

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 7



Título: *Desarrollo del Módulo de Gestión de Reuniones del Sistema de Teleconsulta.*

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

Autores: *Hilda María Rodríguez Gómez
Jorge Moreira Rodríguez*

Tutores: *Ing. Débora González Tolmo
Ing. Miguel Ángel Fernández Marín*

*“Año 53 de la Revolución”
La Habana, junio 2011*

Datos de contacto

Ing. Débora González Tolmo.

Graduada de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2009. Instructora recién graduada en adiestramiento. Durante su trabajo como profesora ha impartido cursos pertenecientes a práctica profesional. Actualmente imparte cursos pertenecientes a práctica profesional.

En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Medicina (SEM) del Centro de Informática Médica (CESIM) y específicamente trabaja en el desarrollo del proyecto alas SIUM donde se desempeña como Analista Principal. Además se ha desempeñado como responsable de PP3-5 de este departamento desde el curso pasado.

Correo electrónico: dtolmo@uci.cu

Ing. Miguel Ángel Fernández Marín.

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas (UCI) en el 2008. Posee la categoría docente de Instructor. Durante su trabajo como profesor, ha impartido las asignaturas de Curso Introductorio de Matemática, Matemática Discreta I, Algebra Lineal. Actualmente imparte la asignatura de Álgebra Lineal.

En la vinculación con la producción pertenece al Departamento de Sistemas Especializados en Salud (SES), del Centro de Informática Médica (CESIM) y específicamente, trabaja en el desarrollo del proyecto alas CSI donde se desempeña como Desarrollador.

Correo electrónico: mafernandez@uci.cu

Dedicatoria

De Hilda:

A mi mami, porque sus brazos siempre se abren cuando necesito un abrazo, su corazón sabe comprender cuándo necesito una amiga. Sus ojos sensibles se endurecen cuando necesito una lección y por su fuerza y amor que me han guiado por la vida.

A mis viejitos, por ser mi fuente de inspiración.

A mis primos, a los que en todo momento he querido servirles de ejemplo.

A mi hermana y mi sobrinito, porque aunque la distancia nos separe siempre me han brindado su amor.

A mi segunda mamá, Loly, por siempre darme la fuerza y confianza que necesitaba en la travesía de estos 5 años.

A mi tío Héctor, por sembrar esa semilla de superación que todos necesitamos.

A mi familia, por ayudarme y apoyarme a llegar hasta aquí.

De Jorge:

Al Rey de reyes y Señor de señores, Jesús de Nazaret, por ser el camino, la verdad y la vida.

A mis nenes lindos: Rolando, Reinier, Cesar y Daniel Moreira, mis hermanitos.

A mis padres y al familión que tengo, porque son un gran regalo de DIOS.

A mis hermanos en la fe y a mis amigos, porque hacen más felices mis días en este peregrinaje.

Agradecimientos

De Hilda y Jorge:

A los profesores, por brindarnos su ayuda, conocimiento y apoyo.

A Iskael Díaz, por contestar a muchas dudas y cuestiones y por ofrecernos su apoyo en todo momento.

A los tutores, por guiar nuestras ideas, no solo en el desarrollo de esta tesis, sino también en nuestra formación como investigadores.

A todos aquellos que de una forma u otra han contribuido con nuestra formación como personas y como profesionales a lo largo de estos años.

A Cuba y a la Revolución por permitirnos ser parte de este sueño y por su aporte invaluable a nuestra educación, crecimiento y madurez a lo largo de nuestra vida.

De Hilda:

Agradecerle a Dios por haberme dado la sabiduría y la fortaleza para que fuera posible alcanzar este triunfo.

A mi mamá por ser lo más maravilloso que he conocido, porque siempre me ha puesto el ejemplo de fortaleza y sacrificio, de entrega y generosidad, de entusiasmo y trabajo, de fe y esperanza, de amor y de lucha, en fin no tengo palabras para agradecerle a Dios el que me haya permitido ser su hija.

A mis abuelitos porque siento un enorme agradecimiento por tenerlos en mi vida, por ser parte de la de ellos y por tener el privilegio de ser su nieta.

A Loly porque sus palabras han sido buenas consejeras para alejar las tormentas y para devolverme la calma, a mi tío Héctor por guiarme y a Ani, por ser una hermana para mí, por hacerme reír siempre y apoyarme incondicionalmente.

A Yisi, Dayi, Mary, Shirly, Jacque porque me siento feliz de tenerlas como amigas, lo cual es un privilegio para mí, por escucharme siempre y dejar huellas en mi corazón de todos los momentos que hemos pasado juntas, por eso nunca las olvidaré. A mis amigas Marianin, Ale, Jose, Denys, Maikelin por cuidar siempre de nosotras como hermanas, hacerme reír y compartir las largas noches de estudio.

A mi compañero de tesis, por su amistad, esfuerzo y dedicación en la realización de este trabajo.

A Yoel, por su apoyo, su compañía, sus consejos y por estar siempre con una palabra de aliento cuando la he necesitado. A mi familia que me ha ayudado a fortalecer mi carácter, me brindan una perspectiva de la vida mucho más amplia y me han enseñado a ser más cautelosa.

De Jorge:

Cierta vez escribí:

“Dios abrió el Mar Rojo para sacar a Israel de la esclavitud de Egipto: “como por tierra seca” y tras de sí cerró el camino de vuelta. Ahora yo soy uno más que ha cruzado ese gran mar, tras la promesa del desierto y el camino angosto...Canaán me espera...”

A eso se le añade hoy:

*Ha sido largo el viaje pero al fin llegué, más las promesas y el peregrinaje continúan. Canaán es toda una belleza y Dios me ha propuesto su conquista. Él me ha pedido: **“Repite siempre lo que dice el libro de la ley de Dios, y medita en él de día y de noche, para que hagas siempre lo que este ordena. Así todo lo que hagas te saldrá bien. Yo soy quien te manda que tengas valor y firmeza. No tengas miedo ni te desanimas porque yo, tu Señor y Dios, estaré contigo dondequiera que vayas.”**(Josué 1: 8 - 9)*

*Y esta ha sido mi respuesta: **“No importa a donde yo vaya siempre que sea contigo. Quiero que en mi viaje seas TU el camino, seas el destino y el paisaje alrededor.”** ¡Prepárate pues, Canaán, porque voy con mi Dios a tu conquista!*

Gracias a ti, Espíritu Santo, por ser mi buen pastor, por haberme rescatado y por hacer de mi lo que soy, porque fuera de ti nada es posible. Conoces mi corazón, y a la verdad que no hay palabras... pero gracias. Al que me escucha digo: debería conocerle, porque su Amor y su Hermosura no tienen fin y nuestra necesidad solo tiene verdadera saciedad en El.

Gracias a Libán, porque fue el Moisés que El Señor usó para sacarme de Egipto; por ser mi amigo. A todos mis hermanos porque cuando necesité el maná me ayudaron a encontrarlo, y aun cuando no lo hallé compartieron su porción. Gracias a cada uno por ser únicos y especiales, por brindarme su apoyo y amistad, por las aventuras que hemos vivido todos estos años. Obligado mencionar a: Odalis, Adriadna, Amarilis, Sirleys y Adolfo, la gente del núcleo, que se han ganado un lugar especial en mi corazón.

Gracias a mi papá y a mi mamá, y a toda mi familia, por aceptar el reto de Dios de darme vida, por protegerme y amarme, por su fe en mí.

Gracias a mis compañeros del aula y del apartamento con los cuales he convivido, porque de una forma u otra han sido instrumentos de Dios para bendecirme. Gracias a mi compañera de tesis, por su paciencia y dedicación, porque es triste soportar a Java y a Eclipse solo; gracias por su ayuda y Gloria a Dios porque nos graduamos juntos.

Resumen:

La práctica de la medicina con la aparición de enfermedades complejas o poco frecuentes, y de diagnóstico difícil, exige a los especialistas médicos la interacción entre ellos. Aunque quizás no estén en la misma área de salud, o se encuentren separados geográficamente por grandes distancias. Debido a esta situación, en la facultad 7 de la Universidad de las Ciencias Informáticas, se desarrolla el proyecto Teleconsulta, una aplicación web que precisa de un módulo de Gestión de Reuniones Virtuales.

Para el desarrollo de la aplicación se emplean herramientas como Gestor de Base de Datos PostgreSQL, el Lenguaje de Programación Java, la Metodología de Desarrollo de Software el Proceso Unificado de Desarrollo (RUP) para especificar, construir y documentar el sistema. Se hace uso del Lenguaje Unificado de Modelado (UML) y Visual Paradigm for UML 6.4 para la creación de los artefactos que se generan durante el ciclo de vida del software y el Framework JBoss Seam que utiliza el patrón Modelo-Vista-Controlador (MVC).

La puesta en marcha de esta aplicación permitirá una mayor eficiencia en la realización de un diagnóstico médico que necesite la opinión de otros especialistas de la salud, sin importar la distancia en que estos se encuentren.

Palabras Claves: Aplicación Web, Diagnóstico Médico, Framework, Reunión Virtual, Software, Teleconsulta.

Tabla de Contenidos:

INTRODUCCIÓN:	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA	5
1.1. Conceptos Asociados.	5
1.2. Antecedentes y Sistemas automatizados existentes.	6
1.2.1. Sistemas de Gestión de reuniones virtuales a nivel internacional.	7
1.2.2. Sistemas de Gestión de reuniones virtuales a nivel nacional.	11
1.3. Descripción de Tecnologías, Herramientas y Metodología a utilizar.	12
1.3.1. Metodología de desarrollo.	12
1.3.2. Tecnologías.	13
1.3.3. Lenguajes.	16
1.3.4. Servidor de aplicaciones.	17
1.3.5. Sistema Gestor de Base de Datos.	18
1.3.6. Herramientas.	19
CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA	23
2.1. Objeto de estudio.	23
2.2. Modelo de Dominio.	24
2.2.1. Conceptos Fundamentales del Dominio.	24
2.3. Especificación de los requisitos de software.	26
2.3.1. Requerimientos Funcionales.	27
2.3.2. Requerimientos No Funcionales.	27
2.4. Modelo de casos de uso del sistema.	32
2.4.1. Actores del Sistema.	32
2.4.2. Listado de casos de uso.	32
2.4.3. Diagrama de Casos de Uso del Sistema.	34
2.4.4. Casos de uso expandidos.	35

CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA.....	41
3.1. Arquitectura.	41
3.2. Estrategias de integración.	43
3.3. Modelo de Análisis.	43
3.4. Modelo de Diseño.	46
CAPÍTULO 4 IMPLEMENTACIÓN	54
4.1. Modelo de Datos.	54
4.2. Modelo de Implementación.	58
4.2.1. Diagrama de despliegue.	58
4.2.2. Diagrama de componentes.	59
4.3. Tratamiento de errores.	62
4.4. Seguridad.	62
4.5. Estrategias de codificación. Estándares y estilos a utilizar.	63
CONCLUSIONES:	69
RECOMENDACIONES:	70
REFERENCIAS BIBLIOGRÁFICAS:.....	71
BIBLIOGRAFÍAS:.....	74
ANEXOS:	77
GLOSARIO DE TÉRMINOS:	99

INTRODUCCIÓN

En la actualidad las computadoras se han convertido en una herramienta de suma importancia para el desarrollo de la ciencia y las nuevas tecnologías, debido a los crecientes avances que estas han alcanzado. La Informática es la ciencia del tratamiento automático de la información a través de un computador (1) y con el surgimiento de esta se ha convertido en necesidad social desarrollar métodos y medios eficaces para recopilar, conservar, buscar y divulgar la información. Mediante la utilización de hardware y software las tecnologías de la información y las comunicaciones (TIC), se encargan del estudio, desarrollo, implementación, almacenamiento y distribución de la información como medio de sistema informático. (2)

El proceso de informatización de la sociedad y la promoción y uso de las TIC en el país, abarca varios renglones. El sector de la salud, es uno de los de mayor importancia en la sociedad cubana, y la digitalización de los procesos de este se considera vital para preservar y agilizar el servicio médico. La utilización de las TIC en el sector de la salud en Cuba ha tenido un avance progresivo, ejemplo de ellos en: Red INFOMED, Desarrollo de la Telemedicina, Universidad Virtual y Biblioteca Virtual. (3)

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. (4)

El desarrollo constante de la medicina a causa del descubrimiento de nuevas enfermedades y tratamientos provoca que los especialistas médicos necesiten comunicarse entre ellos para la revisión de un examen, compartir opiniones o información de un determinado caso. Para ello, utilizan vías de intercambio como: teléfono, correo electrónico y persona-persona.

A través de la vía telefónica, 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, porque 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 intercambio persona-persona es muy efectivo, pues la comunicación es directa, pero se puede necesitar la presencia de otros

Introducción

expertos que podrían no estar en la misma área de salud o separados por grandes distancias para intercambiar opiniones.

Debido a esta situación se hace necesario un encuentro planificado entre los especialistas, con el correspondiente gasto de tiempo y dinero efectuados en el viaje de traslado para dicha reunión. Sin embargo, la tecnología puede sustituir muchas de las reuniones reales, ya que elimina las molestias y los gastos, convirtiéndolas en algo cómodo, rápido y factible, a través de las reuniones virtuales.

Al igual que se prepara una reunión “real”, la reunión “virtual” exige una planificación concreta, y hay que tener en cuenta una serie de códigos de conducta y comunicación propios de la reunión online. Para la realización de una reunión real, se necesita avisar a todos los participantes, acordar lugar, fecha y hora de la misma, los puntos a tener en cuenta en la reunión así como el orden del día, lo cual sería un gasto de tiempo al tratar de localizar a todos los participantes por diferentes vías como la telefónica y unir a todos en un mismo lugar obligándolos a trasladarse. Una de las principales ventajas de las reuniones virtuales es que el proceso de planificación y cambio se realiza mediante notificaciones y/o correos. También se suprime el traslado de un lugar a otro porque todo el intercambio se realiza mediante una estación de trabajo o computadora.

La Universidad de las Ciencias Informáticas (UCI) es una universidad productiva, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio – trabajo como modelo de formación. Desarrolla programas de informatización de la sociedad cubana a través de la relación con entidades nacionales, entre ellas el MINSAP. Los resultados alcanzados se extienden por todo el país, destacándose estos en las esferas de salud, educación, software libre, entre otras.

La Universidad cuenta con varios centros productivos, entre ellos el Centro de Informática Médica (CESIM), al cual pertenece el Departamento Sistemas Especializados de la Facultad 7, donde surge el proyecto Teleconsulta, para desarrollar una herramienta que permita el teletrabajo y que ha de incluir módulos de audioconferencia, videoconferencia y chat, con el objetivo de llevar a cabo reuniones virtuales, para discusión de casos médicos entre especialistas de la salud.

Introducción

A partir de esto fue identificado como **problema a resolver**: ¿Cómo viabilizar el proceso de gestión para la planificación de reuniones virtuales que permita concertar citas entre especialistas para la discusión de casos del Sistema Nacional de Salud?

Este problema se enmarca en el **objeto de estudio**: el proceso de gestión de reuniones virtuales del Sistema Nacional de Salud.

El objeto delimita el **campo de acción**: en la gestión de reuniones virtuales que permita concertar citas planificadas entre especialistas de requerida presencia para discusión de casos del Sistema Nacional de Salud de los diferentes hospitales de Cuba.

Para dar solución al problema fue trazado como **objetivo general**: desarrollar una aplicación para la gestión de reuniones virtuales entre el personal médico que permita concertar citas planificadas entre especialistas de requerida presencia para la discusión de casos del Sistema Nacional de Salud.

Para el cumplimiento del objetivo de este trabajo de diploma se ha propuesto desarrollar las siguientes **tareas de investigación**:

1. Realizar un análisis crítico y valorativo de los sistemas informáticos de gestión de reuniones virtuales existentes a nivel nacional e internacional.
2. Analizar las librerías, tecnologías, metodologías, plataforma, y herramientas propuestas por el proyecto para la solución del problema.
3. Definir los conceptos del dominio asociados a la planificación de reuniones virtuales.
4. Obtener los artefactos correspondientes a los Flujos de Trabajo: “Modelamiento del Negocio”, “Requerimientos”, “Análisis y Diseño” e “Implementación”.
5. Implementar el sistema informático aplicando las pautas de diseño y siguiendo las necesidades de funcionamiento establecidas en la Especificación de Requisitos de Software.

El contenido del presente trabajo de diploma está estructurado de la siguiente manera:

Capítulo 1. Fundamentación teórica: En este capítulo se expone el estado del arte del objeto de estudio de la presente investigación y se definen conceptos fundamentales para un mayor entendimiento

Introducción

del mismo. Se describen las metodologías, tecnologías, herramientas de software y el lenguaje de programación propuestos para dar solución a la problemática planteada y la justificación de su uso, con vistas al cumplimiento del objetivo trazado.

Capítulo 2. Características del sistema: En este capítulo se expone el objeto de estudio, los procesos del dominio. Se representa el modelo del dominio y se definen los requisitos funcionales y no funcionales, así como los actores y casos de uso del sistema.

Capítulo 3. Análisis y diseño del sistema: En este capítulo se describe todo lo referido al flujo de trabajo de análisis y diseño, se define la arquitectura y los patrones a utilizar para el diseño e implementación del sistema. También incluye sus principales artefactos como son los diagramas de clases del análisis y del diseño y su descripción. Además se explicarán las clases más significativas del diseño.

Capítulo 4: Implementación: En este capítulo se representará el modelo de datos, así como la explicación de sus tablas y sus atributos, obteniéndose el diagrama de clases persistentes para generar a partir del mismo la base de datos. Se expone lo relativo a la construcción de la solución propuesta. Se describen los principales artefactos generados en el Flujo de Trabajo de Implementación como son los diagramas de componentes y de despliegue.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

El presente capítulo expone el marco teórico y conceptual asociado a la problemática a resolver para lograr un mayor entendimiento de la misma. Se aborda una panorámica crítica y valorativa de los sistemas de gestión de reuniones virtuales, en el ámbito internacional y nacional. Se describen las metodologías, tecnologías, herramientas de software y el lenguaje de programación propuestos para dar solución a la problemática planteada y la justificación de su uso, con vistas al cumplimiento del objetivo trazado.

1.1. Conceptos Asociados.

El desarrollo de las tecnologías de la información (TIC) ha hecho posible que cualquier ciencia o disciplina se favorezca de ellas, con la práctica de estas en la medicina se ha llegado a nuevas metas para agilizar y fomentar el intercambio entre médicos para la discusión de casos. Es así como, las Telecomunicaciones han permitido que el personal médico pueda prestar servicios de salud a distancia, ya sea en tiempo real o diferido; actividades que dan origen a la **Telemedicina**.

La telemedicina puede ser tan simple como dos profesionales de la salud que discuten 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. Es 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.

También se utiliza para fomentar la educación médica, donde los alumnos de medicina y enfermería pueden aprender remotamente, apoyados por su profesor y con la presencia del paciente, gracias a la transmisión de datos médicos sobre redes adecuadas.

La Organización Mundial de Salud en 1998 define Telemedicina:

"Como la 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

Capítulo 1: Fundamentación Teórica

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. " (5)

El tipo de información generada dentro de la actividad médica hace que la Telemedicina involucre el procesamiento y la transmisión de datos, audio, imagen y video. Se enmarca dentro de un conjunto de aplicaciones como la Teleasistencia, la Teleconsulta, la Teleeducación y la Telesalud.

La **Teleconsulta** consiste en el intercambio de información especializada entre médicos y especialistas sobre opiniones o conocimientos de un determinado tema (6) y facilita así consejos de un experto remoto, pues permite una segunda opinión por otro especialista para confirmar el diagnóstico o para ayudar al médico local a llegar a un diagnóstico correcto.

En realidad la telemedicina y la teleconsulta incrementan su potencial cuando pueden realizarse de manera sencilla sin costosos equipamientos, que no se justifican salvo en las situaciones citadas. En la actualidad ha sido la revolucionaria expansión de Internet y no tanto los avances técnicos lo que ha permitido cambiar los planteamientos de teleconsulta a algo cotidiano.

Con el uso de la Telemedicina y dentro de ella las Teleconsultas se han implementado múltiples alternativas que se tienen hoy día para organizar una **Reunión Virtual**. Esta es la que permite la comunicación a través de Internet en tiempo real de los especialistas médicos, mediante las computadoras, sin importar la distancia en la que se encuentren estos, para intercambiar opiniones y/o conocimientos de un caso médico, pueden compartir imágenes, audio y video mediante la videoconferencia.

Las reuniones virtuales poco a poco se han convertido en una necesidad en el mercado mundial. El verdadero beneficio de las reuniones médicas en línea no se entiende hasta que todos los factores son considerados y la reunión virtual se experimenta.

1.2. Antecedentes y Sistemas automatizados existentes.

Con la evolución de la humanidad las personas han sentido la necesidad de intercambiar opiniones mediante diferentes vías, la común, ha sido el diálogo directo entre ellos. Cuando varias personas se

Capítulo 1: Fundamentación Teórica

agrupan en un momento y espacio dado, voluntaria o accidentalmente, se denomina reunión. La reunión de diferentes individuos en un lugar y momento específico puede llevarse a cabo de manera planificada, con un objetivo delimitado y con un tiempo de duración planeado, pero también puede darse de manera espontánea, por razones casuales y sin mayores propósitos.

Las herramientas de colaboración en línea se han convertido en una vía para organizar reuniones de trabajo, impartir cursos de formación, realizar presentaciones comerciales o simplemente por motivos de consulta. Una de las ventajas de Internet es la oportunidad de comunicarse en tiempo real, desde cualquier computadora conectada, en cualquier punto del mundo, a lo que se añade la ventaja de ahorro de costes y de energía. En el 2002 comienzan a aparecer sitios web promocionando las redes de círculos de amigos en línea (7), cuando el término se empleaba para describir las relaciones en las comunidades virtuales, y se hizo popular en el 2003 con la llegada de sitios tales como MySpace o Xing. Estos sitios permitían una imagen precisa del grupo de amigos con los cuales se deseaba intercambiar.

Con el surgimiento de estas tecnologías utilizadas para el intercambio en línea de varias personas, se comienzan a usar muchas de ellas en la medicina, para realizar consultas online, como medio de enseñanza a través de las videoconferencias y segundas opiniones de un determinado caso. Estas herramientas no cumplen con todos los objetivos y requisitos necesarios para el intercambio de información entre médicos, es por ello que se han desarrollado diferentes sistemas de reuniones virtuales que apoyan y fomentan una mejor comunicación en línea entre los especialistas médicos.

1.2.1. Sistemas de Gestión de reuniones virtuales a nivel internacional.

Existen a nivel internacional muchas tecnologías por los cuales se realizan reuniones virtuales, tales como: Adobe Connect, Skipe, Openmeetings y Webex Communications.

Adobe Connect: Es un sistema de comunicación web segura y flexible que permite a los profesionales ampliar y complementar la funcionalidad de Adobe Acrobat Connect Professional para proporcionar soluciones de comunicación web empresarial para formación, marketing, ventas, conferencias web y colaboración en línea, para reuniones y seminarios en tiempo real. (8)

Capítulo 1: Fundamentación Teórica

Al adquirir una solución del software se puede elegir la versión de servidor alojada, Adobe Connect Enterprise Hosted o la licencia para la versión del servidor de empresa, Adobe Connect Enterprise Server. Adobe Connect Events es un módulo del sistema que permite gestionar el registro de usuarios, la determinación de idoneidad, la notificación, los recordatorios automáticos por correo electrónico y el seguimiento para seminarios en línea a gran escala, además de presentaciones.

Los usuarios registrados, para ambas opciones, host y servidor empresarial, tienen un entorno web en el que pueden planificar sus reuniones, y definir los asistentes y sus roles, a los que pueden invitar por correo electrónico en donde ya está descrito el motivo de la reunión, la hora y el enlace para acceder a la sala virtual, o a través de la agenda del Microsoft Outlook.

Presenta como inconveniente que está enfocado al entorno empresarial y al uso entre usuarios comunes, por lo que no cumple con todos los requisitos necesarios para el intercambio de información entre médicos. Los complementos para MS Outlook de Microsoft Office ¹precisan de 64 MB de RAM por lo que requiere una infraestructura tecnológica que actualmente no se encuentra disponible para todas las provincias y municipios del país, lo cual también representa un gasto elevado de recursos, al no ser conveniente dada la situación actual de la economía.

También las versiones de Adobe Connect que brindan servicios de host, realizan el intercambio a través de servidores externos, y por los problemas de conexión a Internet que el país presenta por motivos del bloqueo económico y además de que habría que compartir la información de casos médicos fuera del país lo que constituye un problema ético y de seguridad, hace que la solución sea no factible.

Skipe: Es un software para realizar llamadas sobre Internet, utiliza el protocolo VoIP, creado en 2003. El código y protocolo permanecen cerrados y propietarios, pero los usuarios interesados pueden descargar gratuitamente la aplicación del sitio oficial. Es multiplataforma. (9)

Skype brinda además servicios de datos, fax, contestador automático, conferencia y videoconferencia, entre usuarios Skype, de forma gratuita; teléfonos celulares y teléfonos de red fija por cobro. También permite el desvío de llamados y mensajes de texto, lo cual significa que los usuarios del sistema con número telefónico Skype asignado, pueden recibir los llamados o mensajes directamente a su

¹ Es un programa de organización ofimática y cliente de correo electrónico de Microsoft, y forma parte de la suite Microsoft Office.

Capítulo 1: Fundamentación Teórica

computador o generar el desvío de estos a un teléfono celular o de red fija en el caso de llamados, por cobro.

Es una tecnología enfocada al entorno empresarial y al uso entre usuarios comunes, por lo que no cumple con todos los requisitos necesarios para el intercambio de información entre médicos. Para el uso de Skype se requiere conexión a Internet y registro de usuarios en el sitio oficial. La conexión ha de ser de banda ancha para que la comunicación sea fluida. Entre las deficiencias del software están que el sistema de administración de contactos y la integración con la web no son óptimos. Además no todos sus servicios son gratuitos, y dada la situación actual de la economía del país esto no es posible afrontarlo.

Openmeetings: Es un software libre que permite realizar conferencias a través de internet que utiliza la Licencia Pública Eclipse (EPL). Actualmente es utilizado para presentaciones, la formación en línea, conferencias web, pizarra de dibujo, colaboración y edición de documentos, intercambio de escritorio del usuario. (10) Para controlar la moderación y tener control sobre la sala que se utiliza es necesario solicitar al administrador del sistema dicho servicio, suministrando el nombre de la persona y la sala a la cual desea tener control como moderador. También permite programar con anticipación una reunión e informar a los asistentes mediante una notificación a través del correo electrónico enviando como parámetro la contraseña para acceder a esta y el tiempo de duración de la misma.

A pesar de que realiza la mayoría de las funcionalidades deseadas presenta como inconveniente que está enfocado al entorno empresarial y al uso entre usuarios comunes, por lo que no cumple con todos los requisitos necesarios para el intercambio de información entre médicos.

WebEx Communications: Brinda servicios de reuniones en línea, conferencias web, llamadas telefónicas en conferencia, reuniones de videoconferencia, seminarios interactivos, capacitaciones en línea y soporte técnico remoto. Solo requiere tener acceso a un navegador de Internet y un teléfono. Al momento de acceso a la reunión se descarga una aplicación ActiveX² o Java, que depende del sistema operativo. Se integra con Microsoft Outlook de forma rápida y sencilla, para planificar las reuniones e

² Es un framework para definir reutilizables componentes de software en un lenguaje de programación de manera independiente.

Capítulo 1: Fundamentación Teórica

invitar a los asistentes. WebEx sólo está disponible como proveedor de servicios de aplicaciones (ASP) proporcionado por los servidores habilitados de la compañía. (11)

Está enfocado al entorno empresarial y al uso entre usuarios comunes, por lo que no cumple con todos los requisitos necesarios para el intercambio de información entre médicos. Realiza el intercambio a través de servidores externos, y por los problemas de conexión a Internet que el país presenta por motivos del bloqueo económico, lo hace una solución no factible, además de que habría que compartir la información de casos médicos fuera del país lo que podría convertirse en problemas éticos y de seguridad que implican. También es un software propietario y hay que pagar la licencia de uso.

Enfocados a la medicina existen diferentes software que permiten la gestión de intercambio entre los especialista médicos de forma online, estos son: Sistema de Teleconsulta Médica (STM-20), Asociación Santiagueña de Ayuda Integral al discapacitado mental (Asaim) y Chat de SEMES Vigo.

STM-20: Es un proyecto para potenciar la telemedicina en los hospitales de la comunidad gallega. Este proyecto co-financiado con Telefónica, compañía telefónica de España, tiene como objetivo atender la radiología de los centros de salud de la zona sanitaria que compete a este Hospital. De momento está implementado como sistema de comunicaciones que conecta el centro de Atención Primaria de Baltar con el Hospital Provincial de Pontevedra.

El STM-20 permite el diálogo, almacenamiento y transferencia de documentos, tanto texto como imágenes, generados en los distintos centros de salud. A través de una Red Digital de Servicios Integrados (RDSI), el especialista entra en contacto directo con el médico de medicina general que comparte la misma información y visualiza los mismos documentos.

El proyecto cuenta con la experiencia de Telefónica en el desarrollo de este tipo de sistemas y la tecnología de Sun Microsystems. El STM-20 está formado por la unión de dos redes locales de un solo nodo y utiliza el protocolo TCP/IP.

Este sistema fue creado y se utiliza para el trabajo a nivel nacional del país de origen, España, para la comunicación de dos hospitales, por lo que no cumple con los requisitos para el trabajo en la red médica de Cuba, pues se requiere una aplicación para acceder a ella desde cualquier hospital del país. No se

Capítulo 1: Fundamentación Teórica

tiene acceso para instalarlo como software, o para acceder libremente al sitio web. Las características e informaciones presentes en este no se corresponden con el sistema a desarrollar.

Asaim: Es un proyecto que facilita permanentemente la Actualización y el Control del Riesgo Profesional de Médicos, promueve la difusión del conocimiento científico en todas las áreas relacionadas con la Internet en Medicina y la Bioinformática. Realiza desarrollos en la educación médica a distancia, reuniones científicas virtuales, software biomédico y en otras áreas de convergencia entre las ciencias de la informática y de la salud. Utiliza para la comunicación de sus usuarios las charlas en línea (chat), mediante la metodología NetMeeting.

Las características e informaciones presentes en este software no se corresponden con el sistema a desarrollar, pues fue creado y se utiliza para el trabajo a nivel nacional del país de origen, Argentina, no se tiene acceso al mismo para instalarlo como software, o para acceder libremente al sitio web. Además se requiere de pagos mensuales para su uso, por lo que no es una opción factible dada la situación económica que atraviesa el país.

Chat de SEMES Vigo: Se encuentra presente en el portal de medicina de Emergencia de España, se activa todos los domingos, a partir de las 22.30 horas, para charlar sobre los temas de actualidad de la semana y los casos más importantes atendidos en esa semana.

Este sistema fue creado y se utiliza para el trabajo a nivel nacional del país de origen, España, para todos los hospitales, pero no se tiene acceso al mismo para instalarlo como software, o para acceder libremente al sitio web. Las características e informaciones presentes en este no se corresponden con el sistema a desarrollar, nada más se puede tener acceso a las reuniones una vez a la semana y esto no es lo requerido para el teletrabajo en la medicina en Cuba.

1.2.2. Sistemas de Gestión de reuniones virtuales a nivel nacional.

En la UCI, la facultad 6 desarrolló un sistema denominado: **alMEDIGEN**. Este es un sistema de Teleconsulta genética, se desarrolló con herramientas y tecnologías libres, entre las que se encuentran PHP como lenguaje de programación, Symfony como framework y el entorno de desarrollo Eclipse, Apache como el Servidor Web que soportará esta aplicación y el gestor de Base de Datos MySQL.

Capítulo 1: *Fundamentación Teórica*

Todas ellas compatibles con las políticas de desarrollo establecidas por Infomed, para permitir su fácil integración a esta red y así dar la posibilidad de que este sistema pueda ser accedido por cada uno de los Centros Genéticos conectados a Infomed. El sistema hace uso de la plataforma LAMP³, tiene a Linux como Sistema Operativo.

AlasMEDIGEN es utilizado para la discusión de casos genéticos mediante un chat, y el alcance del producto Teleconsulta, va dirigido a cualquier especialista médico del país, para posibilitar el intercambio entre ellos sin importar la rama de la salud a la que este pertenece; además las reuniones virtuales se planifican con la posibilidad de intercambiar información mediante audio y video.

Todos los sistemas anteriormente mencionados, se le suma el inconveniente que por sus características no pueden ser integrados al Sistema de Información Hospitalaria (HIS), ya que no cumplen con las pautas del diseño, los estándares de codificación del mismo y la arquitectura de estos no son compatibles.

1.3. Descripción de Tecnologías, Herramientas y Metodología a utilizar.

1.3.1. Metodología de desarrollo.

Una metodología de desarrollo es el conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software. (12) La metodología indica cómo hay que obtener los distintos productos parciales y finales en el desarrollo del software que sigue uno o varios modelos del ciclo de vida. No existe una metodología de software universal, las características de cada proyecto y equipo de desarrollo exigen que el proceso sea configurable.

Proceso Unificado de Desarrollo (RUP): Es una de las metodologías tradicionales o no ágiles. La misma utiliza el Lenguaje Unificado de Modelado (UML) como lenguaje de representación visual. Este proceso unificado define cuatro elementos: los roles, que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los artefactos, que responden a la pregunta ¿Qué? y los flujos de trabajo que responden a la pregunta ¿Cuándo?

³Entorno de programación basado en la unión de las herramientas Linux, Apache, MySQL y de un lenguaje de programación del tipo Perl, PHP o Python.

Capítulo 1: Fundamentación Teórica

Tiene tres características fundamentales: es iterativo e incremental, centrado en la arquitectura y dirigido por casos de usos y consta de cuatro fases: Inicio, elaboración, construcción y transición. Cuenta con nueve flujos de trabajo, de ellos seis de ingeniería (Ingeniería de Negocios, Requerimientos, Análisis y Diseño, Implementación, Pruebas e Instalación) y tres de apoyo (Configuración y administración del cambio, Administración de proyecto y Ambiente), distribuidos en las cuatro fases de desarrollo.

Esta metodología indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto, aunque no cómo hacerlo, además de que ofrece documentación formal de todo el ciclo de vida del desarrollo de software. Como metodología no ágil, es adaptable al contexto y las necesidades de cada organización, asegura la producción de software de calidad, dentro de plazos y presupuestos predecibles. Dadas sus características 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, dando una mejor visión del sistema a desarrollar, pues permite dividir el trabajo en partes más pequeñas.

1.3.2. Tecnologías.

En este epígrafe se tratarán una serie de conceptos relacionados con las tecnologías a utilizar en el proceso para la gestión de reuniones virtuales.

Java Server Faces (JSF): Es un framework basado en el patrón Modelo Vista Controlador (MVC) para aplicaciones Java basadas en web, que facilita y agiliza el diseño de interfaces de usuario en aplicaciones Java Enterprise Edition (Java EE). Tiene el objetivo de normalizar y estandarizar el desarrollo de aplicaciones web.

Además proporciona un conjunto de componentes para la interfaz de usuario, incluyendo los elementos estándares de Lenguaje de Marcado de Hipertexto (HTML) para representar un formulario y la posibilidad de definir nuevos componentes, también define un conjunto simple de clases Java para componentes de la interfaz de usuario, eventos y estado de los mismos.

RichFaces 3.2.0: Es una librería de componentes visuales para JSF y un avanzado framework para la integración de funcionalidades Ajax en dichos componentes visuales, mediante el soporte de la librería Ajax4JSF. Soporta Facelets y temas CSS o skins. Aprovecha al máximo los beneficios del framework JSF incluyendo, la validación y conversión de instalaciones, junto con la gestión estática y dinámica de los recursos. Es un proyecto open source, activo y con una comunidad también activa.

Capítulo 1: Fundamentación Teórica

Facelets: Es un framework de presentación para la construcción de plantillas, centrado en la tecnología JSF. Su uso facilita la creación de plantillas para los componentes y páginas, ayuda a separar los componentes de interfaz de usuario en diferentes archivos, mejora los informes de errores que tiene JSF, soporta completamente el **Expression Language** (EL), valida **EL** en tiempo de construcción y por tanto no es necesaria la configuración Lenguaje de Marcado Extensible (XML). Permite fácilmente la creación de funciones y librerías de componentes.

Ajax4JSF: Es una librería de código abierto que se integra totalmente en la arquitectura de JSF. Se extiende la funcionalidad de las etiquetas de JSF dotándolas con tecnología Ajax, de forma limpia y sin añadir código JavaScript. Además escribe componentes propios con soporte 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. Presenta una arquitectura abierta y soporte para los estándares en la industria para mezclar componentes que son de distintas librerías y trabaja del lado del servidor.

JBossSeam 2.1.1: Es un framework de código libre desarrollado por JBoss. Combina a los frameworks Enterprise JavaBeans (EJB3) y JSF. Se puede acceder a cualquier componente EJB desde la capa de presentación, refiriéndose a él mediante su nombre de componente Seam, ya que integra la capa de presentación (JSF) con la capa de negocios y persistencia (EJB).

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.

Hibernate: Es una herramienta de mapeo objeto-relacional (ORM)⁴ para la plataforma Java que 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 un software libre y permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen.

⁴Más conocido por su nombre en inglés, Object-Relationship mapping, o sus siglas O/RM, ORM, y O/R mapping) es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje POO y el utilizado en una BDR. En la práctica esto crea una base de datos orientada a objetos virtual, sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (básicamente herencia y polimorfismo).

Capítulo 1: Fundamentación Teórica

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible. Es poderoso, tiene alto rendimiento y permite desarrollar clases persistentes en el lenguaje orientado a objetos. También genera las sentencias del Lenguaje de Consulta Estructurado (SQL), que mantienen la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución. Ofrece también un lenguaje de consulta de datos llamado Hibernate Query Language (HQL).

Enterprise Java Beans (EJB3): Es una Interfaz de Programación de Aplicación (API) que forma parte del estándar de construcción de aplicaciones empresariales Java Enterprise Edition (J2EE), detalla cómo los servidores de aplicaciones proveen objetos desde el lado del servidor. También proporciona 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 estos sean flexibles y sobre todo reutilizables. (13)

Java Persistence API (JPA): Es una especificación de Sun Microsystems, desarrollada para la plataforma Java EE e incluida en el estándar EJB3, se basa en las ideas de los principales frameworks de persistencia y APIs como Hibernate. Esta API busca unificar la manera en que funcionan las utilidades que proveen un mapeo objeto-relacional. El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, y usar objetos regulares, lo que permite la persistencia de objetos Java a cualquier base de datos relacional.

Tecnologías horizontales:

Existen un conjunto de tecnologías que horizontalmente trabajan en conjunto con todas las tecnologías antes mencionadas y sirven de soporte a las mismas. Estas se describen a continuación:

Java Platform Enterprise Edition (JavaEE5): Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java, preferiblemente del lado del servidor. Se caracteriza por tener una arquitectura de N niveles distribuida, basándose ampliamente en

Capítulo 1: Fundamentación Teórica

componentes de software modulares, ejecutándose sobre un servidor de aplicaciones. Esta plataforma proporciona a los desarrolladores un potente conjunto de API que reducen el tiempo de desarrollo y la complejidad, al tiempo que mejoran el rendimiento de las aplicaciones. La implementación Java EE puede ser descargada desde la web gratuitamente, y existen disímiles herramientas de código abierto, disponibles para extender la plataforma o para simplificar el desarrollo.

Java Runtime Environment (JRE6): Se corresponde con un conjunto de utilidades que permite la ejecución de programas Java sobre todas las plataformas soportadas. El entorno en tiempo de ejecución Java está constituido por una máquina virtual Java y además por las librerías de clases estándar que implementan el API de Java. Ofrece la funcionalidad 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.

1.3.3. Lenguajes.

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. Representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona, son independientes de las computadoras a utilizar. (14)

Java: Es un lenguaje de objetos desarrollado por un grupo de ingenieros de Sun Microsystems en 1991, utilizado por Netscape posteriormente como base para JavaScript. A diferencia de los lenguajes convencionales, que generalmente están diseñados para ser compilados a código nativo, este es compilado a un código intermedio o bytecode, el cual es interpretado por una máquina virtual de Java. (15)

Es un lenguaje orientado a objetos, lo que permite agrupar en estructuras encapsuladas tanto sus datos como los métodos o funciones que manipulan estos. Fue diseñado para crear software altamente fiable, lo que proporciona a los programadores la disminución de errores, 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. También proporciona una colección de clases para su uso en aplicaciones de red, que permiten

Capítulo 1: Fundamentación Teórica

abrir sockets, establecer y aceptar conexiones con servidores o clientes remotos, para así facilitar la creación de aplicaciones distribuidas.

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 dos últimas características se conocen como la Máquina Virtual Java (JVM). Además 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.

Lenguaje Unificado de Modelado 2.0 (UML): Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. (16) 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.

Entre sus principales características se encuentran las de constituir un lenguaje unificado para la modelación de sistemas, contiene corrección de errores viables en todas las etapas. Es aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente-servidor y los modelos permiten la comunicación con el cliente en todas las etapas.

Es ideal para el modelado de sistemas orientados a objetos ya que incluye la representación de la abstracción, herencia, polimorfismo, encapsulamiento o encapsulación, envío de mensajes, asociaciones y agregación. Permite además detectar con mayor facilidad las dependencias y dificultades implícitas del sistema.

1.3.4. Servidor de aplicaciones.

Un servidor de aplicaciones es un software instalado en un computador para facilitar la ejecución de otras aplicaciones, que generalmente gestiona la mayor parte (o la totalidad) de las funciones de lógica de negocio y de acceso a los datos de la aplicación. Los principales beneficios de la aplicación de la tecnología de servidores de aplicación son la centralización y la disminución de la complejidad en el desarrollo de aplicaciones. Estos típicamente incluyen también middleware (o software de conectividad)

Capítulo 1: *Fundamentación Teórica*

que les permite intercomunicarse con variados servicios, para efectos de confiabilidad, seguridad y no repudiación. Los servidores de aplicación también brindan a los desarrolladores una Interfaz para Programación de Aplicaciones (API), de tal manera que no tengan que preocuparse por el sistema operativo o por la gran cantidad de interfaces requeridas en una aplicación web moderna.

JBoss Server o JBoss AS 4.2: Es un Servidor de Aplicaciones Java EE de Software Libre implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo que lo soporte. Entre sus características más relevantes se encuentra que implementa todo el paquete de servicios de J2EE, es confiable a nivel de empresa, orientado a arquitectura de servicios y presenta flexibilidad consistente. También proporciona servicios que necesitan la mayoría de las aplicaciones empresariales, tales como la seguridad, transaccionalidad, persistencia, vigilancia, gestión de recursos y de acceso remoto.

JBoss AS es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofrece una plataforma de alto rendimiento para aplicaciones de e-business. Combina una arquitectura orientada a servicios con una licencia de código abierto, JBoss AS puede ser descargado, utilizado, incrustado, y distribuido sin restricciones por la licencia. Por este motivo es la plataforma más popular de middleware⁵ para desarrolladores, vendedores independientes de software y, también, para grandes empresas. (17)

1.3.5. Sistema Gestor de Base de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. (18)El propósito general de los sistemas de gestión de bases de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización.

⁵ Software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos.

Capítulo 1: Fundamentación Teórica

PostgreSQL 8.3: Es una poderosa herramienta de código abierto para el manejo de bases de datos relacionales, publicada bajo licencia BSD⁶. (19) Puede ser utilizado en diversos sistemas operativos, incluyendo GNU/Linux, UNIX y Windows. Soporta el almacenamiento de números binarios de gran tamaño, incluyendo imágenes, sonido, y video.

Características de PostgreSQL:

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits. También permite la creación de tipos propios como la de los disparadores e incorpora funciones de diversa índole.
- Incorpora una estructura de datos de arreglos.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto- relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- Ejecuta procedimientos almacenados en varios lenguajes de programación como Java, Perl, Python, Ruby, C, C + +, y PL / pgSQL y soporta casi toda la sintaxis SQL pues tiene soporte total para llaves extranjeras, joins, vistas.
- Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

1.3.6. Herramientas.

Una herramienta es cualquier dispositivo que, cuando se emplea de forma apropiada, mejora el desempeño de una tarea. Las herramientas en el desarrollo de sistemas mejora la productividad del

⁶Es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Capítulo 1: *Fundamentación Teórica*

analista, la eficiencia y la calidad del sistema de información; ya que convierte la arquitectura en soluciones sólidas.

Herramienta CASE:

Las herramientas Ingeniería de Software Asistida por Computadora (CASE) son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases. (20) Entre sus objetivos se encuentran mejorar la productividad en el desarrollo y mantenimiento del software, aumentar la calidad del software, reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos y mejorar la planificación de un proyecto.

Visual Paradigm 6.4: Es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. Ayuda a una más rápida construcción de aplicaciones de calidad y a menor costo. Permite construir todo tipo de diagramas de clases, generar código desde diagramas y generar documentación. Apoya los estándares más altos de las notaciones de Java y de UML. Soporta aplicaciones web y es fácil de instalar y actualizar. Está diseñado para distintos usuarios entre los que se incluyen ingenieros de software, analistas de sistemas, analistas de negocios, arquitectos y desarrolladores. Está orientado a la creación de diseños y se usa el paradigma de programación orientada a objetos. (21)

Características de Visual Paradigm:

- 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 tanto la ingeniería directa como inversa, soportada en varios lenguajes de programación.
- Permite la disponibilidad de múltiples versiones para cada necesidad, capaz de integrarse con los principales IDEs.
- Disponibilidad en múltiples plataformas y soporta el análisis y diseño orientado a objetos.
- Generación de Bases de Datos basado en diagramas.

Capítulo 1: Fundamentación Teórica

Herramienta IDE:

Una herramienta de Ambiente de Desarrollo Integrado (IDE) es un programa compuesto por un conjunto de herramientas para un programador, que se ejecuta a partir de una única interfaz de usuario. (22) Los IDEs pueden ser aplicaciones por si solas o pueden ser parte de aplicaciones existentes.

Eclipse GANYMEDE: Es una nueva versión del conjunto de herramientas de desarrollo de software del proyecto Eclipse (iniciado por IBM), que es competencia directa con otras herramientas como Netbeans (de Sun Microsystems). (23) En la actualidad, funciona bien como un IDE Java, e incluye herramientas de desarrollo Java. Se requiere que tenga el entorno de Sun Java Runtime (JRE) instalada. (24) Es multiplataforma y se utiliza para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Presenta una arquitectura abierta y basada en plugins, lo que permite expandir las capacidades de la plataforma base automáticamente.

Eclipse presenta control del editor de código, de compilador y de un potente depurador, lo que permite establecer puntos de interrupción, modificar e inspeccionar valores de variables e incluso depurar código que resida en una máquina remota, desde una única interfaz de usuario. Su misión consiste en evitar tareas repetitivas, facilitar la escritura de código correcto, disminuir el tiempo de depuración e incrementar la productividad del desarrollador.

Herramienta de Modelado de Base de Datos:

Las herramientas de modelado de base de datos permiten gestionar y documentar los cambios realizados sobre los modelos de datos a lo largo del tiempo. Suelen ser usadas como una forma de sincronización entre el modelo y la base de datos, además de la capacidad de migrar las estructuras de datos de una versión o plataforma de base de datos a otra diferente.

Case Studio v2.22: 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.

Capítulo 1: Fundamentación Teórica

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. Tiene soporte para trabajar con una amplia variedad de formatos de base de datos (Oracle, SQL, MySQL, PostgreSQL, Access) y permite además generar scripts SQL, usar plantillas de diseño personalizables y crear detallados informes en HTML y Formato de Texto Enriquecido (RTF).

Herramienta para el desarrollo de Java.

Java Development Kit (JDK) 1.6: Provisto gratuitamente por Sun Microsystems, compañía creadora de Java. Se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en Java. Incorpora además la posibilidad de ejecutar parcialmente el programa, ya que detiene la ejecución en el punto deseado y estudia en cada momento el valor de cada una de las variables (Debugger). Permite la detección y corrección de errores del lenguaje. Existe también una versión reducida denominada Java Runtime Environment (JRE) destinada únicamente a ejecutar código Java (no permite compilar). (25)

En este capítulo se corroboró con el análisis los sistemas de gestión de reuniones virtuales que no existe un sistema para todas las especialidades médicas que cumpla con las características del Sistema Nacional de Salud y del sistema a desarrollar. También se efectuó un profundo análisis de las metodologías, herramientas y tecnologías a utilizar propuestas por la dirección del proyecto para darle solución al objetivo planteado, llegando a la conclusión que son las adecuadas pues son tecnologías libres y de fácil uso.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

Este capítulo tiene como objetivo describir 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. También en este capítulo se realiza la descripción de los procesos del negocio, se definen actores, trabajadores, los requisitos funcionales y no funcionales, mediante el diagrama de casos de uso del sistema, las relaciones entre los actores y casos de uso del sistema, así como las descripciones textuales de cada uno de ellos.

2.1. Objeto de estudio.

El objeto de estudio de esta investigación es la gestión de la planificación de reuniones virtuales para el Sistema Nacional de Salud, para fomentar la comunicación de manera directa y en tiempo real entre especialistas para la discusión de casos médicos, permitiendo el intercambio entre varios hospitales.

Para que se efectúe satisfactoriamente esta comunicación, 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, brindándose un mejor diagnóstico al paciente, una gran ayuda al profesional de la salud en su formación profesional e información necesaria para futuros diagnósticos.

Para dar cumplimiento al objeto de estudio se propone realizar una aplicación que permita organizar una reunión virtual, proponiendo el tema de la misma, las características a tener y el horario, accederán médicos invitados de todas las especialidades para intercambiar opiniones y determinar un diagnóstico en conjunto con otros profesionales de la salud. Ofrecerá opciones como: planificar, modificar, y suspender una reunión.

Capítulo 2: Características del Sistema

2.2. Modelo de Dominio.

Un Modelo de Dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea de construcción del modelo de dominio, presentado como uno o más diagramas de clases y contiene, no conceptos propios de un sistema de software sino de la propia realidad física. (26)

El Modelo de Dominio tiene como objetivo ayudar a comprender los conceptos que utilizan los usuarios, los conceptos con los que trabajan y con los que deberá trabajar la aplicación, identifica los atributos y las asociaciones, y es la herramienta más importante del análisis orientado a objetos.

El proceso para su elaboración tiene tres pasos:

1. Identificar las clases conceptuales.
2. Dibujarlas en un diagrama de clases.
3. Añadir relaciones y atributos.

Para el desarrollo del Módulo de Gestión de Reuniones de Teleconsulta se define un Modelo Conceptual al no existir 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.

2.2.1. Conceptos Fundamentales del Dominio.

Teleconsulta: Es la prestación de servicios de medicina a distancia. Para su implementación se emplean usualmente tecnologías de la información y las comunicaciones, ofrece 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.

Reunión Virtual: Es la que permite la comunicación a través de Internet en tiempo real de los especialistas médicos, mediante las computadoras, sin importar la distancia en la que se encuentren estos, para intercambiar opiniones y/o conocimientos de un caso médico, pueden compartir imágenes, audio y video a través del audio y videoconferencia.

Capítulo 2: Características del Sistema

Notificación: Es una de las partes fundamentales en el proceso de las reuniones virtuales porque a través de esta se le avisa al médico todo lo relacionado con la reunión, ya sea fecha, hora, tema, título, entre otras características de la reunión.

Usuario: Representa al médico, con el cual se establece la comunicación y los debates.

Moderador: Es un usuario del sistema que va a dirigir el comportamiento de la reunión ya que tiene permisos para planificar, iniciar, modificar y suspender esta, además de que es el que invita y manda las notificaciones a los demás usuarios con la información de la reunión.

Médico: Es un usuario del sistema con una especialidad determinada que participa en la reunión, solicitando información para llegar a un diagnóstico en conjunto con los demás usuarios (especialistas) invitados a esta. Establece comunicación con otro especialista mediante un chat o videoconferencia.

Especialista Externo: Es un usuario con una especialidad determinada, que no se encuentra en el sistema y se le manda un correo, como notificación para que se presente en la discusión de un caso, en esta se le envía un usuario y contraseña temporal del sistema, para que pueda acceder.

Chat: Es la funcionalidad que permite la comunicación e intercambio de información a través de mensajes entre dos usuarios.

Mensaje: Es el proceso de intercambio de información médica que se envía entre médicos a través del chat, el cual constituye un medio de comunicación entre ellos.

Videoconferencia: Es la comunicación simultánea bidireccional de audio y video, permite mantener reuniones con grupos de personas situadas en lugares alejados entre sí.

Capítulo 2: Características del Sistema

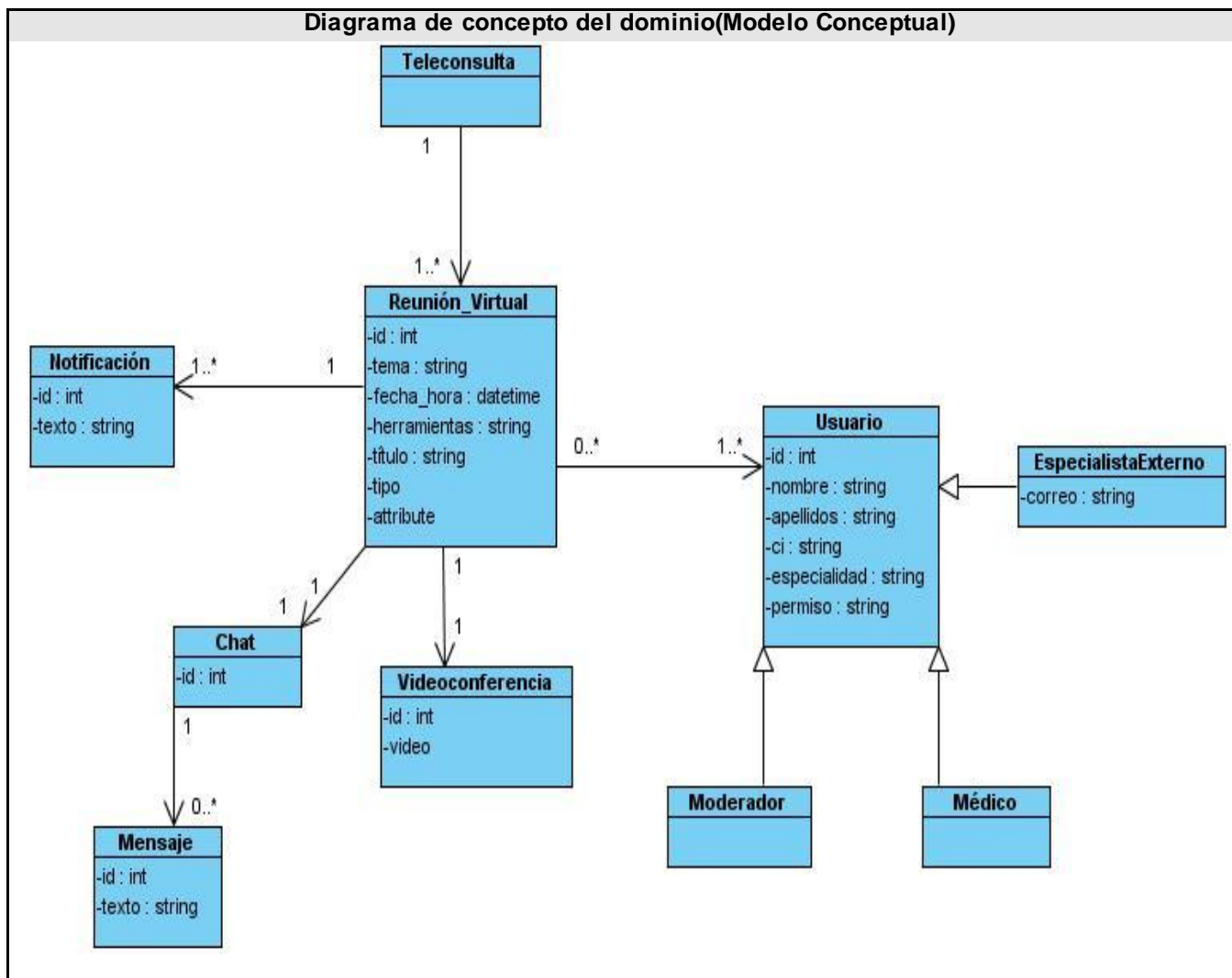


Ilustración 1: Modelo Conceptual

2.3. Especificación de los requisitos de software.

Para llevar a cabo el desarrollo de un sistema que cumpla con los requerimientos establecidos por los clientes finales del software y que cumpla con las normas de calidad de forma concreta se realiza un proceso de descripción de requerimientos. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema.

Capítulo 2: Características del Sistema

La captura de requisitos es de vital importancia debido a que:

- Permite gestionar las necesidades del proyecto en forma estructurada.
- Mejora la capacidad de predecir cronogramas de proyecto proporcionando un punto de partida para controlar actividades específicas.
- Mejora la calidad del software pues si se cumple con todos los requisitos, el software poseerá lo que el cliente desea por lo tanto tendrá buena calidad.
- Evita rechazo de usuarios finales debido a que obliga a los usuarios a considerar sus requerimientos cuidadosamente.

2.3.1. Requerimientos Funcionales.

Los requerimientos funcionales (RF) son características requeridas del sistema que expresan una capacidad de acción del mismo – una funcionalidad; generalmente expresada en una declaración en forma verbal. (27)

RF1: Planificar Reunión.

RF2: Modificar Reunión.

RF3: Suspender Reunión.

RF4: Iniciar Reunión.

RF5: Entrar a Reunión.

RF6: Ver Reunión.

RF7: Crear Audioconferencia.

RF8: Crear Videoconferencia.

RF9: Enviar Notificación.

RF10: Listar Notificaciones.

2.3.2. Requerimientos No Funcionales.

Los requisitos no funcionales (RNF) son características requeridas del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo. (27) Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste.

Capítulo 2: Características del Sistema

Usabilidad: Es la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto.

La Organización Internacional de Normalización (ISO) define usabilidad como:

"La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso" (28)

"Usabilidad es la eficacia, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico" (29)

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en el menor tiempo posible alcanzando un nivel elemental asociado al dominio del sistema y el uso eficiente del mismo, para esto:

- Los usuarios normales necesitarán 20 días de preparación.
- La categoría de usuarios avanzados necesitarán 30 días de preparación.

Fiabilidad: es la probabilidad de que un dispositivo realice adecuadamente su función prevista a lo largo del tiempo, cuando opera en el entorno para el que ha sido diseñado bajo condiciones fijadas. (30) El sistema será diseñado para que se registren todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento mediante los logs, además soportará el uso de firmas digitales para la transferencia de información cuya certificación sea imprescindible para validar el uso de la misma, implementará un control de cambios a determinados campos de información, seleccionados por su importancia, de forma tal que sea posible determinar cuáles han sido las actualizaciones que se le han realizado.

También los usuarios tendrán acceso sólo a los niveles establecidos de acuerdo a la función que realizan, es decir según su rol. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.

Ninguna información que se haya ingresado en el sistema será eliminada, permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas físicamente de la misma,

Capítulo 2: Características del Sistema

independientemente de que para el sistema, este elemento ya no exista y este permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvadas realizadas.

Eficiencia: Es la capacidad para hacer un buen uso de los recursos que se manipula describiendo el rendimiento de la aplicación y los recursos que ésta utiliza bajo condiciones específicas. (31)

El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda. Además permitirá agregar recursos para aumentar el poder de procesamiento y almacenamiento sin afectar los sistemas, garantizando expansiones motivadas por futuros requerimientos.

El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos. Se deberá usar siempre que sea posible el patrón Singleton (instancia única), ya que permite destruir referencias que no están en uso, optimizar el trabajo con cadenas, entre otras buenas prácticas que ayudan a mejorar el rendimiento.

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

Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos por direcciones IP así como la administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación. Se podrán realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados. Va a ser posible el chequeo de las operaciones y acceso de los usuarios al sistema. Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

Seguridad: Se basa en un conjunto de métodos y herramientas destinados a proteger los bienes informáticos de una institución, esta tiene el objetivo de garantizar la confidencialidad, integridad y disponibilidad de los datos.

Capítulo 2: Características del Sistema

El sistema garantizará la confidencialidad de la información manteniéndola protegida contra el acceso de usuarios no autorizados, utilizando mecanismos de validación que puedan garantizar el cumplimiento de: usuario, contraseña y nivel de acceso, de manera que, cada uno pueda tener disponible solamente las opciones relacionadas con su actividad o rol. Así se valida que los datos sean modificados por las personas autorizadas, lo que pone en práctica la integridad de la información.

También se usarán mecanismos de encriptación de los datos que por cuestiones de seguridad, no deben viajar al servidor en texto plano, como es el caso de las contraseñas y se cuenta con la existencia de una bitácora, en la cual se almacena todas las operaciones llevadas a cabo por el usuario, quedando registrado la fecha, la hora, así como la actividad que llevo a cabo el usuario en cuestión.

Interfaz: Entorno que media entre el usuario y los datos almacenados en un archivo o base de datos digital. (32) Este requisito no funcional describe la apariencia del producto y se especifica cómo se tratará de hacer la interfaz de este, contemplando las características del sistema.

Interfaces de usuario: Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con teclas de función y menús desplegables 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 el idioma previamente configurado.

Interfaces de software: Se interactuará con el Sistema de Gestión Hospitalaria (HIS) para la gestión de usuarios y acceso al sistema.

Requisitos para la documentación de usuarios en línea y ayuda del sistema: Se presentará ayuda en línea para orientar en el uso de la interfaz y se desarrollará un manual de usuario para la aplicación.

Hardware: Los requerimientos de hardware estarán dados por la plataforma específica que se utilice para la instalación del sistema, en cuanto a sistema operativo, servidor de aplicaciones y gestor de bases de datos.

En la solución se incluyen estaciones de trabajo que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web, se

Capítulo 2: Características del Sistema

recomienda IE 7, Firefox 3.6 o versiones superiores. Se necesitan estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0 Hz.

Para los servidores la solución estará conformada, fundamentalmente, por alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. Servidores de Base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual – Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual - Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

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.

El sistema debe correr en sistemas operativos Windows, Unix y Linux, utiliza la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL). El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 3.6 o versiones superiores de estos.

Diseño: Especifica o restringe la construcción de un sistema. Son restricciones que han sido ordenadas y deben ser cumplidas estrictamente.

Se utilizará el patrón de diseño Modelo Vista Controlador (MVC). La capa de presentación contendrá todas las vistas y la lógica de la presentación. El flujo web se manejará de forma declarativa y basándose en definiciones de procesos del negocio. La capa del negocio mantendrá el estado de las conversaciones y procesos del negocio que concurrentemente pueden estar bajo la ejecución de varios usuarios. La capa de acceso a datos contendrá las entidades y los objetos de acceso a datos correspondientes a las mismas. El acceso a datos está basado en el estándar JPA y particularmente en la implementación del motor de persistencia Hibernate.

Capítulo 2: Características del Sistema

2.4. Modelo de casos de uso del sistema.

Un modelo de caso de uso es un modelo de los diferentes tipos de usuarios que interactúan con el sistema para resolver un problema. Este sirve como un hilo unificador entre los desarrolladores del sistema. Es usado como la primera especificación de los requisitos funcionales para el sistema, como base para el análisis y diseño, como una entrada para planear la iteración, como las bases para definir los casos de prueba y la documentación del usuario. (33)

2.4.1. Actores del Sistema.

Un actor es algo con comportamiento, como una persona (identificada por un rol), un sistema informatizado u organización, y que realiza algún tipo de interacción con el sistema.

Actores	Justificación
Médico	Es un usuario con una especialidad determinada, participa en la reunión, solicitando información para llegar a un diagnóstico en conjunto con los demás usuarios (especialistas) invitados a esta. Establece comunicación con otro especialista mediante un chat o videoconferencia.
Moderador	Planifica la reunión, envía notificaciones a los usuarios, da la orden de comenzar una reunión y dirige el tema de la conversación en la sala de conferencia.

Tabla 1: Usuarios del Sistema

2.4.2. Listado de casos de uso.

Un caso de uso (CU) es parte del análisis y describe lo que el sistema debe hacer desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

Capítulo 2: Características del Sistema

CU-1	Planificar Reunión.
Actor	Moderador.
Descripción	El CU se inicia cuando se accede a la opción Reunión Virtual. El sistema muestra la interfaz Planificar Reunión, en la cual el usuario podrá introducir un horario, un moderador y un tema, además se le adicionan participantes a los que se les envía una notificación o correo con los datos de la misma.
Referencia	RF 1, RF 9

Tabla 2: Casos de Uso del Sistema: Planificar Reunión

CU-4	Ver Reunión.
Actor	Usuario (Médico, Moderador).
Descripción	El CU se inicia cuando se accede a la opción Ver Reunión, mostrándose un listado de las reuniones a las que está invitado el usuario y las que puede dirigir.
Referencia	RF 6

Tabla 3: Casos de Uso del Sistema: Ver Reunión

CU-5	Iniciar Reunión.
Actor	Moderador.
Descripción	El CU se inicia cuando se accede a la opción Iniciar Reunión, del listado de reuniones, consiste en que todos los invitados podrán comunicarse mediante la red, a través de una videoconferencia o el chat.
Referencia	RF 4, RF 6.

Tabla 4: Casos de Uso del Sistema: Iniciar Reunión

CU-6	Entrar a Reunión.
Actor	Médico.
Descripción	El CU se inicia cuando el usuario desea entrar a la reunión a la cual fue invitado para el intercambio de información, desde su listado de reuniones.
Referencia	RF 5, RF 6.

Tabla 5: Casos de Uso del Sistema: Entrar a Reunión

Capítulo 2: Características del Sistema

2.4.3. Diagrama de Casos de Uso del Sistema.

Un Diagrama de Casos de Uso del Sistema muestra la relación entre los actores y los casos de uso del sistema. Representa las funcionalidades que ofrecerá el sistema. Son importantes para modelar el comportamiento de un sistema, un subsistema o una clase.

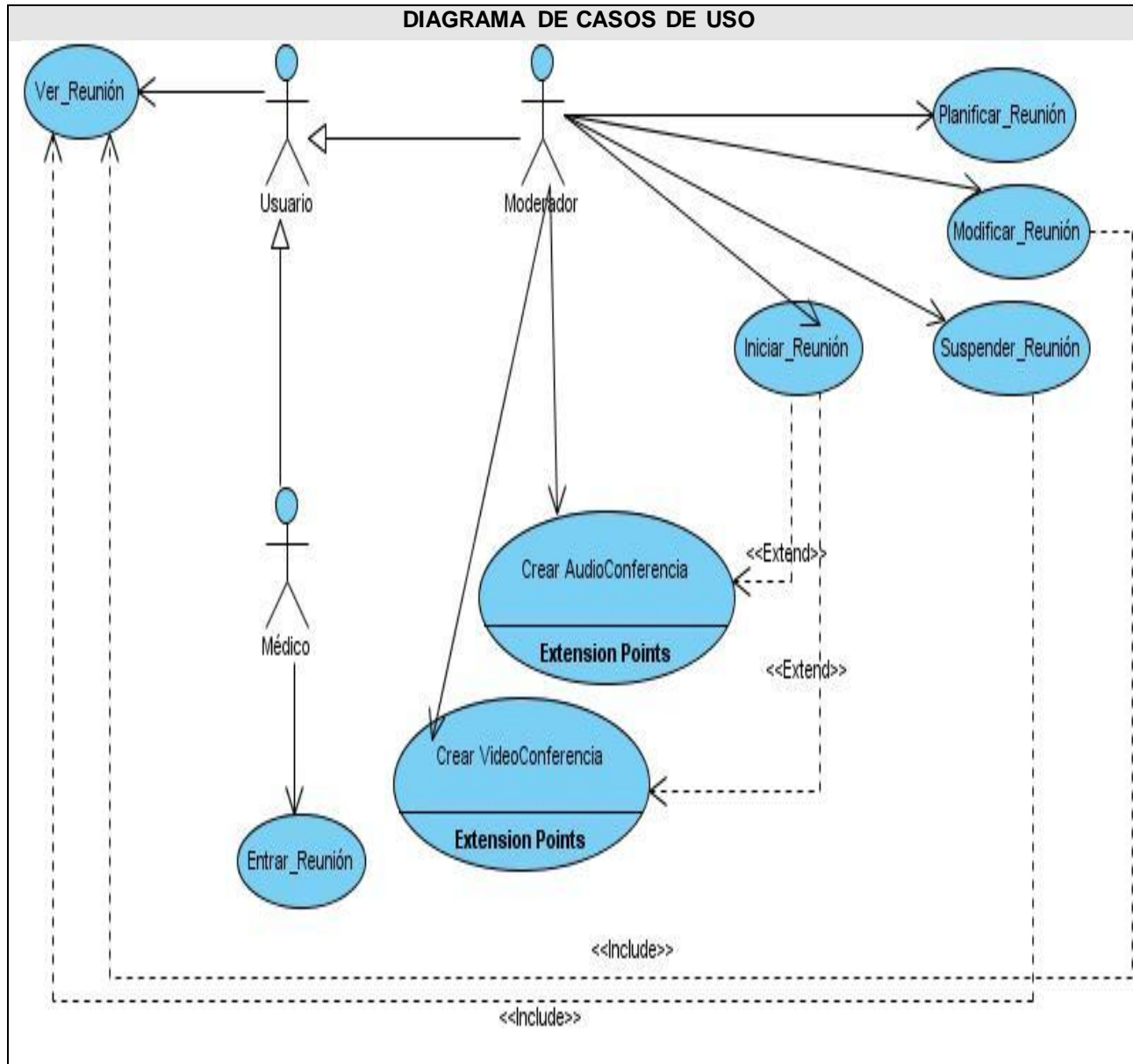


Ilustración 2: Diagrama de Casos de Uso del Sistema

Capítulo 2: Características del Sistema

Casos de uso expandidos.

Un caso de uso esencial expandido UML es una descripción más detallada de los procesos usados para lograr el funcionamiento del sistema. Un caso de uso expandido es construido sobre sobre un caso de uso de alto nivel.

Suspender Reunión.

Caso de uso	
CU-3	Suspender Reunión.
Propósito	Suspender la reunión antes planificada.
Actores	Moderador.
Resumen:	El CU se inicia cuando se accede a la opción Suspender Reunión del listado de reuniones, consiste en suspender una reunión, en caso de no cumplir con los objetivos que se deseaban, luego se enviará una notificación a los invitados informándole de lo sucedido.
Referencias	RF 3, RF 6, RF 9.
Precondiciones:	El usuario se debe haber autenticado. Se debe haber planificado la reunión a suspender.
Acción del actor	Respuesta del sistema
	1. El sistema muestra las reuniones planificadas en la Interfaz Ver Reunión.
2. Selecciona la reunión a suspender y accede a la opción Suspender Reunión.	3. Muestra un mensaje de información "Se suspenderá la reunión seleccionada y quedará eliminada de la base de datos. Al Aceptar se enviará una notificación a los participantes de la misma. ¿Desea continuar?"
4. Selecciona la opción Aceptar	5. Elimina la reunión y se manda una

Capítulo 2: Características del Sistema

4.1 Si selecciona Cancelar. Ver descripción de Flujo Alterno "Cancelar"	notificación a los usuarios invitados a esta
Alternativa "Cancelar".	
Acción del actor	Respuesta del sistema
	4.1. Muestra la interfaz anterior

Tabla 6: Casos de Uso del Sistema: Suspender Reunión

Modificar Reunión.

Caso de uso	
CU-2	Modificar Reunión.
Propósito	Modificar la reunión antes planificada de acuerdo a las nuevas necesidades.
Actores	Moderador.
Resumen:	El CU se inicia cuando se accede a la opción Modificar Reunión del listado de reuniones, consiste en modificar datos de una reunión, el cual tiene como finalidad corregir los datos e invitar a otros participantes de ser necesario, luego se enviaría una notificación con los cambios correspondientes a los participantes.
Referencias	RF 2, RF 6, RF 9
Precondiciones:	El usuario se debe haber autenticado. Se debe haber planificado la reunión a modificar.
Acción del actor	Respuesta del sistema
	1. El sistema muestra las reuniones planificadas en la Interfaz Ver Reunión
2. El actor accede a la opción Modificar Reunión de la reunión deseada.	3. El sistema muestra la interfaz Modificar Reunión con los datos de la reunión seleccionada.
4.1 Si se desea modificar los datos ver Alternativa "Modificar Datos" 4.2 Si lo que desea modificar son los integrantes 4.2.1 Si son usuarios internos "Modificar Internos"	5. Guarda en la base de datos los cambios realizados, muestra el mensaje "Se han modificado los datos correctamente" y envía notificación a todos los usuarios

Capítulo 2: Características del Sistema

<p>4.2.2 Si son usuarios externos "Modificar Externos"</p> <p>4.3 Si se van a modificar todos los datos ver Alternativa 1.</p>	<p>invitados a esta reunión. Culmina el CU.</p>
Flujo alternativo	
Alternativa " Modificar Datos "	
Acción del actor	Respuesta del sistema
<p>4.1.1 Modifica los datos como</p> <p>Titulo</p> <p>Horario</p> <p>Tipo</p> <p>Tema</p> <p>Fecha</p> <p>Descripción</p> <p>Y acepta los cambios realizados.</p>	
Alternativa " Modificar Internos "	
Acción del actor	Respuesta del sistema
<p>4.2.1.1 Si se desea excluir algún invitado Ver Alternativa " Excluir Usuario "</p> <p>4.2.1.2 Si desea invitar usuarios nuevos Ver Alternativa "Nuevos Usuarios"</p>	
Alternativa "Modificar Externos"	
Acción del actor	Respuesta del sistema
<p>4.2.2.1 Si se desea excluir algún invitado Ver Alternativa " Excluir Usuario "</p> <p>4.2.2.2 Si se desea Invitar nuevos usuarios externos ver Alternativa: "Nuevos Usuarios Externos"</p>	
Alternativa1	
Acción del actor	Respuesta del sistema

Capítulo 2: Características del Sistema

4.3.1 (Ver 4.1 y 4.2)	
Alternativa " Excluir Usuario "	
Acción del actor	Respuesta del sistema
4.2.1.1.1 Selecciona el invitado a excluir de la reunión y da clic en el botón "Eliminar".	4.2.1.1.2 Muestra el mensaje "El invitado va a ser excluido de la Reunión, se le hará llegar una notificación de aviso, desea continuar."
4.2.1.1.3 Selecciona la opción "Aceptar." En caso de "Cancelar" ver Alternativa 2	4.2.1.1.4 Se elimina el usuario de la reunión y se le manda una notificación.
Alternativa " Nuevos Usuarios "	
Acción del actor	Respuesta del sistema
4.2.1.2.1 Selecciona los invitados o realiza una búsqueda según el nombre, los apellidos y la especialidad.	
Alternativa " Nuevos Usuarios Externos "	
Acción del actor	Respuesta del sistema
4.2.2.2.1 Escribe la dirección de correo electrónico del invitado.	
Alternativa 2	
Acción del actor	Respuesta del sistema
	4.2.1.1.4 Muestra la interfaz anterior.

Tabla 7: Casos de Uso del Sistema: Modificar Reunión

Caso de uso	
CU-7	Crear Audioconferencia.
Propósito	Se realiza la comunicación mediante audio.
Actores	Moderador.
Resumen:	El CU se inicia cuando el actor va a comenzar la reunión, accede a esta opción para tener comunicación verbal con los demás médicos.

Capítulo 2: Características del Sistema

Referencias	RF 5, RF 6, RF 7.	
Precondiciones:	El usuario se debe haber autenticado. Se debe haber iniciado la reunión por el moderador.	
	Acción del actor	Respuesta del sistema
	1. El CU inicia cuando el actor accede a la opción Iniciar Reunión de la Interfaz Ver Reunión para comenzar el diálogo de la reunión.	2. El sistema para crear la audioconferencia comprueba según el parámetro enviado de la planificación de la reunión guardado en memoria si lo que le pasa es de tipo audioconferencia, comprueba si es usuario y la reunión planificada y crea la audioconferencia y muestra la ventana de la reunión.

Tabla 8: Casos de Uso del Sistema: Crear Audioconferencia

Caso de uso		
CU-8	Crear Videoconferencia.	
Propósito	Se realiza la comunicación mediante video.	
Actores	Moderador.	
Resumen:	El CU se inicia cuando el actor va a comenzar la reunión, accede a esta opción para tener comunicación visual con los demás médicos.	
Referencias	RF 5, RF 6, RF 8.	
Precondiciones:	El usuario se debe haber autenticado. Se debe haber iniciado la reunión por el moderador.	
	Acción del actor	Respuesta del sistema
	1. El CU inicia cuando el actor accede a la opción Iniciar Reunión de la Interfaz Ver Reunión para comenzar el intercambio de información de la reunión.	2. El sistema para crear la videoconferencia comprueba según el parámetro enviado de la planificación de la reunión guardado en memoria si lo que le pasa es de tipo videoconferencia, comprueba si es usuario y la reunión planificada y crea la

Capítulo 2: Características del Sistema

	videoconferencia y muestra la ventana de la reunión.
--	--

Tabla 9: Casos de Uso del Sistema: Crear Videoconferencia

La elaboración y análisis del presente capítulo permitió obtener la descripción de los conceptos del dominio relacionados con la planificación de reuniones virtuales, realizando así el modelo de dominio. Se definieron los Requerimientos a cumplir por parte del sistema, tanto Funcionales como No Funcionales basado en las necesidades del cliente para el éxito del mismo cuando esté terminado, a partir de estos se especificaron los actores y casos de usos del mismo, que son las funcionalidades que se van a implementar.

Capítulo 3: Análisis y diseño del sistema

CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA

Con el desarrollo del capítulo actual se realizará el análisis y diseño de la propuesta de solución, se definirá la arquitectura y los patrones a utilizar para el diseño e implementación del sistema, obteniéndose los principales artefactos como son los diagramas de clases del análisis y del diseño y su descripción. Por último se describirán las principales clases del diseño para lograr un mayor entendimiento de estas para su implementación.

3.1. Arquitectura.

La Arquitectura de Software establece los fundamentos para que analistas, diseñadores, programadores, y demás trabajadores, se enfoquen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. También expresa un esquema organizativo estructural fundamental para el sistema, además de especificar la comunicación entre los componentes del software y puede involucrar varios patrones de diseño.

“La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. (34)

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño es una solución a un problema de diseño. Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.

Capítulo 3: Análisis y diseño del sistema

- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

Para la realización del módulo “Gestión de Reuniones del Sistema Teleconsulta” se propone el uso de los patrones arquitectónicos Modelo Vista Controlador (MVC) y en capas, debido a que son los más utilizados, porque permite la reusabilidad de componentes y el Sistema de Gestión Hospitalaria (HIS), donde se va a integrar este módulo, los tiene definidos, como patrones arquitectónicos.

Modelo Vista Controlador.

El Modelo Vista Controlador es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: el Modelo, las Vistas y los Controladores.

El MVC en una aplicación web, se representa a través de la Vista que constituyen las páginas Lenguaje de Marcado de Hipertexto (HTML) y el código que provee datos dinámicos a la página, que permite mostrarle la información al usuario debido a la captura de datos del sistema al exterior y viceversa. El Controlador gestiona todos los eventos de entrada del usuario y traslada las peticiones de la Vista a la Modelo, mediante el intermediario entre ellas. El Modelo contiene toda la información persistente y reglas de negocio.

El uso de MVC ayuda a darle cierta estructura lógica a las aplicaciones. Su principal objetivo es separar la lógica del negocio de la lógica de presentación o interfaz; permite así tener dividido el sistema, una de las razones principales para poder dar mantenimiento fácil, generalmente esta arquitectura es usada en grandes sistemas a nivel empresarial.

Patrón en capas.

En el diseño de sistemas informáticos actual se suele usar las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables, que pueden ampliarse con facilidad en caso de que las necesidades aumenten. (35)

Capítulo 3: Análisis y diseño del sistema

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

Actualmente el diseño es en tres niveles o en tres capas: (36)

Capa de presentación: Es la que ve el usuario, hay quien la denomina “capa de usuario”, presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio.

Capa de negocio: Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Es donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

Capa de datos: Es donde residen los datos. Está formada por uno o más gestor de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

3.2. Estrategias de integración.

El módulo “Gestión de Reuniones del Sistema Teleconsulta” 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. Modelo de Análisis.

Durante el análisis, se analiza los requisitos con mayor profundidad, utilizando el lenguaje de los desarrolladores, refinándolos y estructurándolos. El objetivo es conseguir una comprensión más precisa de estos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el

Capítulo 3: Análisis y diseño del sistema

sistema entero. El análisis prepara y simplifica la actividad de diseño e implementación, delimitando los temas que debe resolverse y las decisiones que deben tomarse en esas actividades. (37)

Clases de análisis.

Clases de análisis: Se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación / composición, generalización / especialización y tipos asociativos. RUP propone clasificar a las clases en: interfaz, control y entidad.

Clase Interfaz: Se utilizan para modelar la interacción entre el sistema y sus actores. Esta interacción a menudo implica recibir información y peticiones de los usuarios y sistemas externos. Cada clase de interfaz debe asociarse con al menos un actor.

Clase Control: Se usa para coordinar la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso, por lo que definen el flujo de control y las transacciones dentro de un caso de uso delegando el trabajo a otros objetos.

Clase Entidad: Estas modelan información que posee una larga vida y que a menudo es persistente y fenómenos, conceptos y sucesos que ocurren en el mundo real. La fuente principal de obtención son las clases entidades del negocio y el glosario de términos que se ha ido elaborando.

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.

A continuación se muestran los diagramas de clases del análisis de los Casos de Usos: Planificar Reunión, Suspender Reunión y Ver Reunión, para ver los restantes diagramas, consultar [Anexo 1](#).

Capítulo 3: Análisis y diseño del sistema

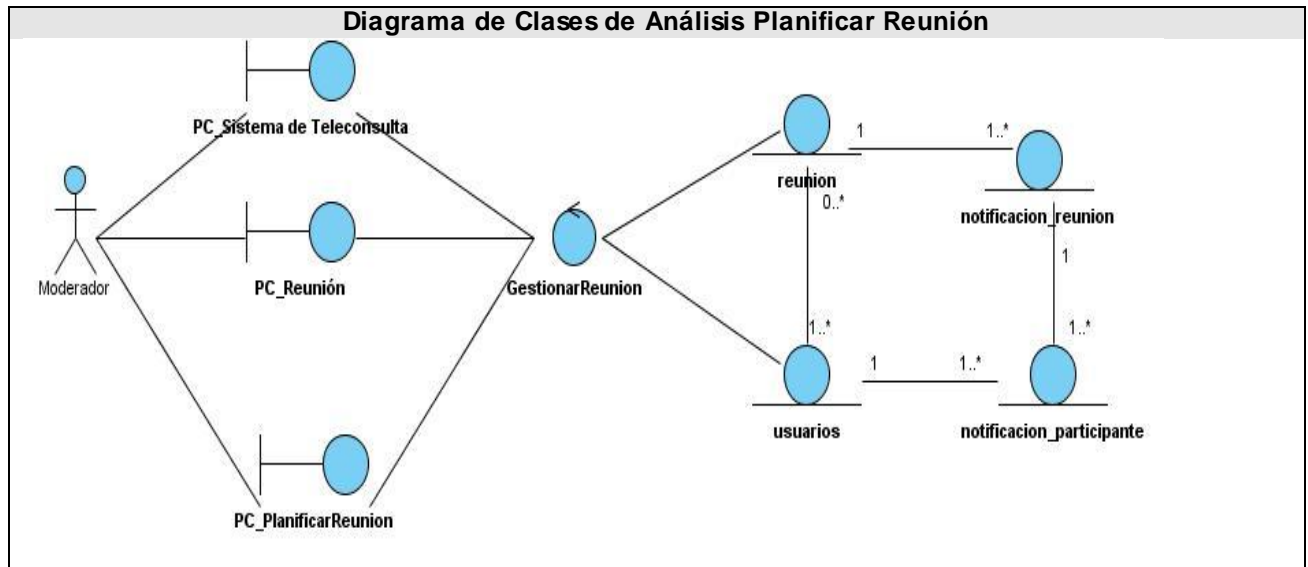


Ilustración 3: Diagrama de Clase del Análisis: Planificar Reunión

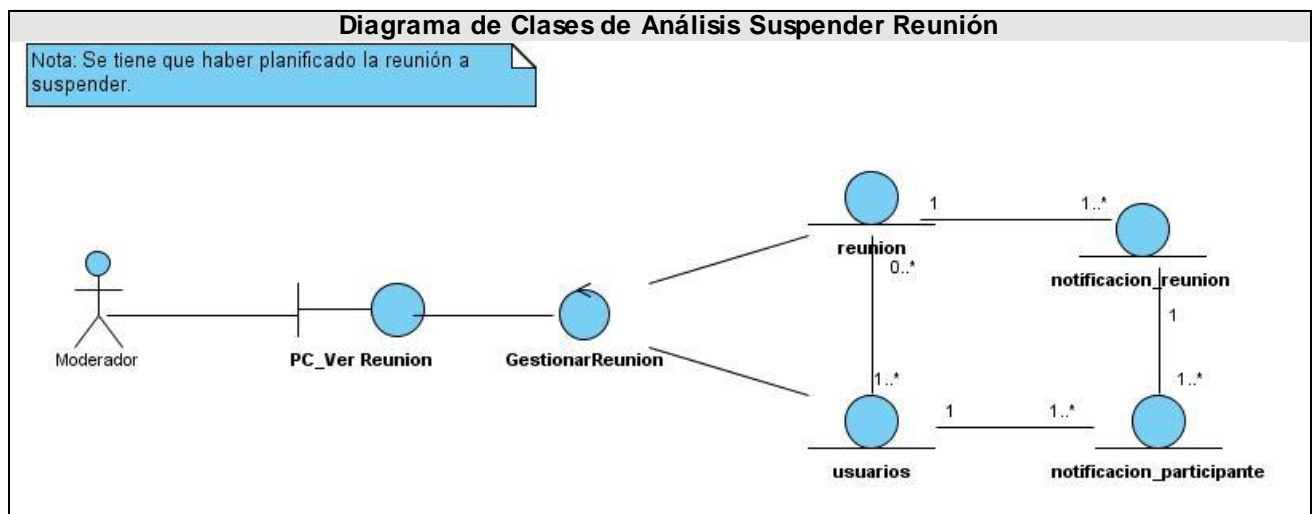


Ilustración 4: Diagrama de Clase del Análisis: Suspender Reunión

Capítulo 3: Análisis y diseño del sistema

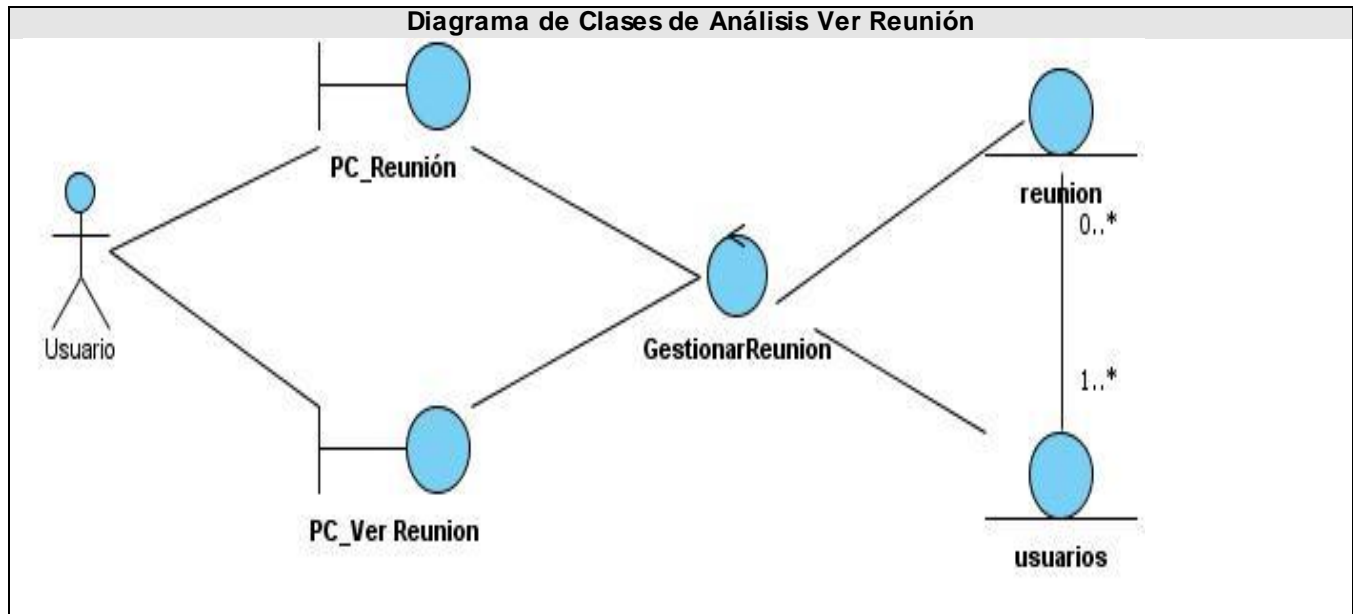


Ilustración 5: Diagrama de Clase del Análisis: Ver Reunión

3.4. Modelo de Diseño.

En el diseño se modela el sistema y es una forma de darle vida a todos los requisitos incorporados en el mismo, incluyen componentes de código que se compilan e integran en versiones ejecutables del sistema. (37)

El diseño debe implementar todos los requisitos explícitos contenidos en el modelo de análisis y debe acumular todos los requisitos implícitos que desea el cliente. Este es una guía que puedan entender los implementadores y los que prueban y mantienen el software.

El modelo de diseño es un modelo de objetos que describe la realización de los casos de usos, y sirve como una abstracción del modelo de implementación y el código fuente. Es usado como una entrada inicial en las actividades de implementación y prueba.

Diagramas de interacción (Secuencia).

Un diagrama de interacción es un artefacto que explica gráficamente cómo los objetos interactúan a través de mensajes para realizar las tareas, modelando el comportamiento dinámico del sistema. Estos

Capítulo 3: Análisis y diseño del sistema

constituyen una de las herramientas más importantes para el análisis y diseño orientado a objetos pues describen la interacción entre los objetos. (38)

Estos diagramas son muy útiles para visualizar, especificar, construir y documentar la dinámica entre dos objetos. Hay dos tipos de diagrama de interacción: Diagramas de Secuencia y Diagramas de Colaboración.

Un diagrama de Secuencia muestra una interacción ordenada según la secuencia temporal de eventos. Representa los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción, sin un orden prefijado. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba abajo. Se pueden colocar etiquetas (como restricciones de tiempo y descripciones de acciones) bien en el margen izquierdo o bien junto a las transiciones o activaciones a las que se refieren. (39)

Para obtener información de los diagramas de secuencia, ver [Anexo 2](#).

Diagramas de Clases del Diseño.

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación para satisfacer los detalles de la implementación. Estos se utilizan para modelar la vista de diseño estática de un sistema y contienen información como clases, asociaciones y atributos, interfaces, con sus operaciones y constantes, métodos, información sobre los tipos de los atributos, navegabilidad y dependencias.

A continuación se muestran algunos diagramas de clases del diseño, pertenecientes a los casos de usos: Planificar Reunión, Suspende Reunión y Ver Reunión, para obtener más información sobre los restantes diagramas consultar: [Anexo 3](#).

Capítulo 3: Análisis y diseño del sistema

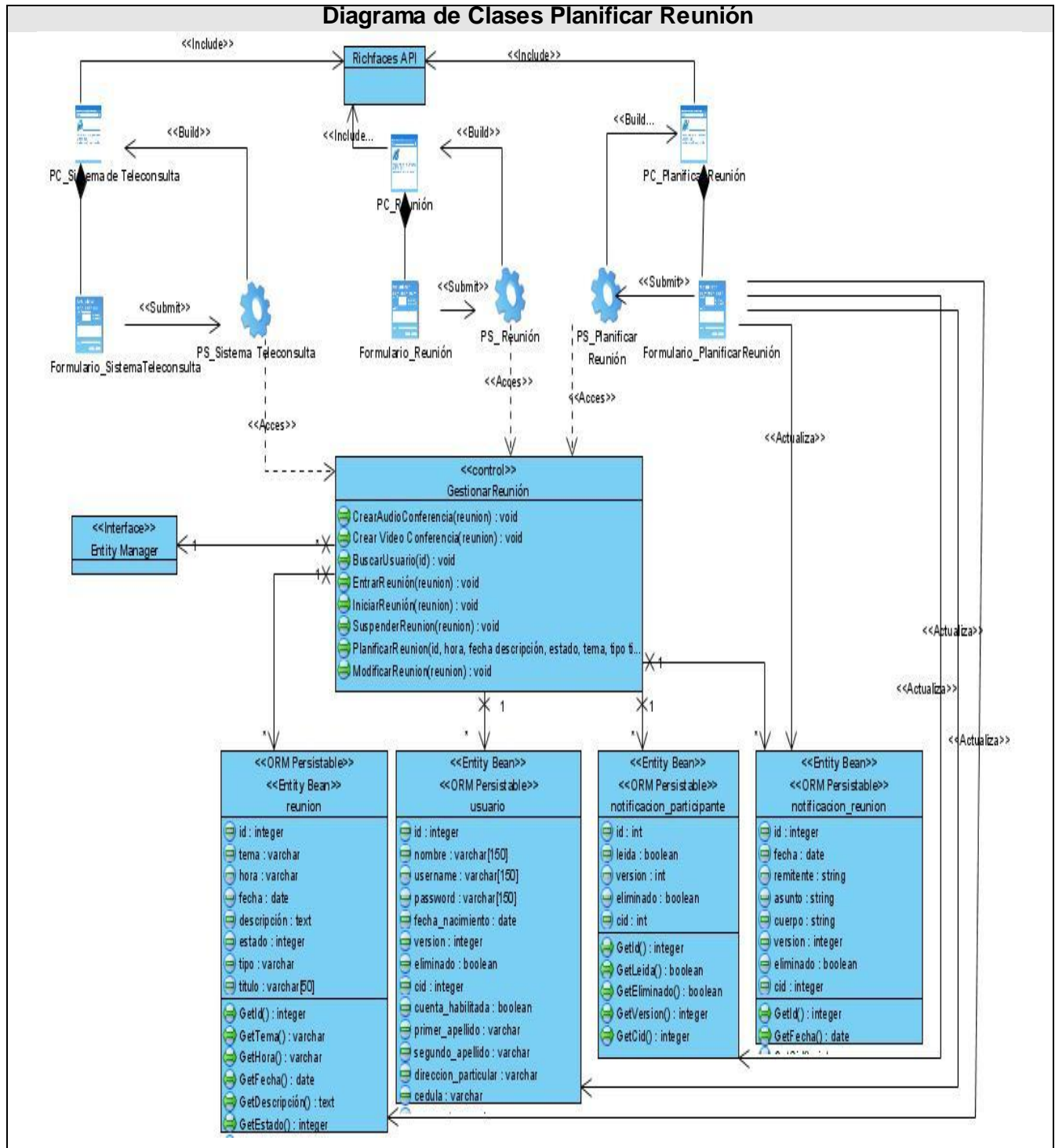


Ilustración 6: Diagrama de Clases del Diseño: Planificar Reunión

Capítulo 3: Análisis y diseño del sistema

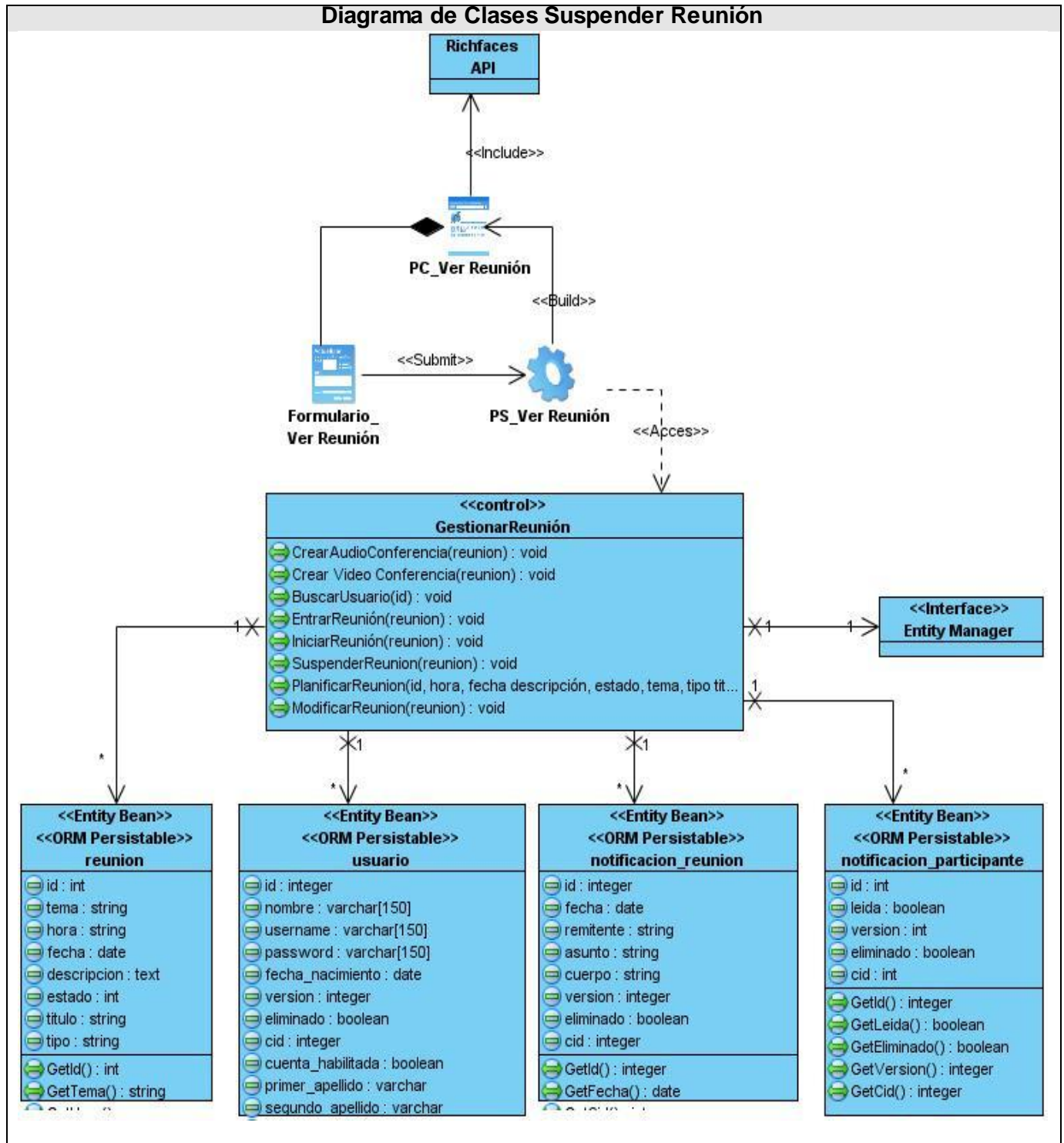


Ilustración 7: Diagrama de Clases del Diseño: Suspende Reunión

Capítulo 3: Análisis y diseño del sistema

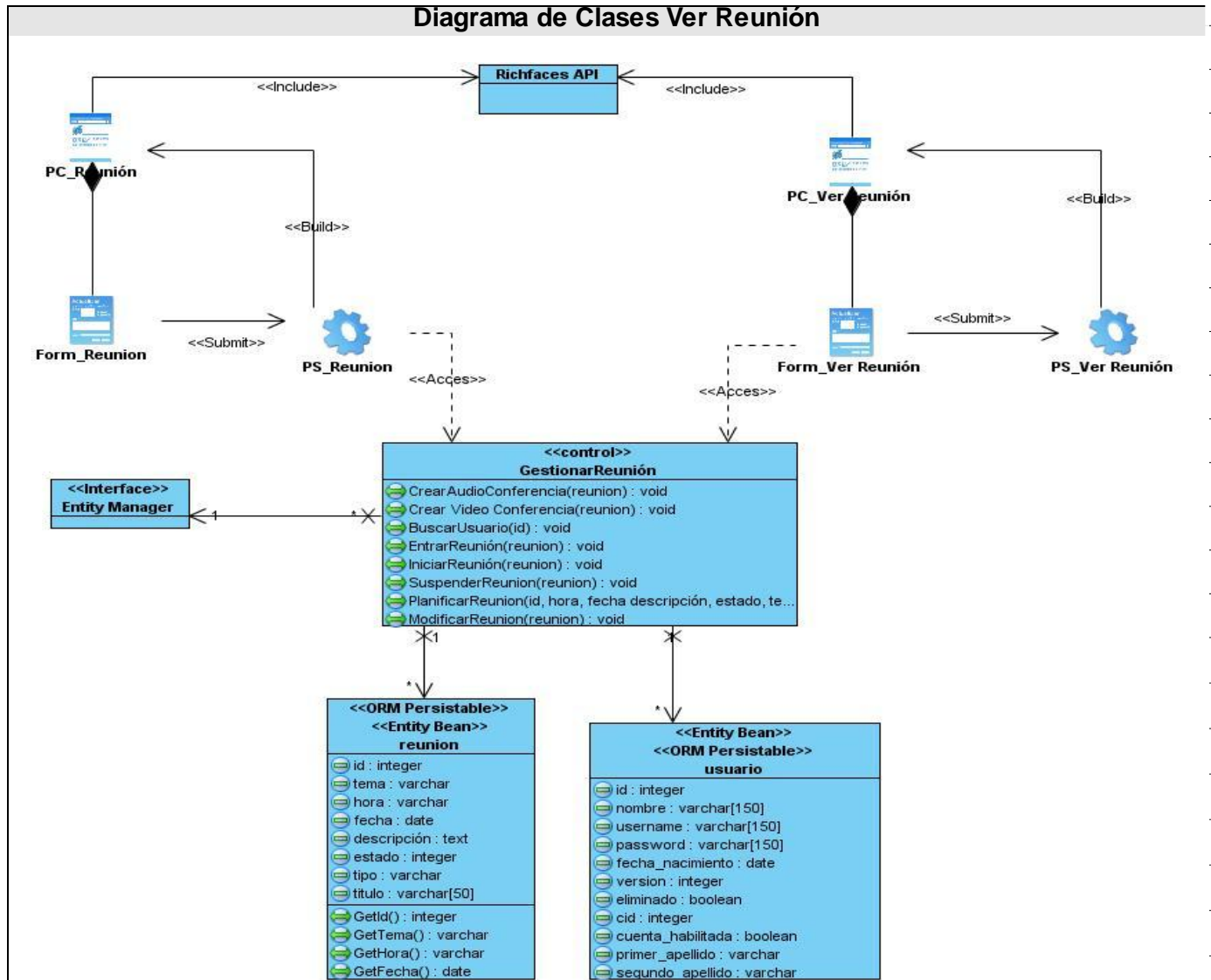


Ilustración 8: Diagrama de Clases del Diseño: Ver Reunión

Descripción de las clases.

En las tablas que a continuación se presentan, no existen acentos, ni caracteres extraños, con el objetivo de evitar errores e incompatibilidades pues no todos los compiladores los reconocen y además la codificación del fichero de código fuente, al cambiar de una estación a otra, puede provocar afectaciones en su visibilidad.

Capítulo 3: Análisis y diseño del sistema

Clases Interfaz:

Nombre: Reunion	
Tipo de clase: Interfaz	
Responsabilidad:	
Descripción:	Representa la página para acceder al menú del Sistema de Teleconsulta.

Tabla 10: Tabla 8: Clase Interfaz: Reunión

Nombre: planificar_reunion	
Tipo de clase: Interfaz	
Responsabilidad:	
Descripción:	Muestra los formularios a llenar para planificar la reunión con todas sus características.

Tabla 11: Clase Interfaz: Planificar Reunión

Nombre: modificar_reunion	
Tipo de clase: Interfaz	
Responsabilidad:	
Descripción:	Muestra las características de la reunión planificada seleccionada y brinda la opción de realizar cambios en ella.

Tabla 12: Clase Interfaz: Modificar Reunión

Nombre: ver_reunion	
Tipo de clase: Interfaz	
Responsabilidad:	
Descripción:	Muestra el listado de las reuniones planificadas a las que está invitado el usuario y las que planificó.

Tabla 13: Clase Interfaz: Ver Reunión

Capítulo 3: Análisis y diseño del sistema

Clases Controladora:

Nombre: GestionarReunion	
Tipo de clase Controladora	
Atributo	Tipo
id	integer
tema	varchar
hora	varchar
fecha	date
descripcion	text
estado	integer
tipo	varchar
titulo	varchar
Para cada responsabilidad:	
Nombre:	Crear_Reunion(idmoderador)
Descripción:	En este método se planifica la reunión con su título, fecha, hora, usuarios, tipo y tema.
Nombre:	Modificar_Reunion (idreunion)
Descripción:	Este método permite realizar cambios en la reunión seleccionada.
Nombre:	Suspender_Reunion(idreunion)
Descripción:	Este método cancela la reunión planificada.
Nombre:	Listar_Reunion(idusuario)
Descripción:	Este método lista las reuniones a que está invitado el usuario y las que dirige en caso de ser moderador.
Nombre:	Buscar_Usuario()
Descripción:	Este método lista los usuarios pertenecientes al sistema.
Nombre:	Entrar_Reunion(idreunion)
Descripción:	Este método permite a los usuarios entrar a la reunión planificada para realizar un intercambio de informaciones.

Capítulo 3: Análisis y diseño del sistema

Nombre:	Iniciar_Reunion(idreunion)
Descripción:	Este método es el que le permite al moderador iniciar la reunión planificada.
Nombre:	Enviar_Notif(reunion)
Descripción:	Este método es el que permite mantener informados a los usuarios de las reuniones planificadas y sus cambios.
Nombre:	Crear_Audio(idreunion)
Descripción:	Este método permite comenzar el intercambio de información mediante audio.
Nombre:	Crear_Video(idreunion)
Descripción:	Este método permite comenzar el intercambio de información mediante video y audio.
Nombre:	Listar_Notif(idusuario)
Descripción:	Este método permite listar las notificaciones de un usuario.

Tabla 14: Clase Controladora: Gestionar Reunión

Para ver las restantes tablas consultar el [Anexo 4](#).

En la confección de este capítulo se profundizó en el estudio de la arquitectura para el desarrollo del Módulo de Gestión de Reuniones del Sistema Teleconsulta: la arquitectura en tres capas y la arquitectura basada en el patrón Modelo-Vista-Controlador, definida por el Sistema de Gestión Hospitalaria (HIS). Se realizaron los diagramas de clases del análisis y los de interacción, que facilitaron una primera aproximación al modelo del diseño, lo que ayudó definir las clases más significativas del mismo, con sus atributos y métodos, y permitió al implementador un mejor entendimiento del diseño del sistema.

Capítulo 4: Implementación

CAPÍTULO IMPLEMENTACIÓN 4

En el presente capítulo se describe todo lo relacionado con la implementación del sistema de acuerdo a la arquitectura propuesta por el diseño. Se representa el modelo de datos, así como la explicación de sus tablas y sus atributos, obteniéndose el diagrama de clases persistentes para generar a partir del mismo la base de datos. Además se modelan los diagramas de componentes y de despliegue, lo que posibilita que quede conformado el modelo de implementación. También, se explica cómo se realiza el tratamiento de errores, la seguridad, los estándares y estilos a utilizar.

4.1. Modelo de Datos.

Un modelo de datos es un conjunto de conceptos, reglas y convenciones que permiten describir y en ocasiones manipular los datos del mundo real que se desee almacenar en la base de datos. Al producto del modelo de datos se le llama esquema, descripción de la estructura de la base de datos, y a los datos en concreto almacenados en la base de datos en ese momento, ocurrencia del esquema. (40)

Un modelo de datos es por tanto 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.

Capítulo 4: Implementación

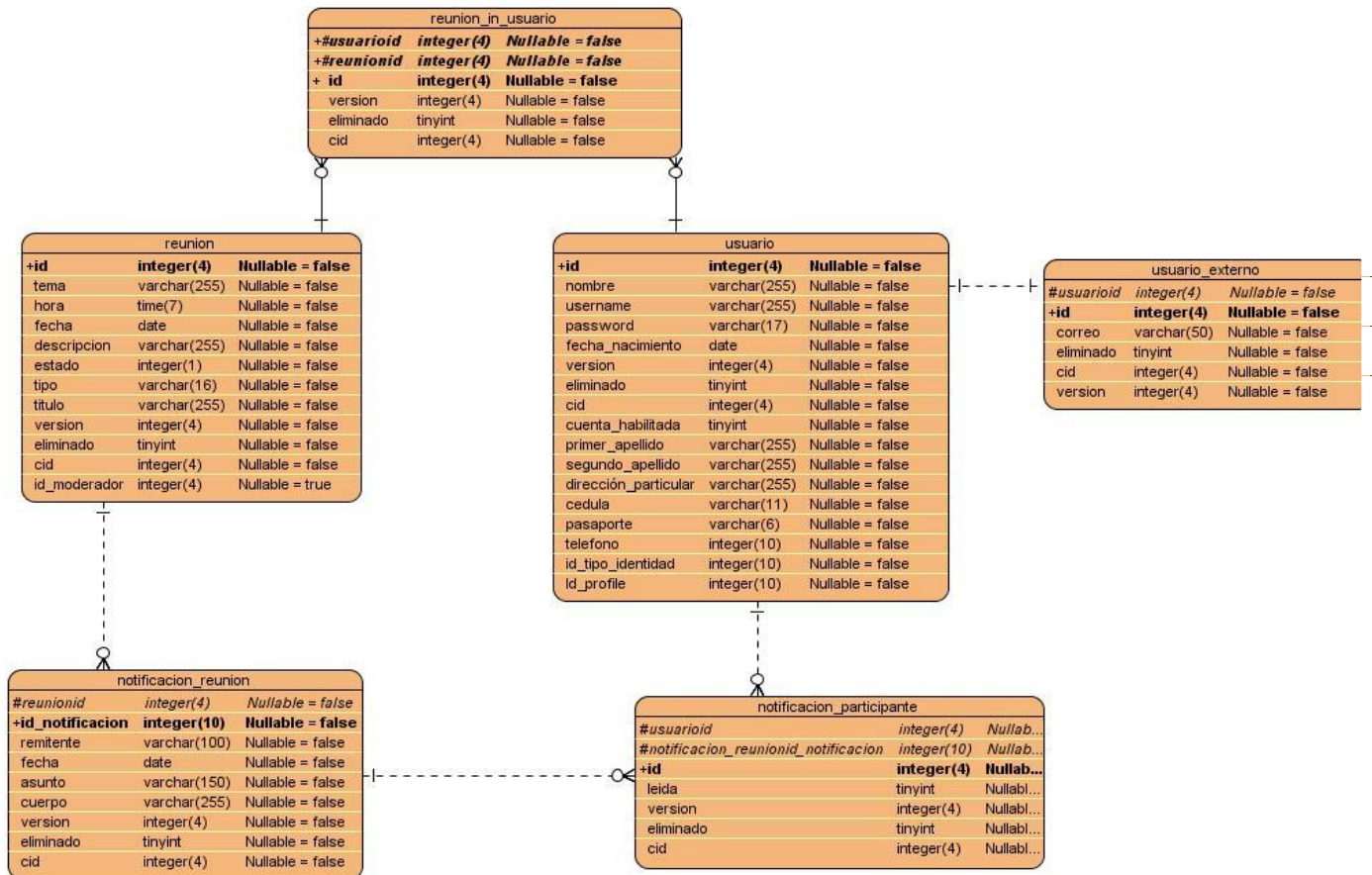


Ilustración 9: Modelo de Datos

Descripción de las tablas del Modelo de Datos.

En las tablas que a continuación se presentan, no existen acentos, ni caracteres extraños, con el objetivo de evitar errores e incompatibilidades pues no todos los compiladores los reconocen y además la codificación del fichero de código fuente, al cambiar de una estación a otra, puede provocar afectaciones en su visibilidad.

A continuación se muestra la descripción de algunas tablas del modelo de datos, para ver las restantes consultar, [Anexo 5](#).

Capítulo 4: Implementación

Nombre: reunion		
Descripción: Esta tabla almacena toda la información necesaria de las reuniones a desarrollar.		
Atributo	Tipo	Descripción
id	integer	En este atributo se almacena el identificador de la reunión.
tema	varchar	En este atributo se almacena el tema de la reunión.
hora	time	En este atributo se almacena la hora en que se va a comenzar la reunión.
fecha	date	En este atributo se almacena la fecha a realizarse la reunión.
descripcion	varchar	En este atributo se almacena todo lo referente que el moderador quiera darle a conocer a los invitados de la reunión.
estado	integer	Este atributo indica el estado de la reunión, que puede ser no iniciada, iniciada o terminada.
id_moderador	integer	En este atributo se almacena el identificador el moderador.
tipo	varchar	Este atributo indica si la reunión va a ser una audioconferencia o videoconferencia.
titulo	varchar	En este atributo se almacena el título de la reunión.
version	integer	En este atributo se almacena el número de actualizaciones a la que ha sido sometida la tupla.
eliminado	boolean	Este atributo indica si la reunión es visible para el sistema.
cid	integer	Este atributo indica el identificador de

Capítulo 4: Implementación

		modificaciones registradas en la bitácora.
--	--	--

Tabla 15: Descripción de las tablas del Modelo de Datos: Reunión

Nombre: notificacion_reunion		
Descripción: Esta tabla almacena todas las notificaciones de las reuniones.		
Atributo	Tipo	Descripción
id_notificacion	integer	En este atributo se almacena el identificador de la notificación perteneciente a la reunión.
remitente	varchar	En este atributo se almacena el usuario que manda la notificación de la reunión.
fecha	date	En este atributo se almacena la fecha en que fue creada la notificación de la reunión.
asunto	varchar	En este atributo se almacena el asunto de la notificación.
cuerpo	varchar	En este atributo se almacena lo que se le desea mandar a los usuarios en la notificación.
version	integer	En este atributo se almacena el número de actualizaciones a la que ha sido sometida la tupla.
eliminado	boolean	Este atributo indica si la notificación es visible para el sistema.
cid	integer	Este atributo indica el identificador de modificaciones registradas en la bitácora.

Tabla 16: Descripción de las tablas del Modelo de Datos: Notificación Reunión

Capítulo 4: Implementación

4.2. Modelo de Implementación.

El Modelo de Implementación muestra cómo se traduce el Modelo de Diseño en los distintos componentes ejecutables de la aplicación a desarrollar. Este consiste en una visión general de lo que tiene que ser implementado en cada iteración, con los componentes y subsistemas a implementar durante esa iteración, así como los resultados de software que se han de obtener y el testeo que se ha de realizar sobre ellos.

El modelo de implementación del Módulo de Gestión de Reuniones del Sistema de Teleconsulta, expone una organización en capas, jerarquías de paquetes y subsistemas de implementación. Los mismos contienen componentes y sus relaciones y divide el sistema en partes más manejables. Esto posibilita la reutilización, que se pueda implementar por separado y disminuye el impacto que pueda traer consigo un cambio.

4.2.1. Diagrama de despliegue.

El despliegue es la etapa del desarrollo que describe la configuración del sistema para su ejecución en un ambiente del mundo real. Para el despliegue se deben tomar decisiones sobre los parámetros de la configuración, funcionamiento, asignación de recursos, distribución y concurrencia.

El Diagrama de Despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. También muestra la configuración de nodos que participan en la ejecución y de los componentes que residen en ellos.

Capítulo 4: Implementación

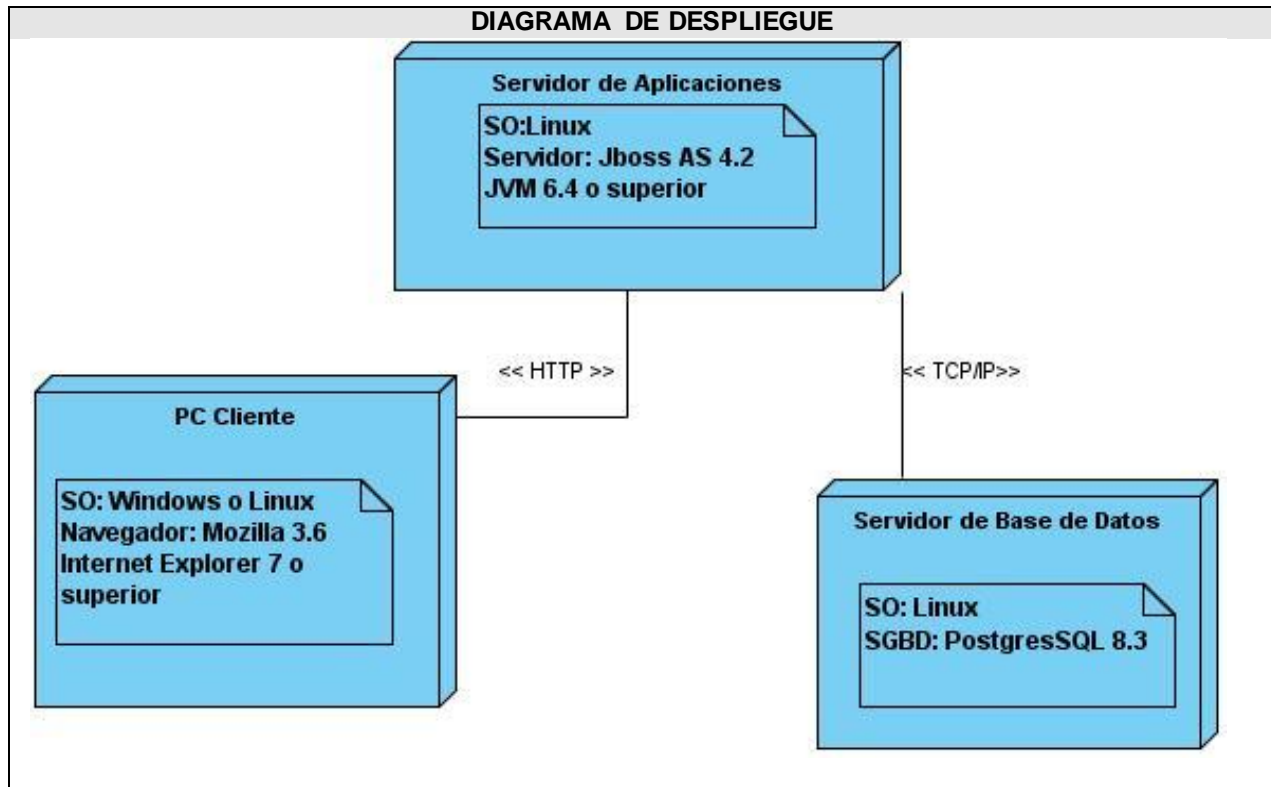


Ilustración 10: Diagrama de Despliegue

Diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas, pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente.

También, los diagramas de componentes permiten visualizar con más facilidad la estructura general del sistema y el comportamiento del servicio que estos componentes proporcionan y utilizan a través de las interfaces. Normalmente un mismo componente se puede implementar por más de una clase, por lo que la representación de un componente abarca una gran parte del sistema.

Capítulo 4: Implementación

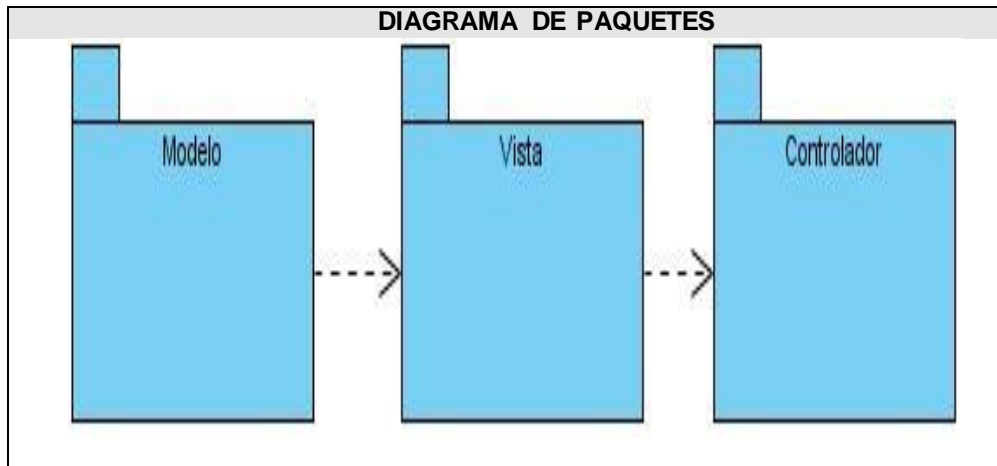


Ilustración 11: Diagrama de Paquetes

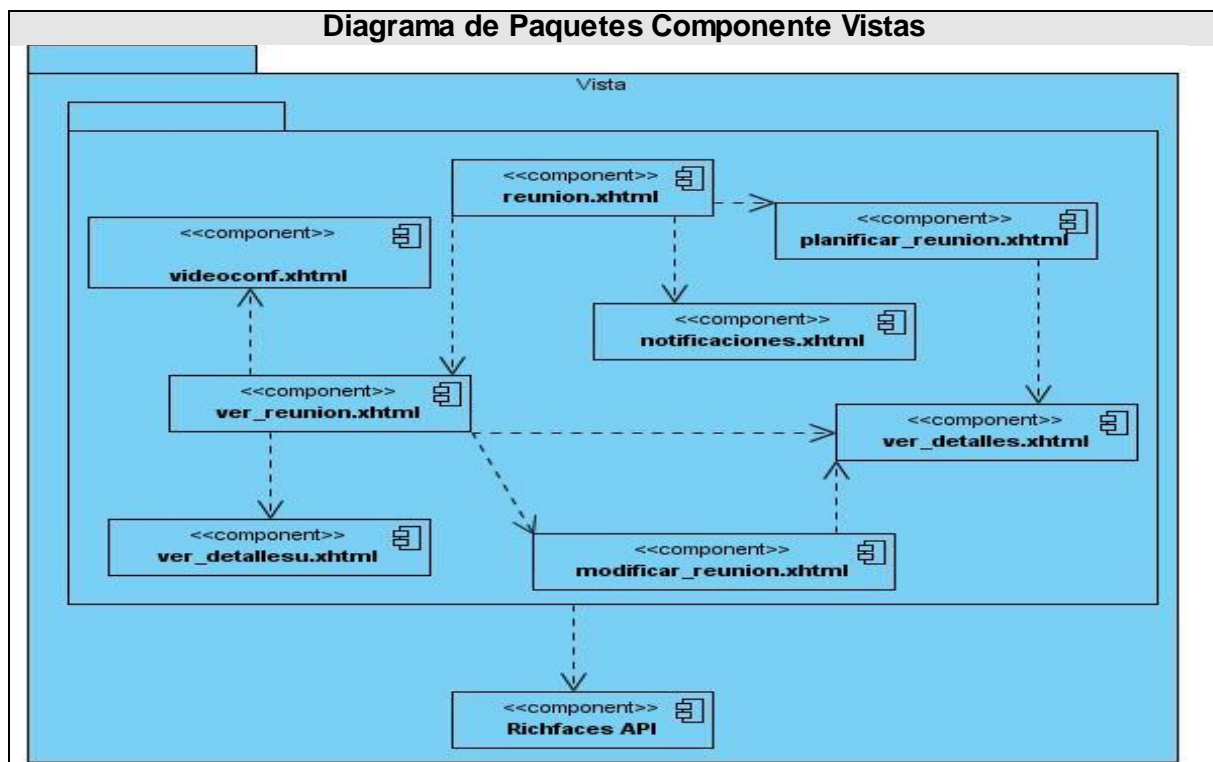


Ilustración 12: Diagrama de Paquetes Componente Vistas

Capítulo 4: Implementación

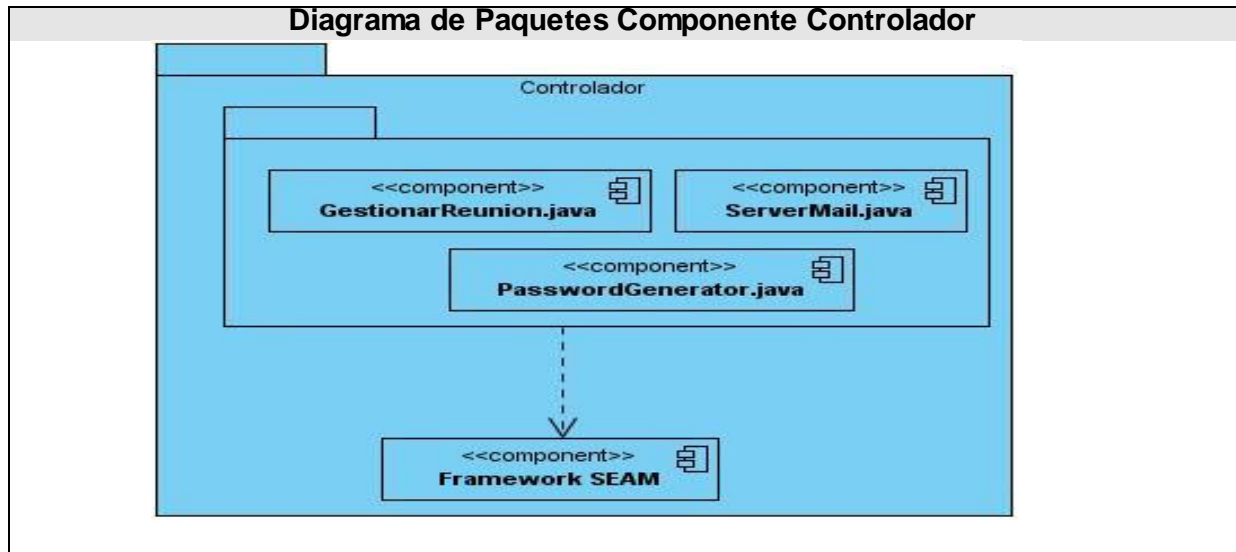


Ilustración 13: Diagrama de Paquetes Componente Controlador

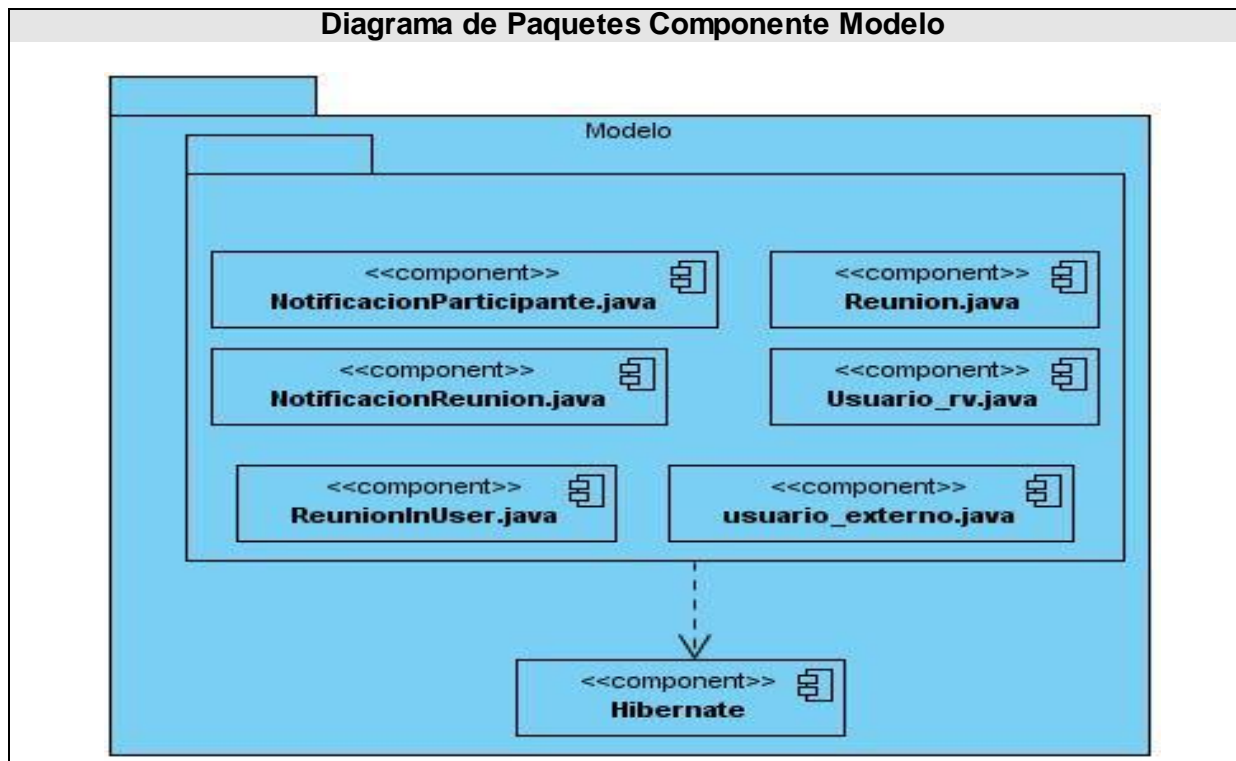


Ilustración 14: Diagrama de Paquetes Componente Modelo

Capítulo 4: Implementación

4.3. Tratamiento de errores.

Un formulario Web bien diseñado debe incluir validaciones de los datos proporcionados por el usuario antes de enviar la información al servidor. Para esto se deben realizar en el cliente tantas comprobaciones como sea posible, así se mejoraría el tiempo de respuesta y se reduciría la carga del servidor.

Para la gestión de reuniones se realizó la validación de todos los campos de entrada al sistema del lado del cliente mediante código JavaScript y del lado del servidor se utilizó la librería Ajax. Al producirse alguna acción incorrecta, se advierte al usuario a través de un mensaje de alerta que no permite que los datos pasen al servidor. En caso de que se necesite el acceso a la base de datos, de producirse un error en la misma, es capturado, especificándosele al usuario la naturaleza del mismo para que conozca en todo momento cual es la fuente del error. Si se desea realizar alguna operación irreversible como la eliminación de algún dato del sistema que se encuentre almacenado en la base de datos, se preguntará al usuario si realmente desea realizar dicha acción mediante un mensaje de confirmación, logrando que se realicen las operaciones que se desean y que se rectifique al cometer un error, aunque la aplicación no permite que se eliminen datos físicamente del servidor de Base de Datos, porque así se definió en el Requisito No Funcional de Fiabilidad.

4.4. Seguridad

La Organización Internacional de Normalización define seguridad:

"Como una subcaracterística de funcionalidad, que indica el grado en que un acceso no autorizado (accidental o deliberado) se prevenga y se permita un acceso autorizado." (41)

En términos informáticos se puede decir que la seguridad informática es un conjunto de métodos y herramientas destinados a proteger los bienes informáticos de una institución y que está estrechamente relacionado con tres aspectos fundamentales de cualquier sistema de computación: la confidencialidad, que radica en que la información o los activos informáticos son accedidos solo por las personas autorizadas; la integridad que reside en que los activos o la información solo pueden ser modificados por las personas autorizadas y de la forma autorizada y por último la disponibilidad que consiste en que los activos informáticos son accedidos por las personas autorizadas en el momento requerido.

Capítulo 4: Implementación

Para analizar la seguridad de un sistema se debe pensar en la forma en que el mismo pudiera sufrir determinada pérdida o daño, para lo cual es necesario identificar las debilidades del sistema. En el caso del sistema en cuestión, la seguridad es llevada a cabo por el proyecto Alas HIS. 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, realizando todo esto a través de un usuario y contraseña. El sistema permite: cerrar sesión y salir del módulo.

Registro de trazas: Cuando el usuario realiza una acción sobre el sistema, que puede ser: inicio o cierre de sesión, acceso al módulo, modificación a un atributo de una entidad o cualquier otra operación sobre el sistema, este registra una traza en la base de datos.

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.

4.5. Estrategias de codificación. Estándares y estilos a utilizar.

Un estándar de codificación es un conjunto de directrices, normas y reglamentos sobre la forma de escribir código. (42) Por lo general, un estándar de codificación incluye pautas sobre cómo nombrar variables, la forma de guión, el código, cómo poner entre paréntesis y palabras clave. La idea es ser constante en la programación de modo que el código sea entendible a cualquier desarrollador del grupo de trabajo y no solo a quien lo creó. Los estándares de codificación garantizan un mantenimiento del sistema más rápido y eficiente, ya sea creando nuevas funcionalidades o modificando las ya existentes.

Para el desarrollo del sistema propuesto se utilizan varios estándares de codificación, tales como:

Capítulo 4: Implementación

Notación Camell: se utiliza para denotar variables, parámetros y métodos. Si el identificador es una palabra simple se escribe todo con minúscula.

Notación C: se utiliza para denotar las variables de las tablas de la base de datos, si la palabra es compuesta se escribe separándola con el guión bajo, ejemplo: id_reunion.

Notación Pascal: se utiliza para denotar los nombres de las clases.

A continuación se muestran algunas excepciones para la nomenclatura basadas en los estándares a utilizar:

Idioma: se debe utilizar como idioma el español, las palabras no se acentuarán.

Indentación:

Las normas de indentación indican la posición en la que se deben colocar los diferentes elementos que se incluyen en el código fuente, por lo que forman parte del estilo de codificación. Su objetivo es lograr una estructura uniforme para los bloques de código, así como para los diferentes niveles de anidamiento.

Inicio y bloque de fin: Se debe 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.

Aspectos generales: El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la PC o la configuración de dicha tecla. Los inicios ({} y cierres (}) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.

Comentarios, separadores, líneas, espacios en blanco y márgenes.

Objetivo: establecer un modo común para comentar el código de forma tal, que sea comprensible con solo leerlo una vez.

Capítulo 4: Implementación

Ubicación de comentarios:

Al inicio de cada clase o función y al final de cada bloque de código. Se recomienda comentar al inicio de la clase o función, especificando el objetivo de la misma, así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.

Líneas en blanco:

Se emplean antes y después de métodos, clases y estructuras. Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

Espacios en blanco:

Entre operadores lógicos y aritméticos. Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código.

Ejemplo de código: `this.minutos = minutos.`

Aspectos generales:

Sobre el 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.

Sobre los 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.

Capítulo 4: Implementación

Variables y constantes:

Apariencia de variables:

Las variables tendrán un prefijo para el tipo de datos en minúscula. El nombre de las variables debe comenzar con la primera letra en minúscula, la cual identificará el tipo de datos al que se refiere, en caso de que sea un nombre compuesto, se empleará notación CamellCasing.

Ejemplo: "sNombreUsuario".

Apariencia de constantes:

Se deben declarar las constantes con todas sus letras en mayúscula.

Aspectos generales:

Nombres de las variables y constantes. El nombre empleado, debe permitir que con solo leer se conozca el propósito de las mismas.

Clases y objetos:

Objetivo: Nombrar las clases e instancias de forma estándar para todas las aplicaciones.

Apariencia de clases y objetos:

Primera letra en mayúscula. 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. Ejemplo: GestionarReunion ().

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 o la notación C.

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. Ejemplo: IniciarReunion (). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set.

Capítulo 4: Implementación

Declaración de parámetro en funciones:

Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal, que estén agrupados por el tipo de dato que contienen, especificando el tipo de datos.

Aspectos generales:

El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con solo leerlo se conozca el propósito de los mismos.

Bases de Datos, Tablas, Esquemas y Campos.

Apariencia de la Base de Datos:

Los nombres de las Bases de Datos deben comenzar con mayúscula y con underscore entre palabras.

Ejemplo: "Nueva_Linea_Base".

Apariencia de las tablas:

Todas las letras en minúscula. El nombre a emplear para las tablas en caso de que sea un nombre compuesto se utilizará underscore para separarlo.

Ejemplo: "notificacion_participante".

Tablas que representen Relaciones:

Todas las letras en minúscula. El nombre a emplear para estas tablas de relación debe comenzar con el nombre de la primera tabla seguido de underscore, luego la palabra "in" y el nombre de la segunda tabla. Ejemplo: "reunion_in_usuario".

Apariencia de los campos:

Todas las letras en minúscula. El nombre a emplear para los campos, debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor y se empleará la notación C. Ejemplo: "id_usuario".

Nombre de los campos:

En caso de identificadores. Todos los campos identificadores van a comenzar con el identificador id seguido de underscore y posteriormente el nombre del campo. Ejemplo: "id_reunion".

Capítulo 4: Implementación

Sentencias SQL:

Todas las letras en mayúscula. Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

Aspectos generales:

El nombre empleado para las Bases de Datos, las vistas, las tablas, los campos y los procedimientos almacenados, deben permitir que con solo leerlos se conozca el propósito de los mismos.

Controles.

Apariencia de los controles:

Los controles tendrán un prefijo para el tipo de datos en minúscula. El nombre que se le da a los controles debe comenzar con las primeras letras en minúscula, las cuales identificarán el tipo de datos al que se refiere, en caso de que sea un nombre compuesto se empleará notación CamellCasing.

Ejemplo: "btnAceptar".

En el capítulo concluido se mostraron los resultados obtenidos durante la etapa de implementación, los cuales permitieron que quedaran conformados los diagramas de componentes, que muestran la distribución de todos los paquetes para su implementación y de despliegue, que describe la estructura de distribución de la aplicación. También, se explicó cómo se realiza el tratamiento de errores, la seguridad llevada a cabo por el sistema y se expusieron las diferentes notaciones que se utilizaron junto con los estándares y estilos para realizar la implementación.

CONCLUSIONES

Como resultado del estudio de la gestión de reuniones virtuales, se determinó que no existe un sistema informático para todas las especialidades médicas, que se adapte a las características del Sistema Nacional de Salud. Por ello, se llevó a cabo el Proceso Unificado de Desarrollo con los primeros tres flujos de trabajo: Modelamiento del negocio, Requerimiento y Análisis y Diseño, lo que permitió construir la arquitectura del sistema a desarrollar, obteniendo así, una solución no funcional del módulo.

A partir de la solución no funcional obtenida, se realizó el flujo de trabajo Implementación, apoyado por el estudio de las técnicas de programación, tecnologías, herramientas y pautas de diseño definidas por el Departamento de Gestión Hospitalaria para el desarrollo de sus aplicaciones, en consecuencia, se obtuvo el Módulo de Gestión de Reuniones del Sistema Teleconsulta, el cual constituye la solución funcional.

RECOMENDACIONES

Para lograr la continuidad de este trabajo, los autores del mismo recomiendan:

- Posibilitar al usuario la opción de responder la notificación al moderador, para que este confirme su presencia en la reunión.
- Permitir al usuario eliminar varias notificaciones a la vez, para facilitarle la interacción con el sistema.
- Listar las reuniones terminadas por el moderador, para que este tenga constancia de sus reuniones realizadas.

REFERENCIAS BIBLIOGRÁFICAS

1. **McGraw-Hill, Editorial.** Introducción al mundo de la Informática. [En línea] 22 de Noviembre de 2010. <http://www.mailxmail.com/curso-introduccion-mundo-informatica/tratamiento-automatico-informacion>.
2. **Kohen, Dr. Alejandro Andrés.** TICs. [En línea] 23 de Noviembre de 2010. http://tics.org.ar/index.php?option=com_content&task=view&id=13&Itemid=28.
3. **Rodriguez, Jorge Martinez.** Programa Social de la Información. [En línea] [Citado el: 11 de Noviembre de 2010.] <http://www.eclac.cl/socinfo/noticias/paginas/6/35876/Cuba.pdf>.
4. **Infomed.sld.cu.** Telemedicina. [En línea] [Citado el: 19 de Octubre de 2010.] <http://www.sld.cu/telemedicina/indice.html>.
5. **System, PrimeCare.** Qué es Telemedicina. [En línea] [Citado el: 15 de Octubre de 2010.] <http://softcare-argentina.com/noticia.php?nId=9>.
6. Voxnet. [En línea] [Citado el: 3 de Octubre de 2010.] <http://www.voxnet.it/home.cfm?ID=1047&ID2=t&espandi=1020>.
7. **Enrique, Javier.** Buenas Tareas. [En línea] [Citado el: 26 de Noviembre de 2010.] <http://www.buenastareas.com/ensayos/Redes-Sociales/263169.html>.
8. **Systems, Adobe.** Adobe-Connect. [En línea] [Citado el: 20 de octubre de 2010.] <http://www.adobe.com/es/products/connect/>.
9. **Skype.** Skype. [En línea] [Citado el: 15 de Octubre de 2010.] <http://www.skype.com/>.
10. Openmeetings. [En línea] [Citado el: 9 de octubre de 2010.] <http://www.openmeetings.de/openmeetings/>.
11. **Systems, Cisco.** Webex. [En línea] [Citado el: 26 de Octubre de 2010.] <http://www.webex.es/espana/product-overview/meeting-center.html>.
12. **Medina, Dr. Eduardo Fernández.** Metodologías de desarrollo 2007. [En línea] [Citado el: 15 de Noviembre de 2010.] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
13. **HRCS, (recursos humanos servicios de consultoría).** EJB 3.0 - JPA. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://www.maperipe.es/cursoejb3/HRCS-EJB3.0-JPA.pdf>.
14. definición.org. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://www.definicion.org/lenguaje-de-programacion>.
15. java. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://java.com/es/about/>.
16. Introduction to OMG's Unified Modeling Language . [En línea] [Citado el: 15 de Diciembre de 2010.] http://www.omg.org/gettingstarted/what_is_uml.htm.
17. JBoss Application Server – Dos Ideas. [En línea] [Citado el: 4 de Octubre de 2010.] http://www.dosideas.com/wiki/JBoss_Application_Server.

Referencias Bibliográficas

18. CAVSI. [En línea] [Citado el: 16 de Diciembre de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-ogdb/>.
19. SCRIBD. [En línea] [Citado el: 16 de Diciembre de 2010.] <http://www.scribd.com/doc/36570462/postgreSQL-investigacion>.
20. Herramientas CASE. [En línea] [Citado el: 16 de Diciembre de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
21. Visual Paradigm. [En línea] [Citado el: 18 de Diciembre de 2010.] <http://www.visual-paradigm.com/news/vpsuite33/vpuml63.jsp>.
22. Definiciones-Es. [En línea] [Citado el: 16 de Diciembre de 2010.] <http://www.definiciones.com/entorno-de-desarrollo-integrado/1/>.
23. Manzana Mecánica. *Eclipse Ganymede*. [En línea] [Citado el: 15 de Noviembre de 2010.] http://manzanamecanica.org/2008/07/eclipse_ganymede.html.
24. **Heng, Christopher**. Thefreecountry.com. [En línea] [Citado el: 17 de Diciembre de 2010.] <http://www.thefreecountry.com/programming/javaide.shtml>.
25. **García, Javier y Rodríguez, José Ignacio**. *Aprenda Java como si estuviera en primero*. Universidad de Navarra : San Sebastián, 2000.
26. Tecnología y Synergix. [En línea] [Citado el: 11 de Enero de 2010.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
27. Tecnología y Synergix. [En línea] [Citado el: 25 de Enero de 2011.] <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.
28. La Norma Iso lec 9126. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://www.mitecnologico.com/Main/LaNormaIsolec9126>.
29. Itákora. [En línea] [Citado el: 10 de Diciembre de 2010.] <http://itakora.com/medir-la-eficiencia/>.
30. Tripod.com. [En línea] [Citado el: 10 de Febrero de 2011.] <http://oggp.tripod.com/teoria2/fiabilidad1.html>.
31. Especificaciones De Requerimientos. [En línea] [Citado el: 10 de Febrero de 2011.] <http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos>.
32. ftapies. [En línea] [Citado el: 11 de Febrero de 2011.] web.ftapies.com/thesaurus/spip.php.
33. Concepto: Modelo de Caso de Uso. [En línea] [Citado el: 11 de Febrero de 2011.] http://epf.eclipse.org/wikis/openupsp/openup_basic/guidances/concepts/use_case_model,_2jyfUAhVEduRe8TeoBmuGg.html.
34. clase7. [En línea] [Citado el: 1 de Febrero de 2011.] <http://ldc.usb.ve/~mgoncalves/IS2/sd07/clase7.pdf>.
35. **Maldonado, Daniel**. El CoDiGo K(Arquitectura de Programación en 3 Capas). [En línea] [Citado el: 8 de Febrero de 2011.] <http://www.elcodigok.com.ar/2007/09/arquitectura-de-programacion-en-3-capas/>.
36. Slideshare. *Arquitectura 3 Capas*. [En línea] [Citado el: 10 de Febrero de 2011.] <http://www.slideshare.net/Decimo/arquitectura-3-capas>.

Referencias Bibliográficas

37. slideshare. *Clase Flujo De Analisis*. [En línea] [Citado el: 17 de Febrero de 2011.] <http://www.slideshare.net/juliopari/13-clase-flujo-de-analisis>.
38. Scribd. *Diagramas de Colaboración*. [En línea] [Citado el: 18 de Febrero de 2011.] <http://www.scribd.com/doc/11802367/diagramas-de-colaboracion>.
39. Clikear.com. *Manual*. [En línea] [Citado el: 18 de Febrero de 2011.] <http://www.clikear.com/manuales/uml/diagramasinteraccion.aspx>.
40. Mundo Geek. *Modelo de Datos*. [En línea] [Citado el: 23 de Marzo de 2011.] <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos/>.
41. CyTA. [En línea] [Citado el: 3 de Abril de 2011.] http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/glosario_9126/glosario_9126.htm.
42. Compute-rs.com. [En línea] [Citado el: 5 de Abril de 2011.] <http://www.compute-rs.com/es/consejos-54651.htm>.

BIBLIOGRAFÍA

1. @saim. [En línea] [Citado el: 15 de Octubre de 2010.] [http://saim.org.ar/.](http://saim.org.ar/)
2. **Aguilar, José Ramón.** El Portal de la Medicina de Emergencia. [En línea] [Citado el: 18 de Octubre de 2010.] <http://www.medynet.com/usuarios/jraguilar/portal%20joserra23.html..>
3. CAVSI. [En línea] [Citado el: 16 de Diciembre de 2010.] [http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/.](http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/)
4. clase7. [En línea] [Citado el: 1 de Febrero de 2011.] [http://ldc.usb.ve/~mgoncalves/IS2/sd07/clase7.pdf.](http://ldc.usb.ve/~mgoncalves/IS2/sd07/clase7.pdf)
5. Clikear.com. *Manual.* [En línea] [Citado el: 18 de Febrero de 2011.] [http://www.clikear.com/manuales/uml/diagramasinteraccion.aspx.](http://www.clikear.com/manuales/uml/diagramasinteraccion.aspx)
6. Compute-rs.com. [En línea] [Citado el: 5 de Abril de 2011.] [http://www.compute-rs.com/es/consejos-54651.htm.](http://www.compute-rs.com/es/consejos-54651.htm)
7. Concepto: Modelo de Caso de Uso. [En línea] [Citado el: 11 de Febrero de 2011.] [http://epf.eclipse.org/wikis/openupsp/openup_basic/guidances/concepts/use_case_model_2jyfUAhVEduRe8TeoBmuGg.html.](http://epf.eclipse.org/wikis/openupsp/openup_basic/guidances/concepts/use_case_model_2jyfUAhVEduRe8TeoBmuGg.html)
8. CyTA. [En línea] [Citado el: 3 de Abril de 2011.] [http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/glosario_9126/glosario_9126.htm.](http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/glosario_9126/glosario_9126.htm)
9. definición.org. [En línea] [Citado el: 15 de Diciembre de 2010.] [http://www.definicion.org/lenguaje-de-programacion.](http://www.definicion.org/lenguaje-de-programacion)
10. Definiciones-Es. [En línea] [Citado el: 16 de Diciembre de 2010.] [http://www.definiciones.com/entorno-de-desarrollo-integrado/1/.](http://www.definiciones.com/entorno-de-desarrollo-integrado/1/)
11. **Enrique, Javier.** Buenas Tareas. [En línea] [Citado el: 26 de Noviembre de 2010.] [http://www.buenastareas.com/ensayos/Redes-Sociales/263169.html.](http://www.buenastareas.com/ensayos/Redes-Sociales/263169.html)
12. Especificaciones De Requerimientos. [En línea] [Citado el: 10 de Febrero de 2011.] [http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos.](http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos)
13. ftapies. [En línea] [Citado el: 11 de Febrero de 2011.] [web.ftapies.com/thesaurus/spip.php.](http://web.ftapies.com/thesaurus/spip.php)
14. **García, Javier y Rodríguez, José Ignacio.** *Aprenda Java como si estuviera en primero.* Universidad de Navarra : San Sebastián, 2000.
15. Haya. *SEMES Capitan. Semes.* [En línea] [Citado el: 19 de Octubre de 2010.] [http://www.semes.org/..](http://www.semes.org/)
16. **Heng, Christopher.** Thefreecountry.com. [En línea] [Citado el: 17 de Diciembre de 2010.] [http://www.thefreecountry.com/programming/javaide.shtml.](http://www.thefreecountry.com/programming/javaide.shtml)

17. Herramientas CASE. [En línea] [Citado el: 16 de Diciembre de 2010.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>.
18. **HRCS, (recursos humanos servicios de consultoría).** EJB 3.0 - JPA. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://www.maperipe.es/coursejb3/HRCS-EJB3.0-JPA.pdf>.
19. IDG. COMPUTERWORLD COMMUNICATIONS. [En línea] [Citado el: 10 de Octubre de 2010.] <http://www.idg.es/computerworld/Hospitales-comunicantes.Sistema-de-Teleconsulta-Me/seccion-ana/articulo-63085..>
20. **Infomed.sld.cu.** Telemedicina. [En línea] [Citado el: 19 de Octubre de 2010.] <http://www.sld.cu/telemedicina/indice.html..>
21. Introduction to OMG's Unified Modeling Language . [En línea] [Citado el: 15 de Diciembre de 2010.] http://www.omg.org/gettingstarted/what_is_uml.htm.
22. Itákora. [En línea] [Citado el: 10 de Diciembre de 2010.] <http://itakora.com/medir-la-eficiencia/>.
23. Java Tutorials and Development. [En línea] [Citado el: 20 de Noviembre de 2010.]
24. java. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://java.com/es/about/>.
25. JBoss Application Server – Dos Ideas. [En línea] [Citado el: 4 de Octubre de 2010.] http://www.dosideas.com/wiki/JBoss_Application_Server.
26. **Kohen, Dr. Alejandro Andrés.** TICs. [En línea] 23 de Noviembre de 2010. http://tics.org.ar/index.php?option=com_content&task=view&id=13&Itemid=28.
27. La Norma Iso lec 9126. [En línea] [Citado el: 15 de Diciembre de 2010.] <http://www.mitecnologico.com/Main/LaNormaIsolec9126>.
28. La. LBD Banda. [En línea] [Citado el: 15 de Octubre de 2010.] [http://www.labandadiario.com/celebraron-los-25-anos-de-historia-de-asaim/..](http://www.labandadiario.com/celebraron-los-25-anos-de-historia-de-asaim/)
29. **Lagos, Prof. Martín.** Hospital Clínico San Carlos. *Telemedicina en España.* [En línea] [Citado el: 10 de Octubre de 2010.] <http://www.revistareduca.es/index.php/reduca/article/viewFile/28/29..>
30. **Maldonado, Daniel.** El CoDiGo K(Arquitectura de Programación en 3 Capas). [En línea] [Citado el: 8 de Febrero de 2011.] <http://www.elcodigok.com.ar/2007/09/arquitectura-de-programacion-en-3-capas/>.
31. Manzana Mecánica. *Eclipse Ganymede.* [En línea] [Citado el: 15 de Noviembre de 2010.] http://manzanamecanica.org/2008/07/eclipse_ganymede.html.
32. **McGraw-Hill, Editorial.** Introducción al mundo de la Informática. [En línea] 22 de Noviembre de 2010. <http://www.mailxmail.com/curso-introduccion-mundo-informatica/tratamiento-automatico-informacion>.
33. **Medina, Dr. Eduardo Fernández.** Metodologías de desarrollo 2007. [En línea] [Citado el: 15 de Noviembre de 2010.] <http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema04.pdf>.
34. Mundo Geek. *Modelo de Datos.* [En línea] [Citado el: 23 de Marzo de 2011.] <http://mundogeek.net/archivos/2004/08/26/modelo-de-datos/>.

35. Nuevo diario Web. [En línea] [Citado el: 17 de Octubre de 2010.] http://www.nuevodiarioweb.com.ar/nota/107662/Banda/Semana_Educacion_Especial_Asim.html.
36. Openmeetings. [En línea] [Citado el: 9 de octubre de 2010.] <http://www.openmeetings.de/openmeetings/>.
37. Oracle. [En línea] [Citado el: 10 de Noviembre de 2010.] <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>.
38. **Rodríguez, Jorge Martínez**. Programa Social de la Información. [En línea] [Citado el: 11 de Noviembre de 2010.] <http://www.eclac.cl/sociinfo/noticias/paginas/6/35876/Cuba.pdf>.
39. SCRIBD. [En línea] [Citado el: 16 de Diciembre de 2010.] <http://www.scribd.com/doc/36570462/postgreSQL-investigacion>.
40. Scribd. *Diagramas de Colaboración*. [En línea] [Citado el: 18 de Febrero de 2011.] <http://www.scribd.com/doc/11802367/diagramas-de-colaboracion>.
41. **Skype**. Skype. [En línea] [Citado el: 15 de Octubre de 2010.] <http://www.skype.com/>.
42. Slideshare. *Arquitectura 3 Capas*. [En línea] [Citado el: 10 de Febrero de 2011.] <http://www.slideshare.net/Decimo/arquitectura-3-capas>.
43. slideshare. *Clase Flujo De Analisis*. [En línea] [Citado el: 17 de Febrero de 2011.] <http://www.slideshare.net/juliopari/13-clase-flujo-de-analisis>.
44. **Suárez, Jose Manuel Sánchez**. Adictos al trabajo. [En línea] [Citado el: 15 de Noviembre de 2010.] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=richFacesJsfIntro..>
45. **System, PrimeCare**. Qué es Telemedicina. [En línea] [Citado el: 15 de Octubre de 2010.] <http://softcare-argentina.com/noticia.php?nld=9>.
46. **Systems, Adobe**. Adobe-Connect. [En línea] [Citado el: 20 de octubre de 2010.] <http://www.adobe.com/es/products/connect/>.
47. **Systems, Cisco**. Webex. [En línea] [Citado el: 26 de Octubre de 2010.] <http://www.webex.es/espana/product-overview/meeting-center.html>.
48. Tecnología y Synergix. [En línea] [Citado el: 11 de Enero de 2010.] <http://synergix.wordpress.com/2008/07/10/modelo-de-dominio/>.
49. Tecnología y Synergix. [En línea] [Citado el: 25 de Enero de 2011.] <http://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.
50. **Tolmo, Debora**. *Trabajo para optar por el título de Ingeniero en Ciencias Informáticas: Diseño e Implementación del Sistema de Teleconsulta*. Ciudad Habana, 2009.
51. Tripod.com. [En línea] [Citado el: 10 de Febrero de 2011.] <http://oggp.tripod.com/teoria2/fiabilidad1.html>.
52. Visual Paradigm. [En línea] [Citado el: 18 de Diciembre de 2010.] <http://www.visual-paradigm.com/news/vpsuite33/vpuml63.jsp>.
53. Voxnet. [En línea] [Citado el: 3 de Octubre de 2010.] <http://www.voxnet.it/home.cfm?ID=1047&ID2=t&expandi=1020>.

ANEXOS:

Anexo 1. Diagrama de Clases de Análisis Modificar Reunión

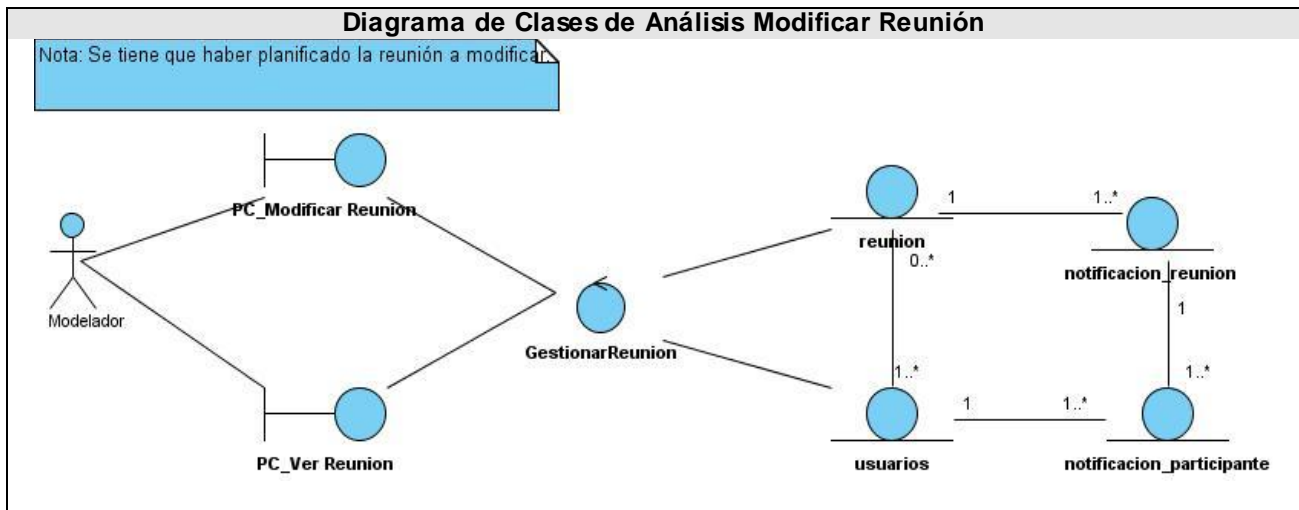


Diagrama de Clases de Análisis Iniciar Reunión

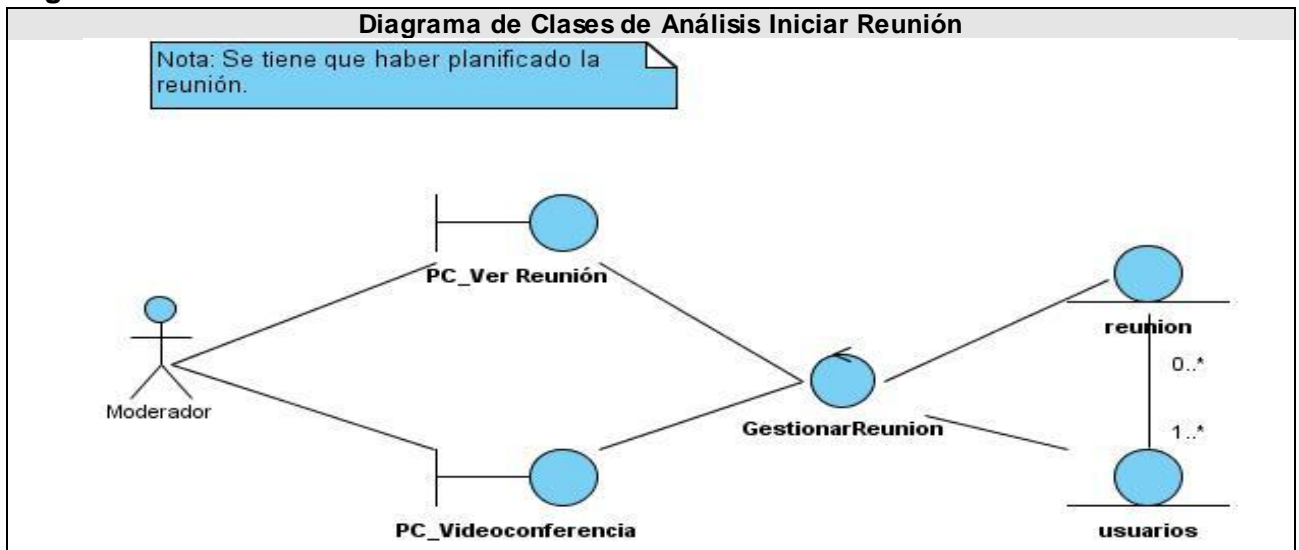


Diagrama de Clases de Análisis Entrar Reunión

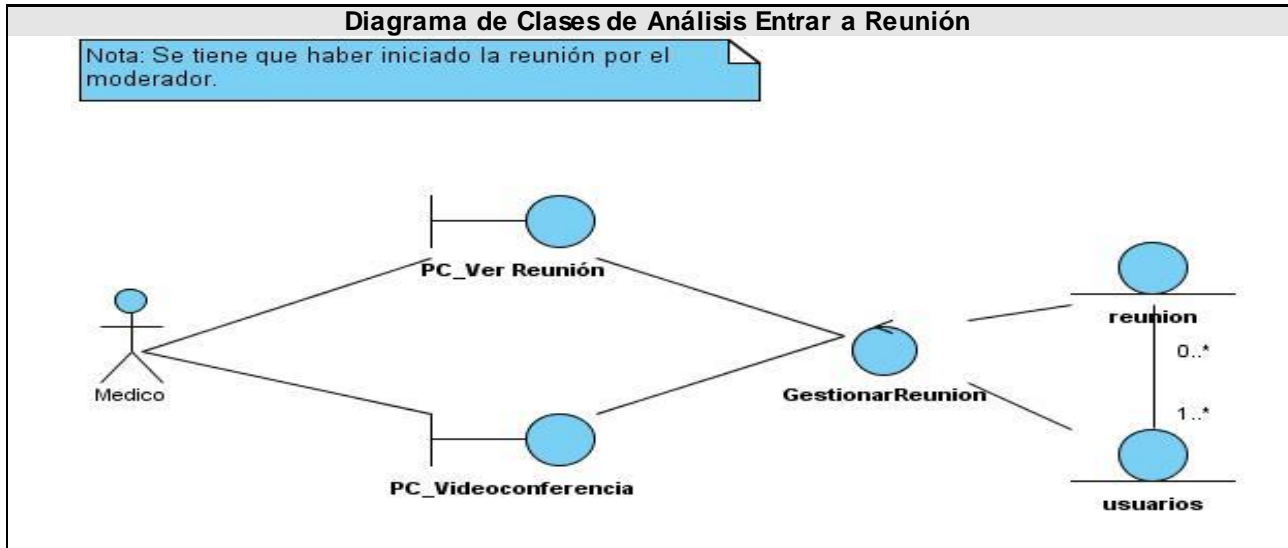
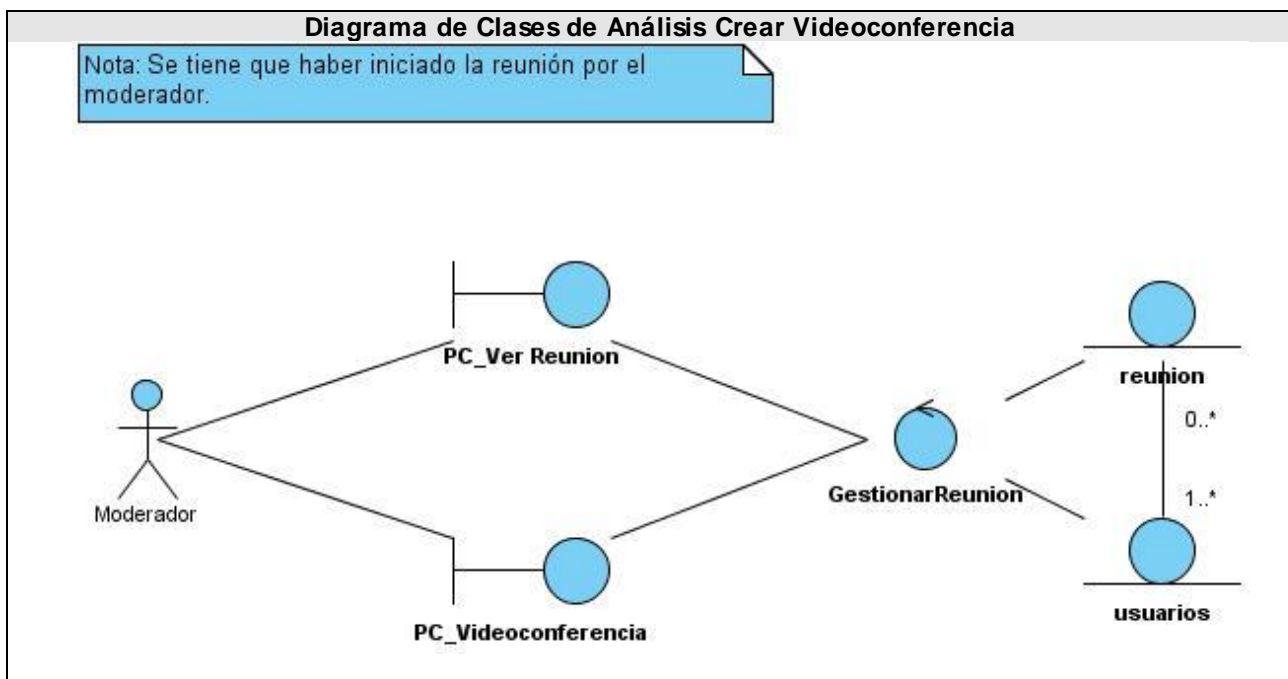


Diagrama de Clases de Análisis Crear Videoconferencia



Anexo 2. Diagrama de Secuencia Planificar Reunión

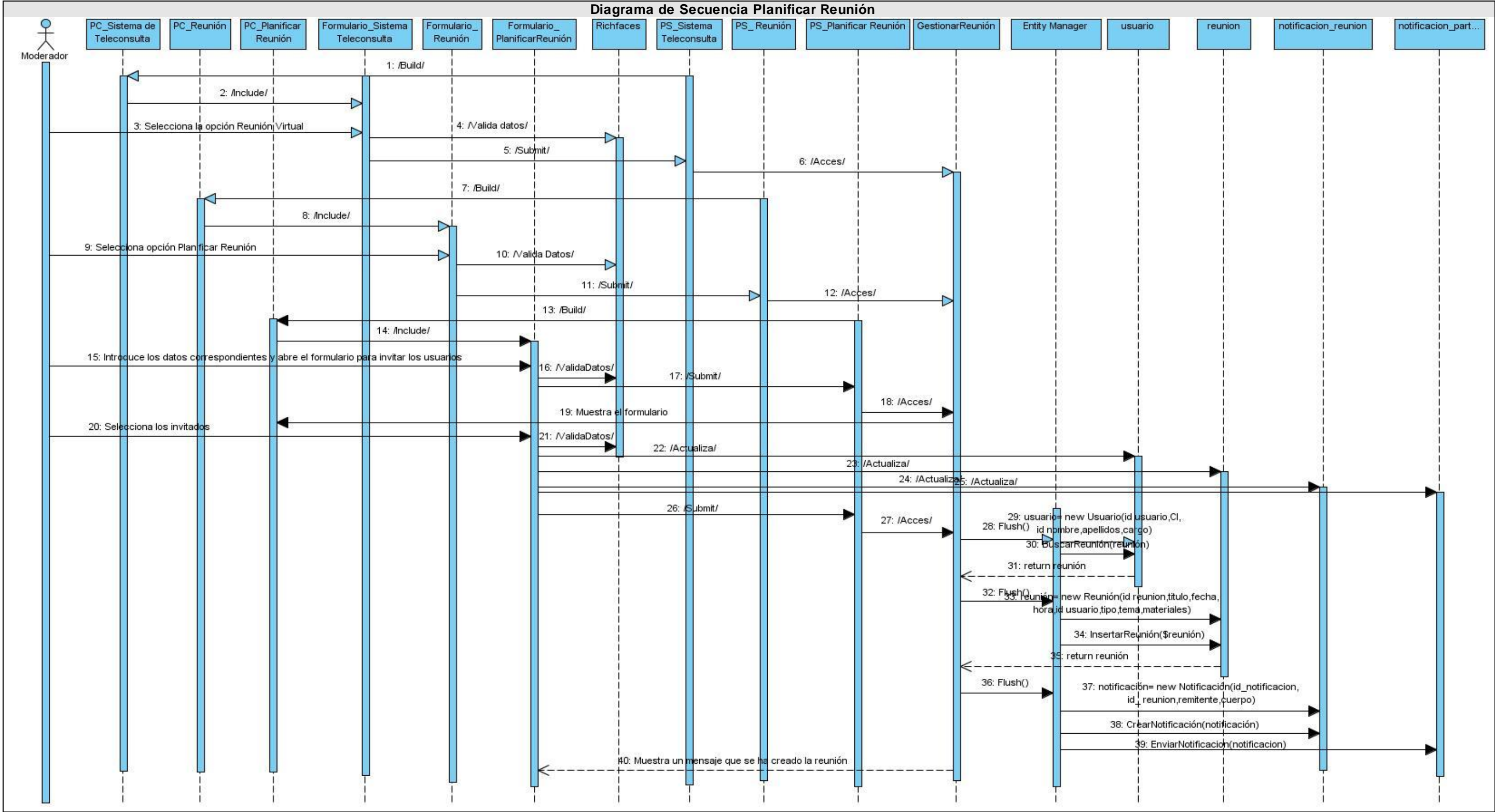


Diagrama de Secuencia Modificar Reunión

Diagrama de Secuencia Modificar Reunión

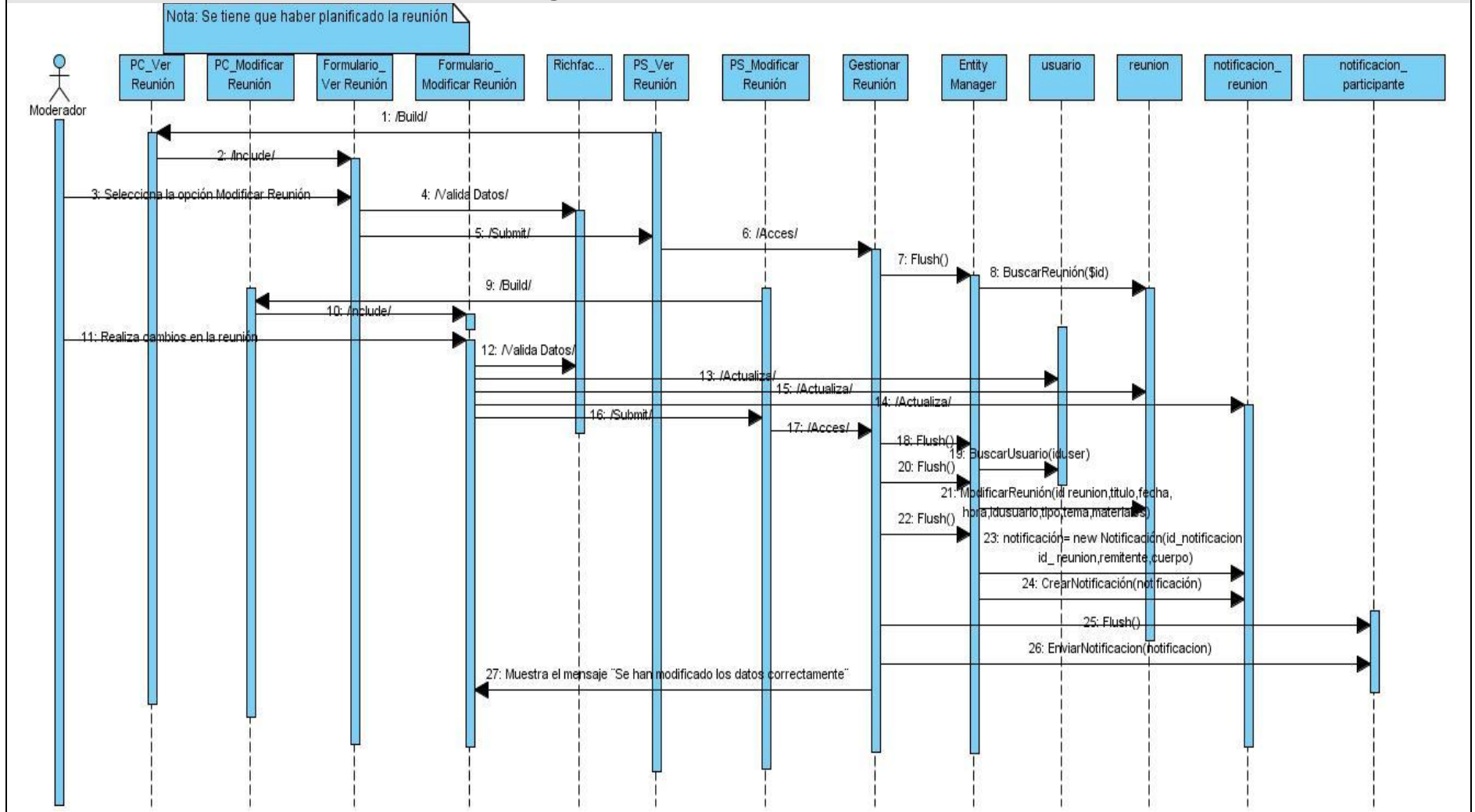


Diagrama de Secuencia Suspender Reunión

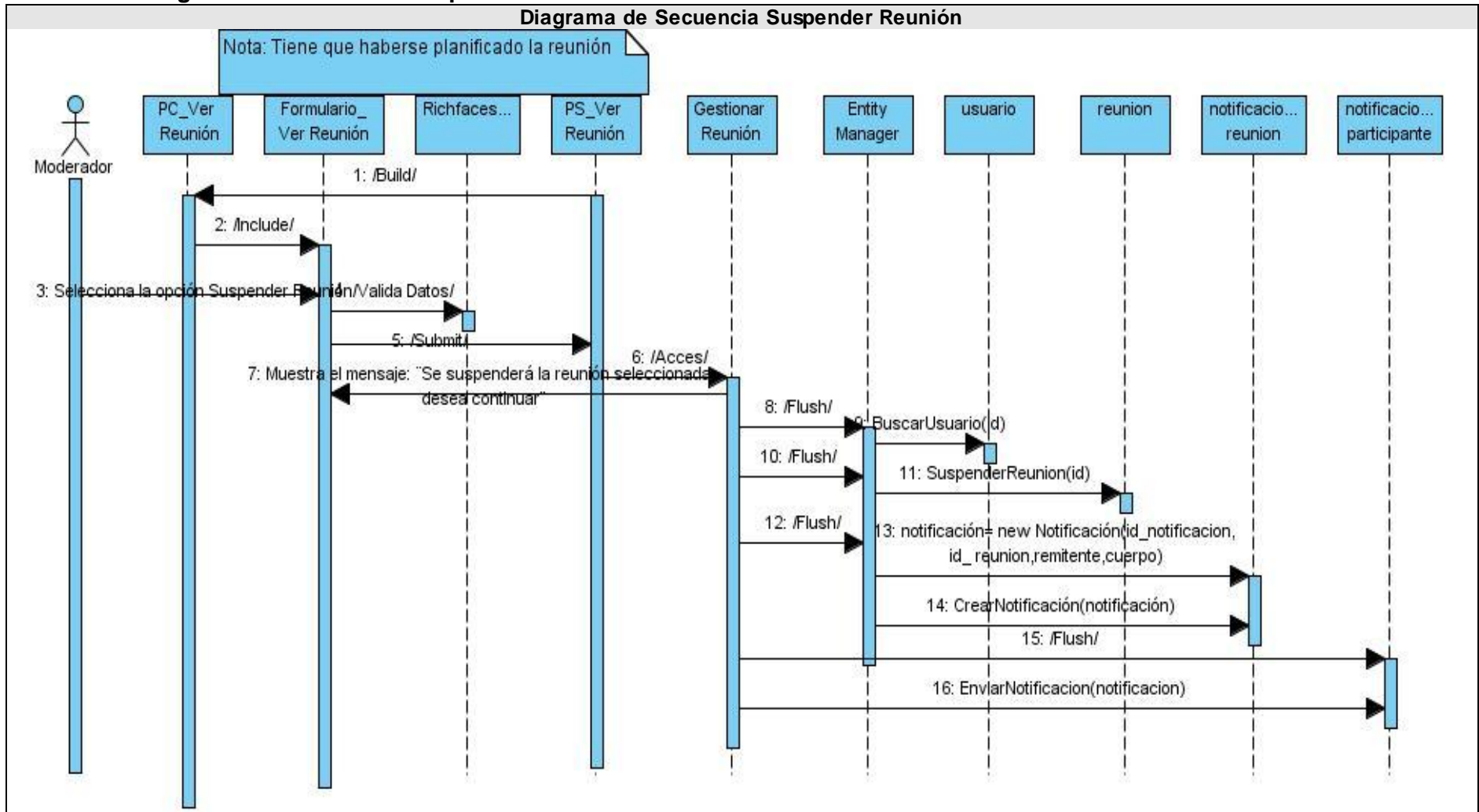


Diagrama de Secuencia Ver Reunión

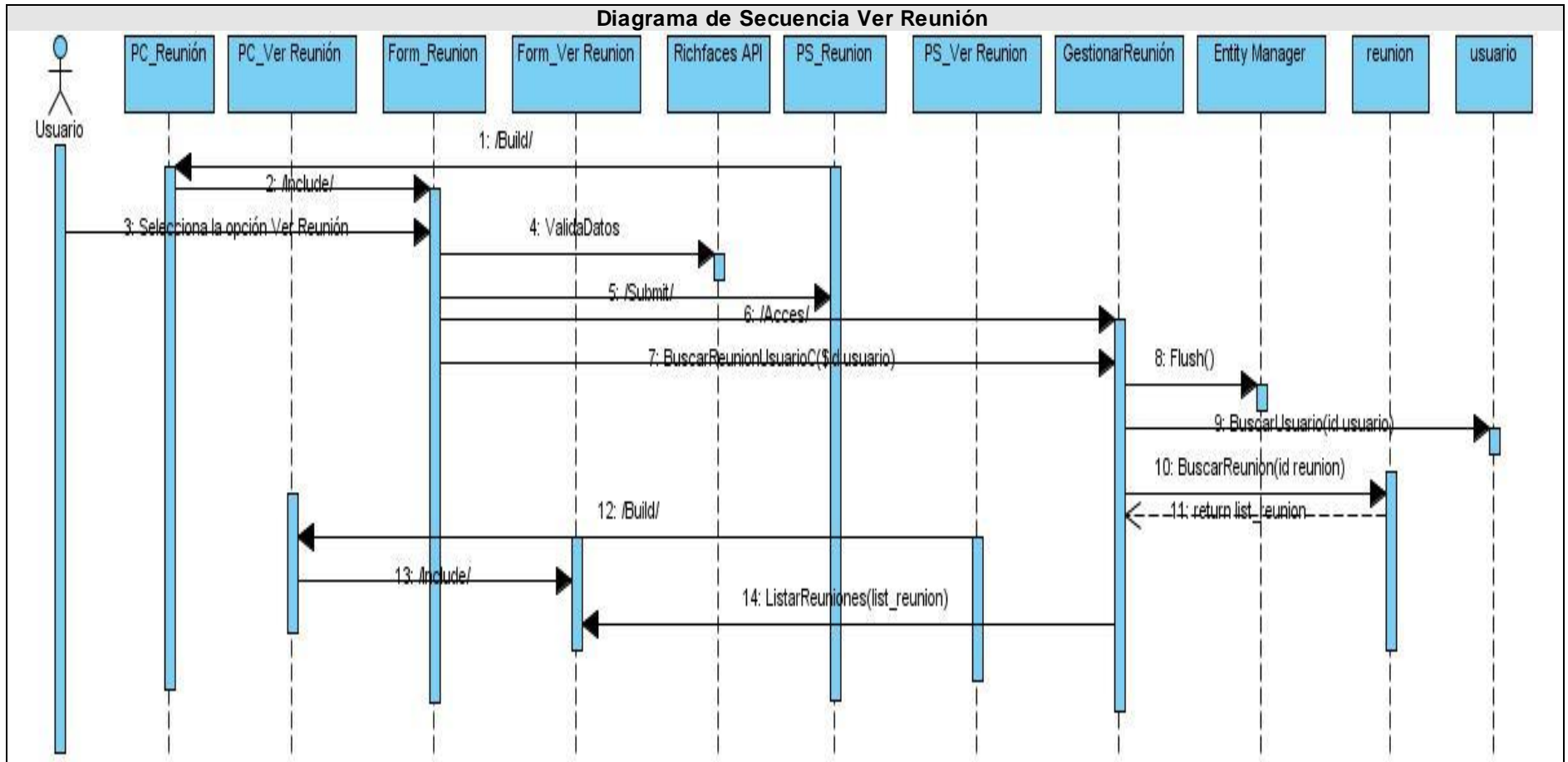


Diagrama de Secuencia Iniciar Reunión.

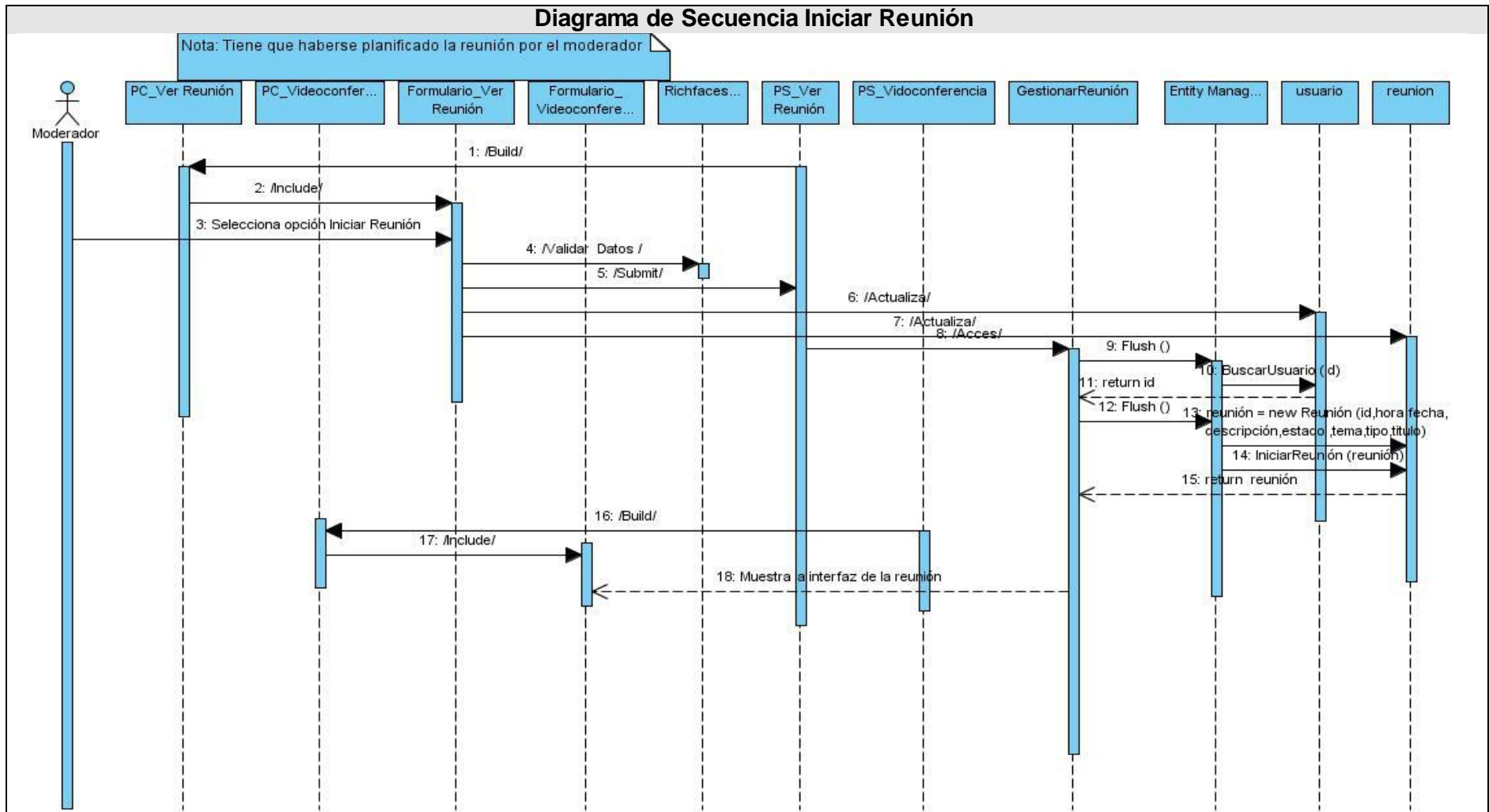


Diagrama de Secuencia Entrar a Reunión.

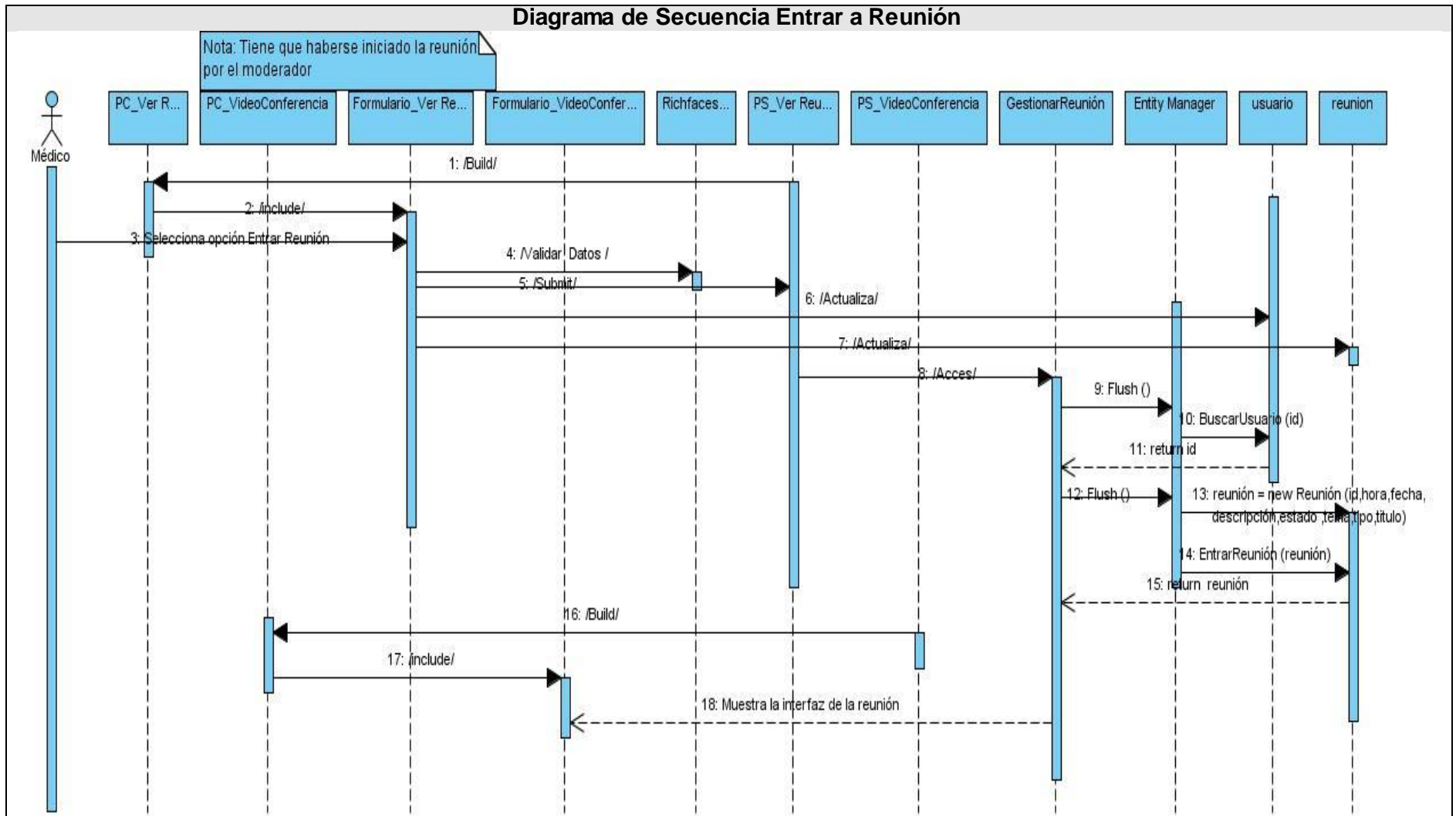


Diagrama de Secuencia Crear Audioconferencia.

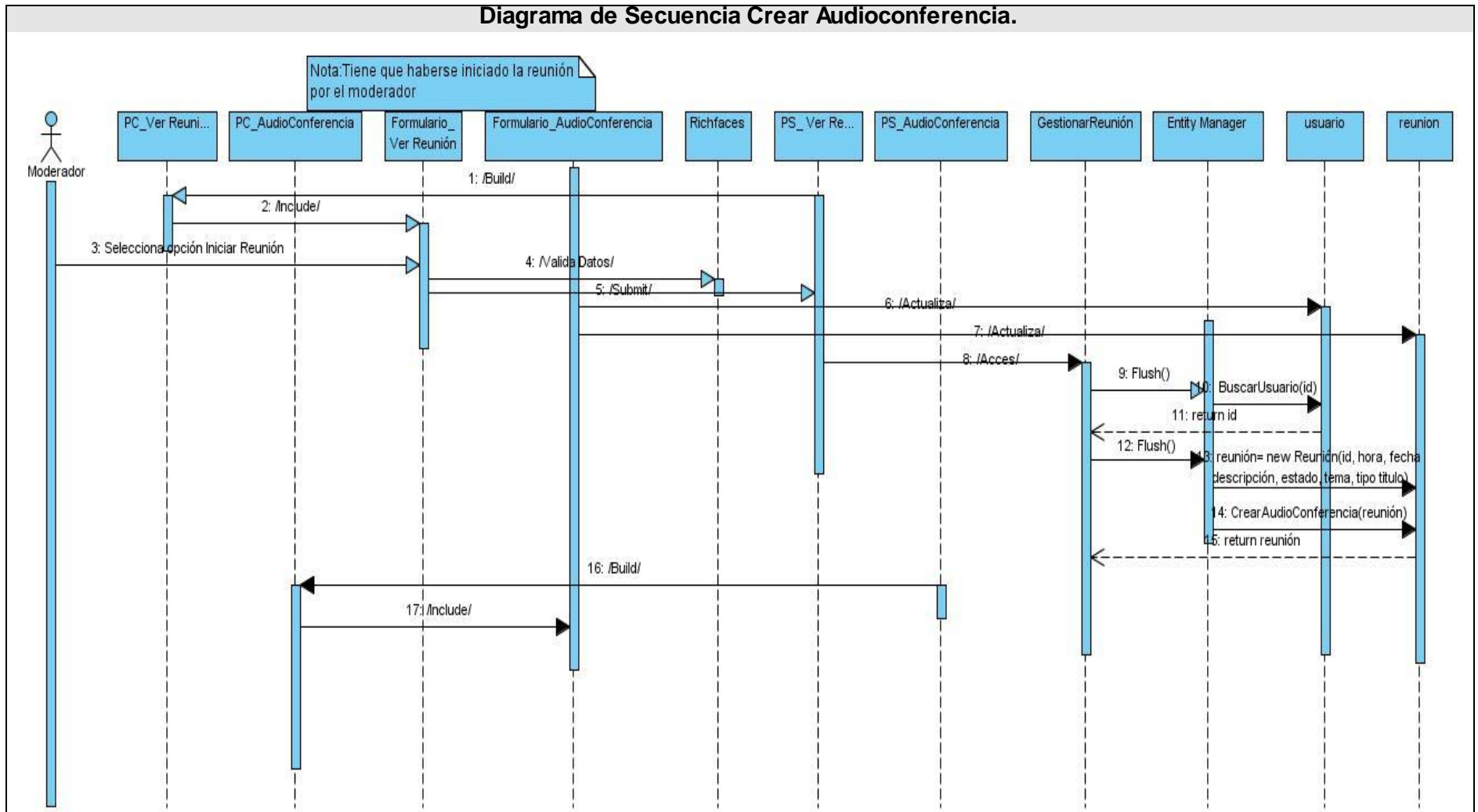
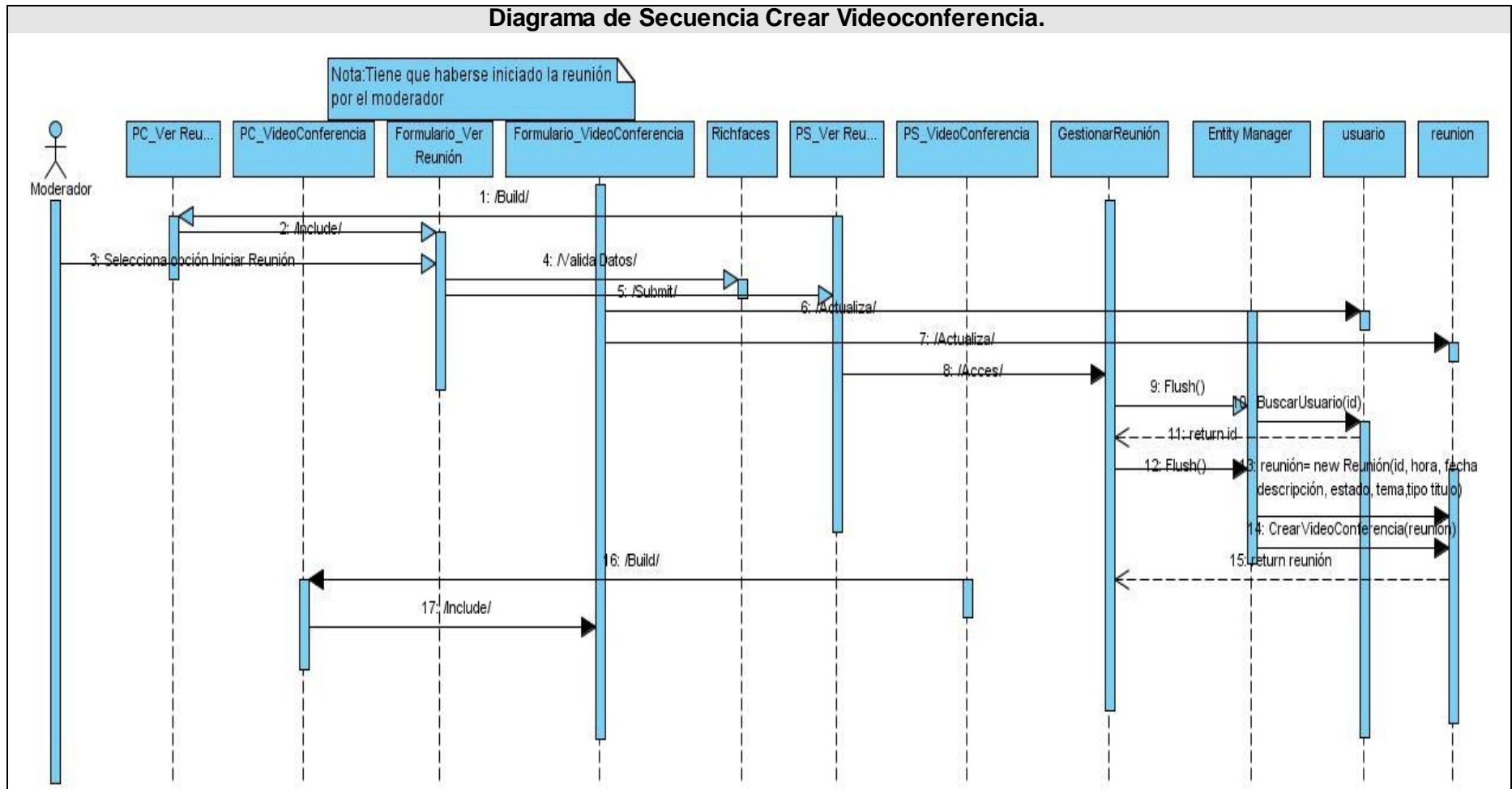


Diagrama de Secuencia Crear Videoconferencia.



Anexo 3. Diagrama de Clases del Diseño Modificar Reunión.

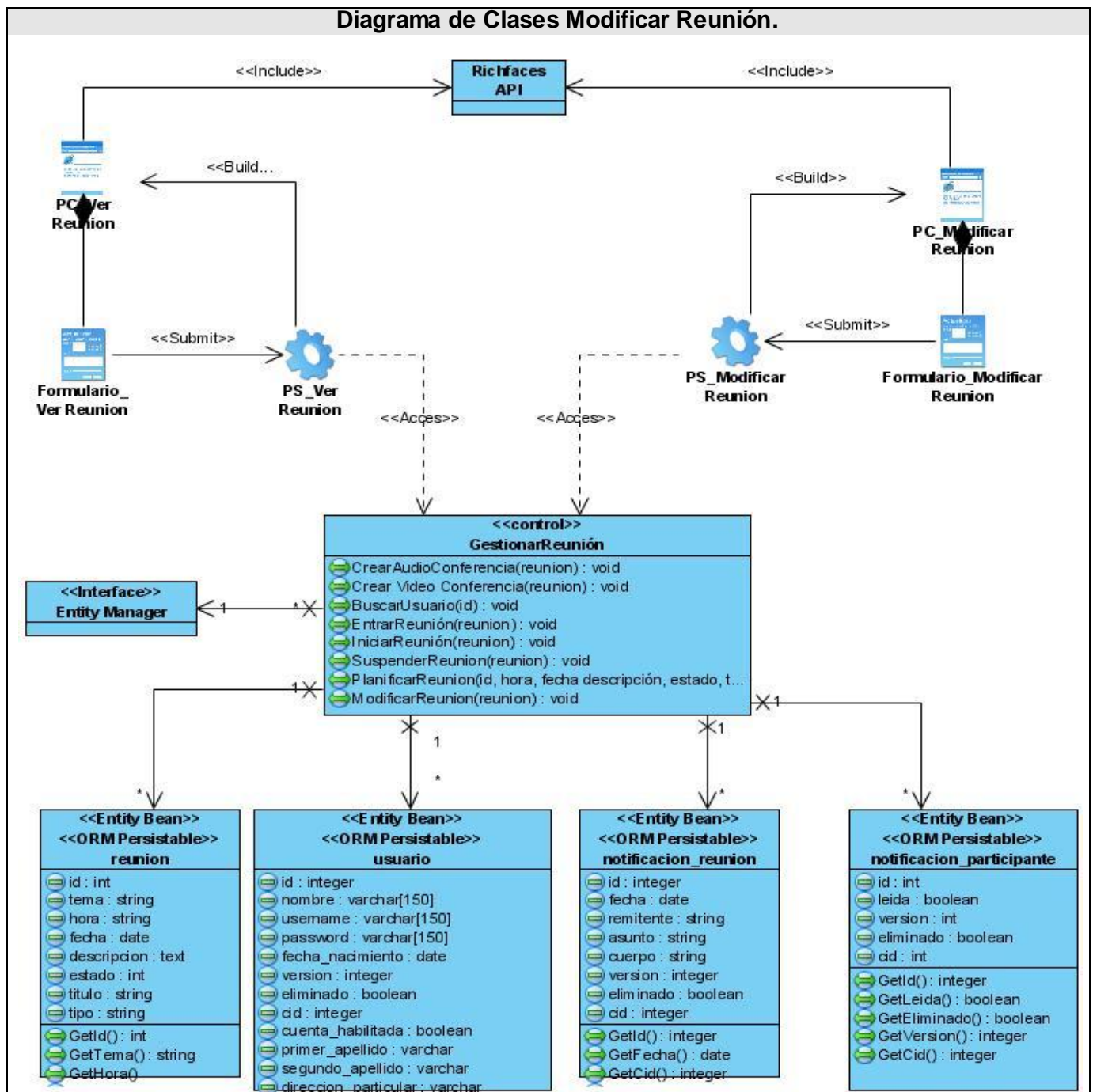


Diagrama de Clases del Diseño Iniciar Reunión.

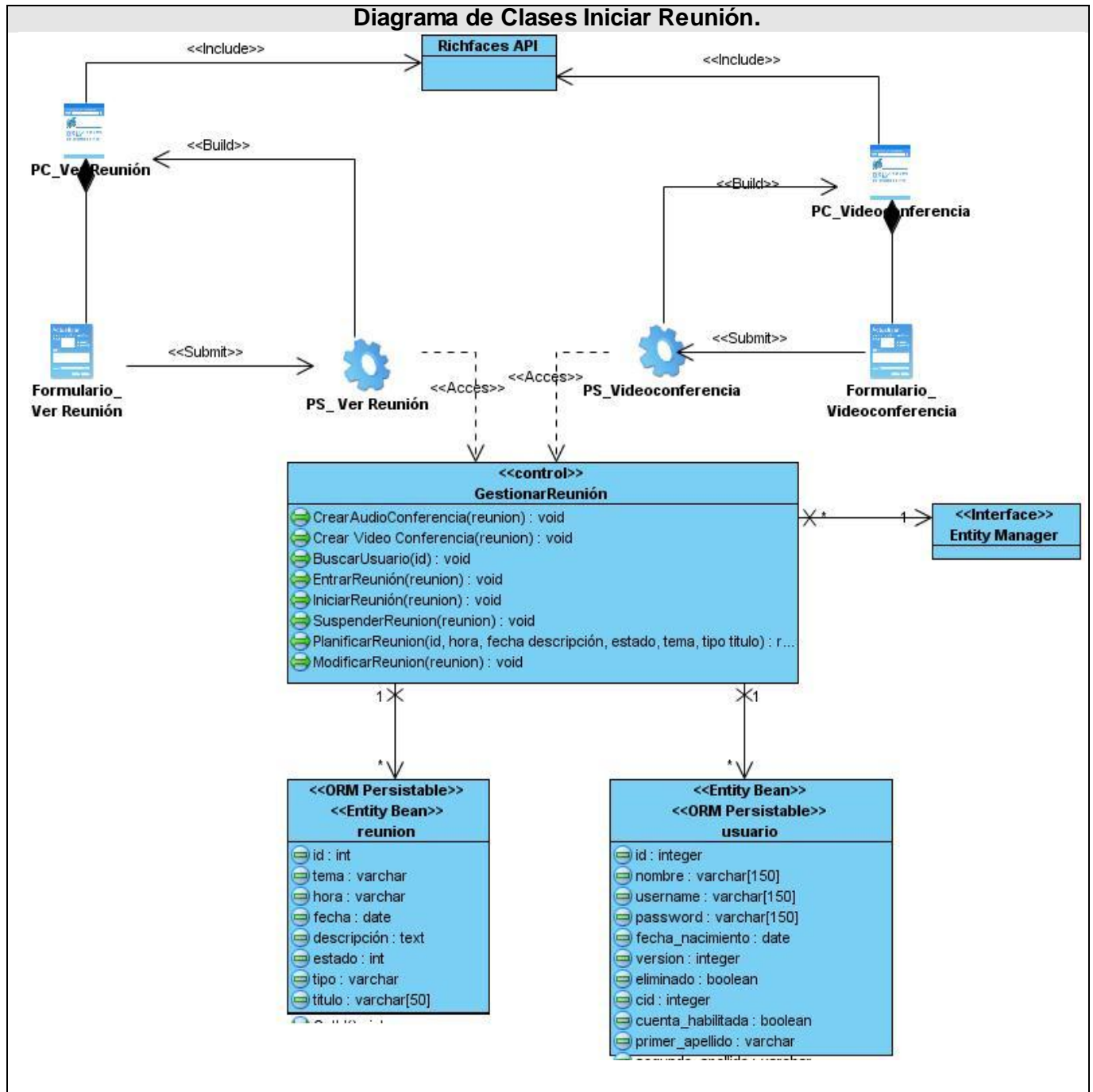


Diagrama de Clases del Diseño Entrar a Reunión.

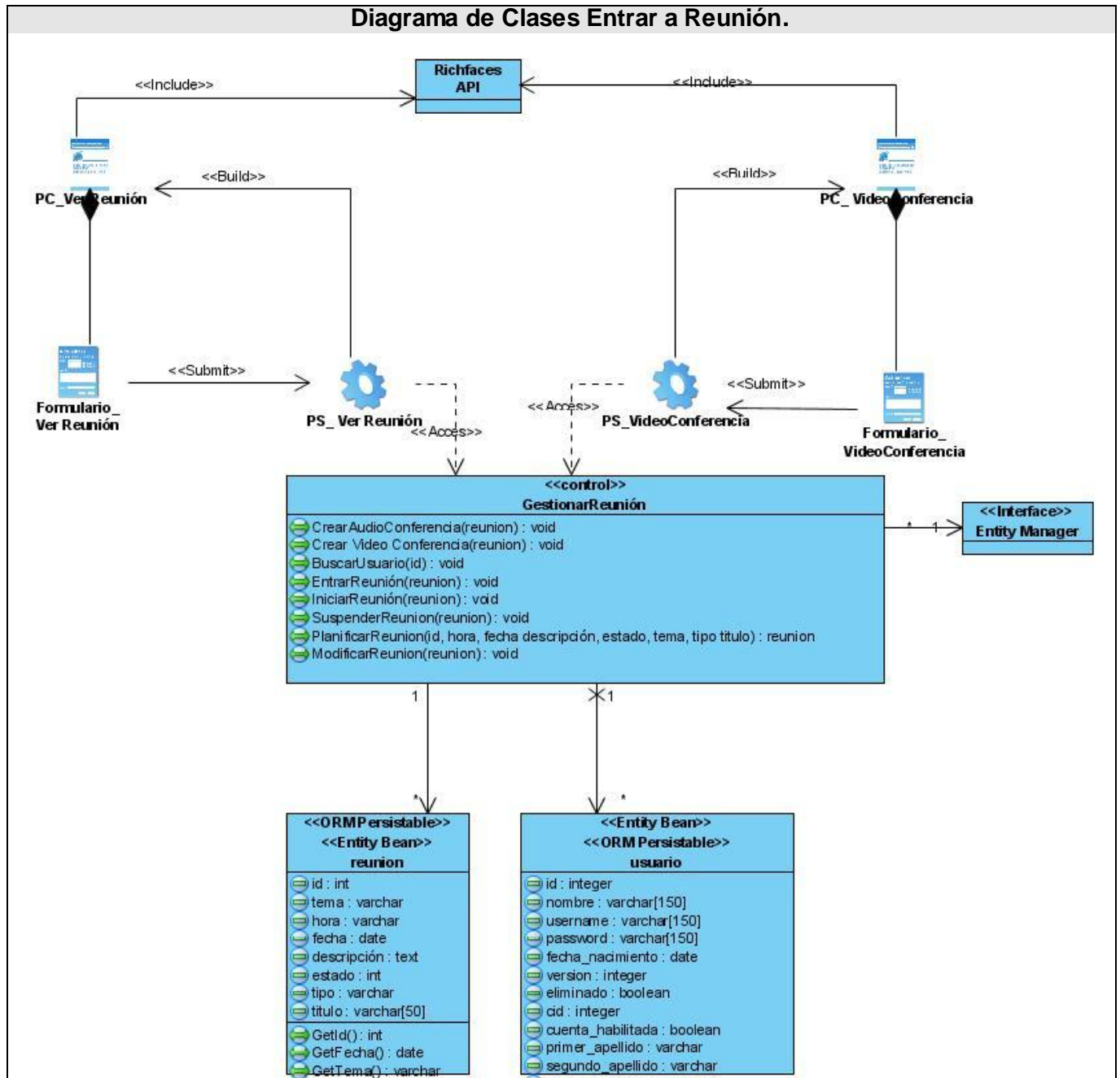


Diagrama de Clases del Diseño Crear Audioconferencia.

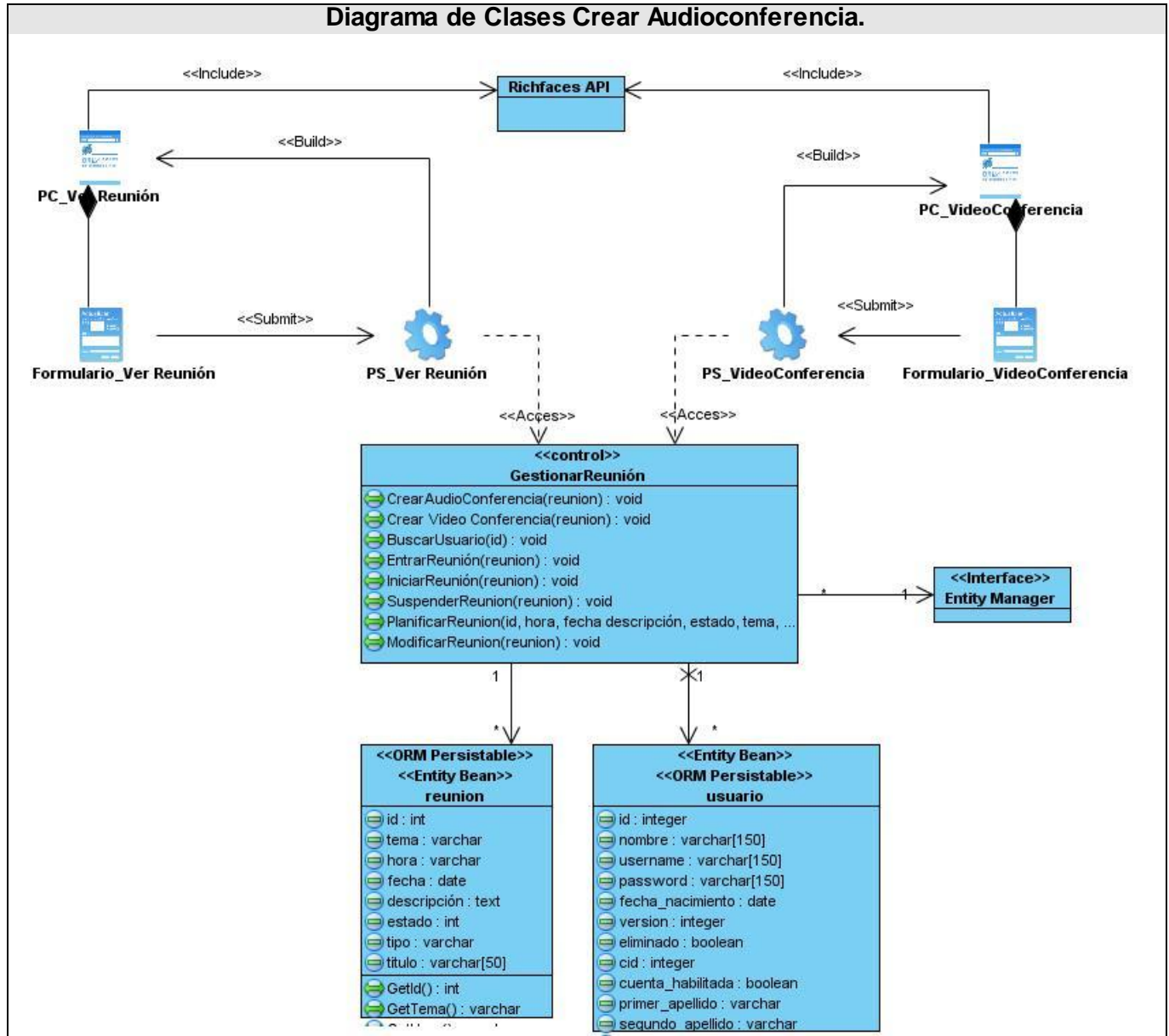
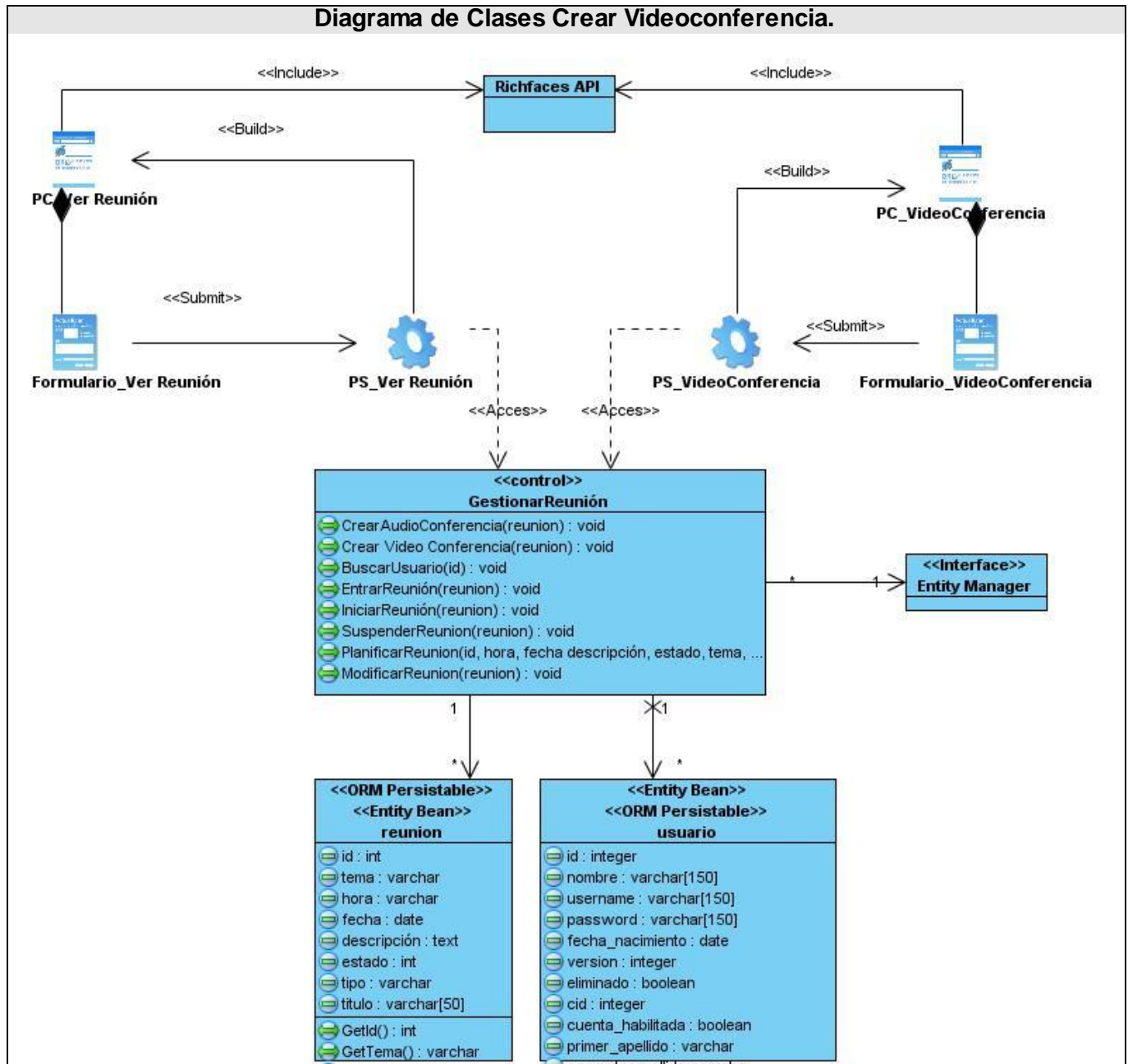


Diagrama de Clases del Diseño Crear Videoconferencia.



Anexo 4. Descripción de las Clases del Diseño.

Nombre: Sistema Teleconsulta	
Tipo de clase: Interfaz	
Responsabilidad:	
Descripción:	Representa la página principal para acceder a la planificación de reuniones.

Nombre: Videoconferencia	
Tipo de clase: Interfaz	
Responsabilidad:	
Descripción:	Muestra la interfaz donde se pueden comunicar los usuarios mediante video y audio.

Nombre: Audioconferencia	
Tipo de clase: Interfaz	
Responsabilidad:	
Descripción:	Muestra la interfaz donde se pueden comunicar los usuarios mediante audio.

Nombre: usuario	
Tipo de clase: Entidad	
Atributo	Tipo
id	integer
nombre	varchar
username	varchar
password	varchar
fecha_nacimiento	date

Nombre: usuario	
Tipo de clase: Entidad	
Atributo	Tipo
version	integer
eliminado	boolean
cid	integer
cuenta_habilitada	boolean
primer_apellido	varchar
segundo_apellido	varchar
dirección_particular	varchar
cedula	varchar
pasaporte	varchar
telefono	varchar
id_tipo_identidad	integer
Id_profile	integer

Nombre: reunion	
Tipo de clase: Entidad	
Atributo	Tipo
id	integer
tema	varchar
hora	varchar
fecha	date
descripcion	text
estado	integer
tipo	varchar
titulo	varchar
version	integer
eliminado	boolean

cid	integer
-----	---------

Nombre: notificacion_reunion	
Tipo de clase: Entidad	
Atributo	Tipo
id	integer
fecha	date
remitente	varchar
asunto	varchar
cuerpo	varchar
version	integer
eliminado	boolean
cid	integer

Nombre: notificacion_participante	
Tipo de clase: Entidad	
Atributo	Tipo
id	integer
leida	boolean
version	integer
eliminado	boolean
cid	integer

Nombre: usuario_externo	
Tipo de clase: Entidad	
Atributo	Tipo
id	serial
id_usuario	integer
correo	varchar
version	integer

Nombre: usuario_externo	
Tipo de clase: Entidad	
Atributo	Tipo
eliminado	boolean
cid	integer

Anexo 5. Descripción de las Clases del Modelo de Datos.

Nombre: usuario		
Descripción: Esta tabla almacena toda la información necesaria de los usuarios del sistema.		
Atributo	Tipo	Descripción
id	integer	En este atributo se almacena el identificador del usuario.
nombre	varchar	En este atributo se almacena el nombre del usuario.
username	varchar	En este atributo se almacena el usuario del especialista.
password	varchar	En este atributo se almacena la contraseña del usuario para acceder al sistema.
fecha_nacimiento	date	En este atributo se almacena la fecha de nacimiento del usuario.
version	integer	En este atributo se almacena el número de actualizaciones a la que ha sido sometida la tupla.
eliminado	boolean	Este atributo indica si el usuario es visible para el sistema.
cid	integer	Este atributo indica el identificador de modificaciones registradas en la bitácora.
cuenta_habilitada	boolean	Este atributo indica si la cuenta del usuario se encuentra habilitada.

Atributo	Tipo	Descripción
primer_apellido	varchar	En este atributo se almacena el primer apellido del usuario.
segundo_apellido	varchar	En este atributo se almacena el segundo apellido del usuario.
dirección_particular	varchar	En este atributo se almacena la dirección del usuario.
cedula	varchar	En este atributo se almacena la cédula del usuario.
pasaporte	varchar	En este atributo se almacena el número de pasaporte del usuario.
telefono	varchar	En este atributo se almacena el teléfono del usuario.
id_tipo_identidad	integer	En este atributo hace referencia a un nomenclador que brinda la información del tipo de documentación.
Id_profile	integer	En este atributo se almacena el perfil del usuario.

Nombre: notificacion_participante		
Descripción: Esta tabla almacena todas las notificaciones de los participantes.		
Atributo	Tipo	Descripción
id	serial	En este atributo se almacena el identificador de la notificación de los participantes.
id_user	integer	En este atributo se almacena el identificador del usuario participante de la reunión.
id_notif	integer	En este atributo se almacena el identificador de la notificación perteneciente a la reunión a la cual está invitado el usuario.

Atributo	Tipo	Descripción
leida	boolean	Este atributo indica si la notificación fue leída por el usuario.
version	integer	En este atributo se almacena el número de actualizaciones a la que ha sido sometida la tupla.
eliminado	boolean	Este atributo indica si la notificación es visible para el sistema.
cid	integer	Este atributo indica el identificador de modificaciones registradas en la bitácora.

Nombre: reunion_in_usuario		
Descripción: Esta tabla almacena todas las notificaciones de los participantes.		
Atributo	Tipo	Descripción
id	serial	En este atributo se almacena el identificador de la relación entre usuarios y reuniones.
id_usuario	integer	En este atributo se almacena el identificador de los participantes a la reunión.
id_reunion	integer	En este atributo se almacena el identificador de la reunión.
version	integer	En este atributo se almacena el número de actualizaciones a la que ha sido sometida la tupla.
eliminado	boolean	Este atributo indica si la reunión es visible en el sistema para los participantes eliminados de la misma.
cid	integer	Este atributo indica el identificador de modificaciones registradas en la bitácora.

Nombre: usuario_extremo		
Descripción: Esta tabla almacena toda la información de los usuarios externos que van a participar en la reunión.		
Atributo	Tipo	Descripción
id	serial	En este atributo se almacena el identificador del usuario externo.
id_usuario	integer	En este atributo se almacena el identificador del usuario.
correo	varchar	En este atributo se almacena el correo del usuario externo.
version	integer	En este atributo se almacena el número de actualizaciones a la que ha sido sometida la tupla.
eliminado	boolean	Este atributo indica si la reunión es visible en el sistema para los participantes eliminados de la misma.
cid	integer	Este atributo indica el identificador de modificaciones registradas en la bitácora.

GLOSARIO DE TÉRMINOS:

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 renderizándola, y luego muestra u oculta porciones de la misma.

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

Bioinformática: Es una disciplina científica emergente que utiliza tecnología de la información para organizar, analizar y distribuir información biológica con la finalidad de responder preguntas complejas en biología.

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

Diagrama de Colaboración: Muestran la interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos y las relaciones entre los roles de los objetos.

Encapsulamiento: Ocultamiento del estado, es decir, de los datos miembro, de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto.

Entorno de Desarrollo Integrado (IDE): Es una aplicación de software que ofrece servicios integrales a los programadores de computadoras para el desarrollo de software.

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

Glosario de Términos

Herencia: Es el mecanismo fundamental para implementar la reutilización y extensibilidad del software.

Hojas de Estilo en Cascada (CSS): Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

Infomed: Coordina y promueve el desarrollo de las actividades de información científica y técnica en el Sistema Nacional de Salud cubano.

Interfaz de Programación de Aplicación (API): Representa un interfaz de comunicación entre componentes software.

Java Enterprise Edition: Es una plataforma de programación, parte de la Plataforma Java, para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N capas.

JavaScript: Es un lenguaje de programación interpretado, se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario.

Lenguaje de Marcado de Hipertexto (HTML): Se utiliza para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Modelo Vista Controlador (MVC): Es un patrón de arquitectura de software compuesto de tres componentes distintos: datos, interfaz de usuario, y lógica del negocio.

MySpace: Es un sitio web, de interacción social constituido por perfiles personales de usuarios que incluye redes de amigos, grupos, blogs, fotos, vídeos y música, además de una red interna de mensajería que permite comunicarse a unos usuarios con otros y un buscador interno.

NetMeeting: Nombre de un programa de Microsoft para charlar, por medio del texto, la voz y de dibujos, a través de Internet.

Netscape: Es la empresa de software creadora del navegador web Netscape.

Glosario de Términos

Ofimática: Equipamiento hardware y software usado para crear, coleccionar, almacenar, manipular y transmitir digitalmente la información necesaria para lograr objetivos básicos.

Polimorfismo: En programación orientada a objetos el polimorfismo se refiere a la capacidad para que varias clases derivadas de una antecesora utilicen un mismo método de forma diferente.

Proceso: Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí, que a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido.

Rational Unified Process (RUP): Proceso unificado de desarrollo de software.

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

Lenguaje de Consulta Estructurado (SQL): es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

Sun Microsystems: Es un proveedor multinacional de ordenadores, software y hardware, y servicios de tecnología de la información, y un contribuyente importante de software de código abierto.

Lenguaje Unificado de Modelado (UML): Es un lenguaje de modelado visual para especificar, visualizar, construir y documentar artefactos de un sistema de software.

VoIP: Define los sistemas de enrutamiento y los protocolos necesarios para la transmisión de conversaciones de voz a través de Internet.

Xing: Es una red social de ámbito profesional. Su principal utilidad es la de gestionar contactos y establecer nuevas conexiones entre profesionales de cualquier sector.