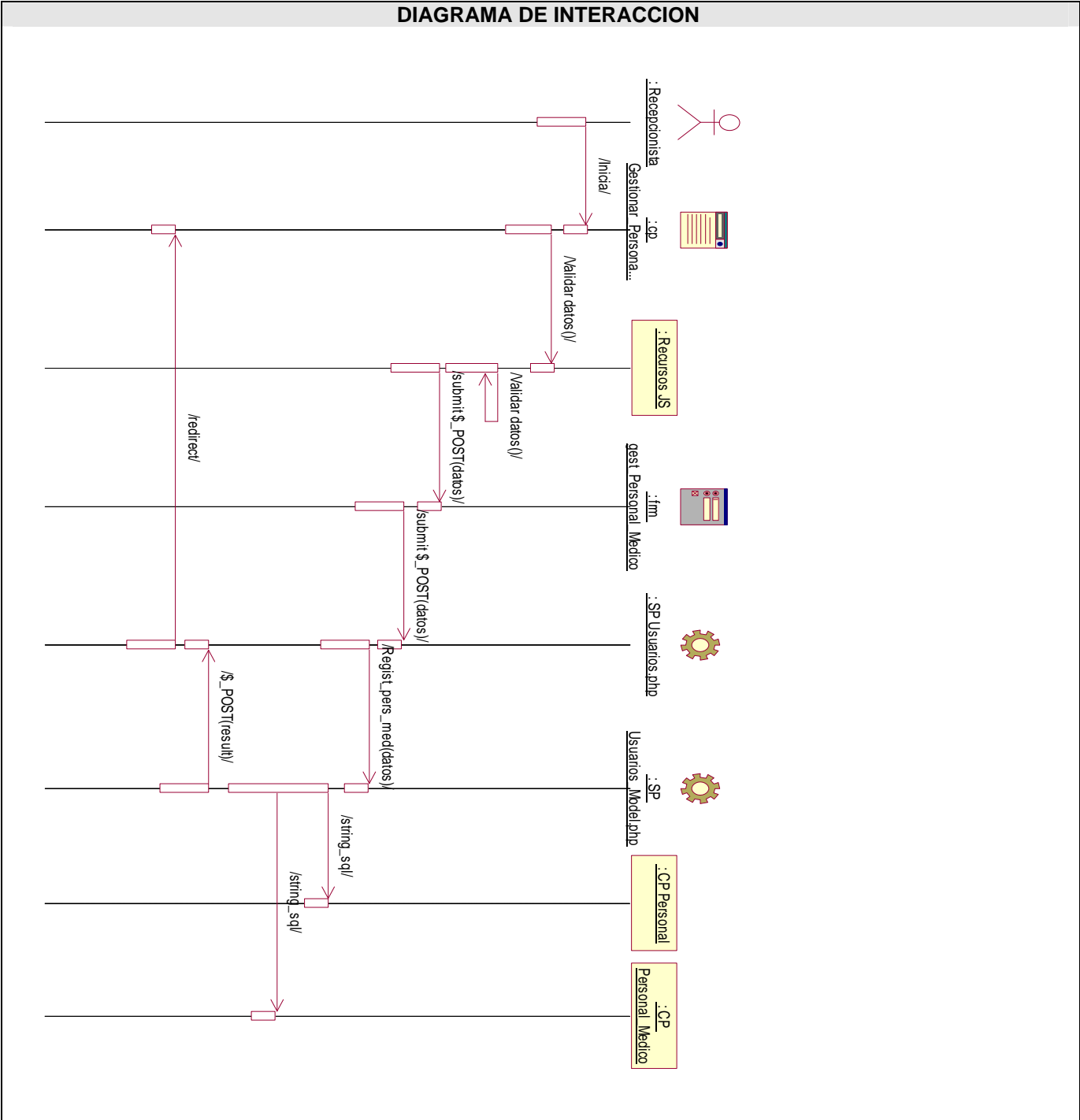
“Registrar Pacientes”.



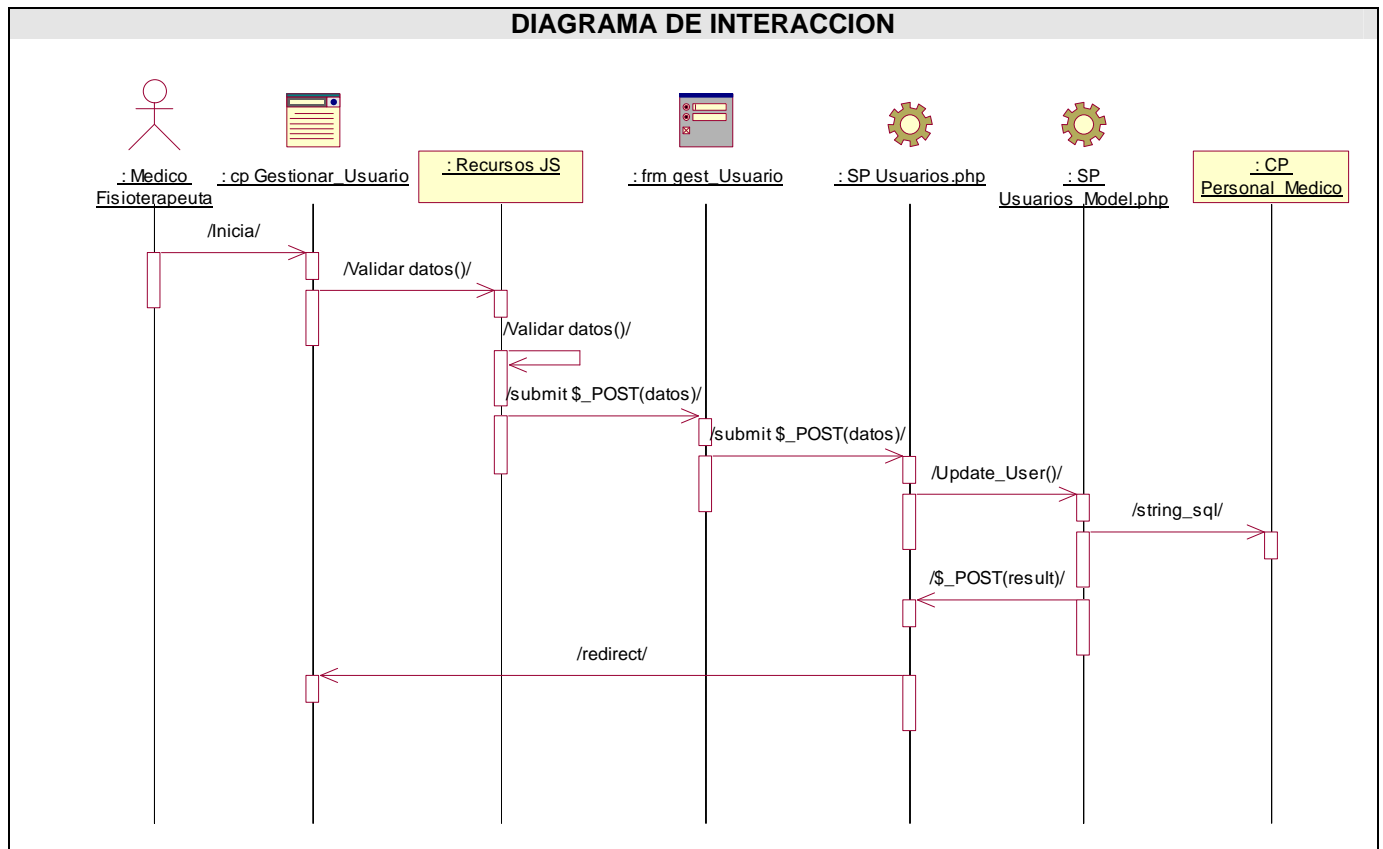
“Modificar Personal Médico”.



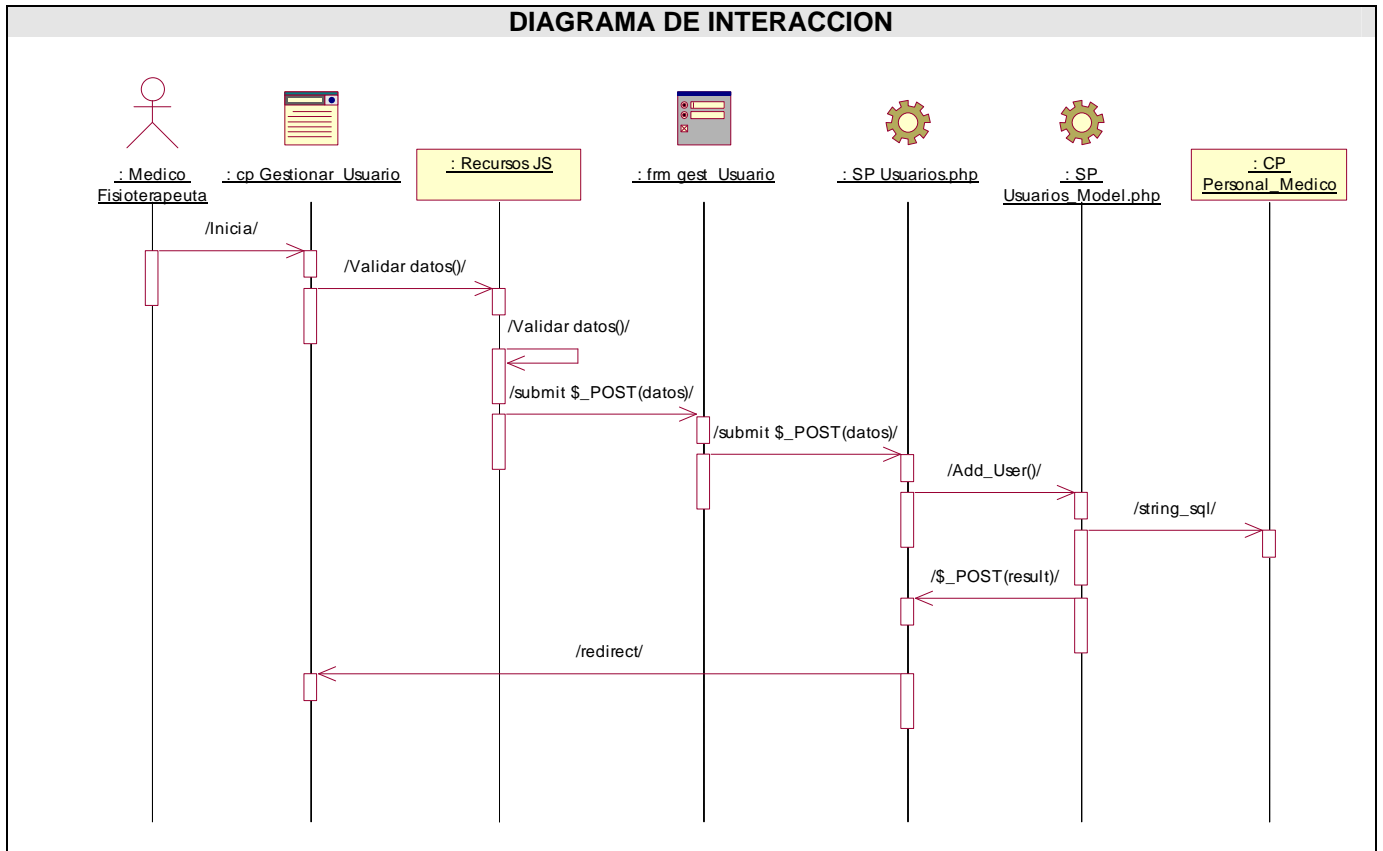
“Registrar Personal Médico”.



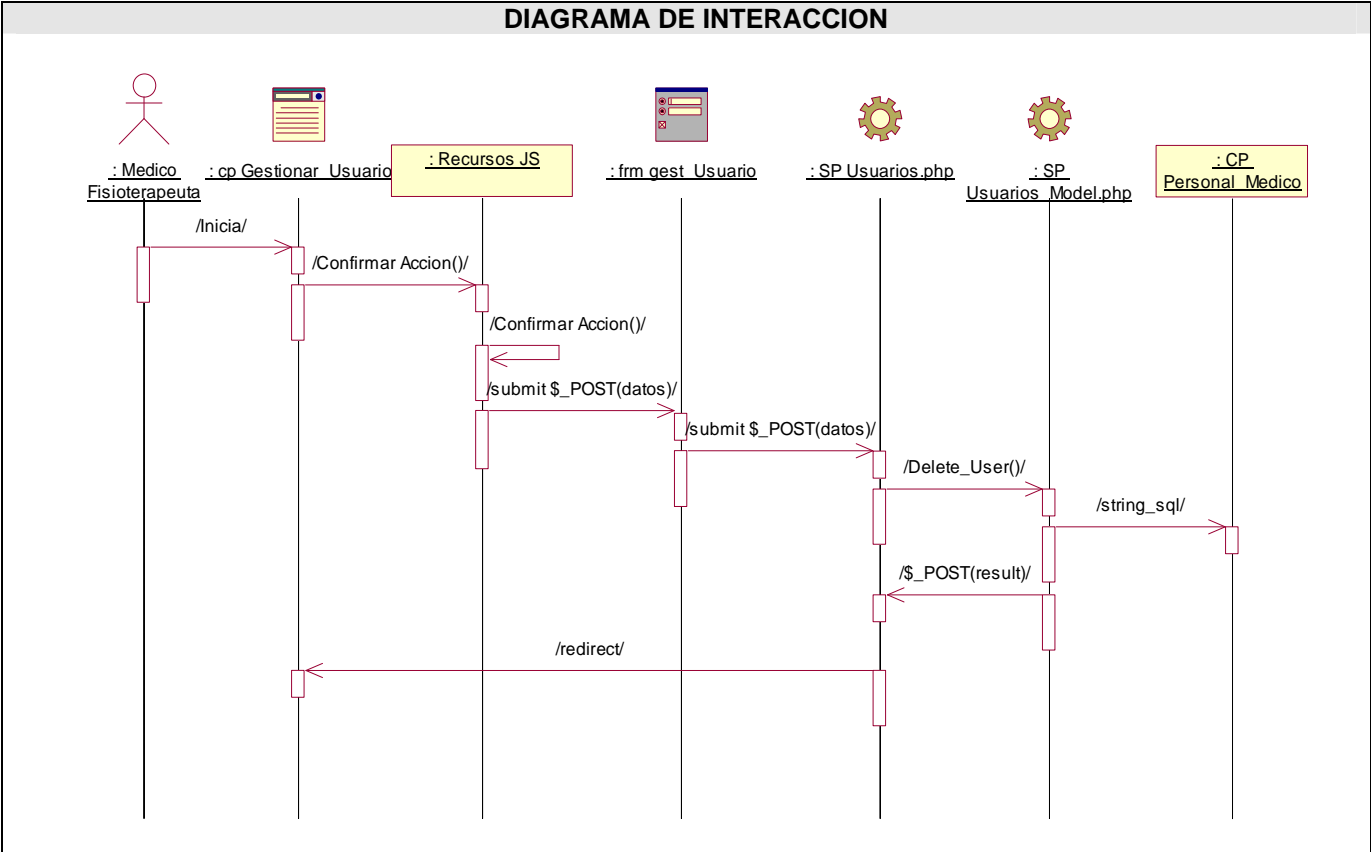
"Modificar Usuarios".



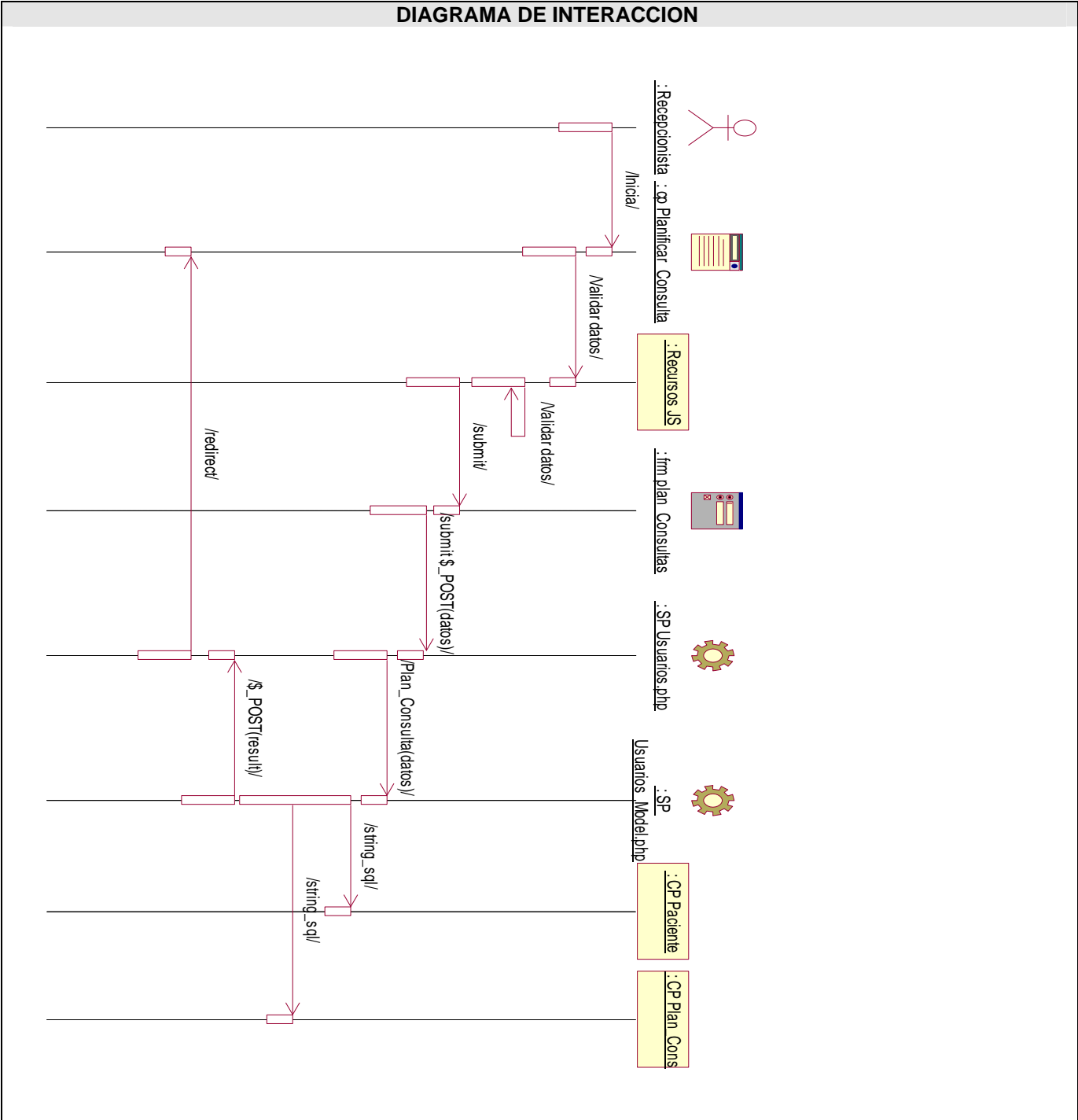
“Agregar Usuarios”.



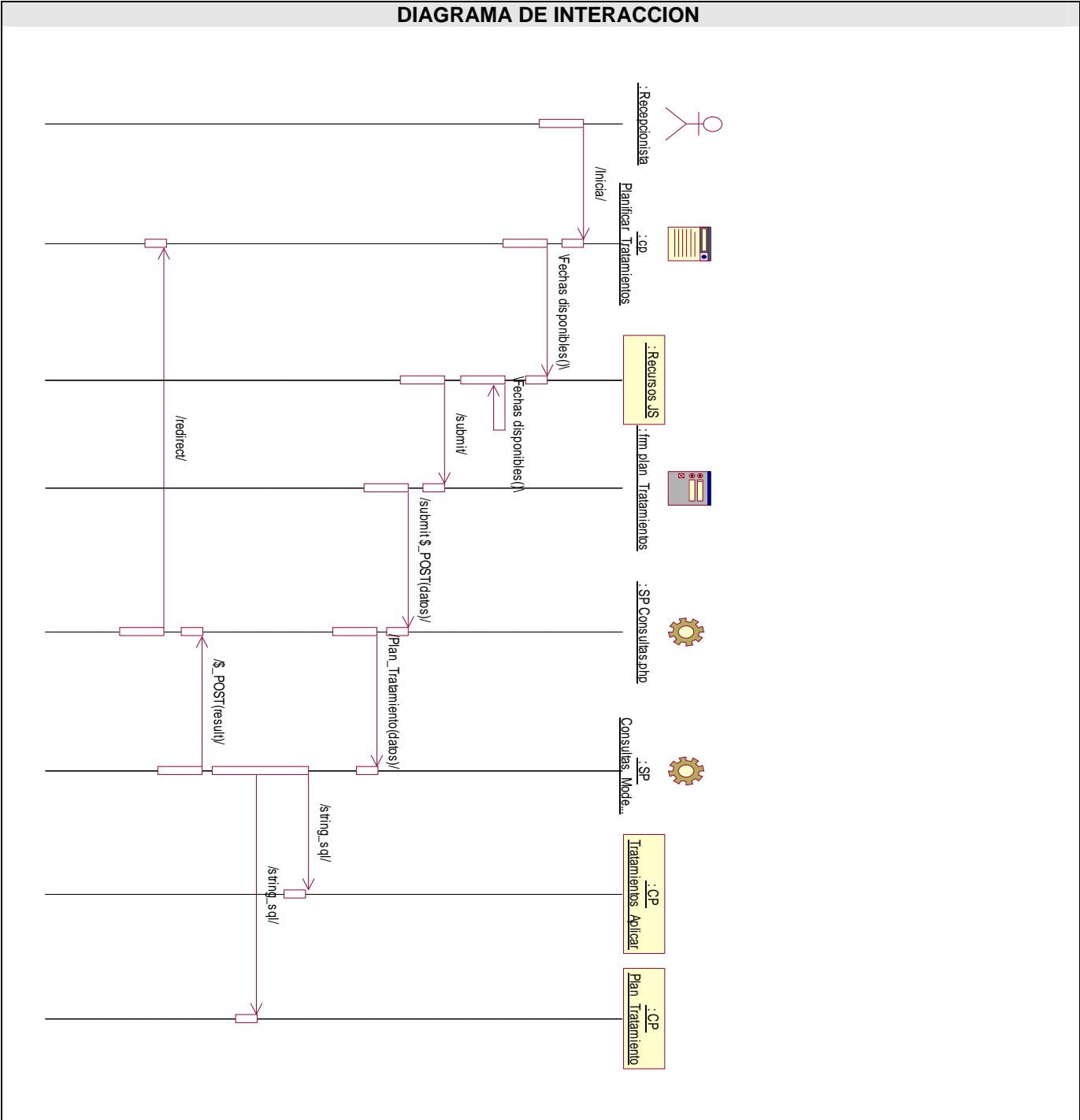
“Eliminar Usuarios”.



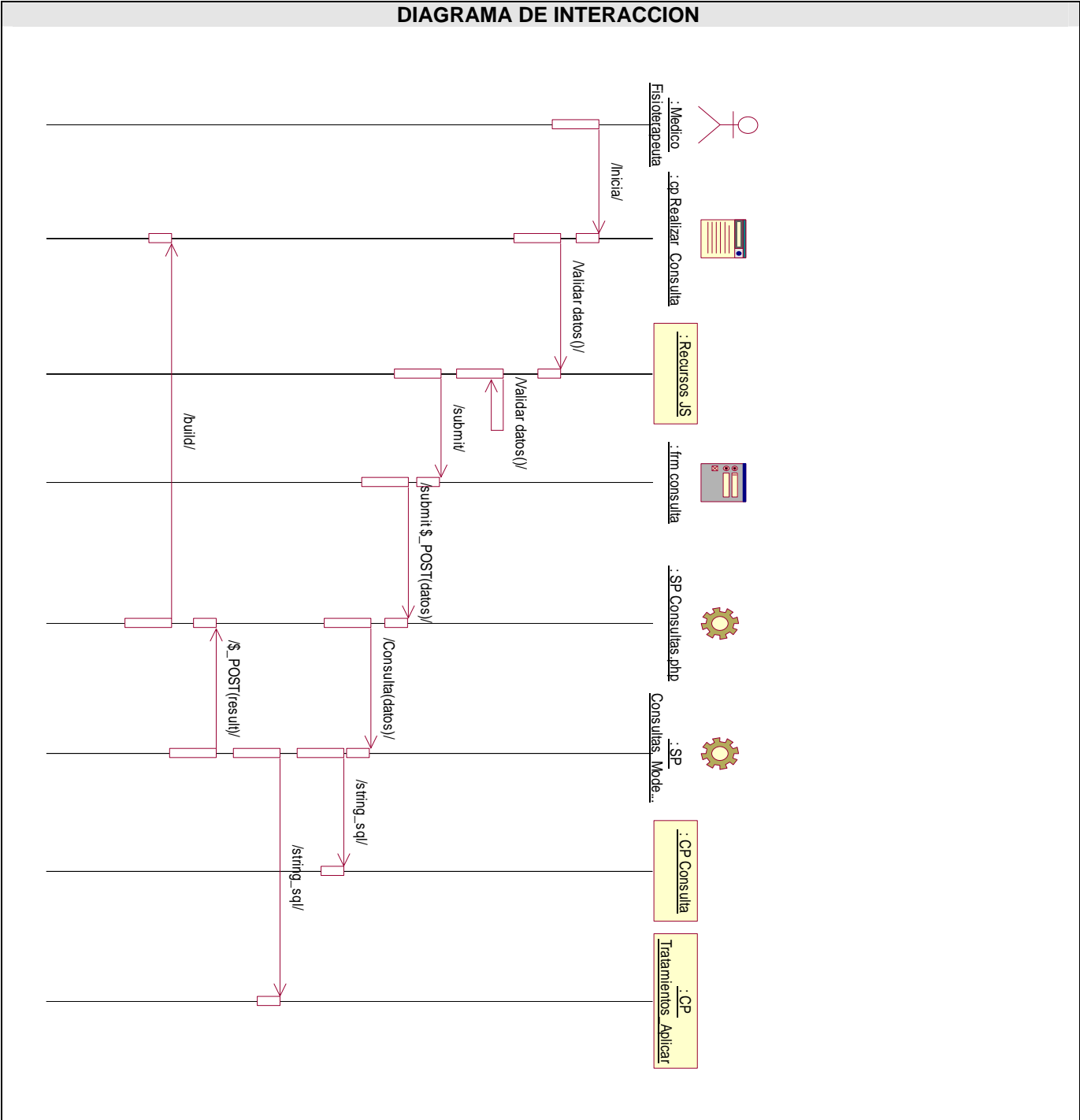
“Planificar Consulta”.



“Planificar Tratamientos”.



“Realizar Consulta”.



GLOSARIO

APACHE: es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1.

Arquitectura Cliente/Servidor: es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en elementos independientes que cooperan entre sí para intercambiar información, servicios o recursos.

CGI: *Common Gateway Interface*.

CUN: *Caso de uso del negocio*.

CUS: *Caso de uso del sistema*.

DHTML: *Dynamic HTML*.

HTML: *HyperText Markup Language*. Lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986. Es un lenguaje de marcas. Los lenguajes de marcas no son equivalentes a los lenguajes de programación aunque se definan igualmente como "lenguajes". Son sistemas complejos de descripción de información, normalmente documentos, que se pueden controlar desde cualquier editor ASCII.

HTTP: *HyperText Transfer Protocol*. Protocolo de Transferencia de Hipertextos. Modo de comunicación para solicitar páginas Web.

Hardware: Componentes electrónicos, tarjetas, periféricos y equipo que conforman un sistema de computación; se distinguen de los programas (software) porque son tangibles.

Internet: Sistema de redes de computación ligadas entre sí, con alcance mundial, que facilita servicios de comunicación de datos como registro remoto, transferencia de archivos, correo electrónico y grupos de noticias. Internet es una forma de conectar las redes de computación existentes que amplía en gran medida el alcance de cada sistema participante.

Linux: Es el nombre de un núcleo, pero se suele denominar con este nombre a un sistema operativo de libre distribución software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.

Macromedia Dreamweaver: Herramienta para el desarrollo de aplicaciones Web de Macromedia. Combina en un único entorno de desarrollo accesible y potente las reconocidas herramientas de presentación visual de Dreamweaver, las características de rápido desarrollo de aplicaciones Web de Dreamweaver UltraDev y ColdFusion Studio, y el extenso soporte de edición de código de HomeSite.

Ofrece una completa solución abierta para las tecnologías Web y estándares de hoy, incluyendo la accesibilidad y servicios Web.

Microsoft: Compañía que manufactura los sistemas de operación DOS y Windows.

MySQL: Es un sistema de gestión de bases de datos relacional que cuentan con todas las características de un motor de BD comercial: transacciones atómicas, triggers, replicación, llaves foráneas entre otras. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar.

MVC: *Modelo Vista Controlador*.

PHP: *PHP: Hypertext Preprocessor*. Es un ambiente script del lado del servidor que permite crear y ejecutar aplicaciones Web dinámicas e interactivas. Con PHP se pueden combinar páginas HTML y scripts. Con el objetivo de crear aplicaciones potentes.

PostgreSQL: es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre.

Personal médico: son los médicos, recepcionistas y técnicos de salud que laboran en los centros rehabilitación.

RUP: *Rational Unified Process* (Proceso Unificado de desarrollo). Metodología para el desarrollo de Software.

Software: Programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

SQL: *Structured Query Language*. Es un lenguaje declarativo de acceso a bases de datos que permite especificar diversos tipos de operaciones sobre las mismas. Aúna características del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar información de interés de una base de datos.

Sitio Web: Es un conjunto de páginas web, típicamente comunes a un dominio de Internet o subdominio en la World Wide Web en Internet.

SGBD: *Sistema de Gestión de Bases de Datos*. 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.

UML: *Unified Modeling Language*. Es una notación estándar para modelar objetos del mundo real como primer paso en el desarrollo de programas orientados a objetos. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema de software.

WEB (WWW): Red de documentos HTML intercomunicados y distribuidos entre servidores del mundo entero.