

Universidad de las Ciencias Informáticas
Facultad 7



**Título: Análisis, Diseño e Implementación del Sistema
de Teleconsulta**

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

Autores: Débora González Tolmo

Leosdan Pozo Águila

Tutor: Ing. Daybert Hernández Hernández

Ing. Luis Mariano Reyna Soler

Asesor: Ing. Miguel Ángel Fernández Marín

Ciudad de La Habana, junio de 2009

“ Año del 50 aniversario del triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los 29 días del mes de junio del año 2009.

Débora González Tolmo

Autora

Leosdan Pozo Águila

Autor

Ing. Luis Mariano Reyna Soler

Tutor

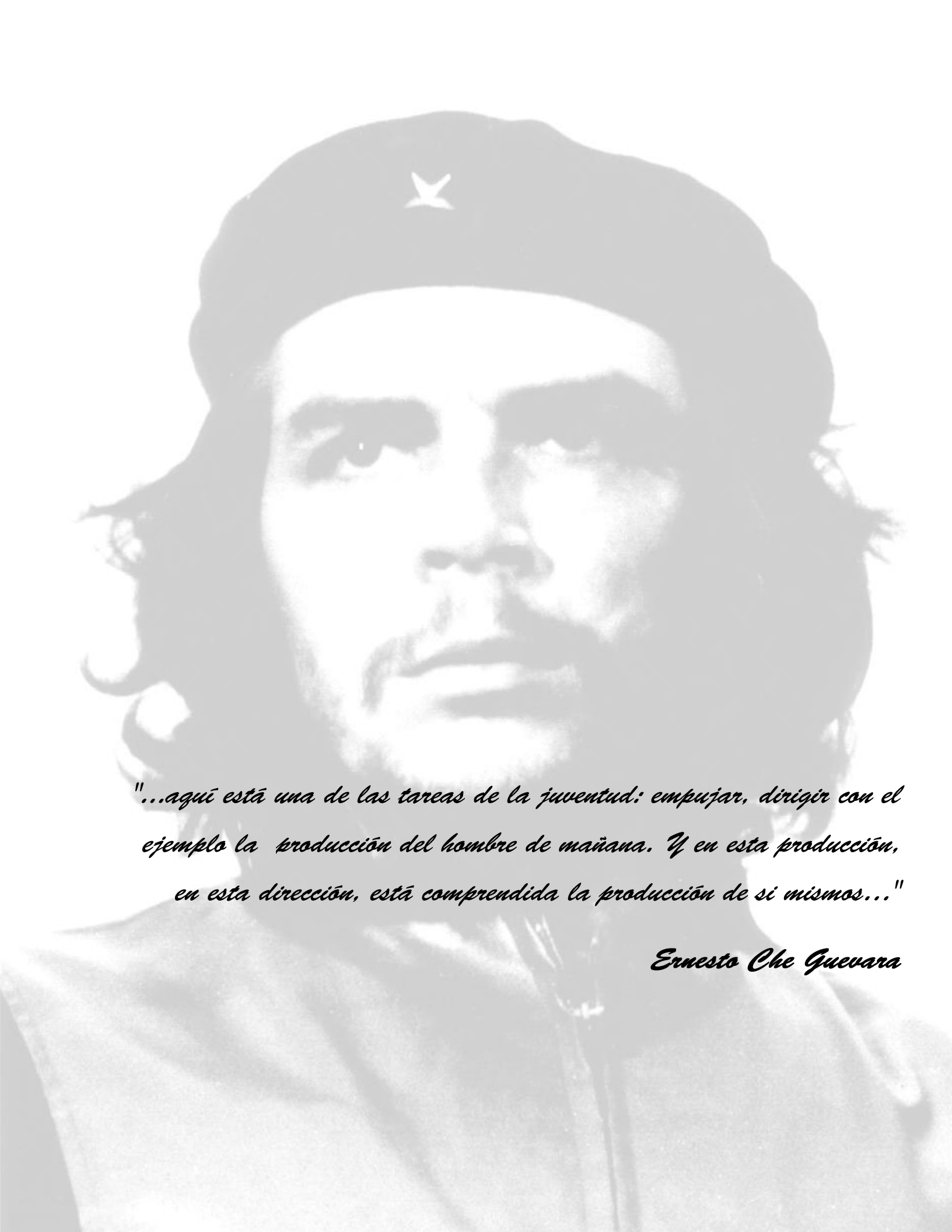
Ing. Daybert Hernández Hernández

Tutor

Datos de Contacto

Daybert Hernández Hernández (dhernandez@uci.cu): Ingeniero en Ciencias Informáticas. Profesor instructor recién graduado y profesor de Idioma Extranjero de la facultad 7.

Luis Mariano Reyna Soler (lreyna@uci.cu): Ingeniero en Informática. Profesor graduado de Ingeniería Informática en la CUJAE en el curso 2005 - 2006, profesor de Introducción a la programación y de Programación 1 desde hace dos años en la facultad 7.



"...aquí está una de las tareas de la juventud: empujar, dirigir con el ejemplo la producción del hombre de mañana. Y en esta producción, en esta dirección, está comprendida la producción de si mismos..."

Ernesto Che Guevara

AGRADECIMIENTOS

Agradecemos:

Al pilar de nuestro país: Fidel Castro Ruz y al Comandante en Jefe Raúl Castro Ruz, por haber hecho realidad los sueños de miles de jóvenes.

A nuestra Universidad de las Ciencias Informáticas, por acogernos y brindarnos la posibilidad de estudiar en una casa de altos estudios.

A nuestros profesores que nos ayudaron a formarnos profesionalmente para el futuro.

A nuestros tutores Daybert y Luis Mariano, por habernos ayudado durante el transcurso de la realización de este trabajo.

Débora González Tolmo:

Agradezco a mi familia, en especial a mis padres: Débora y Manuel, a mis hermanas: Rebeca y Raquel y a mis abuelos: Rosa y Luis Tolmo, por haberme dado su apoyo incondicional y ayudarme a seguir adelante.

Al amor de mi vida, mi esposo: Miguel Ángel, por haberme brindado su amor y paciencia y por estar en los momentos buenos y malos pese a todas las dificultades.

A todos mis amigos y compañeros de estudio, que de una forma u otra contribuyeron a que saliera adelante.

A las profesoras: Lourdes Escalona Peral, Pura Miguel Tamayo y Niurys Lázaro, por su ayuda incondicional.

Gracias a mi compañero y amigo Leosdan Pozo Águila, por estar ahí a pesar de los problemas en el transcurso de la realización del trabajo y que sin él, no hubiera sido posible culminarlo.

Leosdan Pozo Águila:

Quiero agradecer a toda mi familia, en especial a mi mamá y a mi papá por haberme guiado hasta estos momentos, por educarme como lo han hecho, por quererme tanto, y por siempre confiar en mí.

A mi vieja que tanto quiero, a la que tanto le debo en esta vida, a la que siempre ha estado al tanto de todas las cosas que me pasan.

A mi hermanita la que ha sido la fuente de mi inspiración a la que en todo momento he querido servirle de ejemplo.

A todos mis primos que más que primos han sido mis hermanos y mi fuente de apoyo, en especial, Yely, Gretel, Geidy y mi primo Yendry y mis hermanos de toda la vida Yoa y Tuan.

Además a todos mis amigos, a todos aquellos que de una forma u otra me han ayudado a lo largo de estos años. En especial a Alejandro Mario que sin su ayuda no se hubiera podido realizar esta Tesis.

A mi compañera de tesis Débora por haberse esforzado tanto y juntos haber culminado este trabajo, por estar en los buenos y malos momentos, sin importar las dificultades por las que atravesamos.

Quiero agradecer además a una persona muy especial, la cual nunca olvidaré, que desde que comenzamos me ha ayudado mucho, me ha apoyado y estado conmigo en todos los momentos, enfrentándolo todo. Esa persona es la muchacha que más quiero mi Novia Lily.

DEDICATORIA

Dedico este trabajo a mi familia, en especial a mis padres Débora y Manuel, por haberme permitido estudiar esta carrera y por ayudarme en cada paso de mi vida.

A mis abuelos Rosa y Luis Tolmo, ya que son mis segundos padres y me han cuidado en todo momento.

A mis hermanas Rebeca y Raquel, que forman una importante de mí.

A mi amado Miguel Ángel, por su inagotable amor hacia mí y por su ayuda incansable.

Débora González Tolmo

En primer lugar le dedico esta tesis a toda mi familia en especial a mi mamá, a mi papá, a mi vieja y a mi hermana, a todos mis amigos que de una forma u otra me han ayudado.

Leosdan Pozo Águila

RESUMEN

En Cuba, los profesionales de la salud se comunican a través del intercambio persona–persona, a través de vías como: teléfono, correo electrónico y en ocasiones mediante la clínica virtual cubana. Todo esto hace más compleja la evaluación colectiva entre profesionales, de un caso de difícil solución. Ocasionando dificultades en el diagnóstico temprano y en el intercambio de los profesionales de menos experiencia con los médicos más experimentados.

Para solucionar esta problemática, se desarrolló el presente trabajo que tiene como objetivo implementar un sistema informático de tipo WEB que permita facilitar el proceso de comunicación en tiempo real del personal médico del SNS para consultas o satisfacción de dudas médicas entre los hospitales de Cuba que tengan el Sistema de Gestión Hospitalaria.

Para implementar el software se utilizó como lenguaje de programación Java, el gestor de base de datos PostgreSQL 8.3 y como herramientas de desarrollo: Eclipse, al ser una plataforma robusta para Java, Case Studio v 2.22, por su potencia en el diseño de bases de datos y el Visual Paradigm 3.3, para el análisis y diseño del software.

Con el resultado de esta investigación se espera ofrecer servicios de teleconsulta para los profesionales de la salud de cualquier especialidad mediante el intercambio de información médica.

Palabras Claves: Teleconsulta.

TABLA DE CONTENIDOS

TABLA DE CONTENIDOS

AGRADECIMIENTOS.....	I
DEDICATORIA.....	III
RESUMEN.....	V
INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Definiendo las Teleconsultas.....	6
1.2 Antecedentes a nivel internacional.....	7
1.3 Antecedentes en Cuba.....	8
1.4 Conceptos Básicos Relacionados con el Problema Planteado.	9
1.5 Sistemas automatizados o Servicios de teleconsulta online existentes.....	10
1.6 Tendencias y tecnologías actuales utilizadas.....	14
CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA.....	26
2.1 Objeto de Automatización.....	26
2.2 Definición de los casos de uso.....	38
CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA.	47
3.1 Patrones de arquitectura y diseño.....	47
3.2 Estrategias de Integración.....	53
3.3 Análisis.....	54
3.4 Diseño.....	58
CAPÍTULO IV: IMPLEMENTACIÓN.....	91
4.1 Modelo de Datos.....	91
4.2 Implementación.....	95
CONCLUSIONES.....	105
RECOMENDACIONES.....	106
REFERENCIAS BIBLIOGRÁFICAS.....	107
BIBLIOGRAFIA.....	111
ANEXOS.....	116
Anexo 1 Pantalla que aparece antes que el usuario se autentique.	116
Anexo 2 Pantalla de Autenticación del Sistema de Gestión Hospitalaria.....	116
Anexo 3 Pantalla que muestra los módulos del Sistema de Gestión Hospitalaria.....	117

TABLA DE CONTENIDOS

Anexo 4 Pantalla que muestra el vínculo para acceder al Sistema de Teleconsulta	117
Anexo 5 Pantalla principal del Sistema de Teleconsulta	118
Anexo 6 Pantalla que muestra todas las acciones que hace el médico en relación con los contactos.	119
Anexo 7 Pantalla que muestra todas las acciones del médico	120
Anexo 8 Pantalla que muestran la opción de agregar contacto.....	120
Anexo 9 Pantalla que muestra la opción de ver datos adicionales del contacto.....	121
Anexo 10 Pantalla que muestra la opción de ver datos adicionales del contacto.....	121
Anexo 11 Pantalla que muestra la opción de agregar grupo	122
Anexo 12 Pantalla que muestra la opción de renombrar grupo	122
Anexo 13 Pantalla que muestra la opción de modificar la información personal del médico	122
Anexo 14 Pantalla que muestra la opción de modificar otros datos del médico	123
Anexo 15 Pantalla que muestra la opción de cambiar el avatar del médico	123
Anexo 16 Pantalla que muestra los mensajes entre médico-contacto.....	124
Anexo 17 Pantalla que muestra el historial de mensajes de la teleconsulta según el usuario.	124
Anexo 18 Pantalla que muestra la pizarra compartida.	125
Anexo 15 Pantalla que muestra la imagen en la pizarra compartida y un chat.....	126
GLOSARIO DE TERMINOS	127

INTRODUCCIÓN

A lo largo de 50 años de Revolución, se han formado cifras extraordinarias de profesionales de la salud. Los cuales han sido protagonistas de la gigantesca obra que el país ha realizado en esta esfera, lo que ha permitido desarrollar un alto nivel intelectual y cultural en cada especialista. Así como crear cuantiosos centros de atención a la salud y mantener una atención adecuada de la población que posibilite defender la igualdad de derechos.

A pesar de que Cuba es un país bloqueado, la calidad de vida es superior a muchas naciones desarrolladas que disponen de tecnologías y cuantiosos recursos. El desarrollo de la informática y las telecomunicaciones durante todos estos años en la isla ha elevado la necesidad de la informatización de las actividades en la medicina.

En la actualidad, el Ministerio de Salud Pública (MINSAP) es el organismo rector del Sistema Nacional de Salud (SNS), este se encarga de dirigir, ejecutar y controlar la aplicación de la política que el estado y el gobierno ha establecido que debe seguir el sistema de salud cubano [1].

El Ministerio de Salud Pública, ha definido un grupo de premisas y requisitos para informatizar el proceso de gestión de su información. Esto ha posibilitado un avance tecnológico en la medicina, lo que permite entrecruzar los caminos de la ciencia y abre paso a los sistemas de información en la atención al paciente con tecnologías de avanzada, por ejemplo la telemedicina, que constituye un servicio informático que posibilita el intercambio de opiniones médicas entre:

- Profesionales de la salud.
- Médicos y pacientes.
- Médicos y estudiantes de medicina.

El país comenzó a utilizar este servicio a partir de 1970 con la transmisión de señales a través del teléfono o radio, para facilitar la atención médica del país. El mismo está compuesto por diferentes ramas, las cuales son:

INTRODUCCIÓN

- El Telediagnóstico.
- La Teleconferencia.
- La Teleradiología.
- La Telepresencia.
- La Telecirugía.
- La Teleconsulta.

La Teleconsulta es la interacción compartida de imágenes e información médica, donde el diagnóstico primario es realizado por el doctor en la locación del paciente. El propósito es proveer una segunda opinión por un especialista remoto para confirmar el diagnóstico o para ayudar al médico local a llegar a un diagnóstico correcto.

Actualmente existen muchos países que brindan servicios de teleconsulta como: biopsias, diálisis peritoneales, monitorización de pacientes, videoconferencias, asistencia periódica a embarazadas, consultas a distancia, intercambio de opinión entre especialistas para emitir un diagnóstico, consultas especializadas por teléfono, vía correo o por chat.

Todos estos servicios tienen tecnología de avanzada, algunos utilizan los estándares HL7¹, CEN/TC-251², DICOM³, IHE⁴ y para solicitar la gran mayoría de ellos, hay que pagar una determinada suma de dinero o el intercambio es con una hora o día específico. En la actualidad, Cuba se sitúa en la delantera,

¹ **HL7:** Es una organización internacional, iniciada en los Estados Unidos en 1987, que pretende promover el desarrollo y evolución del estándar HL7 (*Nivel de Salud 7*) para el formato de datos e intercambio de información entre diferentes Sistemas de Información de Salud.

² **CEN/TC 251 (CEN Comité Técnico 251):** es un grupo de trabajo que junto a la Unión Europea trabajan en la estandarización en el campo de la Información de la salud.

³ **DICOM** (Imágenes digitales y comunicación en la Medicina): es un estándar para el almacenamiento, transmisión y comunicación de imágenes médicas.

⁴ **IHE** (Integración de Empresas para el cuidado de la salud): es una iniciativa de empresas y profesionales de la sanidad cuya finalidad es mejorar la comunicación entre los distintos sistemas de información sanitarios.

INTRODUCCIÓN

siendo unos de los países subdesarrollados con gran avance científico-técnico a nivel mundial, cuenta con la tecnología necesaria en cada centro de salud del país y no se ha quedado atrás en la utilización de la Teleconsulta, siendo ejemplo de ello los servicios que ofrecen el SIGEMEC y la clínica virtual.

- **SIGEMEC:** Constituye la plataforma telemática de la Red Nacional de Genética Médica, disponible como red científica en la RED INFOMED del Ministerio de Salud Pública e incluye facilidades para teleconsultas médicas en tiempo real. [2]
- **Clínica virtual:** Perteneciente a la Universidad virtual de salud que ofrece 4 servicios: brinda discusiones diagnósticas, preguntas a expertos, temas de actualización clínica e interconsultas.[3]

En Cuba, los profesionales de la salud para comunicarse entre sí, utilizan algunas vías de intercambio como: persona-persona, teléfono, correo electrónico y a través de la clínica virtual cubana. El intercambio persona-persona es muy efectivo, pues la comunicación es directa, pero se pueden encontrar en situaciones en el que se necesite la presencia de otros expertos para intercambiar opiniones.

Por teléfono, las imágenes médicas no pueden ser evaluadas, ya que no hay elementos visuales para una percepción clara. El intercambio por correo electrónico no es eficiente, ya que pueden existir problemas en la red, lo que retrasa la recepción del mensaje. Los archivos pueden ser muy grandes, lo que imposibilitaría su envío o puede existir incompatibilidad en la tecnología de ambos lados, lo que impediría la traducción del mensaje.

El servicio que ofrece la clínica virtual cubana para teleconsultas, no cumple con todas las expectativas, la comunicación entre los médicos es a través de mensajes enviados a especialistas en una determinada rama y la respuesta puede tardar, ya que depende del momento en que él revise su buzón y del envío de la respuesta.

La red médica cubana brinda servicios donde los médicos pueden actualizarse, realizar preguntas o simplemente opinar acerca de un tema en específico, pero no cuenta con un servicio que posibilite el intercambio entre los profesionales de la salud en tiempo real, por lo que atenta contra la evaluación colectiva de los casos médicos más difíciles que se presentan a diario, la rapidez del diagnóstico así como el intercambio de los médicos más experimentados con los de menos experiencia.

INTRODUCCIÓN

La Universidad de las Ciencias Informáticas, a raíz de la informatización del país, se ha trazado varios proyectos para ayudar en esta gran tarea. Específicamente a la Facultad 7, se le ha encomendado la misión de informatizar el sector de salud. El área temática de sistemas especializados perteneciente a esta facultad, tiene dentro de sus muchos proyectos, el proyecto Teleconsulta, el cual se ideó con el propósito de crear un espacio virtual entre profesionales de la salud.

Por todo lo anterior se plantea como problema a resolver: ¿Cómo facilitar la comunicación en tiempo real del personal médico del Sistema Nacional de Salud? Donde el objeto de estudio es el proceso de comunicación en tiempo real del personal médico del Sistema Nacional de Salud.

El campo de acción se enfoca en el proceso de comunicación en tiempo real del personal médico del Sistema Nacional de Salud para consultas o satisfacción de dudas médicas en los hospitales de Cuba que tengan el Sistema de Gestión Hospitalaria (HIS).

Se define como objetivo de la investigación: Implementar un sistema informático de tipo WEB que permita facilitar el proceso de comunicación en tiempo real del personal médico del Sistema Nacional de Salud para consultas o satisfacción de dudas médicas, en los hospitales de Cuba que tengan el Sistema de Gestión Hospitalaria.

Como tareas de la investigación propone:

- Valorar la situación actual de los sistemas de teleconsulta existentes en el mundo.
- Analizar la metodología y herramientas a utilizar para el modelado del sistema informático así como la técnica de programación, lenguaje, plataforma y librerías propuestas por el grupo de arquitectura de la facultad.
- Analizar las necesidades de funcionamiento de la aplicación, describiendo la Especificación de Requisitos de software.
- Realizar el prototipo no funcional del sistema.
- Analizar y diseñar el sistema informático utilizando la metodología escogida.

INTRODUCCIÓN

- Obtener Modelo de Implementación y los artefactos necesarios que describan la Base de Datos.

Como resultado de la investigación se ha obtenido el diseño e implementación de un chat para intercambio de opiniones médicas y de una pizarra compartida capaz de soportar, al menos, imágenes JPG; y el diseño del esquema de la BD del Sistema de Teleconsulta.

El presente trabajo está estructurado en cuatro capítulos, que incluyen todo lo relacionado con la investigación realizada, el diseño e implementación del servicio en cuestión. Los mismos se describen a continuación:

- **Capítulo 1. Fundamentación teórica:** Se hace un estudio del arte de los diferentes productos informáticos existentes en el mundo que aborda el tema de teleconsulta, se definen algunos conceptos importantes, se exponen las tendencias, técnicas, tecnologías, metodología y los lenguajes de programación existentes usados para su desarrollo.
- **Capítulo 2. Características del sistema:** Se expone el objeto de estudio, procesos del dominio y representación del modelo del dominio de las teleconsultas y se argumentan los requisitos no funcionales y funcionales.
- **Capítulo 3. Análisis y diseño del sistema:** Hace un análisis de la arquitectura, de posibles implementaciones, componentes o módulos ya existentes y que puedan ser rehusados y se explican las estrategias de integración.
- **Capítulo 4. Implementación:** Se muestra los modelos de datos, de despliegue y componentes utilizados a través de sus diagramas.

Capítulo I: Fundamentación Teórica

El presente capítulo expone el marco teórico y conceptual asociado a la problemática a resolver, aborda los antecedentes de los sistemas de teleconsultas en tiempo real, se mencionan algunos sistemas automatizados en el ámbito internacional y nacional utilizado para este tipo de servicio médico. Se describen las tecnologías, herramientas, lenguajes y técnicas a utilizar en el desarrollo del problema propuesto.

1.1 Definiendo las Teleconsultas

La medicina trata en el establecimiento del diagnóstico una combinación de datos, de información, que en su mayor parte es susceptible de ser transportada a distancia. El diagnóstico basado en esta información puede realizarse a miles de kilómetros de la localización del paciente.

Desde que los avances técnicos han permitido la transferencia de información electrónica, se ha planteado la posibilidad de realizar diagnósticos remotos, y de recibir consejo médico a distancia. Este avance tecnológico es lo que se conoce hoy en día como telemedicina, que tiene como objetivo prestación de servicios de medicina a distancia.

Para su implementación se emplean usualmente tecnologías de la información y las comunicaciones. La telemedicina puede ser tan simple como dos profesionales de la salud discutiendo un caso por teléfono hasta la utilización de avanzada tecnología en comunicaciones e informática para realizar consultas, diagnósticos y hasta cirugías a distancia y en tiempo real.

En la actualidad, dentro del campo de la Telemedicina, se puede encontrar que la Telemedicina se usa básicamente en dos áreas de trabajo: la práctica y la instrucción, por lo que las teleconsultas entrarían en la parte práctica.

Las teleconsultas incluyen tanto el diagnóstico como el tratamiento y la educación médica. Son un recurso tecnológico que posibilita la optimización de los servicios de atención en salud, ahorra tiempo y dinero y facilita el acceso a zonas distantes para tener atención de especialistas. Otra de las utilidades que presta es el uso de la transmisión de datos médicos para la formación de los profesionales, donde los

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

estudiantes de la salud pueden aprender con el apoyo virtual del profesor y con la presencia física del paciente.

La teleconsulta brinda algunos servicios como:

- Servicios complementarios e instantáneos a la atención de un especialista (obtención de una segunda opinión).
- Diagnósticos inmediatos por parte de un médico especialista en un área determinada.
- Educación vía web de alumnos de las escuelas de enfermería y medicina.
- Servicios de archivo digital de exámenes radiológicos, ecografías y otros.

1.2 Antecedentes a nivel internacional

La Telemedicina surgió por parte del Departamento de Defensa Americano, en los años de la Segunda Guerra Mundial, con el propósito de servir de medio de soporte para mejorar la asistencia médica a los soldados que se encontraban alejados en misiones militares. A partir de 1950, Holter⁵, Gengerelli⁶ y Glasscock⁷, investigan la obtención de parámetros biológicos “sin tocar al hombre” y consiguen recibir por radio el electrocardiograma de personas que deambulaban por la calle a considerable distancia de la estación receptora.

En 1959 se logra transmitir por primera vez imágenes radiológicas a través de la línea telefónica. Ese mismo año, médicos de la Nebraska University utilizaron por primera vez un sistema de televisión interactiva para transmitir exploraciones neurológicas a los estudiantes a través del campus y se comienzan a utilizar en Canadá sistemas para la transmisión de imágenes entre radiólogos. En 1965 se transmiten electrocardiogramas y radiografías desde barcos en alta mar. Aquellas primeras experiencias

⁵ **Holter:** Norman J. Holter, Doctor en Ciencias Físicas y Químicas, inició a mediados del siglo pasado junto a los doctores J.A. Gengerelli y WR.Glasscock, la apasionante aventura científica de “explorar al hombre sin tocarlo”, lo que felizmente permitió años más tarde, el registro ambulatorio del electrocardiograma, mundialmente conocido como Holter.

⁶ **Gengerelli:** Doctor que junto a Norman J. Holter y WR.Glasscock investigaron como registrar el electrocardiograma.

⁷ **Glasscock:** Doctor que junto a Norman J. Holter y WR.Glasscock investigaron como registrar el electrocardiograma.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

lograban superar barreras geográficas y asistencia sanitaria a zonas aisladas o rurales con escasos recursos asistenciales, pero en general no superaron la fase de investigación por el gran coste de los equipos y de las telecomunicaciones.

Posteriormente, tras un período de oscurecimiento, el interés por la Telemedicina resurgió en la década de los noventa, cuando el gobierno estadounidense incluye este concepto en su proyecto “Infraestructura Nacional de Información” (1993); para entonces, las Tecnologías de Información y Comunicación (TIC), experimentaban un notable desarrollo y su coste descendía.

Actualmente grupos de trabajo de todo el mundo tratan de utilizar las nuevas tecnologías, muchas de ellas fruto de la carrera espacial, en todos los campos de la medicina. Como consecuencia de este esfuerzo la telemedicina ha iniciado un período de crecimiento explosivo paralelo al que experimenta Internet, favorecido por la caída continua del coste de la tecnología y el interés de la sociedad en todo lo relacionado con la red. La práctica de la medicina asentada sobre la tecnología Internet ha dado ya lugar a la utilización de un nuevo término de rápido éxito: e-Salud⁸.

1.3 Antecedentes en Cuba

En Cuba, el uso de la telemedicina, surge con la transmisión de señales a través del teléfono o radio en la década de 1970, donde su propósito principal es obtener mayor calidad del diagnóstico. Entre los beneficios fundamentales que brindaría se encuentra: realizar una consulta de segunda opinión y brindar una atención adecuada a la población, todo esto de acorde al ámbito internacional. Entre las líneas de la Telemedicina en que ha trabajado fuertemente el país está el telediagnóstico, dentro de éste, el correspondiente a la imagenología⁹, la telepatología¹⁰ y otras que han sido temas de especial importancia en los que han participado prestigiosos profesionales.

⁸ **E-Salud:** Aplicación de las Tecnologías de Información y Comunicación en el amplio rango de aspectos que afectan el cuidado de la salud, desde el diagnóstico hasta el seguimiento de los pacientes.

⁹ **Imagenología:** Está formada por el conjunto de imágenes médicas, para el estudio más profundo de un caso médico.

¹⁰ **Telepatología:** es la transmisión de imágenes digitales de anatomía patológica por sistemas de telecomunicación, y con fines de consulta, diagnóstico, investigación, o docencia.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

A partir de 1998, se decide por el Ministerio de Salud Pública, abordar de conjunto con el grupo de la electrónica del SIME¹¹; la implementación de una Red de Telediagnóstico para el Sistema Nacional de Salud. Esta estaría soportada en la Red Telemática de la Salud: INFOMED, la cual brinda el soporte necesario de telecomunicaciones para el tráfico de la información, fue implementado con el Sistema PATRIS¹², producidos por EICISOFT¹³.

Desde este momento queda lejos el primitivo concepto de Telemedicina, circunscrito a la utilización de servicios de telediagnóstico y teleasistencia para eludir barreras geográficas. Ahora, su entorno conceptual se ha expandido donde las redes de comunicación y sistemas integrados de información, permiten el intercambio de información sanitaria a distancia para actividades asistenciales como: prevención, formación, investigación y evaluación, lo que abre el camino a unas de las ramas mas importantes para la medicina: la Teleconsulta.

1.4 Conceptos Básicos Relacionados con el Problema Planteado.

Telemedicina: Distribución de servicios de salud, en el que la distancia es un factor crítico, donde los profesionales de la salud usan información y tecnología de comunicaciones para el intercambio de información válida para el diagnóstico, tratamiento y prevención de enfermedades o daños, investigación y evaluación; y para la educación continuada de los proveedores de salud pública, todo ello en interés del desarrollo de la salud del individuo y su comunidad. (OMS 1998)

Teleconsulta: Consulta a distancia entre médicos para contrastar diagnósticos, entre paciente y médico o en algunos casos entre paciente y sistema autónomo.

¹¹ **SIME:** El Ministerio de la Industria Sideromecánica encargado de dirigir, ejecutar y controlar la política del Estado y el Gobierno en cuanto a las actividades de la industria Siderúrgica, Mecánica y de Reciclaje.

¹² **PATRIS:** Sistema Intrahospitalario de Imagenología y Telemedicina, es un modelo de PACS sobre computadoras personales para la automatización de los departamentos de imagenología y la telemedicina.

¹³ **EICISOFT:** Empresa de robótica en Cuba.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

1.5 Sistemas automatizados o Servicios de teleconsulta online existentes.

Los sistemas automatizados de teleconsulta se utilizan con el fin de disminuir el tiempo entre la aplicación de exámenes y la obtención de resultados. Todo esto contribuye a que el paciente no viaje largas distancias, que no tenga que ir a examinarse y la reducción de costos de tiempo y dinero.

Sistemas de Teleconsulta a nivel internacional

- **Doctor Chat®(Bogotá)**

El servicio fue diseñado por la División de Educación de la fundación Santa Fé de Bogotá a través de su programa e-Salud®. Cuenta con la asesoría médica por Internet abierto a toda la comunidad, de manera gratuita, las teleconsultas pueden ser de dos formas.

- **Formulando una pregunta:** Este servicio se solicita a través de un mensaje, el cual será respondido por un especialista en cualquier momento a través del servicio asincrónico de teleconsultas, el cual dará la respuesta de 3 a 5 días por vía correo, publicándola en el sitio o ambos.
- **En tiempo real:** Un invitado resolverá dudas en vivo, en un tema seleccionado, todos los Jueves de 1:00 pm a 2:00 pm.

Doctor Chat® es un servicio que ofrece internet donde brinda teleconsulta online gratuita, pero este servicio no ofrece un intercambio en tiempo real. Este no resuelve el problema de comunicación entre profesionales con el objetivo de realizar valoraciones en conjunto, tampoco permite visualizar imágenes médicas en pizarras compartidas y no posibilita de que un paciente sea consultado de forma online por un médico.

- **Sitio web Apoyomédico**

Este sitio ofrece la posibilidad de intercambiar de forma privada y gratuita con el médico en línea con respecto a alguna duda o necesidad de información sobre algún tema relacionado con la salud y la medicina. Las consultas y sus respuestas se publican de forma anónima y sirve como información

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

referencial o educativa al visitante. Las preguntas se resuelven a manera de opinión médica y su finalidad es de tipo educativa o informativa y de ninguna manera reemplaza a una consulta médica normal, pues para ello, se requiere de la interacción directa entre el médico y el paciente.

El sitio web Apoyomédico a pesar de sus variadas ofertas, no posibilita el intercambio entre profesionales de la salud de forma online. En ocasiones, existe la posibilidad que los profesionales no estén en línea directa para responderles su petición, por tanto, no cumple con los objetivos trazados.

- **Sitio web Soymédico (Bolivia)**

El servicio que ofrece Soymédico para médicos de Bolivia, a través de la tecnología web tagboard (herramienta para crear chat), es para realizar consultas a través de un chat donde un especialista atiende las solicitudes del paciente. En este caso los especialistas están preparados en ginecología y sexología, este sitio se hizo con el propósito de que no hubiera embarazos no deseados.

El Servicio de chat Soymédico tiene la desventaja que es solamente para médicos de Bolivia. El tema que se debate principalmente es acerca de los embarazos y la mujer. Las conversaciones pueden ser vistas por cualquier usuario, lo cual no brinda privacidad y confidencialidad a las consultas que se realicen en línea y estas son realizadas con descuentos, es decir que hay que pagarlas; por tanto, no cumple con los requisitos establecidos.

- **VistaMédica(Argentina)**

El portal VistaMédica es un servicio para médicos donde ofrece información científica, noticias médicas, artículos, chat, novedades de la medicina, clasificados en salud, correo gratis y más. Es un sitio de cooperación libre entre sus integrantes, cuyo objetivo es ofrecer servicios y contenidos para profesionales de la salud de habla hispana, con el propósito de lograr un mayor desarrollo científico y humano. Es independiente de cualquier laboratorio farmacéutico, aseguradora, hospital o cualquier otra empresa. El servicio de chat para los médicos se activa a las 8:00 todos los días para que se comuniquen y expongan sus ideas.

El portal VistaMédica es bastante amplio en su contenido de la salud y útil para todo profesional de la medicina ya que puede documentarse y actualizarse constantemente. El servicio de chat que ofrece, para

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

la comunicación en línea no está moderado, por tanto está propenso a no ofrecer un servicio con la calidad médica requerida. Tiene el inconveniente que para ingresar a él tiene que ser a través de publicidad, donaciones, uso de internet gratis, y cuentas de correo Premium, debido a estas características, no cumple con los requisitos establecidos.

- **Health & Doctor**

Health & Doctor es un canal de atención médica directa que utiliza tecnología de punta a favor de la salud, complementa la medicina tradicional y ofrece un valor añadido de inmediatez y ruptura de barreras geográficas en la asistencia médica. Brinda variados servicios de atención médica como: consultas médicas en línea, emisión de recetas digitales, solicitud de exámenes de laboratorios vía internet, emisión de informes médicos, Telemedicina e información de clínicas y hospitales.

El servicio que ofrece Health & Doctor, es de calidad, ya que utiliza modernas tecnologías que hoy día son parte de la vida diaria para llegar a los pacientes, sin importar el lugar del mundo donde vivan. A pesar de sus variadas ofertas de consultoría y atención a la salud, para beneficiarse de ellos hay que afiliarse por medio de pago de tarjetas magnéticas, lo que lo convierte en un servicio solamente para las personas que puedan pagar. Este servicio no corresponde con el sistema de Cuba, donde es totalmente gratuita la atención médica, por tanto no cumple con las características requeridas.

- **El médico en casa**

El sitio web El médico en casa ofrece consultas online e información sobre lo último en salud. Esto se realiza por medio de consejos y recomendaciones a todos aquellos usuarios y/o pacientes que lo necesiten. El servicio de consultas médicas online es de pago obligatorio y su precio es de 2 €, a realizar tras rellenar el formulario de consulta vía Paypal y se puede adjuntar un archivo que pueda servir de ayuda (analítica, radiografía, fotografía...). La respuesta se envía en menos de 48-72 horas.

El servicio que ofrece el sitio web para los pacientes o cualquier persona es a través de un mensaje que se envía al grupo de expertos médicos, donde en un período de tiempo es respondido. Tiene la desventaja de que para solicitarlo hay que pagar y esperar un tiempo determinado para recibir la respuesta por no ser un servicio online. Este sitio no corresponde con las características del sistema cubano por lo que no lo convierte en un candidato de acuerdo a las necesidades.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

Sistemas de Teleconsulta a nivel nacional

- **Sistema Informático de Genética Médica**

El Sistema Informático de Genética Médica (SIGM), es un sistema realizado por la Facultad 6, de la Universidad de las Ciencias Informáticas, que actualmente se encuentra en despliegue en la Red de Infomed y tiene como objetivo gestionar toda la información asociada a una consulta de genética médica. Es una aplicación web que cuenta con 7 registros, entre los que se encuentra el módulo de Teleconsulta Genética para la discusión a distancia de casos de los que no se tiene un diagnóstico certero.

El módulo de Teleconsulta Genética, permite controlar las consultas entre el Centro Nacional de Genética Médica y los diferentes centros de Genética del país, lo que permite gestionar las solicitudes de los profesionales de la genética en el país. Todo esto se realiza a través de una solicitud que realiza el genetista, debido a una duda que surge en un caso. Esta solicitud se llena a través de un formulario que contiene el sistema, con los datos del paciente, la fundamentación del caso, una foto (si quiere), los participantes que quiere tener en la teleconsulta y otros aspectos importantes para obtener el diagnóstico.

Esta es enviada a un administrador según el nivel de salud, este establece una hora de inicio y fin y se envía una citación a través del correo a los genetistas previamente escogidos y aprobados. La comunicación se realiza a través de un chat con ayuda de un moderador, que es el que comienza y termina la discusión, además de que en caso que la discusión diagnóstica no es suficiente, envía la solicitud a un nivel superior para darle solución a la problemática.

El Sistema Informático de Genética Médica, es una herramienta útil para la Red de Genética Médica, ya que integra varios registros de estudios realizados sobre enfermedades raras y de difícil solución y todos forman parte del sistema y el servicio de intercambio online, que ofrece el módulo de Teleconsulta Genética, es una vía automatizada para lograr un diagnóstico certero de un caso de enfermedad genética a distancia.

A pesar de todas estas ventajas que ofrece, el sistema está destinado a todos los Centros de Genética del país y no corresponde con el negocio propuesto por esta investigación, ya que se centra fundamentalmente en diagnosticar enfermedades genéticas. Los usuarios que intercambian sus opiniones diagnósticas a través del chat, son esencialmente genetistas, lo que no se relaciona con las

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

características del sistema a desarrollar, él cual va dirigido a todos los especialistas en cualquier rama de la salud.

1.6 Tendencias y tecnologías actuales utilizadas

1.6.1 Metodologías de Desarrollo

Metodologías de Desarrollo: Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software. [4]

Las metodologías de desarrollo son importantes para el desarrollo del software ya que ofrecen documentación formal. Estas son útiles para garantizar la eficacia según los requisitos iniciales y la eficiencia al minimizar las pérdidas de tiempo en el proceso de generación de software.

- **Metodología de desarrollo del software ágil**

- **Rational Unified Process (RUP)**

Proceso para el desarrollo de un proyecto de un software que define claramente quién, cómo, cuándo y qué debe hacerse en el proyecto. [5]

RUP es una metodología ágil que se utilizó porque sirve de guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo. Describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica de acuerdo a los requerimientos y a la arquitectura.

Se caracteriza por tener *4 fases* de desarrollo de software (Inicio, Elaboración, Construcción y Trasmisión) ,*6 disciplinas o flujos de ingeniería* (Ingeniería de Negocios, Requerimientos, Análisis y Diseño, Implementación, Pruebas) y *3 flujos de apoyo* (Configuración y administración del cambio, Administración de proyecto y Ambiente).

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

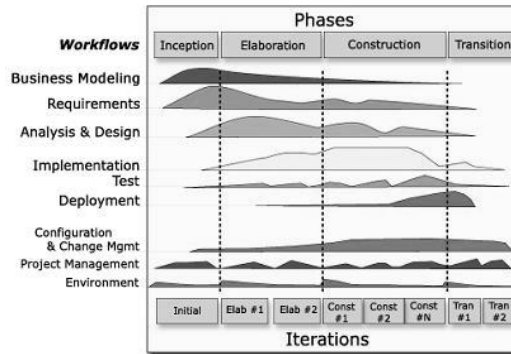


Figura 1 Fases del RUP

1.6.2 Tecnologías de desarrollo

- **Capa de Presentación**

Es la que presenta el sistema al usuario, le comunica la información y captura la que este introduce en un mínimo de procesos. Esta capa se comunica únicamente con la capa de negocio.

- **Facelets**

Es un framework ligero que permite el uso de plantillas en aplicaciones JSF. [6] Este framework ofrece la posibilidad de la construcción de interfaces basadas en plantillas, la rápida creación de componentes por composición y fácil creación de funciones y librerías de componentes.

- **Java Server Faces (JSF).**

Es un framework que facilita y agiliza el diseño de interfaces de usuario. Permite implementar una serie de componentes, estado de los mismos, eventos del lado de servidor y otras.

- **Ajax4JSF**

Es una librería open source (código abierto) que se integra totalmente en la arquitectura de JSF. Se extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax, de forma limpia y sin añadir código Javascript.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

Presenta mejoras sobre los propios beneficios del framework JSF incluyendo el ciclo de vida, validaciones, facilidades de conversión y el manejo de recursos estáticos y dinámicos. Permite definir un evento en una página que invoca una petición Ajax y luego las áreas de la página deberían sincronizarse con el Árbol de Componentes JSF después de que la petición Ajax cambie los datos en el servidor.

El uso de Ajax4jsf aporta al programador una gran ventaja: la utilización de la funcionalidad AJAX dentro de páginas JSF sin la necesidad de crear nuevo código Javascript.

Características:

- Intensifica el conjunto de beneficios de JSF mientras se trabaja con AJAX.
- Añade funcionalidad AJAX a las actuales aplicaciones JSF.
- Escribe componentes propios con soporte built-in (incorporado) para AJAX.
- Dispone de un paquete de recursos con las clases de la aplicación Java.
- Proporciona un avanzado apoyo a la gestión de diferentes recursos: imágenes, código JavaScript y hojas de estilo CSS.
- Brinda una función de “skinnability” que permite definir y administrar fácilmente diferentes esquemas otros parámetros de la interfaz de usuario, con la ayuda de los parámetros denominados en el skin.
- Soporte Ajax para páginas con componentes JSF que tienen comunicación asíncrona y actualizaciones de porciones de página.
- La capacidad de habilitar Ajax a componentes JSF sin cambios en el propio componente.
- Dispone de una arquitectura abierta y soporte para los estándares en la industria para mezclar componentes que son de distintas librerías.
- Trabaja en el lado del servidor.
- Soporta Facelets.
- Total integración con Tomahawk y Trinidad. [7]

➤ **Richfaces**

Es un framework de código abierto para Java y constituye una biblioteca de componentes. Contiene un fuerte apoyo para la skinnability de aplicaciones JSF y aprovecha al máximo los beneficios de JSF

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

framework. Incluye la validación y conversión de instalaciones, junto con la gestión de estática y dinámica los recursos.

- **Capa de Negocio**

- **JBossSeam**

Es un poderoso framework de código abierto desarrollado para la plataforma Java, para construir aplicaciones en Internet. [8]

Este tipo de framework tiene la ventaja de ser de código abierto, al mismo tiempo integra la capa de presentación con la capa de negocios, maneja directamente la lógica de la aplicación y de negocio. Tiene como objetivo simplificar la arquitectura de las aplicaciones, lo que permite integrar tecnologías de forma relativamente transparente y con herramientas de generación de código. Todo esto posibilita que se utilice como el framework para la implementación del software.

- **JBPM**

Es una plataforma de workflow (flujo de trabajo) y automatización de procesos que habilita la coordinación entre aplicaciones y servicios desintegrados. JBPM combina la facilidad de desarrollo de aplicaciones workflow, con un motor flexible y escalable para la ejecución de transacciones de negocio. [9]

- **Drools**

Son reglas de negocio manejadas por un motor de reglas de aplicación en un sistema. El motor de reglas se encuentra implementado por ReteOO, que es una base del algoritmo Rete de Charles Forgy adaptados para el lenguaje Java.

Es lo suficientemente flexible como para que coincida con la semántica del dominio de su problema en determinados idiomas. Provee herramientas de edición gráfica, herramientas basadas en Web y desarrolladores de herramientas de productividad.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

Características:

- **Código abierto:** Tiene licencia de fuente abierta que le hace libre para descargar, usar, integrar y distribuir.
- **Programación Declarativa:** Permite decir “que hacer” y no decir “como hacerlo”.
- **Accesibles reglas de negocio:** Es amigable tanto a desarrolladores como administradores de las reglas de negocio. Permite a los desarrolladores a escribir casi todas las reglas de negocio como un lenguaje natural. Lógica y los datos por separado
- **Basado en Web:** Basado en AJAX, construye desde cero hasta avanzadas reglas de aplicación, control de versiones, y de administración. [10]

- **Capa de Acceso a Datos**

Contiene clases que interactúan con la base de datos. Estas clases altamente especializadas permiten, utilizar los procedimientos almacenados (funciones para interactuar con la base de datos) generados y realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.

- **Mapeo de Objetos Relacional (ORM): Hibernate**

Es una capa de persistencia objeto/relacional y un generador de sentencias sql. [11]

Esta herramienta es útil ya que representa el motor de persistencia que ayuda a la programación en la plataforma Java, facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones. Es poderoso, tiene alto rendimiento y permite desarrollar clases persistentes en el lenguaje orientado a objetos, incluyendo asociación, herencia, polimorfismo, composición, y las colecciones.

- **EJB 3**

Los Enterprise JavaBeans (también conocidos por sus siglas EJB) son una de las API que forman parte del estándar de construcción de aplicaciones empresariales J2EE de Sun Microsystems (ahora JEE 5.0). Su especificación detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor que son, precisamente, los EJB.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

Los EJB proporcionan un modelo distribuido y estándar de componentes que se ejecutan en el servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad y otros); para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

➤ **JPA: Java Persistence API**

Es una especificación de Sun Microsystems para la persistencia de objetos Java a cualquier base de datos relacional. Esta API fue desarrollada para la plataforma JEE e incluida en el estándar de EJB 3.0, su objetivo es no perder las ventajas de la orientación a objetos al interactuar con una base de datos.

Tecnologías horizontales

➤ **JavaEE 5**

Java Platform, Enterprise Edition o Java versión 5, es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java. Se caracteriza por tener una arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares, ejecutándose sobre un servidor de aplicaciones.

➤ **JRE 6**

Es el acrónimo de Java Runtime Environment (entorno en tiempo de ejecución Java) y se corresponde con un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas. El JRE está constituido por una máquina virtual Java y además por las librerías de clases estándar que implementan el API de Java.

La funcionalidad que ofrece JRE de intermediario entre una aplicación programada en Java y el sistema operativo que se esté usando. Sirve para que cualquier aplicación pueda funcionar en cualquier sistema operativo, lo cual es significativo para el desarrollo del software en Java.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

Herramienta para el desarrollo en Java

➤ JDK

Diseñado para programar aplicaciones de software, utiliza principalmente el lenguaje de programación Java. Es un editor especializado en código Java para poder trabajar sin problemas, constituye un soporte para la carga de aplicaciones en la máquina virtual de Java y para línea de comandos, posibilita compilar y ejecutar las aplicaciones para probar si responde a las expectativas.

Servidores

Es un tipo de software cuyo propósito es proveer datos de modo que los usuarios puedan utilizar esos datos.

Servidor de Aplicación

Es el tipo de software que conecta dos aplicaciones, ocupa una gran parte del territorio entre los servidores de bases de datos y el usuario, y a menudo los conectan.

- **Servidor de Aplicación**

- **JBoss Server o JBoss AS**

Se puede utilizar en cualquier sistema operativo que soporta Java para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java, incluyendo servicios adicionales como clustering, caching y persistencia. Es ideal para aplicaciones Java y aplicaciones basadas en la web. Lo cual lo convierte en el servidor de aplicación necesario para el desarrollo del software.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

1.1.3 Lenguajes

- **Lenguaje de modelado**

- **UML (Lenguaje Unificado de Modelado)**

Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. [12]

Su utilización es independiente del lenguaje de programación y de las características de los proyectos. Se diseñó por la necesidad de modelar cualquier tipo de proyecto, tanto informático como de arquitectura, o de cualquier otra rama. Posibilita la modificación de todos sus miembros mediante estereotipos y restricciones.

Características:

- Es una especificación basada en Booch, OMT y OOSE, de allí sus principios.
- Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo.
- Permite describir un sistema en diferentes niveles de abstracción.[13]

- **Lenguaje de programación**

- **Java**

Es una plataforma virtual de software desarrollada por Sun Microsystems, de tal manera que los programas creados en ella puedan ejecutarse sin cambios en diferentes tipos de arquitecturas y dispositivos computacionales. [14]

Esta herramienta de programación por su característica de ser multiplataforma, le posibilita funcionar en diversos tipos de ordenadores y sistemas operativos así como en otros dispositivos inteligentes. Las aplicaciones que se crean con ella son independientes, es "orientado a objetos", lo que facilita que los programas se construyan a partir de módulos independientes, y que estos módulos se pueden transformar o ampliar fácilmente.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

Características:

- Es un lenguaje sencillo para escribir applets y resulta muy fácil su aprendizaje.
- Es orientado a objetos, lo que permite agrupar en estructuras encapsuladas tanto sus datos como los métodos o funciones que manipulan esos datos.
- Proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets, establecer y aceptar conexiones con servidores o clientes remotos, para así facilitar la creación de aplicaciones distribuidas.
- Es compilado, ya que su código fuente se ha transformado en una especie de código máquina y es interpretado, porque sus bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real.
- Fue diseñado para crear software altamente fiable, lo que proporciona a los programadores la disminución de errores (la aritmética de punteros), y la liberación explícita de memoria.
- La seguridad radica en las barreras impuestas en el lenguaje y en el sistema de ejecución en tiempo real.
- Soporta aplicaciones que pueden ser ejecutadas en cualquier entorno de red, arquitecturas distintas y con sistemas operativos diversos.
- Su indiferencia a la arquitectura representa sólo una parte de su portabilidad, especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas, estas 2 últimas características se conocen como la *Máquina Virtual Java* (JVM).
- Permite el soporte de la sincronización de múltiples hilos de ejecución a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

- El lenguaje y su sistema de ejecución en tiempo real son dinámicos en la fase de enlazado, lo que permite el enlace de nuevos módulos de código bajo demanda, procedente de fuentes muy variadas, incluso desde la Red.
- Java puede ser usado para crear dos tipos de programas: aplicaciones independientes y applets.

1.1.4 Herramientas

Herramientas CASE: conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. [15]

- **Herramientas CASE**
 - **Visual Paradigm 6.0**

Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software (análisis y diseño orientados a objetos, construcción, pruebas y despliegue). [16]

Características:

- Permite tanto la ingeniería directa como inversa, soportada en varios lenguajes de programación.
- Brinda un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Presenta un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Permite la disponibilidad de múltiples versiones para cada necesidad, capaz de integrarse con los principales IDEs.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

- **IDE de desarrollo**

- **Eclipse**

Es un proyecto de desarrollo de software de código abierto dedicado a proporcionar una plataforma industrial robusta, con amplias características y con calidad comercial para el desarrollo de herramientas altamente integradas. [17]

Está compuesto de tres subproyectos: la Plataforma Eclipse, la Java Development Tool y el Plug-in Development Environment .Posee un entorno de desarrollo integrado de código abierto multiplataforma que proporciona funcionalidades genéricas y presenta un editor visual con sintaxis coloreada.

Ofrece compilación incremental de código, un potente depurador (que permite establecer puntos de interrupción, modificar e inspeccionar valores de variables e incluso depurar código que resida en una máquina remota).Presenta un navegador de clases, un gestor de archivos y proyectos. Todas estas características lo convierten en una plataforma de herramienta universal, abierta y extensible para el desarrollo del software.

- **Modelado de la base de datos**

- **Case Studio v 2.22**

Es una herramienta profesional que se utiliza para diseñar bases de datos. Permite crear diagramas de relación, modelado de datos, gestión de estructuras, la generación rápida de diagramas gráficos de bases de datos relacionales y de scripts SQL.

Su potencia se basa en la ingeniería inversa, que permite identificar y estructurar bases de datos ya existentes para poder trabajar con ellas sin problemas. Todas estas características permiten que se utilice para la modelación de la base de datos del software.

1.1.5 Sistema Gestor de Base de Datos(SGBD)

Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurar su integridad, confidencialidad y seguridad.

CAPÍTULO I: FUNDAMENTACIÓN TEORICA

- **Sistema Gestor de Base de Datos**

- **PosGretSQL 8.3**

Es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES. [18]

Características de PostgreSQL 8.3

- Es un sistema de base de datos relacional perteneciente al ámbito del software libre. Se destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL.
- Permite aproximar los datos a un modelo objeto-relacional, es capaz de manejar complejas rutinas y reglas, tiene soporte para lenguajes procedurales internos y usa una arquitectura proceso-por-usuario cliente/servidor.
- Soporta operadores, funcionales métodos de acceso y tipos de datos definidos por el usuario y posibilita la integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos.

En este capítulo se abordaron conceptos fundamentales relacionados con los sistemas de teleconsulta. Además, se realizó un estudio sobre los principales sistemas existentes que se encuentran relacionados con las teleconsultas online y se concluyó que ninguna cumplía con las características requeridas. Se detallaron las metodologías y tecnologías a utilizar para el Sistema de Teleconsulta online, que favorecerá su diseño e implementación con la calidad requerida a favor de los resultados esperados.

Capítulo II: Características del Sistema

En el presente capítulo se describen las características del sistema a construir. Debido a la poca definición de los procesos de negocio se hace necesaria la definición de conceptos y sus relaciones, agrupados en un Modelo de Dominio. Este sirve de apoyo para la especificación de las condiciones, capacidades y cualidades que el sistema debe tener; las que se especifican en términos de Requerimientos y Casos de Uso. Esto se muestra en un Diagrama de Casos de Uso en conjunto con los actores que interactúan con los mismos.

2.1 Objeto de Automatización

2.1.1 Descripción del objeto de estudio

El objeto de estudio de esta investigación, es el proceso de comunicación en tiempo real de los especialistas de la salud, el cual se detalla a continuación. En el Sistema de Salud Pública de Cuba, la comunicación entre el personal médico se realiza a través del teléfono, clínica virtual, correo electrónico o simplemente mediante el intercambio persona-persona, lo que no posibilita un intercambio directo y rápido.

Para que la comunicación entre especialistas de la salud en Cuba se efectúe satisfactoriamente, debe de existir en cada centro de salud, un sistema online para el intercambio entre los profesionales de la medicina. Esto resultaría una forma directa de comunicación rápida y de intercambio de opinión especializada y de evaluación de imágenes instantáneamente. Brindaría un mejor diagnóstico al paciente, una gran ayuda al profesional de la salud en su formación profesional e información necesaria para futuros diagnósticos.

- **Propuesta del Sistema**

Se propone realizar un sistema para la teleconsulta online, este consistirá en un chat con pizarra compartida para la evaluación de imágenes. A este sistema accederán médicos de todas las especialidades para intercambiar opiniones y determinar un diagnóstico en conjunto con otros profesionales de la salud. Ofrecerá opciones como: añadir a otros profesionales de la salud, conversar, cambiar su estado, perfil, cargar imagen, moverla y ampliarla. Posibilitará guardar la imagen para futuros

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

diagnósticos y ver el historial de mensajes, por si quieren reevaluar el caso o simplemente para obtener información.

2.1.2 Modelo de Dominio

Es un modelo conceptual que muestra gráficamente los conceptos (clases de objetos), los atributos y las asociaciones más importantes del dominio del problema. [19]

Este tipo de modelo explica los conceptos significativos en un dominio del problema, identifica los atributos y las asociaciones, y es la herramienta más importante del análisis orientado a objetos. Representa un modelo de conceptos de objetos del dominio, no del software y estos conceptos son importantes a la hora de entender lo que se quiere representar, es decir, un diccionario visual que utiliza la notación UML de diagrama de estructura estática.

Incluye:

- Clases de objetos.
- Asociaciones entre clases de objetos.
- Atributos de las clases de objetos.

En el caso del Sistema de Teleconsulta Online se define un Modelo Conceptual al no haber información suficiente para establecer procesos de negocio, por lo que se definieron clases conceptuales relativas al dominio del problema. Esto proporcionará a los usuarios, clientes, desarrolladores y otros interesados, la ventaja de utilizar un vocabulario claro y relacionado con el sistema en cuestión y facilitará la captura de los requerimientos para darle solución al problema.

Conceptos Fundamentales

- **Teleconsulta:** Es el servicio que se ofrece para el intercambio de información especializada entre médicos y/o especialistas sobre opiniones o conocimientos de un determinado tema, donde pueden compartir imágenes e información médica

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

- **Pizarra:** Constituye un espacio electrónico compartido (pantalla), donde los médicos pueden incorporar imágenes para su evaluación, en tanto que los demás médicos pueden ver inmediatamente el resultado en sus correspondientes estaciones de trabajo.
 - **Imagen:** Es la representación visual del objeto de estudio del médico. Esta se carga en la pizarra compartida, donde se evalúa mediante una serie de acciones como dibujar, ampliar o mover para una mejor percepción del médico al cual se le está pidiendo opinión diagnóstica.
 - **Mensaje:** Es la parte fundamental en el proceso de intercambio de información médica que se envía entre médicos a través del chat, el cual constituye el medio de comunicación entre ellos.
 - **Contacto:** Representa al médico, con el cuál se establece la comunicación.
 - **Médico:** Es el que solicita información médica a través del chat para llegar a un diagnóstico en conjunto con el contacto, que constituye el otro especialista con el que entablará un intercambio en tiempo real.
 - **Perfil:** Es la información personal del médico, esta puede ser: nombre, apellidos, correo y teléfono.
- **Roles**

Representa al papel que juega el usuario y como se beneficia de sus resultados. Puede ser cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa.

➤ Usuario

Usuario	Justificación
Médico	Es el responsable de iniciar la comunicación con otro usuario. Se comunica a través de mensajes, en los cuales solicita ayuda profesional médica para mejorar el diagnóstico de un determinado paciente. Tiene la posibilidad de intercambiar información visual a través de la evaluación de imágenes que servirán de apoyo a la información que solicite.

Tabla 1 Usuarios del Dominio

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

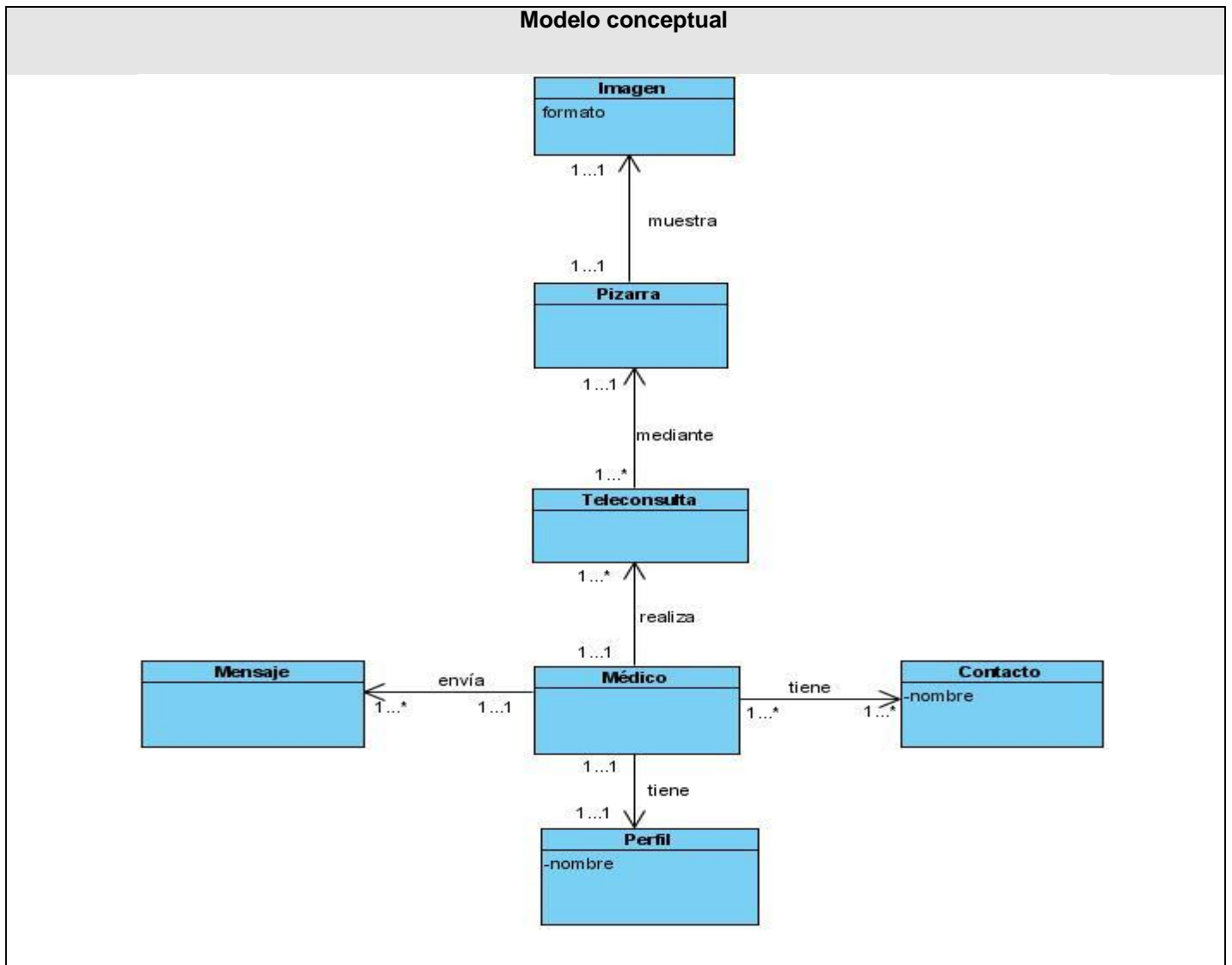


Figura 2 Modelo Conceptual

2.13 Especificación de los requisitos de software

Requerimiento: Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. (IEEE)

Los requerimientos del software son las condiciones necesarias para el desarrollo de un sistema. Estos definen el ámbito del sistema junto a una interfaz de usuario de acuerdo a lo que necesita el cliente o

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

usuario. Se establecen mediante una reunión con los clientes y los desarrolladores para saber cuál es el objetivo del sistema y son utilizados para tener un mejor entendimiento y para la planificación técnica del software.

Los requerimientos deben guiarse por 3 pasos fundamentales:

- **Panorama general:** Describe el objetivo de la creación del sistema informático, en este caso, un Sistema de Teleconsulta Online para los hospitales de Cuba
- **Meta del sistema:** Permitir el intercambio de información médica profesional con la ayuda de la evaluación conjunta de imágenes
- **Funciones del sistema:** Constituye los requerimientos funcionales.
- **Requerimientos Funcionales:**

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir sin alterar la funcionalidad del producto. [20]

Los requerimientos funcionales del Sistema de Teleconsulta Online, especifican lo que las herramientas de desarrollo deben hacer. Estos posibilitaron que se definiera el alcance del sistema teniendo en cuenta las acciones a realizar y el intercambio de datos entre sus diferentes funciones.

A continuación se listan los requerimientos funcionales del sistema propuesto:

RF1. Agregar contacto

RF2. Ver perfil.

RF 3. Mover contacto.

RF4. Eliminar contacto

RF5. Agregar grupo

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

RF6. Renombrar grupo

RF7. Eliminar grupo

RF8. Editar perfil.

RF9. Cambiar avatar

RF10. Enviar Mensaje.

RF11. Historial de Mensaje.

RF12. Borrar Historial

RF13. Compartir pizarra

RF14. Cargar Imagen.

RF15. Dibujar Imagen.

RF16. Ampliar Imagen.

RF17. Mover Imagen.

RF18. Guardar Imagen.

- **Requerimientos no Funcionales:**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. [21]

Los requerimientos no funcionales del sistema, definieron lo que la herramienta de software debe tener en cuenta en cuanto a apariencia, sensación, operabilidad y mantenimiento, para que este sea atractivo, usable, rápido y confiable. Se vinculan directamente con los funcionales y forman una parte significativa de la especificación de los requisitos.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Usabilidad: Describe los niveles apropiados de usabilidad, dados los usuarios finales del producto. Para ello debe revisarse la especificación de los perfiles de usuarios y las clasificaciones de sus niveles de experiencia.

Nº: RNF 1	
Nombre	Usabilidad
Descripción	<ul style="list-style-type: none">- El sistema proporciona un acceso fácil y rápido al usuario, con el objetivo de proveer familiarización con el uso de la misma.- El usuario debe tener un dominio mínimo de computación para utilizar sus funcionalidades.

Tabla 2 Requerimientos No Funcionales: Usabilidad

Rendimiento: Se relaciona con tiempos de respuesta estimados, requeridos y esperados para la ejecución en línea de procesos del sistema. Tiene como base la plataforma tecnológica y escenarios específicos a los que en teoría el sistema estará expuesto y frente a los que deberá responder.

Nº: RNF 2	
Nombre	Rendimiento
Descripción	<ul style="list-style-type: none">- El sistema proporcionará tiempo de respuesta aceptable para los procesos en línea.- Garantizará la velocidad de navegación para los usuarios.

Tabla 3 Requerimientos No Funcionales: Rendimiento

Soporte: Abarca todas las acciones a tomar una vez que se ha terminado el desarrollo del software. Su motivo es asistir a los clientes de este así como lograr su mejoramiento progresivo y evolución en el tiempo.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Nº: RNF 3	
Nombre	Soporte
Descripción	<ul style="list-style-type: none">- El sistema contará con una ayuda, donde el usuario encontrará información referida a como navegar.

Tabla 4 Requerimientos No Funcionales: Soporte

Portabilidad: Describe la capacidad del sistema para migrar de una plataforma a otra. Tiene en cuenta los requisitos técnicos y la configuración del sistema.

Nº: RNF 4	
Nombre	Portabilidad
Descripción	<ul style="list-style-type: none">- La ejecución sobre el sistema operativo será tanto en Windows 95 ó superior como Linux.- Garantizará compatibilidad con otros navegadores.

Tabla 5 Requerimientos No Funcionales: Portabilidad

Seguridad: Se relaciona con la confidencialidad de los datos en la transmisión y en el almacenamiento, junto con las necesidades del sistema. Posibilita evitar intrusiones no autorizadas al mismo y la capacidad para seguir eventos que comprometan esta seguridad a través del tiempo.

Nº: RNF 5	
Nombre	Seguridad
	5.1
Nombre	Confidencialidad

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Descripción	- La información que brinda el sistema estará protegida contra el acceso de usuarios no autorizados.
	5.2
Nombre	Disponibilidad
Descripción	- El sistema estará accesible las 24 horas del día.
	5.3
Nombre	Integridad
Descripción	- El sistema estará accesible cada vez que los usuarios del mismo lo requieran.

Tabla 6 Requerimientos No Funcionales: Seguridad

Software: Se especifica las condiciones o capacidades que el sistema debe cumplir. El software que se usará tanto en la parte del servidor como en la del cliente.

Nº: RNF 7	
Nombre	Software
	7.1
Nombre	Software en el Servidor
Descripción	- El servidor deberá contar con el sistema operativo Windows 95 ó Superior ó Linux; Maquina Virtual de Java versión 1.3 ó Superior; Java Runtime Environment (JRE) 1.5 ó superior, el servidor de aplicaciones JBoss AS 4.2 ó superior y el servidor de base de datos PostgreSQL Server 8.2 ó superior.
	7.2

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Nombre	Software en el Cliente
Descripción	<ul style="list-style-type: none"> - Será necesaria una computadora con el Sistema Operativo Windows o Linux en cualquiera de sus versiones, se podrá acceder desde cualquier Navegador Web recomendándose Internet Explorer 6 ó superior y Firefox 2.0 o superior.

Tabla 7 Requerimientos No Funcionales: Software

Hardware: Elementos de hardware de que se disponen.

Nº: RNF 8	
Nombre	Hardware
	8.1
Nombre	Hardware en el Cliente
Descripción	<ul style="list-style-type: none"> - Para garantizar el correcto funcionamiento de la aplicación, se necesita una computadora con un procesador Pentium IV ó superior, una memoria RAM de 256 MB ó más y un Disco Duro de 10 GB ó más.
	8.2
Nombre	Hardware en el Servidor
Descripción	<ul style="list-style-type: none"> - Se necesitará una computadora con un Intel Xeon Quad Core, 4GB de ram y un raid de discos SCSI de 5 discos y 1.5 teras entre todos.

Tabla 8 Requerimientos No Funcionales: Hardware

Restricciones en el diseño y la implementación: Especifica o restringe la codificación o construcción de un sistema. Son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Nº: RNF 9	
Nombre	Restricciones en el Diseño y la Implementación.
Descripción	<ul style="list-style-type: none">- Se utilizará el patrón de diseño MVC.- Se utilizarán patrones de implementación.- El sistema estará dividido en 3 capas: la capa del negocio, de datos y de presentación. En esta última estarán las validaciones pertinentes.- Para el desarrollo del análisis y el diseño del sistema deberá ser utilizada la metodología RUP, usando el lenguaje de modelado UML y como herramienta para llevarlo a cabo el Visual Paradigm for UML v 6.0.- Para la implementación se utilizará como lenguaje de programación Java, herramienta ID desarrollo Eclipse y para el modelado de la base de datos Case Studio v 2.22.- La biblioteca que se utilizará será RichFaces y como framework Jboss Seam.

Tabla 9 Requerimientos No Funcionales: Restricciones en el Diseño y la Implementación

Apariencia o Interfaz Externa: Describe la apariencia del producto y se especifica cómo se tratará de hacer la interfaz externa del producto contemplando las características del sistema. Se tiene en cuenta la intercomunicación con otros sistemas, que puede ser a través de servicios o de salidas en archivo y se definen los formatos y tecnologías para la exposición o la captura de información.

Nº: RNF 10	
Nombre	Apariencia o Interfaz Externa

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

	10.1
Nombre	Interfaces de Usuario
Descripción	<ul style="list-style-type: none">- Las ventanas del sistema contendrán claro y bien estructurados los datos, además de permitir la interpretación correcta de la información.- La interfaz contará con teclas de función y menús desplegable que faciliten y aceleren su utilización.- La entrada de datos incorrecta será detectada claramente e informada al usuario.- Todos los textos y mensajes en pantalla aparecerán en idioma español.- El diseño de la interfaz del sistema responderá a la ejecución de acciones de una manera rápida, lo que minimizará los pasos a dar en cada proceso. <p>La interfaz será sencilla, amigable, intuitiva y de fácil navegación por el usuario, con el objetivo de evitar la resistencia humana al uso del nuevo sistema.</p>
	10.2
Nombre	Interfaces software
Descripción	<ul style="list-style-type: none">- Se interactuará con el Sistema de Gestión Hospitalaria (HIS) para realizar solicitudes con el fin de obtener resultados de historias clínicas.
	10.3

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Nombre	Interfaces comunicación
Descripción	- Para el intercambio electrónico de datos entre aplicaciones se usará el protocolo HL7 (Health Level Seven).

Tabla 10 Requerimientos No Funcionales: Apariencia o Interfaz Externa

Ayuda y documentación en línea: Documentos ó enlaces que sirvan de guía para el manejo de las funciones del sistema. Esto posibilita que se facilite su uso.

Nº: RNF 11	
Nombre	Ayuda y documentación en línea
Descripción	<ul style="list-style-type: none"> - Presentará ayuda en línea para orientar en el uso de la interfaz. - Se desarrollará un manual de usuario para la aplicación.

Tabla 11 Requerimientos No Funcionales: Ayuda y documentación en línea

2.2 Definición de los casos de uso

- **Definición de los actores**

Actores	Justificación
Médico	Es el responsable de iniciar la comunicación con otro usuario y realiza todas las funcionalidades del Sistema de Teleconsulta. Entre las que se relacionan con un contacto se encuentran: agregarlo, ver su perfil, moverlo para un grupo escogido y eliminarlo de la teleconsulta y al grupo: agregarlo, renombrarlo y eliminarlo. Correspondiente al perfil del médico: editarlo y cambiar el avatar que lo identifica, con el chat: enviar mensajes al contacto, ver el historial de mensajes y borrar el historial y con la pizarra compartida: cargar una imagen, ampliarla, moverla y guardarla.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

Alas-HIS	Sistema externo con el que se comunica el Sistema de Teleconsulta para autenticarse.
-----------------	--------------------------------------------------------------------------------------

Tabla 12 Actores del Sistema

- **Listado de casos de uso**

Los casos de uso son descripciones narrativas en lenguaje natural de los procesos del dominio y para el establecimiento de los casos de uso se tuvo un conocimiento previo de los requerimientos del sistema a desarrollar. Ellos definen verdaderamente lo que se quiere y así se podría concretar toda la información en casos de uso.

CU-1	Agregar contacto
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico inserta un contacto, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF1

Tabla 13 Casos de Uso del Sistema: Agregar contacto

CU-2	Ver perfil
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico ve el perfil del contacto, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF2

Tabla 14 Casos de Uso del Sistema: Ver perfil

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

CU-3	Mover contacto
Actor	Médico
Descripción	El caso de uso se inicia cuando el mueve un contacto para otro grupo, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF3

Tabla 16 Casos de Uso del Sistema: Mover contacto

CU-4	Eliminar contacto
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico elimina un contacto, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF4

Tabla 17 Casos de Uso del Sistema: Eliminar contacto

CU-5	Agregar grupo
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico inserta un grupo, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF5

Tabla 18 Casos de Uso del Sistema: Agregar grupo

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

CU-6	Renombrar grupo
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico modifica un grupo, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF6

Tabla 19 Casos de Uso del Sistema: Renombrar grupo

CU-7	Eliminar grupo
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico elimina un grupo, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF7

Tabla 20 Casos de Uso del Sistema: Eliminar grupo

CU-8	Editar perfil
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico edita su perfil, realiza la operación deseada, finalizando así el caso de uso.
Referencia	RF8

Tabla 21 Casos de Uso del Sistema: Editar perfil

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

CU-9	Cambiar avatar
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico cambia su avatar, realiza la operación deseada, finalizando así el caso de uso.
Referencia	RF9

Tabla 22 Casos de Uso del Sistema: Cambiar avatar

CU-10	Enviar mensaje
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico envía el mensaje, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF10

Tabla 23 Casos de Uso del Sistema: Enviar mensaje

CU-11	Historial de mensaje
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico va al Historial de Mensajes, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF11

Tabla 24 Casos de Uso del Sistema: Historial de mensaje

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

CU-12	Borrar historial
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico borra el historial de mensajes, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF11

Tabla 25 Casos de Uso del Sistema: Borrar historial

CU-13	Compartir Pizarra
Actor	Médico
Descripción	El caso de uso se inicia cuando selecciona la opción de pizarra compartida, la abre y si quiere saber la opinión de otro médico sobre un caso, carga una imagen. Esta última opción se describe en el caso de uso Cargar Imagen. Luego de de esto cierra la pizarra compartida y finaliza así el caso de uso.
Referencia	RF13

Tabla 26 Casos de Uso del Sistema: Compartir pizarra

CU-15	Dibujar Imagen
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico dibuja una parte de la imagen, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF15

Tabla 27 Casos de Uso del Sistema: Dibujar imagen

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

CU-16	Ampliar Imagen
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico amplía la imagen, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF16

Tabla 28 Casos de Uso del Sistema: Ampliar imagen

CU-17	Mover Imagen
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico mueve la imagen, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF17

Tabla 29 Casos de Uso del Sistema: Mover imagen

CU-18	Guardar Imagen
Actor	Médico
Descripción	El caso de uso se inicia cuando el médico guarda la imagen para posteriores consultas, realiza la operación deseada y finaliza así el caso de uso.
Referencia	RF18

Tabla 30 Casos de Uso del Sistema: Guardar imagen

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

- **Casos de Uso Expandidos**

Caso de uso	
CU-14	Cargar imagen
Propósito	El Médico necesita de la opinión diagnóstica de otro médico para evaluar una imagen determinada.
Actores	
Resumen: El caso de uso se inicia cuando el Médico necesita cargar una imagen con la que va a trabajar. El Médico selecciona la opción de cargar imagen en la pizarra para ser evaluada, terminando así el caso de uso.	
Referencias	
Acción del actor	Respuesta del sistema
1.1 El médico selecciona la imagen del directorio de la computadora.	1.1.1 El sistema carga y muestra la imagen.
Puntos de extensión.	
CU-13 Abrir pizarra compartida	

Tabla 31 Casos de Uso del Sistema: Cargar imagen

- **Diagrama de casos de uso del Sistema**

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requerimientos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El mismo contiene actores, casos de uso y sus relaciones y constituye una entrada para el análisis, el diseño y las pruebas. [22]

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

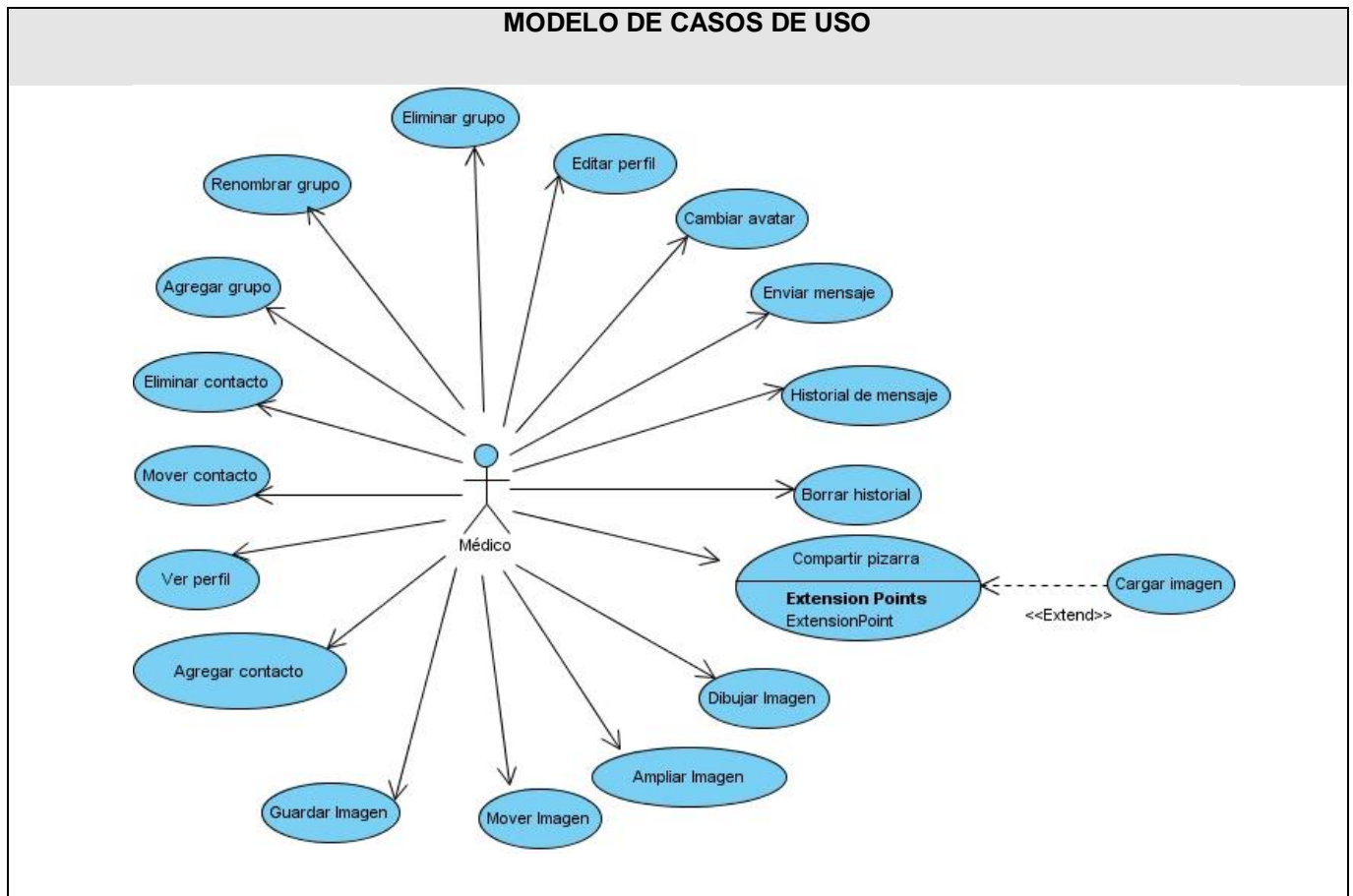


Figura 3 Diagrama de Casos de Uso del Sistema

Este capítulo brinda una visión del sistema en general a través de la descripción del objeto de estudio, se obtuvo un modelo de conceptual por la poca descripción de los procesos del negocio y se propuso una solución para resolver el problema en cuestión. Se describen los casos de uso del sistema obtenidos a través del establecimiento de los requerimientos funcionales y se definieron los requerimientos no funcionales para el éxito del sistema cuando esté terminado.

Capítulo III: Análisis y Diseño del Sistema.

Este capítulo aborda todo lo referido al análisis y diseño del sistema propuesto. Se describe la arquitectura que se utiliza, la cual se verá reflejada en los diagramas de análisis y diseño, a través de la utilización de las tecnologías, lenguaje y metodología antes descritos.

3.1 Patrones de arquitectura y diseño

- **Arquitectura**

La Arquitectura del Software aporta una visión abstracta de alto nivel, posponiendo el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. [23]

Constituye una estructura jerárquica para los módulos y la manera de interactuar entre componentes y la estructura usada entre ellos. Es el diseño de más alto nivel de la estructura de un sistema, programa o aplicación.

- **Estilo arquitectónico: Estilo centrado en datos**

El estilo centrado en datos resulta apropiado para sistemas que se centran en el acceso y actualización de datos en estructuras de almacenamiento que son compartidos por un número indefinido de componentes consumidores. Una de las propiedades más destacables de este estilo arquitectónico es la necesidad de crear persistencia de los datos almacenados. [24]

- **Patrón de pizarra**

En la arquitectura de pizarra, se pueden encontrar dos componentes principales: una estructura de datos que representa el estado actual, y un número independiente de componentes, los cuales realizan sus operaciones sobre él. Esta arquitectura se puede subdividir en dos:

- Si las transacciones en su flujo de entrada, definen los procesos que van a ejecutarse, entonces el repositorio puede ser algo como una BD tradicional.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

- Sin embargo, si el estado de dicha estructura es la que va a disparar los procesos que se ejecuten, el repositorio se llamará pizarra pura. [25]

Aplicación de el patrón pizarra y repositorio

- Aplicaciones que necesitan interpretación de proceso de señales bastantes complejas.
- Implementaciones de estilos de procesos en lotes de BD.
- Programación organizada como colecciones de servicios al rededor de un repositorio común.

Ventajas:

- Hace posible la interacción de agentes contra el sistema.
- Funciona muy bien con los problemas no deterministas (especial para I.A.)
- Se sabe el conocimiento que se lleva en cada momento del proceso.

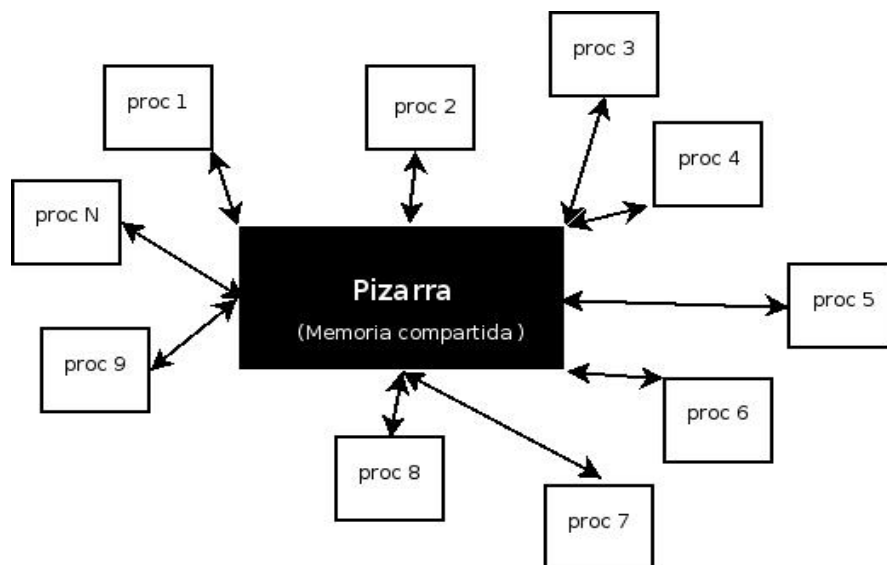


Figura 4 Patrón de arquitectura: pizarra.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

- **Arquitectura en capas**

La estrategia tradicional de utilizar aplicaciones compactas causa gran cantidad de problemas de integración en sistemas software complejos como pueden ser los sistemas de gestión de una empresa o los sistemas de información integrados consistentes en más de una aplicación. Estas aplicaciones suelen encontrarse con importantes problemas de escalabilidad, disponibilidad, seguridad, integración, entre otras. Para solventar estos problemas se ha generalizado la división de las aplicaciones en capas que normalmente serán tres: una capa que servirá para guardar los datos (base de datos), una capa para centralizar la lógica de negocio (modelo) y por último una interfaz gráfica que facilite al usuario el uso del sistema.

- **Arquitectura en 3 capas**

Este tipo de arquitectura se utiliza para programación por capas, ya que es un estilo de programación. Su objetivo primordial es la separación de la lógica de negocios de la lógica de diseño.

- **La capa de presentación o interfaz de usuario.**

Es la capa que ve el usuario, esta formada por formularios, se comunica y captura la información del usuario entrada por el formulario y se comunica solamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable (entendible y fácil de usar) para el usuario.

- **La capa de negocio.**

Es donde residen los programas que se ejecutan, aquí se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio o lógica del negocio porque es aquí donde se establecen todas las reglas que deben cumplirse.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

- La capa de acceso a datos.

Se comunica con la capa de presentación para recibir las solicitudes y presentar los resultados, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

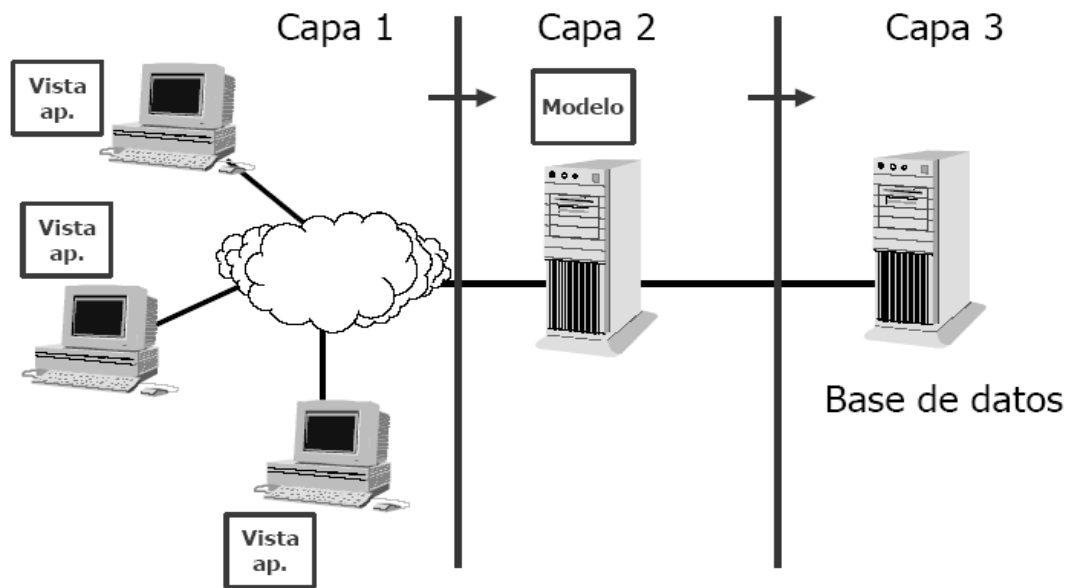


Figura 5 Arquitectura en 3 Capas

Ventajas de la aplicación de la arquitectura en 3 capas:

- Centralización de los aspectos de seguridad y transaccionalidad, que serían responsabilidad del modelo.
- No replicación de lógica de negocio en los clientes: esto permite que las modificaciones y mejoras sean automáticamente aprovechadas por el conjunto de los usuarios, lo que reduce los costes de mantenimiento.
- Mayor sencillez de los clientes.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

- **Patrón de diseño**

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. [26]

Una arquitectura orientada a objetos bien estructurada está llena de patrones. La calidad de un sistema orientado a objetos se mide por la atención que los diseñadores han prestado a las colaboraciones entre sus objetos. Los patrones conducen a arquitecturas más pequeñas, más simples y más comprensibles (Grady Booch).

Los patrones del diseño tratan los problemas del diseño que se repiten y que se presentan en situaciones particulares del diseño, con el fin de proponer soluciones a ellas. Por lo tanto, los patrones de diseño son soluciones exitosas a problemas comunes. No son fáciles de entender, pero una vez entendido su funcionamiento, los diseños resultan más flexibles, modulares y reutilizables.

- **Arquitectura basada en el patrón de diseño: Modelo-Vista- Controlador (MVC).**

La aplicación del Patrón de Diseño Modelo-Vista-Controlador resuelve el problema de la mezcla del código de acceso a datos, el código de la lógica de negocios y el código de presentación, a través de la separación del acceso a datos, la lógica de negocios, la presentación de datos y la interacción del usuario. A partir de su utilización se pueden reutilizar componentes del modelo y se hace fácil darle soporte para nuevos clientes. Por lo tanto el patrón MVC representa un mecanismo de mejora de procesos de desarrollo de software, fácil de comprender y aplicar.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

En el patrón MVC las entradas del usuario, los modelos del mundo exterior y la retroalimentación visual están separados y manejados por tres tipos de objetos, cada uno especializado para un conjunto de tareas específicas.

- **Modelo:** Es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. No tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos.
- **Vista:** Es el objeto que maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario. Interactúa con el Modelo a través de una referencia al propio Modelo.
- **Controlador:** Es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo.

La vista se modeló con la utilización de las clases interfaz, formulario y la servidora, donde ellas pueden acceder a las clases entidad para consultar o actualizar un dato. Está desarrollada con la tecnología JSF, se usó la librería Richfaces la cual se integra con facilidad con Seam UI, junto a otras, las cuales en conjunto mejoran las validaciones y la navegación.

Para el diseño del sistema, el objeto Modelo está representado por las clases entidades, ya que se realizó un mapeo directo desde la base de datos, por lo que no es necesario que se modele en los diagramas. Estas entidades pueden ser autogeneradas (generadas por la base de datos) y las personalizadas (modificadas), ya que estas contienen toda la información posible a la cual acceden las clases controladoras para insertar, modificar, buscar, crear o eliminar datos. Para el acceso a datos, se utilizó JPA de Hibernate para minimizar las configuraciones en XML y los servicios de EJB3, lo que posibilita que se minimicen las transacciones, entre otras cosas.

Para representar la Controladora, se utilizó las clases controladoras autogeneradas (generadas por el entorno), las del proceso (propias del sistema) y las personalizadas (modificadas), todas ellas acceden a

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

las entidades para solicitar alguna información. Para su desarrollo se utilizó el framework Seam, ya que permite la unión de varias tecnologías como JSF, EJB3, JPA, se integra con la librería Richfaces y permite entre sus diferentes innovaciones, facilidades relacionadas con el contexto conversacional y los procesos de negocio, lo que permite que se haga un uso mas eficiente de la memoria, entre otras ventajas.

Ventajas del uso de este patrón:

- Agregar nuevas vistas y nuevas formas de recolectar las órdenes del usuario (interpretar sus modelos mentales) y modificar los objetos de negocios para poder migrar a otra tecnología.
- Las vistas también son susceptibles de modificación sin necesidad de provocar que todo el sistema se paralice. [27]

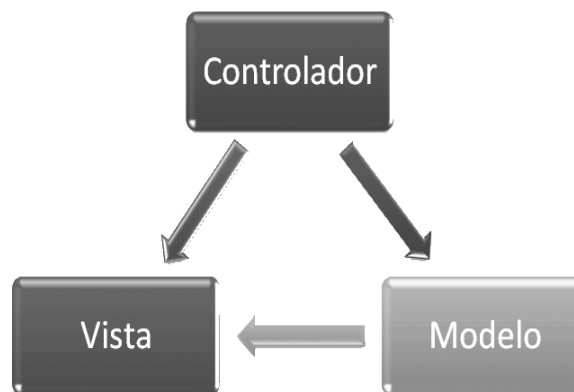


Figura 6 MVC (Modelo-Vista-Controlador)

3.2 Estrategias de Integración

El Sistema de Teleconsulta Online se va integrar al Sistema de Gestión Hospitalaria (HIS) para el acceso a la aplicación. Al hacer uso del servicio de autenticación, el sistema interactuará con la base de datos de usuarios del HIS; con el objetivo de conocer las personas que trabajen en el Ministerio de Salud Pública que estén registrados en el sistema, para poder intercambiar información .

3.3 Análisis

El análisis dice lo que hace el sistema, comprende el análisis de requerimientos donde se genera una fluida comunicación con el usuario y se revelan las necesidades del sistema y se definen sus objetivos, su alcance y se establecen todas sus funciones a contemplar. [28]

El análisis consiste en obtener una visión inicial de cómo la implementación debe ser, por lo que es una guía indiscutible para los programadores. En él se definen las clases que estarán asociadas con el contexto del dominio del problema por lo que representan conceptos y relaciones.

Clases Interfaz

Las clases interfaz representan la interacción entre el sistema y sus actores.

Clases entidad

Estas clases modelan información referente a fenómenos, conceptos y sucesos que ocurren en la vida real, los cuales poseen larga vida y a menudo son persistentes.

Clases de control

Las clases de control regulan el trabajo de uno o unos pocos casos de uso, lo que coordina las actividades de los objetos que implementan la funcionalidad del caso de uso.

- **Modelo de análisis:**

Es usado para representar la estructura global del sistema, describe la realización de casos de uso, sirve como una abstracción del Modelo de Diseño y se centra en los requerimientos funcionales. Este modelo no es un diagrama final que describe todos los posibles conceptos y sus relaciones, es un primer intento por definir los conceptos claves que describen el sistema.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

Diagrama de Clases del Análisis

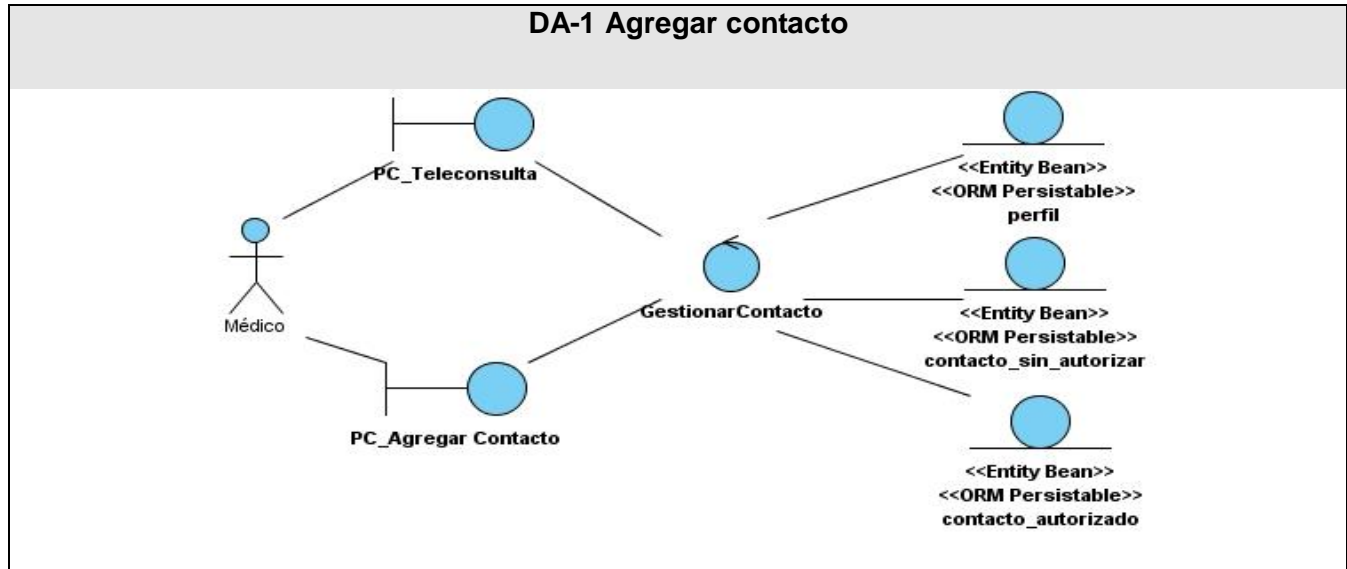


Figura 7 Diagrama de Clases del Análisis: Agregar contacto

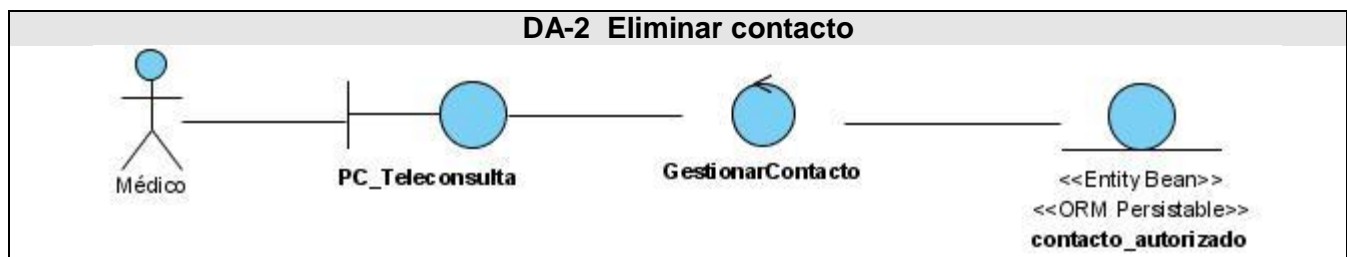


Figura 8 Diagrama de Clases del Análisis: Eliminar contacto

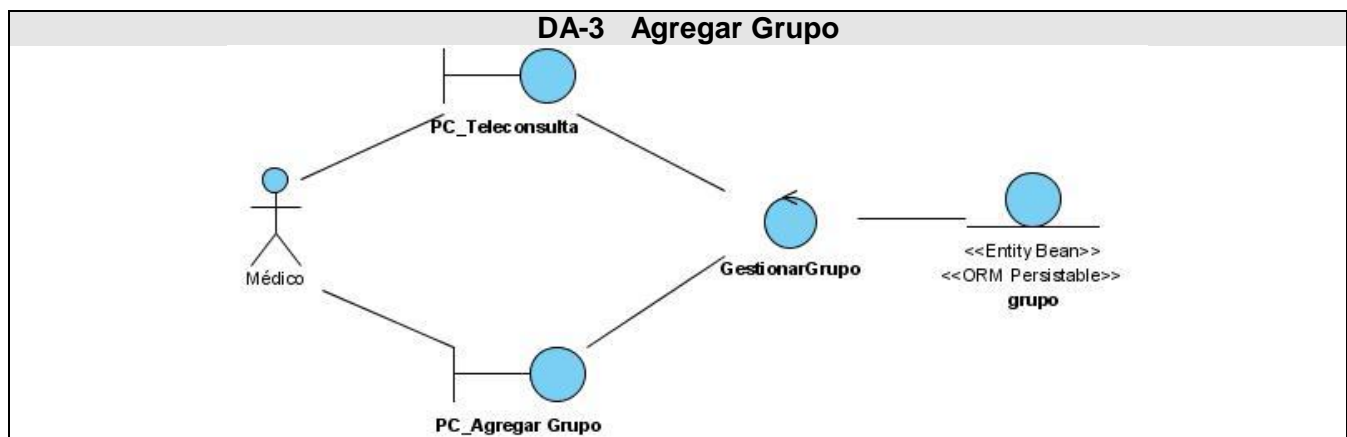


Figura 9 Diagrama de Clases del Análisis: Agregar grupo

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

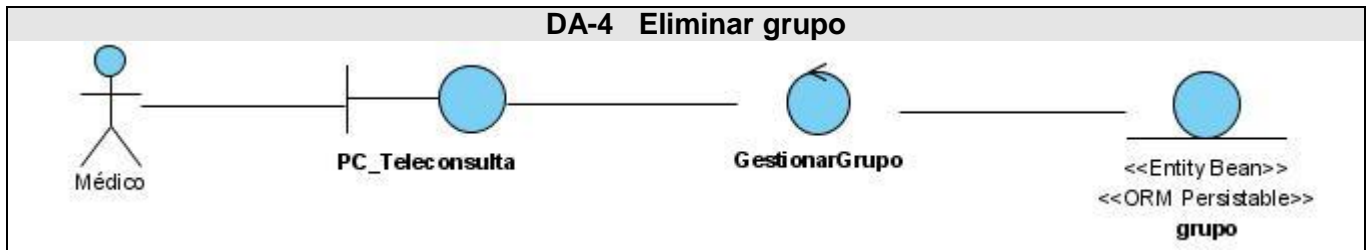


Figura 10 Diagrama de Clases del Análisis: Eliminar grupo

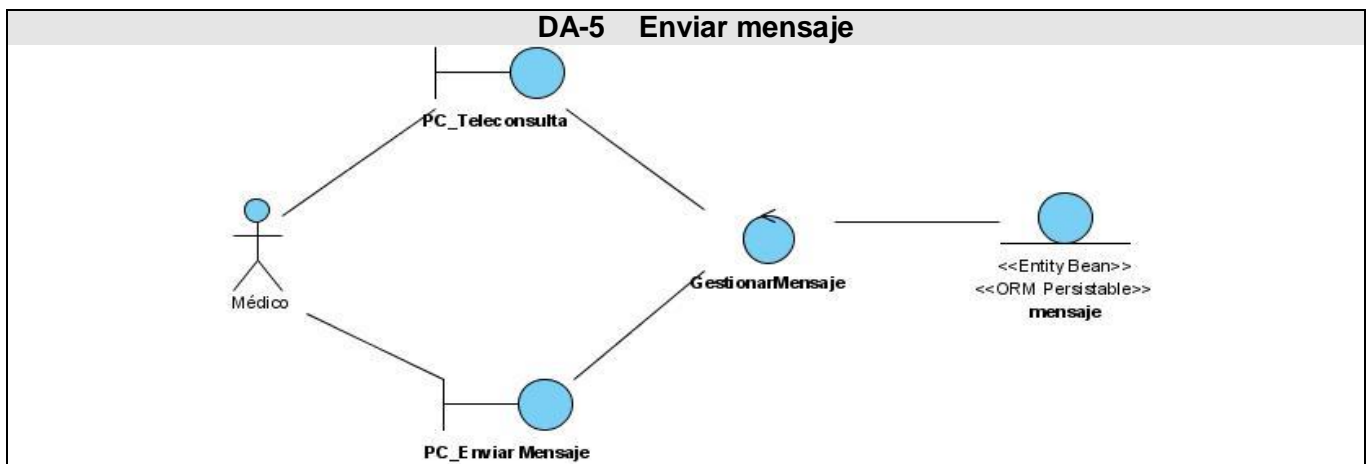


Figura 11 Diagrama de Clases del Análisis: Enviar mensaje

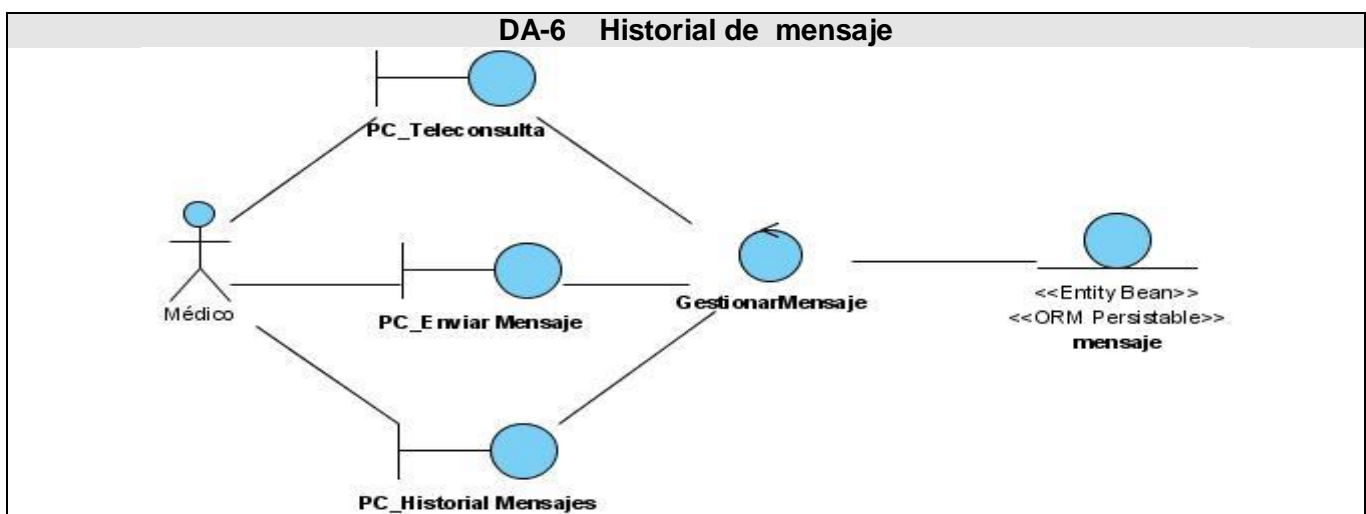


Figura 12 Diagrama de Clases del Análisis: Historial de mensajes

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

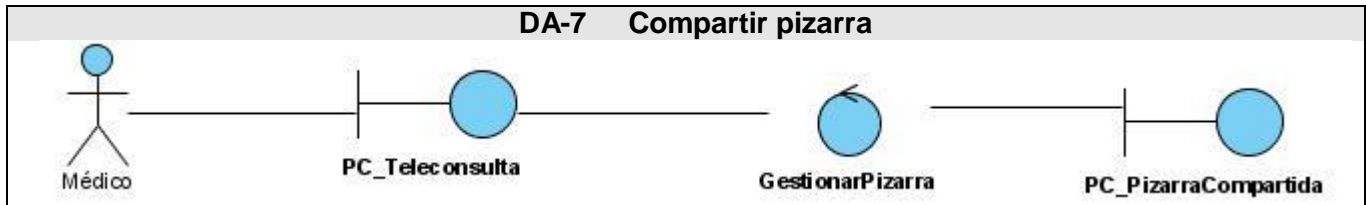


Figura 13 Diagrama de Clases del Análisis: Compartir pizarra

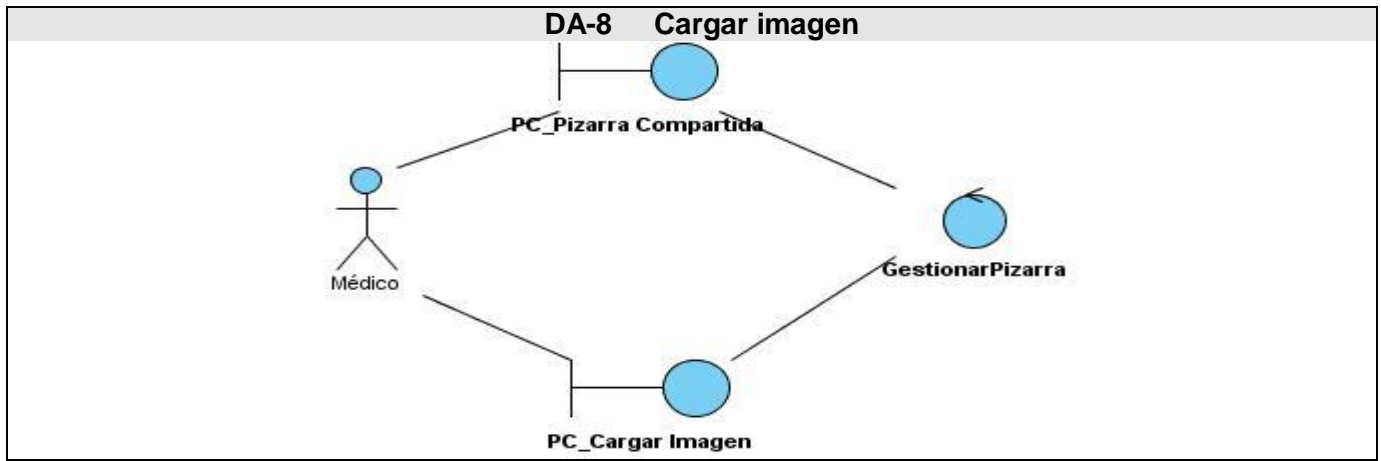


Figura 14 Diagrama de Clases del Análisis: Cargar imagen



Figura 15 Diagrama de Clases del Análisis: Dibujar imagen

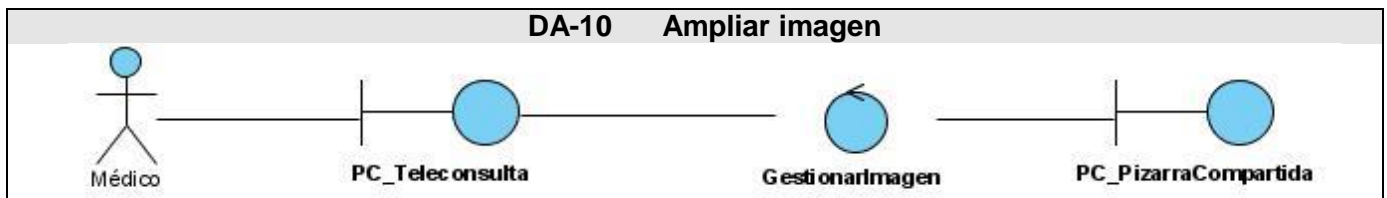


Figura 16 Diagrama de Clases del Análisis: Ampliar imagen

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

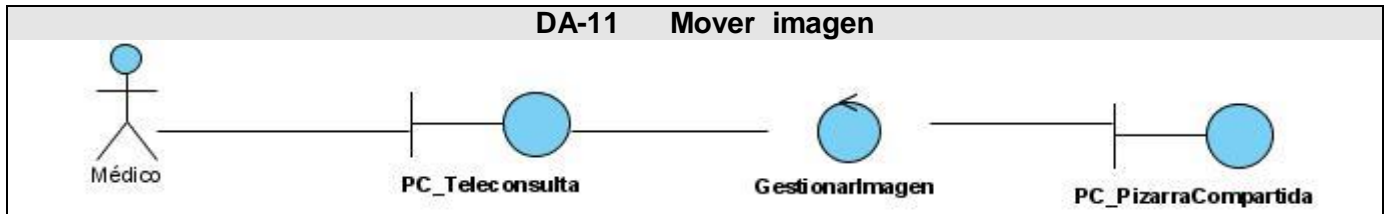


Figura 17 Diagrama de Clases del Análisis: Mover imagen

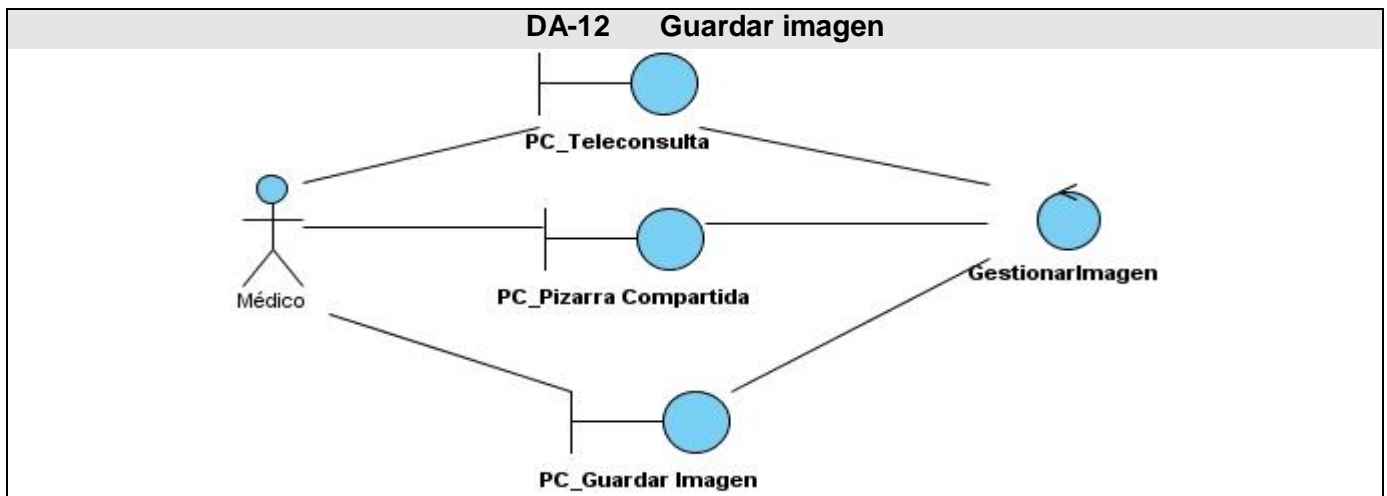


Figura 18 Diagrama de Clases del Análisis: Guardar imagen

3.4 Diseño

El diseño es un refinamiento que toma en cuenta los requerimientos no funcionales, por lo cual se centra en como el sistema cumple sus objetivos, traduce los requerimientos funcionales y no funcionales en una representación del software, constituyendo el primer paso para el desarrollo de cualquier sistema.

- **Modelo de diseño:**

Un Modelo de Diseño es una abstracción del Modelo de Implementación y su código fuente, el cual se emplea para representar y documentar su diseño. Es usado como entrada esencial en las actividades relacionadas a la implementación. Representa a los casos de uso en el dominio de la solución. [29]

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

Para el modelado del diseño se realizó un diagrama de clases por cada caso de uso y un diagrama de secuencia por cada escenario del caso de uso. Se utilizó la arquitectura definida: el patrón Modelo Vista Controlador.

Diagrama de Clases del Diseño

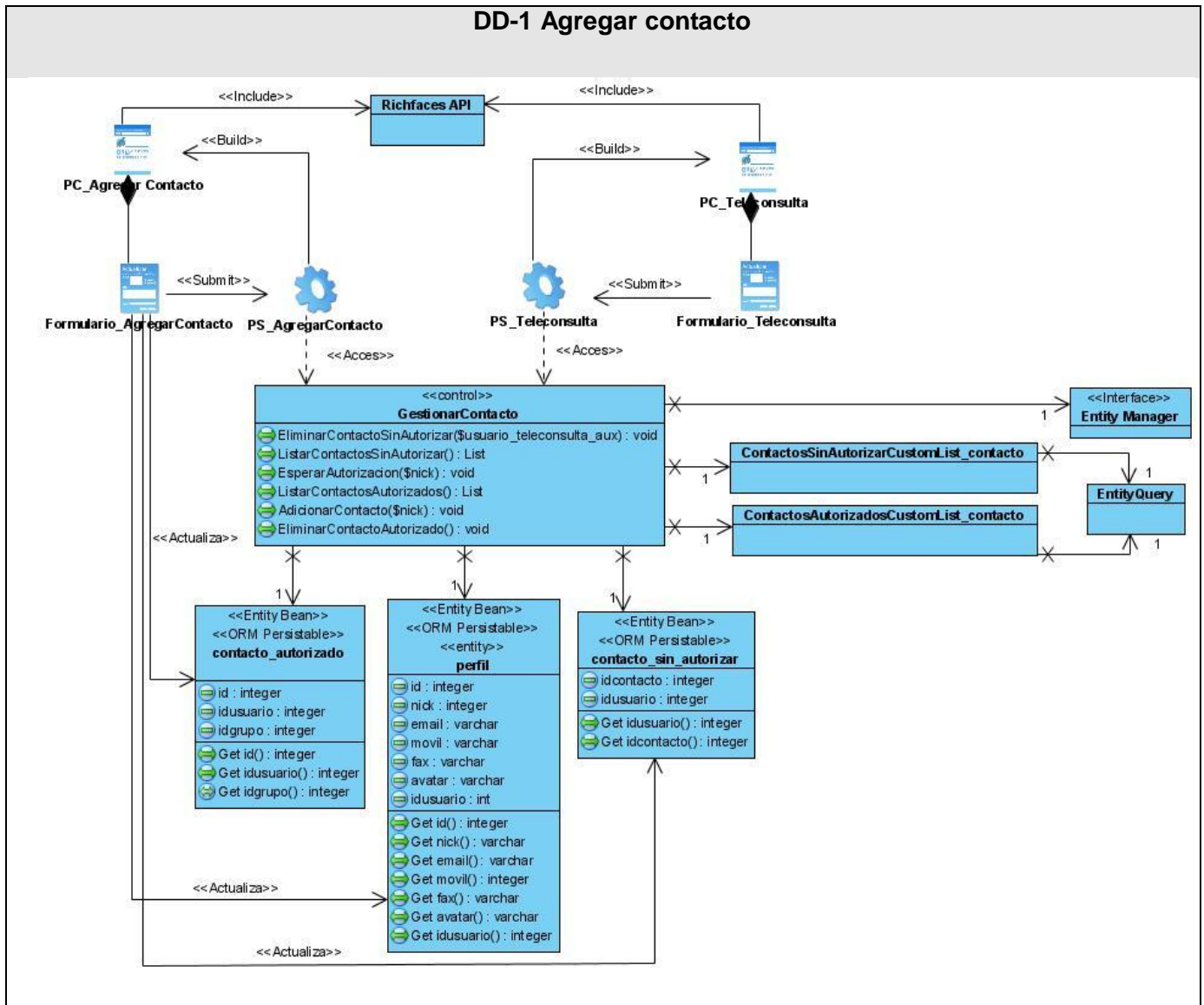


Figura 19 Diagrama de Clases del Diseño: Agregar contacto

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

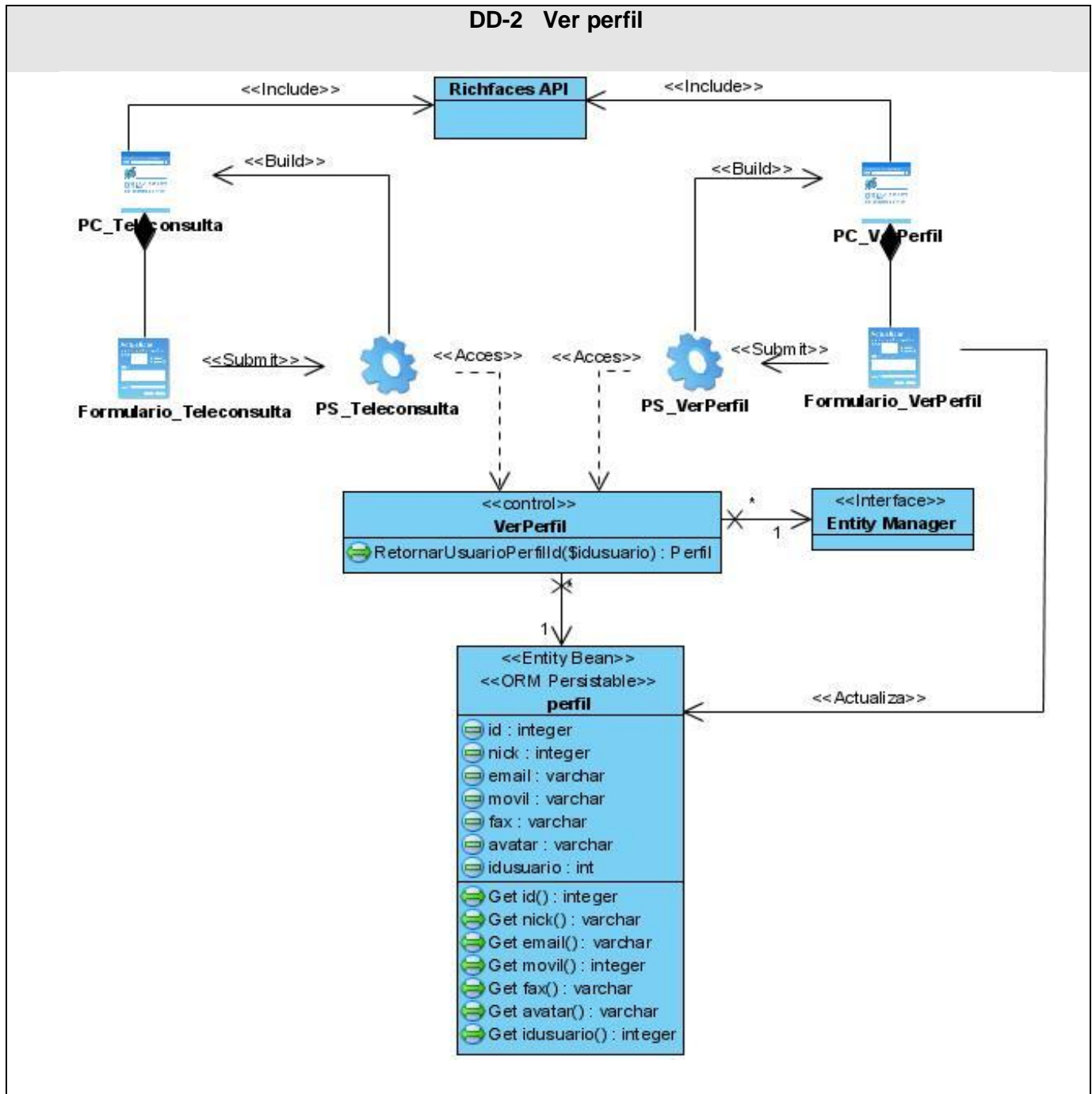


Figura 20 Diagrama de Clases del Diseño: Ver perfil

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

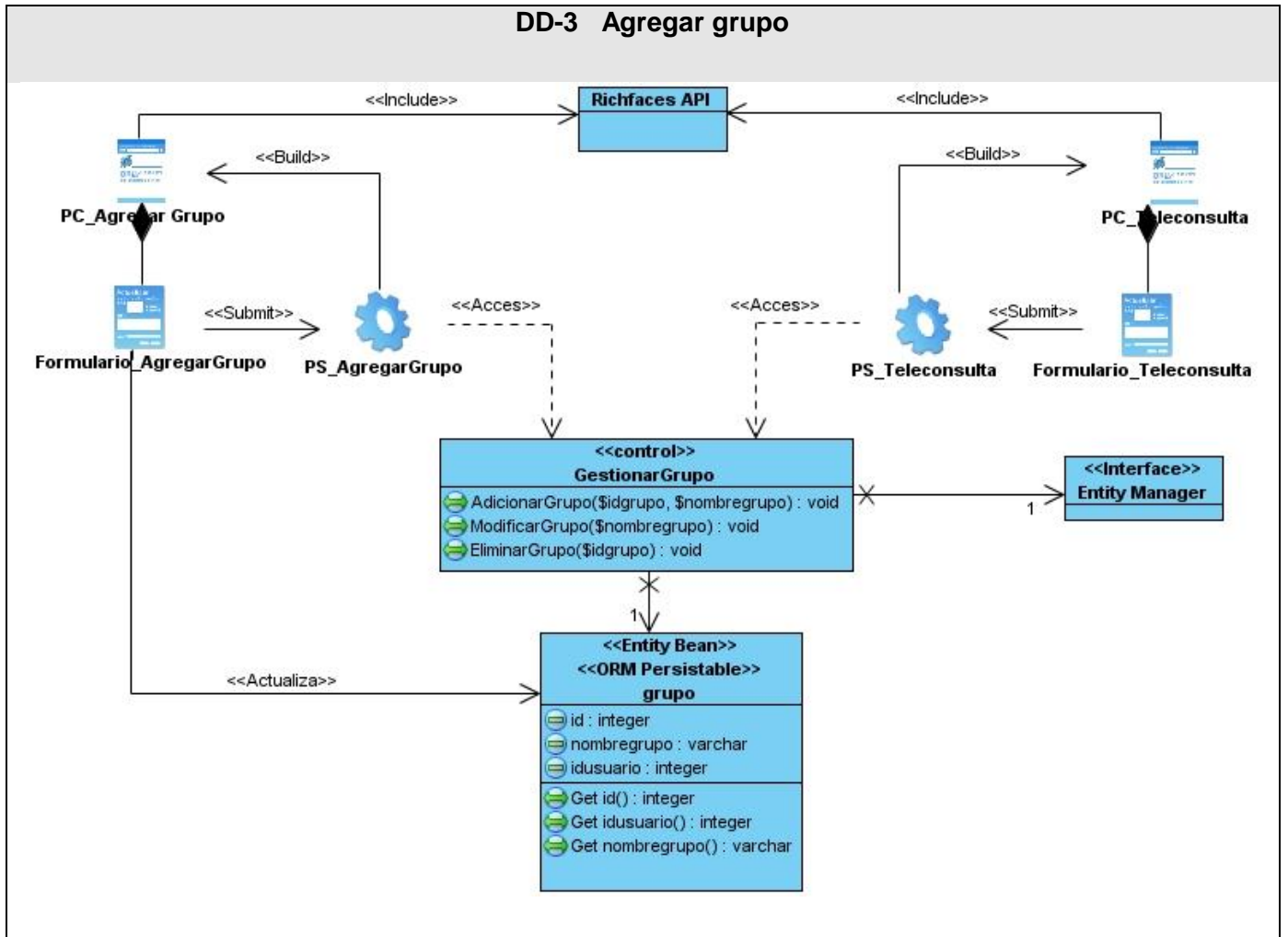


Figura 21 Diagrama de Clases del Diseño: Agregar grupo

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

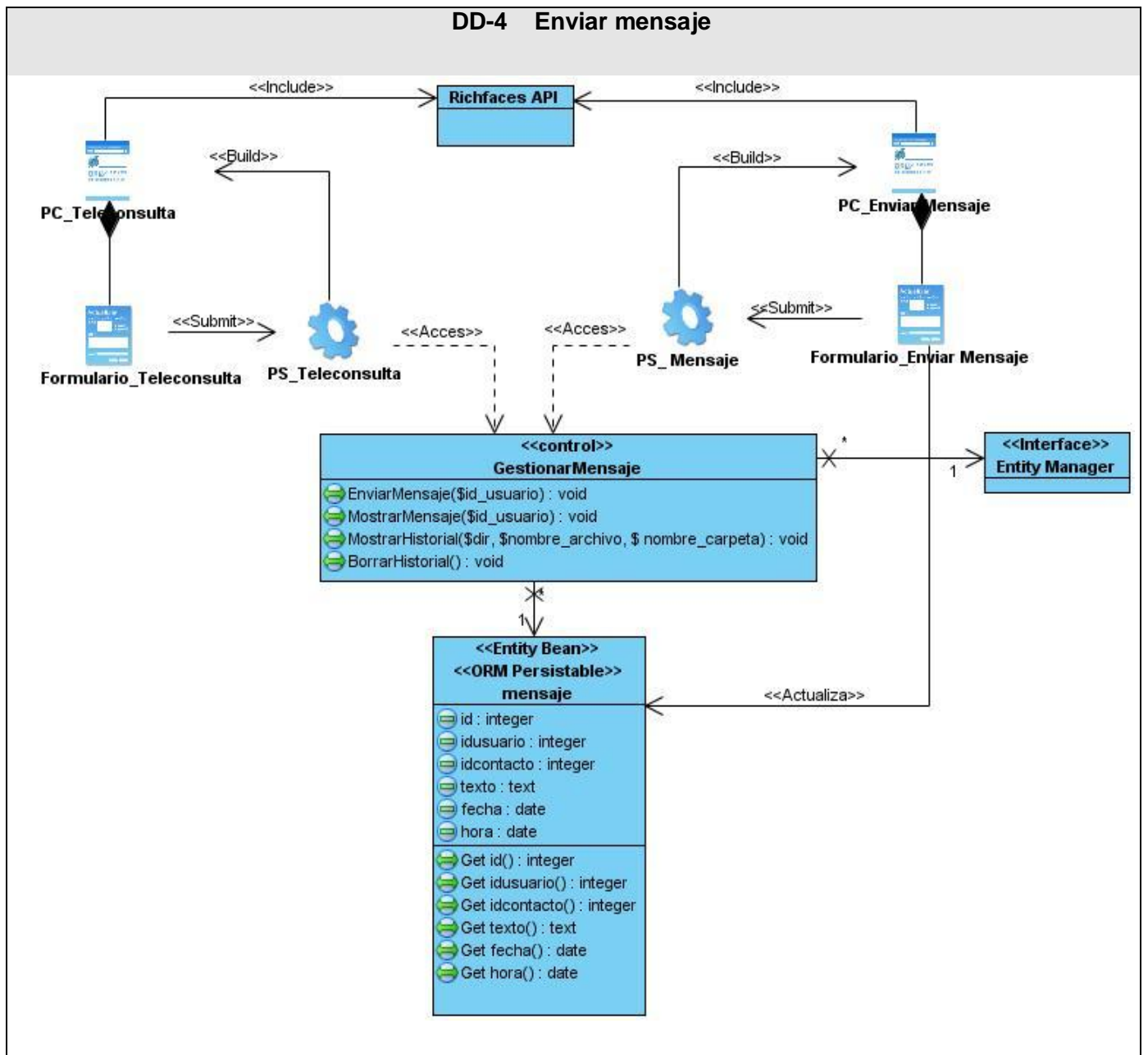


Figura 22 Diagrama de Clases del Diseño: Enviar mensaje

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

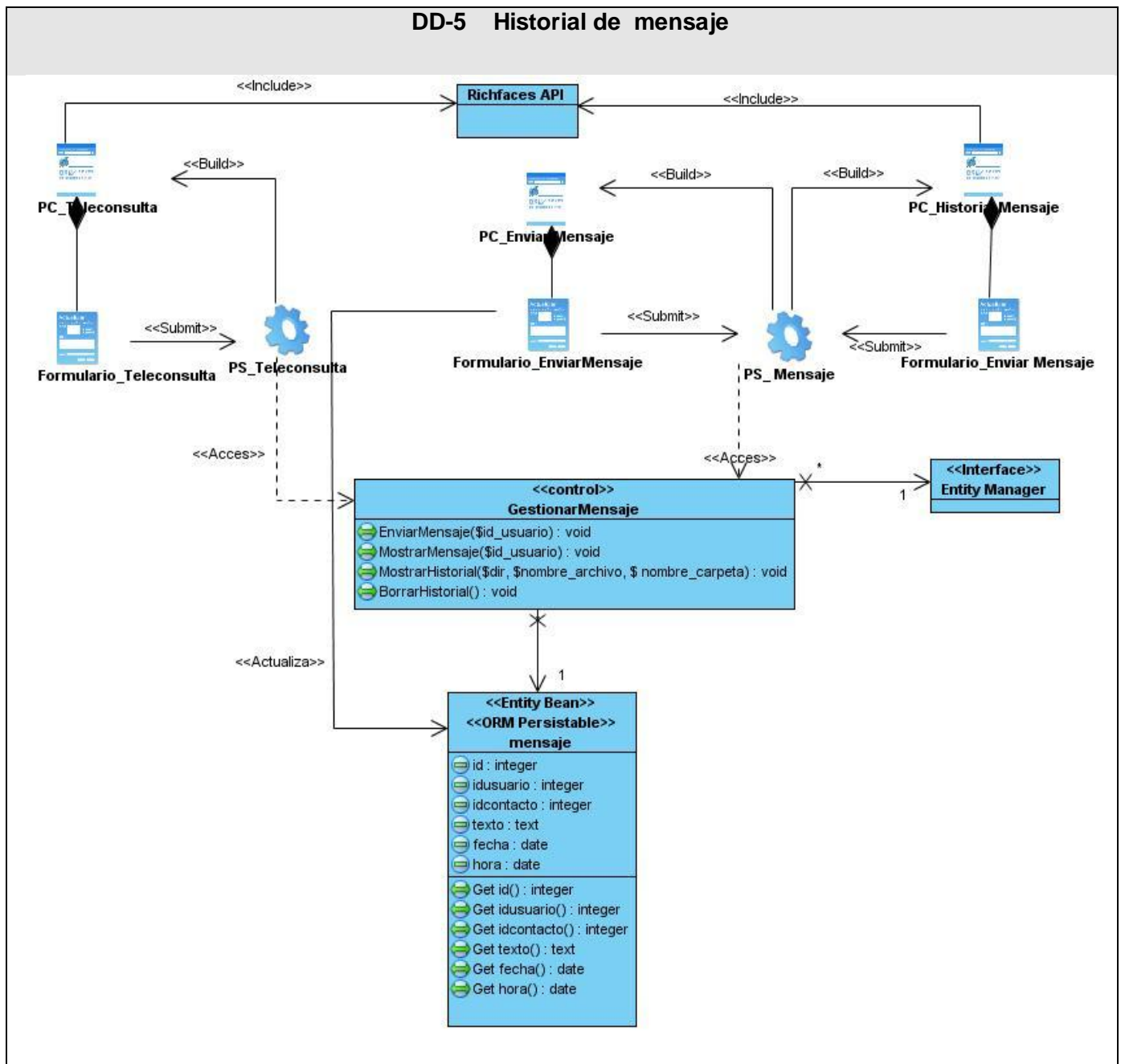


Figura 23 Diagrama de Clases del Diseño: Historial de mensaje

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

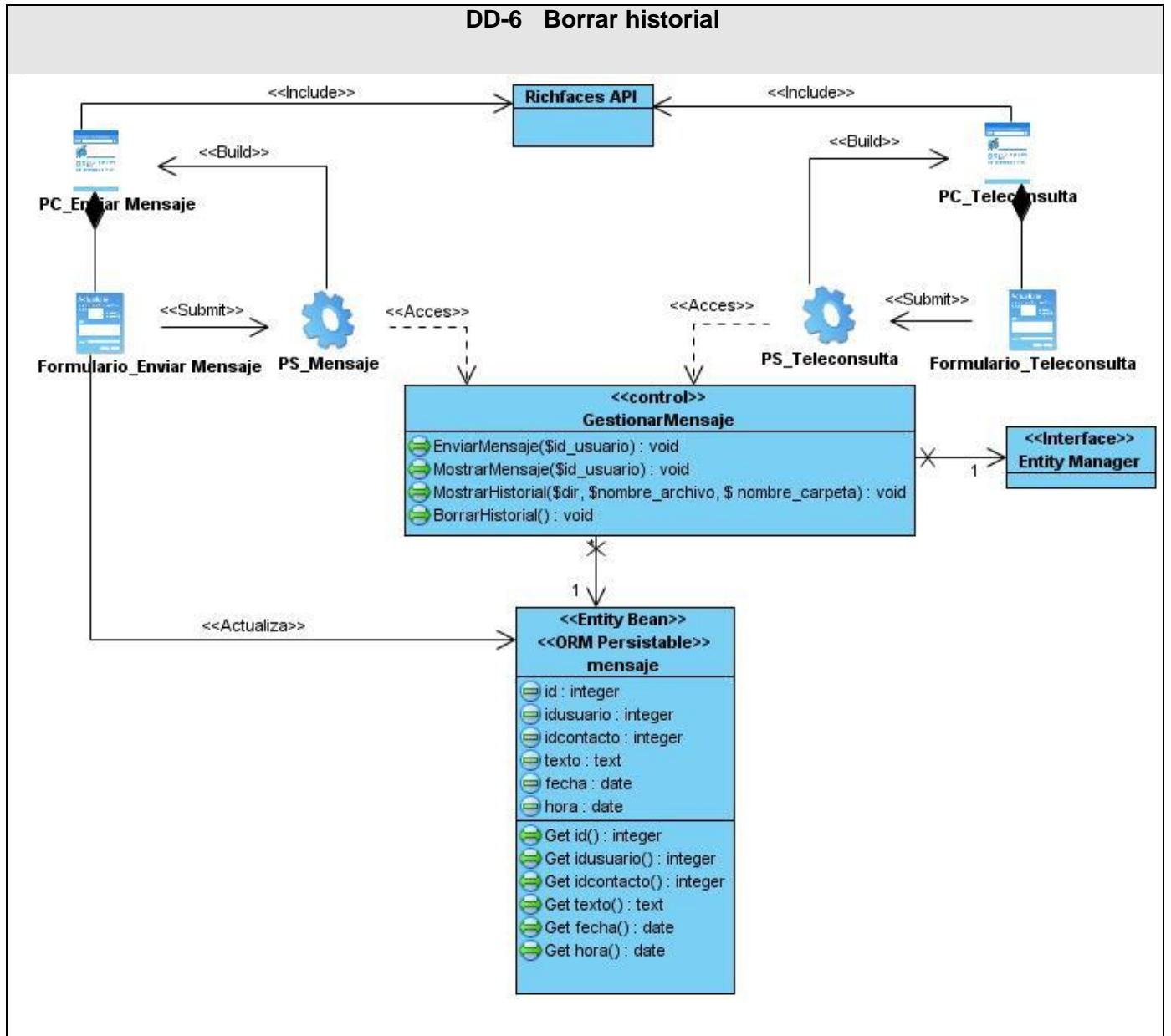


Figura 24 Diagrama de Clases del Diseño: Borrar historial

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

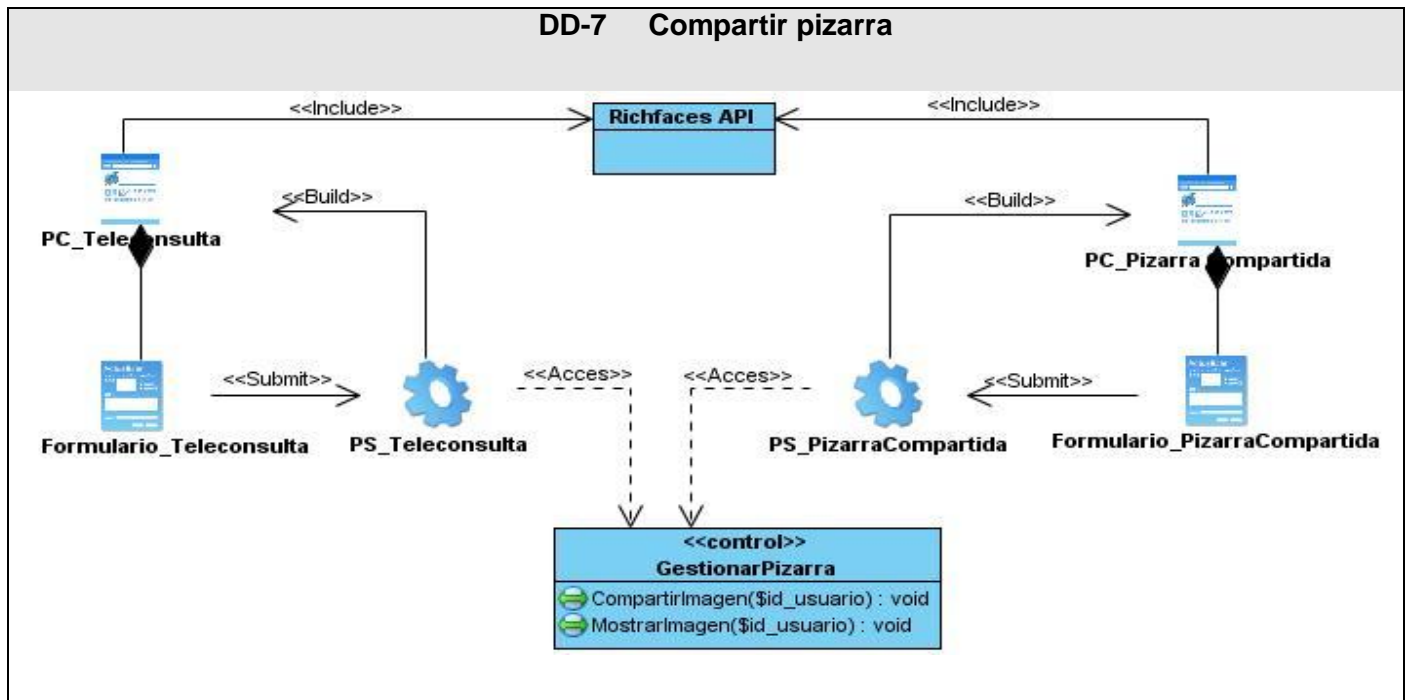


Figura 25 Diagrama de Clases del Diseño: Compartir pizarra

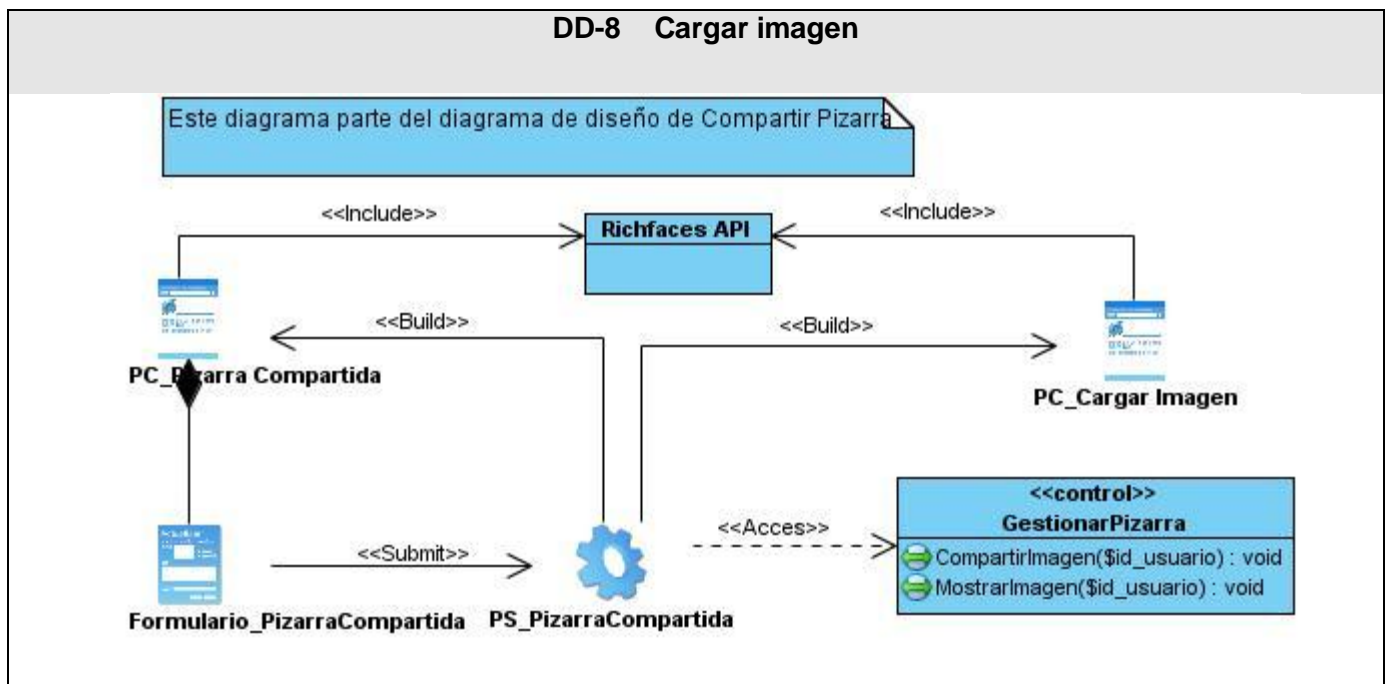


Figura 26 Diagrama de Clases del Diseño: Cargar imagen

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

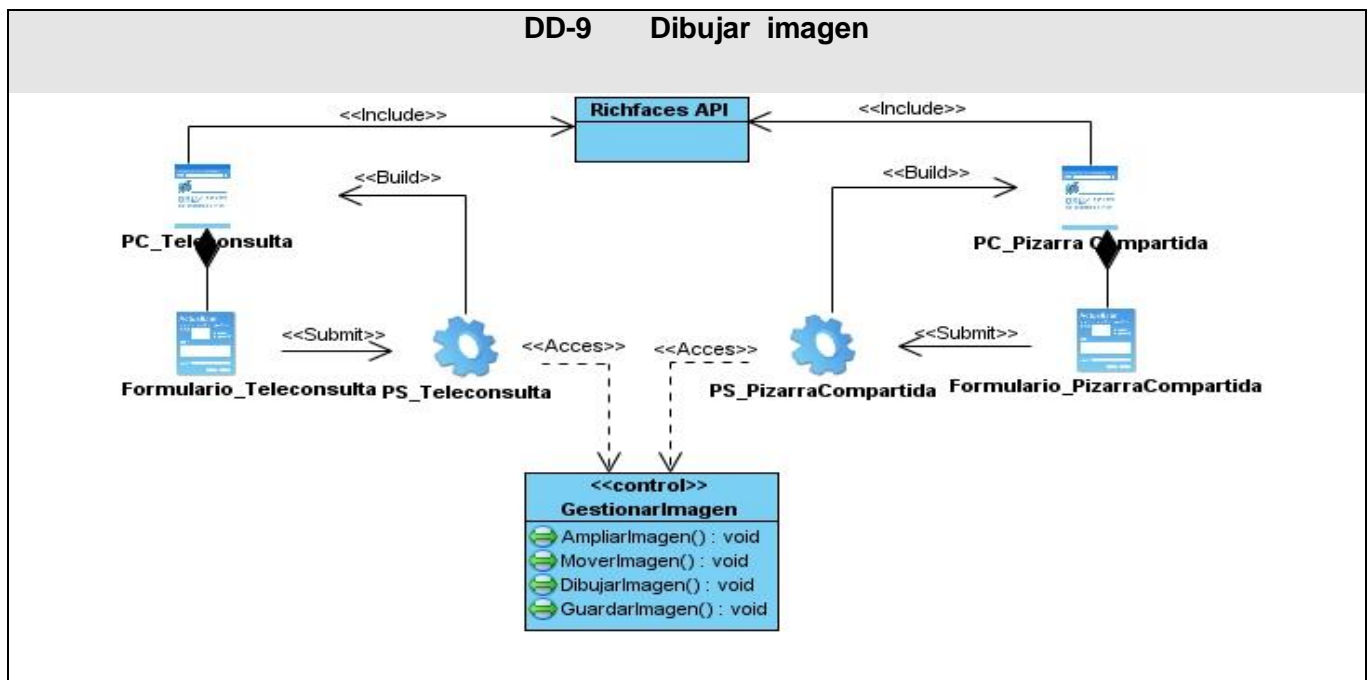


Figura 27 Diagrama de Clases del Diseño: Dibujar imagen

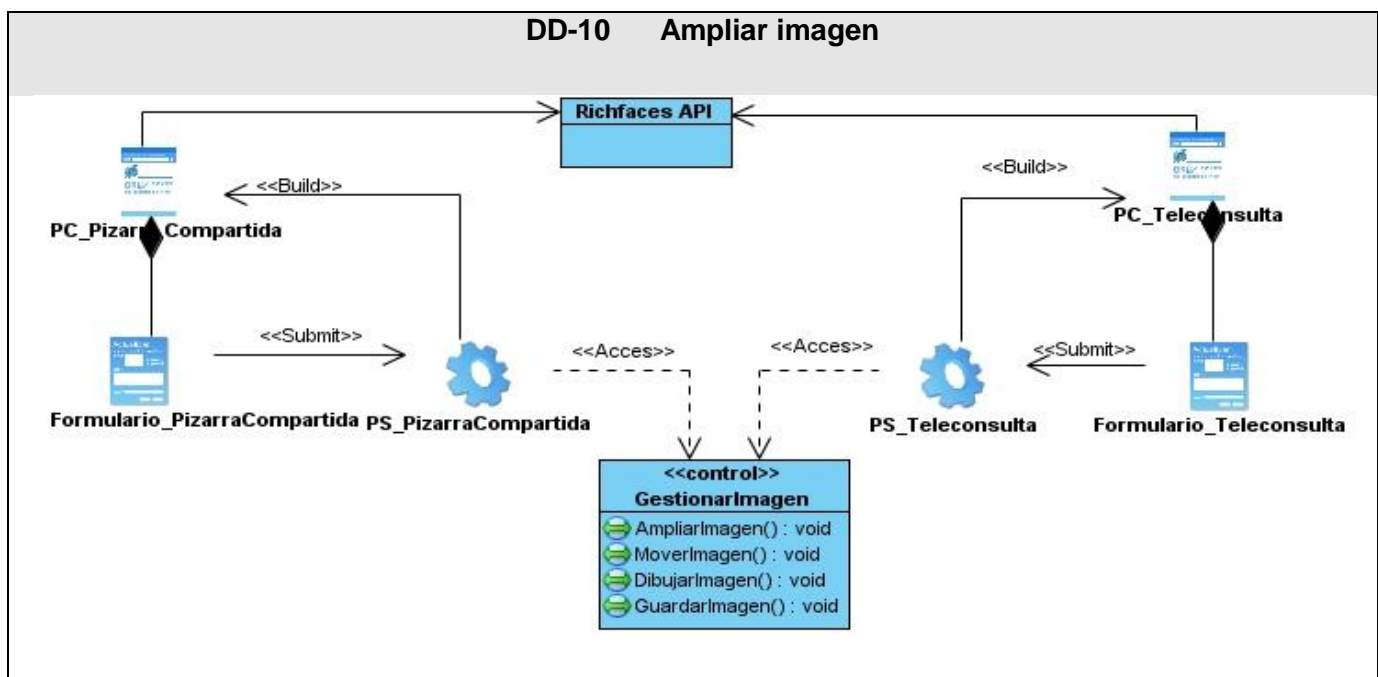


Figura 28 Diagrama de Clases del Diseño: Ampliar imagen

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

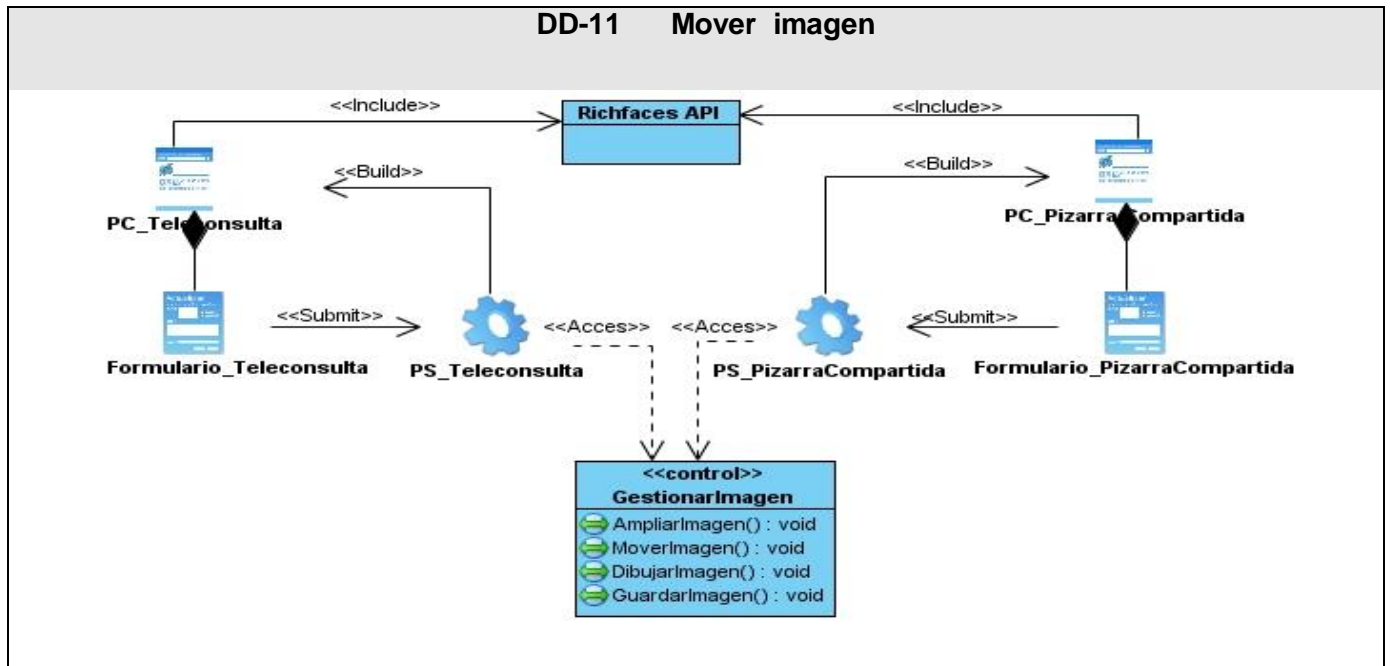


Figura 29 Diagrama de Clases del Diseño: Mover imagen

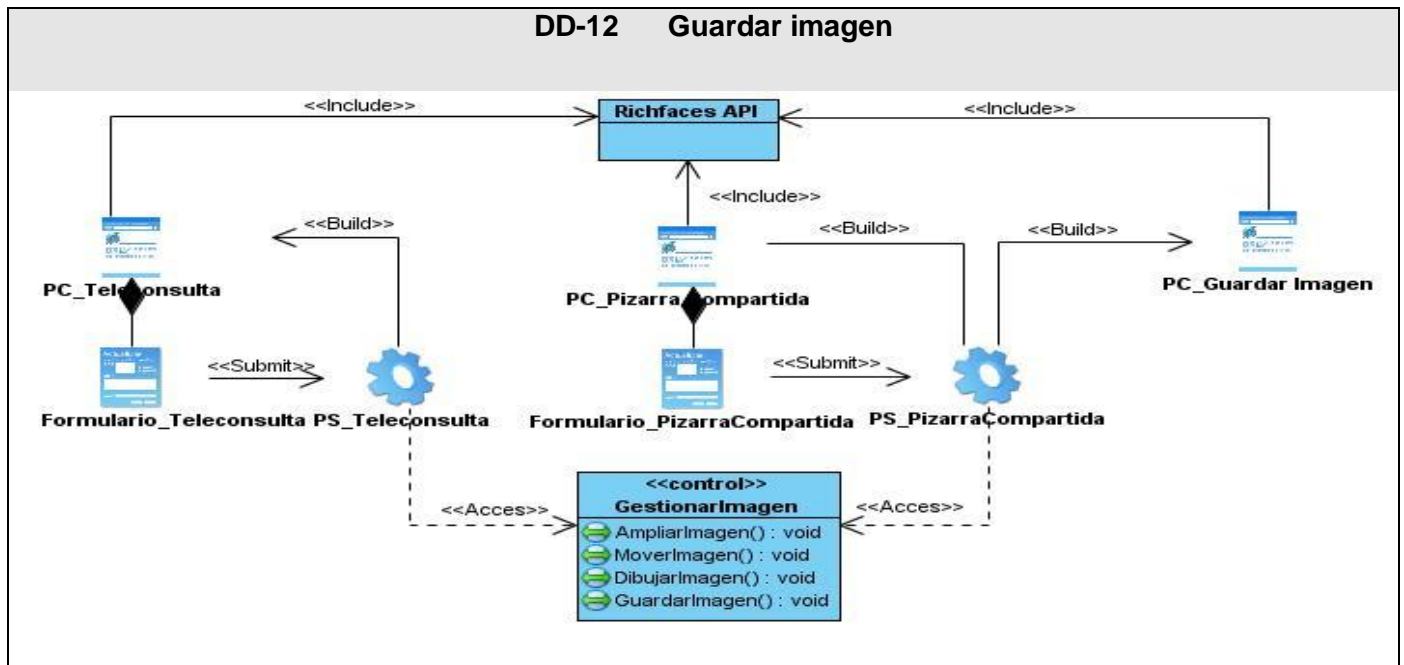


Figura 30 Diagrama de Clases del Diseño: Guardar imagen

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

- Descripción de las clases.

- Clases Interfaz


Nombre: Teleconsulta	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa la página principal del sistema y permite ver los contactos agrupados por grupos y contiene los menús principales de la teleconsulta.
 PC_Teleconsulta	

Tabla 32 Clases Interfaz: Teleconsulta


Nombre: Agregar Contacto	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario agrega el contacto a la teleconsulta.
 PC_Agregar Contacto	

Tabla 33 Clases Interfaz: Agregar Contacto


Nombre: Ver Perfil	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario ve el perfil del contacto.
 PC_Ver Perfil	

Tabla 34 Clases Interfaz: Ver Perfil

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA


Nombre: Editar Perfil	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario puede cambiar sus datos.
 PC_Editado Perfil	

Tabla 35 Clases Interfaz: Editar Perfil


Nombre: Cambiar Avatar	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario puede cambiar el avatar que lo identifica.
 PC_Cambiar Avatar	

Tabla 36 Clases Interfaz: Cambiar Avatar


Nombre: Agregar Grupo	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario agrega el grupo a la teleconsulta.
 PC_Agregar Grupo	

Tabla 37 Clases Interfaz: Agregar Grupo

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA


Nombre: Renombrar Grupo	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario agrega el grupo a la teleconsulta.
 PC_Renombrar Grupo	

Tabla 38 Clases Interfaz: Renombrar Grupo


Nombre: Enviar Mensaje	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario envía el mensaje al contacto.
 PC_Enviar Mensaje	

Tabla 39 Clases Interfaz: Enviar Mensaje


Nombre: Historial Mensaje	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario puede ver el historial de mensajes.
 PC_Historial Mensaje	

Tabla 40 Clases Interfaz: Historial Mensaje

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA


Nombre: Pizarra Compartida	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde el usuario podrá escoger las opciones de cargar, mover, ampliar, dibujar y guardar una imagen para facilitar la teleconsulta.
 PC_Pizarra Compartida	

Tabla 41 Clases Interfaz: Pizarra Compartida


Nombre: Cargar Imagen	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página donde se mostrará la imagen seleccionada por el usuario
 PC_Cargar Imagen	

Tabla 42 Clases Interfaz: Cargar Imagen


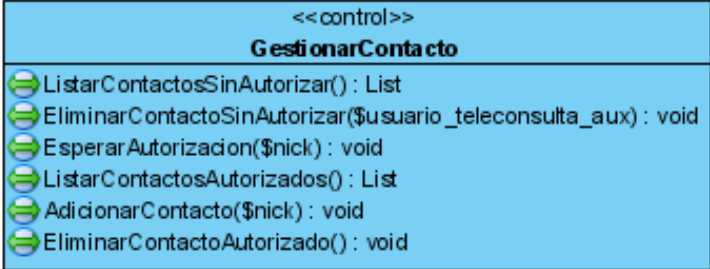
Nombre: Guardar Imagen	
Tipo de clase: (Interfaz)	
Responsabilidad:	
Descripción:	Representa a la página que permitirá guardar la imagen de la teleconsulta.
 PC_Guardar Imagen	

Tabla 43 Clases Interfaz: Guardar Imagen

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

➤ Clases Controladoras

Nombre: GestionarContactos	
Tipo de clase (Controladora)	
 <pre> classDiagram class GestionarContacto { <<control>> ListarContactosSinAutorizar() List EliminarContactoSinAutorizar(\$usuario_teleconsulta_aux) void EsperarAutorizacion(\$nick) void ListarContactosAutorizados() List AdicionarContacto(\$nick) void EliminarContactoAutorizado() void } </pre>	
Para cada responsabilidad:	
Nombre:	AdicionarContacto(\$nick) : void
Descripción:	Se utiliza para adicionar un contacto que ha aceptado la solicitud de suscripción al sistema de teleconsulta online.
Nombre:	ListarContactoAutorizado() : List
Descripción:	Se utiliza para poder listar todos los contactos existentes aceptados por el médico en la página principal del sistema de teleconsulta online.
Nombre:	EliminarContactoAutorizado() : void
Descripción:	Se utiliza para poder eliminar un contacto de la lista de contactos del sistema de teleconsulta online.
Nombre:	ListarContactoSinAutorizar() : List
Descripción:	Se utiliza para poder listar todos los contactos existentes que no han sido aceptados por el médico en la página principal del sistema de teleconsulta online.
Nombre:	EliminarContactoSinAutorizar(\$usuario_teleconsulta_aux) : void
Descripción:	Se utiliza para poder eliminar un contacto de la lista de contactos no autorizados del sistema de teleconsulta online.
Nombre:	EsperarAutorizacion(\$nick) : void
Descripción:	Se utiliza para enviar un mensaje a la página principal de teleconsulta de espera de autorización por parte del usuario.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

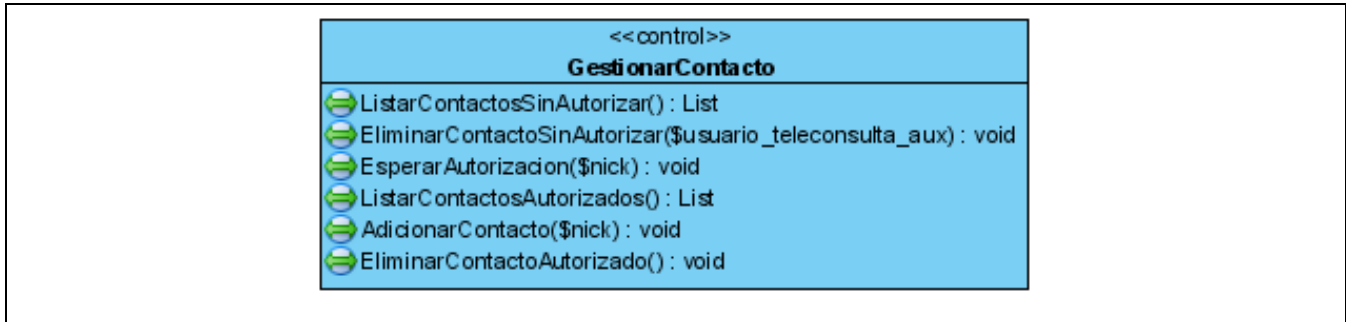


Tabla 44 Clases Controladoras: GestionarContactos

Nombre: MoverContacto	
Tipo de clase (Controladora)	
Para cada responsabilidad:	
Nombre:	RetornarGrupoldUsuario(\$idusuario) : List
Descripción:	Se utiliza para devolver el identificador del usuario de una determinado grupo
Nombre:	MoverContactoNuevoGrupo(\$id_contacto\$nombregrupo) : void
Descripción:	Se utiliza para mover un contacto para un grupo seleccionado por el usuario.
<pre> classDiagram class MoverContacto { <<control>> RetornarGrupoldUsuario(\$idusuario) List MoverContactoNuevoGrupo(\$id_contacto\$nombregrupo) void } </pre>	

Tabla 45 Clases Controladoras: MoverContacto

Nombre: VerPerfil	
Tipo de clase (Controladora)	
Para cada responsabilidad:	
Nombre:	RetornarUsuarioPerfilld(\$idusuario) : Perfil
Descripción:	Se utiliza para conocer el perfil del usuario.
<pre> classDiagram class VerPerfil { <<control>> RetornarUsuarioPerfilld(\$idusuario) Perfil } </pre>	

Tabla 46 Clases Controladoras: VerPerfil

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

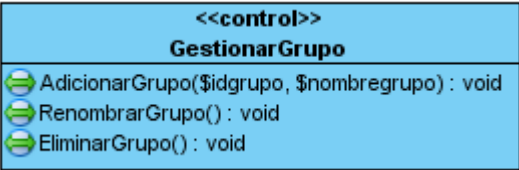
Nombre: GestionarGrupo	
Tipo de clase (Controladora)	
Para cada responsabilidad:	
Nombre:	AdicionarGrupo(\$idgrupo, \$nombregrupo) : void
Descripción:	Se utiliza para adicionar un grupo al sistema de teleconsulta online.
Nombre:	RenombrarGrupo() : void
Descripción:	Se utiliza para cambiarle el nombre al grupo seleccionado por el médico del sistema de teleconsulta online.
Nombre:	EliminarGrupo() : void
Descripción:	Se utiliza para poder eliminar un grupo del sistema de teleconsulta online.
 <pre> classDiagram class GestionarGrupo { <<control>> +AdicionarGrupo(\$idgrupo, \$nombregrupo) : void +RenombrarGrupo() : void +EliminarGrupo() : void } </pre>	

Tabla 47 Clases Controladoras: GestionarGrupo

Nombre: GestionarPerfil	
Tipo de clase (Controladora)	
Para cada responsabilidad:	
Nombre:	RetornarPerfil(\$id_usuario) : Perfil
Descripción:	Se utiliza para conocer algunos datos personales del médico
Nombre:	EditarPerfil() : void
Descripción:	Se utiliza para poder cambiar datos personales e insertar otros de igual importancia del médico.
Nombre:	MostrarAvatar(\$id_usuario) : void
Descripción:	Se utiliza para poder cambiar el avatar del usuario y mostrarlo en la teleconsulta.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

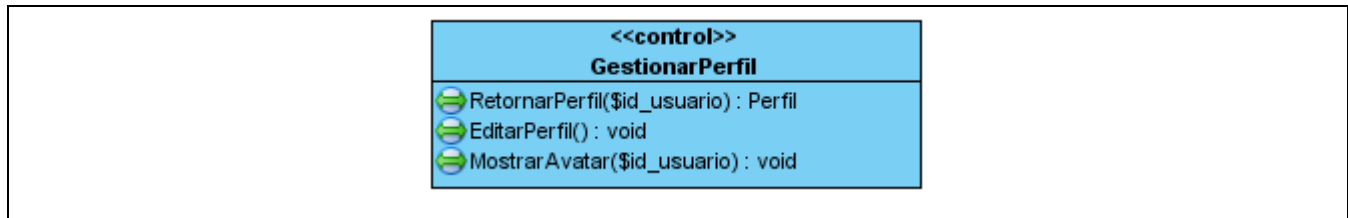


Tabla 48 Clases Controladoras: GestionarPerfil

Nombre: GestionarMensaje	
Tipo de clase (Controladora)	
Para cada responsabilidad:	
Nombre:	EnviarMensaje(\$id_usuario) : void
Descripción:	Se utiliza para enviar mensajes, al contacto del sistema de teleconsulta online.
Nombre:	MostrarMensaje(\$id_usuario) : void
Descripción:	Se utiliza para mostrar los mensajes en la ventana del mensaje.
Nombre:	MostrarHistorial(\$dir, \$nombre_archivo, \$ nombre_carpeta) : void
Descripción:	Se utiliza para ver el historial de los mensajes enviados por los contactos por fecha y hora.
Nombre:	BorrarHistorial() : void
Descripción:	Se utiliza para eliminar los mensajes enviados por los contactos por fecha y hora.
<div style="border: 1px solid black; padding: 5px; background-color: #e0f0ff; margin: 0 auto; width: 80%;"> <p style="text-align: center; margin: 0;"><<control>> GestionarMensaje</p> <ul style="list-style-type: none"> EnviarMensaje(\$id_usuario) : void MostrarMensaje(\$id_usuario) : void MostrarHistorial(\$dir, \$nombre_archivo, \$ nombre_carpeta) : void BorrarHistorial() : void </div>	

Tabla 49 Clases Controladoras: GestionarMensaje

Nombre: GestionarPizarra	
Tipo de clase (Controladora)	
Para cada responsabilidad:	
Nombre:	CompartirImagen(\$id_usuario) : void

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

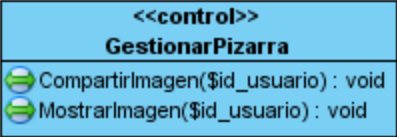
Descripción:	Se utiliza para abrir la ventana compartida que podrá ser vista tanto por el médico como por el contacto que el mismo haya seleccionado y para cargar la imagen.
Nombre:	MostrarImagen(\$id_usuario) : void
Descripción:	Se utiliza para mostrar la imagen seleccionada en la imagen compartida.
 <pre> classDiagram class GestionarPizarra { <<control>> CompartirImagen(\$id_usuario) : void MostrarImagen(\$id_usuario) : void } </pre>	

Tabla 50 Clases Controladoras: GestionarPizarra


Nombre: GestionarImagen	
Tipo de clase (Controladora)	
Para cada responsabilidad:	
Nombre:	AmpliarImagen():void
Descripción:	Se utiliza para poner la imagen más grande para obtener una vista mejor de ella.
Nombre:	MoverImagen():void
Descripción:	Se utiliza para desplazar la imagen por la pantalla
Nombre:	DibujarImagen():void
Descripción:	Se utiliza para dibujar una parte de la imagen seleccionada por el usuario o el contacto.
Nombre:	GuardarImagen():void
Descripción:	Se utiliza para guardar la imagen en la computadora
 <pre> classDiagram class GestionarImagen { <<control>> AmpliarImagen() : void MoverImagen() : void DibujarImagen() : void GuardarImagen() : void } </pre>	

Tabla 51 Clases Controladoras: GestionarImagen

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

➤ Clases Entidad

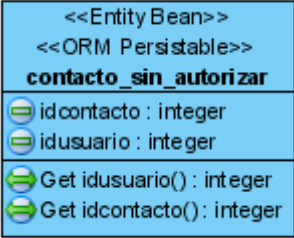
Nombre: (contacto_sin_autorizar)	
Tipo de clase (Entidad)	
Atributo	Tipo
idcontacto	integer
idusuario	integer
Para cada responsabilidad:	
Nombre:	Get idcontacto () : integer
Descripción:	Retorna el id del contacto.
Nombre:	Get idusuario() : integer
Descripción:	Retorna el id del usuario.
 <pre> classDiagram class contacto_sin_autorizar { <<Entity Bean>> <<ORM Persistable>> idcontacto : integer idusuario : integer Get idusuario() : integer Get idcontacto() : integer } </pre>	

Tabla 52 Clases Entidades: contacto_sin_autorizar

Nombre: (contacto_autorizado)	
Tipo de clase (Entidad)	
Atributo	Tipo
idcontacto	integer
idusuario	integer
idgrupo	integer
Para cada responsabilidad:	
Nombre:	Get idcontacto () : integer
Descripción:	Retorna el id del contacto.
Nombre:	Get idusuario() : integer

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

Descripción:	Retorna el id del usuario.
Nombre:	Get idgrupo () : integer
Descripción:	Retorna el id del grupo.
	

Tabla 53 Clases Entidades: contacto_authorized

Nombre: (grupo)	
Tipo de clase (Entidad)	
Atributo	Tipo
idgrupo	integer
nombregroupo	varchar
idusuario	integer
Para cada responsabilidad:	
Nombre:	Get idgrupo () : integer
Descripción:	Retorna el id del grupo.
Nombre:	Get nombregroupo() : integer
Descripción:	Retorna el nombre del grupo.
Nombre:	Get idusuario () : integer
Descripción:	Retorna el id del usuario.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

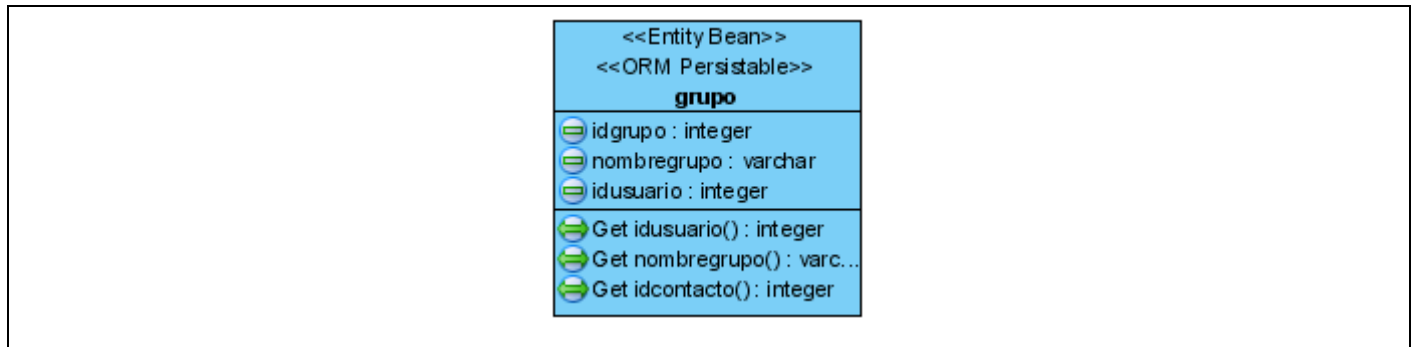


Tabla 54 Clases Entidades: grupo

Nombre: (perfil)	
Tipo de clase (Entidad)	
Atributo	Tipo
idvcard	integer
nick	varchar
email	integer
movil	varchar
fax	varchar
avatar	varchar
idusuario	integer
Para cada responsabilidad:	
Nombre:	Get idvcard():integer
Descripción:	Retorna el id de vcard.
Nombre:	Get nick():varchar
Descripción:	Retorna el nick del médico.
Nombre:	Get email():varchar
Descripción:	Retorna el email del médico.
Nombre:	Get movil():varchar
Descripción:	Retorna el movil del médico.
Nombre:	Get fax():varchar

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

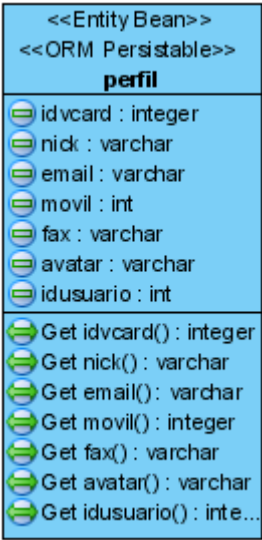
Descripción:	Retorna el fax del médico.
Nombre:	Get avatar():varchar
Descripción:	Retorna el avatar del médico.
Nombre:	Get idusuario():integer
Descripción:	Retorna el id del usuario.
 <pre> classDiagram class perfil { idvcard : integer nick : varchar email : varchar movil : int fax : varchar avatar : varchar idusuario : int Get idvcard() : integer Get nick() : varchar Get email() : varchar Get movil() : integer Get fax() : varchar Get avatar() : varchar Get idusuario() : integer } </pre>	

Tabla 55 Clases Entidades: perfil

Nombre: mensaje	
Tipo de clase (Entidad)	
Atributo	Tipo
idmensaje	integer
idusuario	integer
idcontacto	integer
texto	text
fecha	date
hora	date
Para cada responsabilidad:	
Nombre:	Get idmensaje():integer

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

Descripción:	Retorna el id del mensaje.
Nombre:	Get idusuario():integer
Descripción:	Retorna el id del usuario.
Nombre:	Get idcontacto():integer
Descripción:	Retorna el id del contacto.
Nombre:	Get texto():text
Descripción:	Retorna el texto del mensaje.
Nombre:	Get fecha():date
Descripción:	Retorna la fecha del mensaje.
Nombre:	Get hora():date
Descripción:	Retorna la hora del mensaje.


```

<<Entity Bean>>
<<ORM Persistable>>
mensaje
idmensaje : integer
idusuario : integer
idcontacto : integer
texto : text
fecha : date
hora : date
Get idmensaje() : integer
Get idusuario() : integer
Get idcontacto() : integer
Get texto() : text
Get fecha() : date
Get hora() : date
    
```

Tabla 56 Clases Entidades: mensaje

- **Diagrama de Secuencia**

Un diagrama de secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian de una forma ordenada según su secuencia en el tiempo.

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

Diagrama de Secuencia

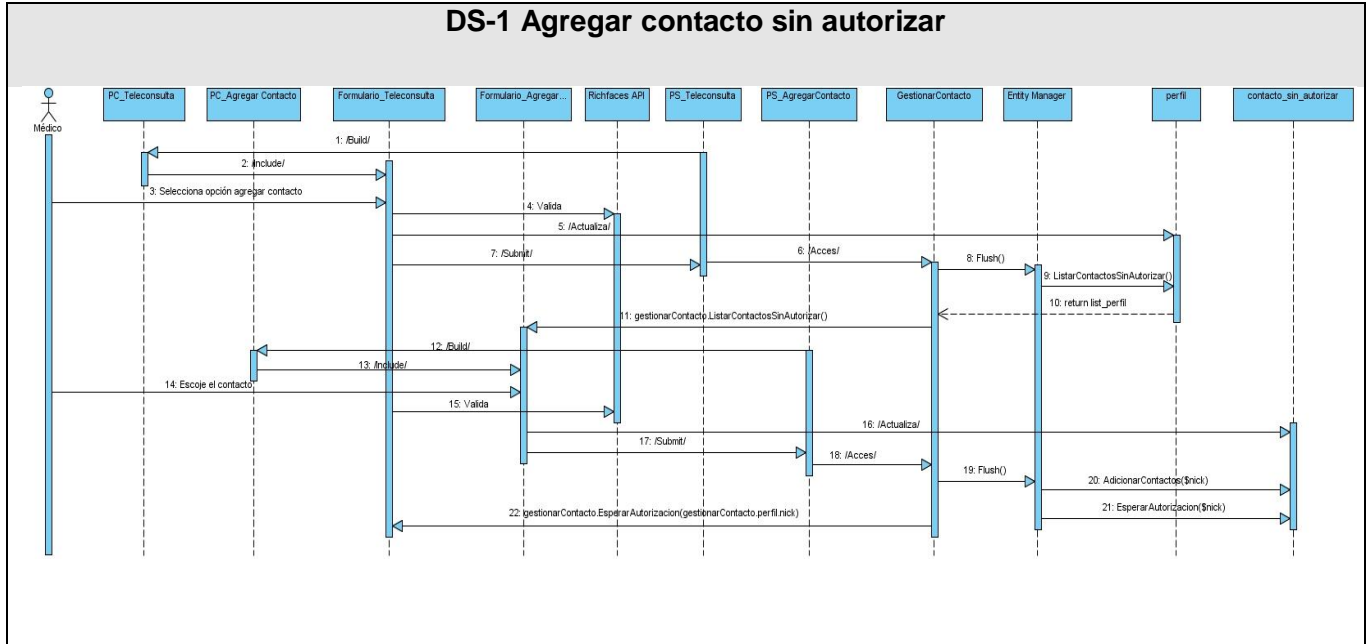


Figura 31 Diagrama de Secuencia: Agregar contacto sin autorizar

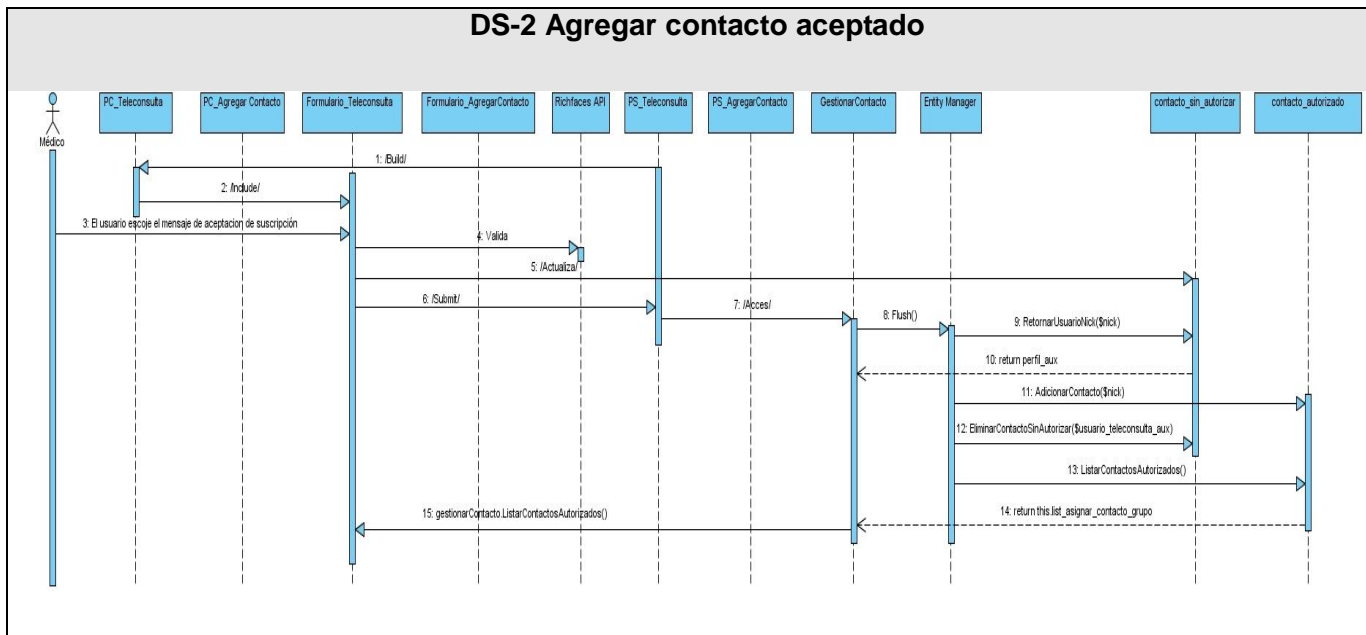


Figura 32 Diagramas de Secuencia: Agregar contacto aceptado

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

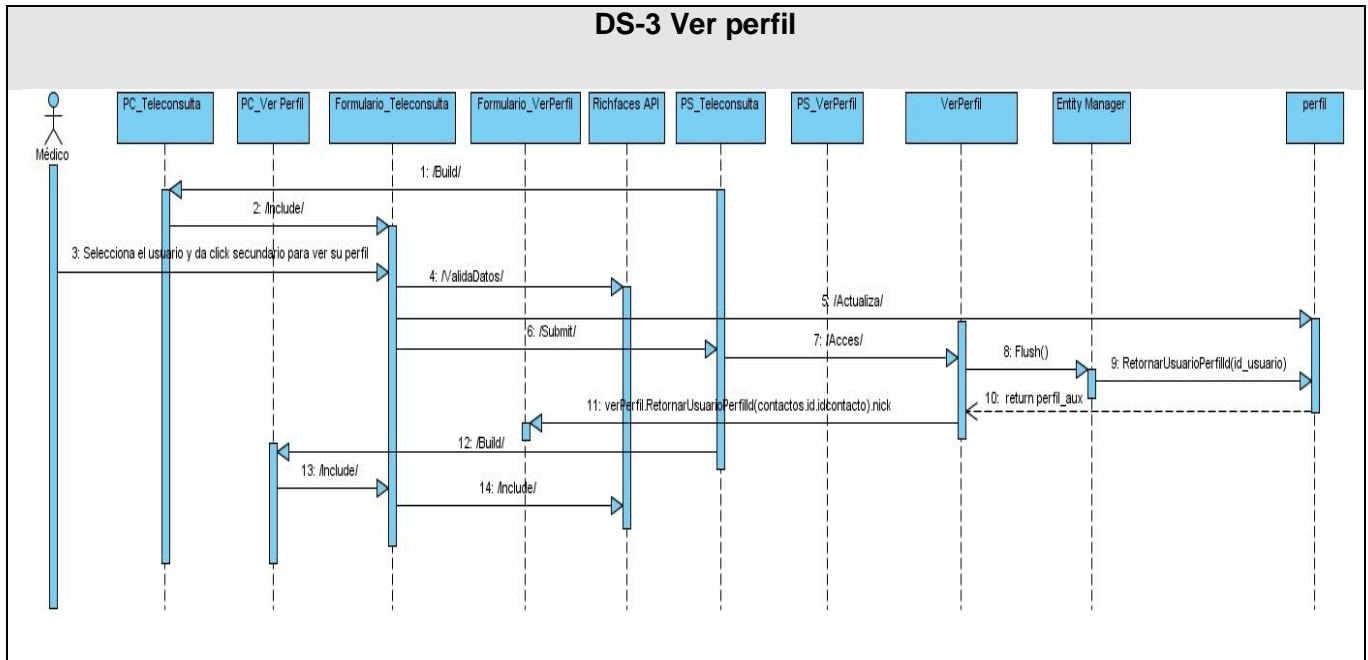


Figura 33 Diagrama de Secuencia: Ver perfil

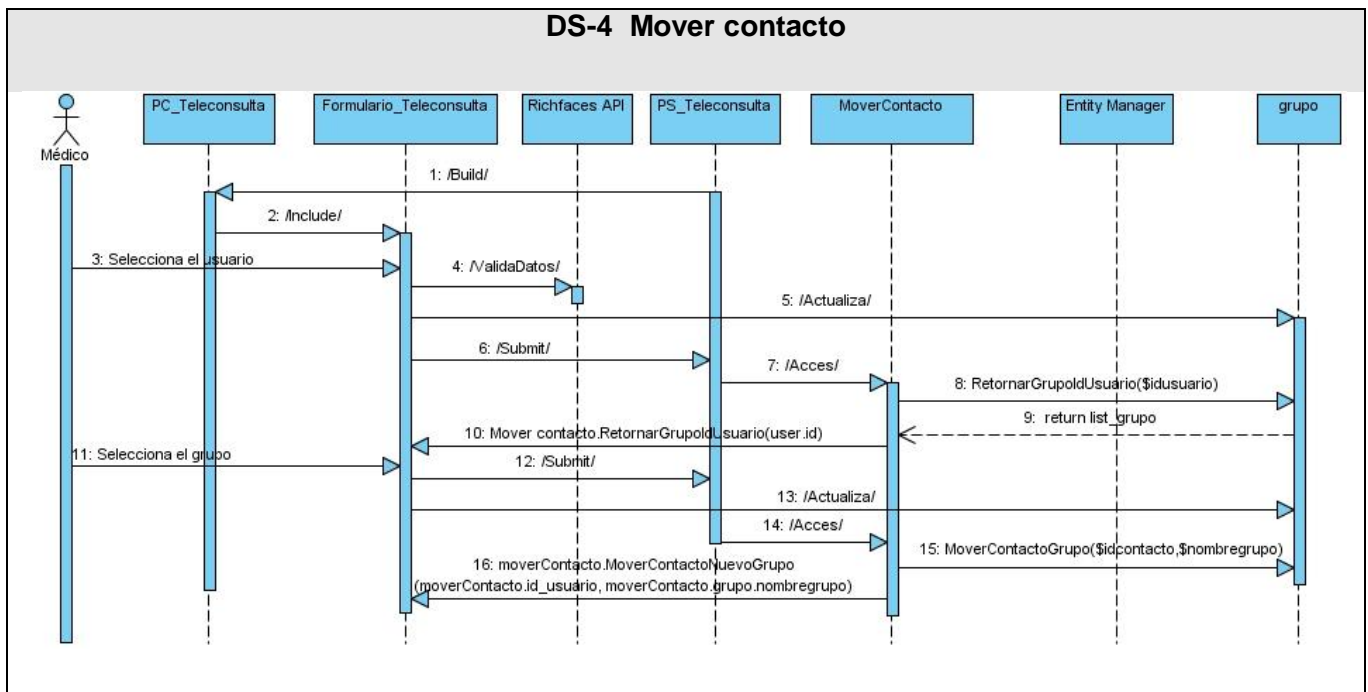


Figura 34 Diagrama de Secuencia: Mover contacto

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

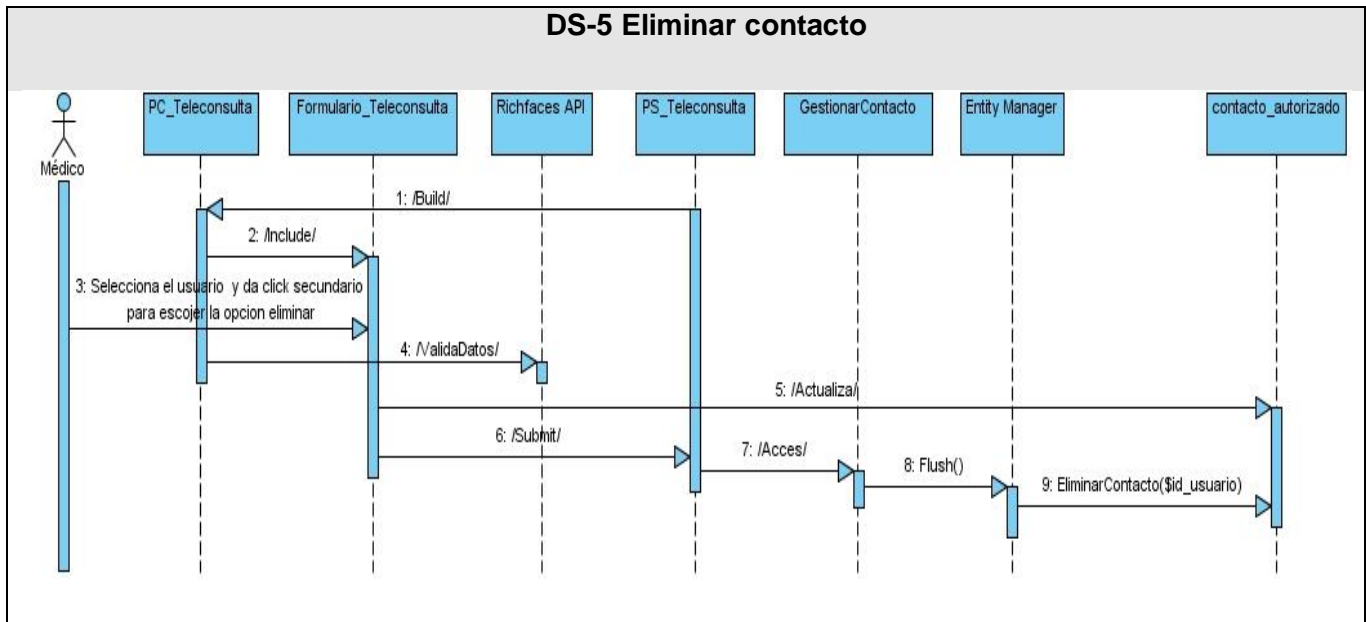


Figura 35 Diagrama de Secuencia: Eliminar contacto

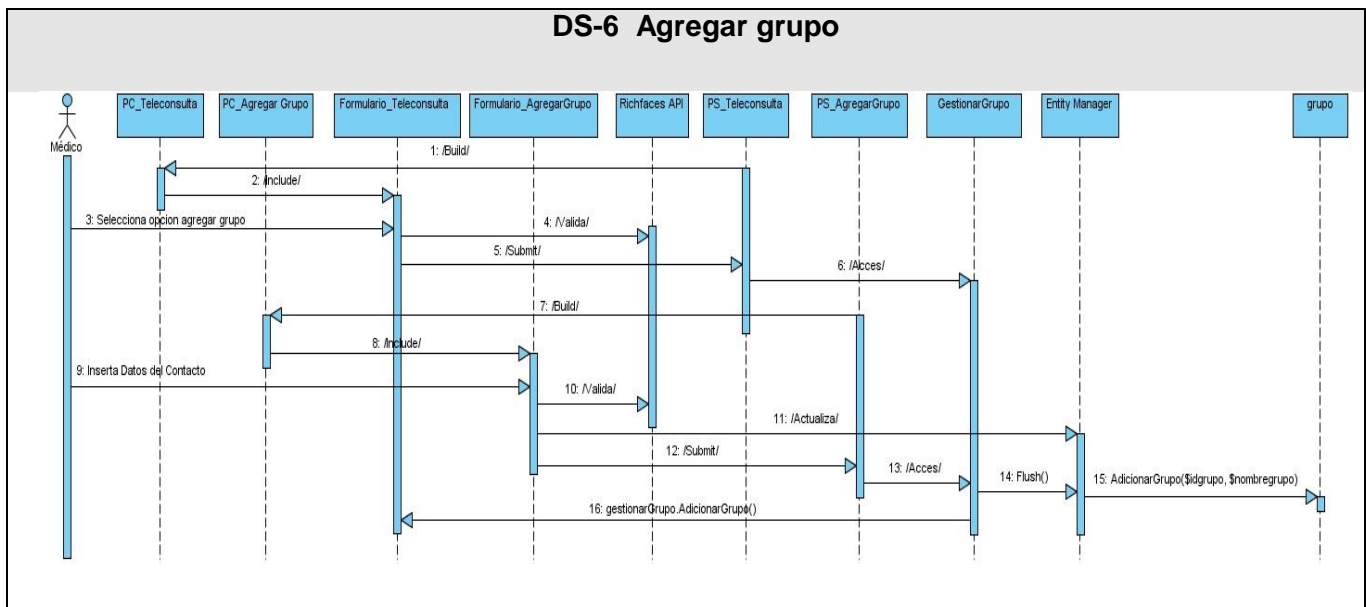


Figura 36 Diagrama de Secuencia: Agregar grupo

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

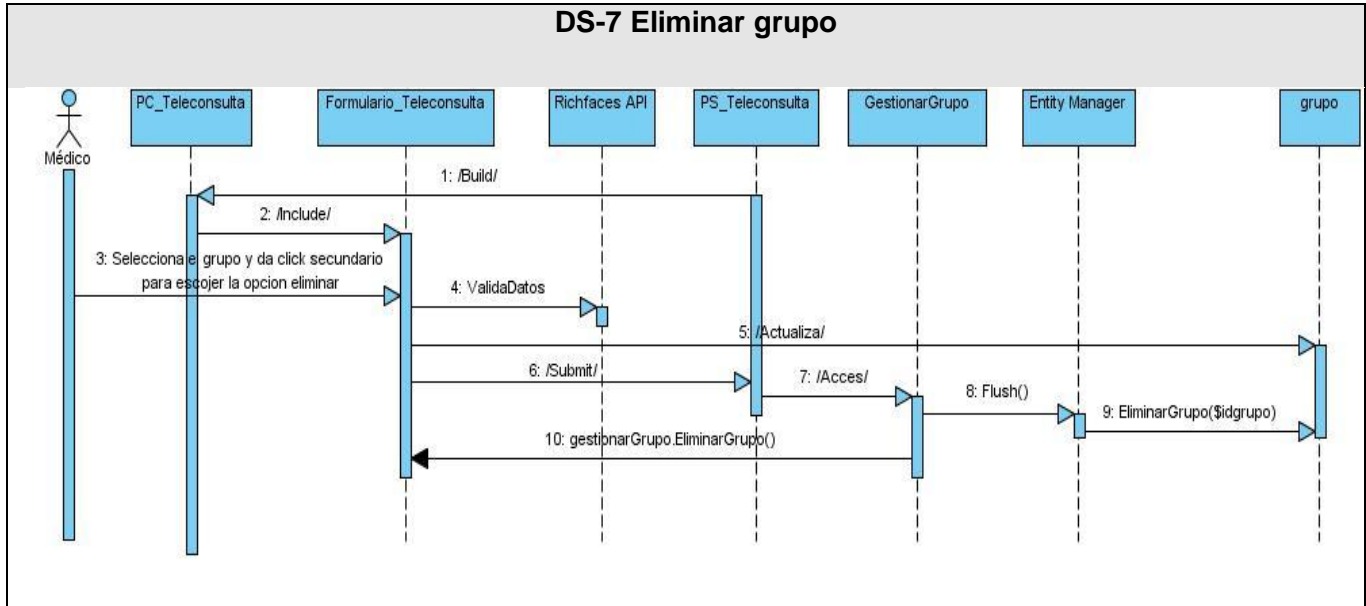


Figura 37 Diagrama de Secuencia: Eliminar grupo

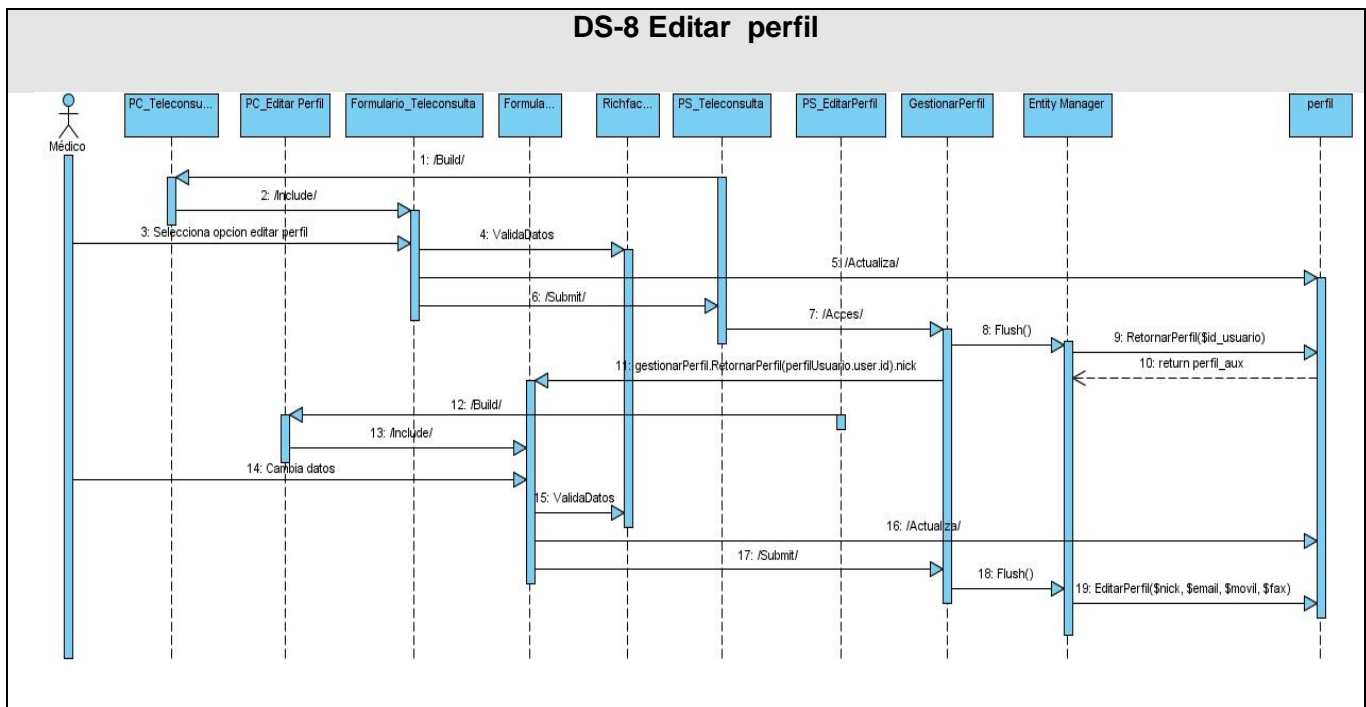


Figura 38 Diagrama de Secuencia: Editar perfil

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

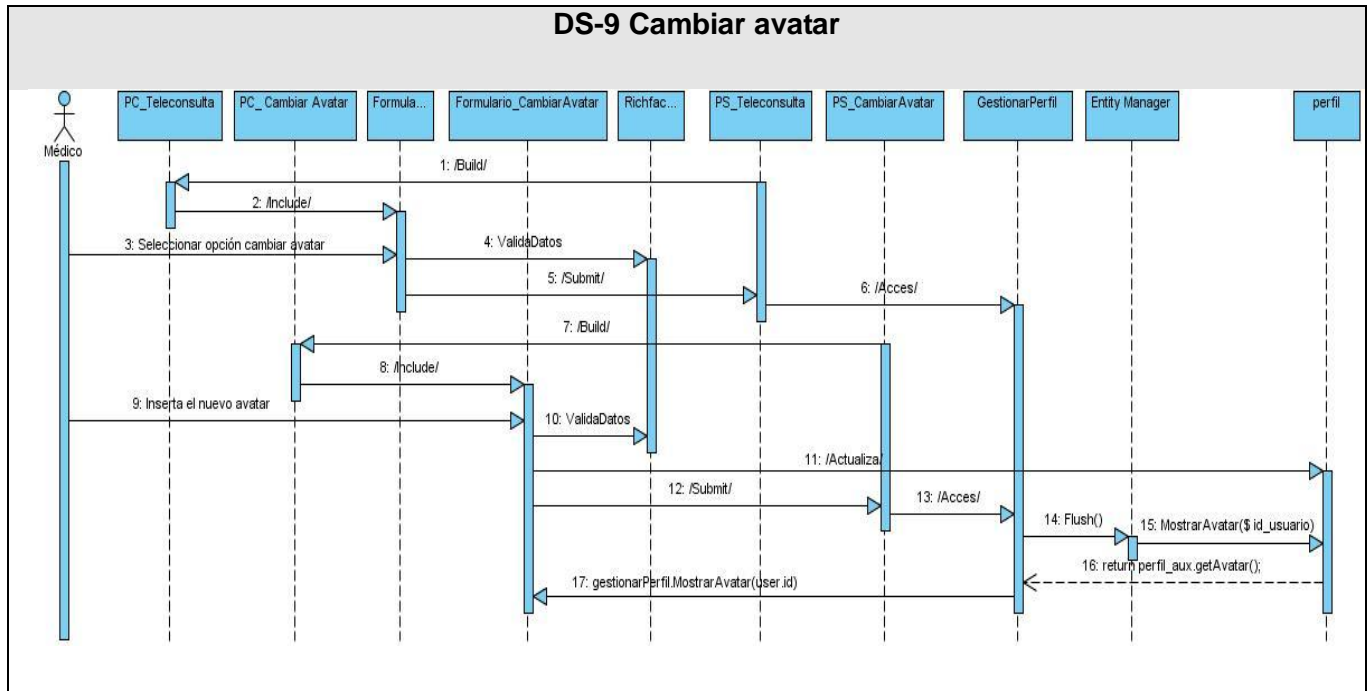


Figura 39 Diagrama de Secuencia: Cambiar avatar

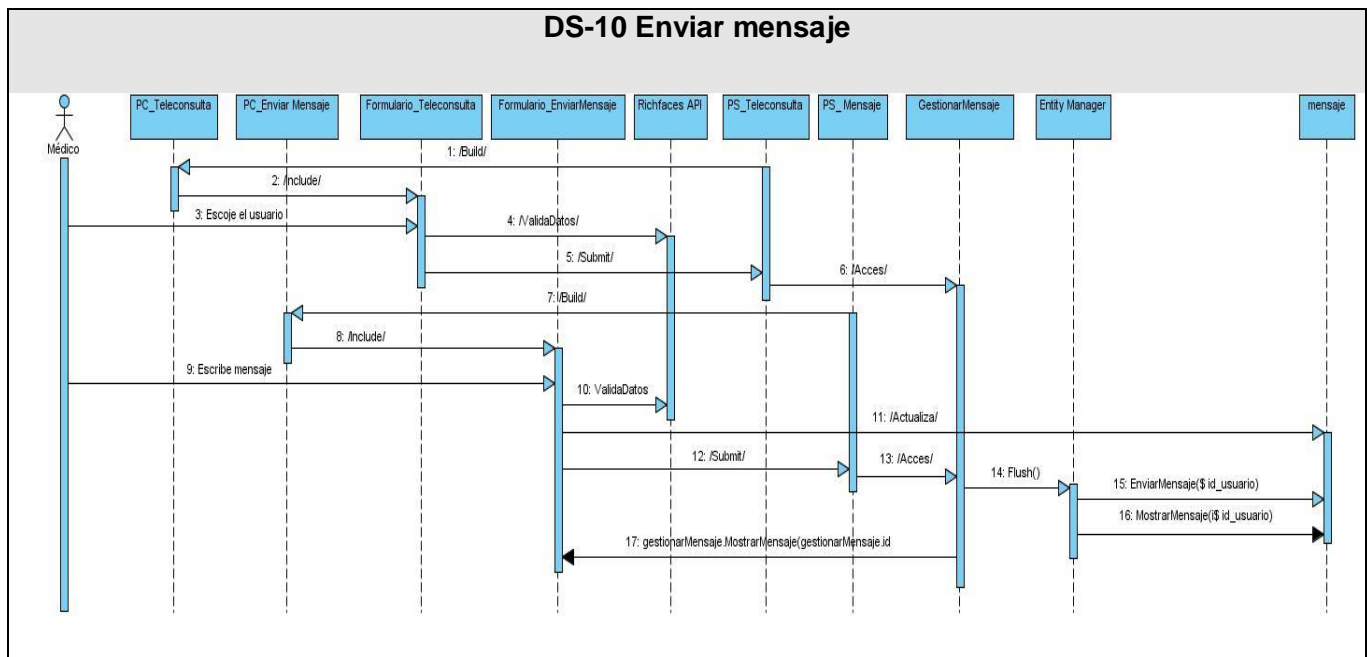


Figura 40 Diagrama de Secuencia: Enviar mensaje

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

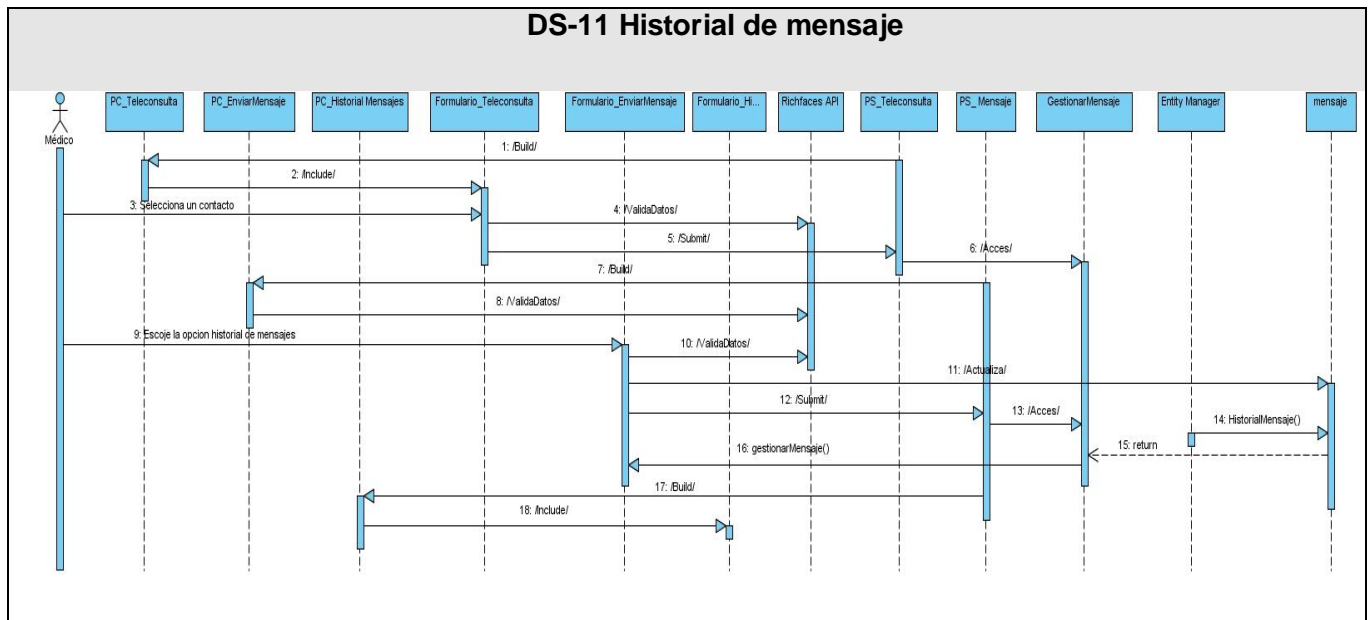


Figura 41 Diagrama de Secuencia: Historial de mensajes

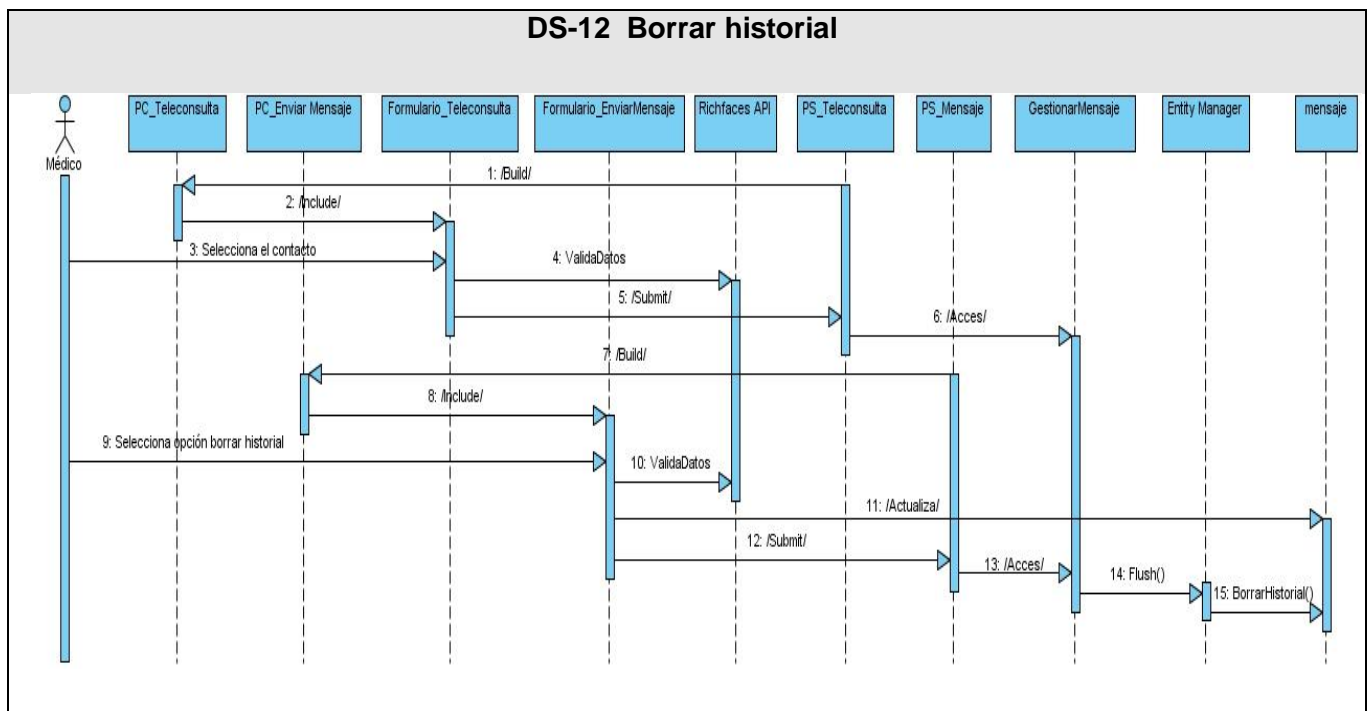


Figura 42 Diagrama de Secuencia: Borrar historial

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

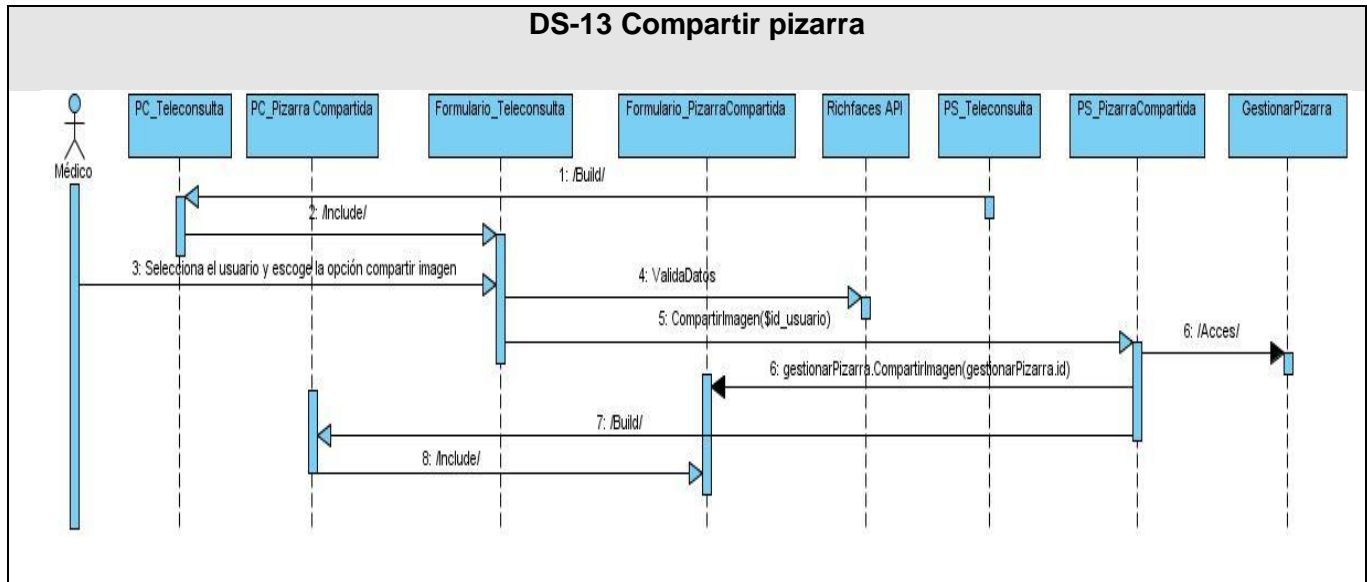


Figura 43 Diagrama de Secuencia: Compartir pizarra

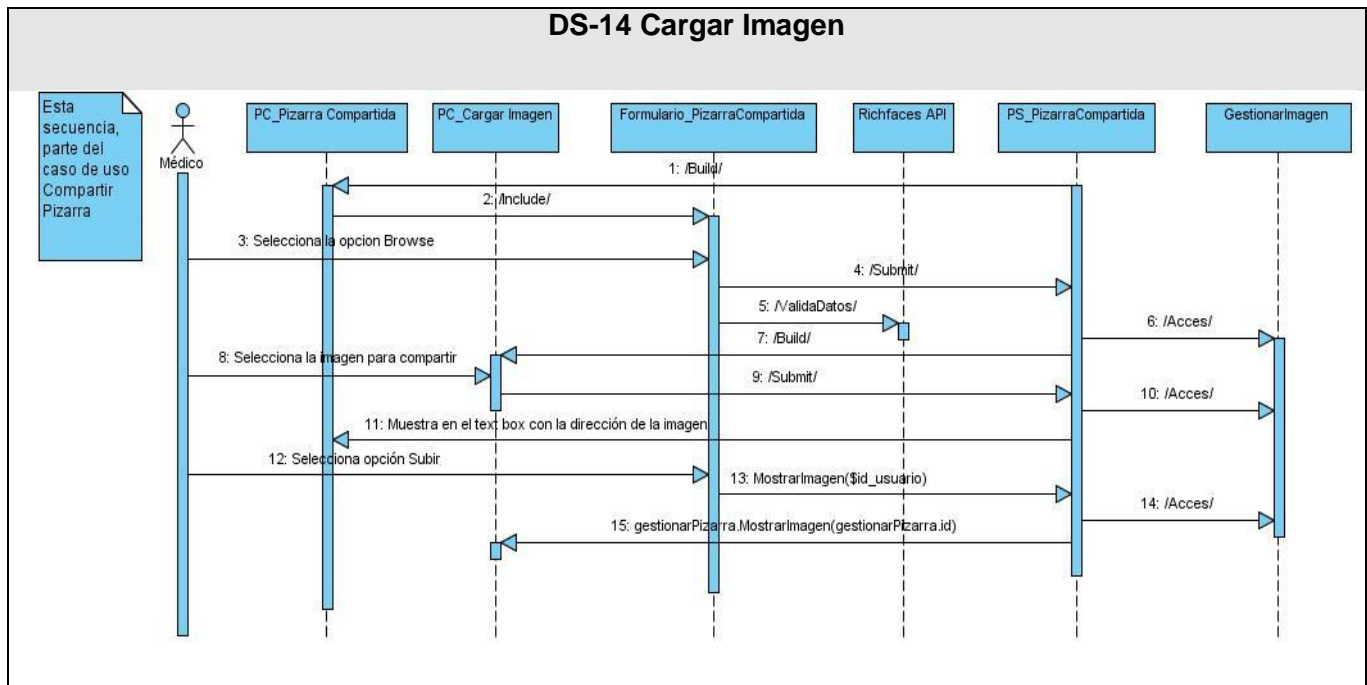


Figura 44 Diagrama de Secuencia: Cargar Imagen

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

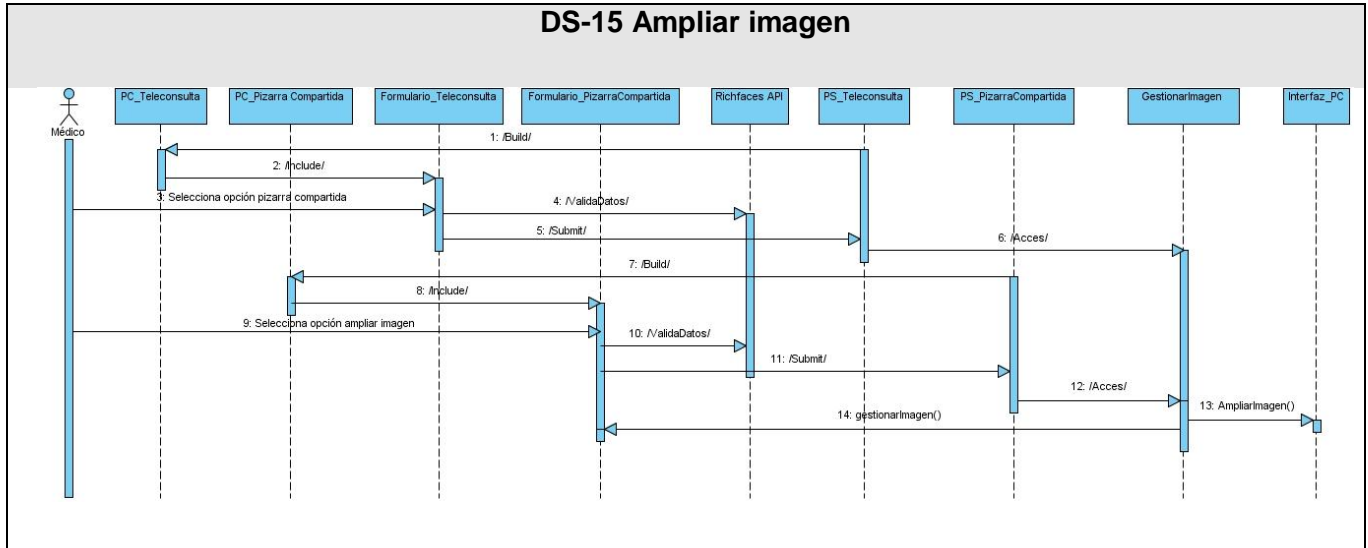


Figura 45 Diagrama de Secuencia: Ampliar Imagen

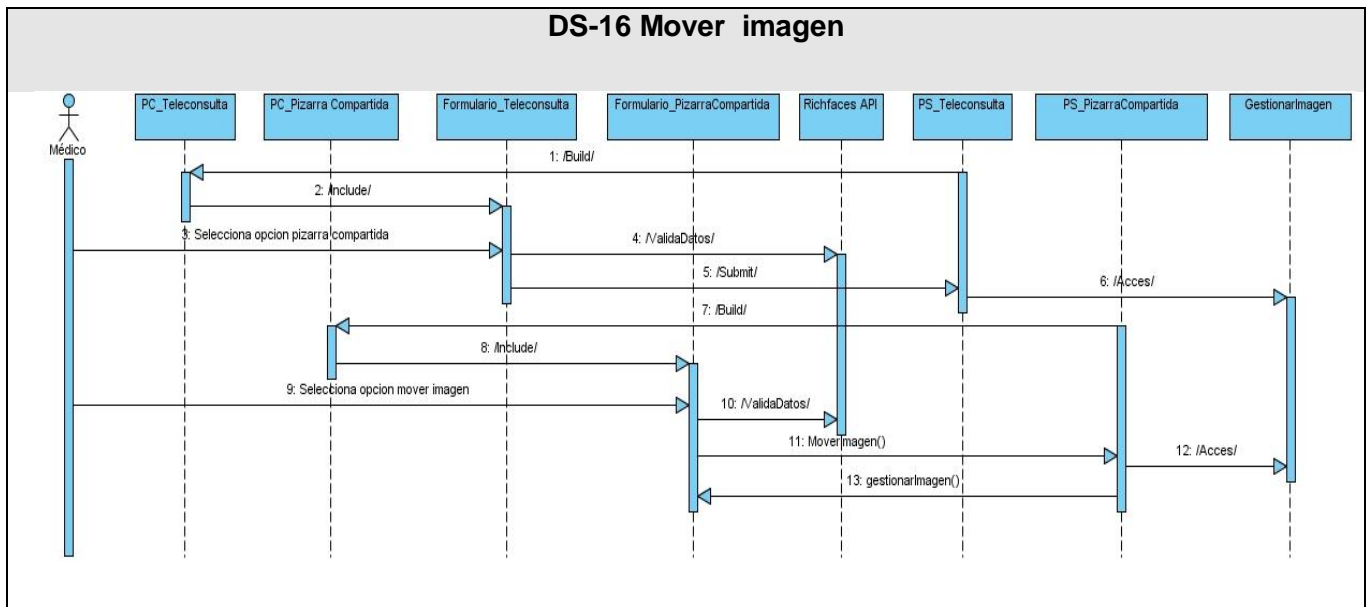


Figura 46 Diagrama de Secuencia: Mover imagen

CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA

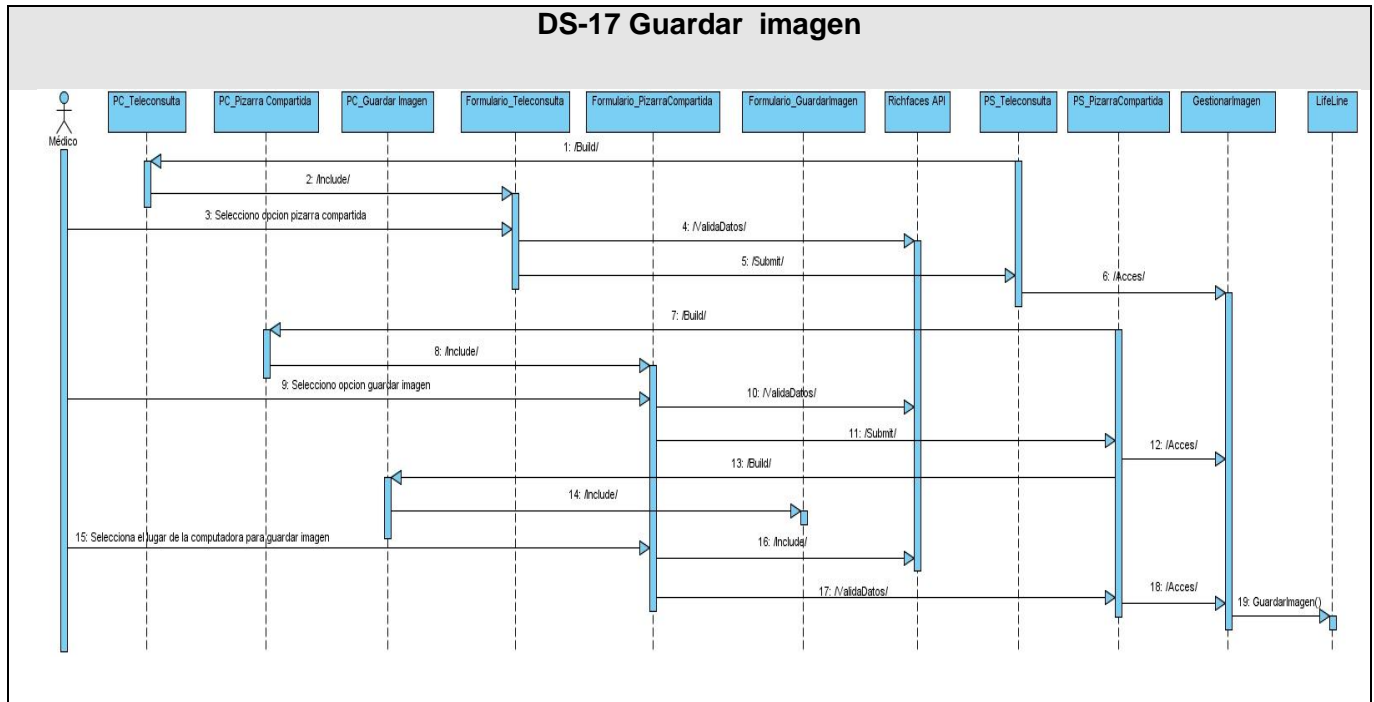


Figura 47 Diagramas de Secuencia: Guardar imagen

Conclusión:

En este capítulo se identificó el tipo de arquitectura con que se contará para el desarrollo del mismo: la arquitectura de pizarra, en 3 capas y la arquitectura basada en el patrón Modelo-Vista-Controlador. Se realizó el análisis y la colaboración, para una mejor comprensión de la programación, el diseño y los diferentes diagramas de secuencia que muestra la secuencia del flujo de mensajes en la programación.

Capítulo IV: Implementación

El siguiente capítulo describe todo lo relacionado con la implementación del sistema de acuerdo a la arquitectura propuesta por el diseño. Se representa la base de datos utilizada a través de un modelo de datos, se modelan los diagramas de componentes y de despliegue, lo que posibilitaría que quedara conformado el modelo de implementación. Además, se explica como se realiza el tratamiento de errores, la seguridad y los estándares y estilos utilizados.

4.1 Modelo de Datos

Para realizar el modelo de datos del sistema, se hizo necesario diseñar un diagrama de clases persistentes. Con la realización de este modelo se logra definir la persistencia de los datos, para lograr que mantengan su valor en el espacio y en el tiempo.

- Diagrama de Clases Persistentes.

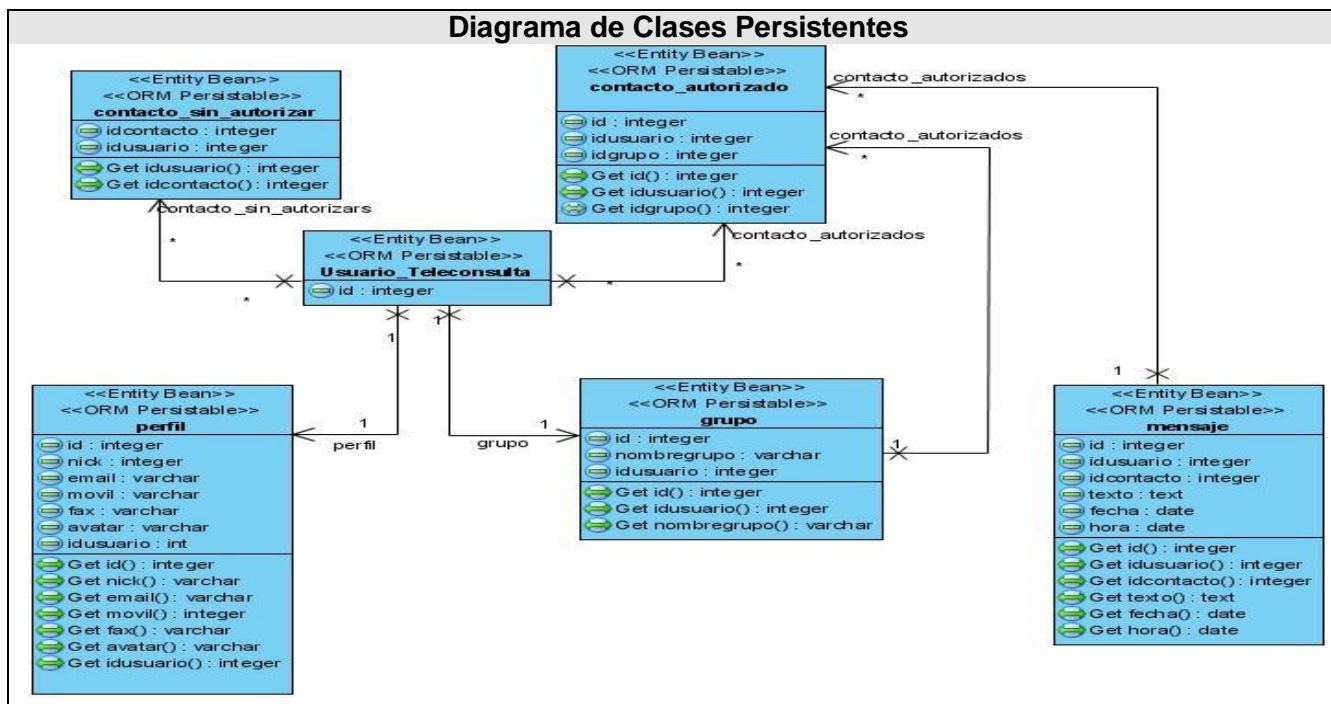


Figura 48 Diagrama de Clases Persistentes

CAPÍTULO IV: IMPLEMENTACIÓN

- **Modelo de Datos**

Una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. [30]

El modelo de datos permite describir el tipo de datos que incluye la base de datos y la forma en que se relacionan. Brinda la condición en que lo datos deben estar para cumplir su objetivo y las diferentes operaciones que se realizan sobre ellos.

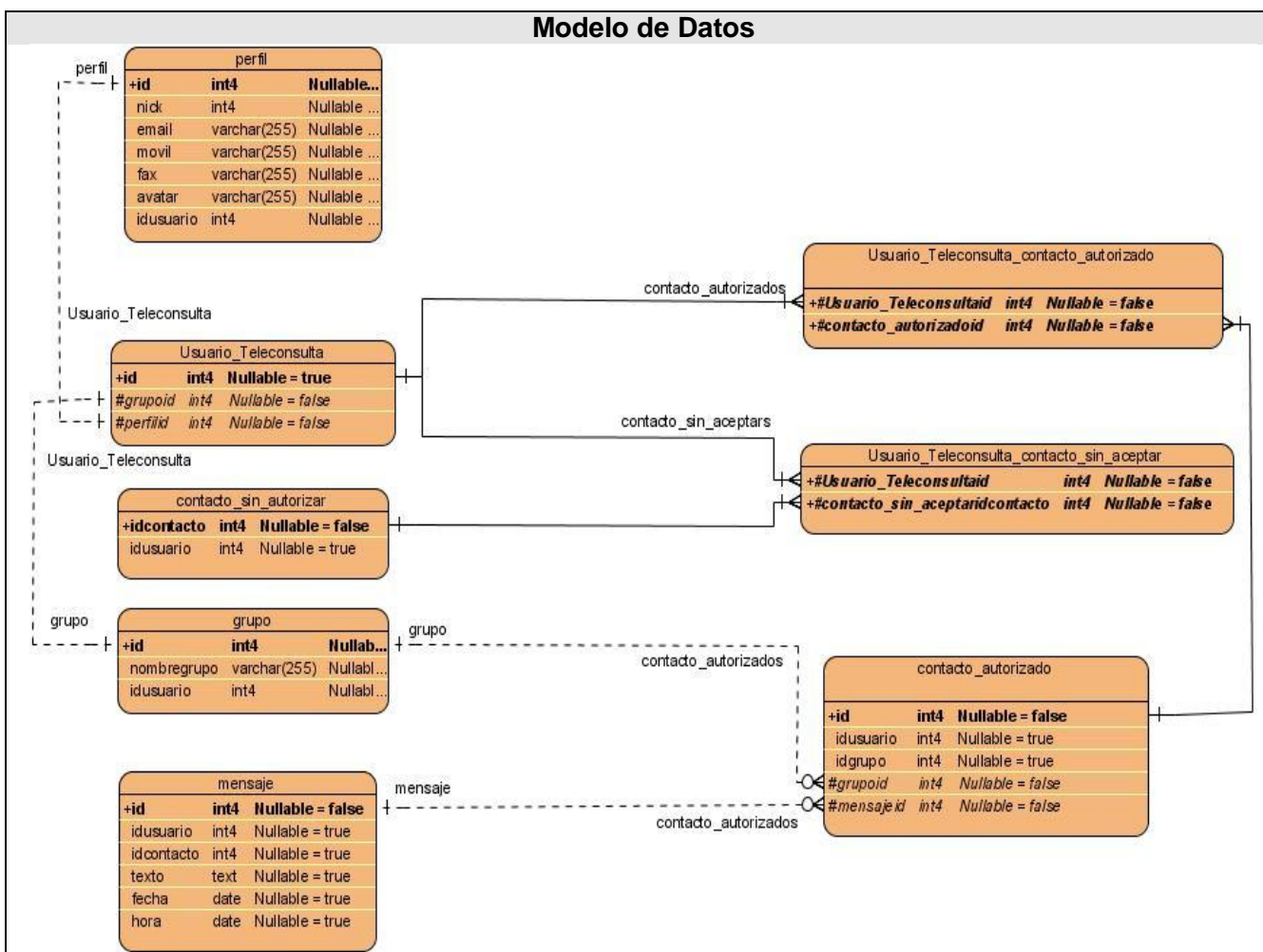


Figura 49 Modelo de Datos

CAPÍTULO IV: IMPLEMENTACIÓN

- Descripción de las tablas.

Nombre: contacto_sin_autorizar		
Descripción: Tabla que almacenará los datos de todos los contactos que todavía no han sido aceptados por el usuario.		
Atributo	Tipo	Descripción
idcontacto	integer	Llave primaria, es el id del contacto no autorizado.
idusuario	integer	Representa al id de usuario.

Tabla 57 Tablas de la BD: contacto_sin_autorizar

Nombre: contacto_autorizado		
Descripción: Tabla que almacenará los datos de todos los contactos que han sido aceptados por el usuario.		
Atributo	Tipo	Descripción
id	integer	Llave primaria, es el id del contacto autorizado.
idusuario	integer	Representa al id de usuario.
idgrupo	integer	Representa al id del grupo al que pertenece el contacto.

Tabla 58 Tablas de la BD: contacto_autorizado

Nombre: grupo		
Descripción: Tabla que almacenará los datos de los grupos para agrupar contactos según las preferencias del usuario.		
Atributo	Tipo	Descripción
id	integer	Llave primaria, es el id del grupo y es auto incremento.
nombregroupo	varchar	Representa al nombre del grupo.
idusuario	integer	Representa al id del usuario.

Tabla 59 Tablas de la BD: grupo

CAPÍTULO IV: IMPLEMENTACIÓN

Nombre:perfil		
Descripción: Tabla que almacenará los datos personales del usuario.		
Atributo	Tipo	Descripción
id	integer	Llave primaria, es el id de perfil y es auto incremento.
nick	varchar	Representa al nombre que identificará al usuario.
email	integer	Representa la dirección de correo.
movil	varchar	Representa el número del móvil del médico.
fax	varchar	Representa al número del fax del usuario.
avatar	varchar	Representa a la imagen que identificará al usuario.
idusuario	integer	Representa al id del usuario.

Tabla 60 Tablas de la BD: perfil

Nombre:mensaje		
Descripción: Tabla que almacenará los mensajes enviados al usuario.		
Atributo	Tipo	Descripción
id	integer	Llave primaria, es el id de mensaje y es auto incremento.
idusuario	integer	Representa al id del usuario.
idcontacto	integer	Representa al id del contacto.
texto	text	Representa el cuerpo del mensaje.
fecha	date	Representa la fecha del mensaje.
hora	date	Representa la hora del mensaje.

Tabla 61 Tablas de la BD: mensaje

Nombre:Usuario_Teleconsulta
Descripción: Tabla que sirve de intermediaria entre la base de datos del Sistema de Gestión Hospitalaria y la del Sistema de Teleconsulta para poder obtener el id del usuario y otros datos importantes.

CAPÍTULO IV: IMPLEMENTACIÓN

Atributo	Tipo	Descripción
id	integer	Llave primaria, es el id de Usuario_Teleconsulta y es auto incremento.

Tabla 62 Tablas de la BD: Usuario_Teleconsulta

4.2 Implementación.

En la implementación se empezó con el resultado del diseño, por tanto, esta parte del desarrollo del sistema lo refina y describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Los diagramas de despliegue y componentes que se representan a continuación, son artefactos que se generan en este flujo de trabajo, los cuales conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

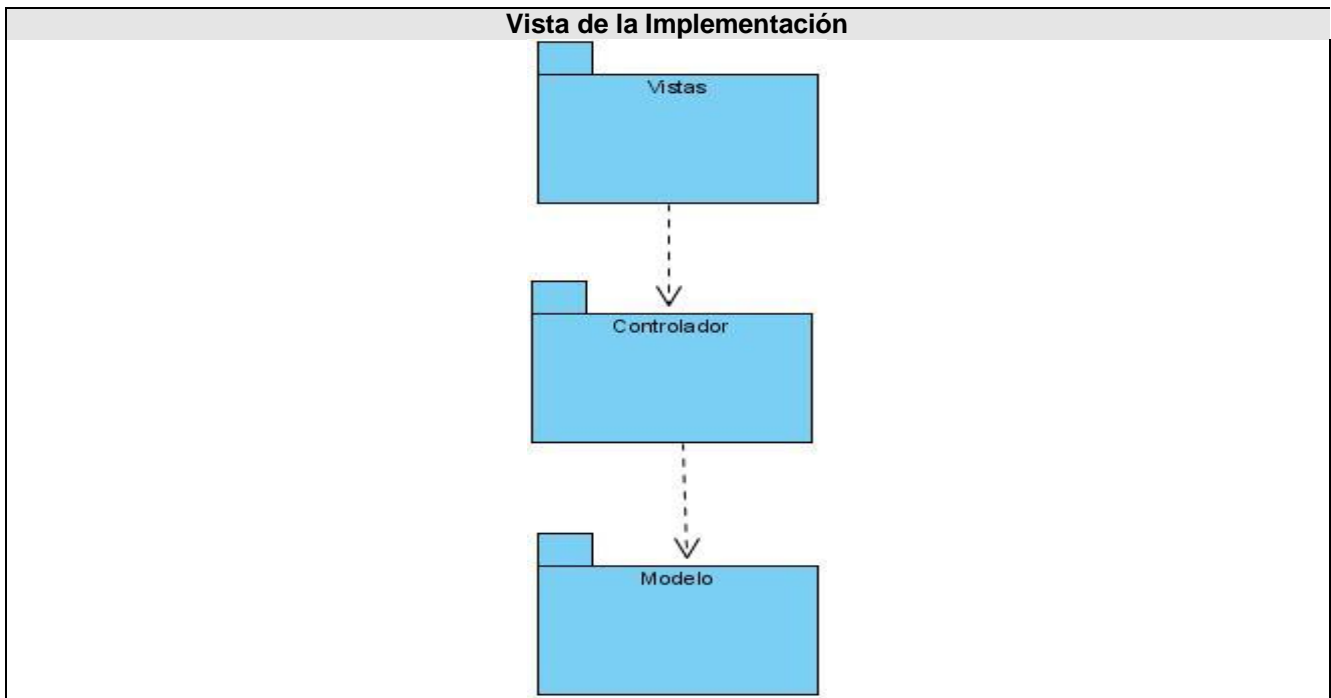


Figura 50 Vista de Implementación

CAPÍTULO IV: IMPLEMENTACIÓN

• Diagrama de despliegue

Modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. [31]

Nodo PC Cliente

Representa las computadoras que utilizarán los usuarios para acceder a la aplicación. Establece comunicación con el servidor de aplicaciones a través del protocolo http.

Nodo Servidor de Aplicaciones JbossServer

Representa el servidor donde estará la aplicación, este nodo mantendrá comunicación con otros nodos:

- Con el nodo Servidor de Base de Datos a través del protocolo ADO.
- Con el nodo Servidor Red5 a través del protocolo ADO.

Nodo Servidor de Base de Datos PostgreSQL 8.3

Representa al servidor donde estará la base de datos de la aplicación.

Nodo Servidor Red5

Representa el servidor que brindará el servicio de la transmisión de imágenes del sistema de teleconsulta.

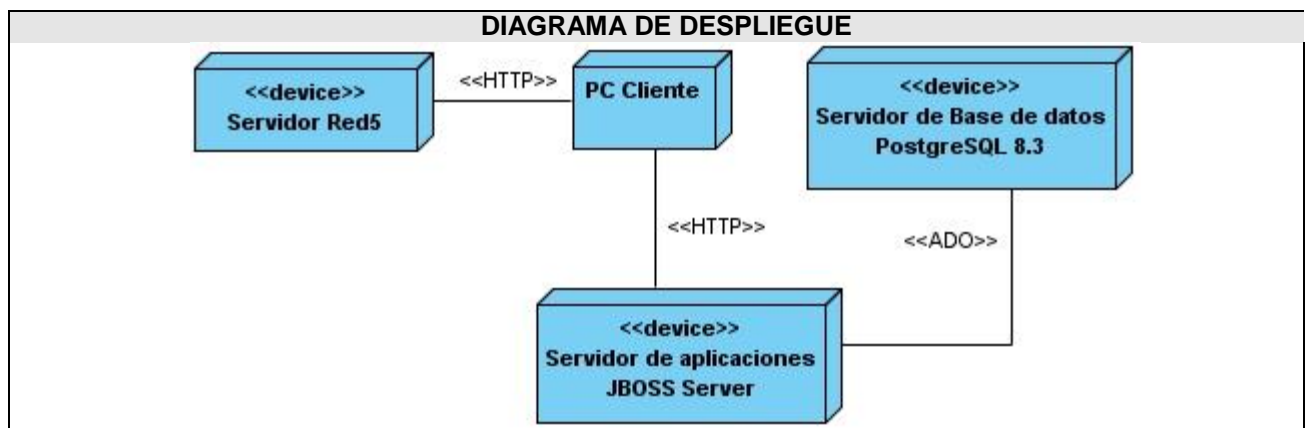


Figura 51 Diagrama de Despliegue

CAPÍTULO IV: IMPLEMENTACIÓN

- Diagrama de componentes.

Muestra un conjunto de componentes y sus relaciones. Se utilizan para describir la vista de implementación estática de un sistema. [32]

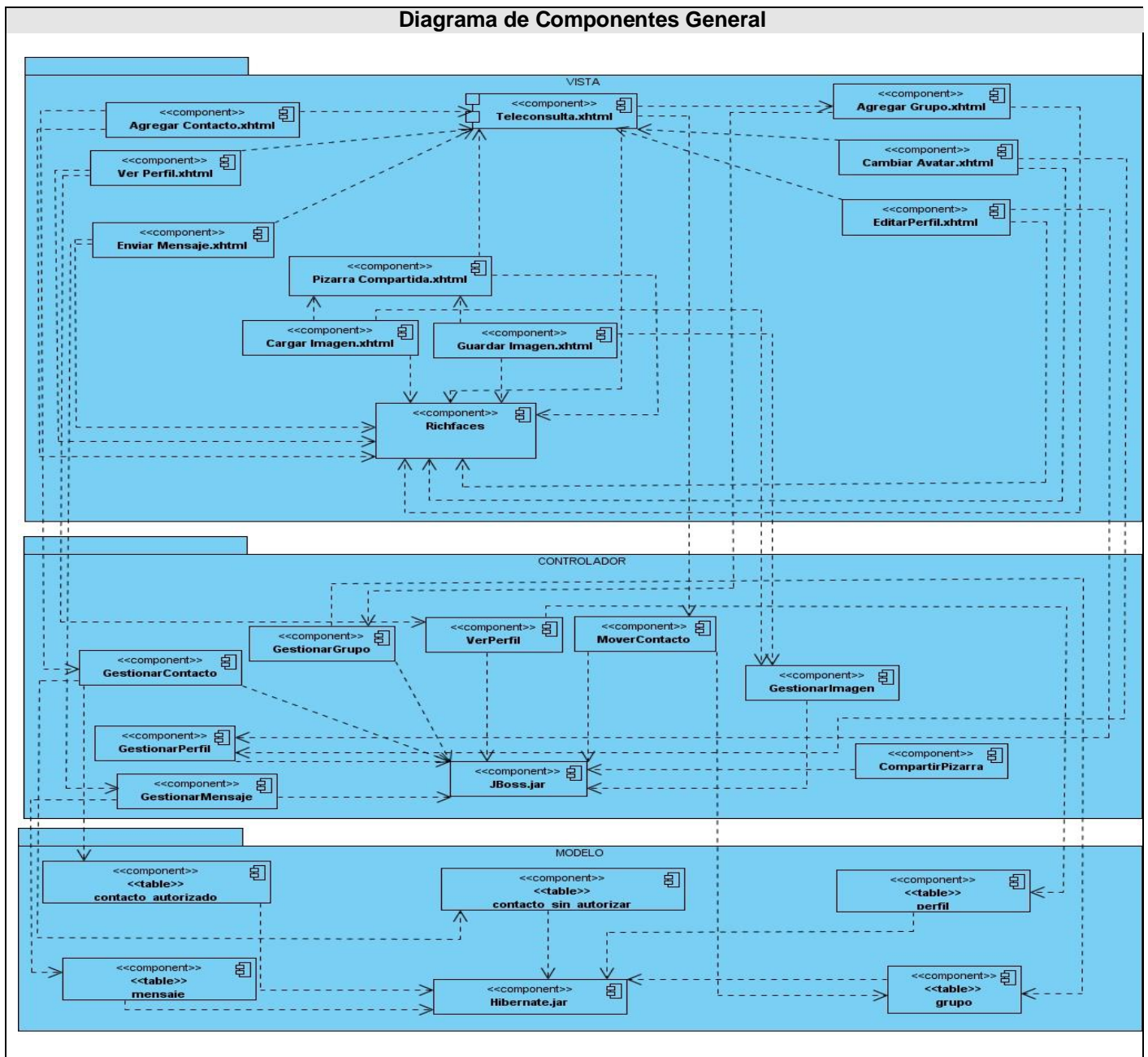


Figura 52 Diagrama de Componentes General

CAPÍTULO IV: IMPLEMENTACIÓN

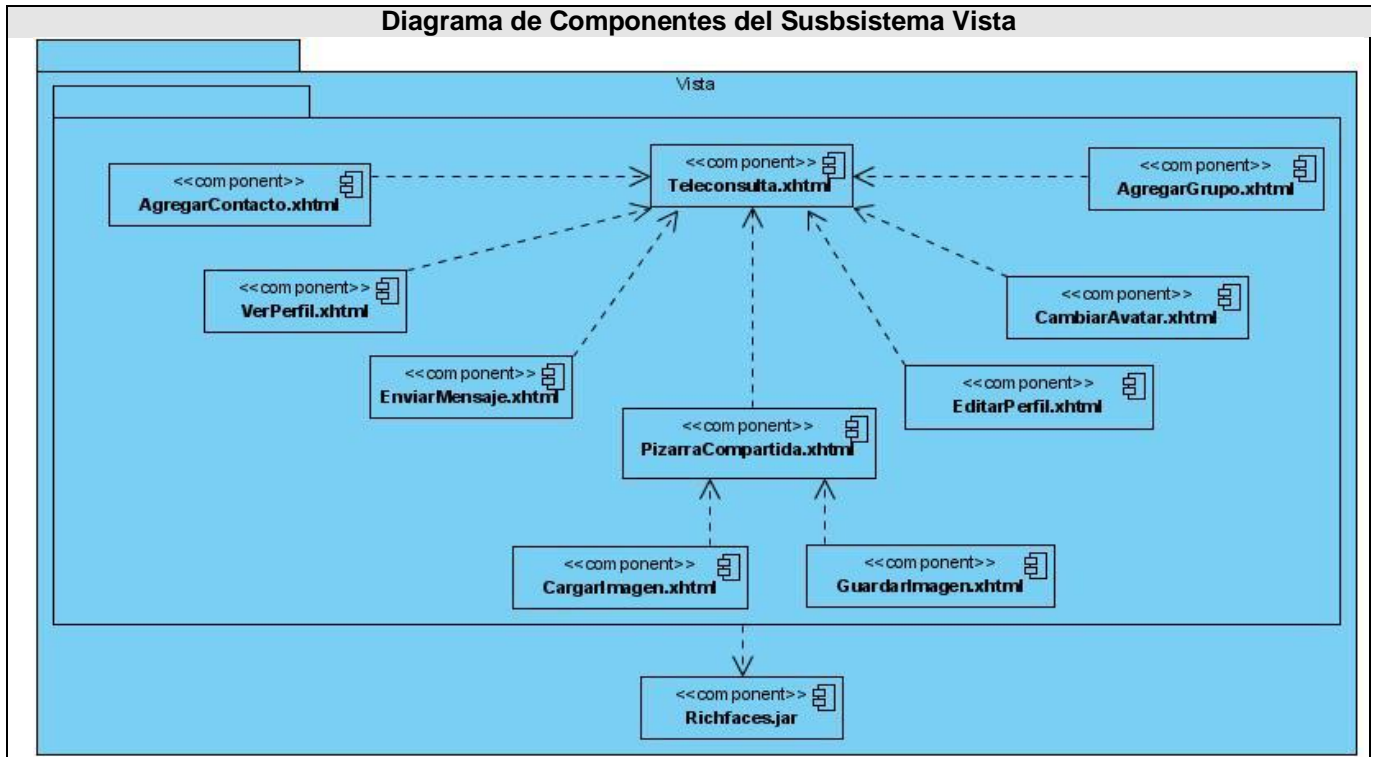


Figura 53 Diagrama para las Clases Vistas de Implementación

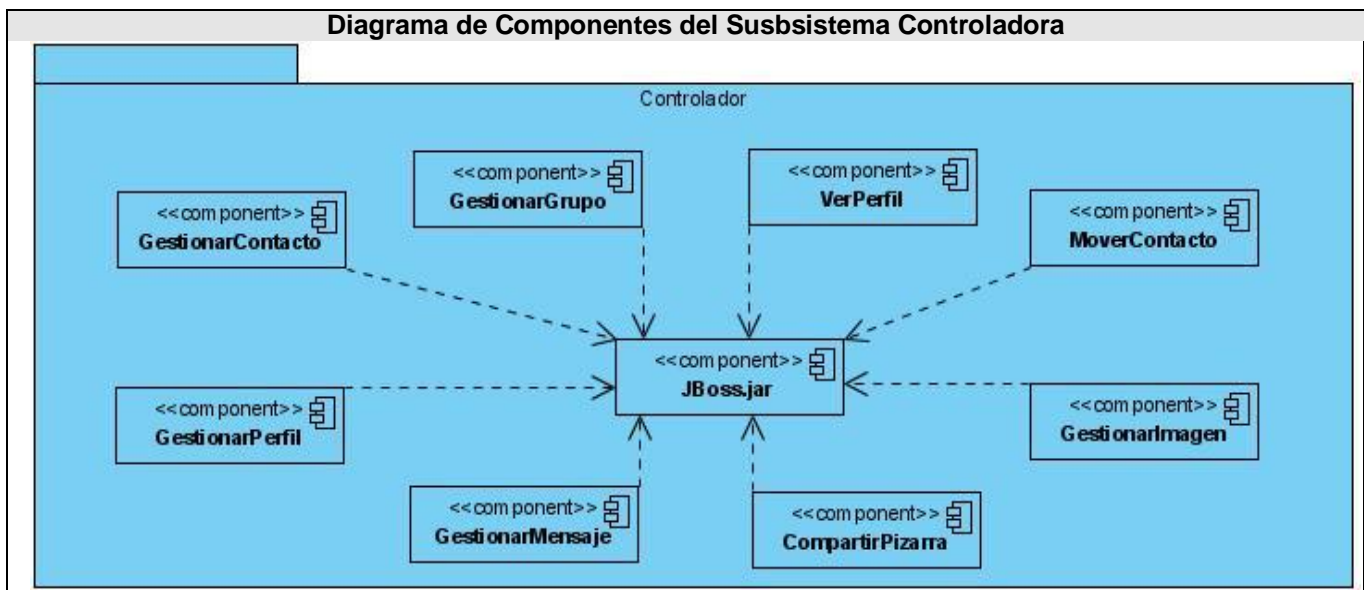


Figura 54 Diagrama para las Clases Controladoras de Implementación

CAPÍTULO IV: IMPLEMENTACIÓN

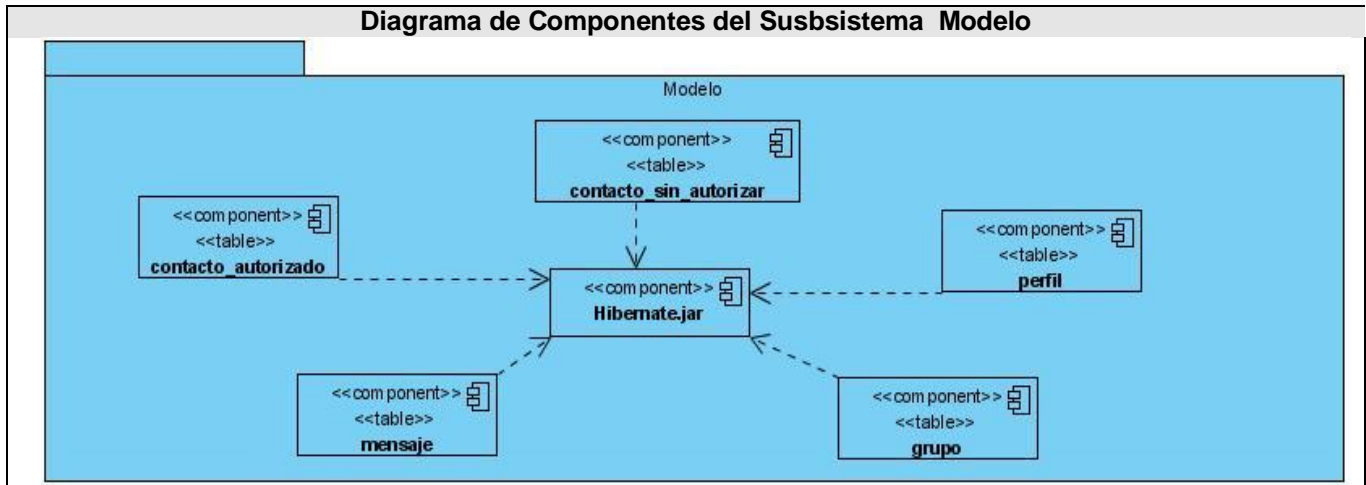


Figura 55 Diagrama para las Entidades de Implementación

- **Tratamiento de errores**

El tratamiento de errores posibilita el buen funcionamiento de una aplicación dándole una mejor apariencia ante los usuarios. En este sistema se trataron los errores de forma tal que las interacciones con la base de datos (inserción, eliminación, modificación y otros), se realicen de forma correcta. Para lograr esto se establecieron mecanismos de validación que comprueban la corrección en el manejo de los datos. En el caso de la entrada de datos por parte del usuario, se implementaron funciones que validen dicha entrada para que, de existir errores, se muestren mensajes que ilustren la incorrecta inserción, modificación o mala manipulación de datos en general.

Ejemplo:

```
public Perfil RetornarPerfilNick(String nick)
{
    Perfil perfil_aux = new Perfil ();
    try{
        perfil_aux = (Perfil) entityManager.createQuery("select p from Perfil p where p.nick = :nick")
            .setParameter("nick", nick)
            .getSingleResult();
        return perfil_aux;
    }
```

CAPÍTULO IV: IMPLEMENTACIÓN

```
    }  
    catch (Exception e) {  
        return null;  
    }  
}
```

- **Seguridad**

La seguridad en un sistema, es de gran importancia, ya que de ella depende la integridad, autenticidad y confiabilidad de su información. En el caso del sistema en cuestión, la seguridad es llevada a cabo por el proyecto ALAS-HIS. Para esto se definieron diferentes tipos de seguridad: acceso al sistema, registro de trazas, administración de seguridad (vista lógica y vista física) y configuración de funcionalidades.

Acceso al sistema: Se definieron diferentes roles que permiten un nivel de acceso distinto para cada uno de ellos, los cuales podrán entrar a los diferentes módulos y podrán usar las funcionalidades de acuerdo a los permisos de su rol, realizado todo esto a través de un usuario y contraseña.

Registro de trazas: Se registran en la base de datos acciones llevadas a cabo por los usuarios como: inicio o cierre de sesión, acceso a un módulo, modificación de un atributo, entidad o cualquier otra operación sobre el sistema.

Administración de seguridad: El sistema brinda la posibilidad de asignar o denegar permiso a los diferentes roles y usuarios, en los módulos y funcionalidades dentro de estos y también, la eliminación de roles y usuarios de las listas de los que se le negó o permitió algún permiso. Todos estos permisos son registrados por el sistema.

Configuración de funcionalidades: Los usuarios del sistema pueden adicionar o eliminar las diferentes funcionalidades y categorías de un módulo en específico.

CAPÍTULO IV: IMPLEMENTACIÓN

- **Estrategias de codificación. Estándares y estilos a utilizar.**

- **Estándares de diseño**

Para el diseño de las vistas del Sistema de Teleconsulta, se tuvo en cuenta el documento: “Pautas de Diseño de Interfaz de Usuario”, del proyecto Sistema de Gestión Hospitalaria (ALAS-HIS). Esto permitirá que exista igualdad a la hora del diseño de acuerdo a lo pautado en este proyecto. A continuación se mencionan algunas pautas que tienen que ver con el sistema:

General:

- En el título de los formularios se pondrá el nombre de la acción en infinitivo, la capitalización es igual a la de las etiquetas.
- Todas las imágenes relacionadas con personas se utilizará la imagen del “desconocido”. [33]

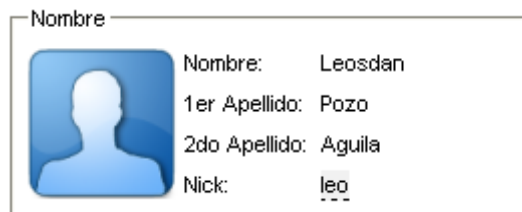


Figura 56 Información del Usuario

Mensajes:

- Los botones de los mensajes deben cumplir las mismas reglas que las de los botones que se utilizan para otras acciones.
- En las ventanas de Información y Error se mostrará solo un botón de “Aceptar”, centrado en la parte inferior. [34]

Botones.

- Los colores están definidos en el estilo CSS.
- El tamaño será el estándar del IDE.

CAPÍTULO IV: IMPLEMENTACIÓN

- Aparecerán alineados en la parte inferior a la derecha y el orden será de tal forma que las acciones positivas al flujo sea de izquierda a derecha. [35]

Etiquetas.

- El tamaño y el color están definidos en el estilo CSS.
- Todas las etiquetas estáticas irán en negro (**R0 G0 B0 - #000000**).
- En caso de mostrar información de solo lectura será mostrada de otro color (**R92 G92 B92 - #5c5c5c**) y se pondrá al lado de la etiqueta que lo identifica la cual estará acompañada de dos puntos. [36]

➤ Estándares de codificación

Un estándar de programación permite definir la nomenclatura de las variables, objetos, métodos y funciones, también el orden y legibilidad del código escrito. Siguiendo esta idea, se puede definir 3 partes principales dentro de un estándar de programación:

- Convención de nomenclatura: Como nombrar variables, funciones, métodos, entre otros.
- Convenciones de legibilidad de código: Como indentar el código, entre otros.
- Convenciones de documentación: Como establecer comentarios, archivos de ayuda, entre otros.

Para la codificación se tuvo en cuenta lo pautado por el proyecto ALAS-HIS, en el documento: “Restricciones de nomenclatura y Estándares de Codificación”. La utilización de estos estándares de codificación permitirá a los programadores el mantenimiento del sistema, lo que facilitará que se entienda los códigos nuevos con rapidez y que el sistema tenga éxito donde quiera que se implante.

Idioma

Se debe utilizar como idioma el español, las palabras no se acentuarán. [37]

CAPÍTULO IV: IMPLEMENTACIÓN

Identación

Inicio y fin de bloque: Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach. [38]

Comentarios, separadores, líneas, espacios en blanco y márgenes

Comentario: Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción. [39]

Espacios en blanco:

- No se debe usar espacio en blanco.
- Después del corchete abierto y antes del cerrado de un arreglo.
- Después del paréntesis abierto y antes del cerrado.
- Antes de un punto y coma. [40]

Variables y constantes

- **Apariencia de variables:** Las variables tendrán un prefijo para el tipo de datos en minúscula.
- **Apariencia de constantes:** Todas sus letras en mayúscula. [41]

Clases y Objeto

- **Apariencia de clases y objetos:** Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing.
- **Apariencia de atributos:** El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto se empleará notación CamellCasing.

CAPÍTULO IV: IMPLEMENTACIÓN

- **Apariencia de las funciones:** Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Se empleará notación PascalCasing. [42]

Tipos de datos

Control	Prefijo	Ejemplo
Botón	btn	btnAceptar
Etiqueta	lbl	lblNombre
Lista/Menú	mn	mnPrincipal
Campo de Texto	txt	txtFecha
Botón de Opción	bpt	optSexo
Casilla de Verificación	chx	chxBorrar
Casilla de Selección	cbx	cbxSexo

Tabla 63 Tabla para los elementos de diseño de interfaz

En este capítulo se mostraron los resultados obtenidos durante la etapa de implementación, los cuales permitieron que quedaran conformados los diagramas de componentes y despliegue, que representan cómo construir y distribuir el sistema; lo que permite la obtención de un producto que cumpla con las funcionalidades propuestas. Se detallan los métodos de implementación más complejos, así como el tratamiento de errores, los principios de codificación y de diseño empleados.

CONCLUSIONES

Al realizar un estudio de los sistemas más difundidos de teleconsulta existentes en el mundo para la comunicación entre los profesionales médicos, se detectó que ninguno cumplía con las características del Sistema Nacional de Salud y con los requerimientos del sistema a desarrollar.

Después de identificar los conceptos fundamentales del dominio del problema, se determinó la realización del “Sistema de Teleconsulta” a través de las tecnologías y herramientas propuestas por el grupo de arquitectura de la facultad.

Debido a que el Sistema de Teleconsulta va a estar integrado al Sistema de Gestión Hospitalaria, la arquitectura que se utilizó esta basada en la de este sistema y los flujos de trabajo se modelaron según lo que propone el Proceso Unificado de Desarrollo (RUP), donde se obtuvo así los artefactos necesarios.

Se implementó el “Sistema de Teleconsulta”, donde todas sus funcionalidades fueron realizadas con éxito a través de una correcta gestión y validación de errores.

RECOMENDACIONES

- Implementación de una funcionalidad que permita cargar imágenes médicas en la pizarra compartida.
- Implementación de una funcionalidad que permita transmitir video y voz para facilitar el intercambio entre los usuarios.

REFERENCIAS BIBLIOGRÁFICAS

[1] . **Infomed**.SLD.CU. *Telemedicina*. [Online] [Cited: 11 13, 2008.].

<http://www.sld.cu/telemedicina/indice.html>.

[2]. **Ferreres, Luis Alfaro** .CONGRESO-INFO. Ponencias. *Aplicaciones de la Telepatología* [Online]

[Cited: 1 18, 2009.]

<http://www.seap.es/telepatologia/telepatologia10.pdf>.

[3]. **Infomed**. Univirtual. *Universidad Virtual de Salud*. [Online] [Cited: 11 25, 2008.]

<http://www.uvirtual.sld.cu/clinica/index.php3>.

[4] Alarcos-inf-cr. Tema04.*Metodologías de Desarrollo* [Online] [Cited: 12 4, 2008.]

<http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>

[5] **Gallego, Juan Gómez**.Scribd.RUP. *Fundamento de la metodología RUP*

<http://www.scribd.com/doc/297224/RUP> [Online] [Cited: 12 4, 2008.]

[6] Adictosaltrabajo. Tutoriales. *Facelets* [Online] [Cited: 12 4, 2008.]

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=migrateJSF2Facelets>

[7] **JUNTA DE ANDALUCIA**. Juntadeandalucia. *MARCO DE DESARROLLO DE LA JUNTA ANTALUCÍA*

[Online] [Cited: 12 4, 2008.]

<http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Ajax4JSF> [Online] [Cited: 12 4, 2008.]

[8] Willmanchamba. *JBoss-Seam-Framework*. [Online] [Cited: 12 4, 2008.]

<http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>.

[9] **JBoss**.JBoss.Dowlands. *JBPM*. [Online] [Cited: 12 4, 2008.]

<http://www.jboss.org/jbossjbpm/downloads/>

[10] assembla.documents.Documentacion.*Drools* [Online] [Cited: 12 4,2008.]

REFERENCIAS BIBLIOGRÁFICAS

http://www.assembla.com/spaces/FallasII/documents/c_YT_le88r3B0oab7jnrAJ/download/Drools%20Documentacion

[11] **Red Hat Middleware.** Hibernate. *Hibernate*. [Online] [Cited: 12 4, 2008.]

<http://hibernate.bluemars.net>

[12] dcc.uchile.uml.introduccion. *Tutorial de UML* [Online] [Cited: 12 6, 2008.]

<http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>

[13]. **Facyt.** Alfa.facyt.uc.edu. *Presentación UML*. [Online] [Cited: 1 16, 2009.]

<http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones20062007/presentacion%20UML.pdf>.

[14] Web Taller.com. Lenguajes. Java.Lecciones. *Que es Java* [Online] [Cited: 1 18, 2009.]

<http://www.webtaller.com/construccion/lenguajes/java/lecciones/que-es-java.php>

[15]. **Barzanallaga, Rafael.** Um. *CASE-principales*. [Online] [Cited: 1 18, 2009.]

http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html.

[16] Freedownloadmanager. Sitio de descargas de software. *Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) 6.0* [Online] [Cited: 1 18, 2009.]

[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).

[17] **Iván García Puebla.** Redksr. *Eclipse: una herramienta profesional al alcance de todos*. [Online] [Cited: 1 18, 2009.]

<http://www.redksr.com/~talos/JAVA/JAVA3D/eclipse.pdf>.

[18] Netpecos. Docs. Mysql_postgres. *Postgres* [Online] [Cited: 1 18, 2009.]

http://www.netpecos.org/docs/mysql_postgres/x15.html#AEN17

[19] **Guerrero, Luis A.** DCC.UCHILE *Aydoo Análisis y Diseño Orientado a Objetos* [Online] [Cited: 1 18, 2009.]

<http://www.dcc.uchile.cl/~luguerre/cc61j/recursos/aydoo.ppt>

REFERENCIAS BIBLIOGRÁFICAS

[20] **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software*. La Habana: Félix Varela, 2004.

[21] Ídem a la referencia 20.

[22] Ídem a la referencia 20.

[23]. **Guiarte Multimedia S.L.** Desarrollo Web. *Artículos*. [Online] [Cited: 1 18, 2009.]

<http://www.desarrolloweb.com/articulos/1622.php>.

[24] **Morfeo Project.** Morfeoproject. *Modelos avanzados de comunicación de recursos* [Online] [Cited: 1 18, 2009.]

http://forge.morfeoproject.org/wiki/index.php/D.3.2.2_Modelos_avanzados_de_comunicaci%C3%B3n_de_recursos.

[25] Ídem a la referencia 24.

[26] **.Granada, José Luis.** Uptoline. *Patrón Modelo Vista Controlador*. [Online] [Cited: 1 18, 2009.]

http://www.utplonline.edu.ec/cursos/diretorio/apoio_5918_40589/Patr%F3n%20de%20dise%F1o%20Modelo-Vista-Controlador.pdf.

[27].MIT.OCW.UNIVERSIA.NET LECTURE-12 *Patrones de Diseño* [Online] [Cited: 2 20, 2009]

<http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>.

[28] **Dirección de Tecnologías de la información.** UNCOMA. Ingeniería *.Área Ingeniería de Software* [Online] [Cited: 2 20, 2009.]

http://www.uncoma.edu.ar/secretaria_general/dti/ingenieria.html

[29] **Ministerio del Poder Popular para las Telecomunicaciones y la Informática.** MeRinde. *Modelo de Diseño* [Online] [Cited: 2 20, 2009]

http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=92&Itemid=296.

[30].**Elies.Elies.** *Bases de datos: Modelos de datos* [Online] [Cited: 3 26, 2009.]

<http://elies.rediris.es/elies9/4-2.htm>

REFERENCIAS BIBLIOGRÁFICAS

- [31]. **Sparx Systems**. Sparxsystems. Uml2 Deploymentdiagram. *Diagrama de Despliegue UML 2*
http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html
- [32]. Tvdi.UML.Node22. Diagramas UML [Online] [Cited: 3 26, 2009.]
<http://tvdi.det.uvigo.es/~avilas/UML/node22.html>
- [33] **ALBET, S.A.** Pautas de Diseño de Interfaz de Usuario. *PAUTAS DE DISEÑO DE INTERFAZ DE USUARIO*. La Habana.Universidad de las Ciencias Informáticas.2009
- [34] Ídem a la referencia 33.
- [35] Ídem a la referencia 33.
- [36] Ídem a la referencia 33.
- [37] **ALAS-HIS**. Restricciones de nomenclatura y Estándares de Codificación. *Restricciones de nomenclatura y Estándares de Codificación*. La Habana. Universidad de las Ciencias Informáticas. 2009
- [38] Ídem a la referencia 37.
- [39] Ídem a la referencia 37.
- [40] Ídem a la referencia 37.
- [41] Ídem a la referencia 37.
- [42] Ídem a la referencia 37.

BIBLIOGRAFIA

1. **ALBET, S.A.** IH-SW-DR-087 ALAS-HIS_ Elementos comunes Modelo de casos de uso del sistema. *Modelo de casos de uso del sistema* .La Habana. Universidad de las Ciencias Informáticas. 2009
2. **ALBET, S.A.** Pautas de Diseño de Interfaz de Usuario. *PAUTAS DE DISEÑO DE INTERFAZ DE USUARIO*. La Habana. Universidad de las Ciencias Informáticas. 2009
3. Adictosaltrabajo. Tutoriales. *Facelets* [Online] [Cited: 12 4, 2008.]
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=migrateJSF2Facelets>
4. Alarcos. Tema04. *Metodologías de Desarrollo* [Online] [Cited: 12 4, 2008.]
<http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>
5. **ALAS-HIS.** Restricciones de nomenclatura y Estándares de Codificación. *Restricciones de nomenclatura y Estándares de Codificación*. La Habana. Universidad de las Ciencias Informáticas. 2009
6. Assembla.documents.Documentacion. *Drools* [Online] [Cited: 12 4,2008.]
http://www.assembla.com/spaces/FallasII/documents/c_YT_le88r3B0oab7jnrAJ/download/Drools%20Documentacion
7. **Barzanallaga, Rafael.** Um. *CASE-principales*. [Online] [Cited: 1 18, 2009.]
http://www.um.es/docencia/barzana/IAGP/Enlaces/CASE_principales.html.
8. **Centro Informático Científico de Andalucía.** CICA. *Arquitectura Modelo Vista Controlador*. [Online] [Cited: 1 18, 2009.] http://www.cica.es/formacion/JavaTut/Apendice/arq_mvc.html.
9. **COPLEC.** Coplec. *Java Persistence API (JPA)* [Online] [Cited: 12 6, 2008.]
<http://www.coplec.org/?q=2008/09/14/java-persistence-api-jpa>

BIBLIOGRAFÍA

10. CS.UALBERTA. Colaboración01. *Conceptos básicos en un Diagrama de Colaboración*[Online] [Cited: 2 20, 2009] <http://www.cs.ualberta.ca/~pfiguero/soo/uml/colaboracion01.html>
11. Dcc.Uml.Introduccion. *Tutorial de UML* [Online] [Cited: 12 8, 2008.] <http://www.dcc.uchile.cl/~lpsalinas/uml/introduccion.html>
12. Definicion.de. *Modelo de Datos*. [Online] [Cited: 3 26, 2009.] <http://definicion.de/modelo-de-datos/>
13. **Dirección de Tecnologías de la información**.UNCOMA. Ingeniería .*Área Ingeniería de Software* [Online] [Cited: 2 20, 2009.] http://www.uncoma.edu.ar/secretaria_general/dti/ingenieria.html.
14. **Dr. Juan Urbino**.Apoyo Medico. *Apoyo Medico*. [Online] [Cited: 12 3, 2008.] <http://www.apoyomedico.com/usuarios.shtml>.
15. DSI. Asignaturas. *Módulo 2 Tema 12: Modelo de Implementación: Diagramas de Componentes y Despliegue* [Online] [Cited: 3 26, 2009.] <http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf>
16. Doctorchat. *Doctor Chat*. [Online] [Cited: 12 3, 2008.] <http://doctorchat.wordpress.com/tag/teleconsulta/>.
17. El medico en casa. El medico en casa [Online] [Cited: 12 3, 2008.] <http://www.elmedicoencasa.com/>
18. Elies.Elies. Bases de datos: Modelos de datos [Online] [Cited: 3 26, 2009.] <http://elies.rediris.es/elies9/4-2.htm>
19. Facyt.Alfa.facyt.uc.edu. Presentacion UML. [Online] [Cited: 1 16, 2009.] <http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones20062007/presentacion%20UML.pdf>.
20. Ferreres, Luis Alfaro .CONGRESO-INFO. Ponencias. Aplicaciones de la Telepatología [Online] [Cited: 1 18, 2009.] <http://www.seap.es/telepatologia/telepatologia10.pdf>.
21. Freedownloadmanager. Sitio de descargas de software. Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) 6.0 [Online] [Cited: 1 18, 2009.]

BIBLIOGRAFÍA

[http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_\(M%C3%8D\)_14720_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/)

22. Gallego, Juan Gómez. Scribd.RUP. Fundamento de la metodología RUP. [Online] [Cited: 12 4, 2008.].
http://www.scribd.com/doc/297224/RUP_JKLJKL
23. Generatorfd. Generatorfd. Arquitectura. [Online] [Cited: 1 19, 2009.].
<http://www.generatorfd.com/Arquitectura.aspx> .
24. Granada, Jose Luis. Uptoline. Patron Modelo Vista Controlador. [Online] [Cited: 1 18, 2009.].
http://www.utplonline.edu.ec/cursos/diretorio/apoyo_5918_40589/Patr%F3n%20de%20dise%F1o%20Modelo-Vista-Controlador.pdf.
25. Grupo BT España. Arrakis. Objetivos del diseño en Java. [Online] [Cited: 1 18, 2009.].
<http://www.arrakis.es/~abelp/ApuntesJava/Introduccion.htm#1.1.%20Objetivos%20de%20dise%20de%20Java>.
26. Guerrero, Luis A. DCC.UCHILE Aydoo Análisis y Diseño Orientado a Objetos [Online] [Cited: 2 39, 2009.].
<http://www.dcc.uchile.cl/~luguerre/cc61j/recursos/aydoo.ppt>. KLJ
27. Guiarte Multimedia S.L. Desarrollo Web. Articulos. [Online] [Cited: 1 18, 2009.].
<http://www.desarrolloweb.com/articulos/1622.php>.
28. Health & Doctor. Health & Doctor [Online] [Cited: 12 3, 2008.].
<http://www.healthanddoctor.com/> K
29. Infomed.SLD. Estrategias. [Online] [Cited: 1 20, 2009.].
http://www.sld.cu/sistema_de_salud/estrategias.html.
30. Infomed.SLD.Telemedicina. [Online] [Cited: 11 13, 2008.].
<http://www.sld.cu/telemedicina/indice.html>.
31. Infomed.Univirtual. Universidad Virtual de Salud. [Online] [Cited: 11 25, 2008.].
<http://www.uvirtual.sld.cu/clinica/index.php3>.

BIBLIOGRAFÍA

32. Informatica2009. Sistema-Informatico-De-Genetica-Medica.SISTEMA INFORMÁTICO DE GENÉTICA MÉDICA [Online] [Cited: 26 05, 2009.]
http://74.125.47.132/search?q=cache:WgBVOCXt4vUJ:informatica2009.sld.cu/Members/aclaro/sistema-informatico-de-genetica-medica/at_download/trabajo+SIGM&cd=1&hl=es&ct=clnk&gl=cu
33. JBoss.JBoss.Dowlands. JBPM. [Online] [Cited: 12 4, 2008.] <http://www.jboss.org/jbossjbp/downloads/>
34. Jboss EPA.Jboss.Jbossas. [Online] [Cited: 12 6, 2008.] <http://www.jboss.org/jbossas>.
35. JUNTA DE ANDALUCIA. Juntadeandalucia. MARCO DE DESARROLLO DE LA JUNTA ANTALUCÍA [Online] [Cited: 12 4, 2008.] <http://www.juntadeandalucia.es/xwiki/bin/view/MADEJA/Ajax4JSF>
36. Lycos. Usuarios. Consultas online 24 hrs [Online] [Cited: 12 3, 2008.]
<http://usuarios.lycos.es/siempreunive/yosano.htm>.
37. Lycos.Usuarios.lycos. Uml. [Online] [Cited: 1 16, 2009.] <http://usuarios.lycos.es/ooopere/uml.htm>.
38. Marañón, Gonzalo Alvarez. IEC.CISC. QuesJava. [Online] [Cited: 1 18, 2009.]
<http://www.iec.csic.es/CRIPTONOMICON/java/quesjava.html>.
39. Ministerio del Poder Popular para las Telecomunicaciones y la Informática. MeRinde. Modelo de Diseño [Online] [Cited: 2 20, 2009]
http://merinde.rinde.gob.ve/index.php?option=com_content&task=view&id=92&Itemid=296.
40. MIT.OCW.UNIVERSIA.NET LECTURE-12 Patrones de Diseño [Online] [Cited: 2 20, 2009]
<http://mit.ocw.universia.net/6.170/6.170/f01/pdf/lecture-12.pdf>
41. Netpecos. Docs. Mysql_postgres.Postgres [Online] [Cited: 1 18, 2009.]
http://www.netpecos.org/docs/mysql_postgres/x15.html#AEN17
42. Orallo, Enrique Hernández.Disca.upv.El lenguaje unificado UML. [Online] [Cited: 12 8, 2008.]
<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.

BIBLIOGRAFÍA

43. Osmosis Latina. Osmosis Latina. Guías de Software. Guía de UML: Diagramas Uso-Caso
<http://www.osmosislatina.com/lenguajes/uml/casos.htm>. [Online] [Cited: 2 20, 2009.]
44. Personales.unican. *Trabajando con Visual Paradigm para UML*. [Online] [Cited: 1 18, 2009.]
<http://personales.unican.es/ruizfr/is1/doc/lab/01/is1-p01-trans.pdf>.
45. PostgreSQL Global Development Group. Postgresql. Lista de Características de PostgreSQL 8.3.
[Online] [Cited: 1 21, 2009.] <http://www.postgresql.org/about/press/features83.html.es>.
46. **Puebla, Iván García**. Redksr. Java. *Eclipse: una herramienta profesional al alcance de todos*. [Online]
[Cited: 1 18, 2009.] <http://www.redksr.com/~talos/JAVA/JAVA3D/eclipse.pdf>.
47. Red Hat Middleware. Hibernate. Hibernate. [Online] [Cited: 12 4, 2008.] <http://hibernate.bluemars.net>.
48. SmallSquid. Aplicaciones Empresariales. Utilizando framework de programación en las empresas.
[Online] [Cited: 1 21, 2009.] <http://www.aplicacionesempresariales.com/utilizando-framework-de-programacion-en-las-empresas.html>.
49. Tvdi.UML.Node22.Diagramas UML [Online] [Cited: 3 26, 2009.]
<http://tvdi.det.uvigo.es/~avilas/UML/node22.html>
50. Vistamedica. Vista Medica [Online] [Cited: 12 3, 2008.] <http://www.vistamedica.com/>
51. Web Taller.com. Lenguajes. Java.Lecciones.Que es Java [Online] [Cited: 1 18, 2009.]
<http://www.webtaller.com/construccion/lenguajes/java/lecciones/que-es-java.php>
52. Willmanchamba. JBoss-Seam-Framework. [Online] [Cited: 12 4, 2008.]
<http://wilmanchamba.wordpress.com/2008/02/20/jboss-seam-framework/>.
53. Universidad de Murcia. Departamento de Informática y Sistemas. DIS Casos de Uso [Online] [Cited:
2 20, 2009.] <http://dis.um.es/~lopezquesada/documentos/casosuso.pdf>.

ANEXOS

Anexo 1 Pantalla que aparece antes que el usuario se autentique.



Figura 57 Prototipo de Interfaz de Usuario: ALAS-HIS

Anexo 2 Pantalla de Autenticación del Sistema de Gestión Hospitalaria



Figura 58 Prototipo de Interfaz de Usuario: Autenticación

Anexo 3 Pantalla que muestra los módulos del Sistema de Gestión Hospitalaria



Figura 59 Prototipo de Interfaz de Usuario: Módulos del HIS

Anexo 4 Pantalla que muestra el vínculo para acceder al Sistema de Teleconsulta



Figura 60 Prototipo de Interfaz de Usuario: Acceso a Sistema de Teleconsulta

Anexo 5 Pantalla principal del Sistema de Teleconsulta

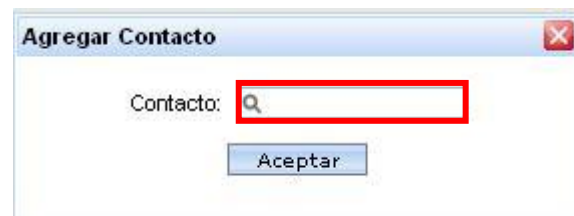


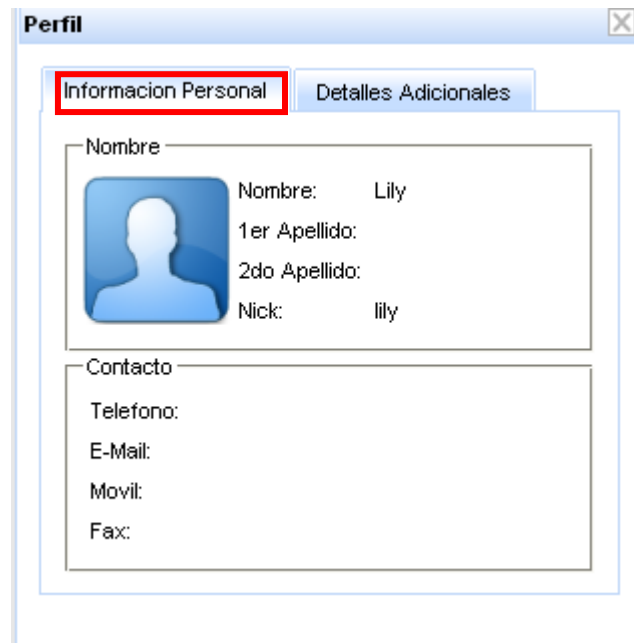
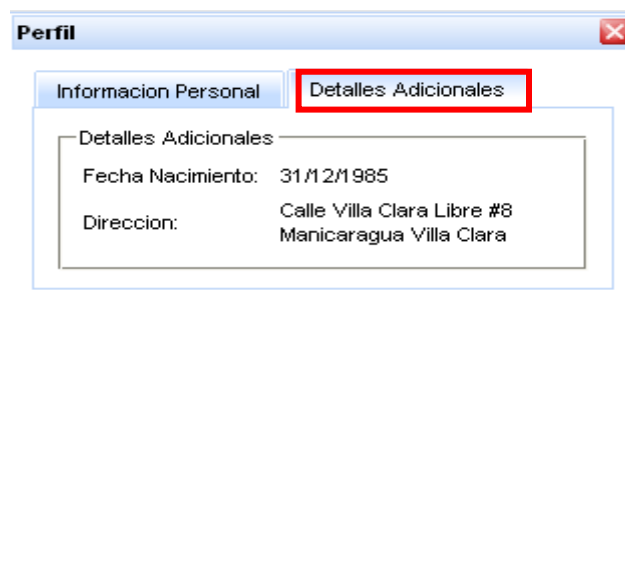
Figura 61 Prototipo de Interfaz de Usuario: Sistema de Teleconsulta


Anexo 6 Pantalla que muestra todas las acciones que hace el médico en relación con los contactos.



Figura 62 Funcionalidades del médico para accionar sobre un contacto

Anexo 7 Pantalla que muestra todas las acciones del médico**Figura 63 Funcionalidades del médico****Anexo 8 Pantalla que muestran la opción de agregar contacto.****Figura 64 Prototipo de Interfaz de Usuario: Agregar contacto**

Anexo 9 Pantalla que muestra la opción de ver datos adicionales del contacto.**Figura 65 Prototipo de Interfaz de Usuario: Ver perfil****Anexo 10 Pantalla que muestra la opción de ver datos adicionales del contacto.****Figura 66 Prototipo de Interfaz de Usuario: Ver perfil**

Anexo 11 Pantalla que muestra la opción de agregar grupo

Una ventana de diálogo con el título "Agregar Grupo" y un botón de cerrar en la esquina superior derecha. El campo de texto "Grupo:" contiene el texto "Click para Agregar Grupo", el cual está rodeado por un recuadro rojo. Debajo del campo de texto hay un botón "Aceptar".

Figura 67 Prototipo de Interfaz de Usuario: Agregar grupo**Anexo 12 Pantalla que muestra la opción de renombrar grupo**

Una ventana de diálogo con el título "Renombrar Grupo" y un botón de cerrar en la esquina superior derecha. El campo de texto "Renombrar Grupo:" contiene el texto "Compañero de Tesis", el cual está rodeado por un recuadro rojo. Debajo del campo de texto hay un botón "Aceptar".

Figura 68 Prototipo de Interfaz de Usuario: Renombrar grupo**Anexo 13 Pantalla que muestra la opción de modificar la información personal del médico**

Una ventana de diálogo con el título "Editar Perfil" y un botón de cerrar en la esquina superior derecha. Hay dos pestañas: "Información Personal" (seleccionada) y "Detalles Adicionales".

Información Personal:

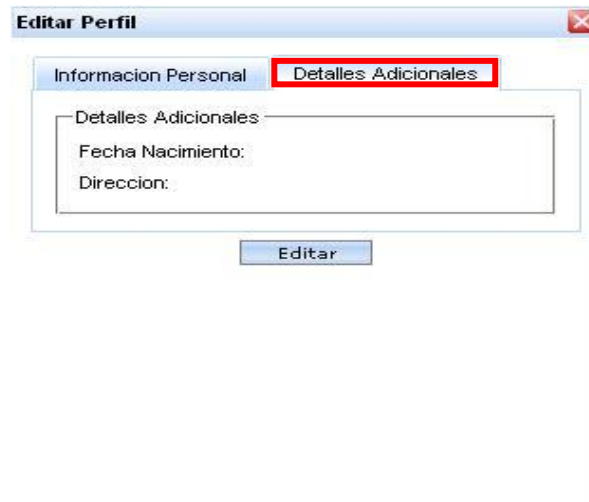
- Nombre: Leosdan
- 1er Apellido: Pozo
- 2do Apellido: Aguila
- Nick: leo

Contacto:

- Telefono: 042491856
- E-Mail: lpozo@estudiantes.uci.cu
- Movil: 042491856
- Fax: 042491856

Debajo de los campos de contacto hay un botón "Editar".

Figura 69 Prototipo de Interfaz de Usuario: Editar perfil

Anexo 14 Pantalla que muestra la opción de modificar otros datos del médico**Figura 70 Prototipo de Interfaz de Usuario: Editar perfil****Anexo 15 Pantalla que muestra la opción de cambiar el avatar del médico****Figura 71 Prototipo de Interfaz de Usuario: Cambiar avatar**

Anexo 16 Pantalla que muestra los mensajes entre médico-contacto

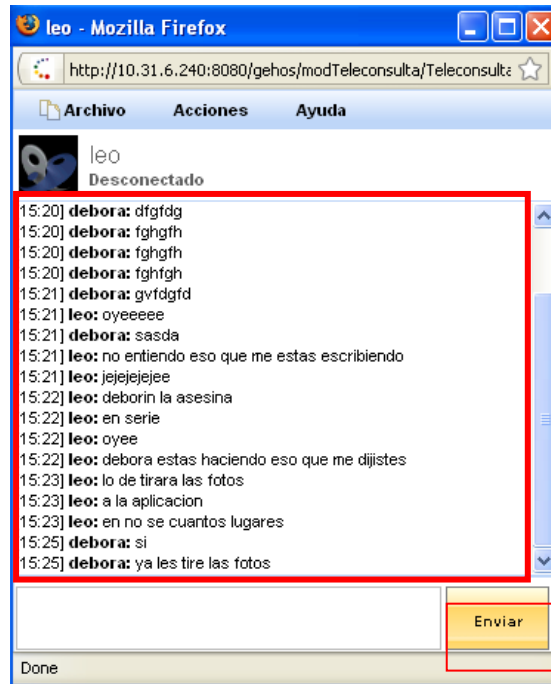


Figura 72 Prototipo de Interfaz de Usuario: Enviar mensaje

Anexo 17 Pantalla que muestra el historial de mensajes de la teleconsulta según el usuario.

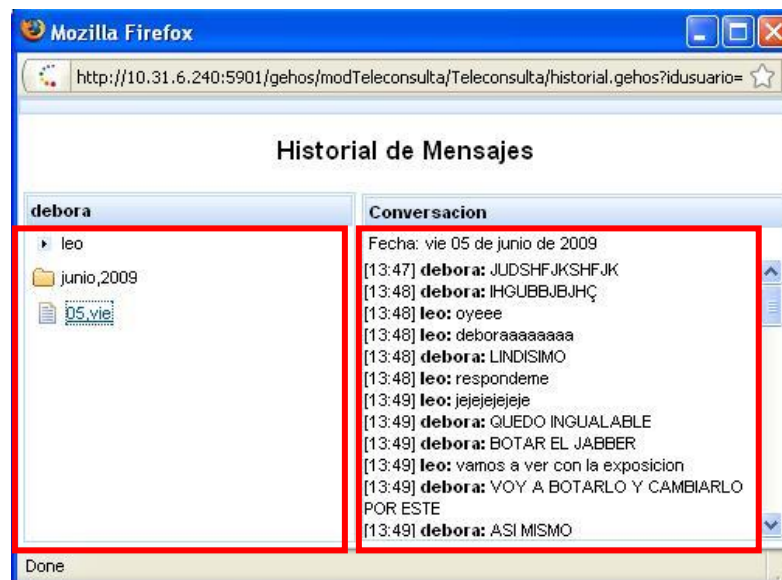


Figura 73 Prototipo de Interfaz de Usuario: Historial de mensaje

Anexo 18 Pantalla que muestra la pizarra compartida.

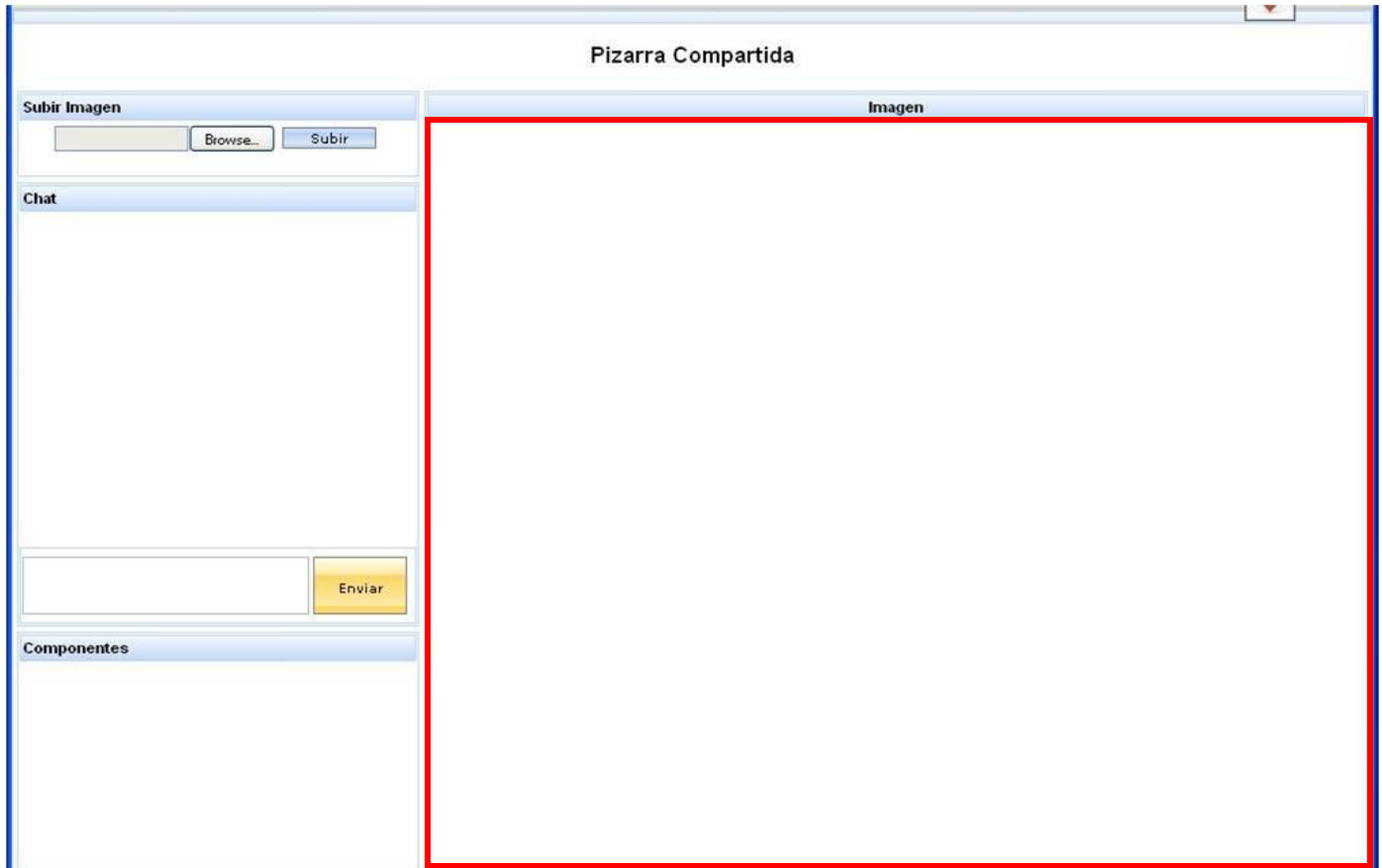


Figura 74 Prototipo de Interfaz de Usuario: Pizarra compartida

Anexo 15 Pantalla que muestra la imagen en la pizarra compartida y un chat.

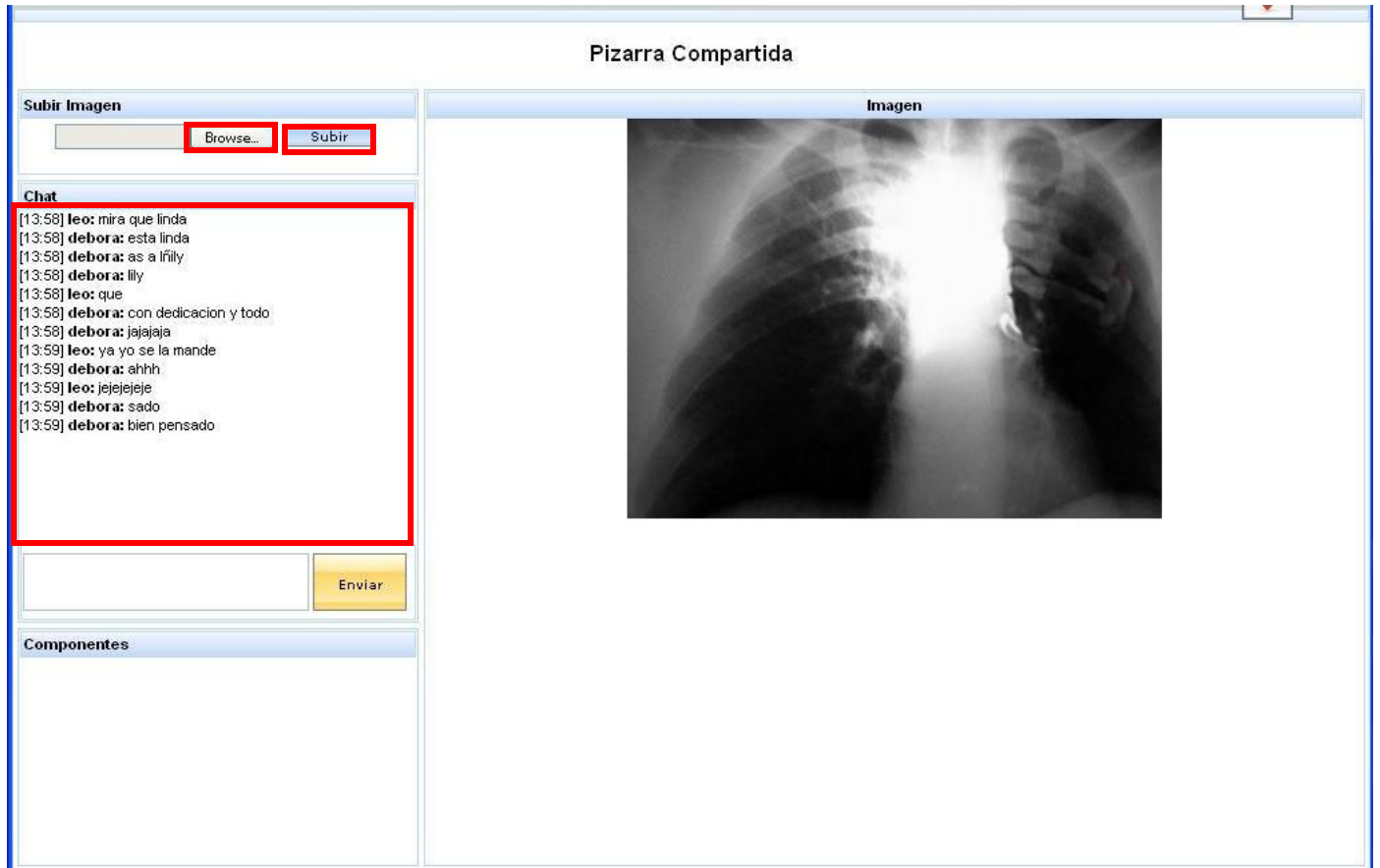


Figura 75 Prototipo de Interfaz de Usuario: Cargar imagen

GLOSARIO DE TERMINOS

Ajax (acrónimo para Asynchronous JavaScript + XML): Cargar una página, al ser necesario ejecutar una acción en el servidor. Esta se realiza a través de una petición asíncrona al mismo, buscando los datos que son usados para actualizar la página solo renderizandola, y luego muestra u oculta porciones de la misma.

API (Application Program Interfase, o interfaz de comunicación entre programas de aplicación): Representa un interfaz de comunicación entre componentes software.

Applets: Es un componente de una aplicación que se ejecuta en el contexto de otro programa.

Aplicación Web: Es una aplicación informática que los usuarios utilizan accediendo a un servidor Web a través de un navegador o browser. Estas son muy son populares debido a la habilidad para actualizar y mantener la información manipulada sin distribuir e instalar el software en miles de potenciales clientes.

Base de datos: Conjunto de datos organizados entre los cuales existe una correlación y que están almacenados con criterios independientes de los programas que los utilizan. La filosofía de las bases de datos es la de almacenar grandes cantidades de datos de una manera no redundante y que permita las posibles consultas de acuerdo a los derechos de acceso.

Booch (metodología de Booch): Es una técnica usada en ingeniería de software. Es un lenguaje de modelado de objetos y una metodología ampliamente usada en el diseño de software orientado a objetos. Fue desarrollada por Grady Booch mientras trabajaba para Rational Software.

Bytecode: Código binario intermedio más abstracto que el código máquina.

Caso de uso: Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores.

Diagrama: Presentación gráfica de un conjunto de elementos y sus relaciones.

Dominio: Área de conocimiento o actividad caracterizada por un conjunto de conceptos y terminología

GLOSARIO DE TÉRMINOS

comprendidos por los practicantes de ese dominio.

Estereotipos: Especificaciones del lenguaje.

Formulario: Parte de una página Web que el usuario completa y devuelve al servidor para su procesamiento.

Framework: Es la representación de una arquitectura de software en la que se modelan las relaciones generales, provee una estructura y una metodología de trabajo. Facilita el desarrollo de software, permite que tanto desarrolladores como diseñadores se enfoquen más en identificar requerimientos del software que con la programación de la aplicación. Utiliza la programación en tres capas Modelo-Vista-Controlador e intenta utilizar herramientas avanzadas.

Hardware: Conjunto de elementos materiales que componen un ordenador. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

IDE (Integrated Development Environment: Entorno de desarrollo integrado): Editor de código que sirve para depurar y facilitar las diferentes tareas necesarias en el desarrollo de cualquier tipo de aplicación.

INFOMED: Es la red de personas e instituciones que trabajan y colaboran para facilitar el acceso a la información y el conocimiento para mejorar la salud de los cubanos y de los pueblos del mundo, mediante el uso intensivo y creativo de las tecnologías de la información y la comunicación.

Infraestructura: Conceptos de bajo nivel.

Interfaz: Es la parte de un programa informático que permite a éste comunicarse con el usuario o con otras aplicaciones permitiendo el flujo de información.

JEE: Plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java.

JSF: Aplicaciones basadas en web.

GLOSARIO DE TÉRMINOS

Librerías: Constituyen el conjunto de procedimientos y funciones agrupados en un archivo con el fin de que sean utilizadas por otros programas.

Notación PascalCasing: Es como la notación húngara pero sin prefijos. En este caso, los identificadores y nombres de variables, métodos y funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula.

Notación CamellCasing: Es común en Java. Es parecido al Pascal-Casing con la excepción que la letra inicial del identificador no debe estar en mayúscula.

Motor de Persistencia: Los motores de persistencia tratan de solucionar el problema de la diferencia entre los dos modelos usados hoy en día para organizar y manipular datos: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional).

Mapeo de Objetos relacional: Técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

OMS: Es el organismo de la Organización de las Naciones Unidas (ONU) especializado en gestionar políticas de prevención, promoción e intervención en salud a nivel mundial. Organizada por iniciativa del Consejo Económico y Social de la ONU, se redactan los primeros estatutos de la OMS.

OOSE: Proceso de software.

OCL: Lenguaje de restricción.

OMT: Es una de las metodologías de análisis y diseño orientados a objetos, más maduros y eficientes que existen en la actualidad. La gran virtud que aporta esta metodología es su carácter de abierta (no propietaria), que le permite ser de dominio público y, en consecuencia, sobrevivir con enorme vitalidad. Esto facilita su evolución para acoplarse a todas las necesidades actuales y futuras de la ingeniería de software.

Restricciones: Comportamiento forzado de una clase o relación.

SIGEMEC: Sitio de Genética Médica.

GLOSARIO DE TÉRMINOS

Skinnability: Cambios de temas.

Sessions beans: Objeto que se instancia durante una sesión.

Sockets: Puntos o mecanismos de comunicación entre procesos que permiten que un proceso hable (emita o reciba información) con otro proceso incluso si estos procesos están en distintas máquinas.

Software: Es la parte lógica del ordenador, esto es, el conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Es el conjunto de instrucciones que permite la utilización del equipo.

XML: Lenguaje de etiquetado extensible.

XMI: Es un estándar para definir los nombres y propiedades de los ítems de datos que se intercambian entre el servicio solicitante y el servicio proveedor.