



**UNIVERSIDAD NACIONAL DE LOJA**  
**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y**  
**LOS RECURSOS NATURALES NO RENOVABLES**

**CARRERA DE INGENIERÍA EN SISTEMAS**

**“DISEÑO E IMPLEMENTACIÓN DE UNA  
APLICACIÓN CON RECURSOS DE  
VEDOCONFERENCIA SOBRE INTERNET 2,  
PARA COLABORACIÓN DE GRUPOS  
MULTIDISCIPLINARIOS DE PERSONAS”**

TESIS PREVIA A OPTAR EL GRADO  
DE INGENIEROS EN SISTEMAS

AUTORES:

*Victor Manuel Gordillo Pacheco*  
*Willan Fernando Mejía Vásquez*

DIRECTOR:

*Ing. Milton Leonardo Labanda Jaramillo*

LOJA - ECUADOR  
2009



## CERTIFICACIÓN

Ing. Milton Leonardo Labanda Jaramillo

**Docente del Área de la Energía, las Industrias y los Recursos Naturales no  
Renovables de la Universidad Nacional de Loja**

### **CERTIFICA:**

Haber asesorado y revisado durante todo el desarrollo, la Tesis titulada: "DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN CON RECURSOS DE VIDEOCONFERENCIA SOBRE INTERNET 2, PARA COLABORACIÓN DE GRUPOS MULTIDISCIPLINARIOS DE PERSONAS", elaborada por los señores egresados Víctor Manual Gordillo Pacheco y Willan Fernando Mejía Vásquez, previo a la obtención del grado de Ingenieros en Sistemas.

En tal virtud cumple los requisitos que exigen las normas de graduación de esta Institución, por lo expuesto autorizo su presentación y defensa.

Loja, junio del 2009

Ing. Milton Leonardo Labanda Jaramillo  
**DIRECTOR DE TESIS**



## AUTORIA

Las ideas y conceptos vertidos en el presente trabajo de investigación han sido elaborados bajo los criterios de los autores, por lo tanto se declaran como autores legítimos de este trabajo de tesis.

**Víctor Manuel Gordillo Pacheco**  
**Willan Fernando Mejía Vásquez**



## CESIÓN DE DERECHOS

Víctor Manuel Gordillo Pacheco y Willan Fernando Mejía Vásquez, egresados de la Carrera de Ingeniería en Sistemas, conceden todos los derechos de autor de la tesis, cuyo tema es: **“DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN CON RECURSOS DE VIDEOCONFERENCIA SOBRE INTERNET 2, PARA COLABORACIÓN DE GRUPOS MULTIDISCIPLINARIOS DE PERSONAS”**, a la Universidad Nacional de Loja , de hacer uso de la misma con la finalidad que estime conveniente.

**Loja, Junio del 2009**

Víctor Manuel Gordillo Pacheco

**EGDO. CARRERA DE INGENIERÍA**

**EN SISTEMAS**

Willan Fernando Mejía Vásquez

**EGDO. CARRERA DE INGENIERÍA**

**EN SISTEMAS**



## **AGRADECIMIENTO**

Agradecemos en especial a la Universidad Nacional de Loja, al Área de la Energía, las Industrias y los Recursos Naturales no Renovables, a la Carrera de Ingeniería en Sistemas ya que en sus aulas se inicio un sueño que se verá por fin culminar.

Además un agradecimiento especial a nuestros distinguidos docentes ya que ellos implantaron sus conocimientos en nuestra vida universitaria, permitiendo así las bases necesarias para ser unos buenos profesionales y así lograr el objetivo planeado.

Nuestros más sinceros agradecimientos al Ingeniero Milton Leonardo Labanda Jaramillo, Director de Tesis, ya que nos a apoyado con sus conocimientos y sin su ayuda desinteresada no hubiésemos podido desarrollar y culminar el presente proyecto.

**Víctor Manuel Gordillo Pacheco**

**Willan Fernando Mejía Vásquez**

## **DEDICATORIA**



El proyecto de investigación lo dedico en especial a Dios, por regalarme la vida, la sabiduría y darme el conocimiento para hacer posible que culmine con mi carrera Universitaria.

A mis queridos padres por el apoyo moral, material y consejos que me ha servido para feliz término.

Ya que junto a ellos estuvieron mis hermanos que siempre me brindaron su apoyo para seguir adelante con positivismo y sobre todo sin retroceder a las adversidades.

**VICTOR                      MANUEL**  
**GORDILLO PACHECO**

Este proyecto lo dedico de corazón para el ángel que me observa desde el cielo Marlon Israel que de Dios goce, a Dios por permitirme llegar hasta aquí, a mi madre Blanca quien con su sacrificio y dedicación me han llevado a cumplir con esta meta, a mis hermanos por su apoyo y comprensión y a Anita por sus palabras de aliento

**WILLAN                      FERNANDO**  
**MEJÍA VASQUEZ**



## RESUMEN

El presente proyecto enfoca todo lo referente a la implementación del servicio de videoconferencia entre docentes y estudiantes de la Universidad Nacional de Loja.

Inicia con una fase de análisis de requerimientos, diseño de la aplicación como la descripción de casos de uso, diagramas de clases, casos de uso, robustez, secuencia, y la implementación de las aplicaciones Husky Videoconferency Servert, Husky Videoconferency client y Husky Web Videoconferency que permiten la comunicación mediante videoconferencia.

Posteriormente se llevó a cabo una fase de pruebas de las aplicaciones para comprobar su correcto funcionamiento, obteniendo resultados que nos permitieron evaluar los programas, así como también se realizó un plan de validación.

Además se explica la configuración del software utilizado para la implementación.



## SUMMARY

The present project focuses all it with respect to the implementation of the videoconference service among educational and students of the National University of Loja. It begins with a phase of analysis of requirements, design of the application like the description of cases of use, diagrams of classes, cases of use, robustness, sequence, and the implementation of the applications Husky Videoconferency Server, Husky Videoconferency Client and Husky Web Videoconferency that allow the communication mediating videoconference.

Later on it was carried out a phase of tests of the applications to check their correct operation, obtaining results that they allowed us to evaluate the programs, as well as was carried out a validation plan.

The configuration of the software used for the implementation is also explained.



## ÍNDICE GENERAL

<b>PORTADA</b>	<b>i</b>
<b>CERTIFICACIÓN</b>	<b>i.e.</b>
<b>AUTORÍA</b>	<b>i.e.</b>
<b>CESIÓN DE DERECHOS</b>	<b>iv</b>
<b>AGRADECIMIENTO</b>	<b>v</b>
<b>DEDICATORIA</b>	<b>vi.</b>
<b>RESUMEN</b>	<b>vii</b>
<b>SUMMARY</b>	<b>viii</b>
<b>ÍNDICE GENERAL</b>	<b>ix</b>
<b>1. INTRODUCCIÓN</b>	<b>1</b>
<b>2. METODOLOGÍA</b>	<b>3</b>
<b>3. MARCO TEÓRICO</b>	<b>7</b>

### CAPITULO I

#### INTERNET 2 Y VIDEOCONFERENCIA

1.1 INTRODUCCIÓN	7
1.2 DEFINICIÓN DE INTERNET 2	8
1.3 PROTOCOLO DE INTERNET 2	9
1.3.1 Tecnología IPv4	9
1.3.1.1 Broadcast y Multicast	9
1.3.2 Tecnología IPv6	10
1.3.3 Ventajas de IPv6	11
1.3.4 Características principales de IPv6	12
1.3.4.1 Multicast y unicast	12
1.4 Videoconferencia	13



1.4.1	Definición	13
1.4.2	Elementos básicos de un sistema de videoconferencia	13
1.4.2.1	La red de comunicaciones	13
1.4.2.2	La sala de videoconferencia	14
1.4.2.3	El codec	14
1.4.3	ESTÁNDAR H323 PARA VIDEOCONFERENCIA	15

## **CAPITULO II**

### **ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS CON UML**

2.1	INTRODUCCIÓN	16
2.2	PROCESO DE DESARROLLO DE SOFTWARE	16
2.2.1	Informática	16
2.2.2	Proceso de desarrollo	16
2.2.3	Importancia del proceso de desarrollo de software	17
2.2.4	Modelo	17
2.2.4.1	Etapas del proceso de desarrollo de software	18
2.2.5	Programación orientada a objetos	18
2.2.5.1	Orientación a objetos	18
2.2.5.2	Principios y fundamentos de programación orientada a objetos	18
2.3	Lenguaje Unificado de Modelado UML	19
2.3.1	Introducción	19
2.3.2	Elementos Estructurales	19
2.3.2.1	Clases	19



2.3.2.1.1 Clase abstracta	20
2.3.2.1.2 Clase concreta	20
2.3.2.2 Objeto	20
2.3.2.2.1 Atributo	20
2.3.2.2.2 Método	21
2.3.2.3 Interfaz	21
2.3.2.4 Colaboración	21
2.3.2.5 Casos de Uso	21
2.3.3 Relación de Asociación	22
2.3.3.1 Tipos de asociaciones	22
2.3.4 Diagramas	24
2.3.4.1 Diagramas de Clases	24
2.3.4.2 Diagramas de Casos de Usos	24
2.3.4.3 Diagramas de Secuencia y de Colaboración	24
2.3.4.4 Diagramas de Componentes	25
2.3.4.5 Diagrama de paquetes	25
2.3.4.6 Diagrama de robustez	25
2.3.5 Modelamiento de Casos de Uso	26
2.3.5.1 Diagrama de casos de uso	26
2.3.5.2 Descripción de casos de uso	27

### **CAPITULO III**

#### **HERRAMIENTAS DE PROGRAMACIÓN Y DESARROLLO**

3.1 INTRODUCCIÓN	28
3.2 LENGUAJE DE PROGRAMACIÓN JAVA	28
3.2.1 Definición	28



3.2.2	Características	29
3.2.3	JavaBeans	30
3.3	JMF -4JAVA MEDIA FRAMEWORK	32
3.3.1	Definición	32
3.3.2	Datos Basados en el Tiempo (Time-Based Media)	32
3.3.3	Características claves de los medios basados en el tiempo	33
3.3.4	Arquitectura de Alto Nivel de JMF	34
3.3.4.1	Presentación	34
3.3.4.2	Procesamiento	35
3.3.4.3	Captura	35
3.3.4.4	Almacenamiento y transmisión de medios	35
3.3.4.5	Clases e interfaces de JMF	35
4.3.4.5.1	DataSource	36
4.3.4.5.2	Player	37
4.3.4.5.3	Processor	38
4.3.4.5.4	Format	40
4.3.4.5.5	Manager	41
3.4	Java Server Pages y Contenedores de Servlets /JSPs	41
3.4.1	Definición	41
3.5	FRAMEWORK JSF – JAVA SERVER FACES	43
3.5.1	Definición	43
3.5.2	Principales Componentes	44
3.5.3	Aplicaciones Java Server Faces	44
3.6	HIBERNATE	55



3.6.1	Definición	55
3.6.2	Características	56
3.7	APACHE TOMCAT	65
3.7.1	Definición	65
3.7.2	Entorno	65
3.7.3	Estructura de directorios	65
3.8	Servidor de Base de Datos MySQL	67
3.8.1	Definición	67
3.8.2	Lenguajes de programación	67
3.8.3	Características distintivas	68
<b>4. DESARROLLO DE LA PROPUESTA ALTERNATIVA</b>		
4.1	INTRODUCCIÓN	69
4.2	DESCRIPCIÓN DE REQUERIMIENTOS	69
4.2.1	Requerimientos Funcionales	69
4.2.2	Requerimientos No Funcionales	70
4.3	Obtención de Clases del Dominio y Definición de Conceptos	71
4.3.1	Modelo del Dominio	72
4.4	MODELAMIENTO DE CASOS DE USO	73
4.4.1	Diagrama de Casos de Uso	74
4.4.2	Prototipo de Pantallas	75
4.4.3	Descripción de Casos de Uso	85
4.4.3.1	Caso de uso: Administrar Usuario	85
4.4.3.2	Caso de uso: Administrar Noticia	89
4.4.3.3	Caso de uso: Administrar foro de discusión	92
4.4.3.4	Caso de uso: Consultar Noticia	95
4.4.3.5	Caso de uso: Participar en foro de discusión	96
4.4.3.6	Caso de uso: Participar en Videoconferencia	98
4.4.4	DIAGRAMAS DE ROBUSTEZ	101
4.4.4.1	Diagramas de Robustez del Caso de Uso Administrar Usuario	101



4.4.4.2	Diagramas de Robustez del Caso de Uso Administrar Noticia	103
4.4.4.3	Diagramas de Robustez del Caso de Uso Administrar Foro de Discusión	105
4.4.4.4	Diagramas de Robustez del Caso de Uso Consultar Noticia	106
4.4.4.5	Diagramas de Robustez del Caso de Uso Participar en Foro de Discusión	107
4.4.4.6	Diagramas de Robustez del Caso de Uso Participar en Videoconferencia	108
4.4.5	DIAGRAMAS DE SECUENCIA	110
4.4.5.1	Diagramas de Secuencia del Caso de Uso Administrar Usuario	110
4.4.5.2	Caso de uso Administrar noticia	113
4.4.5.3	Caso de uso Administrar foro de discusión	116
4.4.5.4	Caso de uso Consultar noticia	118
4.4.5.5	Caso de uso Participar en foro de discusión	119
4.4.5.6	Caso de uso Participar en Videoconferencia	121
4.5	MAPA DE NAVEGACIÓN	124
4.6	DIAGRAMA DE COMPONENTES	125
4.7	DIAGRAMA DE DESPLIEGUE	126
4.8	DIAGRAMA DE ARQUITECTURA	127
4.9	PRUEBAS DE LA APLICACIÓN	128
<b>5.</b>	<b>PLAN DE VALICADIÓN</b>	<b>135</b>
<b>6.</b>	<b>EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN</b>	<b>139</b>
<b>7.</b>	<b>VALORACIÓN TÉCNICO-ECONÓMICA-AMBIENTAL</b>	<b>141</b>
<b>8.</b>	<b>CONCLUSIONES</b>	<b>142</b>
<b>9.</b>	<b>RECOMENDACIONES</b>	<b>143</b>
<b>10.</b>	<b>BIBLIOGRAFÍA</b>	<b>145</b>
<b>11.</b>	<b>ANEXOS</b>	<b>146</b>



## ÍNDICE DE TABLAS

### METODOLOGÍA

<b>Tabla 2.1</b> Métodos e instrumentos utilizados	3
--	---

### DESARROLLO DE LA PROPUESTA ALTERNATIVA

<b>Tabla 4.1</b> Requerimientos Funcionales	69
<b>Tabla 4.2</b> Requerimientos no Funcionales	70
<b>Tabla 4.3</b> Molelamiento de casos de uso	73
<b>Tabla 4.4</b> Relación Casos de Uso – Requerimientos	73
<b>Tabla 4.5</b> Descripción del Caso de Uso Administrar Usuario	85
<b>Tabla 4.6</b> Descripción del Caso de Uso Administrar Noticia	89
<b>Tabla 4.7</b> Descripción del Caso de Uso Administrar foro de discusión	92
<b>Tabla 4.8</b> Descripción del Caso de Uso Consultar Noticia	95
<b>Tabla 4.9</b> Descripción del Caso de Uso Participar en foro de discusión	96
<b>Tabla 4.10</b> Descripción del Caso de Uso Participar en videoconferencia	98

### PLAN DE VALIDACIÓN

<b>Tabla 5.1</b> Plan de validación	134
-------------------------------------	-----

### VALORACIÓN TÉCNICA-ECONÓMICA

<b>Tabla 7.1</b> Aproximación del Costo Real del Proyecto	143
---	-----



## ÍNDICE DE FIGURAS

### CAPITULO II

#### ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS CON UML

<b>Figura 2.1</b> Proceso de desarrollo de software	17
<b>Figura 2.2</b> Notación UML para las Clases	20
<b>Figura 2.3</b> Notación UML para Casos de Uso	22
<b>Figura 2.4</b> Notación UML para Asociaciones	22
<b>Figura 2.5</b> Notación UML para Agregación	22
<b>Figura 2.6</b> Notación UML para Composición	23
<b>Figura 2.7</b> Notación UML para Herencia	23
<b>Figura 2.8</b> Notación UML para Paquetes	23
<b>Figura 2.9</b> Elementos de un Diagrama de Casos de Uso	26

### CAPITULO III

#### HERRAMIENTAS DE PROGRAMACIÓN Y DESARROLLO

<b>Figura 3.1</b> Modelo de Procesamiento de Medios	33
<b>Figura 3.2</b> Arquitectura de Alto Nivel de JMF	34
<b>Figura 3.3</b> Captura, procesamiento y presentación de medios basados	



en el tiempo	35
<b>Figura 3.4</b> Modelo de un Player JMF	37
<b>Figura 3.5</b> Modelo de un Processor JMF	38
<b>Figura 3.6</b> Proceso de petición-respuesta de una página JSP	42

### DESARROLLO DE LA PROPUESTA ALTERNATIVA

<b>Figura 4.1</b> Diagrama Modelado del dominio	37
<b>Figura 4.2</b> Diagrama de Casos de Uso	75
<b>Figura 4.3</b> Pantalla de Inicio	76
<b>Figura 4.4</b> Pantalla de Noticias	76
<b>Figura 4.5</b> Pantalla con el contenido de una noticia	77
<b>Figura 4.6</b> Pantalla de Foros	77
<b>Figura 4.7</b> Pantalla para cada tema con sus respuestas	78
<b>Figura 4.8</b> Pantalla de Login	78
<b>Figura 4.9</b> Pantalla para la Cuenta de Usuario	79
<b>Figura 4.10</b> Pantalla para la Cuenta de Administrador	79
<b>Figura 4.11</b> Pantalla para Búsqueda de Usuarios	80
<b>Figura 4.12</b> Pantalla para ingreso y edición de usuarios	80
<b>Figura 4.13</b> Pantalla para Búsqueda de Noticias	81



<b>Figura 4.14</b>	Pantalla para ingreso y edición de noticias	81
<b>Figura 4.15</b>	Pantalla para Búsqueda de Temas	82
<b>Figura 4.16</b>	Pantalla para ingreso y edición de temas	82
<b>Figura 4.17</b>	Pantalla principal de Husky Videoconferency	83
<b>Figura 4.18</b>	Pantalla principal de Husky Videoconferency con sesión iniciada	83
<b>Figura 4.19</b>	Pantalla principal de Husky Videoconferency con lista de usuarios	84
<b>Figura 4.20</b>	Pantalla principal de Husky Videoconferency con chat	84
<b>Figura 4.21</b>	Mensaje de error	85
<b>Figura 4.22</b>	Mensaje de información	85
<b>Figura 4.23</b>	Mensaje de confirmación	85
<b>Figura 4.24</b>	Diagrama de robustez del caso de uso Administrar Usuario	102
<b>Figura 4.25</b>	Diagrama de robustez del curso alternativo A: Texto vacío del caso de uso Administrar usuario	102
<b>Figura 4.26</b>	Diagrama de robustez del curso alternativo B: Usuario no encontrado del caso de uso Administrar usuario	103
<b>Figura 4.27</b>	Diagrama de robustez del curso alternativo C: Nuevo usuario	



del caso de uso Administrar usuario	103
<b>Figura 4.28</b> Diagrama de robustez del curso alternativo D: Eliminar usuario del caso de uso Administrar usuario	103
<b>Figura 4.29</b> Diagrama de robustez del caso de uso Administrar noticia	104
<b>Figura 4.30</b> Diagrama de robustez del curso alternativo A: Texto vacío del caso de uso Administrar noticia	104
<b>Figura 4.31</b> Diagrama de robustez del curso alternativo B: Noticia no encontrada del caso de uso Administrar noticia	104
<b>Figura 4.32</b> Diagrama de robustez del curso alternativo C: Nueva noticia del caso de uso Administrar noticia	105
<b>Figura 4.33</b> Diagrama de robustez del curso alternativo D: Eliminar noticia del caso de uso Administrar noticia	105
<b>Figura 4.34</b> Diagrama de robustez del caso de uso Administrar foro de discusión	106
<b>Figura 4.35</b> Diagrama de robustez del curso alternativo A: Texto vacío del caso de uso Administrar foro de discusión	106
<b>Figura 4.36</b> Diagrama de robustez del curso alternativo B: Tema no encontrado del caso de uso Administrar foro de discusión	107
<b>Figura 4.37</b> Diagrama de robustez del caso de uso Consultar noticia	107



<b>Figura 4.38</b>	Diagrama de robustez del caso de uso Participar en foro de discusión	108
<b>Figura 4.39</b>	Diagrama de robustez del curso alternativo A: Nuevo tema del caso de uso Participar en foro de discusión	108
<b>Figura 4.40</b>	Diagrama de robustez del caso de uso Participar en Videoconferencia	109
<b>Figura 4.41</b>	Diagrama de robustez del curso alternativo A: Texto vacío del caso de uso Participar en videoconferencia	109
<b>Figura 4.42</b>	Diagrama de robustez del curso alternativo B: Usuario no registrado del caso de uso Participar en videoconferencia	110
<b>Figura 4.43</b>	Diagrama de robustez del curso alternativo C: Multimedia no encontrados del caso de uso Participar en videoconferencia	110
<b>Figura 4.44</b>	Diagrama de secuencia del caso de uso Administrar usuario	111
<b>Figura 4.45</b>	Diagrama de secuencia del curso alternativo A: Texto vacío del caso de uso Administrar usuario	112
<b>Figura 4.46</b>	Diagrama de secuencia del curso alternativo B: Usuario no encontrado del caso de uso Administrar usuario	112
<b>Figura 4.47</b>	Diagrama de secuencia del curso alternativo C: Nuevo usuario del caso de uso Administrar usuario	113



<b>Figura 4.48</b>	Diagrama de secuencia del curso alternativo D: Eliminar usuario del caso de uso Administrar usuario	113
<b>Figura 4.49</b>	Diagrama de secuencia del caso de uso Administrar noticia	114
<b>Figura 4.50</b>	Diagrama de secuencia del curso alternativo A: Texto vacío del caso de uso Administrar noticia	115
<b>Figura 4.51</b>	Diagrama de secuencia del curso alternativo B: Noticia no encontrada del caso de uso Administrar noticia	115
<b>Figura 4.52</b>	Diagrama de secuencia del curso alternativo C: Nueva noticia del caso de uso Administrar noticia	116
<b>Figura 4.53</b>	Diagrama de secuencia del curso alternativo D: Eliminar noticia del caso de uso Administrar noticia	116
<b>Figura 4.54</b>	Diagrama de secuencia del caso de uso Administrar foro de discusión	117
<b>Figura 4.55</b>	Diagrama de secuencia del curso alternativo A: Texto vacío del caso de uso Administrar foro de discusión	118
<b>Figura 4.56</b>	Diagrama de secuencia del curso alternativo B: Texto no encontrado del caso de uso Administrar foro de discusión	118
<b>Figura 4.57</b>	Diagrama de secuencia del caso de uso Consultar noticia	119
<b>Figura 4.58</b>	Diagrama de secuencia del caso de uso Participar en foro de discusión.	120



<b>Figura 4.59</b> Diagrama de secuencia del curso alternativo A: Nuevo tema del caso de uso Participar en foro de discusión	121
<b>Figura 4.60</b> Diagrama de secuencia del caso de uso Participar en videoconferencia.	122
<b>Figura 4.61</b> Diagrama de secuencia del curso alternativo A: Texto vacío del caso de uso Participar en videoconferencia	123
<b>Figura 4.62</b> Diagrama de secuencia del curso alternativo B: Usuario no registrado del caso de uso Participar en videoconferencia	123
<b>Figura 4.63</b> Diagrama de secuencia del curso alternativo C: Multimedia no encontrados del caso de uso Participar en videoconferencia	124
<b>Figura 4.64</b> Diagrama de Navegación	125
<b>Figura 4.65</b> Diagrama de Componentes	126
<b>Figura 4.66</b> Diagrama de Despliegue	127
<b>Figura 4.67</b> Diagrama de arquitectura	128



## 5. INTRODUCCIÓN

La Universidad Nacional de Loja, ha llegado a convertirse en uno de los más importantes centros de estudio superior a nivel regional y nacional, por cuanto ofrece carreras presenciales y a distancia, que forman profesionales que contribuyen a resolver problemas de la sociedad en general.

El alma mater cuenta con cinco áreas académicas-administrativas y emplea un modelo pedagógico denominado “Sistema Académico Modular por Objetos de Transformación (SAMOT), muy importante para impartir y generar conocimientos hacia los estudiantes que se forman en esta institución, mediante la investigación formativa desarrollada por los estudiantes desde los primeros módulos de estudio, involucrándose en la problemática social, con la finalidad de generar alternativas de solución.

El acelerado desarrollo de la tecnología, demanda que la Universidad se adentre a este progreso. Es por ello que actualmente cuenta con los equipos y el personal calificado, para aprovechar adecuadamente las nuevas tecnologías en la formación de los estudiantes que asisten a este establecimiento.

La presente investigación denominada “***DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN CON RECURSOS DE VIDEOCONFERENCIA SOBRE INTERNET 2, PARA COLABORACIÓN DE GRUPOS MULTIDISCIPLINARIOS DE PERSONAS***”, tiene como objetivo la implementación del servicio de videoconferencia entre docentes y estudiantes de la Universidad Nacional de Loja.

El desarrollo de la presente investigación se encuentra estructurado de la siguiente manera:



En la primera parte se describe la metodología utilizada en el desarrollo de las actividades planificadas en el proyecto inicial, así como los métodos y técnicas que permitieron el desarrollo de la investigación.

Seguidamente se puede encontrar el sustento teórico que sirvió de base para comprender de mejor manera el problema en estudio y llegar a plantear las alternativas de solución para el mismo. La teoría utilizada aborda los siguientes temas: conceptos, características y ventajas principales de Internet 2, a continuación cuenta con las definiciones y elementos de un sistema de videoconferencia, así como los estándares y codecs empleados para obtener una buena calidad en el servicio. Se detalla la metodología de desarrollo empleada como es la metodología de desarrollo orientado a objetos con UML, que cuenta con los conceptos más importantes y los diagramas de mayor uso para los proyectos de software. Finalmente se explican los puntos más relevantes de las herramientas de software empleadas para el desarrollo del presente proyecto.

En cuanto a la propuesta alternativa, se presenta la fase de análisis, diseño e implementación de las aplicaciones que permiten la comunicación mediante videoconferencia, cuenta con el análisis de requerimientos, descripción de casos de uso, diagramas de clases, casos de uso, robustez, secuencia, navegación, despliegue, componentes y arquitectura.

Se llevó a cabo una fase de pruebas de las aplicaciones para comprobar su correcto funcionamiento, mediante un cuestionario dirigido a los usuarios del sistema de videoconferencia obtuvimos resultados que nos permitieron evaluar los programas.

Así como también se realizó un plan de validación.

Se incluye además la evaluación de los objetivos planteados, para comprobar el cumplimiento de cada uno de los mismos.



Presentamos también una estimación técnico-económica de los materiales y recursos empleados, con lo que se llegó a determinar el costo real del proyecto.

Finalmente se describe las conclusiones y recomendaciones obtenidas a lo largo del presente proyecto de investigación.



## 6. METODOLOGÍA

El presente trabajo investigativo estuvo dirigido al desarrollo de un sistema de videoconferencia que permita el vínculo a distancia entre estudiantes y docentes de nuestra universidad.

Para la ejecución se utilizó métodos, técnicas y procedimientos mismos que se explican a continuación:

**Tabla 2.1 Métodos e instrumentos utilizados**

TAREA REALIZADA	METODOLOGÍA
<ul style="list-style-type: none"><li>- Investigación bibliográfica y en Internet acerca de Internet 2</li><li>- Investigación bibliográfica acerca del protocolo de Internet versión IPv4 y IPv6</li><li>- Investigación en Internet acerca de videoconferencia sobre redes ip</li><li>- Investigación en libros e Internet acerca de estándares y codecs para videoconferencia</li><li>- Investigación en Internet sobre herramientas para el desarrollo de</li></ul>	<ul style="list-style-type: none"><li>- <b>Método sintético</b> (Consiste en la reconstrucción del todo uniendo sus partes que estaban separadas, facilitando la comprensión cabal del asunto que se estudia o analiza). Este método nos sirvió para obtener solo la información mas relevante que nos permitió conocer el funcionamiento de un sistema de videoconferencia sobre redes ip</li></ul> <p>Técnicas e instrumentos utilizados:</p> <ul style="list-style-type: none"><li>- <b>Observación directa.</b>- Consiste en la inspección y estudio por medio de los sentidos de las características más sobresalientes del hecho o fenómeno por investigar</li><li>- <b>Observación indirecta.</b>- Consiste en el conocimiento del hecho o fenómeno a</li></ul>



<p>sistemas de videoconferencia</p>	<p>través de las observaciones realizadas anteriormente por otra persona</p> <ul style="list-style-type: none"><li>- <b>Lectura comprensiva.</b>- Consiste en el conocimiento ordenado y sistemático de un aspecto de la realidad o de los acontecimientos, hechos o ideas relacionadas con un tema específico</li></ul>
<ul style="list-style-type: none"><li>- Entrevistas con personas con conocimientos acerca de videoconferencia y redes</li><li>- Aplicación de encuestas para obtener los requerimientos que debe cumplir un sistema de videoconferencia</li></ul>	<p>Técnicas e instrumentos utilizados:</p> <ul style="list-style-type: none"><li>- <b>Entrevista.</b>- Consiste en el diálogo entre dos personas: el entrevistador (investigador) y el entrevistado. Se aplicó esta técnica con el fin de obtener información de personas entendidas en el campo de investigación</li><li>- <b>Encuesta.</b>- Consiste en entregar a las personas un cuestionario que debe ser llenado por ellas libremente. Mediante esta técnica se obtuvieron los requerimientos con que debe cumplir un sistema para videoconferencia</li><li>- Observación directa</li><li>- Lectura comprensiva</li></ul>
<ul style="list-style-type: none"><li>- Análisis de la información obtenida en las entrevistas y encuestas</li><li>- Identificación de los requerimientos de los usuarios para un sistema de videoconferencia</li></ul>	<ul style="list-style-type: none"><li>- <b>Método sintético</b> .- Nos permitió obtener solo la información mas real y confiable que nos sirvió para plantear nuestros requerimientos para el sistema de videoconferencia</li><li>- <b>Método inductivo.</b>- Consiste en un proceso mediante el cual se parte del estudio de casos, hechos o fenómenos particulares para llegar al descubrimiento de un principio o ley general que lo rige. Este método nos permitió tener un conocimiento general de un sistema de videoconferencia partiendo de sus</li></ul>



	<p>requerimientos</p> <p>Técnicas e instrumentos utilizados:</p> <ul style="list-style-type: none"><li>- Lectura comprensiva</li></ul>
<ul style="list-style-type: none"><li>- Modelado de la aplicación de videoconferencia Husky</li><li>- Videoconferency</li></ul>	<ul style="list-style-type: none"><li>- <b>Modelado Orientado a Objetos con UML.</b>- Es una Metodología de Desarrollo de Software que emplea el Lenguaje Unificado de Modelado como herramienta necesaria para obtener los planos del software para su programación. Nos permitió obtener todos los diagramas que se usaron en la programación de la aplicación, así como el diseño de las pantallas y el diseño de la base de datos</li></ul> <p>Técnicas e instrumentos utilizados:</p> <ul style="list-style-type: none"><li>- Diagrama de clases</li><li>- Diagrama de casos de uso</li><li>- Diagrama de robustez</li><li>- Diagrama de secuencia</li></ul>
<ul style="list-style-type: none"><li>- Obtener en Internet y configurar adecuadamente el software y las herramientas necesarias para la programación de la aplicación</li></ul>	<ul style="list-style-type: none"><li>- <b>Método analítico.</b>- Consiste en descomponer en partes algo complejo, para mostrarlas, describirlas, numerarlas y para explicar las causas de los hechos, fenómenos que constituyen el todo. Este método nos sirvió para evaluar con cuidado cual es la mejor herramienta que permita la programación de la aplicación de una manera adecuada a los requerimientos</li></ul> <p>Técnicas e instrumentos utilizados:</p> <ul style="list-style-type: none"><li>- Observación directa</li></ul>



	<ul style="list-style-type: none"><li>- Observación indirecta</li><li>- Lectura comprensiva</li></ul>
<ul style="list-style-type: none"><li>- Programación de la aplicación</li><li>- Pruebas del sistema de videoconferencia Husky Videoconferency</li><li>- Depuración de errores</li></ul>	<ul style="list-style-type: none"><li>- <b>Modelo de programación en espiral.</b>- Nos permitió realizar avances sustanciales de la aplicación al tiempo que se puede regresar y retomar acciones de rediseño en base a sus resultados</li><li>- <b>Método científico.</b>- Es el conjunto de procedimientos lógicos, que sigue la investigación para descubrir las relaciones internas y externas de los procesos de realidad natural y social. El mismo nos permitió a través de software y hardware relacionado a videoconferencia realizar pruebas que nos llevaron a definir a detectar y depurar con exactitud errores existentes en la programación y lograr la configuración final de:<ul style="list-style-type: none"><li>• <b>Husky Videoconferency.</b>- Aplicación Web de administración del sistema</li><li>• <b>Husky Videoconferency Server.</b>- Aplicación de escritorio del lado del docente para la comunicación mediante audio, video y texto.</li><li>• <b>Husky Videoconferency Client.</b>- Aplicación de escritorio del lado del estudiante para la comunicación mediante audio, video y texto.</li></ul></li></ul> <p>Técnicas e instrumentos utilizados:</p> <ul style="list-style-type: none"><li>- Observación directa</li><li>- Observación indirecta</li><li>- Lectura comprensiva</li></ul>



## 7. ARCO TEÓRICO

### CAPITULO I

#### INTERNET 2 Y VIDEOCONFERENCIA

##### 1.1 INTRODUCCIÓN

El uso de Internet como herramienta educativa y de investigación científica ha crecido aceleradamente debido a la ventaja que representa el poder acceder a grandes bases de datos, la capacidad de compartir información entre colegas y facilitar la coordinación de grupos de trabajo.

Los seres humanos estamos visualmente orientados. Por tanto las imágenes pueden considerarse como el medio de comunicación más efectivo por contener una mayor cantidad de información cuando se le compara con las palabras escritas o ideas conceptuales.

Dada la sofisticación del sistema de la visión humana, la predilección del ser humano por las imágenes es sorprendente, no sólo una gran parte del cerebro esta dedicada a la visión y al análisis visual sino que también la capacidad de transporte de información (el ancho de banda) de nuestro sistema visual es mucho mayor que el de cualquier otro de nuestros sentidos.

De todas las imágenes y pinturas conocidas, el rostro humano es la más importante como fuente de información. Cuando hablamos cara a cara con otra persona, obtenemos mayor información de las expresiones faciales, más que de sus palabras o calidad de voz combinadas.

El problema es que en el ambiente global de los negocios de ahora las comunicaciones cara a cara han llegado a ser una práctica costosa, con un alto consumo de tiempo por lo que es, frecuentemente omitida. Se hace uso entonces de medios como el teléfono, el fax o el modem para satisfacer las necesidades de



comunicación corporativas. "La videoconferencia ofrece hoy en día una solución accesible a esta necesidad de comunicación, con sistemas que permiten el transmitir y recibir información visual y sonora entre puntos o zonas diferentes evitando así los gastos y pérdida de tiempo que implican el traslado físico de la persona, todo esto a costos cada vez más bajos y con señales de mejor calidad". Estas ventajas hacen a la videoconferencia el segmento de mayor crecimiento en el área de las telecomunicaciones.

## 1.2 DEFINICIÓN DE INTERNET 2

Internet 2 es una red de cómputo con capacidades avanzadas separada de la Internet comercial actual. Su origen se basa en el espíritu de colaboración entre las universidades de diferentes países del continente americano y su objetivo principal es desarrollar la próxima generación de aplicaciones telemáticas para facilitar las misiones de investigación y educación de las universidades, además de ayudar en la formación de personal capacitado en el uso y manejo de redes avanzadas de cómputo<sup>1</sup>.

Permite desarrollar tecnología y aplicaciones avanzadas en la Internet, vitales para las misiones de investigación y educación de las instituciones de educación superiores. Internet 2 trabaja para hacer posibles aplicaciones tales como la telemedicina, bibliotecas digitales y laboratorios virtuales que no serían posibles con la tecnología del Internet de hoy. Como proyecto de la Corporación Universitaria para el Desarrollo Avanzado de la Internet (UCAID), el proyecto Internet 2 no es una simple red aparte, sino que une las aplicaciones de la red y los esfuerzos de desarrollo en ingeniería con redes avanzadas de universidades de los países participantes.

La misión de Internet 2 es desarrollar e implantar aplicaciones y tecnologías que demanden redes avanzadas y de alto rendimiento, para la investigación en tecnología y educación, acelerando la creación de Internet del mañana<sup>2</sup>.

La ingeniería en Internet 2 tiene como objetivo minimizar los costos de acceso a las universidades participantes, proporcionando circuitos de conexión de alta velocidad.

<sup>1</sup> MORENO, Fernando – Internet 2 – [www.monografias.com](http://www.monografias.com)

<sup>2</sup> TAMA WONG, Luis A. – Debate de Redes Avanzadas – [www.ieee.org](http://www.ieee.org)



Además, mediante una arquitectura flexible es posible una interconexión de otros servicios regionales. Para servicios de áreas extensas un solo servicio será necesario: el gigaPoP (gigabits Point of Presence), que es un punto de interconexión de tecnología avanzada y alta capacidad, donde todos los participantes de Internet 2 pueden intercambiar tráfico de servicios avanzados entre sí<sup>3</sup>.

## 1.4 PROTOCOLO DE INTERNET 2

### 1.4.1 Tecnología IPv4

Un Protocolo de Internet (IP, según su sigla en inglés) es un conjunto de normas técnicas que define la forma en que las computadoras se comunican dentro de una red.

IPv4 fue la primera versión del Protocolo de Internet de uso masivo, y todavía se utiliza en la mayoría del tráfico actual de Internet. Existen algo más de 4.000 millones de direcciones IPv4. Si bien es una cantidad importante de direcciones IP, no es suficiente para cubrir todas las necesidades que irán surgiendo en el futuro.

IPv4 es la versión 4 del Protocolo IP (Internet Protocol). Esta fue la primera versión del protocolo que se implementó extensamente, y forma la base de Internet.

IPv4 usa direcciones de 32 bits, limitándola a  $2^{32} = 4.294.967.296$  direcciones únicas, muchas de las cuales están dedicadas a redes locales (LANs). Por el crecimiento enorme que ha tenido del Internet (mucho más de lo que esperaba, cuando se diseñó IPv4), combinado con el hecho de que hay desperdicio de direcciones en muchos casos (ver abajo), ya hace varios años se vio que escaseaban las direcciones IPv4.

Esta limitación ayudó a estimular el impulso hacia IPv6, que está actualmente en las primeras fases de implementación, y se espera que termine reemplazando a IPv4.

#### 1.4.1.1 Broadcast y Multicast

---

<sup>3</sup> TORRES MORENO, Juan M. – Internet 2, Las Nuevas Redes del Futuro– [www.lania.mx](http://www.lania.mx)



Broadcast, en castellano difusión, es un modo de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo

El protocolo IP en su versión 4 también permite la difusión de datos. En este caso no existe un medio de transmisión compartido, no obstante, se simula un comportamiento similar.

La difusión en IPv4 no se realiza a todos los nodos de la red porque colapsaría las líneas de comunicaciones debido a que no existe un medio de transmisión compartido. Tan sólo es posible la difusión a subredes concretas dentro de la red, generalmente, aquellas bajo el control de un mismo enrutador. Para ello existen dos modalidades de difusión:

#### **Difusión limitada (limited broadcast)**

Consiste en enviar un paquete de datos IP con la dirección 255.255.255.255. Este paquete solamente alcanzará a los nodos que se encuentran dentro de la misma red física subyacente. En general, la red subyacente será una red de área local o un segmento de ésta.

#### **Multidifusión (multicast)**

La multidifusión utiliza un rango especial de direcciones denominado rango de clase D. Estas direcciones no identifican nodos sino redes o subredes. Cuando se envía un paquete con una dirección de multidifusión, todos los enrutadores intermedios se limitan a re-enviar el paquete hasta el enrutador de dicha subred. Éste último se encarga de hacerlo llegar a todos los nodos que se encuentran en la subred.

Aquella dirección que tiene todos y cada uno de los bits de la parte de dirección de máquina con valor 1 es una dirección de multidifusión. Por ejemplo, en una red 192.168.11.0/24, la dirección de broadcast es 192.168.11.255. El valor de host 255 en 192.168.11.255 se codifica en binario con sus ocho bits a 1: 11111111.

Aún hoy día la multidifusión se utiliza únicamente como experimento. Existe una propuesta de implementación de videoconferencia utilizando multidifusión, sin embargo, se han estandarizado otros mecanismos.



### 1.4.2 Tecnología IPv6

El notable crecimiento de Internet basado en Internet Protocol versión 4 (IPv4) ha hecho relucir algunas de sus limitaciones fundamentales:

- ❖ IPv4 ofrece solamente 32 bits para direccionamiento.
- ❖ Escasez de direcciones IPv4 Clase B.
- ❖ Gran cantidad de sistemas conectados.
- ❖ Demasiadas entradas en las tablas de enrutamiento.
- ❖ Incremento progresivo del tiempo de búsqueda, DNS, etc.
- ❖ Situación salvada temporalmente mediante conservación, recuperación y técnicas como NAT (Network Address Translation).

Internet Protocol versión 6 (IPv6) enfrenta estos problemas y provee funcionalidad y servicios adicionales mejorados:

- ❖ Expansión de capacidades de enrutamiento y direccionamiento.
- ❖ Solución al problema de direcciones: 128 bits.
- ❖ Soporte de mejores estructuras jerárquicas.
- ❖ Direcciones Unicast, Anycast y Multicast
- ❖ No hay BROADCAST.

Internet 2 está implementando principalmente IPv6, pero continuará dando soporte a IPv4. Sin embargo, hay obstáculos para desplegar IPv6, incluyendo el soporte de routers y TCP/IP, como las actualizaciones de hosts. Esta implementación estimulará el crecimiento de las universidades, proveyéndoles más direcciones IP, necesarias para implementar Internet 2.



### 1.4.3 Ventajas de IPv6

En IPv6, el protocolo IP ha sido rediseñado totalmente, aunque basándose en el actual. Básicamente, la versión 6 presenta las siguientes mejoras respecto a su predecesora:

- Direcciones ampliadas: en IPv6, las direcciones se han ampliado a 128 bits frente a los 32 bits de IPv4.
- Desaparecen los paquetes broadcast (destinados a todos los nodos) y se utilizan en su lugar paquetes multicast (destinados a varios nodos) y anycast (destinada a un nodo de entre varios).
- Configuración más simple.
- Capacidad de encriptación y autenticación.
- Posibilidad de marcar los tráficos.

### 1.4.4 Características principales de IPv6

- Mayor espacio de direcciones.
- Seguridad intrínseca en el núcleo del protocolo
- Calidad de servicio y clase de servicio.
- Multicast Envío de UN paquete a Un receptor a Un GRUPO de receptores.
- Anycast Envío de UN paquete a Un receptor dentro de Un GRUPO.
- Paquetes IP eficientes y extensibles, sin que haya fragmentación en los encaminadores (routers), alineados a 64 bits, y con una cabecera de longitud fija, más simple, que agiliza su procesamiento por parte del encaminador.
- Posibilidad de paquetes con carga útil (datos) de más de 65.535 bytes.
- Encaminado (enrutado más eficiente en el troncal (backbone) de la red, debido a una jerarquía de direccionamiento basada en la agregación.
- Renumeración y multi-homing, que facilita el intercambio del proveedor de servicios.
- Características de movilidad.



#### 1.4.4.1 Multicast y unicast

Otra modificación importante que nos encontramos respecto al protocolo actual es el hecho de que también cambia la forma de representar las máscaras de red. En IPv4 existían dos formas: la más habitual, mediante cuatro números (por ejemplo, 255.255.255.0), y otra, menos conocida, mediante un número añadido a la dirección de la forma «/n», donde «n» representa el número de 1 seguidos que tiene la representación en binario de la máscara (en el ejemplo anterior, sería «/24», dado que poniendo la máscara en binario sería 111111111111111111111111111100000000, es decir, 24 unos y 8 ceros).

Debido a que en IPv6 las máscaras de red son muy largas, se ha optado por utilizar sólo esta segunda representación. Así, una dirección de red con su máscara podría quedar: FE80::/64.

De esta forma, podría decirse que la dirección de red está formada por el número de bits indicado como /n, y que la del nodo está formada por 128-n bits.

### 1.5 VIDEOCONFERENCIA

#### 1.5.1 Definición

Al sistema que nos permite llevar a cabo el encuentro de varias personas ubicadas en sitios distantes, y establecer una conversación como lo harían si todas se encontraran reunidas en una sala de juntas se le llama sistema de "videoconferencia".

Para el presente proyecto se empleará el término "videoconferencia" para describir la comunicación en doble sentido ó interactiva entre dos puntos geográficamente separados utilizando audio y video.<sup>4</sup>

#### 1.5.2 Elementos de un sistema de videoconferencia

Para fines de estudio y de diseño los sistemas de videoconferencia suelen subdividirse en tres elementos básicos que son:

---

<sup>4</sup> Videoconferencia – [www.monografias.com](http://www.monografias.com)



- La red de comunicaciones
- La sala de videoconferencia
- El codec.

A su vez la sala de videoconferencia se subdivide en cuatro componentes esenciales: el ambiente físico, el sistema de video, el sistema de audio y el sistema de control.

A continuación se describe brevemente cada uno de los elementos básicos de que consta un sistema de videoconferencia.

### **1.5.2.1 La red de comunicaciones**

Para poder realizar cualquier tipo de comunicación es necesario contar primero con un medio que transporte la información del transmisor al receptor y viceversa o paralelamente (en dos direcciones). En los sistemas de videoconferencia se requiere que este medio proporcione una conexión digital bidireccional y de alta velocidad entre los dos puntos a conectar.

El número de posibilidades que existen de redes de comunicación es grande, pero se debe señalar que la opción particular depende enteramente de los requerimientos del usuario.

### **1.5.2.2 La sala de videoconferencia**

La sala de videoconferencia es el área especialmente acondicionada en la cual se alojará el personal de videoconferencia, así como también, el equipo de control, de audio y de video, que permitirá el capturar y controlar las imágenes y los sonidos que habrán de transmitirse hacia el(los) punto(s) remoto(s).

El nivel de confort de la sala determina la calidad de la instalación. La sala de videoconferencia perfecta es la sala que más se asemeja a una sala normal para conferencias; aquellos que hagan uso de esta instalación no deben sentirse intimidados por la tecnología requerida, más bien deben sentirse a gusto en la instalación. La tecnología no debe notarse o debe de ser transparente para el usuario.



### 1.5.2.3 El codec

Las señales de audio y video que se desean transmitir se encuentran por lo general en forma de señales analógicas, por lo que para poder transmitir esta información a través de una red digital, ésta debe de ser transformada mediante algún método a una señal digital, una vez realizado esto se debe de comprimir y multiplexar estas señales para su transmisión. El dispositivo que se encarga de este trabajo es el CODEC (Codificador/Decodificador) que en el otro extremo de la red realiza el trabajo inverso para poder desplegar y reproducir los datos provenientes desde el punto remoto. Existen en el mercado equipos modulares que junto con el CODEC, incluyen los equipos de video, de audio y de control, así como también equipos periféricos como pueden ser:

- Proyector de video-diapositivas
- Computador
- Videocámara
- Altavoces y/o auriculares
- Micrófono

### 1.5.3 Estándar H323 para Videoconferencia

El estándar H.323 proporciona una base para las comunicaciones de audio, video y datos a través de una red IP como Internet. Los productos que cumplen con el estándar H.323 pueden ínter operar con los productos de otros, permitiendo de esta manera que los usuarios puedan comunicarse sin preocuparse con problemas de compatibilidad.

H.323 es un estándar bajo el amparo de la ITU, es un conjunto de estándares para la comunicación multimedia sobre redes que no proporcionan calidad de servicio (QoS). Estas redes son las que predominan hoy en todos los lugares, como redes de paquetes conmutadas TCP/IP e IP sobre Ethernet, Fast Ethernet y Token Ring. Por esto, los estándares H.323 son bloques importantes de construcción para un amplio rango de aplicaciones basadas en redes de paquetes para la comunicación multimedia y el trabajo colaborativo.

El estándar tiene amplitud e incluye desde dispositivos específicos hasta tecnologías embebidas en ordenadores personales, además de servir para comunicación punto-



punto o conferencias multi-punto. H.323 habla también sobre control de llamadas, gestión multimedia y gestión de ancho de banda, además de los interfaces entre redes de paquetes y otras redes (por ejemplo RTC).

Este tipo de videoconferencia podría utilizarse, por ejemplo, para establecer discusiones entre los individuos participantes en un proyecto.<sup>5</sup>

---

<sup>5</sup> Universidad de Jaén – Servicio de Videoconferencia. Guía del Servicio – [www.ujaen.es](http://www.ujaen.es)



## CAPITULO II

### ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS CON UML

#### 2.1 INTRODUCCIÓN

Todo gira en torno de una visión, un sistema complejo toma forma cuando alguien tiene la visión de cómo la tecnología puede mejorar las cosas. Los desarrolladores tienen que entender completamente la idea y mantenerla en mente mientras crean el sistema que le dé forma.

El éxito de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quien tiene la idea y el desarrollador.

UML (*Unified Modeling Language*) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh.

Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

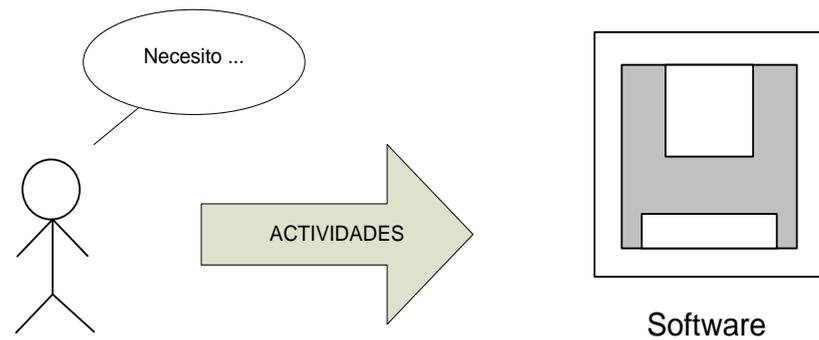
#### 2.2 PROCESO DE DESARROLLO DE SOFTWARE

##### 2.2.1 Informática

Ciencia que involucra un conjunto de actividades para el desarrollo de Sistemas de Información

##### 2.2.2 Proceso de desarrollo

Conjunto de actividades encaminadas a transformar los requerimientos de los usuarios, a software que resuelva un problema



**Figura 2.1 Proceso de desarrollo de software**

El proceso de desarrollo de software se lo debe seguir para todo tipo de aplicaciones tanto para aplicaciones de propósito general como para aquellas de propósito específico

### **2.2.3 Importancia del proceso de desarrollo de software**

- Permite tener un control y seguimiento del proyecto informático.
- Nos asegura aplicaciones de calidad.
- Ayuda a la optimización de recursos.
- Experiencia para futuros proyectos

### **2.2.4 Modelo**

Para producir software que cumpla su propósito hay que obtener los requisitos del sistema, esto se consigue conociendo de una forma disciplinada a los usuarios y haciéndolos participar de manera activa para que no queden “cabos sueltos”. Para conseguir un software de calidad, que sea duradero y fácil de mantener hay que idear una sólida base arquitectónica que sea flexible al cambio. Para desarrollar software rápido y eficientemente, minimizando el trabajo de remodificación y evitando crear miles de líneas de código inútil hay que disponer, además de la gente y las herramientas necesarias, de un enfoque apropiado.



Modelo, descripción que se hace acerca de los elementos que intervienen en el sistema y que permiten entender al mismo.

#### 2.2.4.1 Etapas del proceso de desarrollo de software

1. **Análisis de requerimientos.**- Saber los requisitos y expectativas que tiene el usuario. Identificar los problemas a través de la necesidades
2. **Análisis del sistema.**- Clarificación y entendimiento por parte del analista de cómo está el sistema
3. **Diseño.**- Indicar como se va a solucionar el problema descubierto en las etapas anteriores, de forma lógica
4. **Implementación.**- Traducción de los modelos del diseño a un lenguaje de programación, aplicando la tecnología disponible
5. **Explotación.**- Puesta en marcha del sistema para cubrir las necesidades para las cuales fue concebido

Todas estas etapas se realizan en forma iterativa e incremental, es decir que se encontrará nuevas características al sistema conforme avanza el proyecto y se obtiene un refinamiento (mejora) en la próxima iteración.

#### 2.2.5 Programación orientada a objetos

##### 2.2.5.1 Orientación a objetos

La Programación Orientada a Objetos es una técnica de programación cuyo soporte fundamental es el objeto. Un objeto es una extensión de un tipo abstracto de datos (TAD). Un TAD es un tipo definido por el usuario, que encapsula un conjunto de datos y las operaciones sobre estos datos.

La filosofía orientada a objetos considera que todo es un objeto, permite obtener sistemas apegados a la realidad, rehusables, reducir la mantenibilidad.



### 2.2.5.2 Principios y fundamentos de programación orientada a objetos

- **Abstracción.-** Abstraer los objetos en la mente obteniendo sus esenciales características, también se llama conceptualización.
- **Encapsulación.-** Agrupar los objetos relacionados a esa entidad. La ocultación de la complejidad del detalle. Agrupar las características esenciales de un mismo objeto.
- **Ocultación de la información.-** La restricción del acceso externo por parte de cualquier entidad. Es la manera de mostrar la información.
- **Herencia.-** Una entidad hija hereda características y comportamientos de una entidad padre.
  - *Herencia simple.-* Uno o varios hijos tienen un solo padre
  - *Herencia múltiple.-* Uno o varios hijos tienen dos o más padres
- **Polimorfismo.-** Diferentes entidades pueden responder al mismo mensaje de diferentes maneras.
  - Una entidad muchas formas
  - Un objeto presenta diferentes comportamientos de acuerdo al contexto en que se encuentra
- **Comunicación.-** Se da a través de mensajes, por tanto un mensaje dice lo que debe hacer una entidad<sup>6</sup>

## 2.3 Lenguaje Unificado de Modelado UML

### 2.3.1 Introducción

UML es un lenguaje por que proporciona un vocabulario y las reglas para utilizarlo, y además es un lenguaje de modelado lo que significa que el vocabulario y las reglas se utilizan para la representación conceptual y física del sistema.

UML es un lenguaje que nos ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante

---

<sup>6</sup> Apuntes sobre “Desarrollo de Sistemas Orientado a Objetos con UML”



el desarrollo ya que al ser estándar, los modelos podrán ser interpretados por personas que no participen en su diseño sin ninguna ambigüedad. En este contexto, UML sirve para especificar modelos concretos, no ambiguos y completos.

## 2.3.2 Elementos Estructurales

### 2.3.2.1 Clases

Clase también llamadas conceptos, son una colección o agrupación de objetos que tienen las mismas características y comportamientos. Es una conceptualización de objetos, una plantilla para crear objetos.

#### 2.3.2.1.1 Clase abstracta

Significa que esta clase no tiene instancias de objetos, es decir, que no se pueden crear objetos de esa clase, lo que nos obliga a crear instancias hijas

#### 2.3.2.1.2 Clase concreta

Es aquella a partir de la cual se pueden crear objetos

La clase dice a los objetos:

- Que información van a contener
- Que es lo que pueden hacer

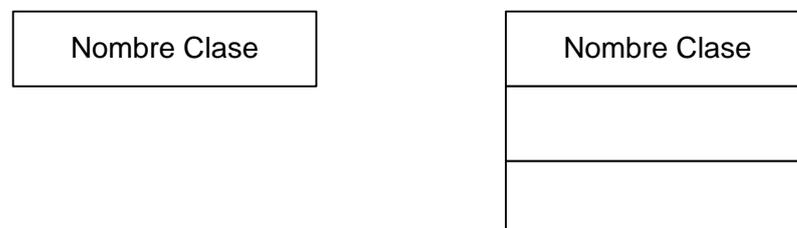


Figura 2.2 Notación UML para las Clases

### 2.3.2.2 Objeto



Tienen existencia, los objetos se crean a partir de las clases. Un objeto se crea al instanciar una clase

Los objetos:

- Conocen algo
- Pueden hacer algo

### 2.3.2.2.1 Atributo

Son las características que tienen los objetos, los atributos indican a los objetos lo que deben conocer. Por lo general se aconseja que todos los atributos de una clase tengan visibilidad privada.

Sintaxis UML:

Visibilidad          nombre\_atributo:      tipo\_dato (clase)

+                    público

# protegido

- privado

### 2.3.2.2.2 Método

Es una operación que pueden realizar las entidades. Nos sirven para operar con los atributos, los métodos son la única forma de acceder a los atributos, los métodos que nos sirven para acceder a la información se llaman métodos Setters and Getters, los conocidos set y get.

Sintaxis UML:

Visibilidad          nombre\_método(lista\_parámetros): tipo\_dato(clase) a devolver

+ público



# protegido

- privado

Lista\_parámetros => nombre\_parámetro: tipo\_dato(clase)

### 2.3.2.3 Interfaz

Una interfaz es una colección de operaciones que especifican un servicio de una determinada clase o componente. Una interfaz describe el comportamiento visible externamente de ese elemento, puede mostrar el comportamiento completo o solo una parte del mismo.

### 2.3.2.4 Colaboración

Define una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos. Las colaboraciones tienen una dimensión tanto estructural como de comportamiento. Una misma clase puede participar en diferentes colaboraciones.

### 2.3.2.5 Casos de Uso

Un caso de uso es la descripción de un conjunto de acciones que un sistema ejecuta y produce un determinado resultado que es de interés para un actor particular. Un caso de uso se utiliza para organizar los aspectos del comportamiento en un modelo. Un caso de uso es realizado por una colaboración.

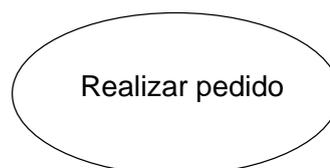


Figura 2.3 Notación UML para Casos de Uso



### 2.3.3 Relación de Asociación

Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos. La agregación es un tipo especial de asociación y representa una relación estructural entre un todo y sus partes.

La asociación se representa con una línea como sigue:

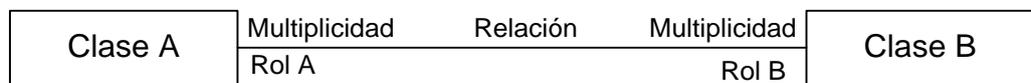


Figura 2.4 Notación UML para Asociaciones

#### 2.3.3.1 Tipos de asociaciones

- **Asociación simple.-** Es la relación más simple, relación que existe entre objetos totalmente independientes uno del otro.
- **Agregación.-** (“es parte de”) Se da entre un todo y sus partes de tal manera que las partes al formar el todo no van a perder su identidad. Se representa con una línea con un diamante apuntado siempre al todo, de la siguiente manera:

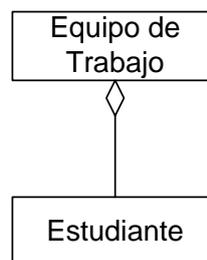


Figura 2.5 Notación UML para Agregación

- **Composición.-** (“es parte de”) Se da entre un todo y sus partes, pero las partes al formar el todo pierden su identidad. Se representa con una línea con un diamante relleno apuntando al todo, como sigue:

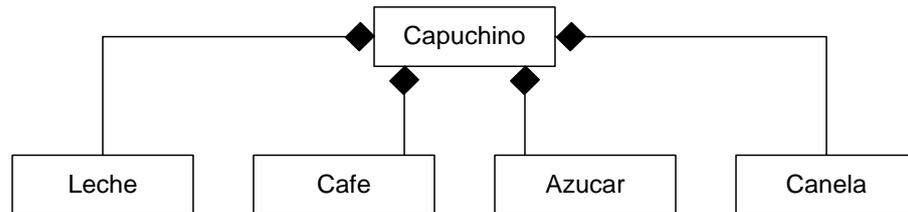


Figura 2.6 Notación UML para Composición

- **Herencia.-** (“es un”, “es una”) Las clases hijas heredan propiedades y comportamientos de las clases padres, a las clases padre también se las conoce como clases base. La herencia también se la llama generalización o derivación.

Se representa como sigue:

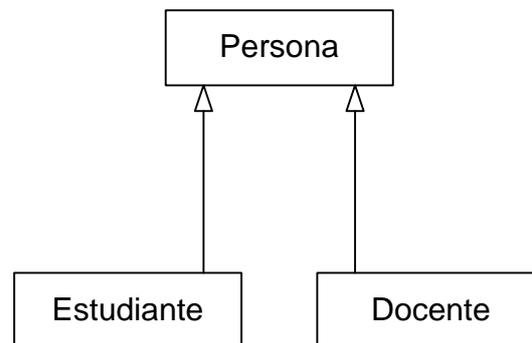
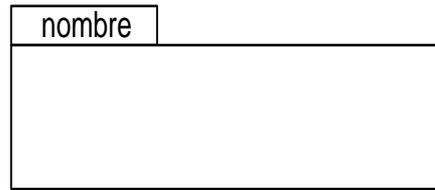


Figura 2.7 Notación UML para Herencia

- ✓ **Herencia simple.-** Cuando las clases hijas tienen solo un padre
- ✓ **Herencia múltiple.-** Cuando las clases hijas tienen dos o más padres
- ✓ **Herencia jerárquica o jerarquía de clases.-** Es un conjunto de clases relacionadas a través de la herencia. Para modelar se debe tomar en cuenta los atributos y métodos que se heredan y que sean similares.
- **Paquete.-** Un paquete es una agrupación de información relacionada, una agrupación de clases que están totalmente relacionadas. Nos sirven para agrupar y organizar clases. Se representa como sigue:



**Figura 2.8 Notación UML para Paquetes**

### **2.3.4 Diagramas**

Los diagramas se utilizan para representar diferentes perspectivas de un sistema de forma que un diagrama es una proyección del mismo.

#### **2.3.4.1 Diagramas de Clases**

Es usado en UML, es una representación gráfica en la que se detalla asociaciones, conceptos, herencias, de un sistema o problema.

Se los usa para el modelamiento del dominio de la aplicación (modelo conceptual). El modelo del dominio de la aplicación es la representación gráfica solo de las clases que intervienen en el problema. Con el modelamiento del dominio se puede decir como se entiende el contexto de la aplicación.

El diagrama de clases también se lo usa para el modelamiento estático, es decir, para el diagrama de clases final que nos va a servir para la implementación. Contiene el modelo del dominio y las nuevas clases del diseño.

Se puede tener uno o mas diagramas de clases tanto en el dominio como en el modelamiento estático.

#### **2.3.4.2 Diagramas de Casos de Usos**



Muestran un conjunto de casos de uso y actores (tipo especial de clases) y sus relaciones. Cubren la vista estática de los casos de uso y son especialmente importantes para el modelado y organización del comportamiento.

#### **2.3.4.3 Diagramas de Secuencia y de Colaboración**

Tanto los diagramas de secuencia como los diagramas de colaboración son un tipo de diagramas de interacción. Constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar unos objetos a otros. Cubren la vista dinámica del sistema. Los diagramas de secuencia enfatizan el ordenamiento temporal de los mensajes mientras que los diagramas de colaboración muestran la organización estructural de los objetos que envían y reciben mensajes. Los diagramas de secuencia se pueden convertir en diagramas de colaboración sin pérdida de información, lo mismo ocurren en sentido opuesto.

#### **2.3.4.4 Diagramas de Componentes**

Muestra la organización y las dependencias entre un conjunto de componentes. Cubren la vista de la implementación estática y se relacionan con los diagramas de clases ya que en un componente suele tener una o más clases, interfaces o colaboraciones.

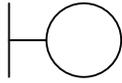
#### **2.3.4.5 Diagrama de paquetes**

Es la representación gráfica de la organización de paquetes y sus clases.

#### **2.3.4.6 Diagrama de robustez**

Es una descripción gráfica del texto del caso de uso. Para hacer los diagramas de robustez se debe haber hecho los casos de uso y el prototipo de pantallas. Por cada caso de uso tendremos un diagrama de robustez.

El diagrama de robustez tiene tres elementos:



Boundary objects.- Objetos de frontera, sirven como nexo de comunicación entre un actor y un sistema. Los objetos de frontera son considerados todos los que pertenecen a una interfaz gráfica de usuario.



Entity objects.- Objetos de entidad, son aquellos que vienen desde el modelo del dominio. Son aquellos objetos que almacenan información, como:

- ✓ Trascendentes.- Almacenan información por un periodo corto de tiempo, luego desaparecen.
- ✓ Persistentes.- Almacenan información por un periodo largo de tiempo como las bases de datos.



Control objects.- Objetos de control o controladores, son como una ventana de comunicación entre los objetos de frontera y los objetos de entidad. Nos ayudan a capturar las políticas y reglas del negocio. Con ellos podemos ver el comportamiento del caso de uso, casi siempre los objetos de control van a pasar a ser métodos de una clase, rara vez pasan a ser clases.

### 2.3.5 Modelamiento de Casos de Uso

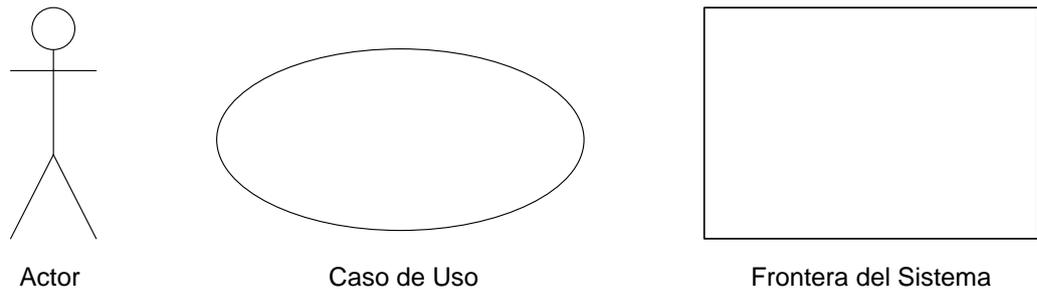
Su esencia es de documentar el comportamiento que va a tener un sistema, por lo tanto se van a identificar actores, casos de uso y relaciones. El comportamiento se lo obtiene en base a los requerimientos. Lo más difícil es encontrar los casos de uso y sus actores.

Los casos de uso nos llevan siempre a la consecución de una meta.



Se debe hacer:

### 2.3.5.1 Diagrama de casos de uso



**Figura 2.9 Elementos de un Diagrama de Casos de Uso**

- Actor.- Persona, empresa, sistema externo a nuestra aplicación, es una entidad que tiene comportamiento. Debe residir fuera del sistema que se está modelando.
  - Primario.- Es aquel que se relaciona directamente con el caso de uso, el cual usa directamente los servicios del sistema.
  - De soporte.- Es aquel que nos provee algún tipo de servicio.
  - Externos o fuera del alcance.- Son aquellos que tienen un interés en el comportamiento del caso de uso pero no son primarios ni de soporte.
- Caso de uso.- Es una descripción hablada o escrita de los escenarios que se presentan en el sistema.
- Frontera del sistema.- Nos indica el límite que va a tener nuestro sistema
- Escenario.- Secuencia específica de acciones e interacciones entre los actores y el sistema.
- Líneas de relación.- Nos indica la relación entre un actor y un caso de uso. También se puede dar entre casos de uso.
- Curso normal de eventos.- Son aquellas acciones que se dan con mayor frecuencia.



- Curso alterno.- aquellas acciones que se dan por un error o acciones no tan frecuentes.

### 2.3.5.2 Descripción de casos de uso

- Casos de uso esenciales (abstractos).- Son descripciones las cuales van a ser independientes de la tecnología y que nos lleva al aclaración y entendimiento de las reglas del negocio. Estos casos de uso nos sirven para el análisis. Una regla de negocio es una operación principal o una política de la empresa.
- Casos de uso reales (o del sistema).- Se llaman reales porque nos van a indicar el comportamiento final que va a tener el sistema dependiente de la tecnología en cuanto a la interfaz gráfica de usuario. Se debe previamente hacer un prototipo de pantallas. Aquí es donde se da solución al problema por lo tanto esto es el diseño<sup>7</sup>

---

<sup>7</sup> Apuntes sobre “Desarrollo de Sistemas Orientado a Objetos con UML”



## CAPITULO III

### HERRAMIENTAS DE PROGRAMACIÓN Y DESARROLLO

#### 3.1 INTRODUCCIÓN

En la actualidad existen diferentes paradigmas de programación, así como herramientas que permiten el desarrollo de software siguiendo las líneas de estos establecidas por estos paradigmas. Es por tanto que se nos presenta el hecho de tomar la decisión más adecuada al tipo de aplicación que deseamos desarrollar.

Un proceso de desarrollo de programas tiene como objetivo la formalización de las actividades relacionadas con la elaboración de sistemas informáticos. La formalización de un proceso de desarrollo tiende a dotar a las empresas de un conjunto de mecanismos que, cuando se aplican sistemáticamente, permiten obtener de manera repetitiva y fiable sistemas de programas de calidad constante. Por naturaleza, la descripción del proceso es general porque no es posible definir autoritariamente un estándar único, adoptando a todas las personas, a todos los tipos de aplicaciones y a todas las culturas. Conviene mas bien hablar de un marco configurable, eventualmente refinado, de manera consensuada por la práctica y la implementación de productos ampliamente adoptados por la comunidad de usuarios.

Para realizar una adecuada decisión se debe tener claramente definido que es lo que vamos a realizar así como de los requerimientos que debe cumplir cuando se encuentre en funcionamiento, por ejemplo si será multiplataforma, manejará un gran número de datos, será una aplicación con entorno de escritorio o con entorno web (o ambas), etc.



Por otro lado debemos también considerar si contamos con una documentación amplia del software y herramientas elegidas. Es por ello que debemos conocer detalladamente las diferentes herramientas existentes y así hacer una elección adecuada para iniciar nuestro proyecto de desarrollo de determinada aplicación.

## 3.2 LENGUAJE DE PROGRAMACIÓN JAVA

### 3.2.1 Definición

Java es un lenguaje de programación moderno, orientado a objetos, tiene extensiones que permiten diseñar interfaces de usuario gráficas (GUI), tiene la sintaxis de C y C++, cuenta también con la gestión de memoria dinámica y la ejecución multihilos de Smalltalk.

Los programas desarrollados en *Java* presentan diversas ventajas frente a los desarrollados en otros lenguajes como C/C++. La ejecución de programas en *Java* tiene muchas posibilidades: ejecución como aplicación independiente (*Stand-alone Application*), ejecución como *applet*, ejecución como *servlet*, etc.

*Java* permite fácilmente el desarrollo tanto de arquitecturas cliente-servidor como de aplicaciones distribuidas, consistentes en crear aplicaciones capaces de conectarse a otros ordenadores y ejecutar tareas en varios ordenadores simultáneamente, repartiendo por lo tanto el trabajo. Aunque también otros lenguajes de programación permiten crear aplicaciones de este tipo, *Java* incorpora en su propia *API* estas funcionalidades<sup>8</sup>

### 3.2.2 Características

- **Simple.-** Java es un lenguaje simple. Para que el aprendizaje no supusiese grandes esfuerzos se diseñó una pequeña cantidad de construcciones en el lenguaje. También se intentó que fuese similar a otros lenguajes de

---

<sup>8</sup> Laboratorio de Informática II - Introducción



programación, como C o C++. Suprimieron una serie de características de C y C++ que llevaban a una mala programación y que raramente se utilizaban.

- Java no permite la sentencia goto, para lo que ofrece otras posibilidades como break etiquetadas y excepciones.
  - Java no tiene ficheros de cabecera y elimina el preprocesador.
  - Como es orientado a objetos se han eliminado las uniones y estructuras.
  - No se utilizan punteros, Java gestiona automáticamente las direcciones (&) e indirecciones (\*) de los objetos.
  - Implementa automáticamente la recolección de basura (garbage collection), por lo que no hay que preocuparse de la gestión de memoria: referencias no válidas a memoria, reserva y liberación de memoria, etc.
- **Independiente de la plataforma.-** Java no se compila a código máquina, genera "bytecodes" para la Máquina Virtual de Java (JVM). Por tanto, se podrá ejecutar en cualquier plataforma que tenga el intérprete. Actualmente, el intérprete se encuentra en las principales plataformas (Windows, Unix, Mac, Linux, etc).
  - **Portable.-** Como el bytecode generado por el compilador de Java es independiente de la arquitectura, una aplicación Java se puede ejecutar en cualquier plataforma que implemente la JVM. Esto es especialmente importante para aplicaciones distribuidas en Internet o en redes heterogéneas.
  - **Multihilo.-** Java no sólo ofrece la posibilidad de crear programas multithread que se ejecuten en distintas plataformas, además, es sencillo de utilizar. La ejecución multithread es la idea de ejecutar más de una tarea al mismo tiempo. Generalmente es la forma de reservar tiempo de un único procesador, si el SO soporta el multiprocesamiento se puede asignar un procesador a cada hilo de ejecución.

La ventaja de soportar la ejecución multithread desde el lenguaje de programación es que el programador no tiene que preocuparse de si hay varios procesos ejecutándose o sólo uno.<sup>9</sup>

### 3.2.3 JavaBeans

---

<sup>9</sup> Laboratorio de Informática II - Introducción



JavaBeans permite escribir componentes software en Java. Los componentes son unidades software reutilizables y auto-contenidas que pueden unirse visualmente en componentes compuestos, applets, aplicaciones y servlets utilizando herramientas visuales de desarrollo de aplicaciones.

Los componentes JavaBean son conocidos como Beans. Una herramienta de desarrollo que soporte JavaBeans, mantiene los Beans en una paleta o caja de herramientas. Se puede seleccionar un Bean de la paleta, arrastrarlo dentro de un formulario, modificar su apariencia y su comportamiento, definir su interacción con otros Beans, y componer un applet, una aplicación, o un nuevo Bean, junto con otros Beans. Todo esto se puede hacer sin escribir una línea de código.

La siguiente lista describe brevemente los conceptos clave de los Beans:

- Las herramientas de desarrollo descubren las características de un Bean (esto es, sus propiedades, sus métodos y sus eventos) mediante un proceso conocido como introspección. Los Beans soportan la introspección de dos formas:
  - Adhiriéndose a las convenciones específicas de nombres conocidas como patrones de nombrado, cuando se nombran las características del Bean. La clase `java.beans.Introspector` examina el Bean buscando esos patrones de diseño para descubrir las características del Bean. La clase `Introspector` se encuentra en el API core `reflection`.
  - Proporcionando explícitamente información sobre la propiedad, el método o el evento con una clase Bean Información relacionada. Esta clase implementa el interface `BeanInfo`. Una clase `BeanInfo` lista explícitamente aquellas características del Bean que están expuestas a la herramienta de desarrollo.
- Propiedades son las características de apariencia y comportamiento de un Bean que pueden ser modificadas durante el diseño. Las propiedades se exponen a las herramientas de desarrollo mediante los patrones de diseño o una clase `BeanInfo`.
- Los Beans exponen sus propiedades para poder ser personalizados durante el diseño. La personalización se soporta de dos formas: utilizando editores de propiedades, o utilizando personalizadores de Beans más sofisticados.
- Los Beans utilizan los eventos para comunicarse con otros Beans. Un Bean que quiere recibir eventos (un Bean oyente) registra su interés con un Bean



que lanza eventos (un Bean fuente). Las herramientas de desarrollo pueden examinar un Bean para determinar que eventos puede disparar (enviar) y cuales puede manejar (recibir).

- La Persistencia permite a los Beans guardar su estado, y restaurarlo posteriormente. Una vez que se han cambiado las propiedades de Bean, se puede guardar su estado y restaurar el Bean posteriormente. Los JavaBeans utilizan la Serialización de Objetos Java para soportar la persistencia.
- Los métodos de un Bean no son diferentes de los métodos Java, y pueden ser llamados desde otros Beans o desde un entorno de scripts. Por defecto, todos los métodos públicos son exportados.<sup>10</sup>

### 3.3 JMF - JAVA MEDIA FRAMEWORK

#### 3.3.1 Definición

Java Media Framework (JMF) es un API (Application Programming Interface), para la incorporación de medios basados en el tiempo (time-based medias) en aplicaciones Java. Los medios basados en el tiempo se caracterizan por cambiar su contenido con el transcurso del tiempo. Dentro de esta categoría encontramos: audio, video, secuencias MIDI y animaciones. Inicialmente, la JMF 1.0 habilitaba a los programadores para desarrollar aplicaciones que simplemente presentaban estos tipos de medios. Actualmente la JMF 2.0, extiende el área de trabajo para proveer soporte para captura y almacenaje de medios. Se puede controlar el tipo de procesamiento que es efectuado durante la reproducción mediante la utilización de plug-ins.<sup>11</sup>

Los objetivos principales del diseño de JMF 2.0 API son:

- Programación más amigable.
- Soporte para la captura de medios.

<sup>10</sup> Pequeño Tutor de Java – [www.redestb.es](http://www.redestb.es)

<sup>11</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – [www.sun.com](http://www.sun.com)



- Habilita el desarrollo de aplicaciones para flujos multimedia y conferencias en Java.
- Mantener la compatibilidad con JMF 1.0 API

### 3.3.2 Datos Basados en el Tiempo (Time-Based Media)

Toda información que tenga cambios significativos con respecto al tiempo puede ser catalogada como un medio basado en el tiempo, como lo son los clips de audio, secuencias MIDI, clips de video, etc. Estos medios pueden ser obtenidos de diversas fuentes, como archivos locales o remotos, cámaras, micrófonos y difusiones en vivo. A continuación se presenta un modelo que describe las características de estos medios y la manipulación que a estos se les debe aplicar.<sup>12</sup>

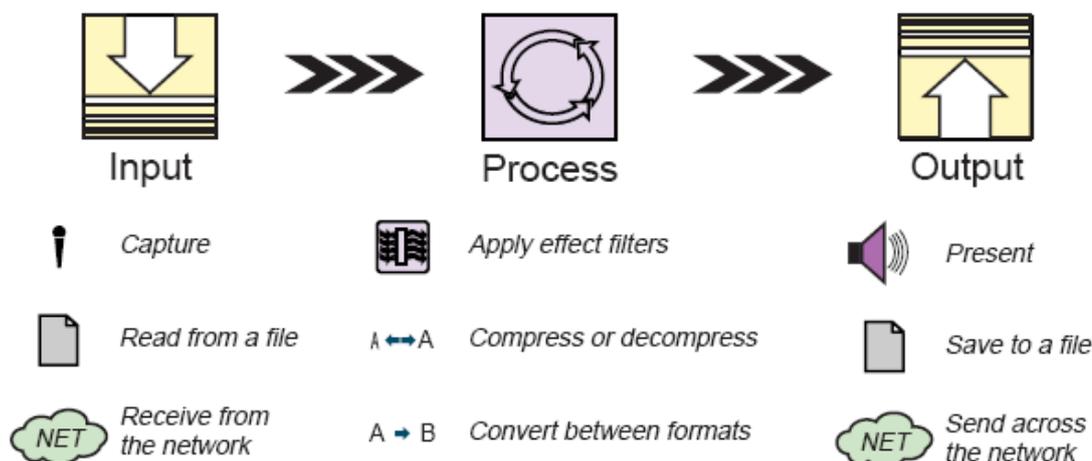


Figura 3.1 Modelo de Procesamiento de Medios

### 3.3.3 Características claves de los medios basados en el tiempo.

- **Flujo de Medios (Streaming Media)**

La característica principal de este medio es que requiere de un tiempo de entrega y de procesamiento, y por esto se debe controlar, ya que una vez iniciado el flujo de datos se deben satisfacer ciertos límites de tiempo.

<sup>12</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – www.sun.com



- **Presentación de Medios (Output)**

La mayoría de estos medios pueden ser presentados a través de dispositivos de salida tales como parlantes y monitores, u otros destinos (ej: a la red). Comúnmente a estos destinos de medios se le llaman DataSinks.

- **Procesamiento de Medios (Process)**

En muchas instancias la información contenida en un medio es manipulada antes de ser presentado al usuario, ya sea multiplexandola, filtrándola, comprimiéndola, o convirtiéndola en otro tipo de medio.

- **Captura de Medios (Input)**

Estos pueden ser capturados desde una fuente en vivo para procesarla y reproducirla o puede ser adquirida de un archivo de forma remota.

JMF proporciona herramientas para la captura, procesamiento y almacenamiento de datos multimedia.

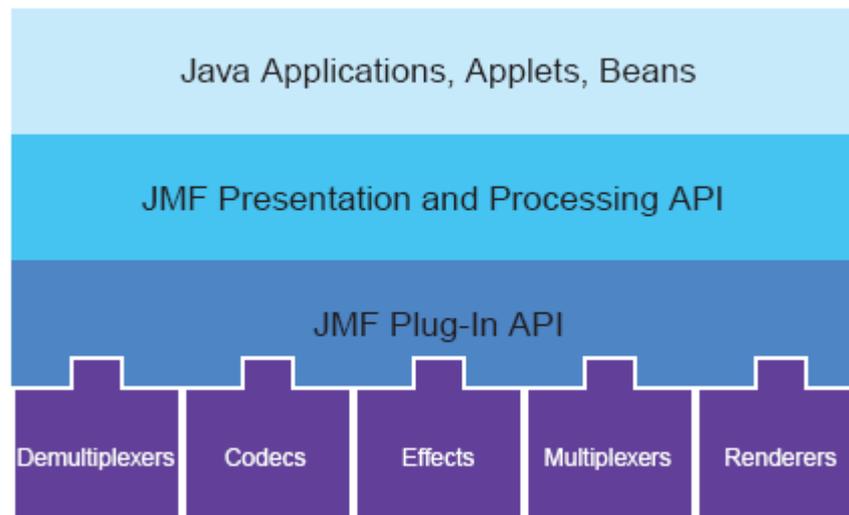
- Permite su transmisión y recepción a través de Internet.
- Más concretamente permite:
  - Reproducir ficheros multimedia en applets y aplicaciones.
  - Reproducir flujos multimedia recibidos en tiempo real a través de la red.
  - Capturar audio y vídeo de un micrófono y una cámara de vídeo.<sup>13</sup>

### 3.3.4 Arquitectura de Alto Nivel de JMF

- Una parte integral de la JMF son las fuentes de datos y los reproductores para la administración de la captura, presentación y procesamiento de los medios, presentando la estructura de la figura 3.2.

---

<sup>13</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – [www.sun.com](http://www.sun.com)



**Figura 3.2 Arquitectura de Alto Nivel de JMF**

#### **3.3.4.1 Presentación**

El proceso de presentación es modelado por la interfaz controladora, que define el estado básico y el mecanismo de control para el objeto presentado o capturado. En este caso se definen 2 tipos de controladores: reproductores y procesadores.

#### **3.3.4.2 Procesamiento**

El procesador es un reproductor que toma una fuente de datos, realiza algunos procesos definidos por el usuario, y luego entrega la información de medio procesada. Las etapas por las que el medio atraviesa son la de multiplexación, transcodificación, demultiplexación y presentación.

#### **3.3.4.3 Captura**

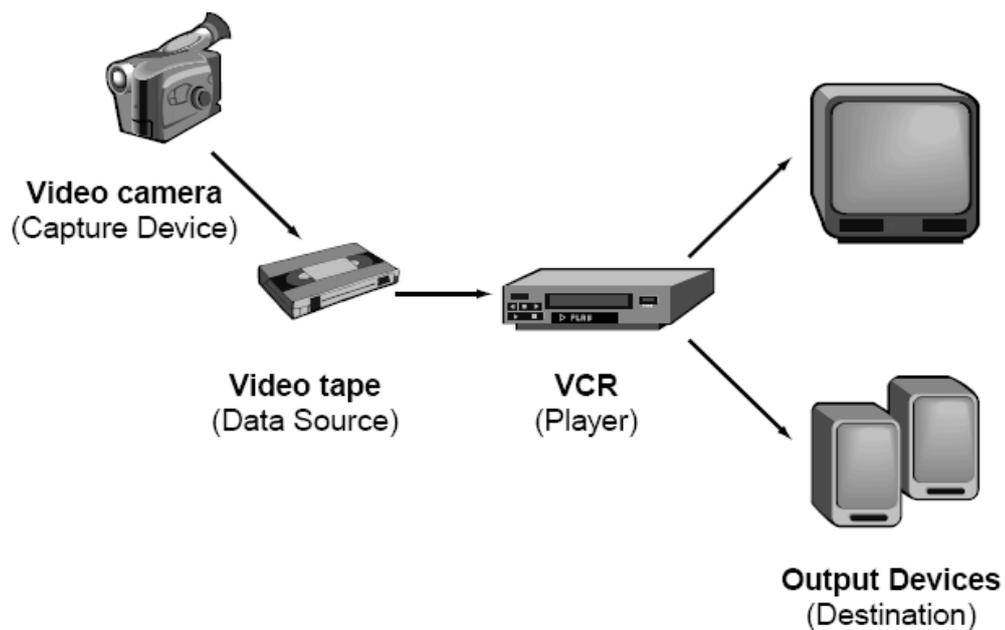
Un dispositivo de captura multimedia puede actuar como una fuente de entrega de medios basados en el tiempo. Algunos dispositivos entregan múltiples flujos de datos que pueden ser separados mediante el procesamiento.

### 3.3.4.4 Almacenamiento y transmisión de medios

Un DataSink es usado para leer medios desde una fuente y presentarlo en algún destino. Particularmente el DataSink puede escribir datos a un archivo, escribir datos a través de la red o funcionar como un difusor RTP.<sup>14</sup>

### 3.3.4.5 Clases e interfaces de JMF

Cualquier aplicación JMF puede ceñirse al siguiente esquema:



Figura

### 3.3 Captura, procesamiento y presentación de medios basados en el tiempo

#### 3.3.4.5.1 DataSource

- Clase de javax.media.protocol.
- Representa el origen de datos multimedia.
- Encapsula información sobre su ubicación, el protocolo para su transferencia al reproductor y el software para tratarlos.
- Soporta todo tipo de contenido multimedia.<sup>15</sup>

<sup>14</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – www.sun.com

<sup>15</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – www.sun.com



## Ejemplo de creación de DataSource

DataSource pertenece al paquete javax.media.protocol por lo que debemos importarlo, además del paquete javax.media que contiene clase como MediaLocator, CaptureDeviceManager y CaptureDeviceInfo que necesitaremos para crear el DataSource.

```
import javax.media.protocol;  
import javax.media.*;
```

...

...

Ahora creamos un objeto DataSource

```
DataSource audioDataSource= null;
```

Para crear un DataSource necesitamos también un objeto de la clase MediaLocator por tanto también creamos un objeto de esta clase

```
MediaLocator audioLocator;
```

...

...

A continuación obtenemos la lista de dispositivos usando la clase CaptureDeviceManager, se utiliza el método getDeviceList al cual le enviamos como parámetro el formato de audio deseado, en este caso nos devolverá una lista con todos los dispositivos que soporten el formato de audio LINEAR

```
Vector devices=  
CaptureDeviceManager.getDeviceList(AudioFormat.LINEAR);
```



```
if (devices.size() != 0) {
```

Obtenemos el primer elemento de la lista el cual contiene la información acerca de este dispositivo, lo almacenamos en un objeto `CaptureDeviceInfo`, el cual encapsula toda la información acerca de este dispositivo

```
CaptureDeviceInfo deviceInfo= devices.firstElement();
```

Ahora creamos un objeto `MediaLocator` a partir de la información del dispositivo almacenada en el objeto creado anteriormente

```
audioLocator= deviceInfo.getLocator();
```

```
}
```

```
try {
```

Finalmente creamos el `DataSource` usando el método `createDataSource` que recibe como parámetro el objeto `MediaLocator` creado

```
audioDataSource =  
    javax.media.Manager.createDataSource(audioLocator);
```

```
} catch (Exception e) {
```

```
    System.out.println(e.toString());
```

```
}
```

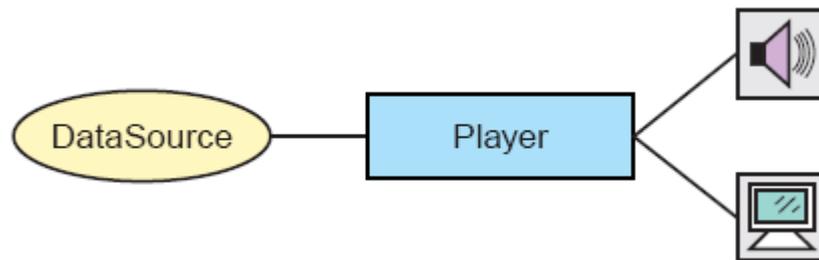
Con esto tenemos un objeto `DataSource` que encapsula información sobre el dispositivo de captura, el cual nos permite capturar datos de audio en formato linear que puede ser enviado a un player para su reproducción, o a un processor para su tratamiento. Este procedimiento se puede emplear de forma similar para capturar datos de video desde un dispositivo.

#### 3.3.4.5.2 Player

- Interface de `javax.media`.
- Actúa como gestor de datos multimedia.
- Asegura que lleguen adecuadamente al dispositivo de salida.



- No hace falta un Player específico para cada tipo de datos.



**Figura 3.4 Modelo de un Player JMF**

- Tiene asociados controles para actuar sobre los datos multimedia que está procesando.
- Antes de que el fragmento pueda reproducirse, el Player debe pasar por una serie de estados.
- Normalmente bastará con invocar al método `start()` sobre el Placer para que pase automáticamente por todos estos estados.<sup>16</sup>

### Ejemplo de creación de Player

Para usar Player debemos importar el paquete `javax.media` al cual pertenece esta clase.

```
import javax.media.*;
```

```
...
```

```
...
```

Creamos un objeto Player que lo usaremos para reproducir datos multimedia

<sup>16</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – [www.sun.com](http://www.sun.com)



```
Player player;
```

```
...
```

```
try {
```

Creamos el player a partir de un DataSource usando la clase Manager

```
player=Manager.createPlayer(audioDataSource);
```

Agregamos un oyente ControllerListener que se encargará de actualizar los datos cada vez que los reciba desde el DataSource así como detener la reproducción cuando llegue al final del stream de datos

```
player.addControllerListener(this);
```

Finalmente invocamos el método start del player para que inicie la reproducción

```
player.start();
```

```
}catch(Exception e){
```

```
    System.out.println(e.toString());
```

```
}
```

El player nos permite reproducir cualquier stream de datos multimedia que pueden provenir de un dispositivo, de un archivo, o de un processor.

#### 3.3.4.5.3 Processor

- Es un tipo de Player, subinterfaz de javax.media.Player.
- Aporta dos características suplementarias:
  - Puede volcar los datos procesados en un DataSource, en lugar de en un dispositivo de salida.
  - Puede cambiar de formato lo datos.

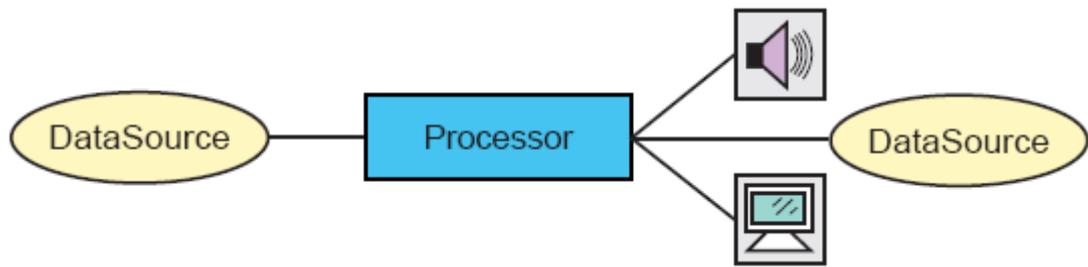


Figura 3.5 Modelo de un Processor JMF

- Un Processor tiene dos estados más que un Player.<sup>17</sup>

### Ejemplo de creación de un processor

Para usar Processor debemos importar el paquete javax.media al cual pertenece esta clase.

```
import javax.media.*;
```

```
...
```

```
...
```

Creamos un objeto Processor que lo usaremos para el tratamiento de datos multimedia

```
Processor processor= null;
```

```
...
```

```
try {
```

Creamos el processor a partir de un DataSource usando la clase Manager

```
processor=Manager.createProcessor(audioDataSource);
```

```
}catch (NoProcessorException e) {
```

```
System.out.println(e.toString());
```

```
}
```

<sup>17</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – www.sun.com



```
}catch(IOException e){  
    System.out.println(e.toString());  
}
```

En este caso haremos que el processor convierta los datos que obtiene desde el dispositivo de captura de audio en formato linear al formato RAW\_RTP para ser transmitidos por la red

Primero debemos esperar hasta que el processor se encuentre en estado configured, el método `waitForState` realiza la tarea de esperar hasta que el processor pase a este estado. El método `waitForState` recibe como parámetro el processor y el estado deseado, por tanto se puede usar para conocer cuando el processor se encuentra en diferentes estados.

```
waitForState(audioProcessor, Processor.Configured);
```

A continuación usando un objeto `ContentDescriptor` especificamos el formato al cual deseamos convertir los datos y lo fijamos al processor usando el método `setContentDescriptor`.

```
ContentDescriptor cd = new  
ContentDescriptor(ContentDescriptor.RAW_RTP);  
audioProcessor.setContentDescriptor(cd);
```

Esperamos a que el processor se encuentre en estado `Realized`

```
waitForState(audioProcessor, Processor.Realized);
```

Obtenemos el `DataSource` que este processor genera como salida con el método `getDataOutput`.

```
DataSource audioDataOutput = audioProcessor.getDataOutput();
```



El objeto DataSource que se obtiene como salida del processor es el resultado de la conversión del formato linear a raw\_rtp, que puede ser enviado por la red o usado por un player para ser reproducido; así también puede convertirse en la entrada de otro processor para recibir otro tratamiento.

#### 3.3.4.5.4 Format

- Es una clase de javax.media.
- Modela el formato de los datos multimedia.
- Tiene dos subclases:
  - ✓ AudioFormat: Encapsula información sobre el formato de datos de audio:
    - Frecuencia de muestreo.
    - Número de bits por muestra.
    - Número de canales.
  - ✓ Video Format: Encapsula información sobre el formato de datos de video.<sup>18</sup>

#### Ejemplo de uso de Format

Para usar Format debemos en primer lugar importar el paquete javax.media

```
import javax.media.*;
```

...

...

Format se puede usar para obtener una lista de dispositivos que soporten un formato en particular, así como para especificar el formato de salida de los datos de un processor

---

<sup>18</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – [www.sun.com](http://www.sun.com)



Para obtener una lista de dispositivos de audio en formato linear con una frecuencia de muestreo de 44100Hz con una muestra de 16 bits y con 2 canales se lo hace de la siguiente forma

```
AudioFormat af= new AudioFormat("linear", 44100, 16, 2);
```

Utilizamos el método `getDeviceList` de la clase `CaptureDeviceManager`, usando como parámetro el objeto de la clase `AudioFormat` creado anteriormente, con ello en el vector tendremos todos aquellos dispositivos que cumplen con el formato especificado y solo hará falta obtener uno de ellos desde el vector. El método `getDeviceList` se usa de la siguiente forma:

```
Vector deviceList= CaptureDeviceManager.getDeviceList(af)
```

#### 3.3.4.5.5 Manager

- Es una clase Final de `javax.media`.
- Contiene métodos estáticos para crear instancias de todas las interfaces y clases anteriores.<sup>19</sup>

#### Ejemplo de uso de Manager

Manager nos permite crear instancias de las clases como son: `Player`, `Processor`, etc., en los ejemplos anteriores se puede ver que se emplea para crear instancias de estas interfaces usando los métodos apropiados para cada caso.

```
player=Manager.createPlayer(audioDataSource);  
processor=Manager.createProcessor(audioDataSource);
```

### 3.4 Java Server Pages y Contenedores de Servlets /JSPs

<sup>19</sup> Sun Microsystems, Inc. – Java Media Framework Api Guide – [www.sun.com](http://www.sun.com)

### 3.4.1 Definición

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

Para empezar, los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propia hebra, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo.

Los JSPs son en realidad servlets: un JSP se compila a un programa en Java la primera vez que se invoca, y del programa en Java se crea una clase que se empieza a ejecutar en el servidor como un servlet. La principal diferencia entre los servlets y los JSPs es el enfoque de la programación: un JSP es una página Web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.

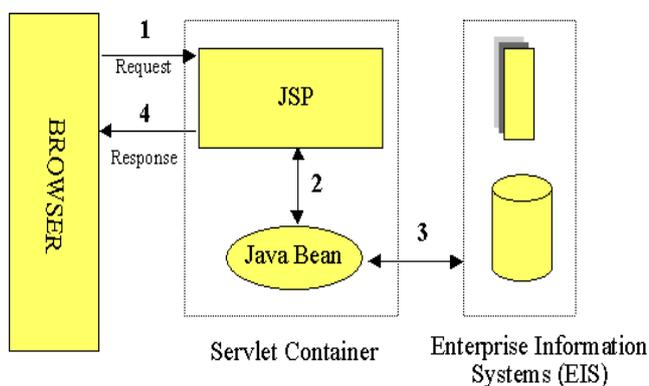


Figura 3.6 Proceso de petición-respuesta de una página JSP



Ambos necesitan un programa que los contenga, y sea el que envíe efectivamente páginas web al servidor, y reciba las peticiones, las distribuya entre los servlets, y lleve a cabo todas las tareas de gestión propias de un servidor web. Mientras que servidores como el Apache están especialmente pensados para páginas web estáticas CGI, y programas ejecutados por el servidor, tales como el PHP, hay otros servidores específicos para servlets y JSPs llamados contenedores de servlets (servlet containers) o servlet engines.

JavaServer Pages (JSP) combinan HTML con fragmentos de Java para producir páginas web dinámicas. Cada página es automáticamente compilada a servlet por el motor de JSP, en primer lugar es recogida y a continuación ejecutada.

JSP tiene gran variedad de formas para comunicarse con las clases de Java, servlets, applets y el servidor web: por esto se puede aplicar una funcionalidad a nuestra web a base de componentes.

Una página JSP es archivo de texto simple que consiste en contenido HTML o XML con elementos JSP. Cuando un cliente pide una página JSP del sitio web y no se ha ejecutado antes, la página es inicialmente pasada al motor de JSP, el cual compila la página convirtiéndola en Servlet, la ejecuta y devuelve el contenido de los resultados al cliente.

### **3.5 FRAMEWORK JSF – JAVA SERVER FACES**

#### **3.5.1 Definición**

Java Server Faces es el estándar presentado por Sun para la capa de presentación Web. Forma parte de la especificación J2EE 5 que deberán cumplir todos los servidores de aplicaciones y se erige como una evolución natural de los frameworks actuales hacia un sistema de componentes.



Es un estándar sencillo que aporta los componentes básicos de las páginas web además de permitir crear componentes más complejos (menús, pestañas, árboles, etc). Ya hay disponibles diferentes implementaciones de la especificación, tanto comerciales como de código abierto, así como librerías de componentes adicionales que amplían la funcionalidad de esos componentes iniciales.

Uno de los patrones más conocidos en el desarrollo web es el patrón MVC (Modelo Vista Controlador). Este patrón nos permite/obliga a separar la lógica de control (sabe que cosas hay que hacer pero no como), la lógica de negocio (sabe como se hacen las cosas) y la lógica de presentación (sabe como interactuar con el usuario).

Utilizando este tipo de patrones conseguimos: más calidad, mejor mantenibilidad, perder el miedo al folio en blanco (tengo un patrón a seguir al empezar un proyecto), pero una de las cosas más importantes es la normalización y estandarización del desarrollo de Software.

JSF (Java Server Faces) es un framework de desarrollo basado en el patrón MVC (Modelo Vista Controlador).

### **3.5.2 Principales Componentes**

- API e implementación de referencia para representar los componentes de interfaz de usuario y manejar su estado, manejo de eventos, validación del lado del servidor y conversión de datos; también permite definir las reglas de navegación entre páginas.
- Librerías de etiquetas Java Server Faces, personalizadas para dibujar componentes de interfaz dentro de una página JSP.

### **3.5.3 Aplicación Java Server Faces**

Es una aplicación como cualquier otra, que se ejecuta dentro de un contenedor Servlet java y típicamente contiene:

- JavaBeans, para contener datos y funcionalidades específicas de la aplicación.
- Oyentes de eventos
- Páginas (JSP)
- Beans, clases de utilidad del lado del servidor.



- Librerías de etiquetas personalizadas para dibujar componentes UI en una página.
- Librerías personalizadas para representar manejadores de eventos, validadores y otras acciones.
- Validadores, manejadores de eventos y manejadores de navegación.

### Ejemplo de uso de JSF

A continuación se presenta un ejemplo de uso de JSF en el desarrollo de aplicaciones web

Para el desarrollo de las aplicaciones por lo general se sigue los pasos a continuación:

- Desarrollar los objetos del modelo, los que contendrán los datos.
- Añadir las declaraciones del bean controlado al fichero de configuración de la aplicación.
- Crear las páginas utilizando las etiquetas de componentes UI y las etiquetas "core".
- Definir la navegación entre las páginas.

### Desarrollar los objetos del modelo

Desarrollar los objetos del modelo es responsabilidad del desarrollador de aplicaciones, el autor de las páginas y el desarrollador de aplicaciones podrían necesitar trabajar juntos para asegurarse que las etiquetas de componentes se refieren a las propiedades del objeto apropiado, que las propiedades del objeto son de los tipos apropiados y para tener cuidado de otros detalles.<sup>20</sup>

A continuación tenemos la clase `UsuarioBean.java` que contiene los datos introducidos en los campos de texto de la página `formulario_usuario.jsp`

```
public class UsuarioBean {
    private String nombre;
    private String apellido;
    private char sexo;
    private String email;
    private String institucion;
    private String ciudad;
    private String pais;
```

<sup>20</sup> Sun Microsystems, Inc. – Introducción a la Tecnología JavaServer Faces – [www.sun.com](http://www.sun.com)



```
private String nickname;
private String password;

/**
 * Getter for property nombre.
 * @return Value of property nombre.
 */
public String getNombre() {
    return this.nombre;
}

/**
 * Setter for property nombre.
 * @param nombre New value of property nombre.
 */
public void setNombre(String nombre) {
    this.nombre = nombre;
}

/**
 * Getter for property apellido.
 * @return Value of property apellido.
 */
public String getApellido() {
    return this.apellido;
}

/**
 * Setter for property apellido.
 * @param apellido New value of property apellido.
 */
public void setApellido(String apellido) {
    this.apellido = apellido;
}

/**
 * Getter for property sexo.
 * @return Value of property sexo.
 */
public char getSexo() {
    return this.sexo;
}

/**
 * Setter for property sexo.
 * @param sexo New value of property sexo.
 */
public void setSexo(char sexo) {
    this.sexo = sexo;
}

/**
 * Getter for property email.
 * @return Value of property email.
 */
```



```
public String getEmail() {
    return this.email;
}

/**
 * Setter for property email.
 * @param email New value of property email.
 */
public void setEmail(String email) {
    this.email = email;
}

/**
 * Getter for property institucion.
 * @return Value of property institucion.
 */
public String getInstitucion() {
    return this.institucion;
}

/**
 * Setter for property institucion.
 * @param institucion New value of property institucion.
 */
public void setInstitucion(String institucion) {
    this.institucion = institucion;
}

/**
 * Getter for property ciudad.
 * @return Value of property ciudad.
 */
public String getCiudad() {
    return this.ciudad;
}

/**
 * Setter for property ciudad.
 * @param ciudad New value of property ciudad.
 */
public void setCiudad(String ciudad) {
    this.ciudad = ciudad;
}

/**
 * Getter for property pais.
 * @return Value of property pais.
 */
public String getPais() {
    return this.pais;
}

/**
 * Setter for property pais.
 * @param pais New value of property pais.
 */
```



```
    */
    public void setPais(String pais) {
        this.pais = pais;
    }

    /**
     * Getter for property nickname.
     * @return Value of property nickname.
     */
    public String getNickname() {
        return this.nickname;
    }

    /**
     * Setter for property nickname.
     * @param nickname New value of property nickname.
     */
    public void setNickname(String nickname) {
        this.nickname = nickname;
    }

    /**
     * Getter for property password.
     * @return Value of property password.
     */
    public String getPassword() {
        return this.password;
    }

    /**
     * Setter for property password.
     * @param password New value of property password.
     */
    public void setPassword(String password) {
        this.password = password;
    }
}
```

Como podemos ver, este bean es como cualquier otro componente JavaBeans. Tiene un método set o accesor y un campo privado o propiedad. Esto significa que podemos concebir referenciar beans que ya hayamos escrito desde nuestras páginas JavaServer Faces.

Dependiendo del tipo de componente que referencia una propiedad del objeto del modelo, esta propiedad puede ser de cualquiera de los tipos básicos primitivos y los tipos referencia. Esto incluye cualquiera de los tipos numéricos, String, int, double, y float. La tecnología JavaServer Faces convertirá automáticamente el dato al tipo especificado por la propiedad del objeto del modelo.



También podemos aplicar una conversión a un componente para convertir los valores de los componentes a un tipo que no esté soportado por el componente.<sup>21</sup>

### Añadir las declaraciones del bean controlado

Después de desarrollar los beans utilizados en la aplicación, necesitamos añadir declaraciones para ellos en el fichero de configuración de la aplicación. Cualquier miembro del equipo de desarrollo puede realizar la tarea de añadir las declaraciones al fichero de configuración de la aplicación.<sup>22</sup>

Las declaraciones se realizan en el archivo `faces-config.xml` del directorio `WEB-INF` de nuestra aplicación. Aquí tenemos la declaración de bean controlado para `UsuarioBean.java`

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE faces-config PUBLIC "-//Sun Microsystems, Inc.//DTD
JavaServer Faces Config 1.1//EN" "http://java.sun.com/dtd/web-
facesconfig_1_1.dtd">
<!-- ===== FULL CONFIGURATION FILE
===== -->
<faces-config xmlns="http://java.sun.com/JSF/Configuration">
  <!--BEANS MANEJADOS-->
  <managed-bean>
    <managed-bean-name>userBean</managed-bean-name>
    <managed-bean-class>
      edu.videoconferencia.negocio.bean.UsuarioBean
    </managed-bean-class>
    <managed-bean-scope>session</managed-bean-scope>
  </managed-bean>
</faces-config>
```

La implementación de JavaServer Faces procesa este fichero en el momento de inicio de la aplicación e inicializa el `userBean` y lo almacena en el ámbito de sesión, entonces el bean estará disponible para todas las páginas de la aplicación.

### Crear las páginas

La creación de las páginas es responsabilidad del autor de páginas. Esta tarea implica distribuir los componentes UI en las páginas, mapear los componentes a los datos de

<sup>21</sup> Sun Microsystems, Inc. – Introducción a la Tecnología JavaServer Faces – [www.sun.com](http://www.sun.com)

<sup>22</sup> Sun Microsystems, Inc. – Introducción a la Tecnología JavaServer Faces – [www.sun.com](http://www.sun.com)



los objetos del modelo, y añadir otras etiquetas importantes (como etiquetas del validador) a las etiquetas de los componentes.<sup>23</sup>

Aquí tenemos la página `formulario_usuario.jsp`

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
    charset=UTF-8">
  <f:loadBundle
    basename="edu.videoconferencia.vista.bundle.messages"
    var="msgs"/>
  <title>Formulario Usuario</title>
  <link
    rel="stylesheet"
    href="resources/style.css" title="default"
    type="text/css"
  >
</head>
<body>
  <f:view>
  <% include file="header.jsp" %>
  <div id="container">
  <% include file="left.jsp" %>
  <div id="right">
  <% include file="date.jsp" %>
  <h3>
    <h:outputLabel
      value="#{msgs.formularioUsuarioPageTitle}"/>
  </h3>
  <h:messages/>
  <h:form>
  <table>
  <tr>
  <td>
    <h:outputText value="#{msgs.adminPrompt}"/>
  </td>
  <td>
    <h:selectBooleanCheckbox value="#{userBean.admin}"/>
  </td>
  </tr>
  <tr>
  <td>
```

<sup>23</sup> Sun Microsystems, Inc. – Introducción a la Tecnología JavaServer Faces – [www.sun.com](http://www.sun.com)



```
<h:outputText value="#{msgs.namePrompt}"/>
</td>
<td>
    <h:inputText                                id="nombre"
        value="#{userBean.nombre}"              required="true"
        styleClass="fields"/>
    <h:message for="nombre" errorClass="mensajeError"/>
</td>
</tr>
<tr>
<td>
    <h:outputText value="#{msgs.lastNamePrompt}"/>
</td>
<td>
    <h:inputText                                id="apellido"
        value="#{userBean.apellido}"            required="true"
        styleClass="fields"/>
    <h:message                                for="apellido"
        errorClass="mensajeError"/>
</td>
</tr>
<tr>
<td>
    <h:outputText value="#{msgs.sexPrompt}"/>
</td>
<td>
    <h:selectOneRadio value="#{userBean.sexo}"
        layout="lineDirection"
        styleClass="radio_button">
        <f:selectItems
            value="#{userBean.sexoItems}"/>
    </h:selectOneRadio>
</td>
</tr>
<tr>
<td>
    <h:outputText value="#{msgs.emailPrompt}"/>
</td>
<td>
    <h:inputText id="email"
        value="#{userBean.email}" required="true"
        styleClass="fields"/>
    <h:message for="email"
        errorClass="mensajeError"/>
</td>
</tr>
<tr>
<td>
    <h:outputText
        value="#{msgs.institutionPrompt}"/>
</td>
<td>
    <h:inputText id="institucion"
        value="#{userBean.institucion}" required="true"
        styleClass="fields"/>
</td>
</tr>
```



```
        <h:message for="institucion"
            errorClass="mensajeError"/>
    </td>
</tr>
<tr>
    <td><h:outputText value="#{msgs.cityPrompt}"/></td>
    <td>
        <h:inputText id="ciudad"
            value="#{userBean.ciudad}" required="true"
            styleClass="fields"/>
        <h:message for="ciudad"
            errorClass="mensajeError"/>
    </td>
</tr>
<tr>
    <td>
        <h:outputText value="#{msgs.countryPrompt}"/>
    </td>
    <td>
        <h:inputText id="pais" value="#{userBean.pais}"
            required="true" styleClass="fields"/>
        <h:message for="pais"
            errorClass="mensajeError"/>
    </td>
</tr>
</table>
<table>
    <tr>
    <td>
        <h:outputText value="#{msgs.nicknamePrompt}"/>
    </td>
    <td>
        <h:inputText id="nickname"
            value="#{userBean.nickname}" required="true"
            styleClass="fields"/>
        <h:message for="nickname"
            errorClass="mensajeError"/>
    </td>
</tr>
<tr>
    <td>
        <h:outputText value="#{msgs.passwordPrompt}"/>
    </td>
    <td>
        <h:inputSecret id="password"
            value="#{userBean.password}" required="true"
            styleClass="fields">
            <f:validateLength maximum="20"
                minimum="4"/>
        </h:inputSecret>
        <h:message for="password"
            errorClass="mensajeError"/>
    </td>
</tr>
<tr>
```



```
<td>
  <h:outputText
    value="#{msgs.passwordConfirmPrompt}"/>
</td>
<td>
  <h:inputSecret id="passwConfirm"
    value="#{userBean.passwordConfirmation}"
    styleClass="fields">
  </h:inputSecret>
  <h:message for="passwConfirm"
    errorClass="mensajeError"/>
</td>
</tr>
</table>
<table>
<tr>
<td>
  <h:commandButton value="#{msgs.button1Prompt}"
    styleClass="buttons"
    action="#{userBean.saveAction}"/>
</td>
<td>
  <h:commandButton immediate="true"
    value="#{msgs.button2Prompt}"
    styleClass="buttons"
    action="#{userBean.cancelAction}"/>
</td>
</tr>
</table>
</h:form>
</div>
</div>
<%@ include file="footer.jsp" %>
</f:view>
</body>
</html>
```

## Definir la navegación por las páginas

Otra posibilidad que tiene el desarrollador de la aplicación es definir la navegación de páginas por la aplicación, como qué página va después de que el usuario pulse un botón para enviar un formulario.

El desarrollador de la aplicación define la navegación por la aplicación mediante el fichero de configuración `faces-config.xml` del directorio `WEB-INF` de la aplicación., el mismo fichero en el que se declararon los beans manejados. Aquí están las reglas de navegación definidas para el ejemplo anterior.



```
<!--Caso de Uso Administrar Usuario-->
<navigation-rule>
  <from-view-id>/cuenta_administrador.jsp</from-view-id>
  <navigation-case>
    <from-outcome>buscarUsuario</from-outcome>
    <to-view-id>/buscar_usuario.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>/buscar_usuario.jsp</from-view-id>
  <navigation-case>
    <from-outcome>new</from-outcome>
    <to-view-id>/formulario_usuario.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>edit</from-outcome>
    <to-view-id>/editar_usuario.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>/formulario_usuario.jsp</from-view-id>
  <navigation-case>
    <from-outcome>success</from-outcome>
    <to-view-id>/buscar_usuario.jsp</to-view-id>
    <redirect/>
  </navigation-case>
</navigation-rule>
<navigation-rule>
  <from-view-id>/editar_usuario.jsp</from-view-id>
  <navigation-case>
    <from-outcome>success</from-outcome>
    <to-view-id>/buscar_usuario.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

Cada regla de navegación define cómo ir de una página (especificada en el elemento `from-view-id`) a otras páginas de la aplicación. El elemento `navigation-rule` puede contener cualquier número de elemento `navigation-case`, cada uno de los cuales define la página que se abrirá luego (definida por `to-view-id`) basándose en una salida lógica (definida mediante `from-outcome`).

La salida se puede definir mediante el atributo `action` del componente `UICommand` que envía el formulario, como en el ejemplo:

```
<h:commandButton
  value="{msgs.button1Prompt}"
  styleClass="buttons"
  action="{userBean.saveAction}"
/>
```



La salida también puede venir del valor de retorno del método invoke de un objeto Action. Este método realiza algún procesamiento para determinar la salida. Un ejemplo es que el método invoke puede chequear si la password que el usuario ha introducido en la página corresponde con la del fichero. Si es así, el método invoke podría devolver "success"; si no es así, podría devolver "failure". Un salida de "failure" podría resultar en la recarga de la página de login. Una salida de "success" podría resultar por ejemplo en que se mostrara una página para búsqueda de usuarios o de administración de usuarios.<sup>24</sup>

### 3.6 HIBERNATE

#### 3.6.1 Definición

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones. Hibernate es un generador de sentencias sql; permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada podremos generar bases de datos en cualquiera de los entornos soportados: Oracle, DB2, MySQL, etc. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

Esta primera aproximación puede ser útil para proyectos o arquitecturas sin casi clases de negocio, ya que el mantenimiento del código esta altamente ligado a los cambios en el modelo de datos relacional de la base de datos, un mínimo cambio implica la revisión de casi todo el código así como su compilación y nueva instalación en el cliente.

Aunque no podemos desechar su utilidad. El acceso a través de SQL directas puede ser utilizado de manera puntual para realizar operaciones a través del lenguaje SQL lo cual seria mucho mas efectivo que la carga de gran cantidad de objetos en memoria.

---

<sup>24</sup> Sun Microsystems, Inc. – Introducción a la Tecnología JavaServer Faces – [www.sun.com](http://www.sun.com)



Si bien un buen motor de persistencia debería implementar mecanismos para ejecutar estas operaciones masivas sin necesidad de acceder a este nivel.

Una aproximación mas avanzada seria la creación de unas clases de acceso a datos (DAO Data Acces Object). De esta manera nuestra capa de negocio interactuaría con la capa DAO y esta sería la encargada de realizar las operaciones sobre la base de datos.

Hibernate es un mapeador objeto-relacional de código abierto más maduro completo. Se utiliza muy ampliamente y se desarrollada activamente. Hibernate también soporta una de las mayores comunidades de Open Source que se puede encontrar.

### 3.6.2 Características

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos usados hoy en día para organizar y manipular datos: El usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todas las bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (*Hibernate Query Language*), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "*criteria*").



Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma.

### Ejemplo de uso de Hibernate

Para usar hibernate en un proyecto de software que implique acceso a una base de datos, en primer lugar se debe realizar un archivo de configuración donde indicaremos la base de datos a usar, el conector que usaremos para comunicarnos con la base de datos, la dirección del servidor, el nombre de la base de datos, además del nombre y ruta del archivo de mapeo para cada una de las clases que deseamos persistir en nuestra base de datos, a continuación se presenta el archivo de configuración empleado para la conexión a la base de datos husky\_db utilizada por la aplicación web Husky, este archivo de configuración debe ser ubicado en la carpeta raíz de nuestro proyecto de software.

```
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-
configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <!-- Database connection settings -->
    <property name="connection.driver_class">
      org.gjt.mm.mysql.Driver
    </property>
    <property name="connection.url">
      jdbc:mysql://localhost/husky_db
    </property>
    <property name="connection.username">root</property>
    <property name="connection.password">adminadmin</property>

    <!-- JDBC connection pool (use the built-in) -->
    <property name="connection.pool_size">1</property>
    <!-- SQL dialect -->
    <property name="dialect">
      org.hibernate.dialect.MySQLDialect
    </property>

    <!-- Enable Hibernate's automatic session context management -->
    <property name="current_session_context_class">
      Thread
    </property>
    <!-- Disable the second-level cache -->
    <property name="cache.provider_class">
      org.hibernate.cache.HashtableCacheProvider
    </property>
```



```
<!-- Echo all executed SQL to stdout -->
<property name="show_sql">true</property>
<!-- Drop and re-create the database schema on startup -->
<property name="hbm2ddl.auto">update</property>
<mapping resource="edu/videoconferencia/negocio/integracion/
  persona.hbm.xml"/>
<mapping resource="edu/videoconferencia/negocio/integracion/
  direccion.hbm.xml"/>
<mapping resource="edu/videoconferencia/negocio/integracion/
  noticia.hbm.xml"/>
<mapping resource="edu/videoconferencia/negocio/integracion/
  foro.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

Lo siguiente que necesitamos es el archivo de mapeo de cada una de las clases que serán almacenadas en la base de datos, a continuación tenemos el archivo de mapeo de la clase persona que tiene una subclase llamada usuario.

```
<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-
    3.0.dtd">

<hibernate-mapping>

    <class name="edu.videoconferencia.negocio.dominio.Persona"
    table="persona_table">

        <id name="personaId" column="persona_id" type="long">
            <generator class="increment"/>
        </id>

        <property name="nombre" column="nombre"/>
        <property name="apellido" column="apellido"/>
        <property name="sexo" column="sexo"/>
        <property name="email" column="email"/>
```



```
<joined-subclass
name="edu.videoconferencia.negocio.dominio.Usuario"
table="usuario_table">

    <key column="persona_id"/>

    <property name="nickname" column="nick"/>

    <property name="password" column="pass"/>

    <property name="admin" column="admin"/>

</joined-subclass>

</class>

</hibernate-mapping>
```

Una vez que contamos con el archivo de configuración y el archivo de mapeo para nuestras clases creamos una instancia de la clase `SessionFactory` que nos devuelve un objeto con el cual podemos realizar las operaciones de acceso a nuestra base de datos, a continuación el código de la clase `HibernateInit` donde podemos ver como crear un `SessionFactory`.

```
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import org.hibernate.HibernateException;

/**
 * La clase HibernateInit provee la configuración
 de Hibernate y
 * los métodos necesarios para crear una conexión a la base de
 datos de la
 * aplicación Husky Videoconferency.
 *
 * @author Willan Mejía
 * @author Victor Gordillo
```



```
* @version 1.0, 19-07-2007
*/

public abstract class HibernateInit {

    /* El SessionFactory para la comunicación a la base de datos */
    private static SessionFactory sessionFactory;

    /**
     * Devuelve un objeto <code>SessionFactory</code> para
     * la comunicación a la base de datos usando Hibernate.
     *
     * @return Un objeto <code>SessionFactory</code> para
     * la comunicación a la base de datos usando Hibernate.
     */

    public static SessionFactory iniciar() {
        try {
            sessionFactory= new
                Configuration().configure().buildSessionFactory();
        }catch(HibernateException ex) {
            ex.printStackTrace();
        }
        return sessionFactory;
    }
}
```

Finalmente desde la clase UsuarioDao usamos el objeto que nos devuelve la clase anterior para trabajar con nuestra base de datos.



```
import edu.videoconferencia.negocio.dominio.Usuario;
import edu.videoconferencia.negocio.dominio.Direccion;
//
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.SessionFactory;
import org.hibernate.Query;
import org.hibernate.HibernateException;
import org.hibernate.criterion.Restrictions;
//
import java.util.List;

/**
 * La clase <code>UsuarioDao</code> provee los métodos necesarios para
 * interactuar con los datos de los usuarios y la base de datos de la
 * aplicación Husky Videoconferency.
 *
 * <p>
 * Esta clase implementa la interfaz <code>HuskyDao</code>
 *
 * @author Willan Mejía
 * @author Victor Gordillo
 * @version 1.0, 06-09-2007
 */

public class UsuarioDao implements HuskyDao {

    /**
     * Este método almacena los datos del objeto enviado
```



```
* como parámetro.
*
* @param obj El <code>Object</code> a ser almacenado.
* @param sessionFactory Un objeto <code>SessionFactory</code>
* para la conexión a la base de datos.
*/

public void store(Object obj, SessionFactory sessionFactory) {
    try {
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        try {
            Usuario usuario= (Usuario) obj;
            session.save(usuario);
        }
        finally {
            tx.commit();
            session.close();
        }
    }
    catch (HibernateException e) {
        e.printStackTrace();
    }
}

/**
 * Este método actualiza los datos del objeto enviado como
 * parámetro.
 *
 * @param obj El <code>Object</code> a ser actualizado.
```



```
* @param sessionFactory Un objeto <code>SessionFactory</code>
* para la conexión a la base de datos.
*/

public void update(Object obj, SessionFactory sessionFactory) {
    try {
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        try {
            Usuario usuario= (Usuario) obj;
            session.update(usuario);
        }
        finally {
            tx.commit();
            session.close();
        }
    }
    catch (HibernateException e) {
        e.printStackTrace();
    }
}

/**
 * Este método borra los datos del objeto enviado como parámetro.
 *
 * @param obj El <code>Object</code> a ser borrado.
 * @param sessionFactory Un objeto <code>SessionFactory</code>
 * para la conexión a la base de datos.
 */
```



```
public void delete(Object obj, SessionFactory sessionFactory) {  
    try {  
        Session session = sessionFactory.openSession();  
        Transaction tx = session.beginTransaction();  
        try {  
            Usuario usuario= (Usuario) obj;  
            session.delete(usuario);  
        }  
        finally {  
            tx.commit();  
            session.close();  
        }  
    }  
    catch (HibernateException e) {  
        e.printStackTrace();  
    }  
}
```

```
/**  
 * Devuelve una lista con todos los usuarios existentes.  
 *  
 * @param sessionFactory Un objeto <code>SessionFactory</code>  
 * para la conexión a la base de datos.  
 *  
 * @return Un <code>Object</code> con la lista de usuarios  
 * existentes.  
 */
```

```
public Object list(SessionFactory sessionFactory) {  
    List listaUsuarios= null;
```



```
try {
    Session session = sessionFactory.openSession();
    Transaction tx = session.beginTransaction();
    try {
        listaUsuarios= session.createQuery("from
            Usuario").list();
    }
    finally {
        tx.commit();
        session.close();
    }
}
catch (HibernateException e) {
    e.printStackTrace();
}
return listaUsuarios;
}

/**
 * Devuelve una lista con todos los usuarios cuyo nombre
 * corresponda con el
 * valor especificado como parámetro.
 *
 * @param value El <code>String</code> que será buscado.
 * @param sessionFactory Un objeto <code>SessionFactory</code>
 * para la conexión a la base de datos.
 *
 * @return Un <code>Object</code> con la lista de usuarios que
 * coinciden con el valor especificado.
 */
```



```
public Object searchForName(String value, SessionFactory
    sessionFactory) {
    List usuarios= null;
    try {
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        try {
            usuarios= session.createCriteria(Usuario.class)
                .add(Restrictions.like("nombre", "%" +
                    value + "%")).list();
        }
        finally {
            tx.commit();
            session.close();
        }
    }
    catch (HibernateException e) {
        e.printStackTrace();
    }
    return usuarios;
}

/**
 * Devuelve una lista con todos los usuarios cuyo apellido
 * corresponda con el valor especificado como parámetro.
 *
 * @param value El <code>String</code> que será buscado.
 * @param sessionFactory Un objeto <code>SessionFactory</code>
 * para la conexión a la base de datos.
```



```
*
* @return Un <code>Object</code> con la lista de usuarios que
* coinciden con el valor especificado.
*/

public Object searchForLastname(String value, SessionFactory
    sessionFactory) {
    List usuarios= null;
    try {
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        try {
            usuarios= session.createCriteria(Usuario.class)
                .add(Restrictions.like("apellido", "%" +
                    value + "%")).list();
        }
        finally {
            tx.commit();
            session.close();
        }
    }
    catch (HibernateException e) {
        e.printStackTrace();
    }
    return usuarios;
}

/**
* Devuelve una lista con todos los usuarios cuyo email
* corresponda con el valor especificado como parámetro.
```



```
*  
* @param value El <code>String</code> que será buscado.  
* @param sessionFactory Un objeto <code>SessionFactory</code>  
* para la conexión a la base de datos.  
*  
* @return Un <code>Object</code> con la lista de usuarios que  
* coinciden con el valor especificado.  
*/
```

```
public Object searchByEmail(String value, SessionFactory  
    sessionFactory) {  
    List usuarios= null;  
    try {  
        Session session = sessionFactory.openSession();  
        Transaction tx = session.beginTransaction();  
        try {  
            usuarios= session.createCriteria(Usuario.class)  
                .add(Restrictions.like("email", "%" +  
                    value + "%")).list();  
        }  
        finally {  
            tx.commit();  
            session.close();  
        }  
    }  
    catch (HibernateException e) {  
        e.printStackTrace();  
    }  
    return usuarios;  
}
```



```
/**
 * Devuelve una lista con todos los usuarios cuyo nickname
 * corresponda con el valor especificado como parámetro.
 *
 * @param value El <code>String</code> que será buscado.
 * @param sessionFactory Un objeto <code>SessionFactory</code>
 * para la conexión a la base de datos.
 *
 * @return Un <code>Object</code> con la lista de usuarios que
 * coinciden con el valor especificado.
 */

public Object searchForNickname(String value, SessionFactory
    sessionFactory) {
    List usuarios= null;
    try {
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        try {
            usuarios= session.createCriteria(Usuario.class)
                .add(Restrictions.like("nickname", "%" +
                    value + "%")).list();
        }
        finally {
            tx.commit();
            session.close();
        }
    }
    catch (HibernateException e) {
```



```
        e.printStackTrace();
    }
    return usuarios;
}

/**
 * Devuelve un objeto correspondiente al valor especificado como
 * parámetro.
 *
 * @param value El nickname del usuario que se desea obtener.
 * @param sessionFactory Un objeto SessionFactory
 * para la conexión a la base de datos.
 *
 * @return Un Object correspondiente al valor
 * especificado.
 */

public Object readUser(String value, SessionFactory
    sessionFactory) {
    Usuario usuario= null;
    try {
        Session session = sessionFactory.openSession();
        Transaction tx = session.beginTransaction();
        try {
            List lst= session.createCriteria(Usuario.class)
                .add(Restrictions.like("nickname", value)).list();
            if(lst.size() > 0) {
                usuario= (Usuario) lst.get(0);
            }
        }
    }
}
```



```
        finally {  
            tx.commit();  
            session.close();  
        }  
    }  
    catch (HibernateException e) {  
        e.printStackTrace();  
    }  
    return usuario;  
}  
}
```



## 3.7 APACHE TOMCAT

### 3.7.1 Definición

Tomcat (también llamado Jakarta Tomcat o Apache Tomcat) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems.

Tomcat es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software Licence*. Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 6.x, que implementan las especificaciones de Servlet 2.4 y de JSP 2.0. A partir de la versión 4.0, Jakarta Tomcat utiliza el contenedor de servlets Catalina.

### 3.7.2 Entorno

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

### 3.7.3 Estructura de directorios



La jerarquía de directorios de instalación de Tomcat incluye:

- bin - arranque, cierre, y otros scripts y ejecutables
- common - clases comunes que pueden utilizar Catalina y las aplicaciones web
- conf - ficheros XML y los correspondientes DTD para la configuración de Tomcat
- logs - logs de Catalina y de las aplicaciones
- server - clases utilizadas solamente por Catalina
- shared - clases compartidas por todas las aplicaciones web
- webapps - directorio que contiene las aplicaciones web
- work - almacenamiento temporal de ficheros y directorios

### **Características principales**

#### **Tomcat 3.x** (distribución inicial)

- Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
- Recarga de servlets
- Funciones básicas HTTP

#### **Tomcat 4.x**

- Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
- Contenedor de servlets rediseñado como Catalina
- Motor JSP rediseñado con Jasper
- Conector Coyote
- Java Management Extensions (JMX), JSP Y administración basada en Struts

#### **Tomcat 5.x**

- Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
- Recolección de basura reducida
- Capa envolvente nativa para Windows y Unix para la integración de las plataformas
- Análisis rápido JSP

#### **Tomcat 6.x**



- Implementado de Servlet 2.5 y JSP 2.1
- Soporte para *Unified Expression Language 2.1*
- Diseñado para funcionar en Java SE 5.0 y posteriores
- Soporte para Comet a través de la interfaz CometProcessor

### 3.8 Servidor de Base de Datos MySQL

#### 3.8.1 Definición

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero las empresas que quieran incorporarlo en productos privativos pueden comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario que proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson, y Michael Widenius.

#### 3.8.2 Lenguajes de programación

Existen varias APIs que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del



driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP.

### 3.8.3 Características distintivas

Las siguientes características son implementadas únicamente por MySQL:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo.



## 8. DESARROLLO DE LA PROPUESTA ALTERNATIVA

### 4.5 INTRODUCCIÓN

Con la finalidad de dar cumplimiento a los objetivos planteados, iniciamos el proceso de desarrollo utilizando para ello la metodología de desarrollo orientado a objetos con UML, para el diseño de nuestra aplicación, posteriormente se emplearon las herramientas para programación como son: NetBeans, java, servidor de base de datos MySQL, Hibernate para realizar la fase de implementación del programa.

Luego de la implementación, se procedió a realizar una serie de pruebas para depurar errores existentes, así como realizar mejoras al programa, para ello utilizamos un cuestionario dirigido a los usuarios del sistema (estudiantes, docentes).

### 4.6 DESCRIPCIÓN DE REQUERIMIENTOS

Para la obtención de los requerimientos con los que debía cumplir la aplicación a desarrollar y siguiendo la metodología de desarrollo orientado a objetos con UML, se procedió a elaborar una encuesta con una serie de preguntas que fueron aplicadas a diferentes personas como son: estudiantes y docentes del AEIRNNR, a continuación presentamos los resultados obtenidos.

#### 4.6.1 REQUERIMIENTOS FUNCIONALES

Tabla 4.1 Requerimientos Funcionales

<b><i>CODIGO</i></b>	<b><i>DESCRIPCIÓN</i></b>	<b><i>CATEGORÍA</i></b>
RF001	Deberá permitir una interacción síncrona de audio y video entre diferentes usuarios al mismo tiempo	Evidente



<b>CODIGO</b>	<b>DESCRIPCIÓN</b>	<b>CATEGORÍA</b>
RF002	Deberá permitir a los usuarios la conversación por texto	Evidente
RF003	Permitirá administrar los usuarios por parte del administrador del sistema	Evidente
RF004	Permitirá administrar la información en la parte web de la aplicación (noticias, foros)	Evidente
RF005	Deberá permitir la consulta de información por parte de los usuarios del sistema en la parte web de la aplicación (noticias, foros)	Evidente
RF006	Deberá permitir a los usuarios la creación y participación en foros de discusión	Evidente
RF007	Deberá solicitar autenticación a los usuarios que deseen usar el sistema	Evidente
RF008	El sistema permitirá un vínculo de estudio a distancia manteniendo la comunicación con todo el alumnado.	Evidente
RF009	El sistema permitirá la interacción mediante reuniones virtuales.	Evidente

#### 4.6.2 REQUERIMIENTOS NO FUNCIONALES

**Tabla 4.2 Requerimientos no Funcionales**

<b>CODIGO</b>	<b>DESCRIPCIÓN</b>
RNF001	Deberá ser una aplicación web en cuanto a la administración del sistema, consulta de noticias y participación en foros de discusión



<b><i>CODIGO</i></b>	<b><i>DESCRIPCIÓN</i></b>
RNF002	Deberá ser de fácil uso para el usuario
RNF003	Estará desarrollado en lenguaje Java, JSP, JSF, HIBERNATE
RNF004	La videoconferencia se realizará con ambiente de escritorio utilizando JMF (java media framework)
RNF005	La videoconferencia utilizará el estándar H323
RNF006	Se usara como servidor de base de datos MySQL,
RNF007	Deberá funcionar sobre Internet 2
RNF009	Utilizará el protocolo de Internet versión 4 (IPv4), con compatibilidad para usar el protocolo de Internet versión 6 (IPv6)
RNF010	El sistema debe tener una interfaz grafica de usuario amigable.

#### 4.7 OBTENCIÓN DE CLASES DEL DOMINIO Y DEFINICIÓN DE CONCEPTOS

- Administrador del Sistema.- Persona encargada de organizar la información en el portal web.
- Usuario.- Persona que se beneficia de los servicios prestados por un sistema de información y comunicaciones.
- Noticia.- Contenido de una comunicación puesta a conocimiento del público a través de diferentes medios.
- Foro.- Característica de un sitio web, que permite la creación de temas de discusión así como responder a uno ya existente, para ello se debe ser un usuario registrado.
- Videoconferencia.- Comunicación en doble sentido, interactiva entre dos



puntos geográficamente separados utilizando audio y video.

- Modulo.- Parte de una aplicación que puede contener información sobre noticias, cursos, eventos.
- Aplicación de software.- Programa que permite utilizar determinadas características del computador de una forma fácil y rápida, permite optimizar las comunicaciones entre personas.



## 4.7.1 MODELO DEL DOMINIO

modelo del dominio

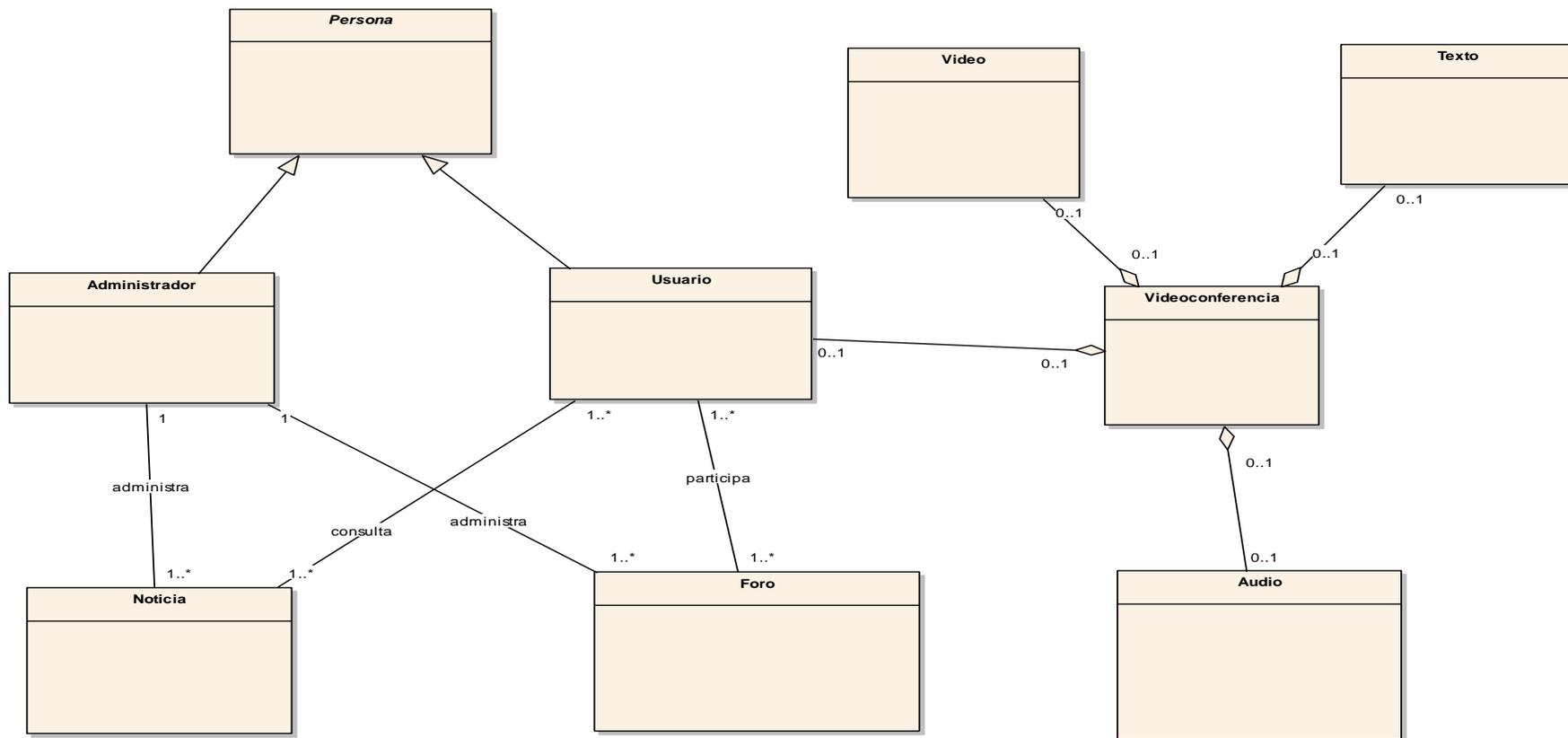


Figura 4.1 Diagrama Modelado del dominio



#### 4.8 MODELAMIENTO DE CASOS DE USO

Para el modelamiento de casos de uso, se procedió a realizar la identificación de cada uno de ellos, así como los actores que participan en los mismos y la meta que deben cumplir, se detalla en la tabla 4.3.

**Tabla 4.3 Modelamiento de casos de uso**

<b>Actor</b>	<b>Meta</b>	<b>CU</b>
Usuario del sistema	x Consultar noticia	x Consultar noticia
	x Participar en foro	x Participar en foro
	x Participar en videoconferencia	x Participar en videoconferencia
Administrador del sistema	x Crear, modificar, eliminar usuarios	x Administrar usuario
	x Ingresar, modificar, eliminar noticias	x Administrar noticia
	x Eliminar foro de discusión	x Administrar foro de discusión

A continuación en la tabla 4.4 se presenta la relación entre casos de uso y los requerimientos que cubre cada uno de ellos.

**Tabla 4.4 Relación Casos de Uso - Requerimientos**

<b>Nombre del Caso de Uso</b>	<b>Referencia de Requerimientos</b>
Administrar usuario	RF003
Administrar noticia	RF004



Administrar foro de discusión	RF004
Consultar noticia	RF005
Participar en foro de discusión	RF006
Participar en videoconferencia	RF001 - RF002 - RF007 - RF008 – RF009

#### 4.8.1 DIAGRAMA DE CASOS DE USO

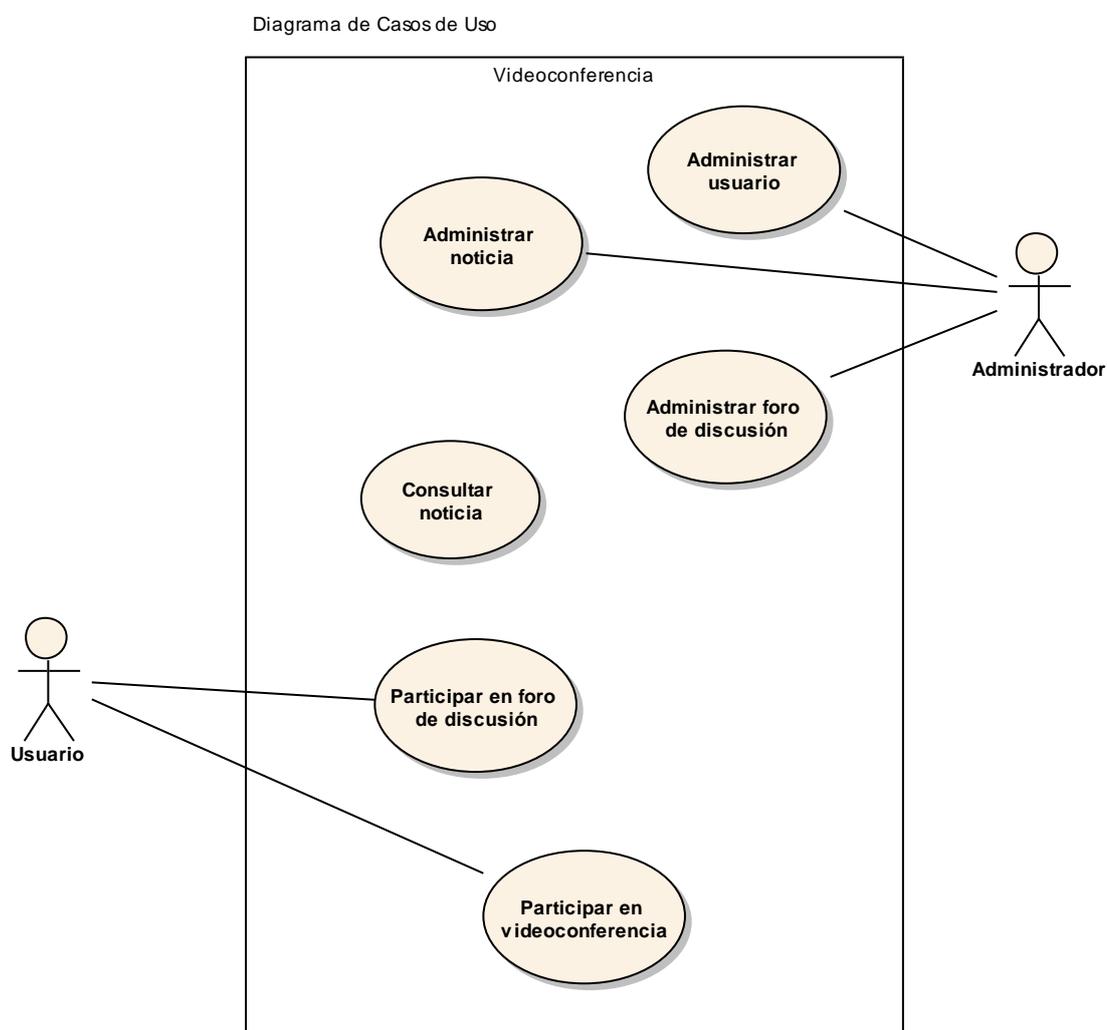


Figura 4.2 Diagrama de Casos de Uso





## 4.8.2 PROTOTIPO DE PANTALLAS

- Aplicación web de administración del sistema, consulta de noticias y participación en foros de discusión

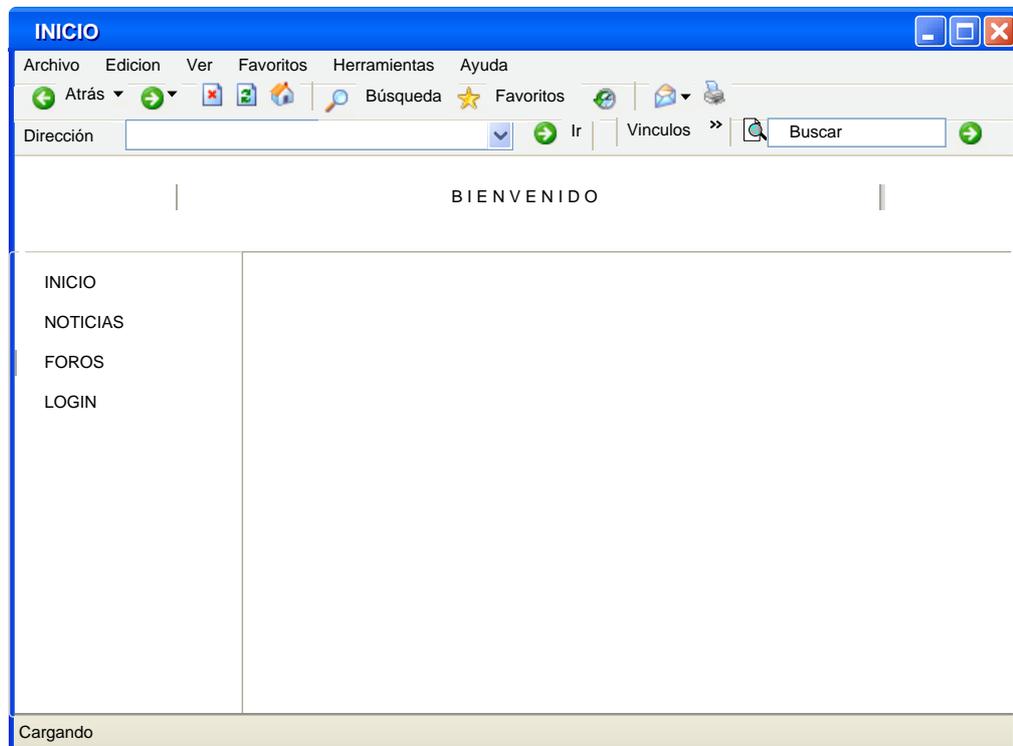


Figura 4.3 Pantalla de Inicio

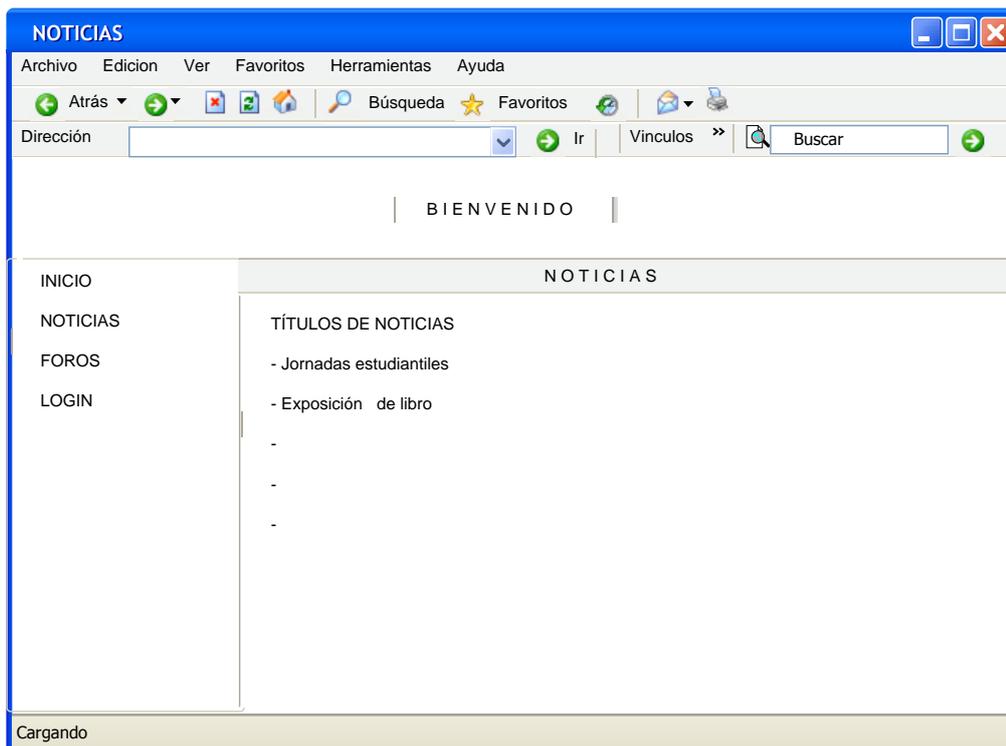


Figura 4.4 Pantalla de Noticias

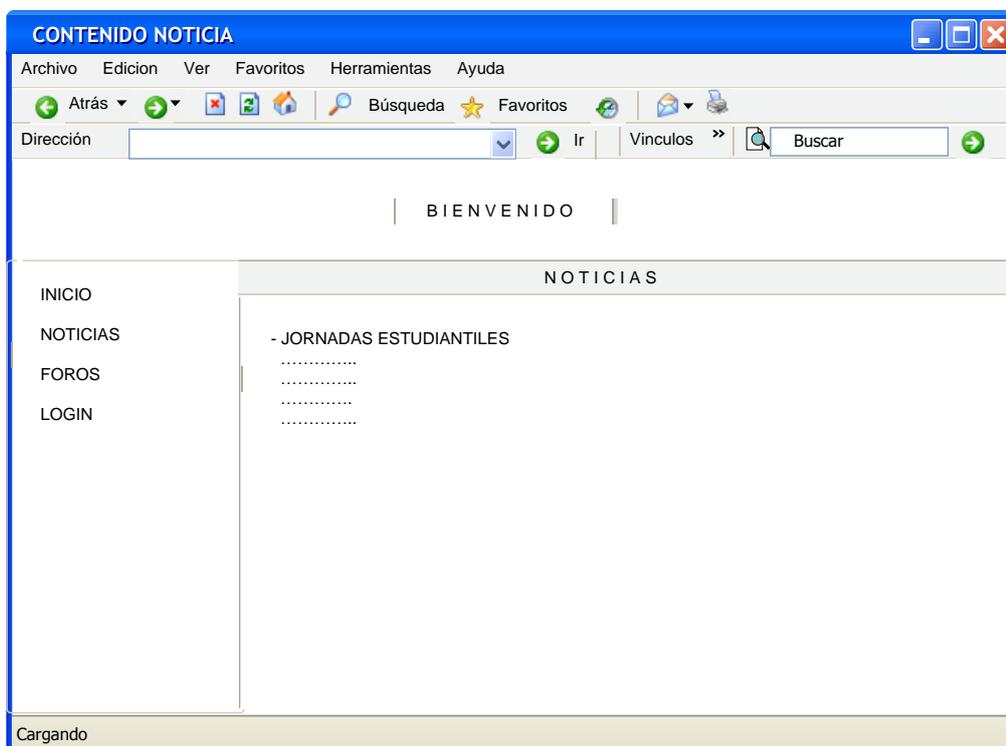


Figura 4.5 Pantalla con el contenido de una noticia

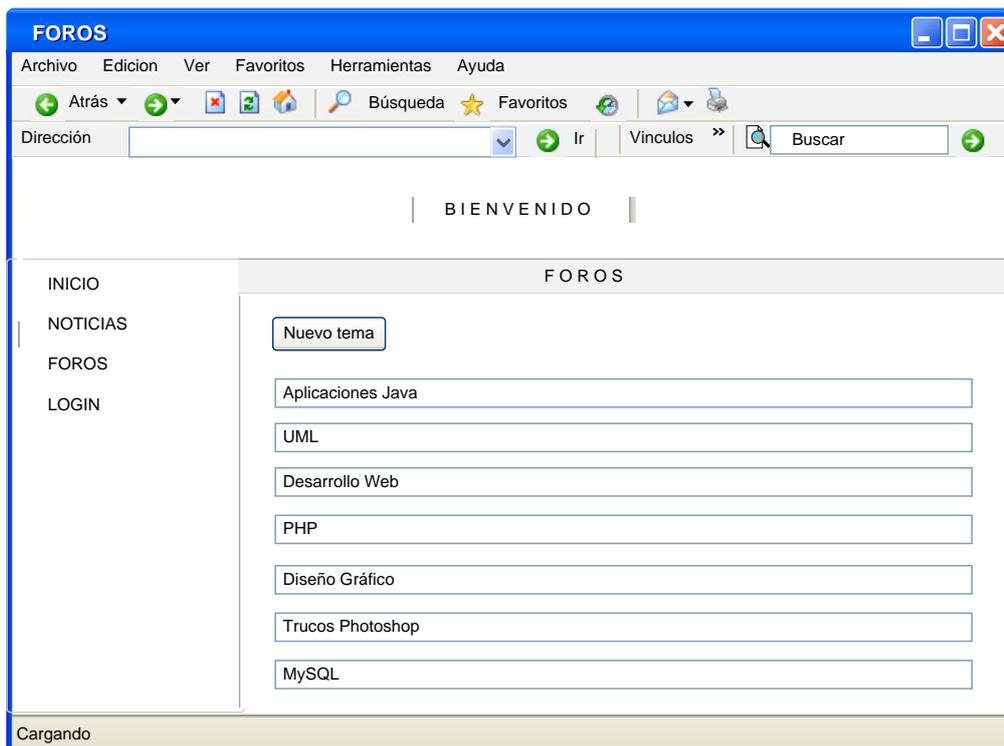


Figura 4.6 Pantalla de Foros

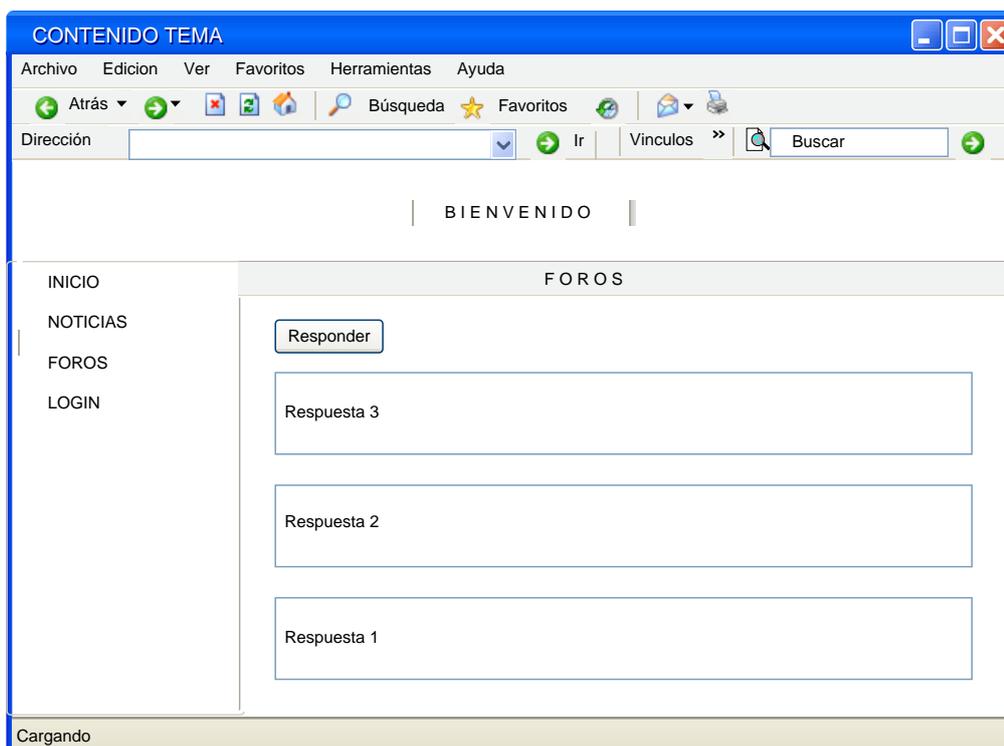




Figura 4.7 Pantalla para cada tema con sus respuestas

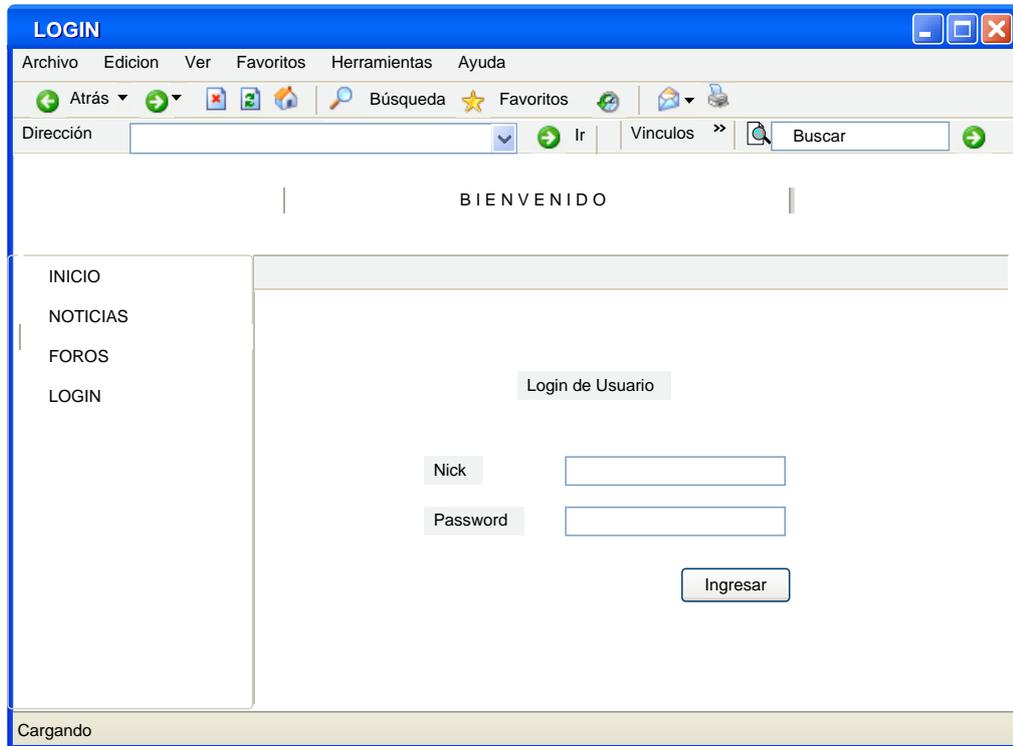


Figura 4.8 Pantalla de Login



CUENTA USUARIO

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Ir Vinculos Buscar

Dirección

USUARIO

País :

Isntitución:

Nick:

e-mail:

Nombre:

Modificar datos Salir

Cargando

Figura 4.9 Pantalla para la Cuenta de Usuario

CUENTA ADMINISTRADOR

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Ir Vinculos Buscar

Dirección

ADMINISTRADOR

País :

Isntitución:

Nick:

e-mail:

Nombre:

Buscar usuario Buscar noticias Buscar terminar foro Modificar datos Salir

Cargando

Figura 4.10 Pantalla para la Cuenta de Administrador



**BUSCAR USUARIO**

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos

Dirección Ir Vinculos Buscar

**USUARIO**

Busqueda:

Buscar por:

**BUSCAR**

Resultado de búsqueda


**NUEVO**

**MODIFICAR**

**ELIMINAR**

Cargando

**Figura 4.11 Pantalla para Búsqueda de Usuarios**



FORMULARIO USUARIO

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección [ ] Ir Vinculos [ ] Buscar [ ]

USUARIO

Datos Personales

Nombre: [ ]

Apellido: [ ]

Institución: [ ]

e-mail: [ ] Sexo [ ]

Dirección

País: [ ]

Ciudad: [ ]

GUARDAR CANCELAR

Cargando

Figura 4.12 Pantalla para ingreso y edición de usuarios

BUSCAR NOTICIA

Archivo Edición Ver Favoritos Herramientas Ayuda

Dirección [ ] Ir Vinculos [ ] Buscar [ ]

NOTICIA

Busqueda: [ ]

Buscar por: [ ] BUSCAR

Resultado de búsqueda


NUEVO

MODIFICAR

ELIMINAR

Cargando

Figura 4.13 Pantalla para Búsqueda de Noticias



FORMULARIO NOTICIA

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Ir Vinculos Buscar

Dirección

NOTICIA

Escriba Texto

Titulo:

Autor:

Texto:

GUARDAR CANCELAR

Cargando

Figura 4.14 Pantalla para ingreso y edición de noticias

BUSCAR TEMAS DEL FORO

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Ir Vinculos Buscar

Dirección

FOROS

Busqueda:

Buscar por:

BUSCAR

NUEVO

MODIFICAR

ELIMINAR

Resultado de búsqueda

Codigo	Autor	Fecha de Publicación

Cargando



Figura 4.15 Pantalla para Búsqueda de Temas

FORMULARIO FORO

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Ir Vinculos Buscar

Dirección

BIENVENIDO

INICIO  
NOTICIAS  
FOROS  
LOGIN

FOROS

Autor:

Título: <?=\$titulo?>

Mensaje: <?=\$mensaje?>

Enviar

Cargando

Figura 4.16 Pantalla para ingreso y edición de temas

- Aplicación de escritorio para videoconferencia

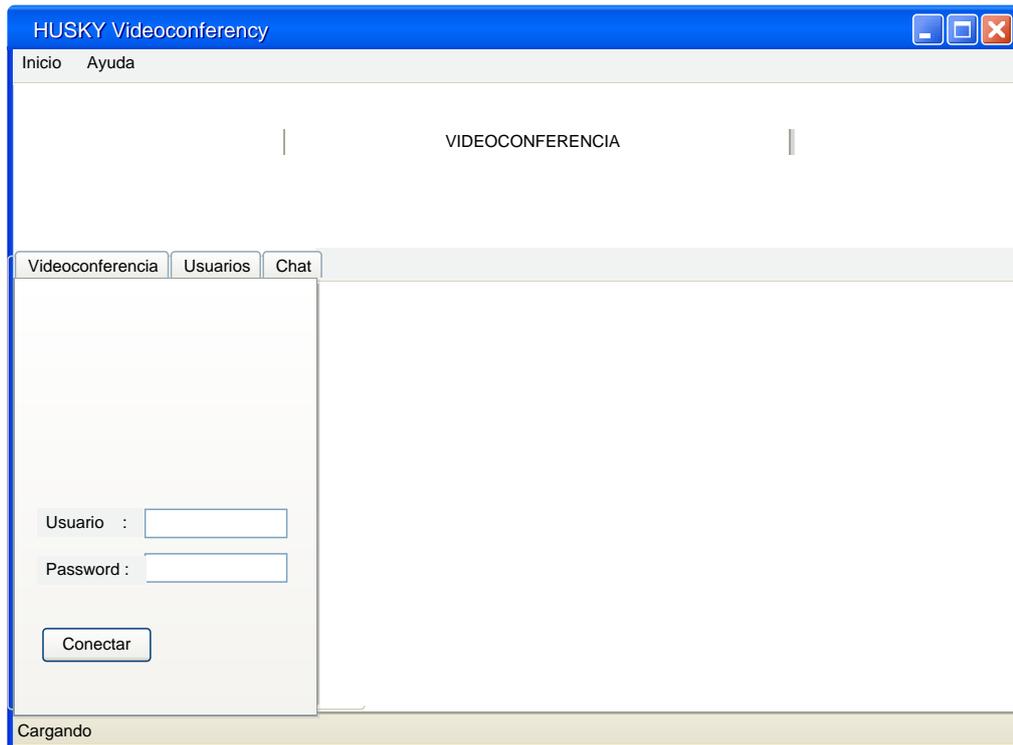


Figura 4.17 Pantalla principal de Husky Videoconferency

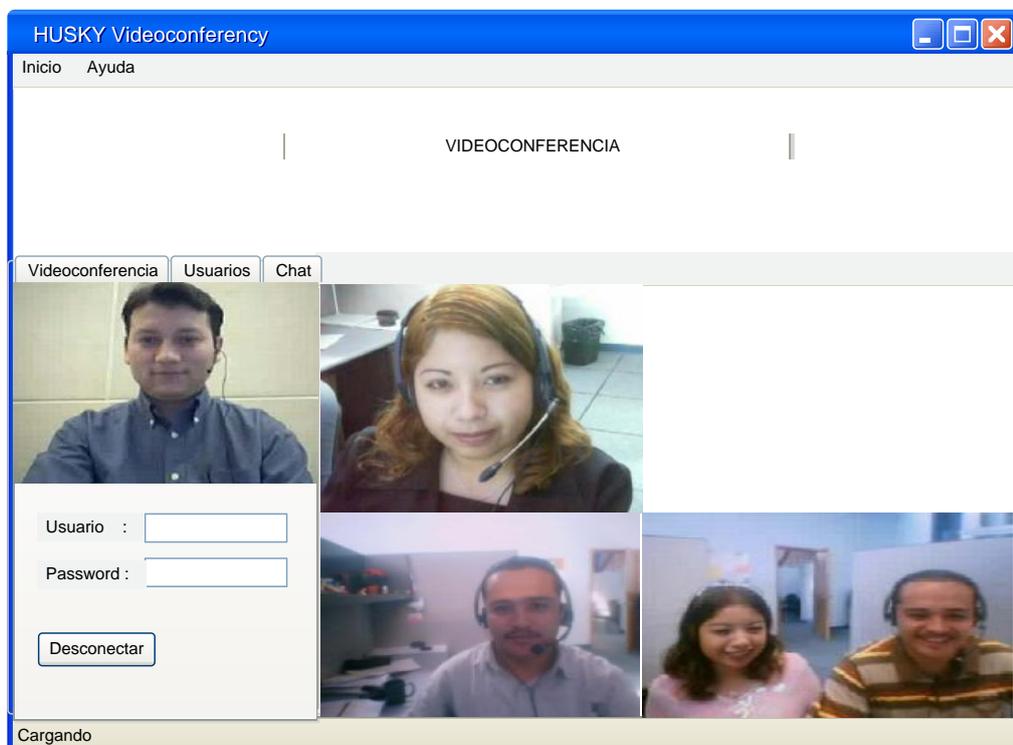


Figura 4.18 Pantalla principal de Husky Videoconferency con sesión iniciada

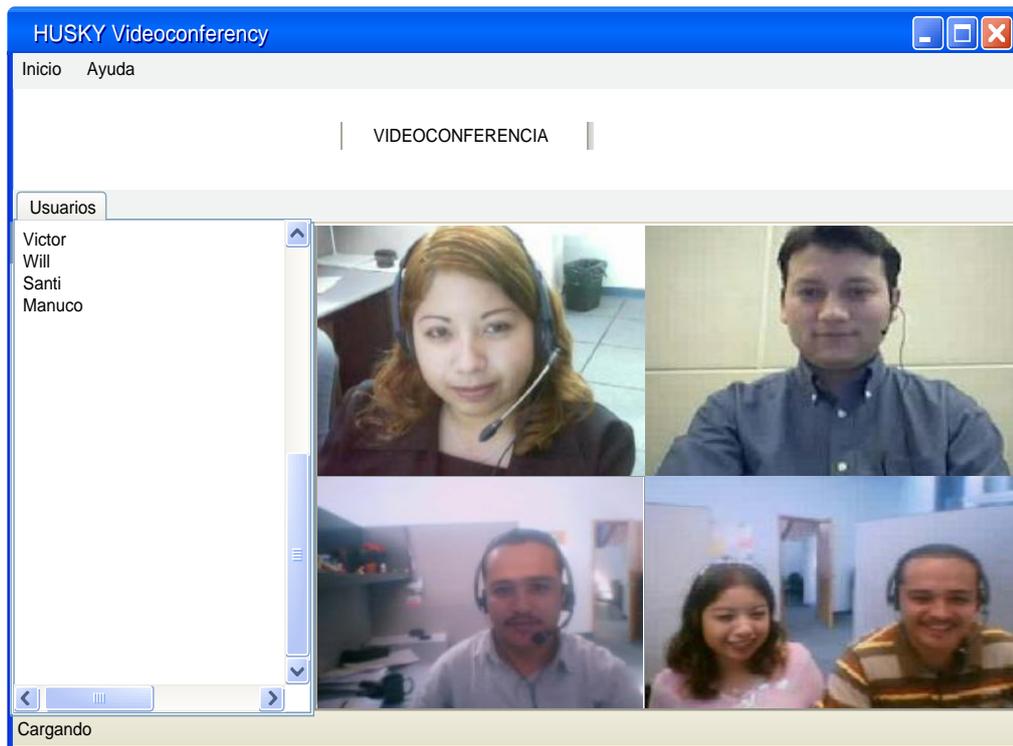


Figura 4.19 Pantalla principal de Husky Videoconferency con lista de usuarios

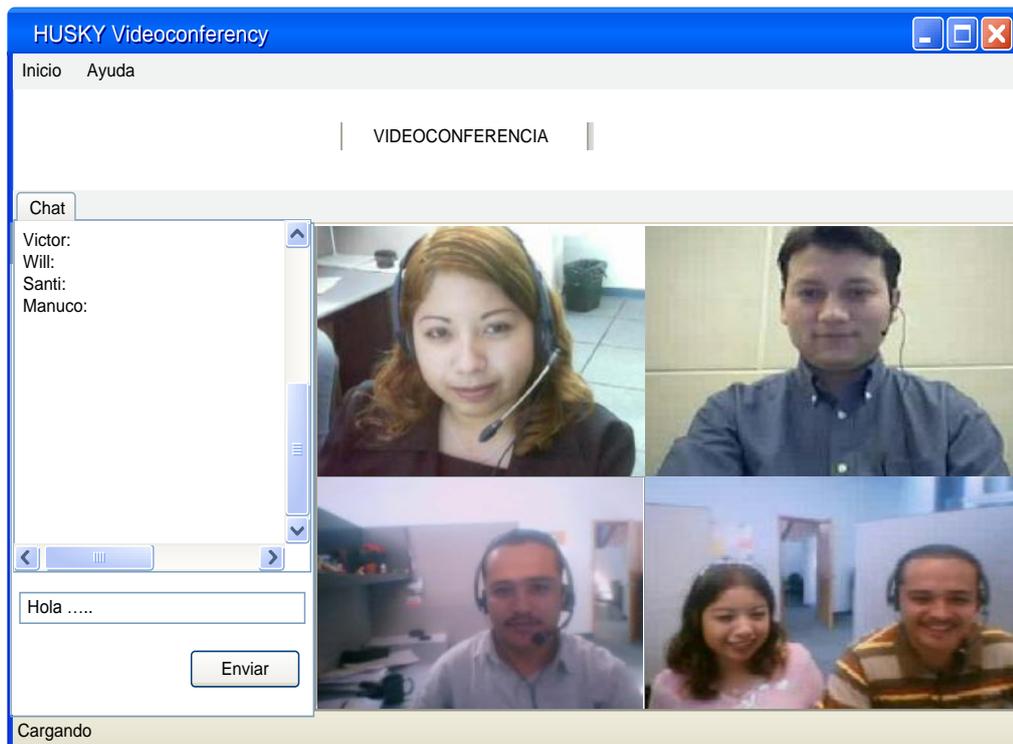




Figura 4.20 Pantalla principal de Husky Videoconferency con chat



Figura 4.21 Mensaje de error

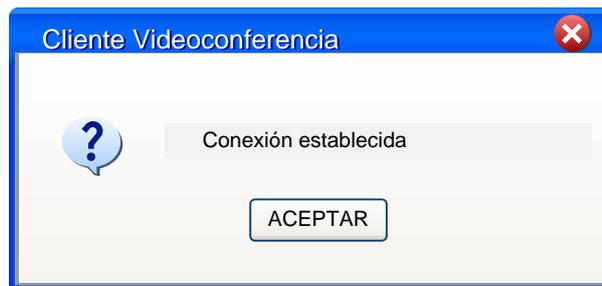


Figura 4.22 Mensaje de información

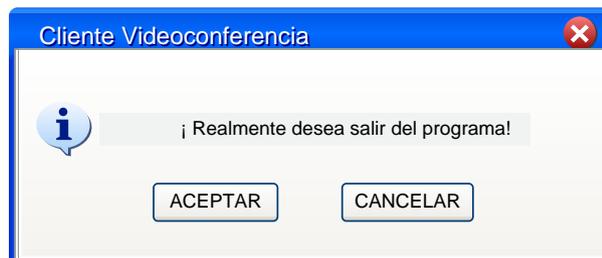


Figura 4.23 Mensaje de confirmación





### 4.8.3 DESCRIPCIÓN DE CASOS DE USO

#### 4.8.3.1 Caso de uso: Administrar Usuario

Tabla 4.5 Descripción del Caso de Uso Administrar Usuario

<b>Nombre CU:</b>	Administrar usuario	<b>Identificación CU:</b>	UC001
<b>Descripción:</b>	El administrador del sistema ingresa nuevos usuarios con sus respectivos datos personales, también puede modificar y dar de baja usuarios		
<b>Actor:</b>	Administrador del sistema		
<b>Referencia de requerimientos:</b>	RF003		
<b>Precondiciones:</b>	1. El administrador haya ingresado al sistema 2. El administrador del sistema haya activado la pantalla <b>Buscar usuario</b>		
<b>Poscondiciones:</b>	1. Buscar usuarios del sistema con sus respectivos datos personales 2. Modificar los datos personales de los usuarios 3. Crear nuevo usuario 4. Dar de baja usuarios		
<b>Tipo CU:</b>	Primario Esencial		
<b>CURSO NORMAL DE EVENTOS</b>			
<b>Acción del actor</b>		<b>Respuesta del sistema</b>	
1. El administrador del sistema ingresa el texto que desea buscar en el campo <b>Búsqueda</b> de la pantalla <b>Buscar usuario</b> .			
2. El administrador del sistema selecciona el criterio de búsqueda en el campo <b>Buscar</b>			



por.

3. El administrador del sistema hace clic en el botón **[BUSCAR]**.

4. El sistema valida que el campo **Búsqueda** no este vacío.

5. El sistema busca los usuarios de acuerdo al texto ingresado en el campo búsqueda y el criterio elegido.

6. El sistema muestra todos los usuarios encontrados en la tabla **Resultados de búsqueda** de la pantalla **Buscar usuario**.

7. El administrador del sistema selecciona un usuario en la tabla **Resultados de búsqueda** de la pantalla **Buscar usuario**.

8. El administrador del sistema selecciona **[MODIFICAR]** de la pantalla **Buscar usuario**.

9. El sistema muestra la pantalla **Formulario usuario** con los datos del usuario seleccionado.

10. El administrador del sistema modifica la información del usuario en la pantalla **Formulario usuario**.

11. El administrador del sistema hace clic en el botón **[GUARDAR]** de la pantalla **Formulario usuario**.



	<p>12. El sistema valida que los campos requeridos de la pantalla <b>Formulario usuario</b> estén llenos.</p> <p>13. El sistema actualiza los cambios efectuados del usuario.</p> <p>14. El sistema presenta un mensaje de confirmación indicando que se realizó la actualización correctamente.</p> <p>15. El sistema cierra el <b>Formulario usuario</b>.</p> <p>16. Finaliza el caso de uso.</p>
<b>CURSOS ALTERNOS DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p><b>A. TEXTO VACIO</b></p> <p>A5. El sistema presenta un mensaje de error indicando que el campo <b>Búsqueda</b> está vacío.</p> <p>A6. El caso de uso continúa en el paso 1 del curso normal de eventos.</p> <p><b>B. USUARIO NO ENCONTRADO</b></p> <p>B6. El sistema muestra un mensaje de aviso indicando que el usuario no existe con ese texto o criterio de búsqueda.</p> <p>B7. El caso de uso continúa en</p>



### C. NUEVO USUARIO

C7. El administrador del sistema hace clic en el botón **[NUEVO]** de la pantalla **Buscar usuario**.

C9. El administrador del sistema ingresa los datos del nuevo usuario en la pantalla **Formulario usuario**.

C10. El administrador del sistema hace clic en el botón **[GUARDAR]** de la pantalla **Formulario usuario**.

### D. ELIMINAR USUARIO

el paso 1 del curso normal de eventos.

C8. El sistema crea un nuevo usuario y muestra la pantalla **Formulario usuario**.

C11. El sistema valida que los campos indispensables de la pantalla **Formulario usuario** estén llenos.

C12. El sistema guarda la información del nuevo usuario.

C13. El sistema presenta un mensaje de confirmación indicando que se guardó la información correctamente.

C14. El caso de uso continúa en el paso 15 del curso normal de eventos.



D8. El administrador del sistema hace clic en el botón **[ELIMINAR]** de la pantalla **Buscar usuario**.

D10. El administrador hace clic en **[ACEPTAR]** en el cuadro de dialogo **Eliminar usuario**.

D9. El sistema presenta un mensaje solicitando confirmación.

D11. El sistema borra la información del usuario.

D12. El sistema presenta un mensaje de confirmación indicando que se borró la información del usuario.

D13. El caso de uso continúa en el paso 16 del curso normal de eventos.



#### 4.8.3.2 Caso de uso: Administrar Noticia

**Tabla 4.6 Descripción del Caso de Uso Administrar Noticia**

<b>Nombre CU:</b>	Administrar noticia	<b>Identificación CU:</b>	UC002
<b>Descripción:</b>	El administrador del sistema ingresa nuevas noticias y da de baja la información que ha cumplido cierto periodo en el sistema.		
<b>Actor:</b>	Administrador del sistema		
<b>Referencia de requerimientos:</b>	RF004		
<b>Precondiciones:</b>	<ol style="list-style-type: none"> <li>1. El administrador haya ingresado sistema</li> <li>2. El administrador del sistema haya activado la pantalla <b>Buscar noticia</b></li> </ol>		
<b>Poscondiciones:</b>	<ol style="list-style-type: none"> <li>1. Buscar noticias</li> <li>2. Modificar la información de las noticias</li> <li>3. Crear nuevo noticia</li> <li>4. Dar de baja noticias</li> </ol>		
<b>Tipo CU:</b>	Primario Esencial		
<b>CURSO NORMAL DE EVENTOS</b>			
<b>Acción del actor</b>		<b>Respuesta del sistema</b>	
<ol style="list-style-type: none"> <li>1. El administrador del sistema ingresa el texto que desea buscar en el campo <b>Búsqueda</b> de la pantalla <b>Buscar noticia</b>.</li> <li>2. El administrador del sistema selecciona el criterio de búsqueda en el campo <b>Buscar por</b> de la pantalla <b>Buscar noticias</b></li> <li>3. El administrador del sistema elige <b>[BUSCAR]</b> en la pantalla <b>Buscar noticia</b></li> </ol>		<ol style="list-style-type: none"> <li>4. El sistema valida que el texto</li> </ol>	



<p>7. El administrador del sistema selecciona una noticia en la tabla <b>Resultados de búsqueda</b> de la pantalla <b>Buscar noticia</b></p> <p>8. El administrador del sistema selecciona <b>[MODIFICAR]</b> de la pantalla <b>Buscar noticia</b></p> <p>10. El administrador del sistema modifica la información de la noticia en la pantalla <b>Formulario noticia</b></p> <p>11. El administrador del sistema elige <b>[GUARDAR]</b> de la pantalla <b>Formulario noticia</b></p>	<p>del campo <b>Búsqueda</b> no este vacío</p> <p>5. El sistema busca las noticias de acuerdo al texto ingresado en el campo búsqueda y el criterio elegidos</p> <p>6. El sistema muestra todas las noticias encontradas en la tabla <b>Resultados de búsqueda</b> de la pantalla <b>Buscar noticia</b>.</p> <p>9. El sistema muestra la pantalla <b>Formulario noticia</b> con los datos de la noticia seleccionada</p> <p>12. El sistema valida que los campos indispensables de la pantalla <b>Formulario noticia</b> estén llenos</p> <p>13. El sistema actualiza la noticia.</p>
---	---



	<p>14. El sistema presenta un mensaje de confirmación indicando que se realizó la actualización correctamente</p> <p>15. El sistema cierra la pantalla <b>Formulario noticia</b></p> <p>16. Finaliza el caso de uso</p>
<b>CURSOS ALTERNOS DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<p><b>A. TEXTO VACÍO</b></p> <p>A5. El sistema presenta un mensaje de error indicando que el texto del campo <b>Búsqueda</b> está vacío</p> <p>A6. El caso de uso continua en el paso 1 del curso normal de eventos</p> <p><b>B. NOTICIA NO ENCONTRADA</b></p> <p>B6. El sistema muestra un mensaje de aviso indicando que la noticia no existe con ese texto o criterio de búsqueda</p> <p>B7. El caso de uso continua en el paso 1 del curso normal de eventos</p>
<p><b>C. NUEVA NOTICIA</b></p> <p>C7. El administrador del sistema elige <b>[NUEVO]</b> de la pantalla <b>Buscar noticia</b></p>	



C9. El administrador del sistema ingresa la información de la noticia en la pantalla **Formulario noticia**

C10. El administrador del sistema hace clic en el botón **[GUARDAR]** de la pantalla **Formulario noticia**

#### **D. ELIMINAR NOTICIA**

D8. El administrador del sistema selecciona **[ELIMINAR]** de la pantalla **Buscar noticia**

D10. El administrador elige **[ACEPTAR]** en el cuadro de dialogo **Eliminar noticia**

C8. El sistema crea un nueva noticia y muestra la pantalla **Formulario noticia**

C11. El sistema valida que los campos indispensables de la pantalla **Formulario noticia** estén llenos

C12. El sistema guarda la información de la nueva noticia

C13. El sistema presenta un mensaje de confirmación indicando que se guardo la información correctamente

C14. El caso de uso continua en el paso 15 del curso normal de eventos

D9. El sistema presenta un mensaje solicitando confirmación



	<p>D11. El sistema borra la información de la noticia</p> <p>D12. El sistema presenta un mensaje de confirmación indicando que se borró la información de la noticia.</p> <p>D13. El caso de uso continua en el paso 16 del curso normal de eventos</p>
--	---

#### 4.8.3.3 Caso de uso: Administrar foro de discusión

Tabla 4.7 Descripción del Caso de Uso Administrar foro de discusión

<b>Nombre CU:</b>	Administrar foro de discusión	<b>Identificación CU:</b>	UC003
<b>Descripción:</b>	El administrador del sistema puede dar de baja aquellos temas que llevan un determinado periodo en el foro, o aquellos que puedan llegar a ser prohibidos		
<b>Actor:</b>	Administrador del sistema		
<b>Referencia de requerimientos:</b>	RF004		
<b>Precondiciones:</b>	<ol style="list-style-type: none"> <li>1. El administrador haya ingresado al sistema</li> <li>2. El administrador del sistema haya activado la pantalla <b>Buscar temas del foro</b></li> </ol>		
<b>Poscondiciones:</b>	<ol style="list-style-type: none"> <li>1. Dar de baja temas del foro con sus respectivas respuestas</li> </ol>		
<b>Tipo CU:</b>	Primario Esencial		
<b>CURSO NORMAL DE EVENTOS</b>			



Acción del actor	Respuesta del sistema
<p>1. El administrador del sistema ingresa el texto que desea buscar en el campo <b>Búsqueda</b> de la pantalla <b>Buscar temas del foro</b></p> <p>2. El administrador del sistema selecciona el criterio de búsqueda en el campo <b>Buscar por</b> de la pantalla <b>Buscar temas del foro</b></p> <p>3. El administrador del sistema elige <b>[BUSCAR]</b> en la pantalla <b>Buscar temas del foro</b></p> <p>7. El administrador del sistema selecciona un tema en la tabla <b>Resultados de búsqueda</b> de la pantalla <b>Buscar temas del foro</b></p> <p>8. El administrador del sistema selecciona <b>[ELIMINAR]</b> de la pantalla <b>Buscar temas del foro</b></p> <p>10. El administrador del sistema elige <b>[ACEPTAR]</b> en el cuadro de dialogo <b>Eliminar</b></p>	<p>4. El sistema valida que el texto del campo <b>Búsqueda</b> no esté vacío</p> <p>5. El sistema busca los temas de acuerdo al texto y criterio de búsqueda elegidos</p> <p>6. El sistema muestra todos los temas encontrados en la tabla <b>Resultados de búsqueda</b> de la pantalla <b>Buscar temas del foro</b></p> <p>9. El sistema presenta un mensaje solicitando confirmación</p>



tema	11. El sistema borra la información del tema del foro  12. El sistema presenta un mensaje de confirmación indicando que se borró el tema  13. Finaliza el caso de uso
<b>CURSOS ALTERNOS DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	<b>A. TEXTO VACIO</b>  A5. El sistema presenta un mensaje de error indicando que el texto del campo <b>Búsqueda</b> está vacío  A6. El caso de uso continua en el paso 1 del curso normal de eventos  <b>B. TEMA NO ENCONTRADO</b>  B6. El sistema muestra un mensaje de aviso indicando que el tema no existe con ese texto o criterio de búsqueda  B7. El caso de uso continúa en el paso 1 del curso normal de eventos





#### 4.8.3.4 Caso de uso: Consultar Noticia

Tabla 4.8 Descripción del Caso de Uso Consultar Noticia

<b>Nombre CU:</b>	Consultar Noticia	<b>Identificación CU:</b>	UC004
<b>Descripción:</b>	El usuario realiza consultas de noticias		
<b>Actor:</b>	Usuario del sistema		
<b>Referencia de requerimientos:</b>	RF005		
<b>Precondiciones:</b>	1. El usuario haya activado la pantalla <b>Home</b>		
<b>Poscondiciones:</b>	1. Consultar noticias con su respectiva información.		
<b>Tipo CU:</b>	Primario Esencial		
<b>CURSO NORMAL DE EVENTOS</b>			
<b>Acción del actor</b>		<b>Respuesta del sistema</b>	
1. El usuario elige la opción [ <b>NOTICIAS</b> ] de la pantalla <b>Home</b>			
3. El usuario elige la noticia de su interés en la pantalla <b>Noticias</b>		2. El sistema presenta la pantalla <b>Noticias</b> con la lista de noticias existentes  4. El sistema presenta la pantalla <b>Contenido noticia</b> con la noticia seleccionada y su contenido  5. Finaliza el caso de uso	





#### 4.8.3.5 Caso de uso: Participar en foro de discusión

Tabla 4.9 Descripción del Caso de Uso Participar en foro de discusión

<b>Nombre CU:</b>	Participar en foro de discusión	<b>Identificación CU:</b>	UC005
<b>Descripción:</b>	El usuario puede responder a temas existentes en el foro, también puede crear nuevos temas		
<b>Actor:</b>	Usuario del sistema		
<b>Referencia de requerimientos:</b>	RF006		
<b>Precondiciones:</b>	1. El usuario se haya identificado en el sistema 2. El usuario haya activado la pantalla <b>Home</b>		
<b>Poscondiciones:</b>	1. Responder a temas existentes 2. Creación de nuevos temas de discusión		
<b>Tipo CU:</b>	Primario Esencial		
<b>CURSO NORMAL DE EVENTOS</b>			
<b>Acción del actor</b>		<b>Respuesta del sistema</b>	
1. El usuario escoge la opción <b>[FOROS]</b> de la pantalla <b>Home</b>			
3. El usuario elige el tema de su interés en la pantalla <b>Foros</b>		2. El sistema presenta todos los temas existentes en la pantalla <b>Foros</b>  4. El sistema presenta todas las respuestas relacionadas al tema seleccionado en la pantalla <b>Contenido tema</b>	



<p>5. El usuario elige la opción <b>[RESPONDER]</b> de la pantalla <b>Contenido tema</b></p> <p>7. El usuario llena los campos con la información requerida</p> <p>8. El usuario elige <b>[ENVIAR]</b> en la pantalla <b>Formulario foro</b></p>	<p>6. El sistema presenta la pantalla <b>Formulario foro</b></p> <p>9. El sistema valida que los campos requeridos de la pantalla <b>Formulario foro</b> estén llenos</p> <p>10. El sistema actualiza la información del tema de discusión</p> <p>11. El sistema presenta un mensaje de confirmación indicando que se actualizó la información con éxito</p> <p>12. Finaliza el caso de uso</p>
--	---

**CURSOS ALTERNOS DE EVENTOS**

<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p><b>A. NUEVO TEMA</b></p> <p>A3. El usuario elige la opción <b>[NUEVO TEMA]</b> de la pantalla <b>Foros</b></p>	<p>A4. El sistema crea un nuevo tema y presenta la pantalla <b>Formulario foro</b></p>



<p>A5. El usuario llena los campos con la información requerida</p> <p>A6. El usuario elige <b>[ENVIAR]</b> en la pantalla <b>Formulario foro</b></p>	<p>A7. El sistema valida que los campos requeridos de la pantalla <b>Formulario foro</b> estén llenos</p> <p>A8. El sistema guarda la información del tema de discusión</p> <p>A9. El sistema presenta un mensaje de confirmación indicando que se guardó la información correctamente</p> <p>A10. El caso de uso continua en el paso 12</p>
---	--

#### 4.8.3.6 Caso de uso: Participar en Videoconferencia

Tabla 4.10 Descripción del Caso de Uso Participar en videoconferencia

<b>Nombre CU:</b>	Participar en Videoconferencia	<b>Identificación CU:</b>	UC006
<b>Descripción:</b>	El usuario se identifica con el sistema, se comunica con otros usuarios a través de audio, video y texto.		
<b>Actor:</b>	Usuario del sistema		
<b>Referencia de</b>	RF001 - RF002 - RF007 - RF008 - RF009		



<b>requerimientos:</b>	
<b>Precondiciones:</b>	El usuario haya ingresado al programa de videoconferencia y se ubique en la pestaña <b>videoconferencia</b>
<b>Poscondiciones:</b>	Accede a los servicios de Videoconferencia.
<b>Tipo CU:</b>	Primario Esencial
<b>CURSO NORMAL DE EVENTOS</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El usuario del sistema ingresa su nombre en el campo <b>Usuario</b> de la pestaña <b>videoconferencia</b>  2. El usuario del sistema ingresa su contraseña en el campo <b>Contraseña</b> de la pestaña <b>videoconferencia</b>  3. El usuario del sistema elige <b>[CONECTAR]</b> de la pestaña <b>videoconferencia</b>	4. El sistema valida el nombre y contraseña del usuario.  5. El sistema verifica que exista el usuario registrado en el sistema.  6. El sistema verifica que estén disponibles los periféricos multimedia  7. El sistema da inicio a la captura de datos a través de los periféricos multimedia



<p>12. El usuario elige <b>[DESCONECTAR]</b> de la pestaña <b>videoconferencia</b></p> <p>14. El usuario elige <b>[Salir]</b> de la pantalla <b>Husky Videoconferency</b></p>	<p>8. El sistema transmite los datos (audio, video, texto) a través de la red</p> <p>9. El sistema recibe los datos (audio, video, texto) desde la red</p> <p>10. El sistema presenta los datos de video en el panel <b>Vista</b> de la pantalla <b>Husky Videoconferency</b></p> <p>11. El sistema emite el audio a través de los altavoces</p> <p>13. El sistema finaliza la captura, transmisión y recepción de datos</p> <p>15. El sistema cierra la pantalla <b>Husky Videoconferency</b></p> <p>16. Finaliza el caso de uso</p>
<p style="text-align: center;"><b>CURSOS ALTERNOS DE EVENTOS</b></p>	
<p style="text-align: center;"><b>Acción del actor</b></p>	<p style="text-align: center;"><b>Respuesta del sistema</b></p>
	<p><b>A. TEXTO VACIO</b></p> <p>A5. El sistema presenta un mensaje de error indicando que el nombre del usuario y/o contraseña están vacíos.</p> <p>A6. El caso de uso continúa en el paso 1 del curso normal de</p>



eventos.

#### **B. USUARIO NO REGISTRADO**

B6. El sistema muestra un mensaje de error indicando que el nombre de usuario y/o contraseña son incorrectos.

B7. El caso de uso continúa en el paso 1 del curso normal de eventos.

#### **C. PERIFÉRICOS MULTIMEDIA NO ENCONTRADOS**

C7. El sistema muestra un aviso indicando que los periféricos no se encuentran disponibles

C8. El caso de uso continúa en el paso 1 del curso normal de eventos

#### 4.8.4 DIAGRAMAS DE ROBUSTEZ

##### 4.8.4.1 Diagramas de Robustez del Caso de Uso Administrar Usuario

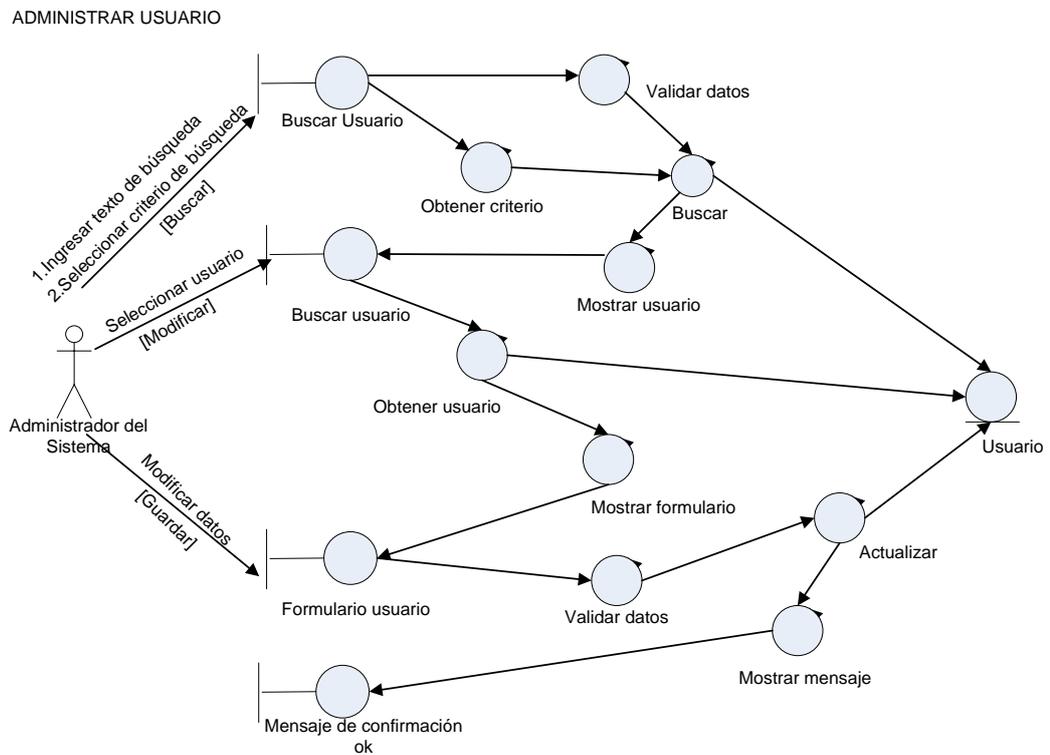


Figura 4.24 Diagrama de robustez del caso de uso Administrar usuario

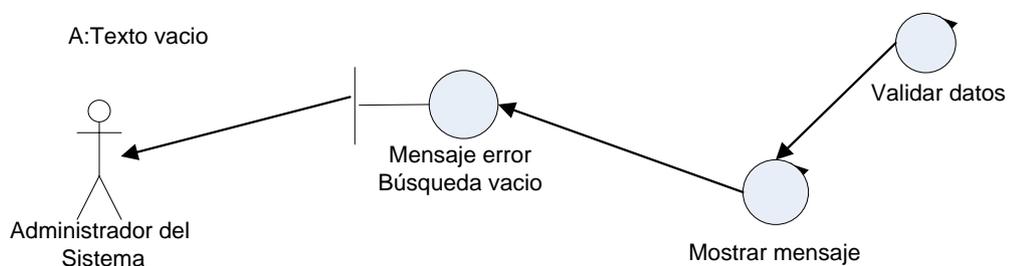


Figura 4.25 Diagrama de robustez del curso alternativo A: Texto vacío del caso de uso Administrar usuario

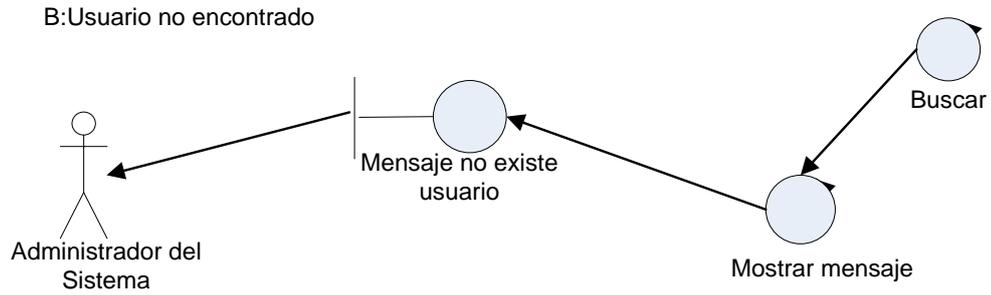


Figura 4.26 Diagrama de robustez del curso alternativo B: Usuario no encontrado del caso de uso Administrar usuario

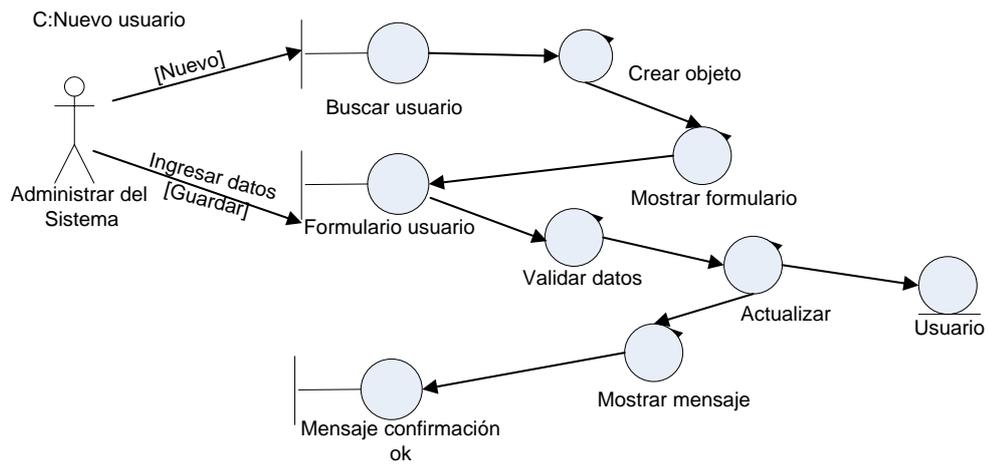


Figura 4.27 Diagrama de robustez del curso alternativo C: Nuevo usuario del caso de uso Administrar usuario

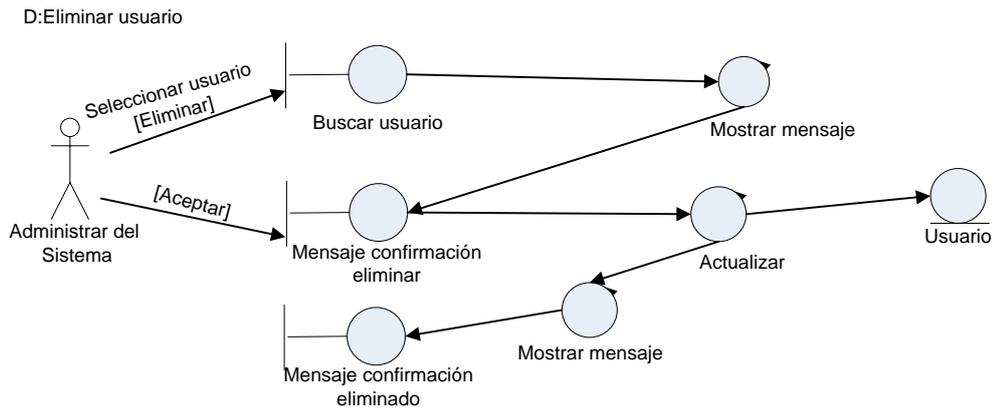


Figura 4.28 Diagrama de robustez del curso alternativo D: Eliminar usuario del caso de uso Administrar usuario

#### 4.8.4.2 Diagramas de Robustez del Caso de Uso Administrar Noticia

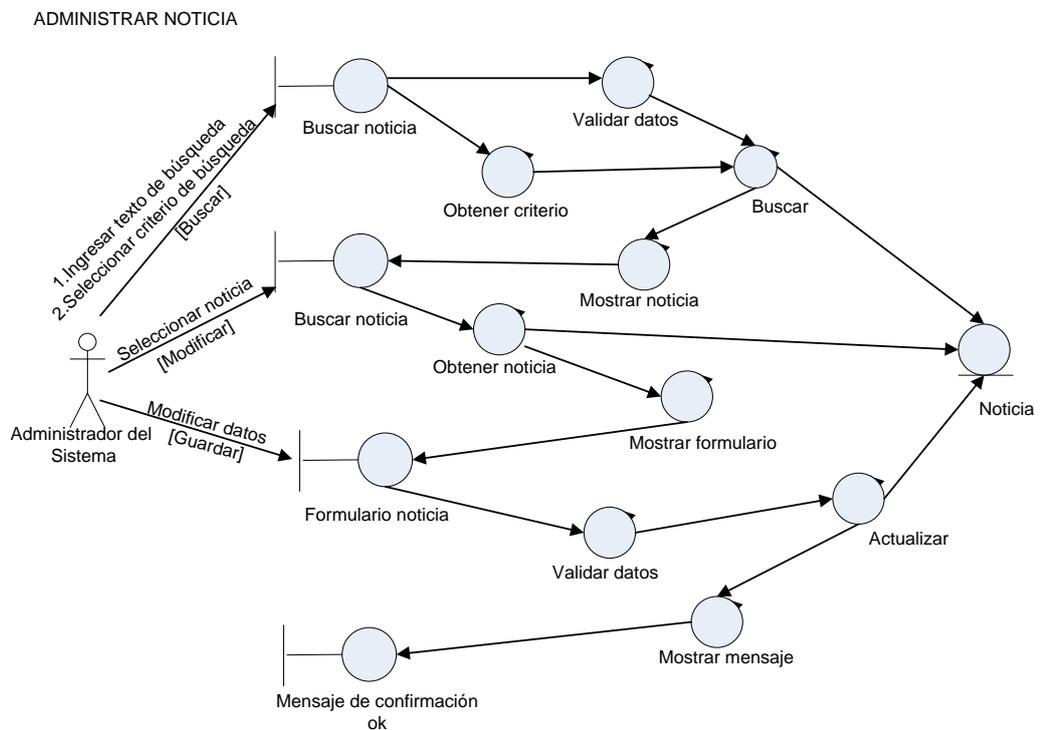
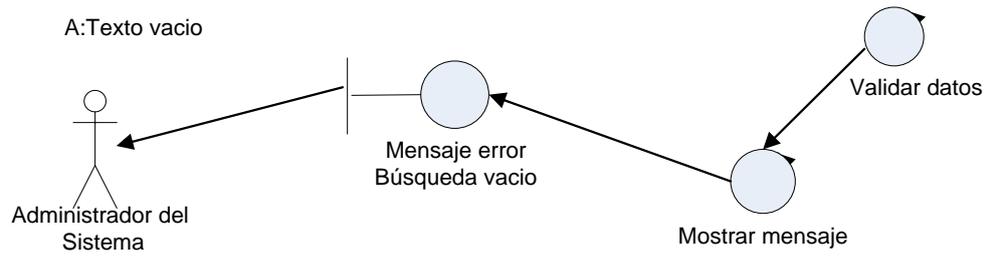
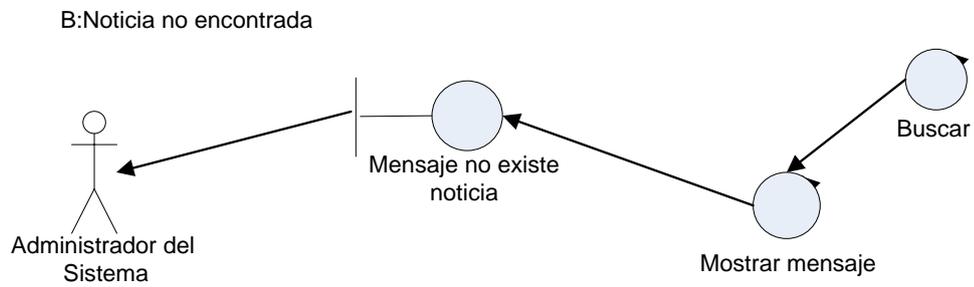


Figura 4.29 Diagrama de robustez del caso de uso Administrar noticia



**Figura 4.30 Diagrama de robustez del curso alternativo A: Texto vacío del caso de uso Administrar noticia**



**Figura 4.31 Diagrama de robustez del curso alternativo B: Noticia no encontrada del caso de uso Administrar noticia**

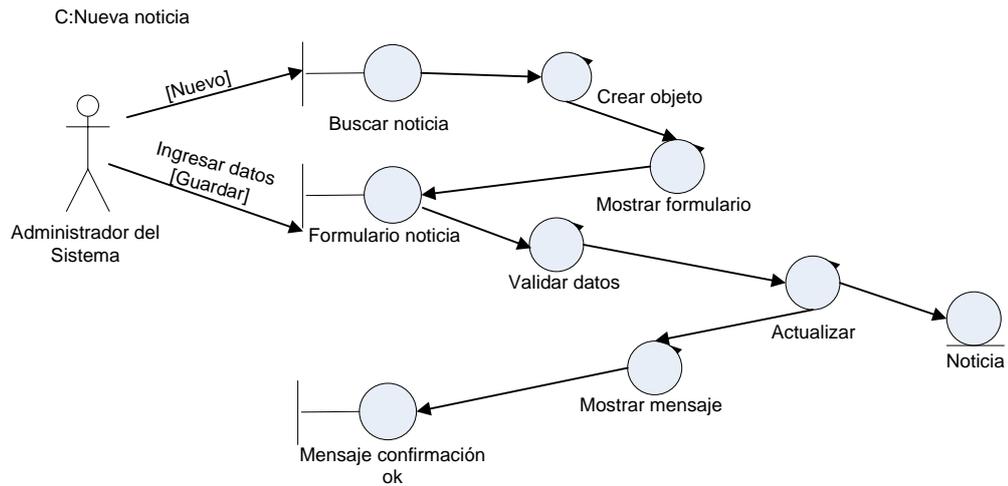


Figura 4.32 Diagrama de robustez del curso alternativo C: Nueva noticia del caso de uso Administrar noticia

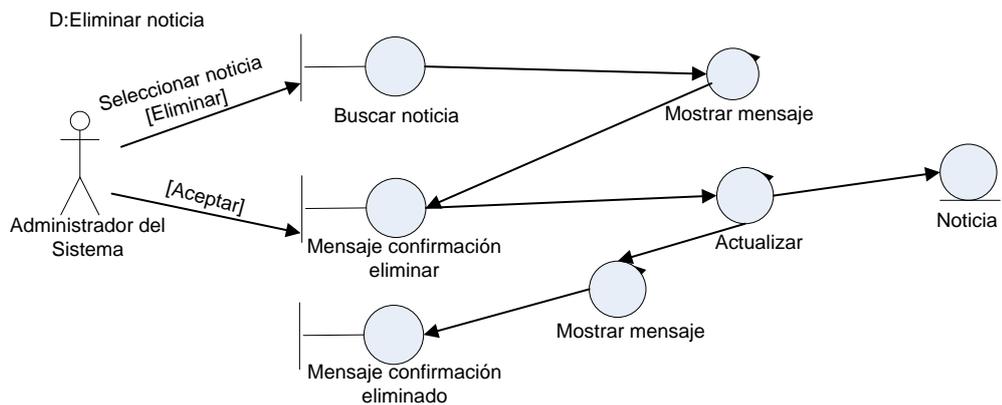


Figura 4.33 Diagrama de robustez del curso alternativo D: Eliminar noticia del caso de uso Administrar noticia

#### 4.8.4.3 Diagramas de Robustez del Caso de Uso Administrar Foro de Discusión

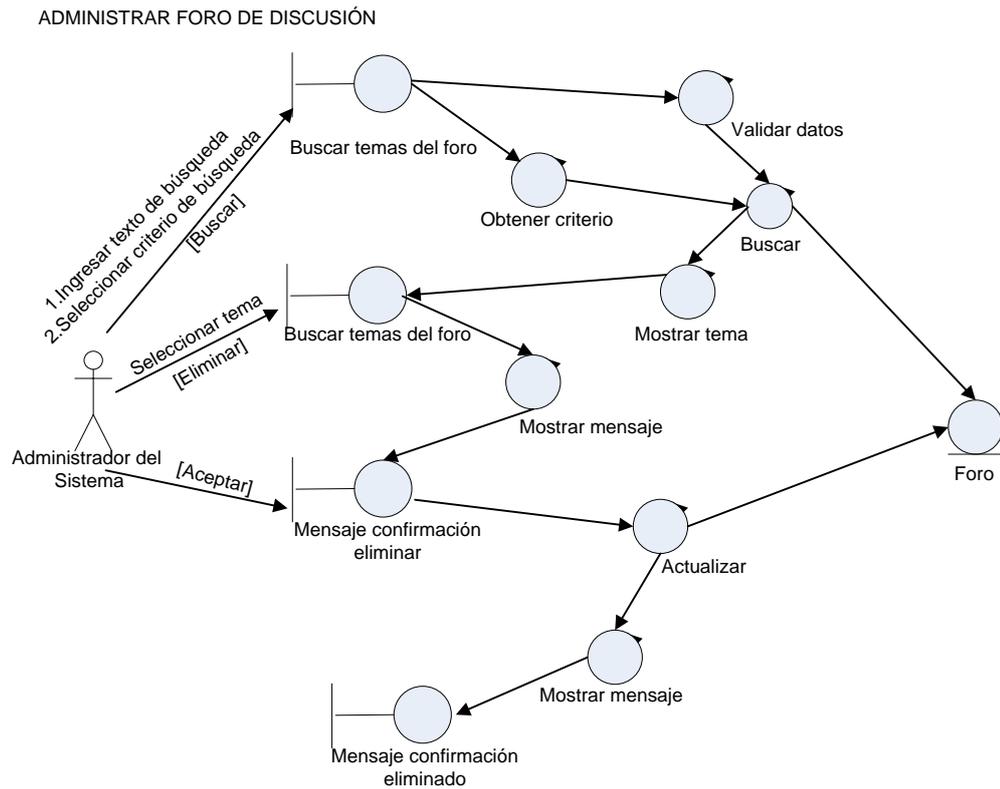
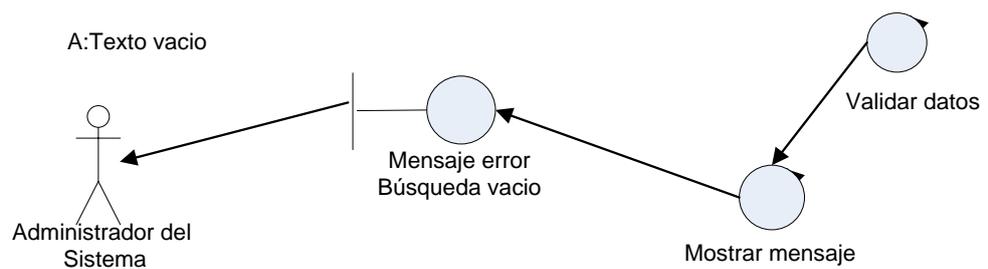
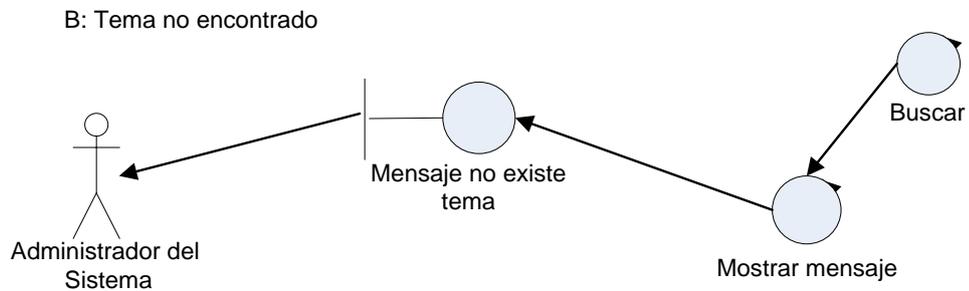


Figura 4.34 Diagrama de robustez del caso de uso Administrar foro de discusión

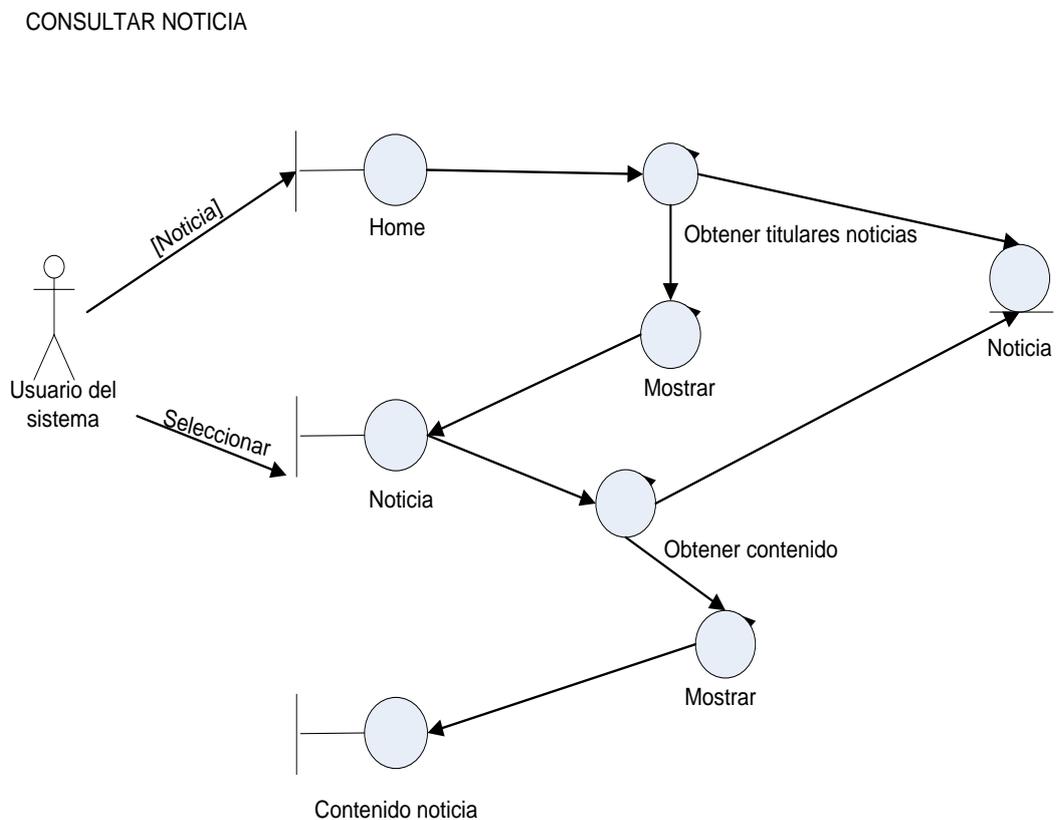


**Figura 4.35 Diagrama de robustez del curso alternativo A: Texto vacío del caso de uso Administrar foro de discusión**



**Figura 4.36 Diagrama de robustez del curso alternativo B: Tema no encontrado del caso de uso Administrar foro de discusión**

#### 4.8.4.4 Diagramas de Robustez del Caso de Uso Consultar Noticia



**Figura 4.37 Diagrama de robustez del caso de uso Consultar noticia**



#### 4.8.4.5 Diagramas de Robustez del Caso de Uso Participar en Foro de Discusión

PARTICIPAR EN FORO DE DISCUSIÓN

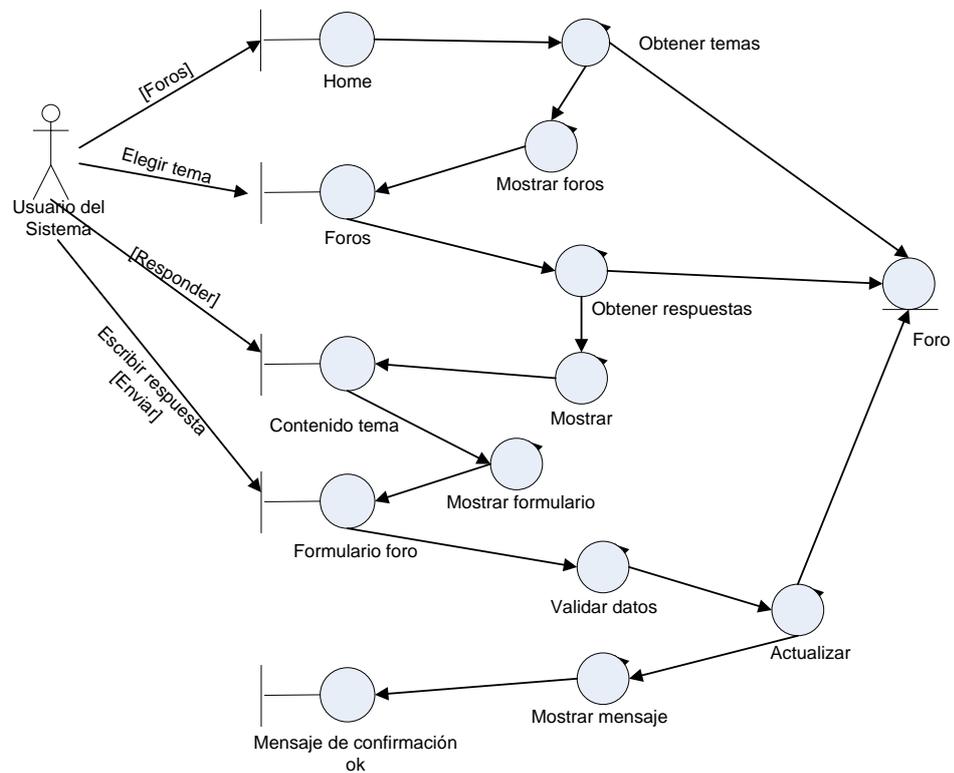
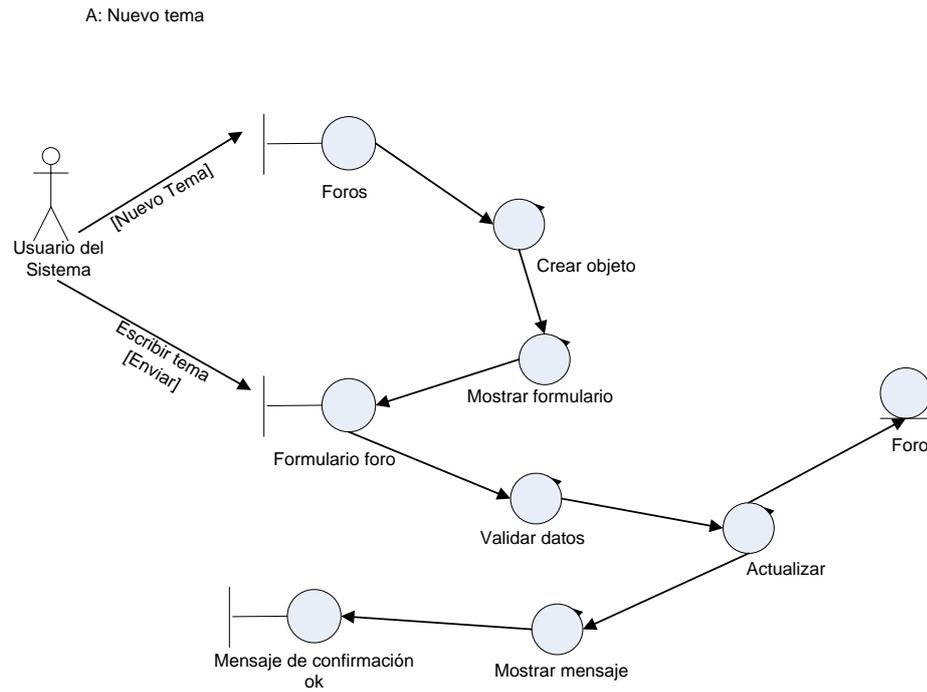


Figura 4.38 Diagrama de robustez del caso de uso Participar en foro de discusión



**Figura 4.39 Diagrama de robustez del curso alterno A: Nuevo tema del caso de uso Participar en foro de discusión**

#### 4.8.4.6 Diagramas de Robustez del Caso de Uso Participar en Videoconferencia

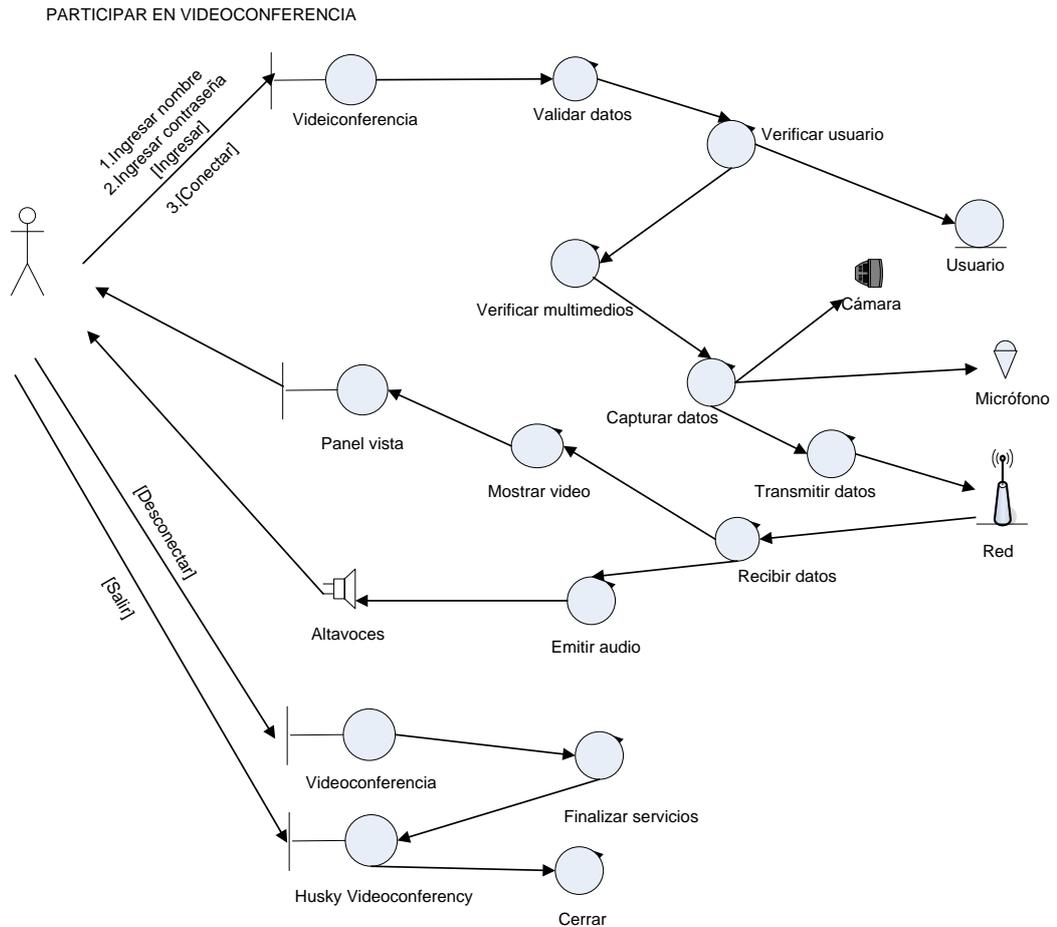


Figura 4.40 Diagrama de robustez del caso de uso Participar en Videoconferencia

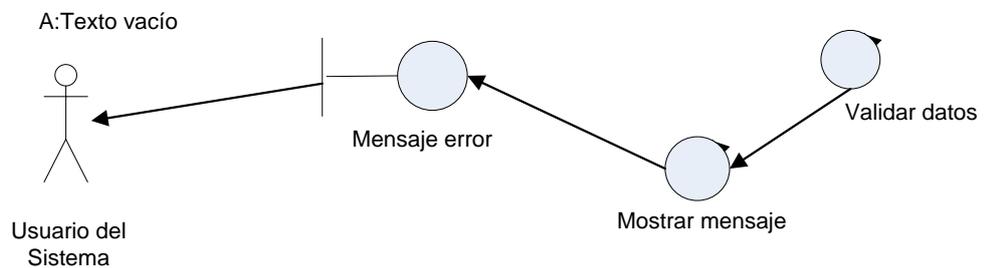
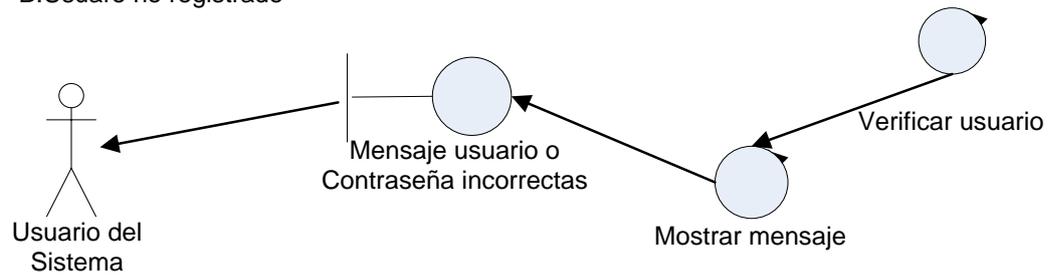


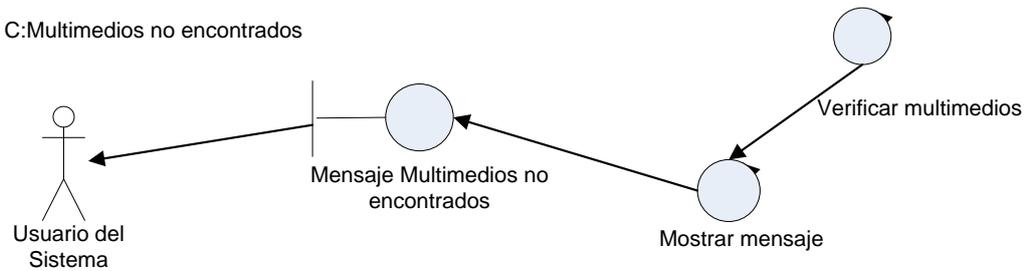
Figura 4.41 Diagrama de robustez del curso alternativo A: Texto vacío del caso de uso Participar en videoconferencia

B: Usuario no registrado



**Figura 4.42 Diagrama de robustez del curso alternativo B: Usuario no registrado del caso de uso Participar en videoconferencia**

C: Multimedia no encontrados



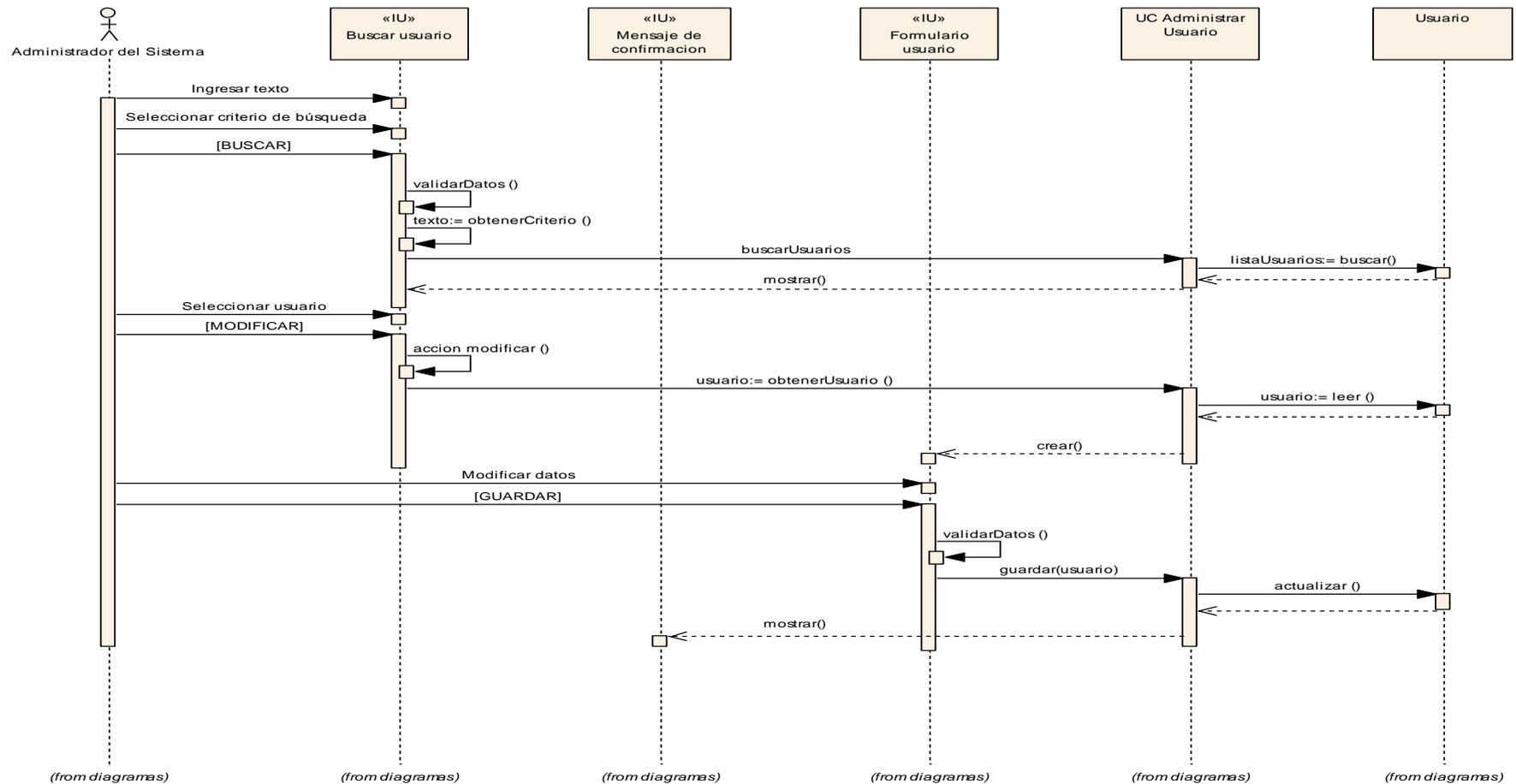
**Figura 4.43 Diagrama de robustez del curso alternativo C: Multimedia no encontrados del caso de uso Participar en videoconferencia**



## 4.8.5 DIAGRAMAS DE SECUENCIA

### 4.8.5.1 Diagramas de Secuencia del Caso de Uso Administrar Usuario

UC administrar usuario

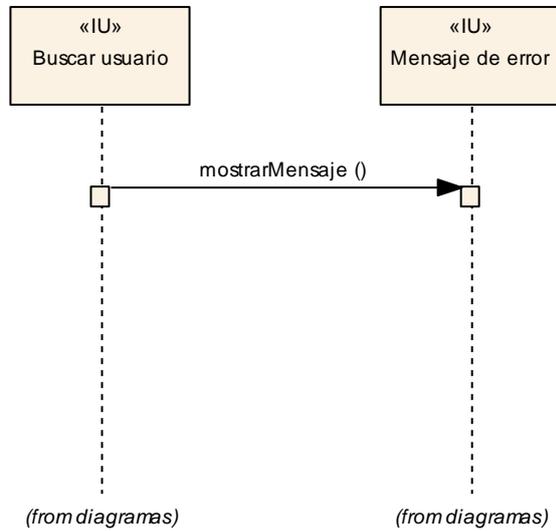




**Figura 4.44 Diagrama de secuencia del caso de uso Administrar usuario**

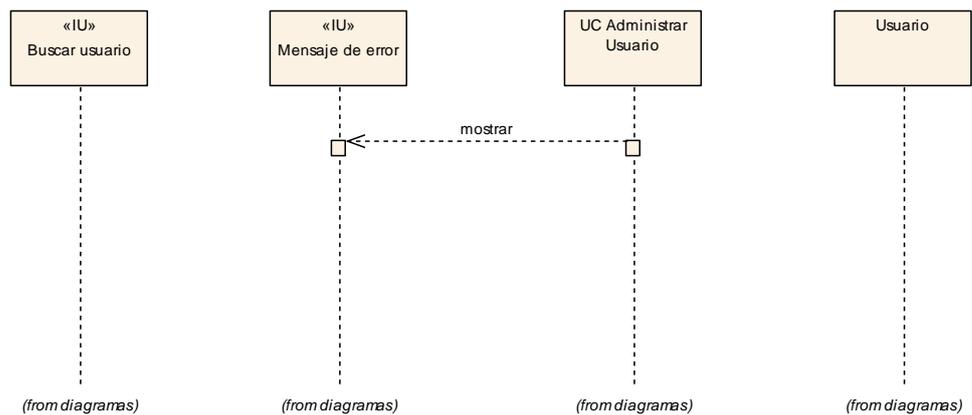


A: Texto vacío



**Figura 4.45 Diagrama de secuencia del curso alternativo A: Texto vacío del caso de uso Administrar usuario**

B: Usuario no encontrado



**Figura 4.46 Diagrama de secuencia del curso alternativo B: Usuario no encontrado del caso de uso Administrar usuario**



C: Nuevo usuario

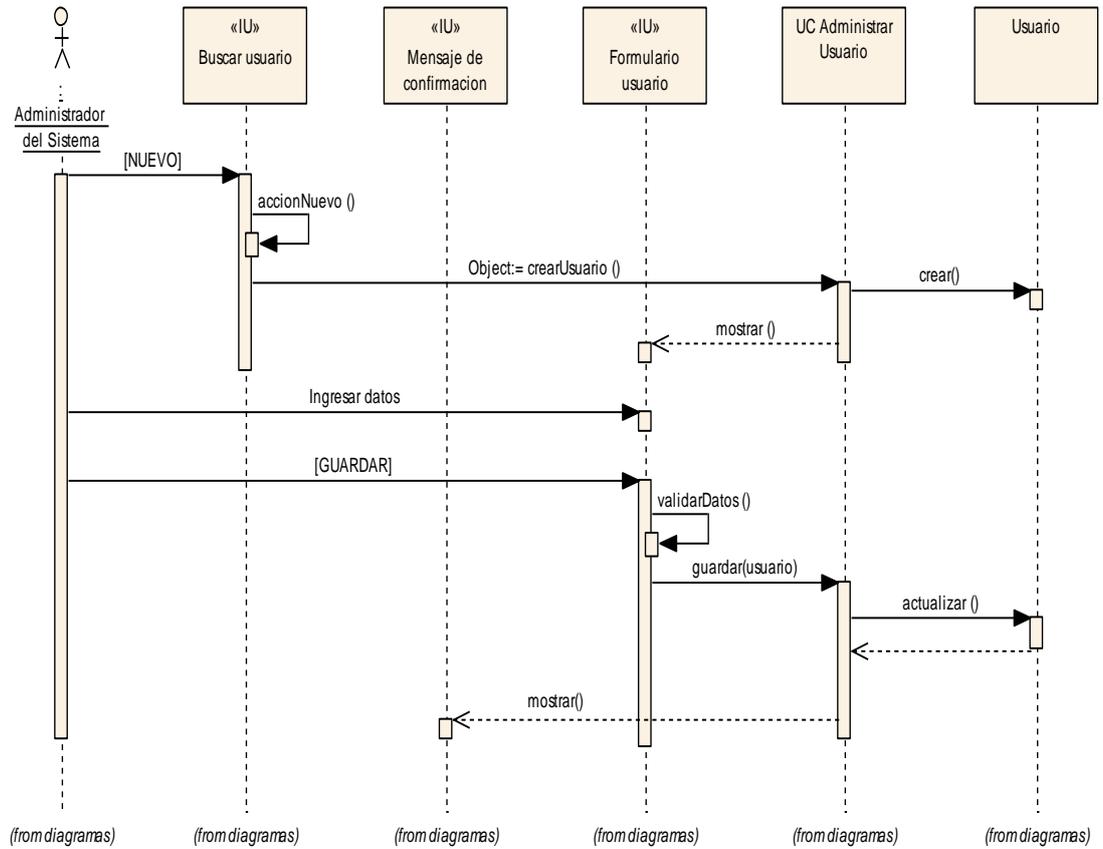
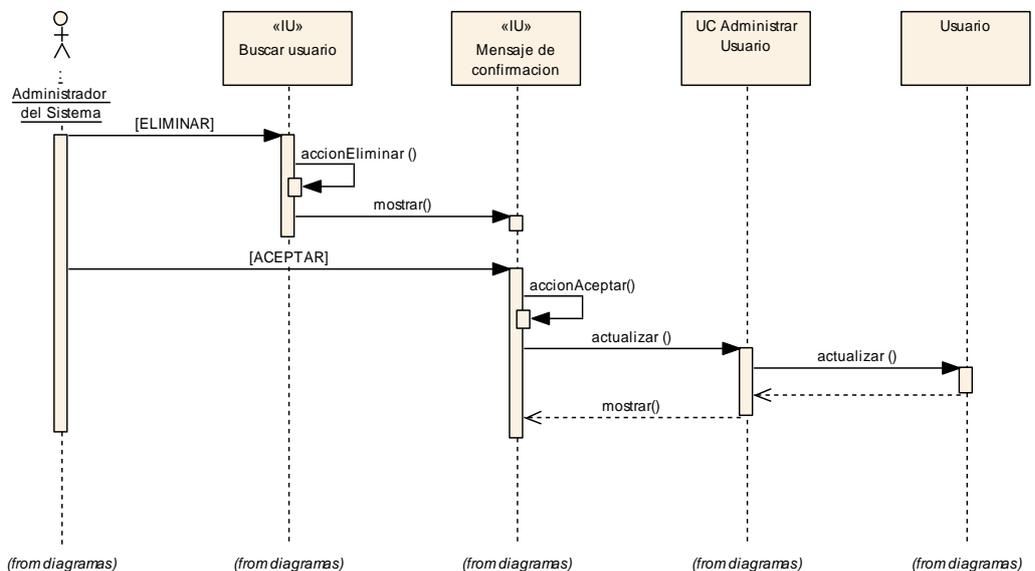


Figura 4.47 Diagrama de secuencia del curso alternativo C: Nuevo usuario del caso de uso Administrar usuario

D: Eliminar usuario





**Figura 4.48 Diagrama de secuencia del curso alterno D: Eliminar usuario del caso de uso Administrar usuario**



### 4.8.5.2 Caso de uso Administrar noticia

UC administrar noticia

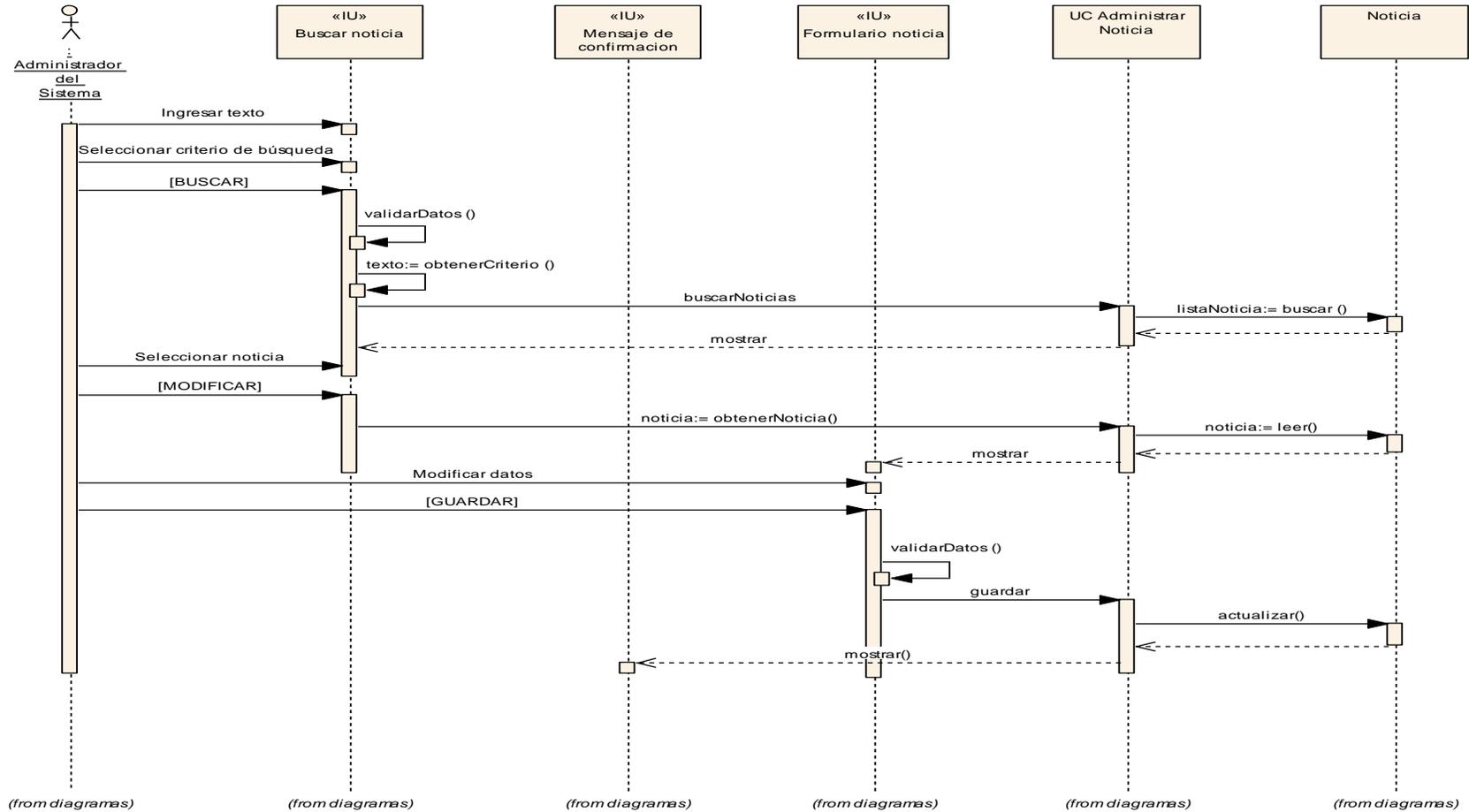


Figura 4.49 Diagrama de secuencia del caso de uso Administrar noticia



A: Texto vacío

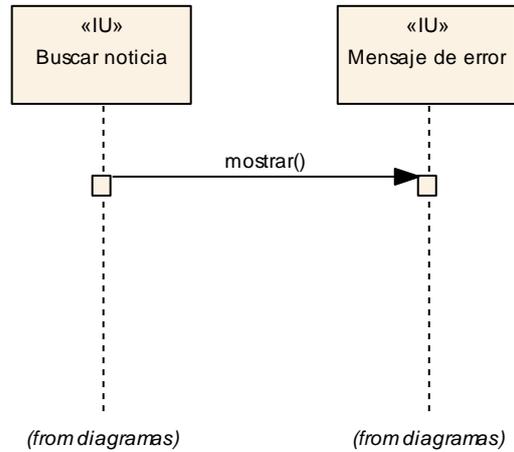


Figura 4.50 Diagrama de secuencia del curso alternativo A: Texto vacío del caso de uso Administrar noticia

B: Noticia no encontrada

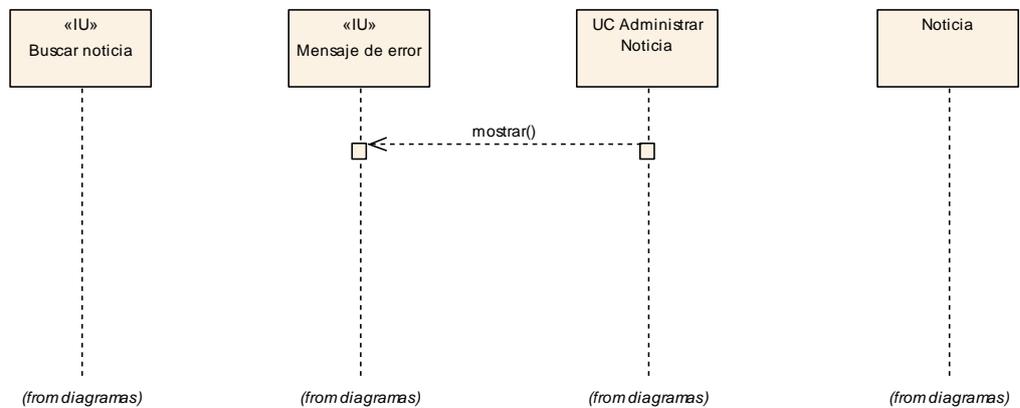


Figura 4.51 Diagrama de secuencia del curso alternativo B: Noticia no encontrada del caso de uso Administrar noticia



C: Nueva noticia

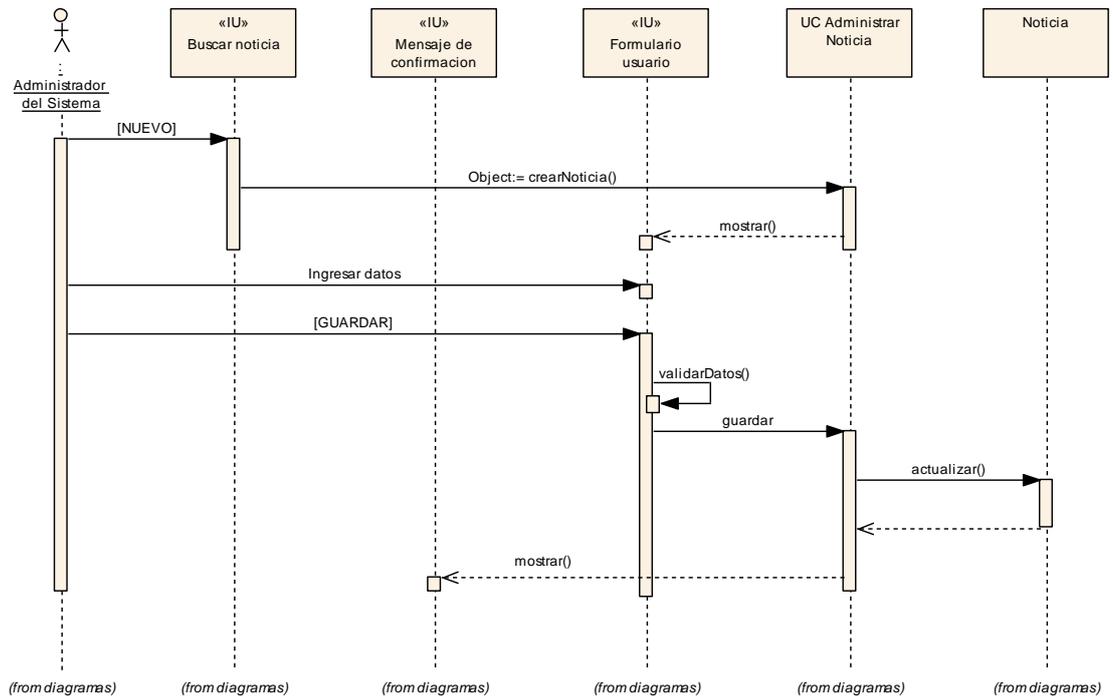
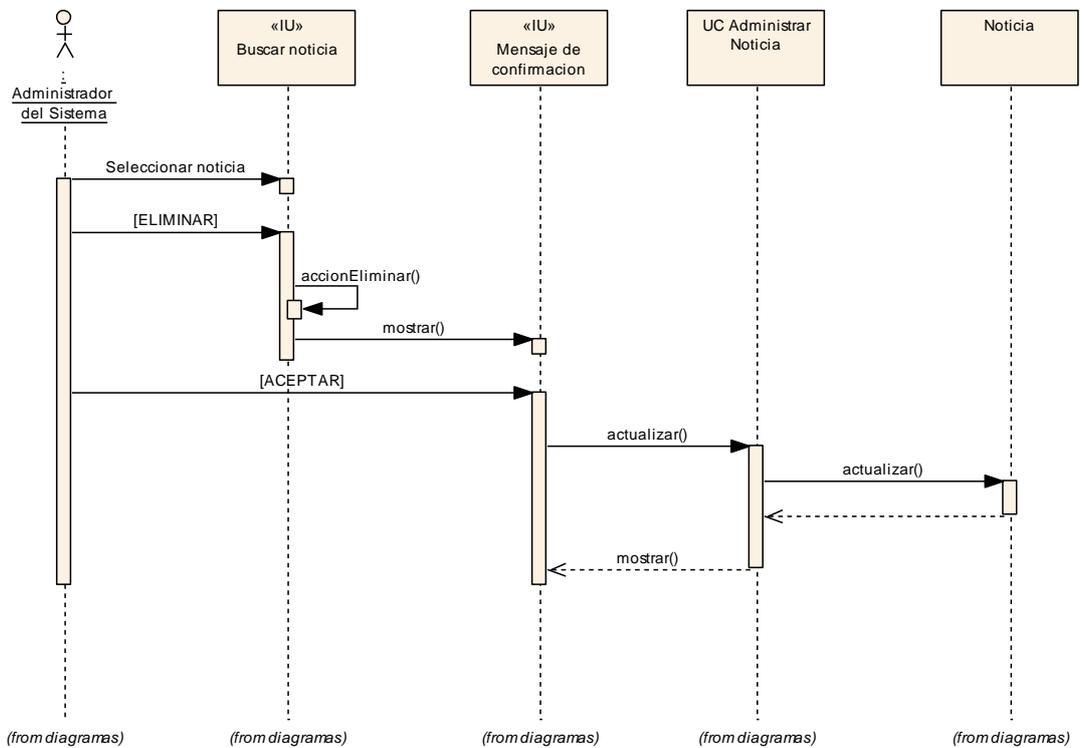


Figura 4.52 Diagrama de secuencia del curso alternativo C: Nueva noticia del caso de uso Administrar noticia

D: Eliminar noticia



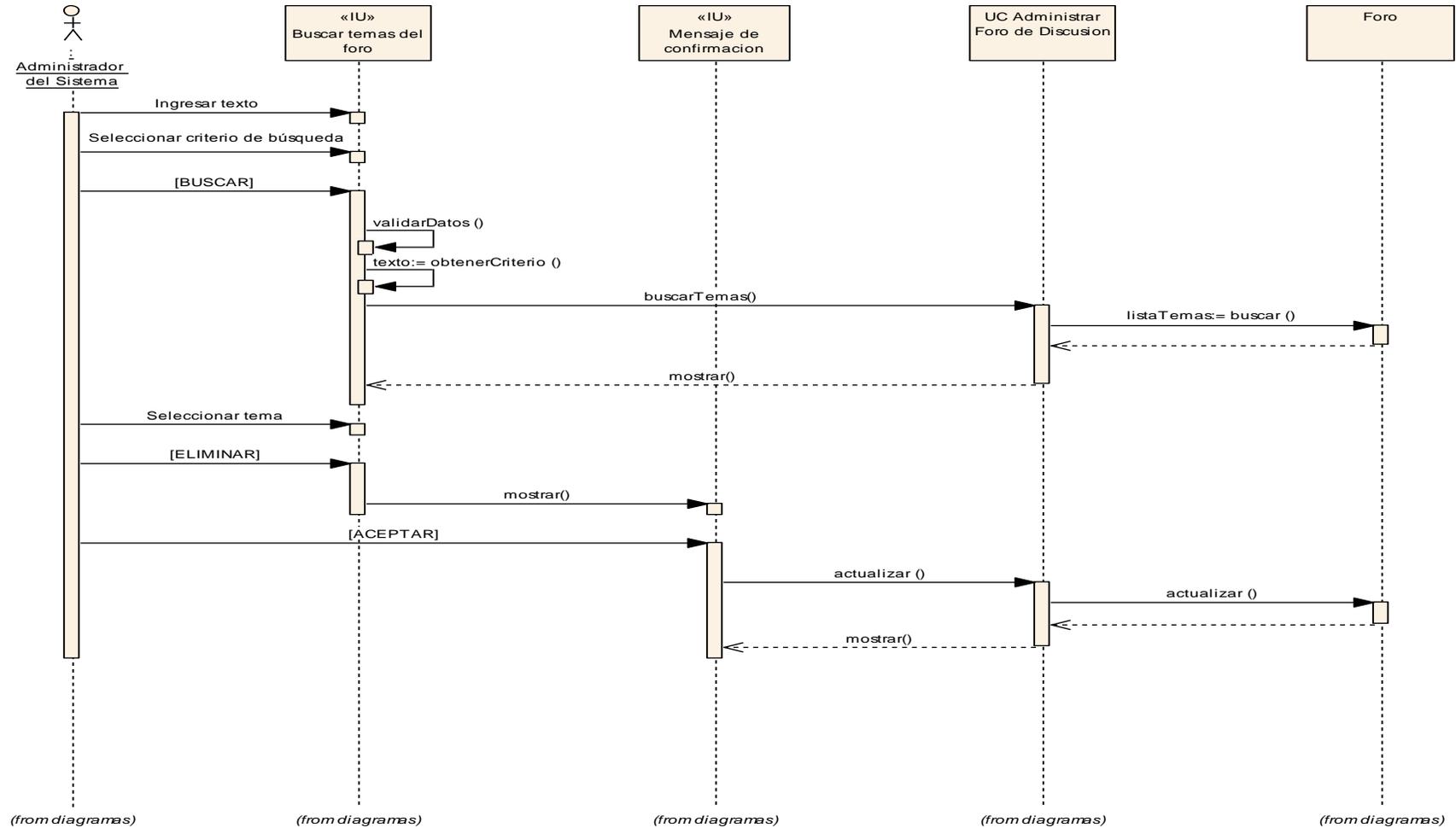


**Figura 4.53 Diagrama de secuencia del curso alternativo D: Eliminar noticia del caso de uso  
Administrar noticia**



### 4.8.5.3 Caso de uso Administrar foro de discusión

UC administrar foro de discusion





**Figura 4.54 Diagrama de secuencia del caso de uso Administrar foro de discusión**



A: Texto vacío

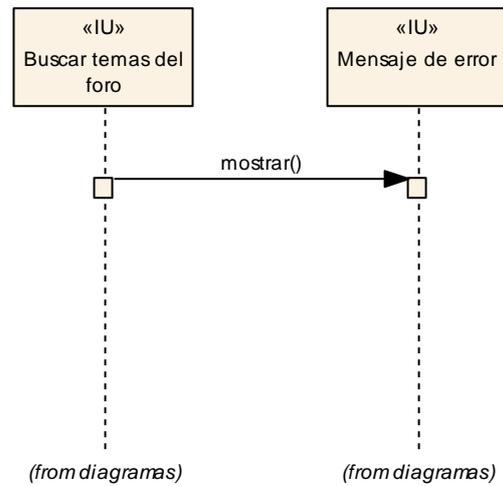


Figura 4.55 Diagrama de secuencia del curso alternativo A: Texto vacío del caso de uso Administrar foro de discusión

B: Tema no encontrado

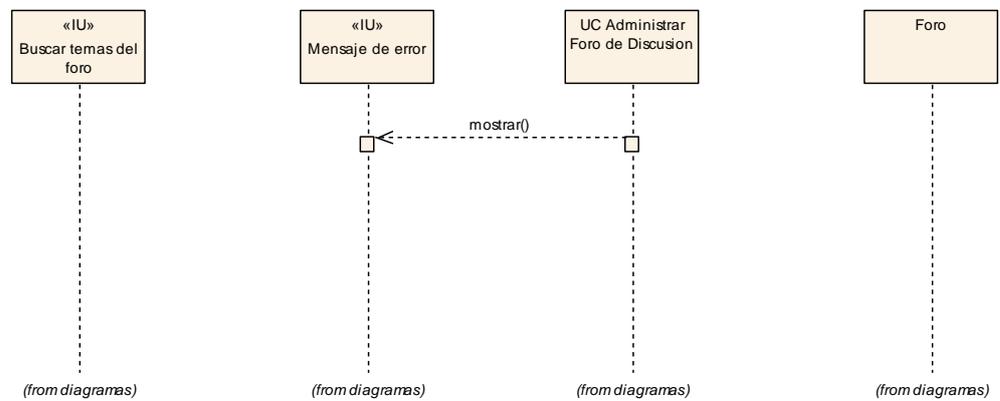
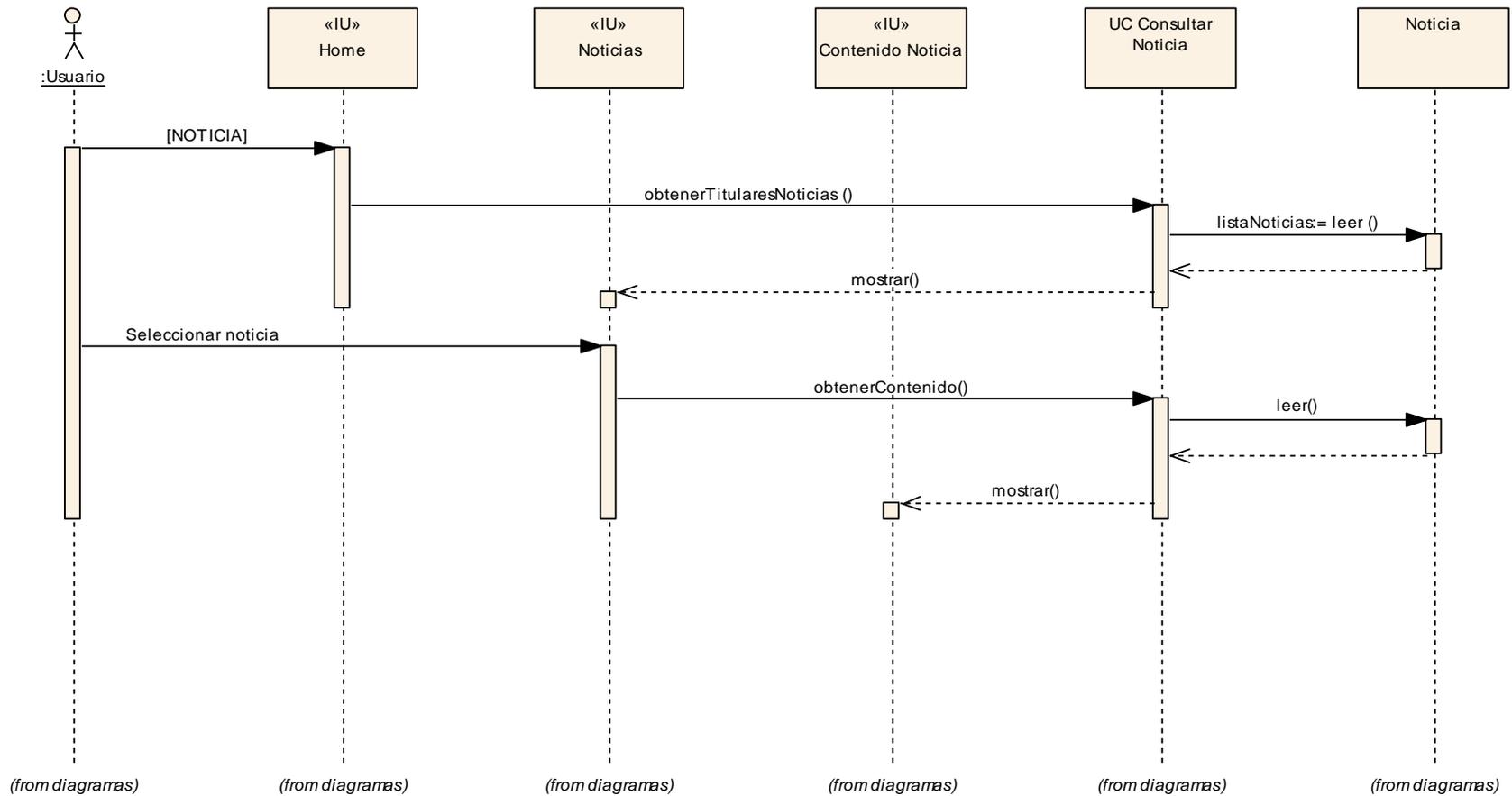


Figura 4.56 Diagrama de secuencia del curso alternativo B: Texto no encontrado del caso de uso Administrar foro de discusión



#### 4.8.5.4 Caso de uso Consultar noticia

UC consultar noticia





**Figura 4.57 Diagrama de secuencia del caso de uso Consultar noticia**

#### **4.8.5.5 Caso de uso Participar en foro de discusión**



UC participar en foro de discusión

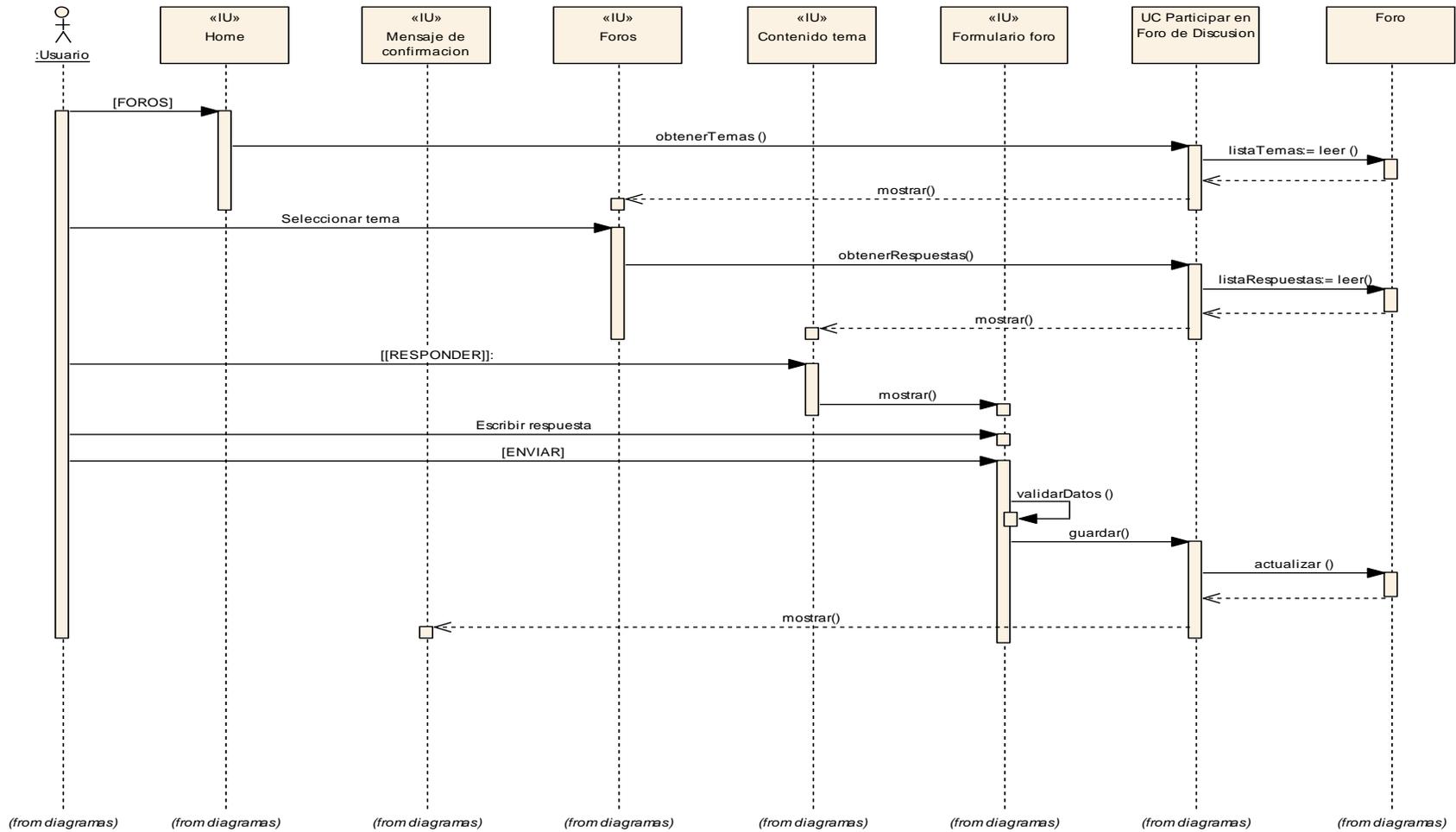


Figura 4.58 Diagrama de secuencia del caso de uso Participar en foro de discusión



A: Nuevo tema

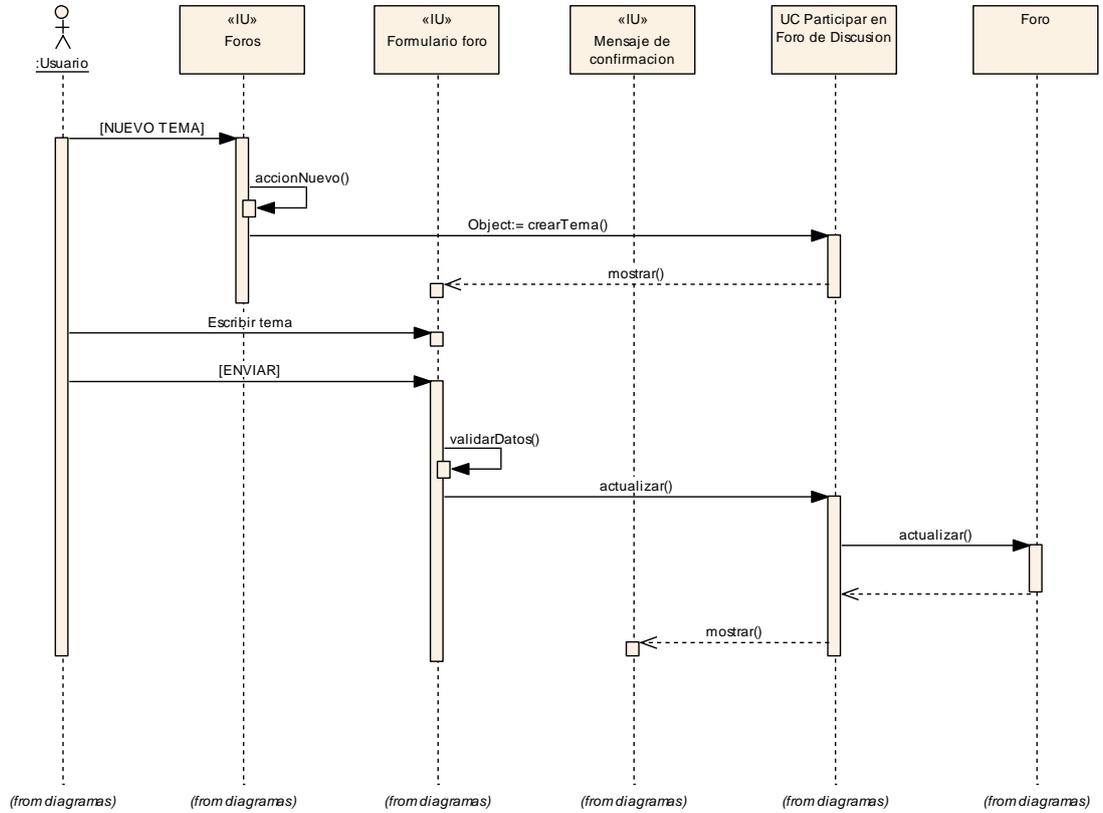


Figura 4.59 Diagrama de secuencia del curso alternativo A: Nuevo tema del caso de uso Participar en foro de discusión



### 4.8.5.6 Caso de uso Participar en Videoconferencia

UC participar en videoconferencia

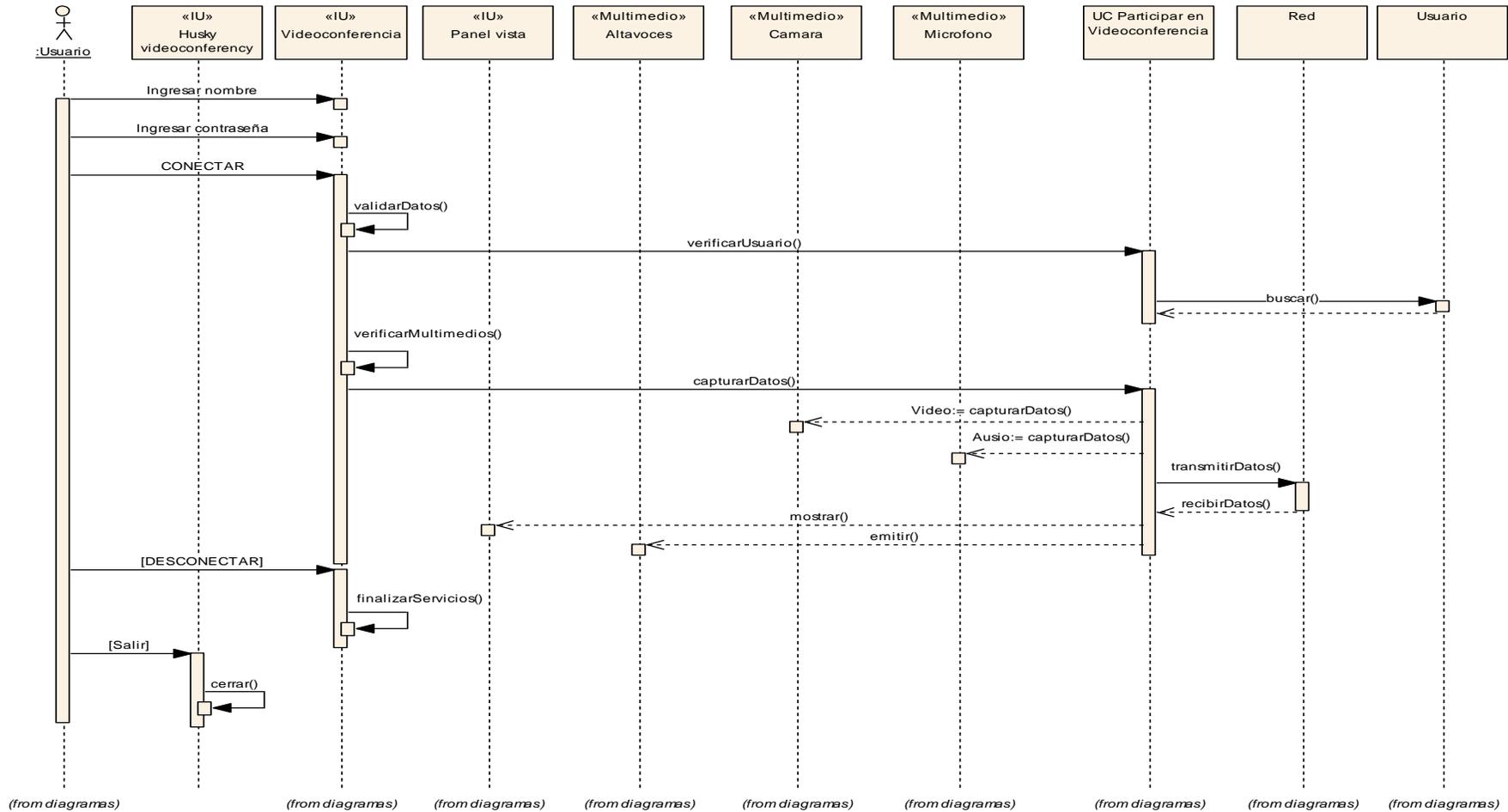


Figura 4.60 Diagrama de secuencia del caso de uso Participar en videoconferencia



A: Texto vacío

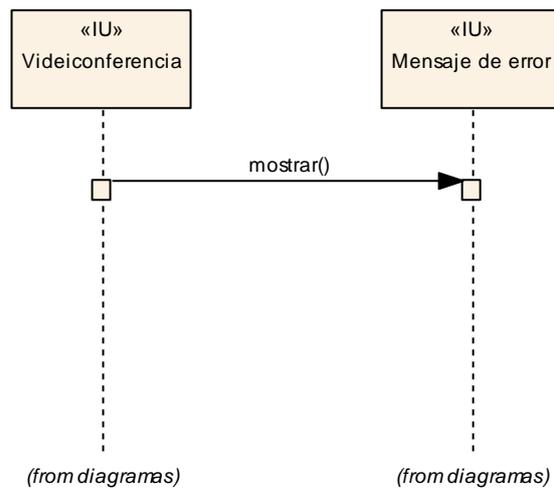


Figura 4.61 Diagrama de secuencia del curso alternativo A: Texto vacío del caso de uso Participar en videoconferencia

B: Usuario no registrado

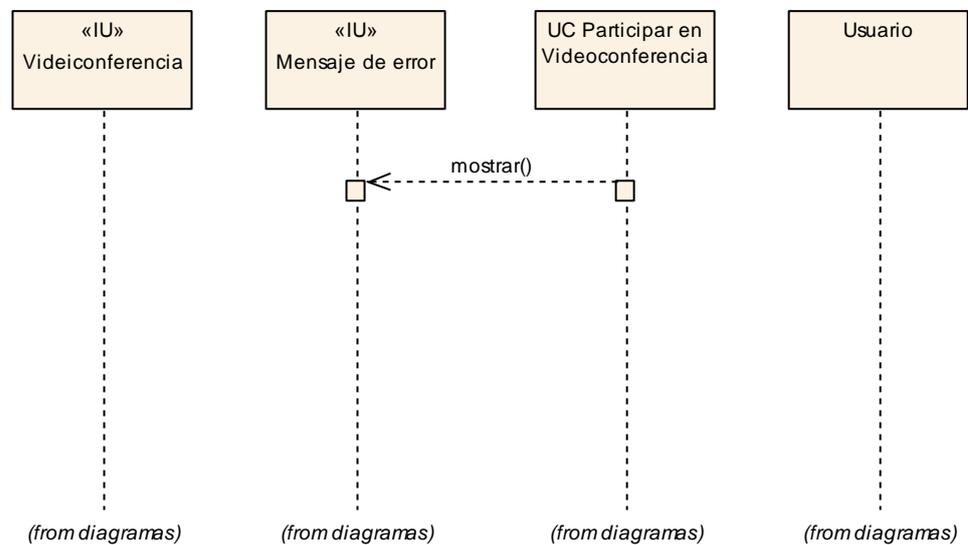


Figura 4.62 Diagrama de secuencia del curso alternativo B: Usuario no registrado del caso de uso Participar en videoconferencia



C: Multimedia no encontrados

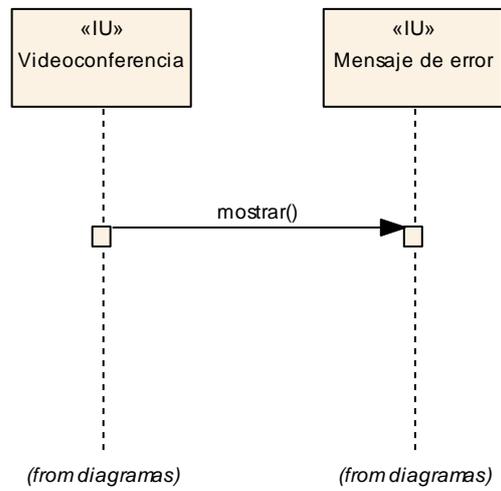


Figura 4.63 Diagrama de secuencia del curso alterno C: Multimedia no encontrados del caso de uso Participar en videoconferencia

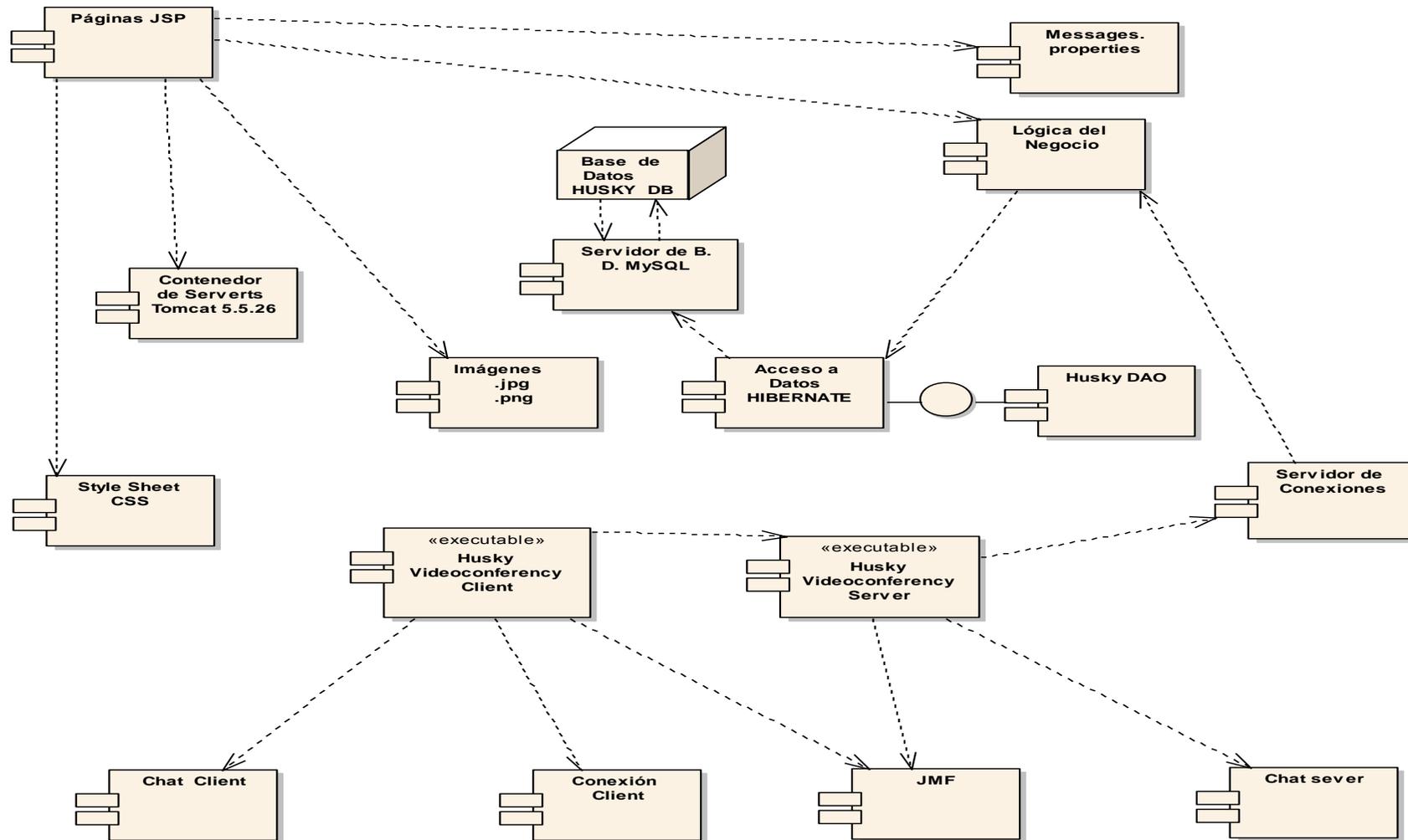




**Figura 4.64 Diagramas de Navegación**



#### 4.6 DIAGRAMA DE COMPONENTES





**Figura 4.65 Diagrama de Componentes**



## 4.7 DIAGRAMA DE DESPLIEGUE

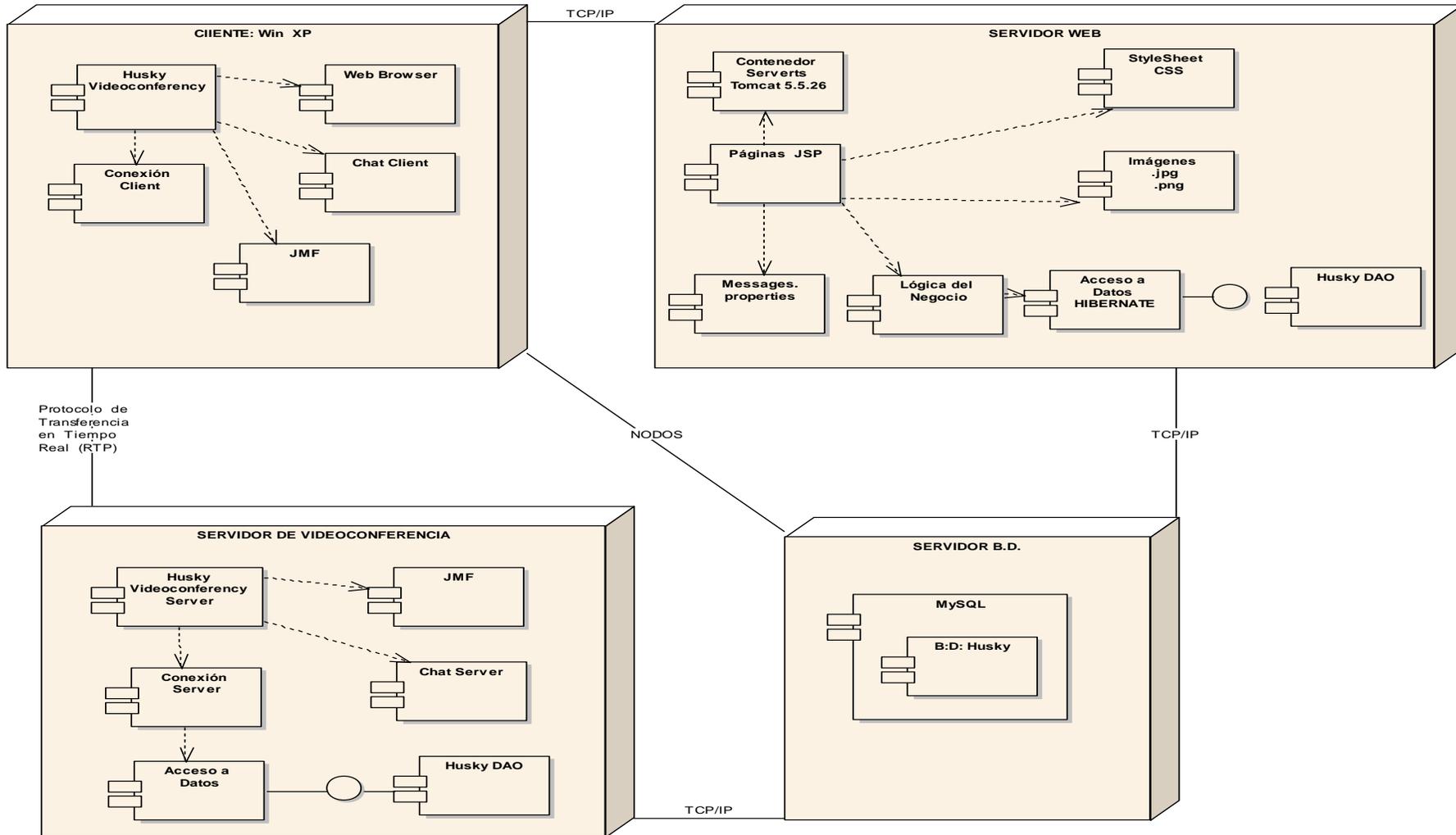
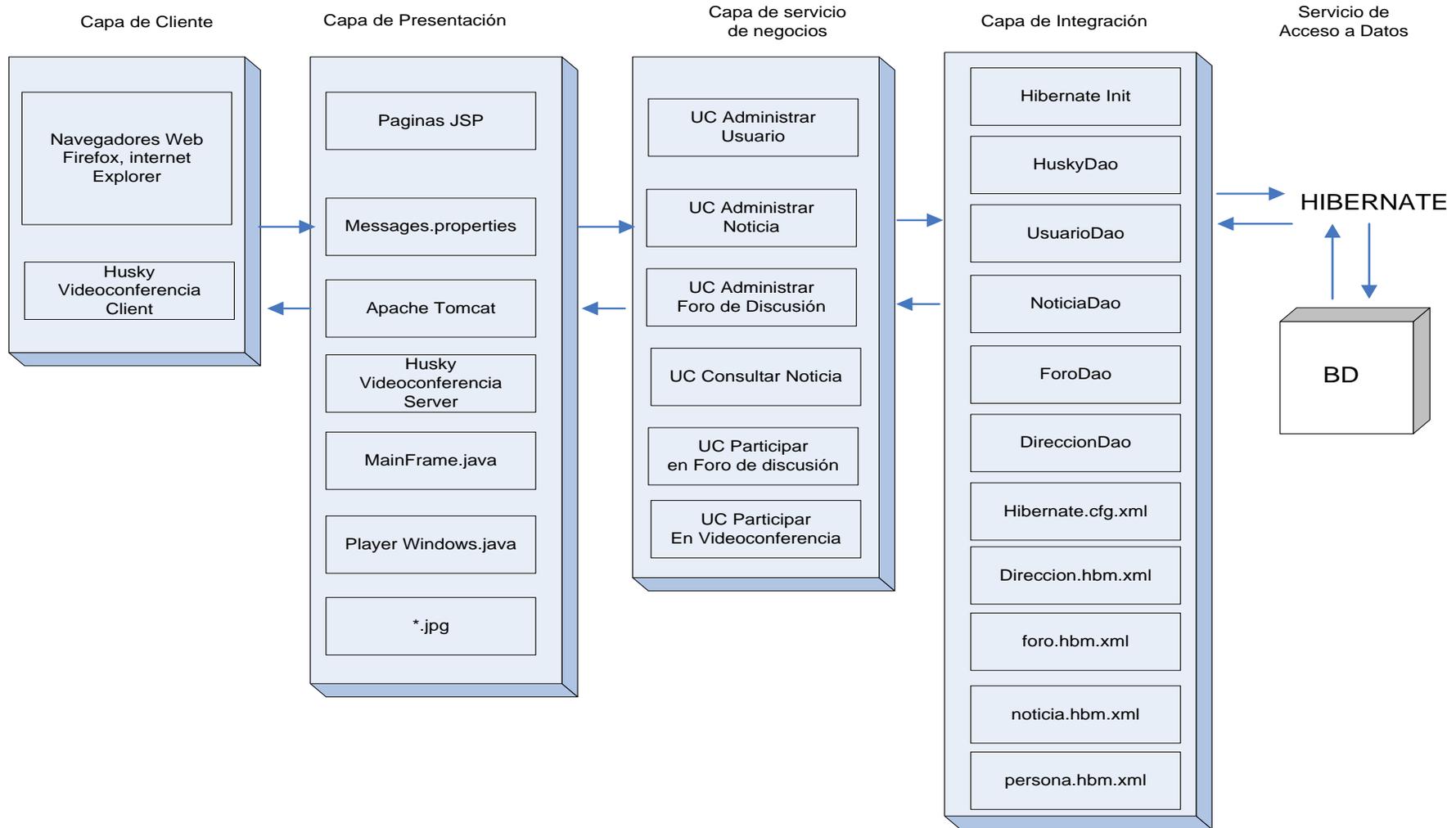




Figura 4.66 Diagrama de Despliegue

#### 4.8 DIAGRAMA DE ARQUITECTURA





**Figura 4.67 Diagrama de Navegación**



#### 4.9 PRUEBAS DE LA APLICACIÓN

Como parte del proceso de pruebas y validación, se ha realizado encuestas dirigidas a los docentes, estudiantes y a la Ing. Ketty Palacios Coordinadora de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, quienes nos colaboraron amablemente en las pruebas realizadas de la aplicación y en emitir sus criterios en las encuestas que se les pidió contestar.

Con el fin de recabar información sobre el servicio de videoconferencia, así como también depurar algunos errores encontrados en esta etapa, se realizó la encuesta que se adjunta a continuación.



Resultados de la encuesta aplicada a los usuarios del Sistema de Husky Videoconferency.

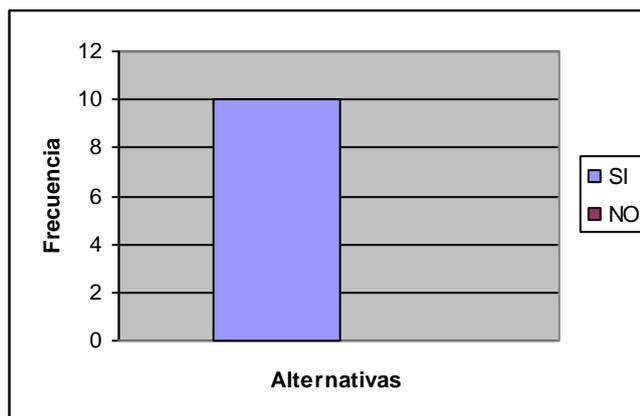
11. ¿Desde su punto de vista cree que la aplicación Husky Videoconferency facilitará la comunicación entre docentes y estudiantes del AEIRNNR?

Tabla 1. Resultados Pregunta 1

Alternativas	Frecuencia	%
SI	10	100
NO	0	0
TOTAL	10	100

Fuente: Encuesta a Usuarios

Elaboración: Los Autores



**Figura 1. Resulta Pregunta 1**

**Fuente: Encuesta a Usuarios**

**Elaboración: Los Autores**

De lo anterior concluimos que el 100% de los encuestados opinan que la aplicación Husky Videoconferency si facilitará la comunicación entre docentes y estudiantes, por las siguientes razones:

- Permite a los docentes dictar sus clases a los estudiantes a distancia
- El AEIRNNR no posee una herramienta de este tipo
- Facilita la comunicación entre docentes y estudiantes
- Se puede llegar a un gran número de personas

**11. Considera UD que la interfaz de usuario del sistema Husky Videoconferency es:**

**Tabla 2. Resultados Pregunta 2**

Alternativas	Frecuencia	%
--------------	------------	---



Amigable al usuario	10	100
Medianamente amigable al usuario	0	0
Nada amigable al usuario	0	0
TOTAL	10	100

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

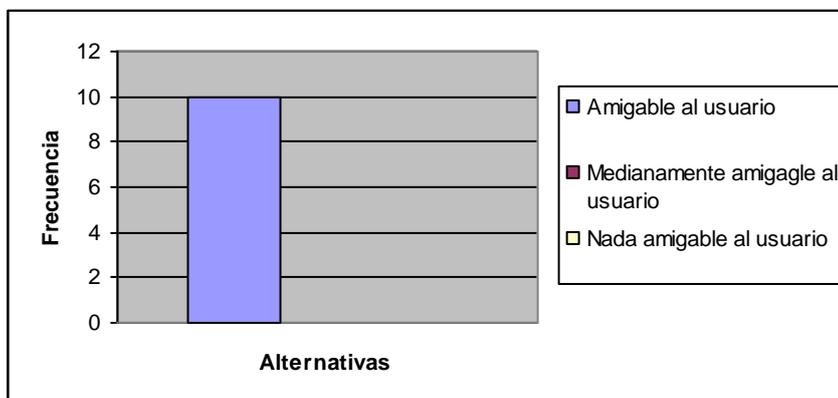


Figura 2. Resulta Pregunta 2

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

El 100% de los encuestados consideran que la aplicación Husky Videoconferency cuenta con una interfaz de usuario amigable y fácil de utilizar.

### 11. La calidad en la presentación de video le parece a UD que es:

Tabla 3. Resultados Pregunta 3



Alternativas	Frecuencia	%
De buena calidad	7	70
Medianamente buena calidad	3	30
De mala calidad	0	0
<b>TOTAL</b>	<b>10</b>	<b>100</b>

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

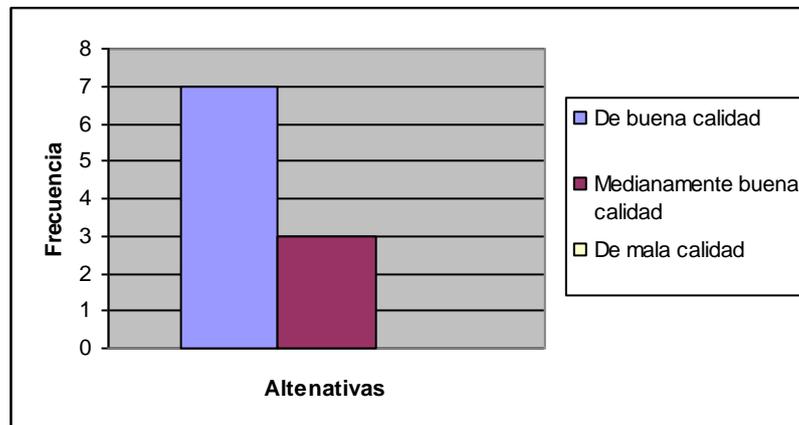


Figura 3. Resulta Pregunta 3

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

El 70% de los encuestados opinan que la presentación de video de la aplicación Husky Videoconferency es de buena calidad, mientras que el 30% consideran que la presentación de video es de medianamente buena calidad.

#### 11. La calidad en la nitidez de audio opina UD que es:

Tabla 4. Resultados Pregunta 4

Alternativas	Frecuencia	%
--------------	------------	---



De buena calidad	5	50
Medianamente buena calidad	5	50
De mala calidad	0	0
TOTAL	10	100

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

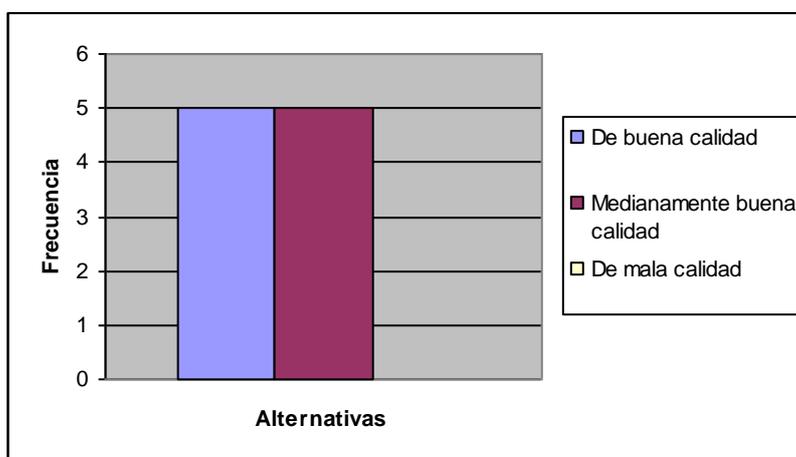


Figura 4. Resulta Pregunta 4

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

En base a los resultados el 50% de encuestados consideran que la nitidez de audio de la aplicación Husky Videoconferency es de buena calidad, mientras que el otro 50% opinan que la nitidez de audio es de medianamente buena calidad.

11. ¿Cree UD que la función de comunicación mediante texto ayuda a hacer la comunicación más eficiente?

Tabla 5. Resultados Pregunta 5



Alternativas	Frecuencia	%
SI	10	100
NO	0	0
TOTAL	10	100

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

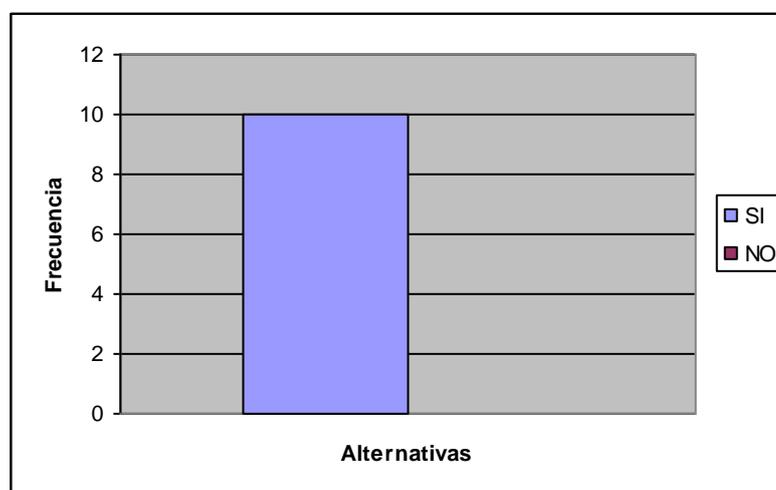


Figura 5. Resulta Pregunta 5

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

El 100% de encuestados consideran que la comunicación mediante texto ayuda a hacer la comunicación más eficiente por las siguientes razones:

- Permite reforzar la comunicación en caso de que existan problemas con el audio
- Permite realizar preguntas más concretas, además hace la conferencia amena
- Se puede detallar las ideas que se desea expresar



11. En cuanto a la velocidad de transmisión de los datos de audio y video considera UD que:

Tabla 6. Resultados Pregunta 6

Alternativas	Frecuencia	%
Es aceptable	10	100
Medianamente aceptable	0	0
Nada aceptable	0	0
TOTAL	10	100

Fuente: Encuesta a Usuarios

Elaboración: Los Autores

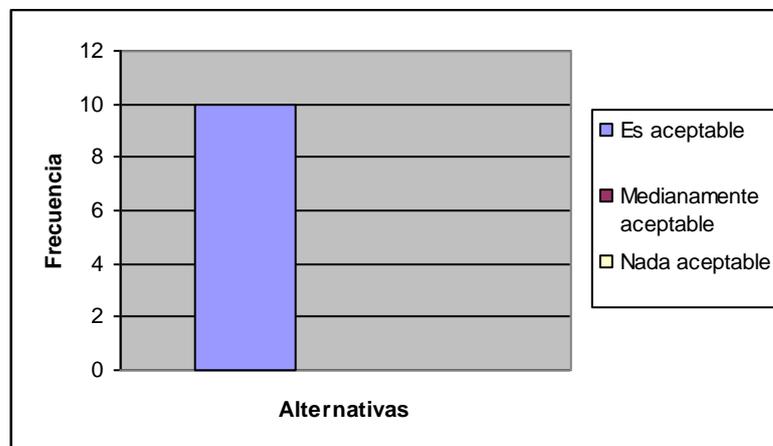


Figura 6. Resulta Pregunta 6

Fuente: Encuesta a Usuarios

Elaboración: Los Autores



De lo anterior, el 100% de los encuestados consideran que la velocidad en la transmisión de los datos es aceptable, por cuanto existe un mínimo retardo en la comunicación.

Como resultado de las pruebas realizadas, se obtuvo lo siguiente.

Los encuestados consideran:

- Que Husky Videoconferency puede facilitar la comunicación entre múltiples usuarios de diferentes disciplinas tanto de nuestra universidad como usuarios ajenos a ella.
- Que la interfaz de usuario de la aplicación Husky Videoconferency es intuitiva y fácil de usar.
- La calidad de video depende del dispositivo de captura empleado, este caso se obtuvo buena calidad ya que se utilizaron cámaras de buena resolución (640x480 píxeles).
- La nitidez de audio es aceptable, ya que se equipara al de un teléfono convencional.



- Que los factores que degradan el servicio de videoconferencia pueden ser:  
El ancho de banda (red de comunicación), debido a la pérdida de paquetes  
Una sala inapropiada para videoconferencia.  
Equipos obsoletos (dispositivos de presentación de audio y video)
- Que la comunicación por texto sirve como un respaldo a la videoconferencia en caso de que no funcione correctamente alguno de los dispositivos de captura o reproducción de audio de los participantes en la videoconferencia.



## 5. PLAN DE VALIDACIÓN

Las pruebas se realizaron empleando como base los requerimientos obtenidos en la etapa de análisis, en el siguiente cuadro se muestran los resultados obtenidos para cada uno de ellos.

PLAN DE VALIDACIÓN	
REQUERIMIENTOS	RESULTADOS
<b>Facilidad de uso</b>	Se efectuaron pruebas con diferentes usuarios en base a cuyas respuestas se concluye que la aplicación de escritorio Husky Videoconferency, así como la aplicación Husky Web Videoconferency cuentan con interfaces gráficas intuitivas y de fácil uso para los usuarios.
<b>Administración del sistema</b>	La administración (creación, modificación, eliminación) de los usuarios del sistema, de los contenidos de noticias y el manejo de foros de discusión se puede realizar de forma fácil mediante la aplicación web.
<b>Creación de foros</b>	La aplicación web permite la creación de foros de discusión y la participación en los mismos mediante la creación de respuestas a diferentes temas.
<b>Presentación de video</b>	La aplicación Husky Videoconferency ofrece una calidad de video aceptable empleando dispositivos de captura con un mínimo de 640x480 píxeles de resolución utilizando formato de video de 24 bits en RGB, cumple con el estándar H323 definido para videoconferencia sobre redes IP, el mismo que define el estándar H263 para transmisión de video sobre una red IP.



<b>Nitidez de audio</b>	La aplicación permite escuchar el audio recibido desde la red de forma clara y entendible, ofrece una calidad de audio comparable con el de un teléfono convencional, cumple con el estándar H323 para videoconferencia sobre redes IP, el mismo que define el estándar G723 para transmisión de voz sobre IP.
<b>Comunicación mediante texto</b>	Husky Videoconferency ofrece la posibilidad de comunicación mediante texto, que permite mantener una comunicación más eficiente, además sirve como respaldo a la videoconferencia en caso de un mal funcionamiento en el dispositivo de audio de alguno de los participantes en la misma
<b>Multiplataforma</b>	<p>Para la validación del sistema de videoconferencia se emplearon algunos sistemas operativos para comprobar el funcionamiento de la aplicación en cada uno de ellos</p> <p><b>Sistema Operativo:</b> Windows XP Profesional service pack 2 y service pack 3</p> <p><b>Dispositivos de video</b></p> <ul style="list-style-type: none"><li>• Cámara chicony usb2.0</li><li>• USB PC Cámara (SN9C101)</li><li>• Logitech quickCam Chat</li><li>• Genius Look III V 1.0</li></ul>



### Resultados

La aplicación funcionó adecuadamente en cuanto a los dispositivos de captura de audio y video, además de la recepción y presentación de los datos, sin embargo se comprobó que el soporte IPv6 de java en este sistema no funciona de acuerdo a lo esperado, pues aunque los sockets de comunicación para identificación de usuarios y la obtención de puertos de comunicación con el servidor funcionaron de forma adecuada, la transmisión y recepción de datos utilizando el protocolo mencionado no fueron satisfactorias, por tanto se concluye que en XP solo se puede utilizar la aplicación usando internet e internet 2 pero con el protocolo de internet versión 4

**Sistema Operativo:** Windows Vista service pack 1

### Dispositivos de video

- Cámara chicony usb2.0
- Logitech quickCam Chat
- USB PC Camera

### Resultados

Las pruebas realizadas en este sistema fueron satisfactorias, ya que permite la utilización de la aplicación Husky Videoconferency utilizando el protocolo de internet versión 6 (IPv6), sin embargo para tal efecto se debe ingresar manualmente la dirección local (dirección IPv6



del cliente) así como la dirección remota (dirección IPv6 del servidor de videoconferencia). Por tanto se concluye que en Windows Vista se puede utilizar la aplicación usando internet e internet 2 con el protocolo de internet versión 6.

**Sistema Operativo:** Mandriva Linux Power Pack 2008.1 y 2009

**Dispositivos de video**

- Cámara chicony usb2.0
- Logitech quickCam Chat
- USB PC Camera

**Resultados**

Este sistema operativo presentó problemas con el dispositivo de captura de video y JMF.

Esto se debe a que JMF no detecta correctamente el dispositivo de video y presenta el siguiente mensaje de error: "Can't open video card". Por tanto Husky Videoconferency no funciona adecuadamente sobre Mandriva Linux.

**Sistema Operativo:** Fedora Linux 9 y Ubuntu Linux 8.04.1



	<p><b>Dispositivos de video</b></p> <ul style="list-style-type: none"><li>• Cámara chicony usb2.0</li><li>• Logitech quickCam Chat</li><li>• USB PC Camera</li></ul> <p><b>Resultados</b></p> <p>Estos sistemas operativos presentaron el mismo inconveniente que Mandriva Linux, JMF no reconoce el dispositivo de captura de video, por lo tanto Husky Videoconferency no se puede ejecutar adecuadamente sobre sistemas Linux debido a esta dificultad.</p>
<p><b>Protocolo IPv6</b></p>	<p>De las pruebas realizadas en los diferentes sistemas operativos se concluye que el soporte para el protocolo de internet versión 6 (IPv6) funciona solo utilizando el sistema operativo Windows Vista, para ello se deberá conocer la dirección del equipo cliente así como la dirección del servidor de videoconferencia e ingresarlas manualmente en los campos definidos para tal efecto en la aplicación Husky Videoconferency.</p>

**Tabla 5.1. Plan de validación**



## 6. EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN

Luego de finalizado el presente proyecto de investigación, presentamos la evaluación de los objetivos planteados al inicio del desarrollo de la tesis.

### Objetivo Especifico 1

Implementar reuniones virtuales, la interacción en tiempo real por medio de audio, video y conversación por texto entre diferentes usuarios del sistema.

Con el fin de dar cumplimiento a este objetivo en base a los requerimientos obtenidos se realizó la programación de dos aplicaciones con entorno de escritorio llamadas: Husky Videoconferency Server y Husky Videoconferency Client, que permiten la comunicación mediante audio, video y texto entre diferentes usuarios. Para la programación se emplearon herramientas con licencia de distribución libre como son: JDK 1.0.6\_05, NetBeans IDE 5.5.1, JMF 2.1.1e, Hibernate 3.2, MySQL server 5.0.41.

### Objetivo Especifico 2

Establecer la administración de usuarios del sistema por medio de interfaces sencillas.

Para cumplir con este objetivo creamos la aplicación con entorno web llamada: Husky Web Videoconferency, en la cual el administrador del sistema puede realizar toda la administración de los usuarios como es ingresar nuevos usuarios, editar la información de usuarios ya registrados en el sistema, dar de baja usuarios del sistema, todo mediante interfaces muy intuitivas, fáciles de utilizar y navegar. Para esta aplicación se usaron herramientas con licencia de distribución libre como son: JDK 1.0.6\_05, NetBeans IDE 5.5.1, JSF 1.2, Hibernate 3.2, MySQL server 5.0.41, Contenedor de Servlets Apache Tomcat 5.5.26. El administrador también puede realizar la gestión de noticias y foros de discusión del sistema desde esta sección.



### **Objetivo Especifico 3**

Permitir la consulta de noticias y la creación de foros de discusión.

Dentro de la aplicación Husky Web Videoconferency se ha creado una sección para consulta de noticias por parte de usuarios que ingresen al sistema, además de otra sección en la que los usuarios pueden crear temas de discusión dentro de los foros, así como participar dejando sus respuestas a los diferentes temas existentes en esta parte, con lo descrito anteriormente se da cumplimiento al presente objetivo. Para la programación de la sección de foros se utilizaron las mismas herramientas mencionadas anteriormente, en otras palabras, se realizó la implementación de un foro hecho en java por parte de nuestro equipo de trabajo.



## 7. VALORACIÓN TÉCNICO-ECONÓMICA-AMBIENTAL

En el presente proyecto de investigación se presenta la valoración técnico-económica-ambiental mediante la descripción de los recursos humanos, técnicos, materiales que intervinieron en la realización del mismo, además de la aproximación del costo real del proyecto una vez que entró en ejecución.

Tabla 7.1 Aproximación del Costo Real del Proyecto

DESCRIPCIÓN	CANTIDAD	# HORAS	V/U	V/T
<b>Recursos Humanos</b>				
Investigadores	2	450	2,50	2250,00
Director de Tesis – Docente de la Carrera de Ingeniería en Sistemas	1		0,00	0,00
<b>Recursos Técnicos</b>				
<b>Hardware</b>				
Cámara Digital PC Camera	1		15	15
Cámara Digital Logitech	1		15	15
Switch C-Net de 8 puertos	1		30	30
<b>Software</b>				
Microsoft Windows Vista Service Pack 1			0,00	0,00
JDK 1.0.6_05			0,00	0,00
NetBeans IDE 5.5.1			0,00	0,00
JMF 2.1.1e			0,00	0,00
Hibernate 3.2			0,00	0,00
JSF 1.2			0,00	0,00



MySQL Server 5.0.41			0,00	0,00
Apache Tomcat 5.5.26			0,00	0,00
Office 2007			0,00	0,00
Project 2007			0,00	0,00
Enterprise Architecture			0,00	0,00
Editor de Imágenes Gimp			0,00	0,00
<b>Recursos materiales</b>				
Resma de papel para impresión 500 hojas, 75 gramos, formato A4	3		3,70	11,10
Cartuchos de tinta para impresión	3		21,00	63,00
Caja de 10 CDs Imation	10		0,50	5,00
<b>Recursos tecnológicos</b>				
Internet		200	0,80	160
<b>Varios (5%)</b>				
			127,45	127,45
<b>Costo total del proyecto</b>				<b>2676,55</b>

## CONCLUSIONES

- La aplicación “Husky Videoconferency” permite la comunicación mediante audio, video y texto utilizando el protocolo de internet versión 6, empleando el sistema operativo Windows Vista.



- La aplicación “Husky Web Videoconferency” provee al administrador del sistema todo el control necesario sobre el manejo de usuarios, noticias, foros de discusión, con lo cual el mantenimiento del sistema se puede realizar de manera sencilla.
- La aplicación “Husky Web Videoconferency” permite la creación de foros de discusión, así como la creación de respuestas a los diferentes temas.
- La velocidad de transmisión y recepción de datos por parte de la aplicación “Husky Videoconferency” depende de la red sobre la cual sea implantada.
- La calidad de audio y video ofrecida por la aplicación “Husky Videoconferency” está determinada por el tipo de periféricos empleados (cámara, micrófono).
- Windows Vista provee soporte para usar el protocolo IPv6, sin embargo la pila de direcciones de este sistema operativo aún necesita ser consolidada para uso con aplicaciones desarrolladas en lenguaje java.
- La metodología de desarrollo de software orientado a objetos con UML, permite obtener una aplicación estandarizada, de fácil mantenimiento y que posibilita la adaptabilidad de la misma a diferentes entornos de trabajo con un mínimo de configuración.
- Las herramientas con licencia de distribución libre como son: java, jmf, MySQL, hibernate, jsf y Contenedor de Servlets Apache Tomcat utilizadas para el desarrollo del presente proyecto nos permitieron conocer las limitaciones de algunos sistemas operativos como son: Windows XP, Mandriva Linux, Ubuntu Linux, en cuanto al desarrollo de aplicaciones que emplean el protocolo de internet versión 6.

## RECOMENDACIONES



- Probar los dispositivos de captura de audio y video para asegurarse que se encuentran instalados y funcionando correctamente en el computador, para ello se puede utilizar la herramienta JMStudio de JMF.
- Utilizar un dispositivo de captura de video con una definición mínima de 640x480 píxeles, con un formato de video de 24 bits en modo RGB, además de un ambiente con buena iluminación, con la finalidad de que la imagen visualizada por el receptor sea adecuada.
- Antes de utilizar la aplicación “Husky Videoconferency” leer el manual del usuario con la finalidad de conocer y obtener un correcto funcionamiento del programa.
- Emplear preferentemente Mozilla Firefox para el uso de la aplicación web de administración del sistema, consulta de noticias y participación en foros, por cuanto el motor de renderizado de Firefox ofrece mayor compatibilidad con JSF y permite visualizar la página de acuerdo a su diseño.
- Deshabilitar el tema de escritorio Aero de Windows Vista para evitar posibles conflictos con JMF y Husky Videoconferency, además habilitar los puertos de transmisión y recepción de audio y video para la aplicación, en el firewall que se encuentre activo en el computador.
- En caso de que la Universidad Nacional de Loja considere implementar la aplicación “Husky Videoconferency”, se recomienda que utilice una sala adecuada con buena iluminación, libre de ruido y distracciones.
- Utilizar la metodología de desarrollo ICONIX con modelos UML, para el desarrollo de aplicaciones que sigan el patrón modelo, vista, controlador con lenguaje de programación java, por cuanto permite llevar un control adecuado de la aplicación en desarrollo mediante diagramas que facilitan su comprensión y la elaboración de la documentación respectiva.



## BIBLIOGRAFÍA

### LIBROS

- ✓ ALARCON, Raúl; Diseño Orientado a Objetos con UML
- ✓ OLSSON, Tommy; O'BRIEN, Paúl; The Ultimate CSS Reference; Primera Edición; Febrero 2008.
- ✓ LLOYD, Ian; The Ultimate HTML Referente, Primera Edición; Mayo 2008.
  
- ✓ Apuntes sobre “Desarrollo de Sistemas Orientado a Objetos con UML”

### SITIOS WEB

- ✓ Sitio web “Adictos al Trabajo”  
URL: <http://www.adictosaltrabajo.com>  
  
Descripción: Sitio que ofrece gran cantidad de tutoriales y ejemplos acerca de tecnologías java, mysql, servidores de aplicaciones web, etc.
  
- ✓ Sitio web “Albanet”  
URL: <http://www.albanet.com.mx>  
  
Descripción: Sitio dedicado a ofrecer artículos acerca de ciencia y tecnología java, jsp, servlets.
  
- ✓ Sitio web  
URL: <http://www.angelfire.com/la/hmolina>  
  
Descripción: Pagina personal de Hender Molina donde está disponible un tutorial acerca de Internet.
  
- ✓ Sitio web “Apache”  
URL: <http://www.apache.org>  
  
Descripción: Sitio dedicado al desarrollo del servidor web apache, y el contenedor de servlets tomcat, entre otros proyectos, se puede obtener las últimas versiones así como documentación del contenedor de servlets.
  
- ✓ Sitio web  
URL: <http://www.ayuda-internet.net/tutoriales>



Descripción: Sitio en el que se encuentra una colección de tutoriales de programación cliente servidor, java, jsp y diversos temas de estudio.

- ✓ Sitio web “Monografías”

URL: <http://www.monografias.com>

Descripción: Sitio dedicado a la recopilación y difusión de trabajos realizados acerca de Internet, java, jsp y variados temas de estudios.

- ✓ Sitio web “MySQLAB”

URL: <http://www.mysqlab.com>

Descripción: Sitio dedicado al desarrollo del servidor de bases de datos MySql, se puede encontrar las últimas versiones y documentación del mismo.

- ✓ Sitio web “Java en Castellano”

URL: <http://www.programacion.com>

Descripción: Sitio dedicado a ofrecer tutoriales y cursos sobre programación en java, servidores web, etc.

- ✓ Sitio web “Sun Microsystems”

URL: <http://www.sun.com>

Descripción: Sitio dedicado al desarrollo de la máquina virtual de java, donde se puede obtener las últimas versiones de java, documentación, tecnologías java y jsp, ide netbeans para desarrollo java, java media framework para trabajar con audio y video, java server faces para el desarrollo de aplicaciones web, etc.

- ✓ Sitio web “Wikipedia”

URL: <http://es.wikipedia.org>

Descripción: Sitio que ofrece una enciclopedia libre con contenidos acerca de Internet, sistemas de comunicación, tecnologías web, etc.





# ANEXOS

## ANEXO 1: FORMULACIÓN DEL PROYECTO DE DESARROLLO DE TESIS DE GRADO





## **ANEXO 2: ENCUESTA APLICADA EN LA FASE DE PRUEBAS**

