

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

DESARROLLO DE UNA
APLICACIÓN COLABORATIVA
VÍA WEB PARA REUNIONES ELECTRÓNICAS ASÍNCRONAS

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela
Por la Bachiller:

Yolibert Coromoto Ghaleb Saba

Para optar al título de
Licenciada en Computación

Tutora

Profa. Nora Montaña

Caracas, Octubre de 2009

Acta

Quienes suscriben, miembros del Jurado designado por el Consejo de Escuela de Computación, para examinar el Trabajo Especial de Grado presentado por la bachiller Yolibert Coromoto Ghaleb Saba, C.I.:16.871.094, con el título: “**Desarrollo de una Aplicación Colaborativa vía Web para Reuniones Electrónicas Asíncronas**”, a fines de optar al título de Licenciada en Computación, dejan constancia lo siguiente:

Leído como fue, dicho trabajo por cada uno de los miembros del Jurado, se fijó el día 21 de octubre de 2009 a las 11:00 a.m, para que sus autores lo defiendan en forma pública, se hizo en el salón PB3 ubicado en la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, mediante una presentación oral de su contenido. Finalizada la defensa pública del Trabajo Especial de Grado, el Jurado decidió aprobarlo con una nota de puntos.

En fe de lo cual se levanta la presente Acta, en Caracas a los veintiún días del mes de octubre del año dos mil nueve, dejándose también constancia de que actuó como Coordinador del Jurado la profesora Nora Montaña.

Prof. Nora Montaña
(Tutor)

Prof. Yosly Hernández
(Jurado Principal)

Prof. Joali Moreno
(Jurado Principal)

Prof. Andrés Sanoja
(Jurado Suplente)

Prof. Antonio Silva
(Jurado Suplente)

Agradecimientos

Primero que nada quiero agradecerle a Dios por haberme dado la oportunidad de llevar a cabo mis estudios profesionales en la universidad donde siempre quise estudiar y abrirme los caminos durante toda la carrera para vencer las dificultades que se me presentaron.

A mis padres y hermanos por siempre estar ahí en los momentos de felicidad y angustias, apoyándome con sus palabras y consejos. En especial a mi mamá por tener la paciencia y el amor necesario que me levantó en las circunstancias más difíciles de mi vida cuando estuve a punto de decaer, sin ti no lo hubiese podido lograr gracias por existir.

A mis amigos de la carrera con los que he vivido momentos de alegría, pena, trasnochos por noches de estudio, nos hemos aconsejado, ayudado y apoyado.

A cada uno de los profesores que me impartieron los conocimientos necesarios que hoy en día hacen este sueño realidad, formarme como profesional de la ilustre Universidad Central de Venezuela la casa que vence las sombras.

A mi tutora Nora Montaña por tener la paciencia y guiarme a lograr exitosamente la culminación de mi Trabajo Especial de Grado.

Resumen

Actualmente es de gran interés el proveer herramientas usables que faciliten la comunicación entre los miembros de una organización, han surgido soluciones que superan las barreras de tiempo y espacio para facilitar la comunicación, como lo son el chat y los foros. Sin embargo, en busca de mejorar la comunicación se requiere evolucionar estas soluciones combinándolas con protocolos específicos que representan dinámicas de comunicación que surgen en ciertos niveles organizacionales. El objetivo de este Trabajo Especial de Grado consiste en el desarrollo de una Aplicación Colaborativa vía Web para coordinar, organizar y ejecutar reuniones electrónicas asíncronas orientadas bajo un protocolo de colaboración de Agendas y Actas.

El análisis se centró en caracterizar los diferentes aspectos al momento de organizar y llevar a cabo una reunión entre un grupo de trabajo. Algunos de estos aspectos son: discrepancia entre los miembros del grupo para llegar a un acuerdo que permita definir cuando se ejecutará la reunión y la necesidad de contar con un protocolo de colaboración que ayude al coordinador de la reunión y a los participantes en el proceso de interacción, que facilite alcanzar los objetivos propuestos.

La aplicación se elaboró siguiendo el método de desarrollo basado en el Modelado Ágil (MA), especificando como artefactos principales: el modelado de casos de uso y diagramas de flujo de trabajo. Patrones de interacción propuestos por Welie para la descripción de la interfaz y como tecnología a utilizar Ruby on Rails con la base de datos MySQL.

Tabla de Contenidos

Introducción.....	1
CAPÍTULO 1: MARCO CONCEPTUAL.....	5
1.1 Trabajo Cooperativo Mediado por el Computador CSCW (Computer Supported Cooperative Work)	5
1.2 Aplicaciones Colaborativas	7
1.2.1 Características de las Aplicaciones Colaborativas	8
1.2.2 Clasificación de las Aplicaciones Colaborativas	10
1.2.3 Las 3Cs De las Aplicaciones Colaborativas.....	13
1.2.4 Beneficios de las Aplicaciones Colaborativas	15
1.2.5 Desventajas de las Aplicaciones Colaborativas	15
1.3 Caracterización de un Modelo Conceptual de Reuniones	16
1.3.1 Reuniones Electrónicas	19
CAPÍTULO 2: MARCO METODOLÓGICO	21
2.1 Enfoque de Desarrollo: Modelado Ágil (AM).....	21
2.1.1 Agile Model Driven Development (AMDD).....	23
2.3 Patrón de Diseño MVC (Modelo Vista Controlador).....	27
2.4 Tecnologías a Utilizar en el Desarrollo de la Aplicación	28
2.4.1 Ruby on Rails.....	29
2.4.2 MySQL 5.0.....	31
CAPÍTULO 3: MARCO DE DESARROLLO.....	34
3.1 Análisis de la Solución Propuesta.....	34
3.2 Iteración 0: Iniciación del Proyecto	39
3.3 Desarrollo de las Iteraciones.....	53

3.4 Liberación del Producto.....	66
3.4.1 Producto Final	67
3.4.2 Pruebas a la Aplicación.....	68
Conclusiones.....	76
Referencias Bibliográficas.....	78
Anexos	80

Índice de Figuras

Figura 1. 1: Esquema de trabajo de CSCW. Interacción humano-computador, computador-computador y humano-humano (Bannai, 1996).....	7
Figura 2. 1: Ciclo de Vida de AMDD: Modelado de actividades del ciclo de vida del proyecto (Ambler, 2002)	23
Figura 2. 2: AMDD a través del Ciclo de Vida de Desarrollo Ágil (Ambler,2002)	24
Figura 2. 3: Los Requerimientos Agiles cambian la Gestión del Proceso (Ambler,2002)...	26
Figura 2. 4: Relación entre los diferentes Componentes de MVC (Catalani, 2007)	28
Figura 3. 1: Ciclo de Vida de una Reunión Asíncrona.....	35
Figura 3. 2 Modelo de Actividades. Ciclo de Vida de una Reunión Electrónica.	37
Figura 3. 3 Modelo de Actividades. Proceso Pre-Reunión.	37
Figura 3. 4 Modelo de Actividades. Proceso Ejecución de la Reunión.	38
Figura 3. 5 Modelo de Actividades. Proceso Post Reunión.	39
Figura 3. 6 Iteración 0 del proyecto.....	39
Figura 3. 7 Nivel 1 de los Casos de Uso, Usuario Autenticado	40
Figura 3. 8 Pila de Requerimientos de la Aplicación	44
Figura 3. 9 Lenguaje de Patrones de la aplicación.	46
Figura 3. 10 Patrón del Sistema.....	48
Figura 3. 11 Arquitectura MCV en Ruby on Rails de la Tesis.....	52
Figura 3. 12 Modelado del RF-1	54
Figura 3. 13 Modelo de Interfaz: Recuperar Contraseña.....	54
Figura 3. 14 Modelo de Interfaz: Modificar Perfil	55
Figura 3. 15 Modelo de Interfaz: Cambiar Contraseña	55
Figura 3. 16 Modelado del RF-6	56
Figura 3. 17 Modelo de Interfaz: Administrar Reunión	57
Figura 3. 18 Modelo de Interfaz: Modificar Reunión Tiempo	58
Figura 3. 19 Modelo de Interfaz: Modificar Reunión Agenda.....	58
Figura 3. 20 Modelo de Interfaz: Modificar Reunión Participantes.....	59
Figura 3. 21 Modelo de Interfaz: Crear Reunión.....	59

Figura 3. 22 Modelo de interfaz. Realizar Sugerencias. Vista Participante fase 1 de la reunión.....	60
Figura 3. 23 Modelo de interfaz. Editar Puntos. Vista Coordinador fase 1 de la reunión....	61
Figura 3. 24 Modelo de interfaz. Comentar Puntos. Vista Participante fase 2 de la reunión.	61
Figura 3. 25 Modelo de interfaz. Punto en Votación. Vista Coordinador fase 2 de la reunión.....	62
Figura 3. 26 Modelo de interfaz. Ver Agenda de la Reunión.....	63
Figura 3. 27 Modelo de interfaz. Ver Resultados de Reuniones Pasadas.	64
Figura 3. 28 Modelado del RF-5	65
Figura 3. 29 Modelo de interfaz. Agregar un Nuevo Contacto.	66
Figura 3. 30 Modelo de interfaz. Ver mis Contactos.....	66
Figura 3. 31 Modelo de interfaz. Usuarios en Línea.	66

Índice de Tablas

Tabla 1. 1: Cuadro de Percepción (McDaniel, 1996).....	10
Tabla 1. 2: Aplicaciones Colaborativas Clasificación Espacio/ Temporal	12
Tabla 2. 1 MVC en Ruby on Rails	30
Tabla 3. 1 Descripción breve de los casos de uso	42
Tabla 3. 2 Requerimientos No Funcionales de la Aplicación	43
Tabla 3. 3 Componentes para describir los patrones.....	46
Tabla 3. 4 Taxonomía de los Patrones de Interacción (Tidwell, 1999).....	46
Tabla 3. 5 Refinamiento del requerimiento funcional 6. Descripción breve de del casos de uso.....	57
Tabla 3. 6 Refinamiento del requerimiento funcional 5. Descripción breve de del casos de uso.....	65
Tabla 3. 7 Prueba Escenario Registro de Usuario.	69
Tabla 3. 8 Prueba Escenario Olvido su Contraseña.	70
Tabla 3. 9 Prueba Escenario Iniciar Sesión.	70
Tabla 3. 10 Prueba Escenario Modificar Perfil.	71
Tabla 3. 11 Prueba Escenario Cambiar Contraseña.	71
Tabla 3. 12 Prueba Escenario Ver Agenda de la Reunión.....	72
Tabla 3. 13 Prueba Escenario Crear Reunión.....	73
Tabla 3. 14 Prueba Escenario Modificar Reunión.....	74
Tabla 3. 15 Prueba Escenario Mis Contactos.	75
Tabla 3. 16 Prueba Escenario Ver Resultados de Reuniones Pasadas.	75

Introducción

Actualmente es de vital importancia cuando se está trabajando en un grupo mantener constantemente la comunicación entre sus miembros. La comunicación se transforma en un proceso social que permite la interacción entre las personas, a través del intercambio de ideas, opiniones, comentarios y sugerencias, las cuales ayudan a lograr los objetivos propuestos.

El uso de Internet mediante el computador y los dispositivos móviles, han facilitado el proceso comunicativo. Es posible encontrar en el mercado diferentes aplicaciones colaborativas, algunas de ellas son: BSCW (Basic Support for Cooperative Work, Soporte Básico para Trabajo en Grupo), Webex Meeting Center, IBM Lotus Sametime, PhpGroupware, las cuales contienen herramientas síncronas y asíncronas como: chat, foros de discusión, email, videoconferencias entre otras, que permiten llevar a cabo reuniones electrónicas y realizar otras funciones.

Sin embargo, es posible detectar que el uso de estas aplicaciones colaborativas carecen de protocolos de colaboración que ayuden a facilitar la creación y desarrollo de reuniones electrónicas, por tal motivo surgió la iniciativa de evolucionar estas herramientas combinándolas con protocolos de colaboración que faciliten el proceso comunicativo, la coordinación y ejecución de las reuniones.

Problema

La planificación y ejecución de una reunión requiere de muchos detalles y coordinación, esto aumenta al trabajar con un grupo numeroso de personas, durante ese proceso es posible encontrar ciertos conflictos como los que se nombran a continuación.

- Dificultad para llegar a un consenso entre los participantes a qué hora y día de la semana será efectuada la reunión en caso de trabajar con herramientas síncronas como: chat o videoconferencias, ya que cada uno puede tener diferentes obligaciones.

- Si se trabaja con una herramienta asíncrona como foros de discusión a pesar de que el aporte de las opiniones no tiene que ser en el mismo instante de tiempo, no se cuenta con un protocolo de colaboración que ayude a garantizar que realmente se logre los objetivos planteados. El coordinador de la reunión se ve limitado a solo leer las opiniones que han dado los participantes y aportar las suyas, más sin embargo no puede realizar acciones mucho más contundentes que permitan tratar de alcanzar el éxito de la reunión.

Objetivo General

Desarrollar una aplicación colaborativa vía web que permita llevar a cabo reuniones electrónicas asíncronas, basadas bajo un protocolo de colaboración de Agendas y Actas, utilizando un método de desarrollo Ágil soportado mediante la tecnología de Ruby on Rails.

Objetivos Específicos

- Caracterizar el protocolo a seguir en una reunión de tipo Agendas y Actas, para poder determinar las diferentes facetas y funcionalidades de la reunión.
- Aplicar el modelado ágil como método a seguir para el desarrollo de la aplicación propuesta mediante el lenguaje de programación Ruby on Rails.
- Diseñar las interfaces de usuario correspondiente a cada una de las funcionalidades que brinda la aplicación, siguiendo los patrones de interacción.
- Identificar las diferentes validaciones que tienen que ser llevadas a cabo en el sistema para su correcto funcionamiento.
- Realizar las pruebas necesarias que certifiquen que el sistema está funcionando correctamente.

Justificación de la Aplicación a Desarrollar

El desarrollo de esta Aplicación Colaborativa permitirá a sus usuarios poder planificar y coordinar reuniones electrónicas asíncronas, en el momento que consideren necesario llevar a cabo una reunión.

Al tratarse de una Aplicación vía Web y asíncrona las limitaciones de tiempo y espacio geográfico se reducen, ya que los participantes podrán estar ubicados en diferentes localidades y de igual forma participar en la reunión.

Adicionalmente las discusiones que se presentaban comúnmente entre los miembros de un grupo para poder llegar a un acuerdo sobre el instante en que se ejecutará la reunión desaparece, las reuniones serán planificadas por el coordinador quien especificará los intervalos de días en que durará, y los invitados tienen la opción de aceptar o rechazar la invitación.

Uno de los aspectos más importantes que provee la Aplicación es su protocolo de colaboración, el cual se basa en una reunión de Agendas y Actas. Este tipo de reuniones se centra básicamente en dos fases: la primera de ellas Orden de Día donde los participantes de la reunión dan sugerencias acerca de los puntos que conforman la agenda. En esta fase el coordinador decide qué acciones tomar si editar los puntos de la agenda, agregar o eliminar.

En la segunda fase Discusión de la Agenda ya se tiene la agenda definitiva se comienzan a discutir los puntos no informativos y el coordinador cuenta con privilegios que le permiten determinar cuándo colocar un punto a votación, en donde se especifica una resolución definitiva o en estado de listo; es decir que no se pueden hacer más comentarios sobre ese punto o votar. De igual forma el coordinador puede tomar la decisión en cualquier momento de cancelar la reunión si lo cree conveniente.

Al finalizar la reunión la aplicación genera un acta que contiene los diferentes comentarios realizados por cada participante a un punto y las estadísticas de votación sobre la resolución tomada.

Con el seguimiento de este protocolo de colaboración se puede facilitar el desempeño y organización de una reunión electrónica. Si hay disposición por parte de los participantes y el coordinador se puede obtener resultados favorables para el grupo de trabajo, al concluir la ejecución de la reunión en el sistema.

Este trabajo abarcará tópicos que se encuentran estrechamente relacionados con el tema de investigación en estudio. A continuación se presenta la estructura del contenido de cada uno de los capítulos de este trabajo:

CAPÍTULO 1: MARCO CONCEPTUAL. Se presentan los aspectos teóricos que respaldan el Trabajo Especial de Grado, tomando en cuenta los conceptos necesarios para ello. Definición del CSCW (Trabajo Cooperativo Mediado por el Computador), aspectos relacionados con las Aplicaciones Colaborativas: sus características, clasificación, beneficios y desventajas, concepto de Reunión Electrónica.

CAPÍTULO 2: MARCO METODOLÓGICO. Se describe la metodología de desarrollo, el patrón de diseño a seguir y por último se explica la tecnología a utilizar.

CAPÍTULO 3: MARCO DE DESARROLLO. Se esbozan las actividades de desarrollo enmarcadas en la metodología adoptada, con la finalidad de exponer o documentar formalmente el proceso de implementación del producto propuesto.

Finalmente se presentan las conclusiones que fueron posibles obtener con la realización del Trabajo Especial de Grado.

CAPÍTULO 1: MARCO CONCEPTUAL

En este capítulo se presentan y explican algunos conceptos teóricos que respaldan el Trabajo Especial de Grado desarrollado.

Uno de los objetivos que persigue la aplicación a desarrollar es proveer a los miembros que conforman un grupo el ambiente propicio y las herramientas necesarias para que puedan trabajar en conjunto por medio del uso de las tecnologías, con el fin de lograr los objetivos propuestos. Por lo tanto surge el concepto de CSCW (Trabajo Cooperativo Mediado por el Computador).

Adicionalmente se procede a la definición de Aplicaciones Colaborativas, las características que deben presentar, su clasificación, ventajas y desventajas y finalmente se establece el contexto de reuniones electrónicas dentro del marco del CSCW.

1.1 Trabajo Cooperativo Mediado por el Computador CSCW (Computer Supported Cooperative Work)

CSCW (Computer Supported Cooperative Work) son las siglas en inglés de “Trabajo cooperativo mediado por el computador”. El CSCW fue utilizado por primera vez en 1984 por Irene Greif, del Massachusetts Institute of Technology, y Paul Cashman, de la Digital Equipment Corporation. En ese año tuvo lugar en Estados Unidos el primer taller de CSCW, cuyo objetivo fue reunir a personas de diferentes disciplinas para analizar las características de comunicación y coordinación en grupos de trabajo (Martínez, 2005).

Fue en 1984, debido a un enfoque del trabajo en grupo, que ocurrió la llamada “automatización de oficinas”, la cual cristalizó un esfuerzo para integrar las aplicaciones de un solo usuario para soportar grupos.

Pero en la automatización de oficinas no había el conocimiento necesario para entender cómo es que la gente trabaja en grupos y organizaciones y cómo la tecnología afecta su trabajo. Lo que trajo consigo un interés hacia el estudio de CSCW por parte de economistas, psicólogos sociales, antropólogos teóricos y organizacionales, educadores y muchos otros que pudiesen aportar un mejor entendimiento de las actividades de grupo.

En la actualidad ha llegado a ser un área para que los desarrolladores compartan experiencias y discutan posibilidades y restricciones tecnológicas referentes al trabajo grupal.

No existe una definición universalmente aceptada de CSCW, pero entre las más recurridas es posible mencionar las siguientes:

- “Es un término genérico que combina el entender la manera en que la gente trabaja en grupos, con la habilitación de tecnologías de redes de computadoras y el hardware, software, servicios y técnicas asociadas” (Wilson, 1991).
- “Es una combinación de tecnología, gente y organización que facilita la comunicación y coordinación necesarias para que un grupo trabaje unido con eficiencia en un propósito común y logre así una ganancia en todos sus miembros” (Ramage, 1999).
- “Es un campo de investigación multidisciplinario que incluye a las ciencias de la computación, la economía, la sociología y la psicología. Las investigaciones se enfocan al desarrollo de nuevas teorías y tecnologías para la coordinación de personas que trabajan juntas” (Martínez, 2005).

A partir de todas estas definiciones es posible extraer una definición propia:

CSCW es un campo encargado de la investigación, desarrollo de principios, herramientas y técnicas que sirvan de apoyo al trabajo cooperativo mediante el uso del computador, por parte de un grupo de trabajo previamente establecido los cuales pueden mantener la comunicación, cooperación y coordinación gracias al uso de la combinación de las diferentes tecnologías redes, hardware, software y aplicaciones que presten los servicios necesarios para conseguir un objetivo común que beneficie a todos sus miembros.

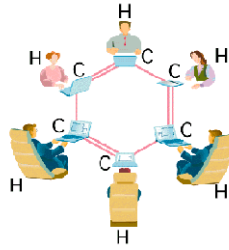


Figura 1. 1: Esquema de trabajo de CSCW. Interacción humano-computador, computador-computador y humano-humano (Bannai, 1996)

En la figura 1.1 se puede apreciar como es el esquema de trabajo de CSCW, en donde la interacción humano-computador, computador-computador y humano-humano puede ocurrir gracias a la interconexión de las redes de computadoras.

1.2 Aplicaciones Colaborativas

Como se mencionó anteriormente el CSCW se encarga de toda la parte investigativa y el desarrollo de técnicas que sirvan de apoyo al trabajo cooperativo entre los grupos, mediante el uso del computador a través de sistemas denominados hoy en día como aplicaciones colaborativas, los cuales se pueden definir de la siguiente manera:

El uso del computador debe estar orientado por un elemento de software que permita la comunicación, colaboración y cooperación entre un grupo de trabajo, que se reúnen con el fin de trabajar en conjunto sobre una tarea común para alcanzar objetivos específicos de manera coordinada.

En la década de los 80 estas aplicaciones eran conocidas como **Groupware** que fue empleado por primera vez por Peter y Trudy Johnson Lenz. Quienes lo definieron como: “Procesos de Grupos intencionales y procedimientos para realizar propósitos específicos” más “Herramientas de software diseñadas para soportar y facilitar el trabajo en grupo” (Johnson-Lenz, 1981).

Las Aplicaciones Colaborativas deben estar orientadas a:

- **Proveer** un ambiente de colaboración, en el que realmente se perciba que el trabajo en grupo se lleva a cabo.
- **Mantener** la información en un solo sitio común para todos los miembros.

- **Interactuar** con otros usuarios, de forma escrita, voz o video.

1.2.1 Características de las Aplicaciones Colaborativas

Para que toda Aplicación Colaborativa pueda funcionar correctamente y brinde un ambiente adecuado que ayude a la comunicación, cooperación y coordinación entre los grupos de trabajo, es necesario que contengan una serie de características que se mencionan a continuación.

a) Memoria Grupal

La memoria grupal es el espacio común donde los miembros de un grupo almacenan información en forma ordenada referente al desarrollo de la actividad realizada por el mismo (Ellis, 1991).

Este espacio es creado con la finalidad de brindar al grupo un dispositivo efectivo de comunicación, ya que los individuos autorizados podrán acceder en cualquier momento y visualizar la información publicada por otros miembros del grupo. La Memoria Grupal debe contar con dispositivos de comunicación tanto sincrónicos como asincrónicos, dependiendo del contexto de la aplicación colaborativa implementada.

b) Roles

Un rol es un conjunto de privilegios y responsabilidades atribuidas a una persona, o a veces a un módulo de sistema.

c) Protocolos de Colaboración

Son las distintas maneras de interactuar de las personas consensuadas por el grupo. Son reglas que permiten a los individuos comunicarse entre sí de tal forma que cada uno pueda enviar y recibir señales comprensibles para los demás.

Un protocolo de colaboración es de suma importancia en el proceso de trabajo en grupo, ya que el mismo tiene como principal objetivo proporcionar una serie de instrucciones preestablecidas, que ayuden a lograr los objetivos planteados.

Los protocolos grupales o de colaboración pueden ser intuitivos o informales. Estos tienen su origen en el entorno social y se hacen necesarios siempre que las personas colaboran en una tarea y se transforman en protocolos de acceso a la información cuando éstas lo hacen mediante el computador, pues el lenguaje, las palabras, los gestos se traducen en datos, los cuales necesitan ser almacenados, accedidos, comprendidos y transmitidos (Ellis, 1991).

d) Percepción

Debido a que la distancia física disminuye dramáticamente la comunicación entre los miembros de un grupo, se hace necesario crear mecanismos para proveer información sobre la actividad del grupo (McDaniel, 1996). Es por esta razón que la percepción juega un papel de gran importancia en las aplicaciones colaborativas, donde no se da una interacción cara a cara entre los individuos que conforman un grupo de trabajo.

Se puede definir percepción como: toda información que provee una conciencia grupal al individuo que forma parte de un grupo. La manera como ésta conciencia se obtiene es suministrando información, la cual el usuario interpreta dependiendo de sus requerimientos.

Ésta información se puede dar de dos formas: mediante la percepción de datos y la percepción de usuarios. La percepción de usuarios provee información sobre los miembros del grupo, por ejemplo: informar quiénes están conectados y lo que estos hacen. La percepción de datos suministra información referente a los cambios efectuados sobre los datos. Parece interesante notar que tanto en el caso cara a cara como en el distribuido asincrónico, la percepción que casi siempre es requerida es la de datos. (McDaniel, 1996).

Según (McDaniel, 1996), percepción es cualquier respuesta a las preguntas: quién, qué, cuándo, dónde, por qué y cómo. Para clarificar esta propuesta se muestra la Tabla 1.1, en este ejemplo se ha tomado el caso de la comunicación mediante computador, es por esta razón que aparece el caso sincrónico y el asincrónico:

	Sincrónico	Asincrónico
Quién	Quién está integrado a la sesión Quién está disponible	Quién agregó determinado artefacto
Qué	Qué están haciendo los otros durante la sesión compartida Qué información propia es enviada a otros	Qué datos son enviados a otros
Cuándo	Cuándo los documentos compartidos están siendo usados por otros	Cuándo los artefactos compartidos han sido cambiados por otros Cuándo los otros han enviado mensajes
Dónde	Dónde se encuentran situados los otros	Dónde los otros han enviado mensajes
Porqué	Porqué los otros han ejecutado alguna acción	Porqué los otros han cambiado un artefacto compartido
Cómo	Cómo los miembros de grupo se comunican	Cómo se manipulan los artefactos

Tabla 1. 1: Cuadro de Percepción (McDaniel, 1996)

1.2.2 Clasificación de las Aplicaciones Colaborativas

La clasificación de las Aplicaciones Colaborativas hasta ahora se ha hecho de muchas maneras, pero lo cierto es que todas ellas se basan en una matriz inicial ideada por Johansen en 1988 denominada “Matriz Espacio/Temporal”.

a) Espacio Temporal:

En esta matriz de clasificación se toman en cuenta dos aspectos fundamentales en el entorno de trabajo en grupo: el aspecto temporal o la forma de interactuar entre los miembros del grupo, bien sea al mismo tiempo o en diferentes momentos, y el aspecto espacial o dispersión geográfica de los miembros del grupo.

Aspecto Temporal

En el aspecto temporal indica que hemos de considerar si la comunicación entre los miembros del grupo es síncrona o asíncrona (Hernández, 1996).

- **Interacción síncrona:** hace referencia a las herramientas que facilitan la comunicación en un mismo lugar (espacio) y en un mismo momento (tiempo). Dentro de los ejemplos clásicos se tendrían: las reuniones cara a cara, las conversaciones telefónicas, y entre los ejemplos de interacciones síncronas mediante el uso del computador podemos mencionar reuniones electrónicas, chats, videoconferencias.
- **Interacción Asíncrona:** en esta clasificación encontramos las herramientas que sirven como facilitadoras de los procesos de comunicación y permiten compartir información entre personas que no interactúan simultáneamente. Ejemplos de este tipo tenemos el correo electrónico, las conferencias electrónicas, los sistemas de flujo de documentos, foros de discusión.

Aspecto Geográfico

En el aspecto geográfico o espacial hace referencia a que los miembros del grupo pueden trabajar juntos en el mismo lugar o estar dispersos en diferentes lugares.

En este caso hay que mencionar que la comunicación síncrona o asíncrona se puede dar en diferentes lugares que es lo que se conoce como *distribuida o remota* o en el mismo sitio y se conoce como *local*.

En la tabla 1.2 se puede apreciar las diferentes herramientas que pueden estar o no presentes en una aplicación colaborativa, de acuerdo a su clasificación tomando como referencia el aspecto temporal y geográfico mencionado anteriormente.

	Síncrono (Mismo Tiempo)	Asíncrono (Diferente Tiempo)
Local (Mismo Lugar)	Interacción cara a cara Reuniones Virtuales Pantallas Compartidas	Herramientas de coordinación Organizadores de proyectos Sistemas de escritura colaborativa
Distribuido o Remoto (Diferente Lugar)	Conversación Telefónica Pizarra Electrónica Chat Video Conferencias Sistema para la Toma de Decisiones	Transferencia de Archivos Correo Electrónico Foros de Debate Flujo de Trabajo Wiki Grupos de Discusión Blog

Tabla 1. 2: Aplicaciones Colaborativas Clasificación Espacio/ Temporal

Otra de las clasificaciones que se le pueden hacer a las Aplicaciones Colaborativas tiene que ver con la funcionalidad que brindan a los grupos de trabajo.

b) Según su Funcionalidad

Las aplicaciones colaborativas necesitan trabajar en conjunto con una serie de herramientas que permitan brindar la mayor cantidad de funcionalidades a los miembros del grupo, para que estos puedan lograr los objetivos que se han propuestos.

En función de todo esto podemos encontrar herramientas basadas en Web de gran utilidad para los miembros, las cuales se describen a continuación:

Herramientas de comunicación electrónica: que envían mensajes, archivos, datos o documentos entre personas y facilitan el compartir información (colaboración asíncrona), como por ejemplo:

- Correo electrónico
- Correo voz
- Publicación en web

Herramientas de conferencia electrónica: que facilitan el compartir información de forma interactiva (colaboración síncrona), como por ejemplo:

- Pantallas Compartidas: cada participante coloca datos en una pizarra que todos los otros pueden leer y modificar.
- Video Conferencia: los participantes comparten video y señales de audio para intercambiar información.
- Discusión Virtual para facilitar el manejo en línea de mensajes de texto.
- Conferencias de voz: teléfonos que permiten interactuar a los participantes.
- Salas de chat o mensajería instantánea: una plataforma de discusión que facilita el intercambio inmediato de mensajes
- Sistemas para facilitar reuniones: un sistema de conferencias integrado en una sala. Estas salas suelen disponer de un avanzado sistema de sonido y presentación que permite una mejor interacción entre participantes en una misma sala o entre salas separadas. Ejemplos de ello son los Sistemas de soporte a decisiones

Herramientas de gestión colaborativa: que facilitan las actividades del grupo, como por ejemplo:

- Calendarios electrónicos: para acordar fechas de eventos y automáticamente enviar notificaciones y recordatorios a los participantes.
- Sistemas de gestión de proyectos: para organizar y hacer seguimiento de las acciones en un proyecto hasta que este finaliza.
- Sistemas de control de flujos de trabajo (workflow): para gestionar tareas y documentos en un proceso organizado de forma estructurada.
- Sistemas de soporte a redes sociales: para organizar las relaciones de los grupos.
- Wikis, Blogs, pizarras compartidas y grupos de discusión.

1.2.3 Las 3Cs De las Aplicaciones Colaborativas

Hoy en día las Aplicaciones Colaborativas se están volviendo más populares dentro de las organizaciones, ya que resulta más barato instalar una Intranet y comprar o implementar un sistema de colaboración a estar transportando personal de un lugar a otro.

Además si se necesita tomar una decisión urgente y las personas se encuentran en diferentes partes del mundo, esto ya no es un factor limitante debido al uso de las Aplicaciones Colaborativas las cuales ayudan a romper estas barreras.

Es por esto que las Aplicaciones Colaborativas deben proporcionar tres funciones esenciales dentro de un grupo, llamadas las 3Cs (Comunicación, Colaboración y Coordinación), a continuación la definición de cada una de ellas:

a) Comunicación:

Es la base de las Aplicaciones Colaborativas. Los miembros de un grupo de trabajo necesitan comunicarse los unos a los otros para intercambiar información. Para ello utilizan herramientas de comunicación electrónica que envían mensajes, archivos, datos o documentos entre personas. Estas herramientas permiten a los usuarios comunicarse fácil y rápidamente entre sí.

b) Colaboración:

Las Aplicaciones Colaborativas no sólo fomentan la colaboración entre los miembros del equipo, sino que la impone.

La idea es compartir y propagar el conocimiento, además de proponer un marco en el que cada miembro puede aportar ideas con el fin de que se pueda conseguir una conclusión final que ayude a lograr las metas propuestas, las cuales persiguen un objetivo común para cada integrante del grupo formado.

c) Coordinación:

Las Aplicaciones Colaborativas permiten la coordinación de los distintos miembros del grupo para la realización de un determinado proyecto común. Esta coordinación puede estar reflejada mediante el seguimiento de una serie de protocolos previamente establecidos.

Al unir estas tres características Comunicación, Colaboración y Coordinación, las Aplicaciones Colaborativas permiten que las metas propuestas se puedan realizar de una manera más eficiente y sin la necesidad de consumir muchos recursos. Esto gracias a que se

pueden ejecutar las actividades de forma coordinada, reduciendo o eliminando por completo las barreras que antes nos separaban como lo son las de tiempo y geografía y haciendo más rápido el proceso de toma de decisiones.

1.2.4 Beneficios de las Aplicaciones Colaborativas

Algunos beneficios que se buscan al implementar las aplicaciones colaborativas para el trabajo en proyectos son:

- Estimulan la cooperación dentro de una organización y ayuda a las personas a comunicarse y colaborar en proyectos comunes.
- Coordinan gente y procesos.
- Ayudan a definir el flujo de documentos y después definen el trabajo que se debe hacer para terminar un proyecto.
- Proporcionan a los usuarios una manera única para compartir información, construyéndola en documentos estructurados. El documento se convierte entonces en el lugar central en donde se almacena la información compartida.
- Idealmente, las aplicaciones colaborativas deben poder ayudar a cada persona en un proyecto de colaboración a realizar su trabajo en específico de una manera más eficiente.

1.2.5 Desventajas de las Aplicaciones Colaborativas

Entre las desventajas que se pueden presentar al trabajar con una aplicación colaborativa se destacan:

- Diferencias en los niveles de experiencia y conocimiento de tecnologías por parte de los miembros en los equipos de trabajo.
- Puede que algunos de los miembros tengan dificultades para utilizar y entender ciertas tecnologías.
- En el uso de tecnologías puede haber éxito para un grupo y puede ser fracaso para otros.
- Limitaciones al acceso de la tecnología mediante internet.

A continuación se destaca como debe ser un modelo de reunión, esto con la finalidad de tomar las bases teóricas que se exponen como referencia, en el momento de implementar la aplicación colaborativa propuesta, la cual persigue como objetivo principal proveer de herramientas que permitan a los usuarios llevar a cabo reuniones electrónicas asíncronas planificadas por los mismos.

1.3 Caracterización de un Modelo Conceptual de Reuniones

Es común que las personas tengan la necesidad de reunirse para lograr objetivos específicos como por ejemplo: intercambiar opiniones sobre algún tema, intercambiar conocimientos, tomar una decisión, seminarios, consejos, congresos, entre otros. Por todo esto podemos afirmar que la reunión es una herramienta importante de cooperación, coordinación y comunicación.

El hecho de que se realice una reunión entre un grupo de personas no garantiza su éxito; es decir que se logren los objetivos propuestos. Muchas veces la organización de una reunión amerita enormes gastos tanto de tiempo como económicos, y resulta mucho más difícil que un grupo compuesto de varias personas logren obtener una conclusión o decisión final. Por lo tanto esto es importante tomar en cuenta una serie de factores que sirvan de apoyo a la reunión y que permitan garantizar en gran parte su éxito.

Para el éxito de una reunión, es importante considerar las características del grupo, las cuales se detallan a continuación (Jay, 1976):

- **Tamaño:** Si es grande o pequeño
- **Proximidad:** Si están en un mismo sitio o distribuido en distintos lugares.
- **Composición:** Si las relaciones entre ellos son jerárquicas o son a un mismo nivel, si trabajan en un mismo proyecto, o en proyectos paralelos o si están en diferentes proyectos y el único interés entre ellos radica en dar cuenta de sus objetivos.
- **Compenetración:** Su grado de cohesión, pues las relaciones interpersonales agradables reducen la intimidación y permiten un cambio de orientación hacia los objetivos del grupo.

- **Frecuencia:** Una reunión diaria es diferente a una semanal y esta diferente a una mensual, etc.
- **Motivación:** Si los miembros del grupo tienen un objetivo común o es el deseo del éxito de la reunión el que hace la unificación.

Aunque son muy importantes las características del grupo, también se debe tomar en cuenta la peculiaridad de las acciones que deben realizarse para llevar a cabo una tarea, las cuales pueden ser (Nunamaker, 1991):

- Generación de ideas.
- Toma de decisiones.
- Elección de cursos de acción a seguir.
- Mecanismos para obtener metas.

También es importante tomar en cuenta el entorno en el que se desarrolla la reunión: si hay presión de tiempo, si el tono de la reunión es crítico o apoyado, es decir si se estimula o no la participación, y si la reunión está bien estructurada y coordinada.

De igual forma no se puede dejar a un lado las categorías de reuniones, las cuales pueden ser (Jay, 1976):

- **La reunión diaria:** Es aquella donde las personas están unidas en un proyecto con un objetivo común y el alcance de las decisiones es informal por un acuerdo general.
- **La reunión semanal o mensual:** Es aquella donde los miembros trabajan en diferentes pero paralelos proyectos y donde hay ciertos elementos de competitividad y una gran probabilidad de que el que preside la reunión sea el que toma las decisiones.
- **La reunión irregular:** Es una reunión ocasional o por un proyecto especial, y está compuesta por personas que normalmente no están en contacto y tienen pocas relaciones en común.

La organización de las reuniones es de vital importancia ya que permite a los miembros del grupo prepararse antes de la reunión, para que los mismos puedan adquirir información

valiosa que les permite entablar una discusión coherente con el grupo sobre cada uno de los puntos a discutir, lo que ayuda a reducir las equivocaciones.

Las preguntas más importantes respecto a la finalidad de la reunión son:

- ¿Qué se pretende lograr con la reunión?
- ¿Cuál(es) sería(n) la(s) consecuencia(s) probable(s) de no celebrarse la reunión?

Para que una reunión sea exitosa se debe planear con anticipación, por lo que se debe proveer todo un conjunto de acciones tendientes a la realización exitosa de la misma. Primero se tiene que nombrar el coordinador de la reunión quien ayudara a tratar de garantizar el éxito por medio de la conducción de la misma y tomando las decisiones más relevantes. También se debe analizar las actividades a realizar antes de la reunión. Aquí se deben estudiar los problemas que son susceptibles a ser tratados en la misma, de forma tal que oriente a los participantes en posibles soluciones para así agilizar la reunión.

Otro punto sumamente importante antes de llevarse a cabo una reunión es determinar la lista de participantes que serán invitados. El éxito de la reunión puede verse amenazado si se ha invitado a un número de personas relativamente grande; por lo que el coordinador debe realizar un análisis previo que permita determinar quienes son las personas más indicadas para que asistan y distribuir el material necesario.

Adicionalmente otra de las acciones previas a la reunión sería definir los ítems de la agenda, los cuales pueden ser de cuatro tipos (Jay, 1976):

- **Informativo:** Se refiere a información de hechos, los cuales pueden ser introducidos en la reunión pero no necesitan conclusiones, ni decisiones o acciones a tomar
- **Constructivo:** Es aquel ítem que requiere una nueva idea, tal como una nueva política, un nuevo plan de mercadeo, un nuevo producto, un nuevo procedimiento, y así sucesivamente.
- **Responsabilidades ejecutivas:** Este tipo de ítem responde a la pregunta ¿cómo debemos hacerlo?. Son muy importantes pues permiten que los miembros de un grupo encuentren el camino para alcanzar sus objetivos, además de habilitar a cada miembro a entender y a influenciar la manera en que su trabajo encaja con el trabajo

del grupo. También compromete a cada miembro del grupo a cumplir su parte y lo hace colectivamente responsable del éxito del proyecto.

- **Marco legislativo:** Son los ítems referentes a qué se acordó y cómo se realizarán los acuerdos convenidos en la reunión.

1.3.1 Reuniones Electrónicas

Las reuniones electrónicas son sistemas que ofrecen el ambiente necesario para simular una reunión cara a cara, esto lo logran mediante el apoyo del hardware y software los cuales componen el cuerpo de la aplicación más la logística de negocio implementada en la misma.

Generalmente las reuniones electrónicas se llevan a cabo mediante diferentes estaciones de trabajo interconectadas en la red. Así como también la aplicación brinda todas las funcionalidades necesarias para que pueda llevarse a cabo la reunión, estas funcionalidades pueden ser de tipo asíncronas o síncronas. Estos sistemas incluyen toda la ayuda necesaria para que se pueda crear la reunión, herramientas de generación de ideas, organización y priorización de ideas propuestas y sistemas de apoyo a las decisiones.

Entre las ventajas potenciales de las reuniones electrónicas podemos destacar:

- La generación automática de una memoria de la reunión, la cual aumenta la sinergia y una gran cantidad de información disponible, que conlleva la disminución del uso inadecuado o parcial de dicha información.
- Una comunicación paralela que genera más información en menos tiempo, ya que es más estimulante para los participantes generar información en el contexto de un trabajo en equipo, dando como ventaja también la disminución del fenómeno de dominación por un sólo miembro de la reunión, como muchas veces sucede en las reuniones presenciales.
- Permite la posibilidad del anonimato, lo que aumenta la objetividad de la información y disminuye la presión ocasionada por la jerarquía en los miembros del grupo.

En cuanto a la tecnología, se puede avanzar más resolviendo los problemas de transmisión de grandes volúmenes de datos, principalmente reduciendo los costos, ya que las reuniones electrónicas son una alternativa para disminuir la necesidad de reuniones cara a cara.

Esto es especialmente interesante cuando los integrantes de un grupo de discusión se encuentran distantes físicamente, pues así pueden realizar frecuentemente reuniones en su tiempo libre y no dedicar con demasiada frecuencia una hora determinada para realizar reuniones cara a cara.

CAPÍTULO 2: MARCO METODOLÓGICO

En la ingeniería de Software, las metodologías de desarrollo proveen mecanismos para diseñar estrategias de desarrollo de software que ayuden al codificador a lograr exitosamente la culminación del proyecto trazado y que al mismo tiempo permitan mantener la comunicación con el cliente. Por esta razón es de vital importancia seleccionar la metodología adecuada para realizar un proyecto, su selección va a depender del tipo de software que se quiere realizar y los requerimientos que el mismo presente.

En este capítulo se presenta la metodología seleccionada para el desarrollo de la aplicación propuesta específicamente el modelado ágil, adicionalmente se explica las tecnologías que serán utilizadas para su producción, se destaca como lenguaje de programación Ruby apoyado bajo el framework de Rails, los cuales trabajan con el patrón de diseño MVC, apoyado mediante la base de datos MySQL.

2.1 Enfoque de Desarrollo: Modelado Ágil (AM)

El enfoque seleccionado para llevar a cabo el desarrollo de la aplicación propuesta es el modelado ágil, ya que el mismo permite dividir el ciclo de vida del proyecto en diferentes fases, que serán realizadas en varias iteraciones de acuerdo al grado de extensión del proyecto.

Otras de las razones del porque trabajar con esta metodología es el hecho de que este enfoque permite mantener una constante comunicación entre el cliente y el desarrollador, de esta forma se pueden realizar pruebas que certifiquen si los resultados alcanzados cumplen con las expectativas del cliente en ese instante de tiempo. Adicionalmente el modelado ágil provee de todo lo necesario para tratar de garantizar que la aplicación se pueda realizar completamente en un intervalo de tiempo relativamente corto.

Origen

En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término ágil aplicado al desarrollo de software. En esta reunión participan un grupo de diecisiete expertos de la industria del software, incluyendo algunos de los creadores o impulsores de

metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó The Agile Alliance, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos.

Modelado Ágil

Según Ambler (2002) el Modelado Ágil (AM) es una metodología basada en prácticas para una efectiva modelación y documentación de sistemas de software. AM simplemente es una colección de valores, principios y prácticas para la modelación de software que puede ser aplicado en un proyecto de desarrollo de software de una forma efectiva y sencilla. El secreto de la metodología AM no son las técnicas de modelado en sí mismas, tal como modelos de caso de uso, modelos de clases, modelos de datos, o modelos de interfaces de usuario, pero sí cómo son aplicadas. Se deberá tener un modelado ágil enfocándose en los requerimientos, análisis, arquitectura y diseño.

Principios de AM

AM define una serie de principios que pueden ser aplicados en un proyecto de desarrollo de software para crear las bases para un conjunto de prácticas de modelado. Entre los cuales podemos destacar:

- Lo principal en el modelado ágil es la satisfacción del cliente, esto se pretende conseguir mediante tempranas y continuas entregas del software que le aporten un valor al mismo.
- Flexibilidad a los cambios, se capturan los cambios para que el cliente tenga ventajas competitivas.

- Entregar frecuentemente módulos del software que funcionen correctamente, estas entregas tienen que ser en el menor intervalo de tiempo posible entre entregas.
- Los clientes y los desarrolladores deben trabajar en conjunto para lograr los objetivos propuestos.
- El modelado ágil promueve un desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deben ser capaces de mantener el paso constante indefinidamente.
- La calidad del código y diseño debe ser continua.
- La simplicidad es lo mejor.
- Los desarrolladores se organizan entre ellos mismos se administran los cargos, muchas veces los mejores diseños surgen de esto.
- El equipo se toma tiempo para pensar en lo que está realizando y como mejorarlo.

2.1.1 Agile Model Driven Development (AMDD)

Propuesto por Scott W. Ambler, con el Agile Model Driven Development (AMDD) en vez de desarrollar modelos extensos antes de escribir el código se crean modelos ágiles, los cuales son modelos simples usados para explorar y analizar los requerimientos del sistema. AMDD es básicamente una estrategia crítica para ir escalando los procesos de desarrollo ágiles más allá de los proyectos pequeños.

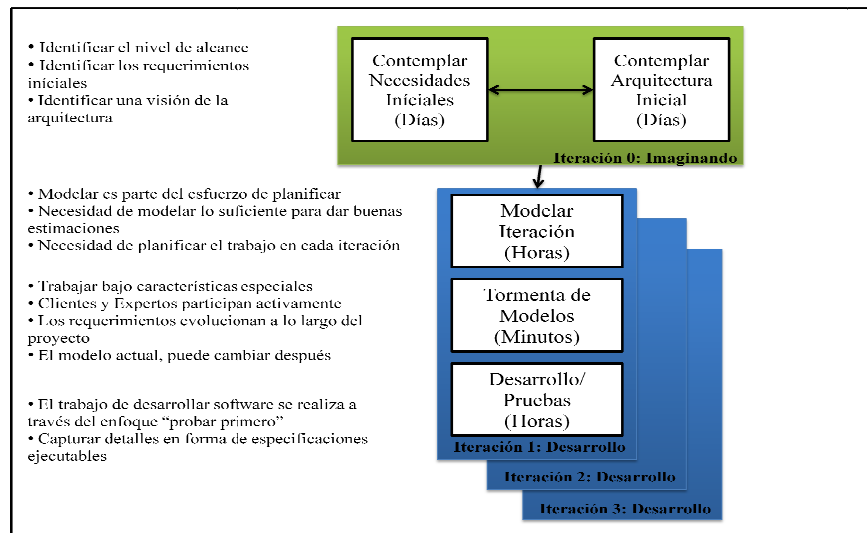


Figura 2. 1: Ciclo de Vida de AMDD: Modelado de actividades del ciclo de vida del proyecto (Ambler, 2002)

La Figura 2.1, representa el ciclo de vida según el modelado ágil para el desarrollo de un sistema, donde cada caja representa una actividad de desarrollo.

La primera actividad la cual está representada por la caja de color verde, está integrada por dos sub-actividades que son: contemplar las necesidades iniciales y contemplar la arquitectura inicial, la misma se lleva a cabo durante la iteración 0 de desarrollo, las actividades restantes Modelar Iteración, Tormenta de Modelos y Desarrollo/Pruebas se pueden ejecutar en cualquier iteración no hay restricciones para esto, incluyendo en la iteración 0. El tiempo indicado en cada caja representa el tamaño de una sesión promedio.

La figura 2.2 que aparece a continuación refleja como encajan las actividades de AMDD en las iteraciones del ciclo de vida de desarrollo ágil. Es simplemente otra manera de mostrar que un proyecto ágil comienza con algún modelaje inicial y que este modelaje sigue ocurriendo en cada una de las iteraciones.



Figura 2. 2: AMDD a través del Ciclo de Vida de Desarrollo Ágil (Ambler,2002)

En el modelado ágil generalmente durante la primera semana de desarrollo del proyecto se realizan las investigaciones y análisis necesarios para contemplar el alcance que tendrá el sistema; así como también la arquitectura probable para tratarla. Para poder hacer esto es indispensable obtener los requerimientos de alto nivel y modelar la arquitectura de lo necesitado. El objetivo no es escribir especificaciones detalladas, que sean arriesgadas, sino explorar las necesidades y llegar a una estrategia global.

Para los proyectos cortos (quizás de varias semanas de duración) es posible hacer este trabajo en las primeras horas y para proyectos largos (entre doce o más meses) se puede tomar la decisión de invertir dos semanas en este esfuerzo. Se recomienda no invertir más tiempo del indicado, ya que se corre el riesgo de modelar algo que tiene demasiados problemas.

Etapas que intervienen en el Modelado Ágil

A continuación se explicara en detalle en qué consiste cada una de las etapas que intervienen en el modelado ágil.

Contemplar necesidades iniciales: Su objetivo es construir un entendimiento común, no es para escribir documentación detallada. Un factor clave del éxito para lograr contemplar las necesidades iniciales, es utilizar las técnicas de modelado que permitan la participación activa de los interesados.

Para esta etapa es necesario un modelo que permita realizar el análisis de cómo los usuarios trabajan con el sistema, un modelo inicial del dominio en el que se identifiquen los requerimientos fundamentales de la entidad de negocio y las relaciones, y un modelo inicial de interfaz que explore la interfaz de usuario y la usabilidad.

Contemplar la Arquitectura Inicial: El objetivo es tratar de identificar una arquitectura que brinde grandes oportunidades para realizar el desarrollo de la aplicación. Esto es sumamente importante, ya que permite establecer la técnica más viable para llevar a cabo el desarrollo del proyecto y proporciona información suficiente para dividir el equipo de trabajo alrededor de la arquitectura. Por el lado de la arquitectura se suelen crear los diagramas de forma libre, aquellos que representen la infraestructura técnica, los modelos iniciales de dominio para explorar las entidades de negocio principales y sus relaciones.

En iteraciones posteriores los requerimientos iniciales y la arquitectura inicial tendrá que evolucionar a medida que van creciendo las definiciones de más partes, pero por los momentos lo importante es obtener el modelado de una arquitectura inicial que permita al grupo desarrollador ponerse en marcha.

Modelado de las Iteraciones: Al comienzo de la construcción de cada una de las iteraciones para el desarrollo del sistema, el equipo desarrollador deberá estimar y programar que se hará en cada iteración.

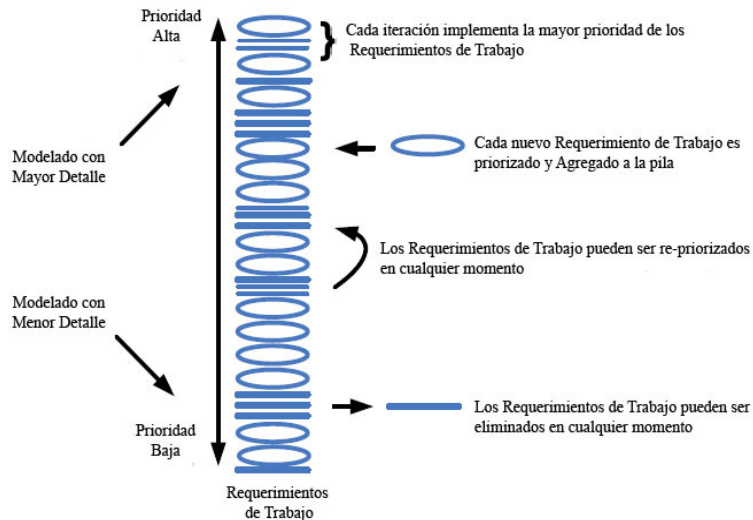


Figura 2. 3: Los Requerimientos Agiles cambian la Gestión del Proceso (Ambler,2002)

En la figura 2.3 se puede observar los diferentes valores que pueden tomar los requerimientos de trabajo en una iteración, el requerimiento de mayor importancia se encuentre en la cima de la pila. Para poder hacer esto con éxito es necesario calcular con exactitud el tiempo de dedicación para cada requerimiento, basándose en la rapidez de las iteraciones previas, cuánto trabajo se pudo lograr en ellas, calcular un tiempo promedio de procesamiento de requerimiento.

Es importante destacar que para poder estimar el tiempo necesario para realizar un requerimiento se debe entender el trabajo necesario para ponerlo en ejecución, es aquí donde entra en juego el modelado.

Tormenta de Modelos: El equipo de trabajo se reúne en tiempos relativamente cortos generalmente no exceden de los 30 minutos, con ayuda de una herramienta de trabajo por ejemplo una pizarra se modela en forma compartida, con el fin de identificar los problemas que se tendrán que resolver, algunos durante este tiempo son identificados y el equipo de trabajo procede a su resolución.

Desarrollo/Pruebas: Durante el desarrollo es bastante común que el modelado de las ideas dure varios minutos y, a continuación se implemente el código de la aplicación esto puede tardar varias horas o incluso varios días, en esta etapa el equipo de trabajo consume el mayor tiempo.

En el modelado ágil los equipos de trabajo tratan de realizar la modelación de sus ideas de la forma más clara y detallada posible, muchas veces esto se realiza junto al cliente para que posteriormente no hayan dificultades o mediante pruebas de desarrollo. Las pruebas conducidas al desarrollo originan pruebas para validar el código de la aplicación y las especificaciones del mismo. Las pruebas del cliente, también llamadas pruebas de aceptación, son realizadas con el fin de mejorar el diseño que se tiene y reforzar los requerimientos.

2.3 Patrón de Diseño MVC (Modelo Vista Controlador)

El patrón de diseño Modelo Vista Controlador o MVC, describe una forma muy utilizada en la Web de organizar el código de una aplicación separando los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos. A continuación se explicara el papel que juega cada uno de estos componentes al momento de desarrollar una aplicación Web (Catalani, 2007).

Modelo: representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

Vista: maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo y muestra los datos al usuario.

Controlador: este responde a eventos generalmente a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista.

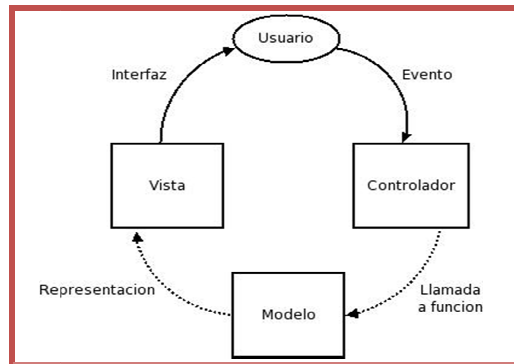


Figura 2. 4: Relación entre los diferentes Componentes de MVC (Catalani, 2007)

Beneficios que ofrece el Patrón MVC

Entre las ventajas que tiene usar un patrón modelo vista controlados podemos destacar:

- Menor acoplamiento: desacopla las vistas de los modelos, desacopla los modelos de la forma en que se muestran e ingresan los datos.
- Mayor cohesión: cada elemento del patrón está altamente especializado en su tarea (la vista en mostrar datos al usuario, el controlador en las entradas y el modelo en su objetivo de negocio).
- Las vistas proveen mayor flexibilidad y agilidad: se puede crear múltiples vistas de un modelo, las vistas pueden anidarse, se puede cambiar el modo en que una vista responde al usuario sin cambiar su representación visual, se puede sincronizar las vistas, las vistas pueden concentrarse en diferentes aspectos del modelo.
- Más claridad de diseño.
- Facilita el mantenimiento.
- Mayor escalabilidad.

2.4 Tecnologías a Utilizar en el Desarrollo de la Aplicación

Tomando como base principal que se está utilizando el modelado ágil como metodología de desarrollo de la aplicación propuesta, es prescindible contar con la tecnología adecuada que nos permita desarrollar la aplicación en el menor tiempo posible, eficientemente y adicionalmente estas tecnologías tienen que ser de código libre, ya que no se cuenta con los recursos para adquirir un software propietario.

Por todo esto y a través del análisis de los requerimientos funcionales que debe proveer la aplicación propuesta, se realizaron pruebas que permitieron determinar que el lenguaje de programación Ruby es una de los más indicados para cumplir con estas necesidades; así como también el manejador de base de datos MySQL el cual tiene un alto grado de compatibilidad con este lenguaje.

2.4.1 Ruby on Rails

Ruby on Rails es un entorno de desarrollo web de código libre (*open source*), que facilita el desarrollo, despliegue y mantenimiento de las aplicaciones Web.

Ruby 1.8.6 es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python, Perl con características de programación orientada a objetos similares a Smalltalk. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre (Cem, 2009).

Entre las características del lenguaje se encuentran (Davila, 2006):

- Posibilidad de hacer llamadas directamente al sistema operativo.
- Muy potente para el manejo de cadenas y expresiones regulares.
- No se necesita declarar las variables.
- La sintaxis es simple y consistente.
- Gestión de memoria automática.
- Todo lo que se puede manipular es un objeto y el resultado de esas manipulaciones, a su vez, son objetos.
- Métodos Singleton.

Rails 2.2.2 es un marco de trabajo (*framework*) para el desarrollo de aplicaciones Web con base de datos, que está basado en el modelo MVC. Desde el Ajax en la vista, a la petición y respuesta en el controlador, hasta el modelo, Rails proporciona un entorno de desarrollo de Ruby.

Principios de Ruby on Rails

Ruby on Rails, se basa en dos principios fundamentales: “*No te Repitas*” y “*Convención sobre Configuración*”.

No te repitas: se refiere a que las definiciones deberían hacerse una sola vez, pues los componentes están integrados de manera que no hace falta establecer puentes entre ellos. Cada pieza de conocimiento en un sistema, deberá ser expresada en un sólo lugar. Este principio se basa en escribir menos líneas de código para implementar la aplicación. Si el código es pequeño quiere decir que el desarrollo es más rápido y con menos errores, lo que hará que el código sea fácil de entender, mantener y mejorar. Por ejemplo, en ActiveRecord, las definiciones de las clases no necesitan especificar los nombres de las columnas; Ruby puede averiguarlos a partir de la propia base de datos, de forma que definirlos tanto en el código como en el programa sería redundante.

Convención sobre configuración: significa que el programador sólo necesita definir aquella configuración que no es convencional. En lugar de archivos de configuración, utilizamos una serie de convenciones simples que permiten averiguarlo todo.

MVC en la arquitectura Rails

La arquitectura Rails utiliza por defecto el patrón de diseño MVC, por lo tanto el framework obliga al desarrollador a mantener la modularidad del código separando los códigos que van en el modelo, la vista y el controlador. Utilizar el entorno de desarrollo ruby on rails también permite establecer una nueva correspondencia entre modelo, vista y controlador; pero ahora con componentes propios de rails, escritos en ruby. Esta correspondencia es la que se muestra a continuación.

Modelo	ActiveRecord
Vista	ActionView
Controlador	ActionController

Tabla 2. 1 MVC en Ruby on Rails

ActiveRecord: Rails gestiona el modelo mediante la clase ActiveRecord que se encarga de mapear nuestras tablas a objetos.

ActionView: En principio, existe una vista por cada acción de cada controlador. Estas vistas son plantillas en formato Embedded Ruby, las cuales definen la presentación que tendrán los datos en los archivos HTML que se entregarán al navegador. El formato Embedded Ruby, es una combinación de código HTML con ruby embebido, formando archivos cuya extensión es .rhtml, los cuales albergan el contenido de cada vista.

ActionController: Un ActionController se encuentra entre un ActiveRecord (interface para la base de datos) y un ActionView (motor de presentación de los datos). El ActionController provee facilidades para la manipulación y organización de los datos que se extraen de la base de datos, y la que viene como entrada en los formularios web; para luego redireccionar a la plantilla de vista correspondiente con los datos resultantes del procesamiento previo.

Ventajas de Ruby on Rails

Entre las ventajas que brinda Ruby on Rails se pueden destacar:

- Ruby on Rails favorece la convención frente a la configuración, es decir, seguir unas normas ya establecidas a la hora de, por ejemplo, llamar a las tablas, campos, archivos, etc; evita (o minimiza) el tener que crear y mantener archivos de configuración y parametrización.
- Es *código libre*, cosa que lo hace accesible a todos los desarrolladores.
- Es multiplataforma.
- Cuenta con soporte para múltiples bases de datos.
- Permite el desarrollo de un código más corto, simple, flexible y mantenible.
- En particular el entorno de desarrollo Rails ofrece módulos, como *Action Mailer* (para enviar correo electrónico), *Action Web Service* para dar soporte a SOAP y XML-RPC, y el muy utilizado últimamente AJAX.

2.4.2 MySQL 5.0

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy

eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (**SQL**) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Características de MySQL

Algunas de las características que presenta este manejador de base de datos son:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas

Tipos de Datos

Entre los tipos de datos soportados por MySQL se pueden destacar:

- Soporte de diferentes tipos de datos en distintas categorías: tipo numérico, fecha, hora y string (enteros con signos de 1, 2, 3, 4 y 8 bytes de longitud, FLOAT, DOBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, TIMESTAMP, YEAR, SET, ENUM).
- Data de Longitud fija y variable.

Ventajas

Las ventajas que presenta MySQL son:

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

Desventajas

Entre las desventajas que podemos encontrar al utilizar este manejador de base de datos se pueden mencionar las siguientes:

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- Sólo se puede establecer integridad referencial si el motor de almacenamiento de las tablas es InnoDB, los cuales son las únicas que lo soportan.

Este gestor se creó con la rapidez en mente, de modo que no tiene muchas de las características de los gestores comerciales más importantes, como Oracle o SQL Server.

CAPÍTULO 3: MARCO DE DESARROLLO

En este capítulo se presenta como fué el proceso de desarrollo de la aplicación adaptado a la metodología seleccionada, el Modelado Ágil. A su vez, se explica el análisis de la solución propuesta donde se expone en qué consiste, con la finalidad de que se puedan comprender mejor los detalles del sistema y como se sustentan las bases teóricas presentadas en el capítulo 1 sobre la aplicación elaborada.

El ciclo de vida de un software elaborado mediante el modelado ágil persigue tres etapas primordiales: iniciación del proyecto, desarrollo de las iteraciones y por último liberación del producto.

En la primera etapa se realizó el análisis de los requerimientos, se contempló la arquitectura inicial y se planificó como se llevará a cabo el desarrollo de la aplicación durante una serie de iteraciones convenidas. En la segunda etapa se describe los hechos realizados durante cada iteración para poder desarrollarla exitosamente. Finalmente, en la tercera etapa se especifica cómo fueron las labores realizadas para colocar el producto final en producción y las diferentes pruebas que se ejecutaron para determinar su correcto funcionamiento.

3.1 Análisis de la Solución Propuesta

Como se mencionó anteriormente el objetivo de este Trabajo Especial de Grado se fundamenta en el desarrollo de una Aplicación Colaborativa vía Web que permita llevar a cabo reuniones electrónicas asíncronas bajo un protocolo de colaboración, específicamente Agendas y Actas.

Durante la presentación del marco conceptual se explicó varios conceptos. En esta sección se expone cómo fueron orientadas dichas definiciones en la aplicación.

Toda Aplicación Colaborativa debe cumplir con cuatro características primordiales: protocolo de colaboración, memoria grupal, percepción de usuarios y roles. A continuación se presenta cómo se orientó dichas características en la aplicación desarrollada.

Protocolo de Colaboración basado en Agendas y Actas:

Al hacer referencia de las reuniones en el marco conceptual se señaló que es fundamental tener coordinación de la reunión que se desea llevar a cabo, ya que esto facilitará el alcance de los objetivos propuestos, por tener previamente establecidos y claros los pasos a seguir al planificarla y desarrollarse.

En consecuencia, se creó el protocolo de colaboración Agendas y Actas el cual servirá de guía a los participantes y el coordinador a la hora de crear y ejecutar una reunión electrónica en la aplicación.

El modelo conceptual de una reunión electrónica bajo el protocolo de colaboración de Agendas y Actas en la aplicación, se basa en tres procesos fundamentales que se explican y presentan a continuación en la figura 3.1:

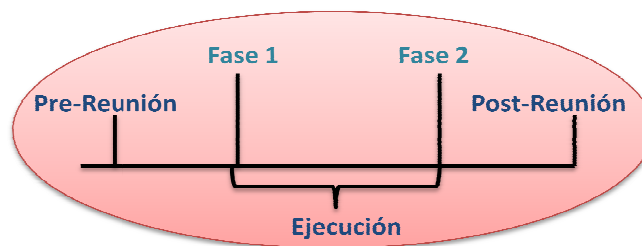


Figura 3. 1: Ciclo de Vida de una Reunión Asíncrona

Pre Reunión: durante este proceso el coordinador de la reunión realiza la planificación de las características que tendrá la reunión que desea ejecutar. Especifica su fecha de inicio y culminación, el tipo de reunión pública o privada (pública cualquier usuario de la aplicación puede ver la agenda, privada solo los participantes y el coordinador pueden ver la agenda), selecciona el comité de trabajo con el cuál desarrollará la reunión, los detalles de la agenda y finalmente determina la duración de cada fase de la reunión. En la figura 3.3 se puede observar el modelo de actividades del proceso Pre-Reunión explicado.

Ejecución de la Reunión: la ejecución de una reunión en la aplicación está representada por medio de dos fases. Por ser de tipo asíncrona la forma de llevarse a cabo una reunión, la duración de cada una de las fases en el sistema será en intervalos de días, especificados por el coordinador al momento de crear la reunión.

- **Fase 1 Orden del Día:** en esta fase los participantes de la reunión pueden observar la agenda que creó el coordinador durante la pre-reunión y aportar sugerencias sobre los puntos que la conforman. Por otra parte el coordinador decide en función de las sugerencias aportadas por los participantes si editar o incluir nuevos puntos a la agenda. Al finalizar se tiene constituida la agenda definitiva que será discutida en la siguiente fase.
- **Fase 2 Discusión de la Agenda:** en esta fase la agenda definitiva de discusión ya se encuentra establecida, los participantes de la reunión pueden proceder a realizar comentarios sobre los puntos no informativos de la agenda, sino están en estado de votación o listo.

El coordinador de la reunión decide cuando cambiar el estado de un punto a votación o listo. Si el punto cambia a estado de votación el coordinador ingresa una resolución final sobre ese punto y los participantes de la reunión pueden aprobar, rechazar o emitir voto nulo sobre él. En caso contrario si el punto está en estado de listo solo se podrán observar los comentarios y la resolución final del punto.

En la figura 3.4 se presenta el modelo de actividades del proceso Ejecución de la Reunión descrito.

Post Reunión: Al concluir la fase 2 de la reunión el sistema genera automáticamente el acta de la reunión, donde se exponen los diferentes comentarios que suministraron cada uno de los participantes a los puntos que conforman la agenda; así como también se presenta la resolución final del punto y las estadísticas de votación. En la figura 3.5 se puede observar el modelo de actividades del proceso Post-Reunión explicado.

En función de lo señalado anteriormente se procedió a realizar diagramas de actividades mediante los cuales es posible representar gráficamente los pasos que se llevan a cabo durante la ejecución de un proceso en una reunión electrónica bajo el protocolo de colaboración de Agendas y Actas y los actores que intervienen en él.

En la figura 3.2, se presenta el modelo de actividades de los procesos y la comunicación entre cada uno de ellos.

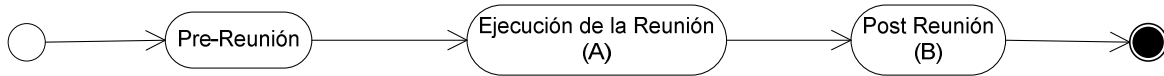


Figura 3. 2 Modelo de Actividades. Ciclo de Vida de una Reunión Electrónica.

Refinamiento de cada uno de los procesos

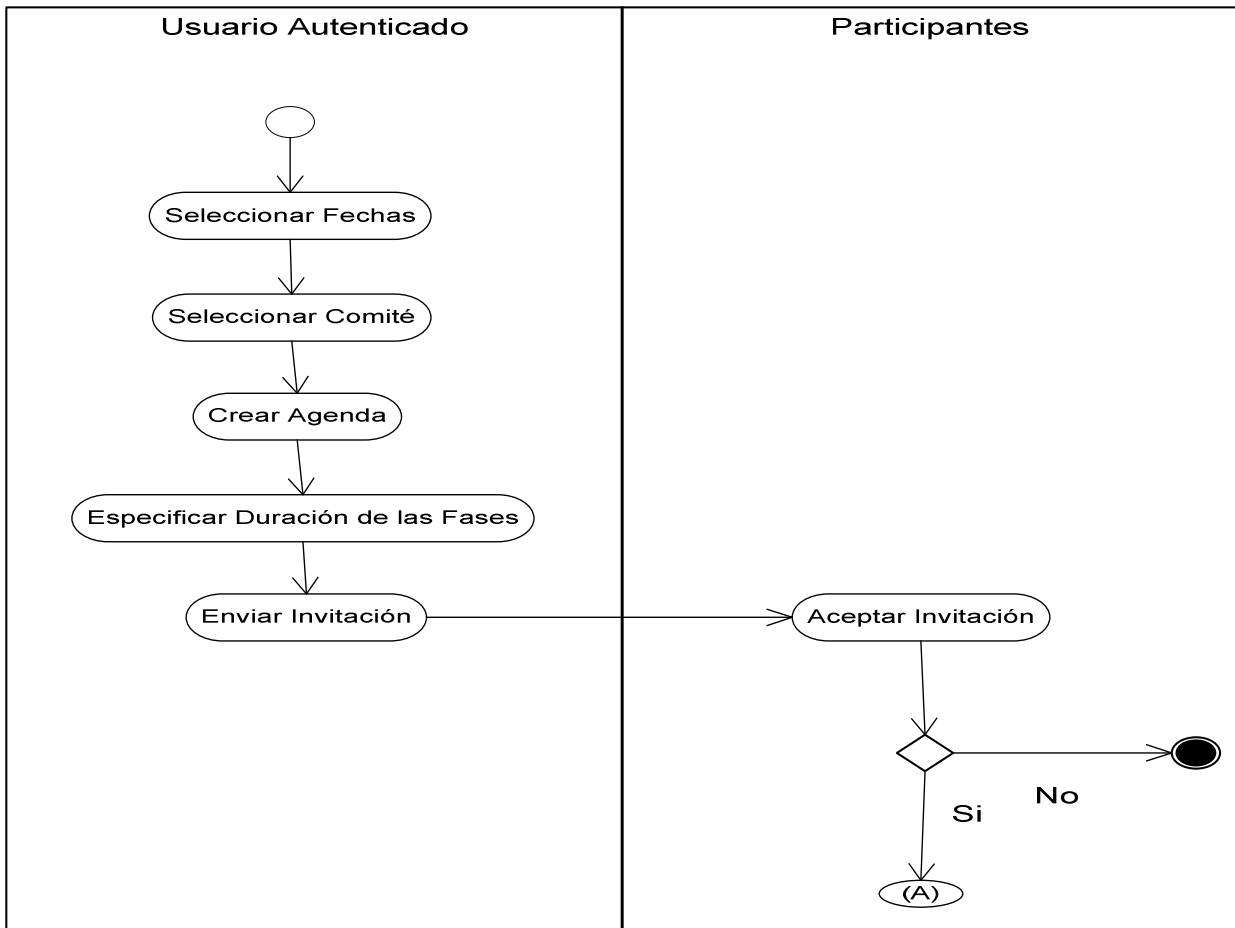


Figura 3. 3 Modelo de Actividades. Proceso Pre-Reunión.

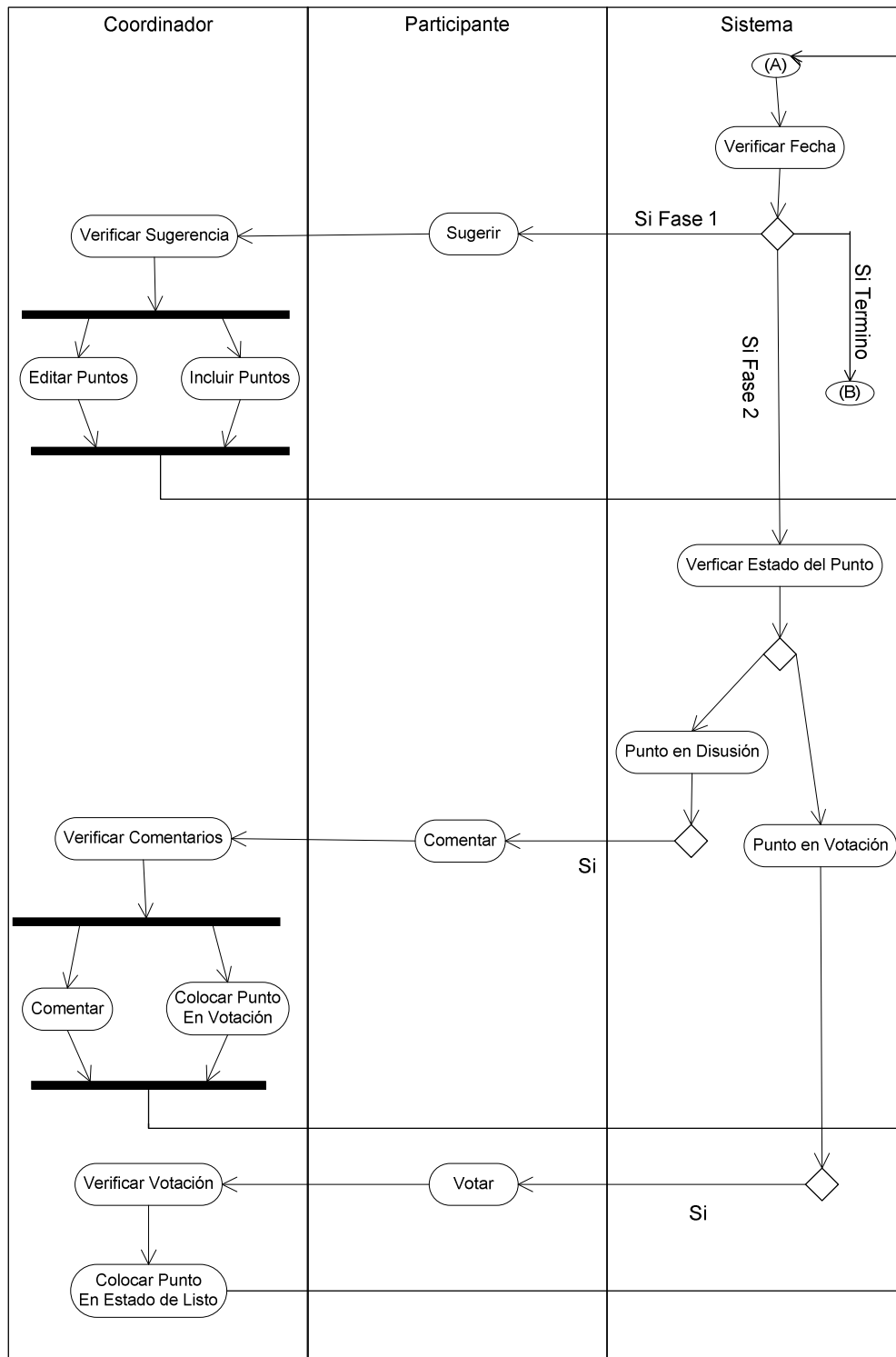


Figura 3. 4 Modelo de Actividades. Proceso Ejecución de la Reunión.

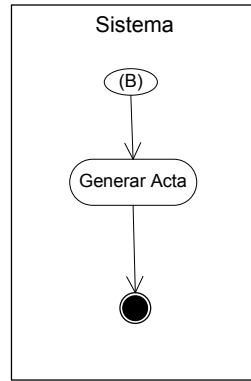


Figura 3. 5 Modelo de Actividades. Proceso Post Reunión.

La forma en que fue representada las características restantes en la aplicación: roles, percepción de usuarios y memoria grupal, serán descritos más adelante a medida que se explica el proceso de desarrollo del software.

3.2 Iteración 0: Iniciación del Proyecto

En la iteración 0, se realizó un análisis de los requerimientos del sistema donde se permitió determinar las funcionalidades que la aplicación provee, los actores participantes y las diferentes relaciones entre los mismos. De igual forma se determinó la arquitectura inicial la cual sigue el esquema de una aplicación Web Cliente-Servidor, desarrollado bajo el patrón de diseño MVC de Ruby on Rails.

Por último, se destacó la Pila de Requerimientos que contiene la distribución de cómo se desarrolló cada modulo funcional de la aplicación y en que iteración, la descripción de la interfaz; a través de los patrones de interacción. En la figura 3.6 se detallan las acciones a tomadas en la iteración 0.

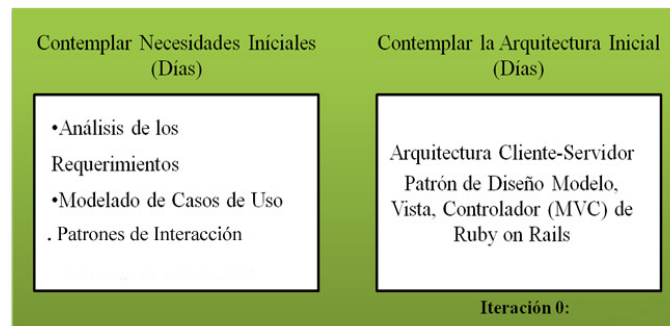


Figura 3. 6 Iteración 0 del proyecto

Requerimientos Funcionales de la Aplicación

Los requerimientos funcionales son todos aquellos que describen las distintas funcionalidades que se espera el sistema provea y las interacciones del mismo con su entorno.

Una forma de representar estos requerimientos es a través de la realización del modelo de casos de uso de la aplicación ya que permiten describir las funcionalidades que debe proveer el sistema. En la figura 3.7 se muestra el modelo de los casos de uso realizados y posteriormente la descripción de cada uno.

Como la iteración que se está presentando es la cero, los diagramas de casos de uso están hechos hasta un nivel uno de detalle. En iteraciones más avanzadas se llevó a cabo el refinamiento de estos casos de uso y fueron explotados a un nivel siguiente.

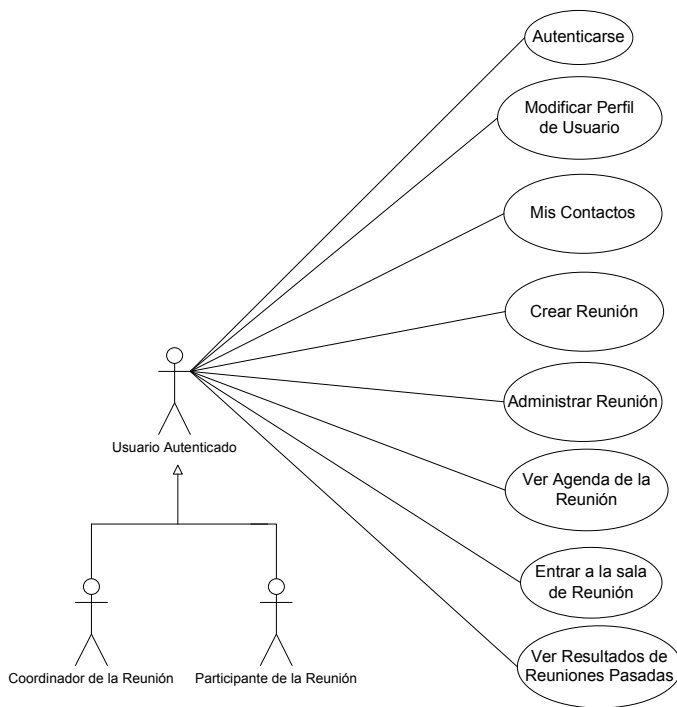


Figura 3. 7 Nivel 1 de los Casos de Uso, Usuario Autenticado

Descripción de los Casos de Uso

A continuación se describe cada uno de los casos de uso modelados en esta iteración; así como también los actores que intervienen:

Actores

Al describir los actores que están presente en la aplicación se cumple con otras de las características de las aplicaciones colaborativas, en este caso la de Roles. Cada uno de estos actores personifican los diferentes roles que puede tomar un usuario en el sistema y sus privilegios.

- **Usuario:** es aquella persona que ha ingresado al sistema pero aún no ha iniciado sesión, esto puede ser por varias causas: no está registrado en el sistema, perdió su contraseña o simplemente no se ha autenticado. Tiene acceso restringido a ciertas funcionalidades de la aplicación.
- **Usuario Autenticado:** es aquel que posee un registro en el sistema, ha iniciado sesión exitosamente y puede proceder a crear una reunión, lo que lo convierte automáticamente en coordinador de esa reunión.
- **Participante de la Reunión:** es un invitado a la reunión, el mismo puede participar en una reunión programada y aportar sus opiniones. Tiene que estar registrado en la aplicación y haber iniciado sesión exitosamente.
- **Coordinador de la Reunión:** se encarga de mantener el control de una reunión, selecciona los participantes que cree conveniente, toma las decisiones finales, puede configurar la reunión y la agenda.

Descripción de los Requerimientos Funcionales

A continuación se presenta una breve descripción de cada uno de los casos de usos presentados en la figura 3.7. Si desea ver una descripción mucho más detalla de cada uno, consulte los anexos del Trabajo investigativo.

Nombre	Descripción
RF- 1 Autenticarse	Permitir a un usuario poder llevar a cabo las funcionalidades básicas de toda aplicación, como lo son: el registro de usuario, inicio de sesión, recuperar contraseña o cambio de contraseña.
RF- 2 Crear Reunión	Permitir a un usuario autenticado en la aplicación crear una reunión.
RF- 3 Ver Resultados de Reuniones Pasadas	Los usuarios de la aplicación estén enterados de las decisiones tomadas en reuniones que se llevaron a cabo, donde participaron y puedan ver el acta generada.
RF- 4 Modificar Perfil de Usuario	Permitir a un usuario autenticado actualizar los datos personales que ingresó al momento de su registro.
RF-5 Mis Contactos	Permite a un Usuario Autenticado ver sus contactos, eliminar algún contacto de su lista o agregar un nuevo contacto.
RF- 6 Administrar Reunión	Permite al coordinador de la reunión modificar, eliminar, o ver una reunión. El participante de la reunión solo puede ver los detalles de la reunión.
RF- 7 Entrar a la Sala de Reunión	Los usuarios autorizados (Participantes invitados y coordinador) pueden ingresar a la reunión programada en esa fecha.
RF- 8 Ver Agenda de la Reunión	Los usuarios de la aplicación puedan ver las agendas de las reuniones públicas o privadas programadas.

Tabla 3. 1 Descripción breve de los casos de uso

Requerimientos No Funcionales de la Aplicación

Los requerimientos no funcionales del sistema están relacionados con todos aquellos que permiten juzgar la calidad del sistema; así como también permiten determinar si las funcionalidades del sistema, se cumplen de manera eficaz y eficientemente. Como una de las prioridades del sistema en desarrollo es promover el trabajo en grupo de una manera organizada y la toma de decisiones, estos requerimientos deben ayudar a garantizar dichos objetivos. A continuación se presenta la descripción de aquellos requerimientos no funcionales que se tomaron en cuenta en la aplicación colaborativa elaborada.

Requerimiento	Descripción
RNF-1 Seguridad	El sistema debe garantizar que los datos suministrados por los usuarios no sufran discrepancia alguna si hay varios usuarios modificando al mismo tiempo un espacio de trabajo. Debe permitir una alta concurrencia. Así como también el manejo de sesión, para poder verificar si el usuario que accede a alguna funcionalidad de la aplicación tiene los permisos necesarios. Los participantes de una reunión solo podrán ingresar a la sala de reuniones, ver las actas y las agendas de aquellas reuniones en las que haya aceptado la invitación.
RNF-2 Confiabilidad	El sistema debe comportarse de manera adecuada brindando lo que se espera de él y en el tiempo correcto, de esta forma se evita que los usuarios puedan atemorizarse ante el comportamiento excepcional del sistema
RNF-3 Robustez	El sistema tiene que estar en la facultad de emitir alguna respuesta ante situaciones que no hayan sido anticipadas en los requerimientos
RNF-4 Usabilidad	El sistema debe ser lo más sencillo de usar con metáforas adecuadas y fáciles de predecir su funcionalidad poco propenso a generar errores, el diseño de la interfaz debe poseer los colores, letras y tamaños más adecuados.
RNF-5 Integridad	Garantizar que los datos suministrados por los usuarios de la aplicación estén correctos en todo momento, para poder lograr esto hay que realizar validaciones tanto del lado del cliente como del lado del servidor.
RNF-6 Rendimiento	Garantizar una velocidad estable de navegación para los usuarios del sistema. Optimizar los algoritmos en la aplicación para que los tiempos de respuestas sean los más eficiente posible.

Tabla 3. 2 Requerimientos No Funcionales de la Aplicación

Pila de Requerimientos

Siguiendo los principios del modelado ágil y las diferentes etapas que intervienen en él, el desarrollo del Trabajo Especial de Grado se realizó mediante iteraciones previamente establecidas y analizadas. Para poder determinar la cantidad de iteraciones, se procedió a la captura de los requerimientos funcionales y no funcionales del sistema.

Los requerimientos fueron colocados en la pila por orden de prioridades. El tiempo que llevó el desarrollo de cada uno de los requerimientos dependió de la extensión de los mismos; así como también se realizó un análisis previo que permitió definir cómo se

ejecutarían cada uno de los requerimientos. Este análisis consistió básicamente en determinar la comunicación que debe existir entre cada modulo que compone el sistema y sus necesidades. A continuación en la figura: 3.8 se presenta la pila de requerimientos implementada:

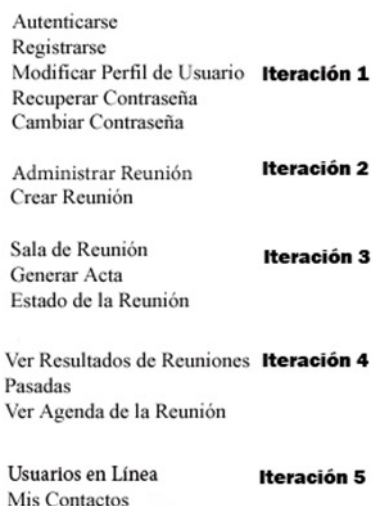


Figura 3. 8 Pila de Requerimientos de la Aplicación

Interfaz de la Aplicación

Una vez que se ha efectuado el análisis de los requerimientos funcionales y no funcionales de la aplicación, se procedió a realizar el modelo de interfaz de la aplicación.

Para realizar este modelado, se tomó en cuenta que la aplicación a desarrollar es colaborativa lo que es de vital importancia que el diseño de la interfaz incentive a los usuarios a interactuar con otros usuarios. Por otro lado la interfaz debe poseer metáforas conocidas por el usuario que faciliten la navegabilidad y la interacción con el sistema, los componentes tienen que estar ubicados en un espacio acorde, tomar en cuenta atributos como: el color, el tamaño y la forma.

Partiendo del hecho que la aplicación colaborativa persigue como objetivo principal servir de apoyo para realizar reuniones electrónicas asíncronas, las herramientas que provee al usuario de la aplicación deben permitir: escribir sus opiniones sobre algún tema que se esté discutiendo, esto se puede lograr mediante la simulación de foros de discusión, almacenar,

editar documentos como por ejemplo: la agenda de una reunión programada, enviar información por medio del correo electrónico, entre otras. Estas herramientas tienen que estar sustentadas a través de la interfaz elaborada.

Otro de los aspectos que se toma en consideración, son los diferentes actores que interactúan en la aplicación (Usuario, Usuario Autenticado, Coordinador de la Reunión, Participante de la Reunión), con la finalidad de crear interfaces que satisfagan las necesidades que cada uno presenta. En el caso de un Usuario Autenticado y el Coordinador de la Reunión deben poseer una interfaz única, donde administren las reuniones y la agenda. El Usuario y los Participantes de la Reunión requieren de una interfaz usable que facilite su navegación en el sistema.

Para poder lograr todo esto, se utilizaron los patrones de interacción que se definen como: describe una solución exitosa a un problema recurrente concerniente a la interfaz de usuario, en un contexto dado.

Según (Erickson, 2001) un patrón de interacción es un medio de comunicación que expresa una solución de interfaz en una forma textual y gráfica, a fin que sea entendida por todas las personas involucradas en el diseño, las cuales, generalmente, tienen diferentes formaciones profesionales.

Los patrones de interacción capturan la experiencia y conocimientos de expertos, quienes han producido soluciones exitosas a problemas; sin embargo no siempre proveen la solución definitiva, por lo que los patrones utilizados fueron adaptados de acuerdo a las necesidades que presentó la aplicación. Es importante destacar que para efectos del desarrollo del sistema se tomaron en cuenta los patrones de interacción propuestos por (Welie, 2008), que denotan interfaces exitosas que mejorará la usabilidad de la aplicación.

En la literatura se encuentran diversas formas de describir un patrón de interacción (Borchers, 2000), las cuales coinciden en algunos componentes básicos. Para efecto de la descripción de los patrones utilizados en la aplicación, se tomó en cuenta aquellos componentes obligatorios como: nombre, problema, solución y contexto. En la tabla 3.3 se presenta el significado que tienen los componentes seleccionados.

Componente	Descripción
Nombre	Comunica la idea central del patrón
Problema	Describe la interacción desde el punto de vista del usuario
Solución	Describe, en forma detallada y gráfica, la solución del problema
Contexto	Presenta las condiciones bajo las cuales se usa este patrón

Tabla 3. 3 Componentes para describir los patrones.

Taxonomía de los Patrones de interacción

Notación	Categoría
	Patrones de dominio
	Patrones de sistema
	Patrones de tareas
	Patrones de elementos compuestos
	Patrones de elementos simples
	Patrones de usuario

Tabla 3. 4 Taxonomía de los Patrones de Interacción (Tidwell, 1999).

A partir de lo anteriormente descrito y tomando como base los patrones de interacción planteados, se realizó el lenguaje de patrones de la aplicación que puede ser apreciado en la figura: 3.9, donde se describe diferentes patrones de tareas que representan las características que cumple la interfaz de la aplicación en funcionalidades específicas, los cuales fueron instanciados al momento de explicar la interfaz en cada una de las iteraciones desarrolladas.

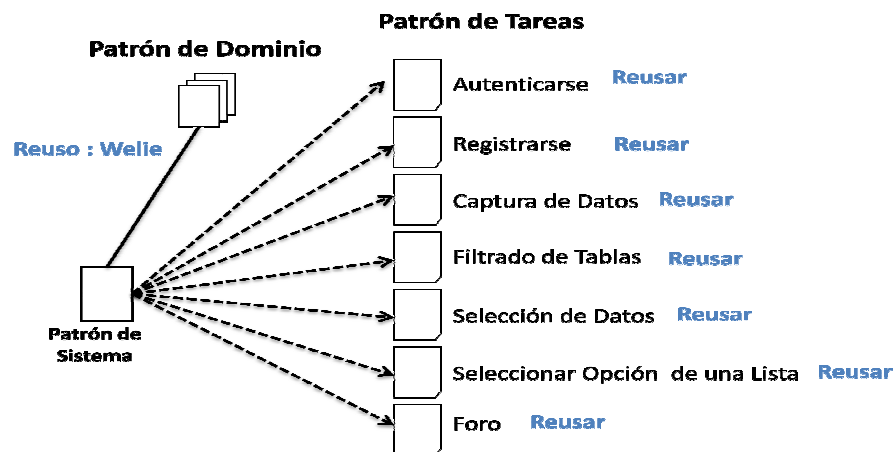


Figura 3. 9 Lenguaje de Patrones de la aplicación.

En la figura 3.9 al destacar el término Reusar quiere decir que se trabajó tomando como punto de referencia los patrones de diseño de interacción propuestos por (Welie, 2008).

Descripción de los Patrones

Patrón del Usuario

Nombre	Usuario Internet 😊
Problema	Caracterizar al usuario de la Aplicación Colaborativa
Solución	El usuario maneja el uso del hipertexto, hipervínculos, tiene cierto dominio de internet. Su objetivo es poder programar una reunión electrónica o participar en ella, en caso de que esté invitado. Estas reuniones deben servir de herramientas para el trabajo en grupo y la toma de decisiones.
Contexto	Aplicación Colaborativa para reuniones electrónicas asíncronas coordinadas.

Patrón del Sistema

Nombre	Areunirse
Problema	El usuario necesita una aplicación que facilite el proceso de llevar a cabo una reunión electrónica asíncrona.
Solución	<p>Diseño de una página donde dependiendo del rol del usuario se habilite un menú con funcionalidades de acuerdo a su cargo y que contenga las características que debe poseer toda aplicación colaborativa.</p> <p>Entre estas características se destaca la percepción de usuarios a través de un panel ubicado en la interfaz de la aplicación donde se permite visualizar los usuarios conectados actualmente.</p> <p>Los elementos estarán ubicados en un lugar correcto y de fácil acceso.</p>

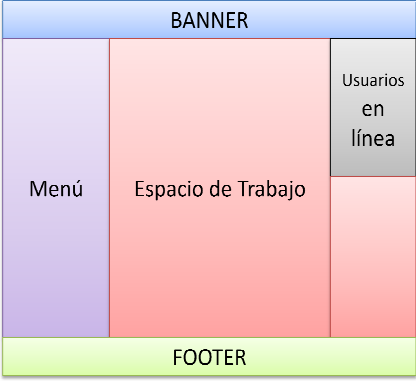
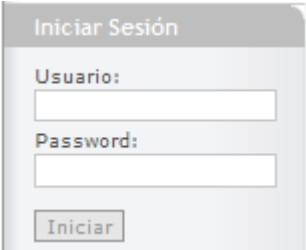
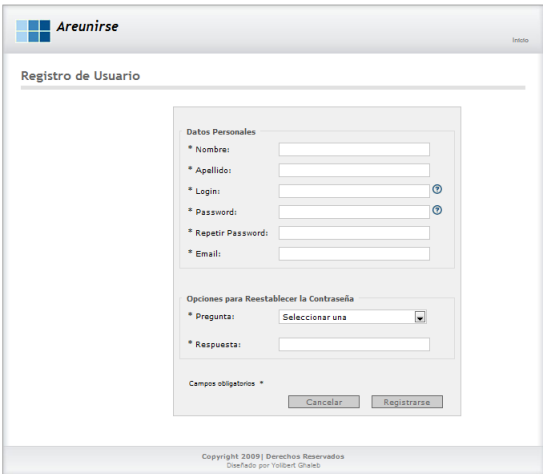
	
Contexto	Aplicación Colaborativa para reuniones electrónicas asíncronas



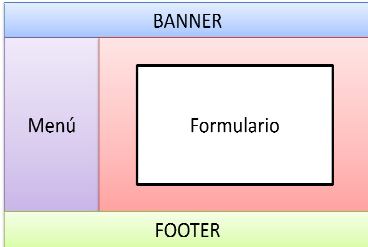
Figura 3. 10 Patrón del Sistema

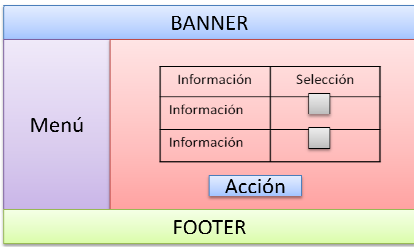
Patrón de Tareas

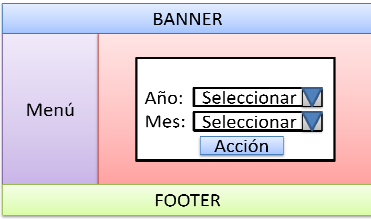
Nombre	Autenticarse
Problema	Los usuarios necesitan autenticarse para poder utilizar las funcionalidades que requieren de privilegios en la aplicación.
Solución	Cuando es necesario, que los usuarios inicien sesión a través de la combinación de un nombre de usuario y contraseña. 
Contexto	El usuario se ha interesado por ingresar a la aplicación y decide iniciar sesión.

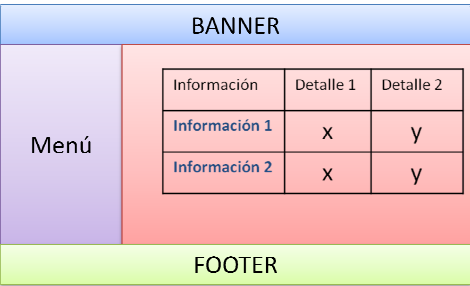
Nombre	Registrarse
Problema	Es necesario contar con un módulo que permita la recolección de los datos personales del usuario, para que puedan utilizar las funcionalidades de la aplicación.
Solución	Ofrecen a los usuarios la posibilidad de almacenar su información personal para su uso posterior. 
Contexto	El usuario desea ingresar a la aplicación y poder utilizar sus

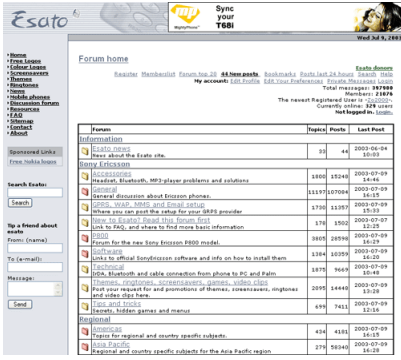
	funcionalidades, pero es precisa su autenticación antes.
--	--

Nombre	Captura de Datos
Problema	Los usuarios tienen que proveer información específica y personal, que será utilizada en la aplicación para su correcto funcionamiento.
Solución	Ofrece a los usuarios un formulario con los elementos necesarios. <div style="text-align: center;">  </div>
Contexto	El usuario desea llevar a cabo funcionalidades que requieren de la captura de datos, los cuales serán almacenados en la base de datos para su procesamiento y uso en acciones posteriores.

Nombre	Filtrado de Tablas
Problema	Los usuarios necesitan tener el control de la información que se presenta en la tabla y poder realizar con ella la acción que deseen.
Solución	Permitir a los usuarios seleccionar un subconjunto de los elementos de información directamente sobre la tabla. <div style="text-align: center;">  </div>
Contexto	El usuario ingresa a funcionalidades en la aplicación donde la información se presenta en tablas y el procesamiento de ellas requiere de su selección.

Nombre	Selección de Datos
Problema	Los usuarios necesitan consultar cierta información y para esto es necesario ingresar datos de una manera adecuada que garanticen el éxito de la búsqueda.
Solución	Solo permiten que el usuario introduzca los datos, en la sintaxis correcta. 
Contexto	El usuario ingresa a funcionalidades en la aplicación donde para obtener los resultados sobre la búsqueda en una serie de datos es necesario el ingreso exacto de la información.

Nombre	Seleccionar Opción de una Lista
Problema	Los usuarios necesitan gestionar una colección de resultados producto de alguna consulta realizada.
Solución	Crear una visión general de los objetos que juntos es significativa para los usuarios. 
Contexto	El usuario ingresa a funcionalidades en la aplicación donde aparecen los resultados en tablas que contienen una descripción breve y a partir de su selección pueden verlo con más detalles y llevar a cabo acciones sobre ella.

Nombre	Foro
Problema	Los usuarios requieren discutir u opinar sobre los diferentes puntos de la agenda.
Solución	<p>Crear una lista de temas y permitir a los usuarios colocar comentarios relacionados al tema.</p> 
Contexto	El usuario ingresa a la sala de reuniones, puede aportar sugerencias y comentarios sobre los diferentes puntos que integran la agenda.

Arquitectura de Ruby on Rails adaptada a la Aplicación Colaborativa

Como se mencionó en el capítulo del marco metodológico la arquitectura de Ruby on Rails es MVC puro, lo que obliga a los desarrolladores a trabajar de una manera más organizada separando los códigos en los modelos, las vistas y los controladores. En la figura 3.11, se refleja el cuerpo del Trabajo Especial de Grado siguiendo el patrón MVC, en el cual se encuentran una serie de directorios con funcionalidades específicas.

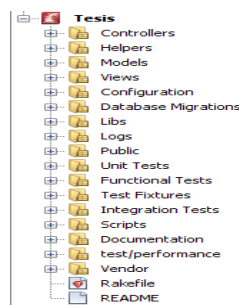


Figura 3. 11 Arquitectura MCV en Ruby on Rails de la Tesis

Descripción de los diferentes directorios del Proyecto

- **app:** almacena los componentes de la aplicación: vistas y helpers, controladores y modelos.
- **config:** contiene la configuración de la aplicación de la base de datos (*database.yml*), estructura del entorno de Rails (*environment.rb*) y el enrutador de peticiones Web (*routes.rb*).
- **public:** es similar al directorio public de un servidor Web. Contiene los archivos con código de funciones escritas en JavaScript, imágenes, hojas de estilo y código HTML común para las interfaces.
- **script:** este directorio contiene scripts que inician y administran las diversas herramientas que se pueden utilizar en Rails; por ejemplo, se tienen los scripts que generan código (*generate*) e inician un servidor Web (*server*).

Es importante destacar que en la descripción de la iteración cero aún no se cuentan con controladores, vistas y modelos, ya que en esta parte del desarrollo de la aplicación solo se hace un análisis que nos permita tener un bosquejo inicial del sistema.

3.3 Desarrollo de las Iteraciones

Como se pudo observar en la Pila de Requerimientos realizada el desarrollo de los módulos que conforman la aplicación fue dividido en una serie de iteraciones. A continuación se presenta la descripción de cada una de ellas.

Iteración 1

En esta iteración se realizaron aquellas funcionalidades básicas que debe proveer una aplicación Web, en donde los usuarios que interactúan con el sistema tienen la necesidad de autenticarse para poder realizar ciertas acciones que requieren tener privilegios.

Entre las funcionalidades elaboradas se destacan: registro de usuario, iniciar sesión, modificar perfil de usuario, recuperar contraseña y cambiar contraseña. En la figura 3.12 se presenta el caso de uso autenticarse a un nivel 2 de detalle.

Refinamiento de las Funcionalidades

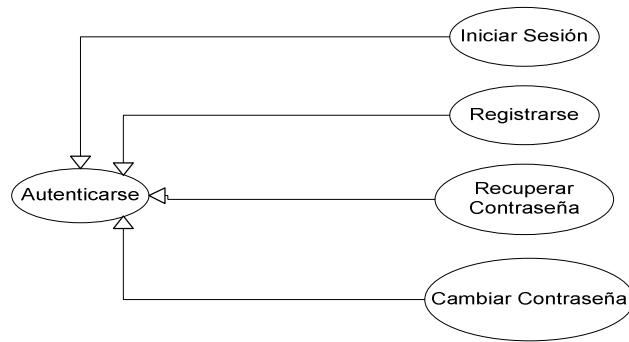


Figura 3. 12 Modelado del RF-1

Descripción de la Interfaz

Se instancian los patrones de tareas: autenticarse, registrarse y captura de datos para los módulos de recuperar contraseña, modificar perfil de usuario y cambiar contraseña descritos durante el desarrollo de la iteración 0.

A continuación se pueden visualizar las diferentes interfaces que se generaron para esta primera iteración.



Figura 3. 13 Modelo de Interfaz: Recuperar Contraseña

Figura 3. 14 Modelo de Interfaz: Modificar Perfil

Figura 3. 15 Modelo de Interfaz: Cambiar Contraseña

Iteración 2

En esta iteración se llevó a cabo el desarrollo del módulo Administrar y Crear Reunión. Para poder realizarlo fue necesario efectuar un análisis, mediante el cual, se permitiera determinar cuáles son las diferentes facetas que presenta una reunión de tipo Agendas y

Actas, sus restricciones y los datos que son necesarios recopilar para su adecuada ejecución al momento de habilitarse la reunión en la aplicación.

Al momento de desarrollar Crear Reunión, se realizó la instalación de la gema tsmail (0.0.1) que permite el envío de correo electrónico a los invitados. Es importante destacar que la aplicación está trabajando internamente con el proveedor de servicio de correo electrónico de Gmail, quien es el encargado de enviar las invitaciones o notificaciones a los participantes e invitados de una reunión por medio de la cuenta Areunirse@gmail.com.

Adicionalmente, por pruebas que fueron realizadas se pudo determinar que algunos proveedores de correo electrónico, rechazan los correos enviados a través de la aplicación por ejemplo: Yahoo. Esto debido a las políticas de seguridad que manejan, al desconocer la fuente es visto como un correo malicioso y no lo aceptan. Sin embargo para proveedores como Gmail y Hotmail esto no ocurre.

Al principio el proveedor de ciens también actuaba de igual forma que yahoo, pero una vez que fue colocada la aplicación en producción en un servidor dentro de la universidad esto dejó de ocurrir, ya que se conocía la fuente y la ip era válida para las políticas de seguridad que el maneja. En la figura 3.16 se presenta el caso de uso administrar reunión a un nivel 2 de detalle.

Refinamiento de las Funcionalidades

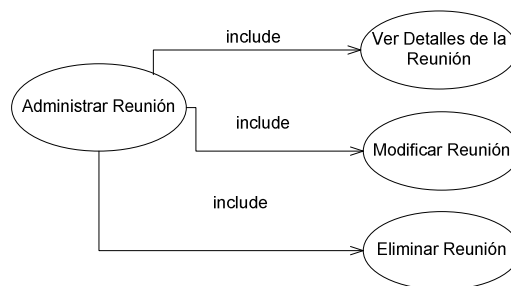


Figura 3. 16 Modelado del RF-6

Descripción del Modelado del RF-6

Nombre	Descripción
RF-6.1 Ver Detalles de la Reunión	El participante o coordinador de la reunión, podrán ver las características de las reuniones donde están involucrados.
RF- 6.2 Modificar Reunión	El coordinador de la reunión puede proceder a modificar los datos de la reunión, en caso de que no haya iniciado, concluido o se encuentre en estado de cancelada.
RF- 6.3 Eliminar Reunión	El coordinador de la reunión puede proceder a eliminar una reunión por completo de la aplicación.

Tabla 3. 5 Refinamiento del requerimiento funcional 6. Descripción breve de del casos de uso.

Descripción de la Interfaz

Se instancia los patrones de tareas descritos en la iteración 0:

- Captura de datos: utilizado en el momento de crear una reunión, modificar reunión específicamente el tiempo y la agenda.
- Seleccionar opción de una lista: para modificar o eliminar reunión.
- Filtrado de tablas: para modificar reunión específicamente los participantes.

A continuación se presentan las diferentes interfaces generadas en está iteración:

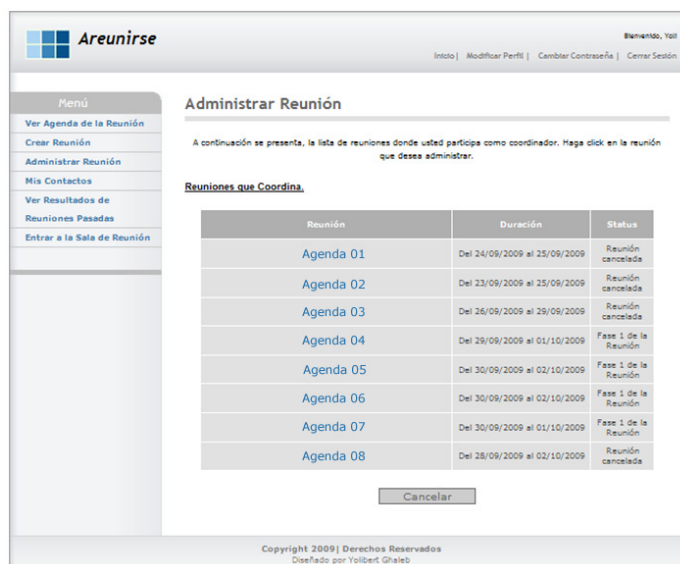


Figura 3. 17 Modelo de Interfaz: Administrar Reunión

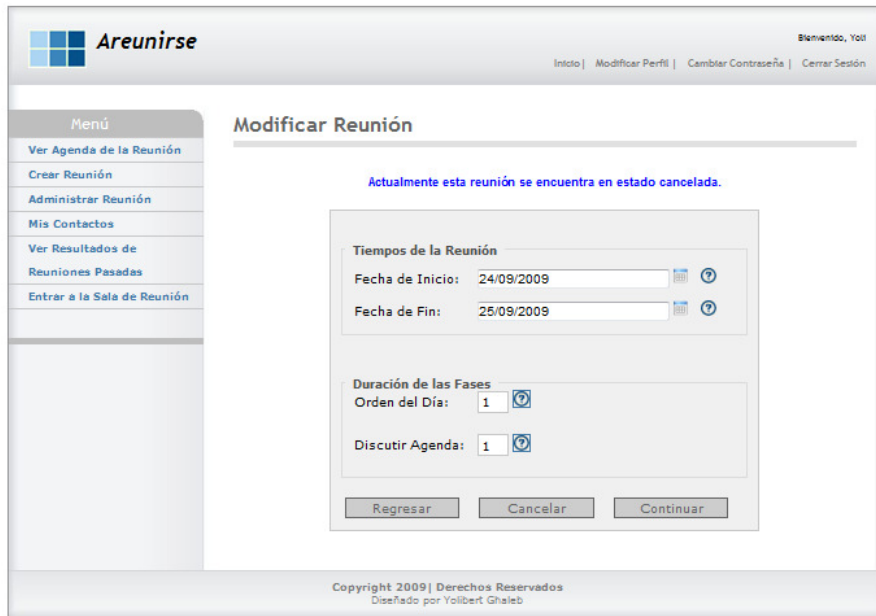


Figura 3. 18 Modelo de Interfaz: Modificar Reunión Tiempo

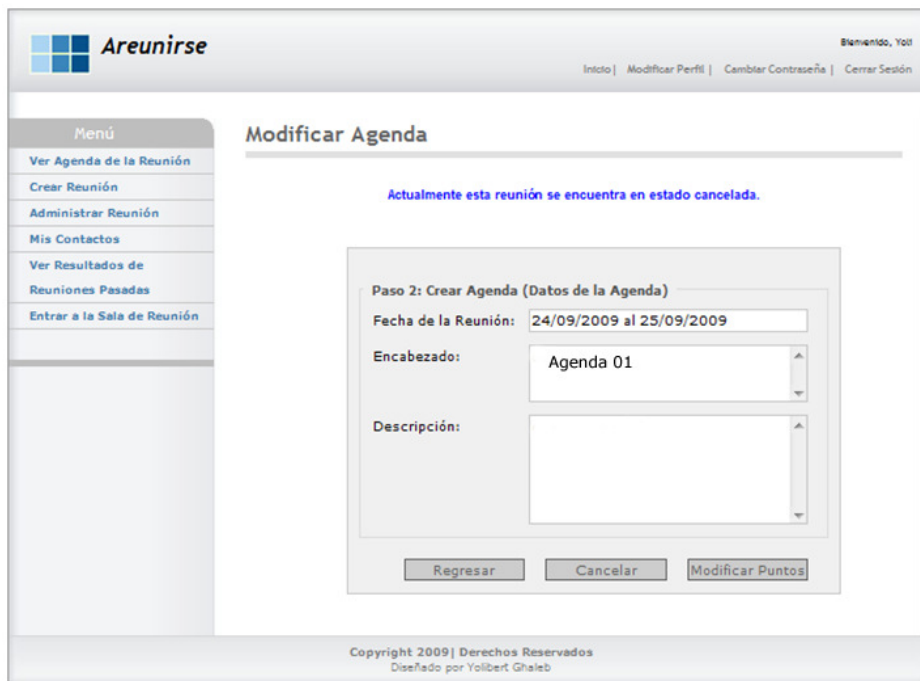


Figura 3. 19 Modelo de Interfaz: Modificar Reunión Agenda

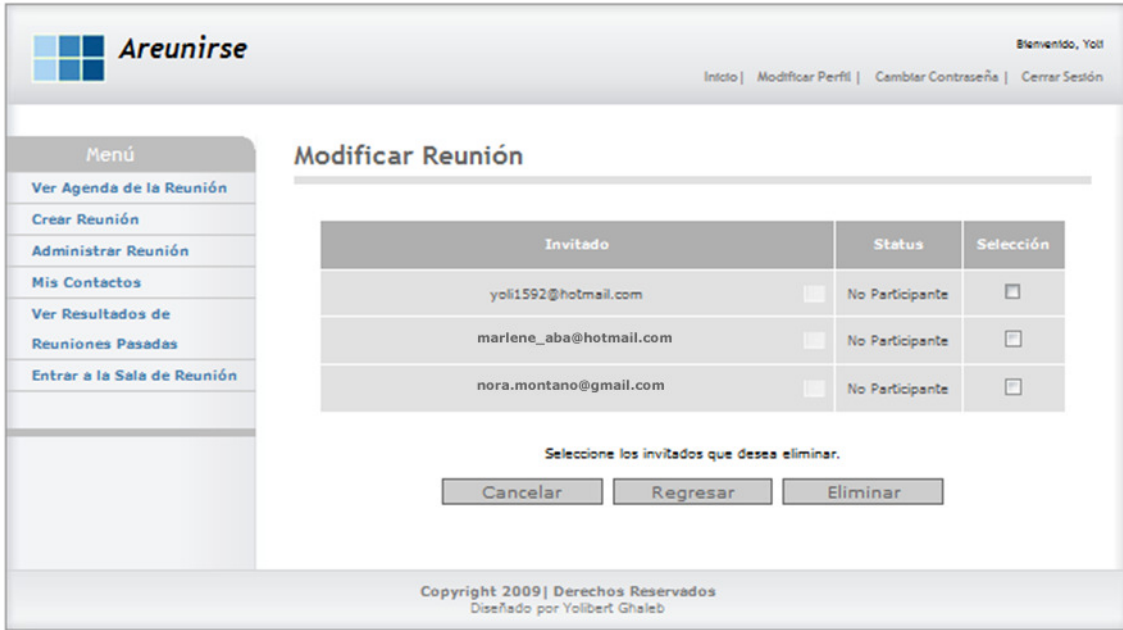


Figura 3. 20 Modelo de Interfaz: Modificar Reunión Participantes

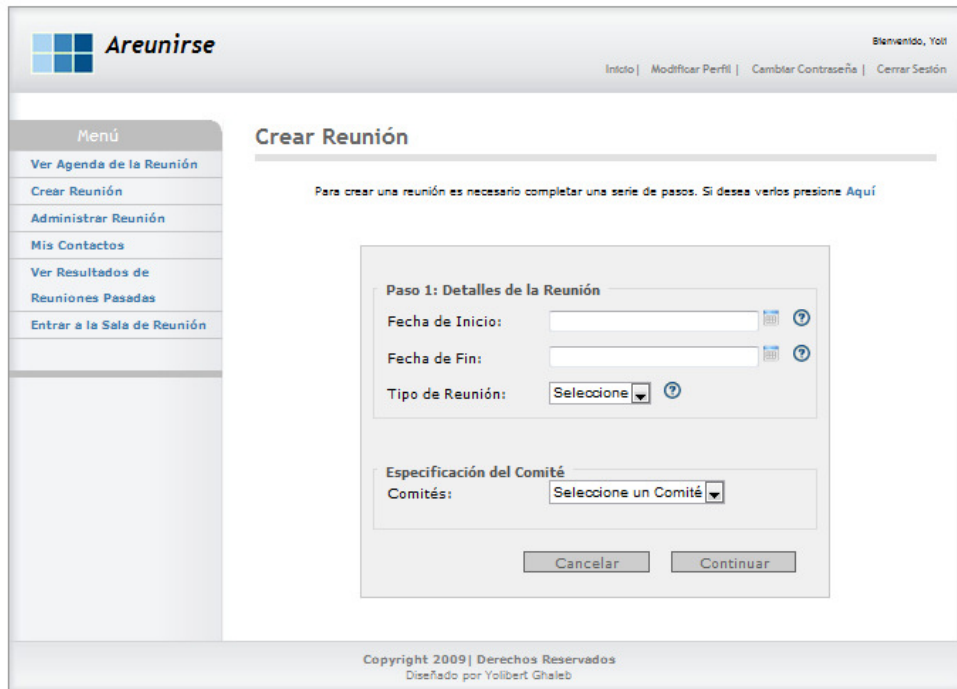


Figura 3. 21 Modelo de Interfaz: Crear Reunión

Iteración 3

En esta iteración se desarrolló la Sala de Reuniones. Mediante el análisis realizado en el apartado 3.1, donde se realizó diagramas de actividades y se describe el modelo conceptual de la reunión fue posible precisar el comportamiento de una reunión electrónica bajo el protocolo de colaboración de Agendas y Actas, lo que facilitó en gran parte esta etapa de desarrollo.



Descripción de la Interfaz

Se instancia los patrones de tareas descritos en la iteración 0:

- Captura de datos: utilizado en la vista del coordinador al momento de editar la agenda en la fase 1 de la reunión, en la fase 2 de la reunión si el coordinador decide colocar un punto en votación al momento de insertar la resolución sobre el punto.
- Foro: en la fase 1 de la reunión los participantes pueden realizar sugerencias sobre la agenda, en la fase 2 el participante y el coordinador pueden realizar comentarios a un punto no informativo.

A continuación se presentan las diferentes interfaces generadas en esta iteración:

Fase 1 de la Reunión: Consideración de la Agenda

Reunión:  02/10/2009 a las 12:00am Rol:  Ver Agenda Actualizar

A continuación se presenta, las sugerencias que han emitido los invitados de la reunión, acerca de la agenda a tratar. Si usted lo desea puede proceder a emitir su sugerencia.

Por: Mama
02 de Octubre a las 04:34PM

Considero que cada uno de los puntos que conforman la agenda, son los adecuados para ser discutidos en la siguiente fase de la reunión.

Por: Yoli
02 de Octubre a las 04:33PM

A mi me parece que el punto 1 puede ser descartado de la agenda, que no es necesaria su discusión ya que es relevante.

Figura 3. 22 Modelo de interfaz. Realizar Sugerencias. Vista Participante fase 1 de la reunión.

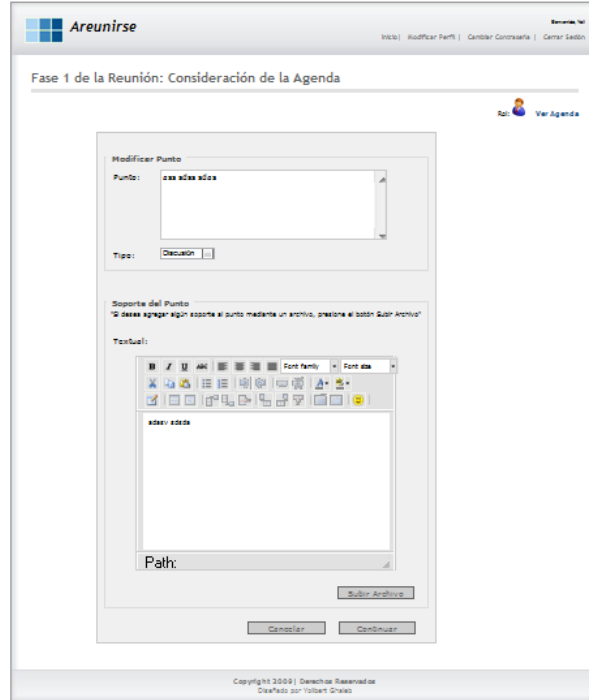


Figura 3. 23 Modelo de interfaz. Editar Puntos. Vista Coordinador fase 1 de la reunión.

Fase 2 de la Reunión: Discusión de la Agenda

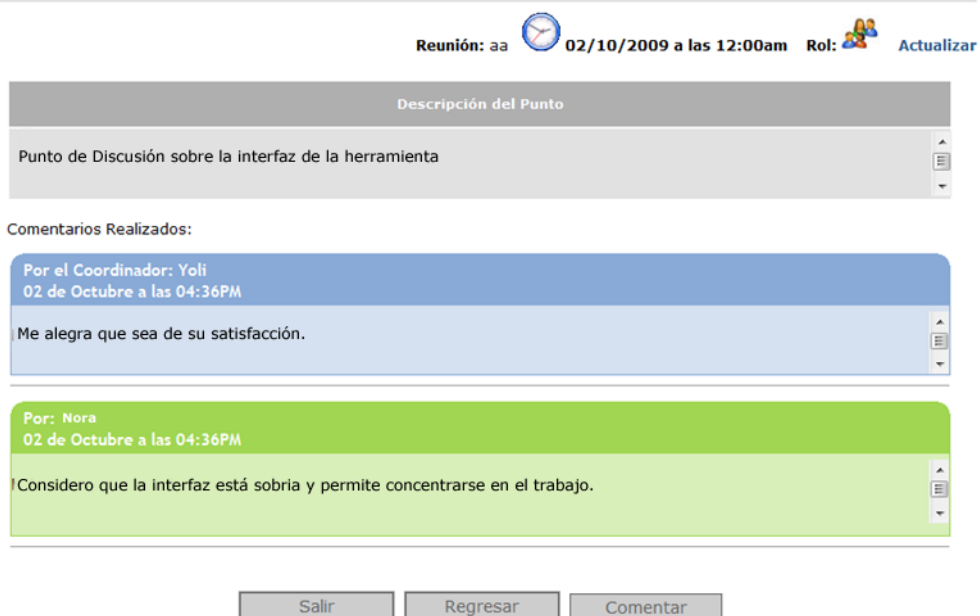


Figura 3. 24 Modelo de interfaz. Comentar Puntos. Vista Participante fase 2 de la reunión.

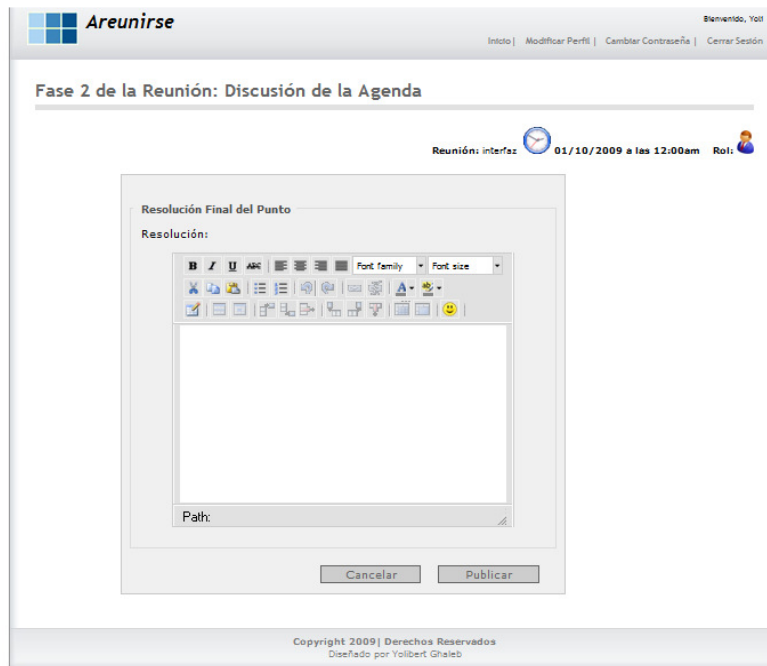


Figura 3. 25 Modelo de interfaz. Punto en Votación. Vista Coordinador fase 2 de la reunión.

Iteración 4

Fueron desarrollados los módulos correspondientes a Ver Agenda de la Reunión y Ver Resultados de Reuniones Pasadas.

Estos módulos ayudan a satisfacer otra de las características de las aplicaciones colaborativas presentadas en el marco conceptual, Memoria Grupal. Es necesario que los usuarios de la aplicación puedan observar cada vez que lo consideren necesario la información publicada por otros miembros, por ejemplo: la agenda de la reunión, las sugerencias o comentarios realizados por los participantes y el coordinador en la reunión, los detalles de la reunión.

Adicionalmente, aquella información generada producto de los resultados obtenidos por procesos que se ejecutaron en el sistema como: las actas al concluir una reunión, las cuales contienen los diferentes comentarios que se realizaron a cada uno de los puntos en discusión, la resolución tomada por el coordinador a cada punto y las estadísticas finales de la votación efectuada.

Para poder llevar a cabo el modulo de Ver Resultados de Reuniones Pasadas, fue necesario la instalación de las gemas de rails que permiten la generación de documentos tipo PDF, ya que las actas se generan en la aplicación en ese formato. Estas gemas son: pdf-writer (1.1.8) y ptools (1.1.6).

Descripción de la Interfaz

Se instancia los patrones de tareas descritos en la iteración 0:

- Seleccionar Opción de una Lista: ver agenda de la reunión.
- Selección de Datos: ver resultados de reuniones pasadas.

A continuación se presentan las diferentes interfaces generadas en está iteración:

Areunirse

Inicio | Modificar Perfil | Cambiar Contraseña | Cerrar Sesión

Menú

- Ver Agenda de la Reunión
- Crear Reunión
- Administrar Reunión
- Mis Contactos
- Ver Resultados de Reuniones Pasadas
- Entrar a la Sala de Reunión

Agenda de la Reunión

Seleccione la reunión a la que desea ver su agenda.

Reuniones que Coordina.

Reunión	Duración	Estado
Agenda 01	Del 24/09/2009 al 25/09/2009	Reunión cancelada
Agenda 02	Del 23/09/2009 al 25/09/2009	Reunión cancelada
Agenda 03	Del 26/09/2009 al 29/09/2009	Reunión cancelada
Agenda 04	Del 29/09/2009 al 01/10/2009	Fase 2 de la Reunión
Agenda 05	Del 30/09/2009 al 02/10/2009	Fase 1 de la Reunión
Agenda 06	Del 30/09/2009 al 02/10/2009	Fase 1 de la Reunión
Agenda 07	Del 30/09/2009 al 01/10/2009	Fase 2 de la Reunión
Agenda 08	Del 28/09/2009 al 02/10/2009	Reunión cancelada
Agenda 09	Del 30/09/2009 al 01/10/2009	Fase 2 de la Reunión

Cancelar

Copyright 2009 | Derechos Reservados
Diseñado por Yoibert Galea

Figura 3. 26 Modelo de interfaz. Ver Agenda de la Reunión.

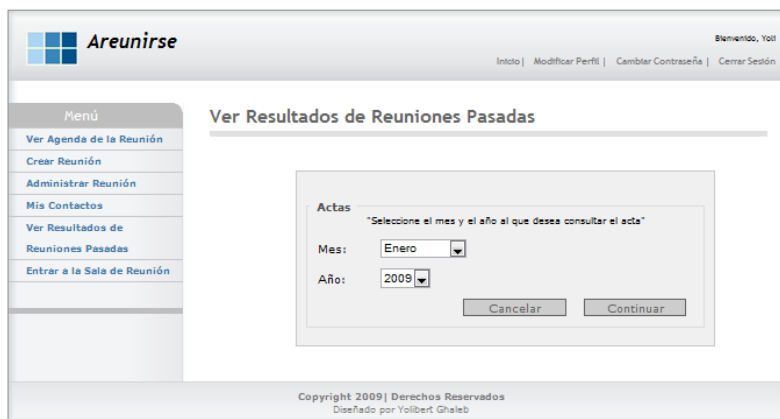


Figura 3. 27 Modelo de interfaz. Ver Resultados de Reuniones Pasadas.

Iteración 5

Finalmente se presenta la última iteración de la aplicación desarrollada donde se realizó el módulo de Mis Contactos y Usuarios en Línea.

Mis Contactos tiene como objetivo principal proporcionarle a los usuarios de la aplicación la posibilidad de administrar su grupo de trabajo, esto con el fin de facilitarles el proceso de crear una reunión en el sistema ya que podrán importar su grupo de trabajo y automáticamente enviarles la invitación a cada uno de ellos; así como también seleccionar otros invitados que no pertenezcan a ese grupo. Posteriormente se presenta el refinamiento de este caso de uso donde es posible observar las funcionalidades que ofrece.

Usuarios en Línea fue elaborado con el fin de satisfacer la última de las características que debe proveer una aplicación colaborativa Percepción de Usuarios. Cuando se está trabajando con este tipo de aplicaciones es importante mantener informado a los miembros del grupo las actividades que desarrollan los otros participantes, a través de este modulo es posible observar que usuarios se encuentran en línea actualmente.

Adicionalmente otra forma de satisfacer esta característica en la aplicación, es por medio de la presentación de quienes han realizado los comentarios y sugerencias en la sala de reuniones; así como también informar sobre las acciones que ha llevado a cabo el coordinador de la reunión por ejemplo: si ha cambiado el estado de un punto en discusión o

si ha decidió cancelar la ejecución de la reunión. En la figura 3.28 se presenta el caso de uso mis contactos en un nivel 2 de detalle.

Refinamiento de las Funcionalidades

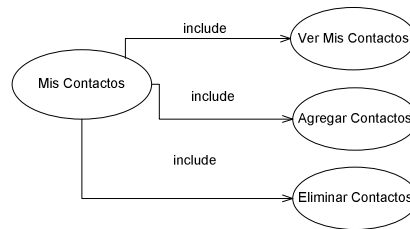


Figura 3. 28 Modelado del RF-5

Descripción del Modelado del RF-5

Nombre	Descripción
RF- 5.1 Ver Mis Contactos	El usuario autenticado puede ver los diferentes contactos que posee en su lista.
RF- 5.2 Agregar Contactos	El usuario autenticado puede agregar contactos a su lista.
RF- 5.3 Eliminar Contactos	El usuario autenticado puede seleccionar aquellos contactos que ya no desea pertenezcan a su lista.

Tabla 3. 6 Refinamiento del requerimiento funcional 5. Descripción breve de del casos de uso.

Descripción de la interfaz

Se instancia los patrones de tareas descritos en la iteración 0:

- Captura de Datos: al momento de agregar un nuevo contacto.
- Filtrado de Tablas: para ver mis contactos.

A continuación se presentan las diferentes interfaces generadas en está iteración:

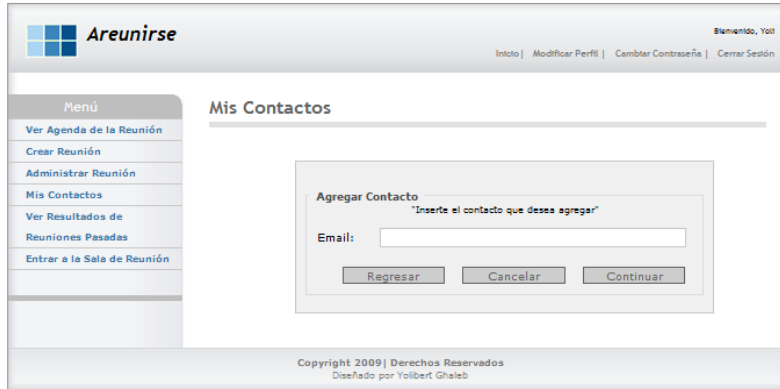


Figura 3. 29 Modelo de interfaz. Agregar un Nuevo Contacto.

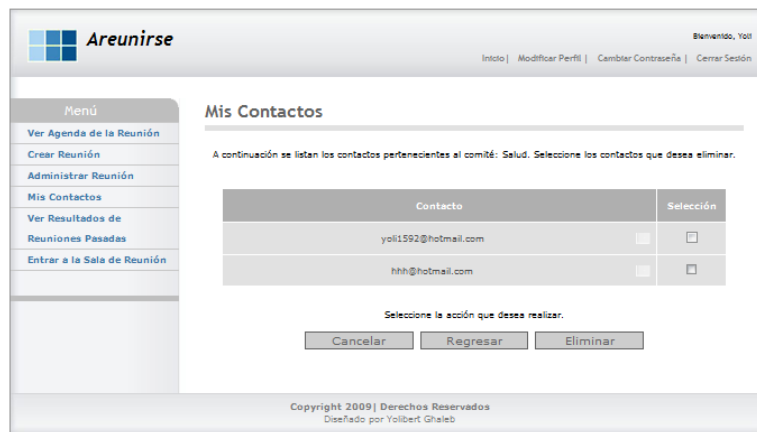


Figura 3. 30 Modelo de interfaz. Ver mis Contactos.

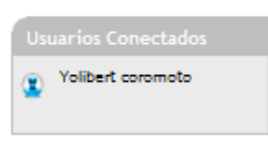


Figura 3. 31 Modelo de interfaz. Usuarios en Línea.

3.4 Liberación del Producto

En esta sección se presenta una descripción detallada de cómo se llevó a cabo la última etapa que caracteriza la realización de un software. Se expone los pasos que se realizaron para montar la aplicación final en producción; así como también las diferentes pruebas efectuadas a la aplicación.

3.4.1 Producto Final

Una vez que concluyó el desarrollo de la aplicación, se procedió a realizar la liberación del producto, este proceso duró aproximadamente una semana y media. Para lograr efectivamente esta labor fue necesario efectuar una serie de pasos que se presentan a continuación:

- Búsqueda de un servidor dentro de la universidad para colocar la aplicación final, que contará con salida hacia afuera ya que era fundamental para poder realizar las pruebas al software por parte de los usuarios, desde cualquier ubicación.
- Instalación en el servidor de cada una de las gemas de rails utilizadas para el correcto funcionamiento de la aplicación como: Calendar_date_select versión 1.15., color versión 1.4.0, color_tools versión 1.3.0, htmlentities versión 4.2.0, mime-types versión 1.16, mislav-will_paginate versión 2.3.11, mongrel versión 1.1.5, mysql versión 2.7.3, pdf-writer versión 1.1.8, ptools versión 1.1.6, tiny_mce_helper versión 0.3.0, tldmail versión 0.0.1.
- Instalación de la base de datos MySQL Server 5.0 y cargar el script de la aplicación en la base de datos.

Durante este proceso se presentaron varios inconvenientes que ocasionaron el retardo de la puesta en marcha de la aplicación.

Al principio fue proporcionado el servidor Ambar, para instalar la aplicación ya que el mismo tiene salida exterior, sin embargo no se contaba con el percance que se presentó. Posteriormente de haber realizado todas las instalaciones necesarias y poner a prueba la aplicación, la misma ocasionaba conflictos con la base de datos que estaba instalada en el servidor ya que poseía una configuración diferente con la que se había venido trabajando.

Luego de varios días tratando de solucionar éste problema se decidió migrar a otro servidor en Linux, se procedió a realizar nuevamente las instalaciones requeridas. Al finalizar y correr la aplicación se comprobó que la misma trabaja correctamente y actualmente reside en este servidor.

Dicho servidor no cuenta con salida hacia afuera de la universidad, por lo que surgió la necesidad de mapear un servidor que si posee salida exterior el cual será el encargado de atender las peticiones que realizan los usuarios y redireccionarlas al servidor donde reside la aplicación.

Actualmente es posible acceder a la aplicación mediante el link <http://lia.ciens.ucv.ve/tesis/principal>.

Paquete de Instalación

Para montar la aplicación desde cualquier servidor es necesario realizar una serie de instalaciones que se presentan a continuación:

- Ruby 1.8.6, el cual puede ser descargado de la página <http://rubyonrails.org/>
- Rails versión 2.2.2. Esto se realiza por consola a través del comando `gem install rails -v 2.2.2`
- Base de Datos MySQL Server 5.0
- Base de Datos para Ruby por consola a través del comando `gem install mysql`
- Instalar cada una de las gemas de Ruby on Rails necesarias para el correcto funcionamiento de la aplicación: `Calendar_date_select` versión 1.15., `color` versión 1.4.0, `color_tools` versión 1.3.0, `htmlentities` versión 4.2.0, `mime-types` versión 1.16, `mislav-will_paginate` versión 2.3.11, `mongrel` versión 1.1.5, `mysql` versión 2.7.3, `pdf-writer` versión 1.1.8, `ptools` versión 1.1.6, `tiny_mce_helper` versión 0.3.0, `tlsmail` versión 0.0.1.
- Los navegadores Internet Explorer 8 o Mozilla Firefox.
- Cargar el script de la base de datos de la aplicación.
- El servidor web Apache.

3.4.2 Pruebas a la Aplicación

El proceso de pruebas del software tiene como objetivo principal verificar la calidad del producto final. Durante el desarrollo de la aplicación se llevaron a cabo diferentes

pruebas realizadas por el desarrollador, las cuales permitieron determinar el correcto funcionamiento del software elaborado.

Estas pruebas fueron divididas según el escenario actual y el rol que desempeña en ese instante de tiempo, un usuario de la aplicación.

A continuación la descripción de las pruebas con sus casos validos e inválidos. Cuando se presenten casos inválidos la interfaz despliega automáticamente un mensaje de error, en caso contrario se realiza la acción deseada exitosamente.

Escenario Registro de Usuario

Descripción: el registro de usuario se presenta al iniciar la aplicación permite a los usuarios que no se encuentran aún registrados en la aplicación, la posibilidad de registrarse para poder comenzar a utilizar las funcionalidades que brinda.

Usuarios: Usuario no Registrado en la aplicación.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Registro de Usuario	Que inserte todos los campos obligatorios correctamente	<ul style="list-style-type: none"> - Que inserte el nombre o apellido acompañado de algún carácter numérico. - Que el password no cumpla con las condiciones de contener mínimo 6 caracteres y máximo 12. - Que no se repita el password correctamente. - Que el email ingresado no cumpla con las características de una dirección de correo electrónico valido. - Que no seleccione alguna pregunta de seguridad. - Que deje algún campo vacío ya que todos son obligatorios.

Tabla 3. 7 Prueba Escenario Registro de Usuario.

Escenario Olvido su Contraseña

Descripción: permite a un usuario registrado en el sistema recuperar su contraseña en caso de que la haya olvidado.

Usuarios: Usuario Registrado en la aplicación.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Olvido su Contraseña	<ul style="list-style-type: none">- Que inserte el login con el cual se registro en la aplicación.- Que responda la pregunta de seguridad que se le realizó correctamente.- Que inserte la contraseña nueva y la repita adecuadamente.	<ul style="list-style-type: none">- No ingreso un login de usuario.- Dejo el campo respuesta vacio.- Ingreso una contraseña que no corresponda con mínimo 6 caracteres y máximo 12.- Se dejo el campo contraseña o repetir password vacios.- Que el campo repetir password no coincida con el password ingresado al principio.

Tabla 3. 8 Prueba Escenario Olvido su Contraseña.

Escenario Iniciar Sesión

Descripción: permite a un usuario registrado en el sistema poder autenticarse.

Usuarios: Usuario Registrado en la aplicación.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Iniciar Sesión	Que ingrese el login y el password correctamente	<ul style="list-style-type: none">- No ingreso el login de usuario.- No ingreso el password.- El login o el password que insertó no se encuentre en la base de datos de la aplicación.

Tabla 3. 9 Prueba Escenario Iniciar Sesión.

Escenario Modificar Perfil

Descripción: permite a un usuario autenticado en el sistema, poder modificar los datos que ingresó el día de su registro en la aplicación.

Usuarios: Usuario Autenticado en la aplicación.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Modificar Perfil	Que inserte todos los campos obligatorios correctamente	<ul style="list-style-type: none">- modifiko su nombre o apellido y le inserte algún carácter numérico.- Insertó un correo electrónico que ya se encuentra registrado en la base de datos por otro usuario.- El email ingresado no cumple con las características de una dirección de correo electrónico valido.- Que deje algún campo vacio.

Tabla 3. 10 Prueba Escenario Modificar Perfil.

Escenario Cambiar Contraseña

Descripción: permite a un usuario autenticado en el sistema, poder modificar su contraseña.

Usuarios: Usuario Autenticado en la aplicación.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Cambiar Contraseña	<ul style="list-style-type: none">- Que ingrese su contraseña anterior correctamente.- Que la contraseña nueva ingresada haga correspondencia al repetirla.	<ul style="list-style-type: none">- La contraseña anterior ingresada no sea la misma, a la que se encuentra registrada en la base de datos.- Se dejo algún campo vacio.- La nueva contraseña no sea de mínimo 6 caracteres y máximo 12.

Tabla 3. 11 Prueba Escenario Cambiar Contraseña.

Escenario Ver Agenda de la Reunión

Descripción: permite a los usuarios no registrados en la aplicación poder observar las agendas de reuniones públicas programadas, en caso de que haya. Si hay reuniones

privadas pautadas solo podrán ser vistas por los participantes y el coordinador de esa reunión.

Usuarios: Usuario Autenticado en la aplicación, Usuario no Registrado, Participantes y Coordinador de una reunión.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Ver Agenda de la Reunión	Que seleccione alguna reunión, de las desplegadas en la interfaz. En caso de que haya más de una reunión programada.	No contiene casos inválidos

Tabla 3. 12 Prueba Escenario Ver Agenda de la Reunión.

Escenario Crear Reunión

Descripción: permite a los usuarios autenticados en la aplicación poder crear una reunión en el sistema, especificando sus características principales como: tiempo de inicio y finalización, tipo de reunión pública o privada (En una reunión pública cualquier usuario puede ver su agenda, en las privadas solo el coordinador y sus participantes), especificar el comité de esa reunión, ingresar el encabezado, la descripción y los puntos de la agenda, especificar la duración en días que tendrá cada fase de la reunión en la aplicación (Fase 1: Orden del Día, Fase 2: Discusión de la Agenda), ingresar los participantes que se desean invitar a la reunión.

Usuarios: Usuario Autenticado en la aplicación, Participantes y Coordinador de una reunión.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Crear Reunión Paso 1: Detalles de la Reunión	Insertar y seleccionar todos los campos del formulario.	<ul style="list-style-type: none"> - No se selecciono una fecha de inicio. - No se selecciono una fecha de fin de la reunión. - La fecha de inicio es igual a la fecha de fin de la reunión. - La fecha de fin de la reunión es mayor a la fecha de inicio. - No se escogió el tipo de reunión. - No se selecciono un comité.

		<ul style="list-style-type: none"> - En caso de crear un nuevo Comité deo alguno de sus campos vacios o inserto manualmente algún comité que ya está en la base de datos y no lo selecciono de la lista.
<p>Crear Reunión Paso2: Crear Agenda</p>	<p>Insertar el encabezado y la descripción de la agenda.</p>	<ul style="list-style-type: none"> - No se insertó el encabezado de la agenda de la reunión.
<p>Crear Reunión Paso2: Puntos de la Agenda</p>	<p>Insertar el punto y seleccionar su tipo. El soporte es opcional</p>	<ul style="list-style-type: none"> - Al insertar un punto no se coloco su descripción. - No se selecciono el tipo de punto (Informativo o de Discusión). - En caso de agregarle soporte si es textual que intente insertar el soporte y deje el campo vacio. - Si el soporte es subir un archivo, tiene que ingresar alguno que este guardado en su computador y que su tamaño no sea muy pesado, ya que el tiempo de duración de la sesión en el servidor es limitado.
<p>Crear Reunión Paso3: Especificación de los Tiempos</p>	<p>Insertar la distribución de los días que dura la reunión entre las dos fases: Orden del día y Discusión de la Agenda, adecuadamente.</p>	<ul style="list-style-type: none"> - Se inserto un carácter no numérico en la especificación de los tiempos de las fases. - La suma entre los días ingresados para las dos fases no coincide con la cantidad de días disponibles para efectuar esa reunión ya sea porque excede o es menor.
<p>Crear Reunión Paso4: Enviar Invitación</p>	<p>Se ha seleccionado o insertado algún email, el cual corresponde al participante que se desea invitar a esa reunión.</p>	<ul style="list-style-type: none"> - No se haya seleccionado algún contacto perteneciente a él o los comités asociados, en caso de poseerlos. O no se ha agregado manualmente algún participante. - Al momento de agregar manualmente algún participante, el email insertado no corresponde con las características que debe cumplir un correo electrónico valido. - Si se desea agregar manualmente más de un participante, la lista de emails no estén separados correctamente por el signo de (,).

Tabla 3. 13 Prueba Escenario Crear Reunión.

Escenario Modificar Reunión

Descripción: permite al coordinador de una reunión modificar las características de una reunión programada siempre y cuando la misma aún no haya comenzado o esté en estado de cancelada. Se selecciona lo que se desea modificar sus tiempos, agenda o participantes.

Usuarios: Coordinador de la reunión.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Modificar Reunión: Los Tiempos	Ingresar una fecha de inicio y fin de la reunión validas y distribuir adecuadamente los días en que durará cada fase.	<ul style="list-style-type: none">- Se cambio la fecha de inicio de la reunión y coincide con la de fin de la reunión.- Se distribuyeron mal los días de duración de cada fase puede exceder o ser menor a los permitidos.
Modificar Reunión: La Agenda	Se cambió el encabezado, la descripción o los puntos de la agenda adecuadamente.	<ul style="list-style-type: none">- Se dejo algún campo vacio en caso de que sea obligatorio.
Modificar Reunión: Los Participantes	Se elimino algún participante o participante pendiente correctamente o se decidió agregar un nuevo participante a esa reunión.	<ul style="list-style-type: none">- Se desea eliminar algún participante y no fue seleccionado.- Se desea invitar un nuevo participante a la reunión y el email ingresado no cumple con las características de un correo electrónico valido.- Si se desea agregar manualmente más de un participante, la lista de emails no estén separados correctamente por el signo de (,).

Tabla 3. 14 Prueba Escenario Modificar Reunión.

Escenario Mis Contactos

Descripción: permite a un usuario autenticado eliminar algún contacto de su lista si lo desea o agregar un contacto a su lista completa o por comité.

Usuarios: Usuario Autenticado.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Eliminar Mis Contactos	Selecciono los contactos que desea eliminar y procedió a su eliminación.	Desea eliminar algún contacto y no lo selecciono de la lista.
Agregar un Contacto	Ingreso un correo que cumple con las características de un correo electrónico valido.	<ul style="list-style-type: none"> - Desea agregar algún contacto y no insertó el correo electrónico en el formulario. - Insertó un correo electrónico no valido.

Tabla 3. 15 Prueba Escenario Mis Contactos.

Escenario Ver Resultados de Reuniones Pasadas

Descripción: permite a los usuarios registrados en la aplicación observar las diferentes actas generadas al concluir una reunión, donde hayan participado o coordinado.

Usuarios: Usuario Autenticado en la aplicación, Participante, Coordinador.

<i>Escenario</i>	<i>Casos Validos</i>	<i>Casos Inválidos</i>
Ver Resultados de Reuniones Pasadas	Que seleccione el mes y el año al que desea consultar las actas asociadas.	No contiene casos inválidos

Tabla 3. 16 Prueba Escenario Ver Resultados de Reuniones Pasadas.

El haber realizado cada una de estas pruebas permitieron detectar los errores que se presentaban al ejecutar cada modulo descrito, lo que condujo a realizar mejoras a la aplicación que mejoraran la usabilidad de la misma.

Conclusiones

La Aplicación Colaborativa Areunirse, brindará las herramientas necesarias para que cualquier usuario autenticado en la aplicación, que tenga la necesidad de organizar y coordinar una reunión entre un grupo de trabajo pueda planificarla con las características que considere más idóneas.

Una reunión en Areunirse está basado bajo el protocolo de colaboración de una reunión de tipo Agendas y Actas, este protocolo fue creado con la finalidad de tratar de garantizar el éxito de la reunión ya que el mismo provee de pasos preestablecidos que si son seguidos de manera adecuada por parte de los participantes y el coordinador de la reunión se puede alcanzar los objetivos propuestos y la toma de decisiones.

Es importante destacar que el éxito de una reunión en la aplicación dependerá de la ética de cada uno de los participantes y el coordinador de la reunión, si están dispuestos a trabajar en grupo y colaborar entre ellos aportando sus opiniones, el protocolo de colaboración creado se ejecutara adecuadamente y se obtendrán los resultados esperados, en caso contrario será un fracaso.

La aplicación elaborada cumple con las características que debe poseer una aplicación colaborativa entre ellas se destacan: protocolo de colaboración, esta característica está representada por medio del protocolo diseñado denominado Agendas y Actas, la memoria grupal a través de la funcionalidad ver resultados de reuniones pasadas donde se puede consultar las actas que representan el historial de cada una de las reuniones que se han desarrollado en la aplicación y el modulo ver agenda de la reunión que permite a los usuarios de la aplicación consultar las agendas de las reuniones donde participan o coordinan.

Otras de las características que se satisfacen son: la percepción de usuarios mediante el cual es posible observar los usuarios que se encuentran actualmente en línea y por último los roles que está representado en la aplicación por cuatro tipos: usuario, usuario autenticado, participante de la reunión y coordinador de la reunión.

Durante el proceso de desarrollo de la aplicación se pudo apreciar que la metodología del modelado ágil apoyada mediante el lenguaje de programación Ruby on Rails, facilitó la creación de la aplicación ya que nos permitió realizar un análisis exhaustivo de la problemática presente y permitió idear mecanismos para poder solventarla, dividiendo la carga de trabajo en iteraciones más pequeñas que se fueron realizando de acuerdo al orden de prioridades de cada una.

Mediante las pruebas realizadas se pudo proceder a corregir errores presentes en la aplicación; así como también comprobar su efectividad y calidad.

Extensiones del Trabajo

Es necesario resaltar que la aplicación desarrollada puede ser extendida con la finalidad de brindar nuevas funcionalidades a sus usuarios. Entre las mejoras que se podrían implementar se destacan:

- Diseñar nuevos protocolos de colaboración, esto con el fin de que una reunión en Areunirse se pueda realizar siguiendo otros pasos y no estén restringidas a reuniones de tipo Agenda y Actas.
- Se puede reutilizar el protocolo de colaboración Agenda y Actas para desarrollar el protocolo Tormenta de Ideas ya que presenta algunas características similares al elaborado.
- Agregar nuevos módulos a la aplicación como por ejemplo un calendario dinámico de eventos.

Referencias Bibliográficas

Ambler, S. (2002) Agile Modeling: Effective Practices for Extreme Programming and the Unified Process. Extraído el 29 de Enero de 2009 de:

<http://www.agilemodeling.com/essays/introductionToAM.htm>

Borchers J. (2000). Breaking the interdisciplinary limits of computer-human interaction design: A pattern approach. SIGCHI. Vol 32,1

Catalani, E. (2007). ARQUITECTURA Modelo/Vista/Controlador. Extraído el 30 de Enero de 2009 de: <http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>

Cem, B. (2009). Ruby. Extraído el 30 de Enero de 2009 de:

<http://es.wikipedia.org/w/index.php?title=Ruby&oldid=23605586>

Davila, R. (2006). Introducción a Ruby on Rails. Extraído el 30 de Enero de 2009 de:

<http://www.maestrosdelweb.com/editorial/rubyonrails/>

Ellis, C, & Gibbs, S, & Rein, G. (1991): "Groupware some issues y experiences", Comm. of the ACM, Vol. 34 No. 1. Extraído el 19 de Octubre de 2008 de:

<http://www2.ing.puc.cl/~group/sisco/cap2.tesis.html#2.1.3>

Erickson, T. (2001). The need for a Lingua franca for design: from sacred places to pattern languages. Obtenido el 2 de marzo de 2009, de IM T.J. Watson Research Center en

<http://www.visi.com/snowfall/DesignLinguaFranca.html>

Hernández, M., Rodríguez, M. (1996). Trabajo en grupo y "groupware". Bogotá: CIFIUNIANDES. Extraído el 25 de Octubre de 2008.

Jay, A. (1976), "How to Run a Meeting", ,Readings in Groupware and Computer-Suport Cooperative Work, San Mateo, CA, Morgan Kaufmann Publishers, 1993, pp.130-145.

Johnson, Lenz & Peter. (1981). Consider the Groupware: Design and Group Process Impacts on Communication in the Electronic Medium. In Hiltz, S. and Kerr, E. (eds.).

Studies of Computer-Mediated Communications Systems: A Synthesis of the Findings, Computerized Conferencing and Communications Center, New Jersey Institute of Technology, Newark, New Jersey. Extraído el 23 de Octubre.

Martínez, M. (2005). Trabajo cooperativo asistido por computadora. Revista de Divulgación Científica y Tecnológica De La Universidad Veracruzana, Volumen XVIII Numero 2. Extraído el 21 de Octubre de 2008 de:
<http://www.uv.mx/cienciahombre/revistae/vol18num2/articulos/tcac/index.htm>

McDaniel, S. (1996). "Providing Awareness Information to Support Transitions in Remote Computer-Mediated Collaboration", Extraído el 23 de Octubre de 2008 de:
<http://www.crew.umich.edu/mcdaniel/doc.cons.html>.

Numamaker, J., Dennis, A., George, J., Valacich, J. & Vogel, D. (1991). "Electronic meeting systems to support group work: Theory and practice at Arizona". Communications of the ACM, pp. 40-61. Extraído el 23 de Octubre de 2008 de:
<http://www2.ing.puc.cl/~group/sisco/cap3.tesis.html>

Ramage M. (1999). The Learning way: Evaluating co-operative systems. Tesis Doctoral publicada en línea. Lancaster University, Reino Unido. Extraído el 28 de Octubre de:
<http://systems.open.ac.uk/objects/MagnusR/LearningWay.pdf>

Tidwell J. (1999). Common Ground: A Pattern Language for Human-Computer Interface Design. Extraído el 28 de Octubre de: http://www.mit.edu/~jtidwell/interaction_patterns.html

Welie M. (2008). Patterns in Interaction Design. Extraído el 15 de Julio de 2009 de:
<http://www.welie.com/patterns/>

Wilson. (1991). Soporte Informático para trabajo colaborativo. Extraído el 21 de Octubre de 2008 de: <http://www.lsi.uned.es/tcolab/material/trabajoColaborativo.pdf>

Anexos

A continuación se presenta la descripción detalla de cada uno de los casos de uso presentados durante el capítulo 3 del marco de desarrollo.

Descripción

Caso de Uso Autenticarse

RF- 1 (Fig. 3.7)	Autenticarse	
Descripción	Permitir a un usuario poder llevar a cabo las funcionalidades básicas de toda aplicación, como lo son: el registro de usuario, inicio de sesión, recuperar contraseña o cambio de contraseña.	
Precondición	El usuario que hace la solicitud puede o no estar registrado en el sistema.	
Secuencia Normal	Paso	Acción
	1	Identificar si es un usuario que se encuentra registrado.
	2	Nuevo Usuario, lo que implica que se tiene que registrar.
	3	Recuperar contraseña en caso de olvido.
	4	Cambiar contraseña.
Post condición	El usuario puede realizar cualquier funcionalidad que necesite, para convertirse en un usuario autenticado con más privilegios.	
Rendimiento	Paso	Acción
	3	10 segundos
Frecuencia Esperada	10 veces/día	
Estabilidad	Alta	
Comentarios	La frecuencia será mucho mayor al momento de llevarse a cabo una reunión	

Caso de Uso Crear Reunión

RF- 2 (Fig. 3.7)	Crear Reunión	
Descripción	Permitir a un usuario autenticado en la aplicación crear una reunión.	
Precondición	El usuario tiene que estar registrado en el sistema.	
Secuencia Normal	Paso	Acción

	1	El usuario ingresa a la opción crear reunión.
	2	El sistema despliega las interfaces acordes para su creación.
	3	Se procede a enviar la invitación por medio del correo electrónico
Post condición	El usuario ha creado una reunión en el sistema, y se puede desarrollar.	
Rendimiento	Paso	Acción
	3	10 segundos
Frecuencia Esperada	2 veces/día	
Estabilidad	Alta	
Comentarios	Ninguno	

Caso de Uso Ver Resultados de Reuniones Pasadas

RF- 3 (Fig. 3.7)	Ver Resultados de Reuniones Pasadas	
Descripción	Que los usuarios de la aplicación estén enterados de las decisiones tomadas en reuniones que se llevaron a cabo, donde participaron y puedan ver el acta generada.	
Precondición	Se ha efectuado alguna reunión.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la opción ver resultados de reuniones pasadas
	2	Selecciona el mes y el año al que desea consultar sus actas asociadas.
	3	El sistema descarga automáticamente una interfaz con los resultados.
Post condición	El usuario puede visualizar las actas, producto de las reuniones desarrolladas.	
Rendimiento	Paso	Acción
	3	10 segundos
Frecuencia Esperada	2 veces/día	
Estabilidad	Alta	
Comentarios	Ninguno	
RF- 4 (Fig. 3.7)	Modificar Perfil de Usuario	
Descripción	Permitir a un usuario autenticado actualizar los datos personales que ingreso al momento de su registro.	

Precondición	El usuario tiene que estar registrado en la aplicación.	
Secuencia Normal	Paso	Acción
	1	Identificar si es un usuario que se encuentra registrado.
	2	El usuario ingresa a la opción de modificación de perfil y el sistema despliega la interfaz que contiene el formulario con sus datos editables.
	3	El usuario procede a actualizar los campos que desea.
	4	El sistema envía una notificación.
Post condición	El usuario actualizó sus datos en la base de datos.	
Excepciones	Paso	Acción
	3	Que no haya insertado correctamente los datos, en el campo que desea actualizar. Paso 4.
	4	Enviar un mensaje al usuario indicando que hay un error.
Rendimiento	Paso	Acción
	4	5 segundos
Frecuencia Esperada	1 veces/día	
Estabilidad	Alta	
Comentarios	Ninguno	

Caso de Uso Mis Contactos

RF- 5 (Fig. 3.7)	Mis Contactos	
Descripción	Permite a un Usuario Autenticado ver sus contactos, eliminar algún contacto de su lista o agregar un nuevo contacto.	
Precondición	El usuario Autenticado tiene que haber iniciado sesión exitosamente.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la opción Mis contactos.
	2	El sistema despliega las opciones: de ver lista de contactos o agregar un nuevo contacto.
	3	El usuario escoge entre alguna de las opciones.
	4	Dependiendo de la opción seleccionada el sistema despliega la interfaz correspondiente.
Post condición	El usuario puede proceder a realizar la acción que dese.	
Rendimiento	Paso	Acción
	2, 4	10 segundos
Frecuencia Esperada	1 vez por semana	

Estabilidad	Alta
Comentarios	Ninguno

Caso de Uso Administrar Reunión y Entrar a la Sala de Reunión

RF- 6 (Fig. 3.7)	Administrar Reunión	
Descripción	Permite al coordinador de la reunión modificar, eliminar, o ver una reunión. El participante de la reunión solo puede ver los detalles de la reunión.	
Precondición	El usuario tiene que ser coordinador o participante de la reunión y estar registrado en el sistema.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la opción administrar reunión.
	2	El sistema despliega tres opciones: modificar, eliminar reunión o ver detalles de la reunión.
	3	El coordinador escoge entre alguna de las opciones
	4	Dependiendo de la opción seleccionada el sistema despliega las acciones pertinentes.
Post condición	El coordinador o el participante pueden proceder a realizar la acción que desee correspondiente a una reunión.	
Rendimiento	Paso	Acción
	2, 4	10 segundos
Frecuencia Esperada	1 veces/día	
Estabilidad	Alta	
Comentarios	Ninguno	

RF- 7 (Fig. 3.7)	Entrar a la Sala de Reunión	
Descripción	Los usuarios autorizados (Participantes invitados y coordinador) pueden ingresar a la reunión programada en esa fecha.	
Precondición	Se ha creado previamente una reunión en el sistema y los participantes de la reunión tienen que estar autenticados.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la opción entrar a reunión
	2	El sistema despliega automáticamente las distintas fases de la reunión programada para esa fecha.
Post condición	El coordinador y los participantes pueden comenzar a llevar a cabo la reunión, para lograr al final el objetivo planteado.	
Rendimiento	Paso	Acción

	2	10 segundos
Frecuencia Esperada	5 veces al día	
Estabilidad	Alta	
Comentarios	Aumentará la frecuencia dependiendo de la cantidad de participantes invitados a la reunión	

Caso de Uso Ver Agenda de la Reunión

RF- 8 (Fig. 3.7)	Ver Agenda de la Reunión	
Descripción	Los usuarios de la aplicación puedan ver las agendas de las reuniones públicas o privadas programadas.	
Precondición	Que haya reuniones programadas.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la opción ver agenda de la reunión.
	2	Selecciona la fecha de la reunión al que desea ver su agenda en caso de que haya más de una. Si el usuario no ha iniciado sesión y hay reuniones públicas programadas se les despliegan sus respectivas agendas. El tratamiento de la agenda para reuniones de tipo privada es distinto a una pública, ya que estas agendas solo pueden ser vistas por los participantes y el coordinador de esa reunión.
	3	El sistema descarga automáticamente una interfaz con los resultados.
Post condición	El usuario puede visualizar las agendas de las reuniones programadas.	
Rendimiento	Paso	Acción
	3	10 segundos
Frecuencia Esperada	2 veces/día	
Estabilidad	Alta	
Comentarios	Ninguno	

Descripción del Modelado del RF-6

RF- 6.1 (Fig. 3.17)	Ver Detalles de la Reunión	
Descripción	El participante o coordinador de la reunión, podrán ver las características de las reuniones donde están involucrados.	
Precondición	Es necesario estar autenticado en la aplicación.	
Secuencia Normal	Paso	Acción

	1	El usuario ingresa a la opción administrar reunión.
	2	El sistema despliega las diferentes reuniones asociadas a ese usuario. El usuario selecciona la reunión que desea consultar y se despliega sus detalles.
Post condición	Se pueden observar los datos de la reunión sus participantes, los puntos que conforman la agenda, la duración de las fases.	

RF- 6.2 (Fig. 3.17)	Modificar Reunión	
Descripción	El coordinador de la reunión puede proceder a modificar los datos de la reunión, en caso de que no haya iniciado, concluido o se encuentre en estado de cancelada.	
Precondición	La reunión no puede haber comenzado aún o estar ejecutándose.	
Secuencia Normal	Paso	Acción
	1	El coordinador ingresa a la opción administrar reunión, el sistema despliega las diferentes reuniones asociadas a él. El usuario selecciona la de su preferencia y específica que desea modificar: tiempo, agenda o participantes.
	2	El coordinador selecciona la opción de su preferencia, el sistema carga la interfaz para su modificación.
Post condición	Se modifico los detalles de la reunión y se procedió a informar a sus participantes por medio del envió de correo electrónico.	

RF- 6.3 (Fig. 3.17)	Eliminar Reunión	
Descripción	El coordinador de la reunión puede proceder a eliminar una reunión por completo de la aplicación.	
Precondición	La reunión no puede haber comenzado aún o estar ejecutándose.	
Secuencia Normal	Paso	Acción
	1	El coordinador ingresa a la opción administrar reunión, el sistema despliega las diferentes reuniones asociadas a él.
	2	El coordinador selecciona la reunión de su preferencia, el sistema carga la interfaz para su eliminación.
Post condición	El coordinador elimino la reunión y se informó a sus participantes por medio del envió del correo electrónico.	

Descripción del Modelado del RF-5

RF- 5.1 (Fig. 3.29)	Ver Mis Contactos	
Descripción	El usuario autenticado puede ver los diferentes contactos que posee en su lista.	

Precondición	Es necesario estar autenticado en la aplicación.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la opción mis contactos y escoge ver mis contactos por selección de comité o por lista completa.
	2	El sistema despliega la lista de los contactos asociados.
Post condición	Se pueden observar los diferentes contactos de ese usuario.	

RF- 5.2 (Fig. 3.29)	Agregar Contactos	
Descripción	El usuario autenticado puede agregar contactos a su lista.	
Precondición	Es necesario estar autenticado en la aplicación.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la opción mis contactos y selecciona agregar contacto por comité o a la lista completa.
	2	El sistema despliega la interfaz que contiene un formulario donde puede insertar la dirección de correo electrónico, correspondiente al contacto que se desea agregar.
Post condición	Se inserto un nuevo contacto a la lista.	

RF- 5.3 (Fig.3.29)	Eliminar Contactos	
Descripción	El usuario autenticado puede seleccionar aquellos contactos que ya no desea pertenezcan a su lista.	
Precondición	Es necesario estar autenticado en la aplicación.	
Secuencia Normal	Paso	Acción
	1	El usuario ingresa a la opción mis contactos y escoge ver mis contactos por selección de comité o por lista completa.
	2	El sistema despliega la lista de los contactos asociados y adicionalmente habilita checks al lado de cada contacto, para que el usuario seleccione aquellos que desea eliminar.
Post condición	Se pueden editar la lista de contactos.	