

INSTITUTO TECNOLÓGICO DE COSTA RICA
DEPARTAMENTO DE COMPUTACIÓN
PROGRAMA DE MAESTRÍA

IMPLEMENTACIÓN DE UN FORO DE DISCUSIÓN JERÁRQUICO, BASADO EN XML, CON CONSIDERACIONES DE PLAUSIBILIDAD

Tesis para optar al grado de
Magister Scientiae en Computación

Por: Raúl Enrique Dutari Dutari

Profesor Asesor: Dr. Ulises Agüero Arroyo

Cartago, Costa Rica
Agosto del 2001

APROBACIÓN DE TESIS

Implementación de un Foro de Discusión Jerárquico, Basado en XML, con Consideraciones de Plausibilidad

TRIBUNAL EXAMINADOR

Giannina Núñez Marín, MSc.
PROFESIONAL EXTERNA

Mario Daniel Ramírez, MSc.
PROFESOR

Dr. José Enrique Araya
PROFESOR

Dr. Ulises Agüero A.
PROFESOR-ASESOR

Dr. Francisco Torres
DIRECTOR/DEPARTAMENTO

Agosto, 2001

IMPLEMENTACIÓN DE UN FORO DE DISCUSIÓN JERÁRQUICO, BASADO EN XML, CON CONSIDERACIONES DE PLAUSIBILIDAD

RESUMEN

El ser humano nace y vive en sociedad, pero en muchos aspectos, es obligado a desarrollar sus actividades desde un punto de vista individualista, pesando más su desempeño individual que su desempeño en equipo. Este problema se presenta desde que el individuo ingresa al sistema educativo formal, porque la mayoría de las actividades que se desarrollan en él, son de índole individualista, en especial la evaluación. La toma de decisiones en grupo, no es un componente diario del trabajo escolar.

El foro de discusión es una de las herramientas de uso más generalizado, en el trabajo colaborativo apoyado por computador, pero no apoya plenamente los procesos de toma de decisiones, aunque ya existen construcciones formales que sustentan la implementación de herramientas colaborativas más eficientes.

Esta investigación plantea la implementación de un Foro de Discusión Jerárquico, que apoya los procesos de toma de decisiones grupales. Es capaz de calcular el estado de plausibilidad de cada hilo de discusión del foro, a medida que se desarrolla el diálogo entre sus participantes.

Se presenta un compendio de las teorías que pueden servir de apoyo a los procesos de toma de decisiones grupales. Se analizan algunas herramientas computarizadas que facilitan los procesos de toma de decisiones en grupo. Se examinan algunas de las tecnologías que se pueden utilizar en la implementación de este tipo de herramientas.

Posteriormente se expone la especificación del Foro de Discusión, en términos del análisis de sus casos de uso, sus diagramas de colaboración, y el diagrama de paquetes.

Luego se expone la arquitectura interna del sistema, en términos de los diagramas de emplazamiento, y de clases estáticas, así como las estructuras de datos XML, el esquema de identificación de enunciados, y la estructura de los mensajes que intercambiarán las aplicaciones que constituyen el sistema.

Finalmente, se explica la validación del sistema contra la formalización que se utiliza como fundamento teórico de esta investigación.

Palabras clave: teoría de plausibilidad, foro de discusión jerárquico, aplicación colaborativa, XML, UML, Java.

IMPLEMENTATION OF A HIERARCHICAL FORUM OF DISCUSSION, BASED IN XML, WITH CONSIDERATIONS OF PLAUSIBILITY

ABSTRACT

The human being is born and he lives in society, but in many aspects, it is forced to develop his activities from an individualistic point of view, weighing more his individual acting that his acting in team. This problem is presented since the individual enters to the formal educational system, because most of the activities that are developed in him are of individualistic nature, especially the evaluation. The taking of decisions in-group, is not a daily component of the schoolwork.

The discussion forum is one of the tools of more widespread use, in the "Computer-Supported Collaborative Work", but it doesn't support the processes fully of taking of decisions, although formal constructions that sustain the implementation of more efficient collaborative tools already exist.

This investigation outlines the implementation of a Hierarchical Forum of Discussion that supports the processes of taking of groups decisions. It is able to calculate the state of plausibility of each thread of discussion of the forum, as the dialogue is developed among its participants.

A summary of the theories is presented that can serve from support to the processes of taking of groups decisions. Some on-line tools are analyzed that facilitate the processes of taking of decisions in-group. Some of the technologies are examined that can be used in the implementation of this type of tools.

Later on, the specification of the Forum of Discussion is exposed, in terms of the analysis of its cases of use, its diagrams of collaboration, and the diagram of packages.

Then the internal architecture of the system is exposed, in terms of the location diagrams, and of static classes, as well as the structures of data XML, the identification outline of having enunciated, and the structure of the messages that you/they will exchange the applications that constitute the system.

Finally, the validation of the system is explained against the formalization that is used like theoretical foundation of this investigation.

Keywords: theory of plausibility, hierarchical forum of discussion, collaborative application, XML, UML, Java.

Dedicatoria

*Al supremo creador, porque sin su apoyo
es imposible lograr algo en esta vida.*

*A mis padres, Juan B. y María E., quienes
con sus esfuerzos, apoyo y amor me
alentaron para alcanzar la realización de
uno de mis más preciados sueños.*

*A Yodania del Carmen Quirós Aguilar
“Yodita”, porque su amor, su cariño, su
respeto y su amistad, son el impulso, el
“motorcito” que me animó en los
momentos más difíciles de este proyecto.
Cuando todo lo daba por perdido, ella fue
-y es- mi motivación más importante para
continuar superándome a nivel
académico, profesional y personal, pese
a las grandes dificultades que enfrenté y,
seguramente, seguiré enfrentando. A
ella, a quién deseo dedicar mi vida; quiera
Dios que nuestro amor dure por siempre,
en esta vida y en la otra.*

Agradecimiento

Son muchas las personas e instituciones que han intervenido, directa o indirectamente, en la realización de mis estudios de maestría. A todos, les extiendo mi más profunda gratitud.

Es imposible presentar este trabajo sin agradecer profundamente al Servicio Alemán de Intercambio Académico “DAAD”, por ofrecerme la oportunidad de realizar mis estudios de maestría.

El personal docente y administrativo del Instituto Tecnológico de Costa Rica “TEC” tiene mi gratitud permanente, por apoyarme, cada uno en la medida de sus posibilidades, en el desarrollo de mis estudios de maestría. Las enseñanzas de profesores como los Doctores Carlos González y José Enrique Araya son modelos de docentes que guiarán mi quehacer en el futuro. La ayuda, siempre oportuna y desinteresada, del personal administrativo y de la Jefatura del Departamento de Computación del TEC, en las personas del Doctor Francisco Torres, Grace, Patricia, Norma y Mónica, no se puede olvidar; en el momento en que corresponda, pueden contar con mi ayuda incondicional, y con mi eterno agradecimiento.

Agradezco también, enormemente el apoyo y la confianza que el Dr. Ulises Agüero Arroyo depositó en mí, al aceptar ser mi tutor en el desarrollo de esta investigación. Confío en haber alcanzado sus expectativas en cuanto a la realización de este proyecto, y de ser posible, que podamos encontrar otras áreas de interés común donde desarrollar otros proyectos de investigación, en el futuro.

Me siento profundamente agradecido por las sugerencias y críticas constructivas que recibí de mis compañeros de maestría, para mejorar y ampliar la calidad de este proyecto de investigación. En especial, de Alicia y Salvador; porque sin sus atinadas observaciones y sugerencias, esta investigación no sería igual a la que presento.

Finalmente, agradezco profundamente el apoyo brindado por todas aquellas personas que, de una u otra forma, han contribuido a que este trabajo de graduación sea una realidad, pero que por una u otra razón, no han sido mencionadas.

ÍNDICE GENERAL

RESUMEN	III
ABSTRACT	V
ÍNDICE GENERAL	IX
ÍNDICE DE ILUSTRACIONES.....	XIX
ÍNDICE DE TABLAS.....	XXIII
1. INTRODUCCIÓN, MARCO TEÓRICO Y METAS.....	1
1.1. INTRODUCCIÓN.	1
1.2. DESCRIPCIÓN DEL CONTENIDO.....	2
1.3. MARCO TEÓRICO.	3
1.3.1. TOMA DE DECISIONES EN GRUPO.....	4
1.3.2. SISTEMAS COLABORATIVOS.	6
1.3.3. HERRAMIENTAS COMPUTARIZADAS QUE FACILITAN LA TOMA DE DECISIONES EN GRUPO.....	9
1.3.3.1. SISTEMAS QUE APOYAN LOS PROCESOS DE TOMA DE DECISIONES EN GRUPO.	9
1.3.3.2. FOROS DE DISCUSIÓN.	13

1.3.4.	TECNOLOGÍAS QUE SE PUEDEN UTILIZAR PARA LA IMPLEMENTACIÓN DE HERRAMIENTAS QUE APOYEN LOS PROCESOS DE TOMA DE DECISIONES.	16
1.3.4.1.	INTERNET.	16
1.3.4.2.	JAVA.	18
1.3.4.3.	XML.....	19
1.3.4.4.	DOM.....	21
1.3.4.5.	FRAMEWORKS QUE FACILITAN EL DESARROLLO DE APLICACIONES COLABORATIVAS.	22
1.4.	METAS.....	24
1.4.1.	ANTECEDENTES DEL PROBLEMA.	24
1.4.2.	OBJETIVO GENERAL.	24
1.4.3.	OBJETIVOS ESPECÍFICOS.....	25
1.4.4.	PRODUCTOS DE LA INVESTIGACIÓN.....	25
1.4.5.	JUSTIFICACIÓN.....	26
1.4.6.	ORIGINALIDAD E IMPACTO.....	26
1.4.7.	DELIMITADORES.....	28

2.	ESPECIFICACIÓN DEL FORO DE DISCUSIÓN JERÁRQUICO.....	30
2.1.	DESCRIPCIÓN GENERAL DEL PROBLEMA.	31
2.2.	TIPOS DE USUARIO.....	31
2.3.	PROBLEMA QUE SE RESOLVERÁ.....	32
2.4.	FUNCIONES.....	34
2.5.	CASOS DE USO.....	35
2.5.1.	CASO DE USO: INICIAR LA SESIÓN.	36
2.5.2.	CASO DE USO: CALCULAR LA PLAUSIBILIDAD DEL HILO DE DISCUSIÓN PRIMARIO.	38
2.5.3.	CASO DE USO: CALCULAR LA PLAUSIBILIDAD DEL DIÁLOGO.....	38
2.5.4.	CASO DE USO: INGRESAR A LA SESIÓN.	38
2.5.5.	CASO DE USO: AGREGAR ENUNCIADO.....	39
2.5.6.	CASO DE USO: MODIFICAR ENUNCIADO.....	41
2.5.7.	CASO DE USO: CONSULTAR EL DIÁLOGO.	42
2.5.8.	CASO DE USO: SALIR DE LA SESIÓN.....	43

2.5.9.	CASO DE USO: FINALIZAR SESIÓN.	44
2.6.	MODELO CONCEPTUAL.	45
2.7.	DIAGRAMAS DE COLABORACIÓN.....	50
2.7.1.	COLABORACIÓN: INICIAR LA SESIÓN.	50
2.7.2.	COLABORACIÓN: CALCULAR LA PLAUSIBILIDAD DEL HILO DE DISCUSIÓN PRIMARIO.	51
2.7.3.	COLABORACIÓN: CALCULAR LA PLAUSIBILIDAD DEL DIÁLOGO.....	53
2.7.4.	COLABORACIÓN: INGRESAR A LA SESIÓN	54
2.7.5.	COLABORACIÓN: AGREGAR ENUNCIADO.....	55
2.7.6.	COLABORACIÓN: MODIFICAR ENUNCIADO.....	56
2.7.7.	COLABORACIÓN: CONSULTAR EL DIÁLOGO.....	58
2.7.8.	COLABORACIÓN: SALIR DE LA SESIÓN.	59
2.7.9.	COLABORACIÓN: FINALIZAR SESIÓN.	60
2.8.	ADICIÓN DE MÉTODOS AL MODELO CONCEPTUAL.....	61
2.9.	DIAGRAMA DE PAQUETES.....	65

3.	ARQUITECTURA Y DISEÑO DEL FORO DE DISCUSIÓN JERÁRQUICO.....	68
3.1.	DIAGRAMA DE EMPLAZAMIENTO: ARQUITECTURA CLIENTE / SERVIDOR DEL FDJ.....	68
3.2.	INTERACCIÓN ENTRE LAS CLASES DEL DIAGRAMA DE EMPLAZAMIENTO.	71
3.3.	MODELO DE IDENTIFICACIÓN DE LOS ENUNCIADOS DENTRO DEL DIÁLOGO.....	75
3.4.	ESTRUCTURA DE LOS DOCUMENTOS XML UTILIZADOS.....	79
3.4.1.	LISTA DE USUARIOS.	80
3.4.2.	DIÁLOGO DEL FORO.	82
3.4.3.	CONSULTA GENERAL.	88
3.4.4.	CONSULTA DEL CLIENTE.	91
3.5.	ESTRUCTURA DE LOS MENSAJES QUE SE INTERCAMBIAN ENTRE EL CLIENTE Y EL SERVIDOR.....	97
3.5.1.	FORMATO DE LOS MENSAJES SIMPLES.	97
3.5.2.	FORMATO DE LOS MENSAJES COMPLEJOS.....	99

3.5.3.	MENSAJES QUE SE ENVÍAN DESDE EL SERVIDOR HACIA LOS CLIENTES.	100
3.5.4.	MENSAJES QUE SE ENVÍAN DESDE LOS CLIENTES HACIA EL SERVIDOR.	103
3.6.	DIAGRAMA DE CLASES ESTÁTICAS.	104
3.7.	DISTRIBUCIÓN DE ATRIBUTOS EN EL DIAGRAMA DE CLASES.	106
3.8.	ADICIÓN DE MÉTODOS AL DIAGRAMA DE CLASES ESTÁTICAS.	113
4.	VALIDACIÓN DEL SISTEMA, CONTRIBUCIONES, CONCLUSIONES Y TRABAJOS DERIVADOS.	121
4.1.	VALIDACIÓN DEL SISTEMA CONTRA LA ESPECIFICACIÓN DEL <i>FDJ</i>	121
4.2.	CONTRIBUCIONES REALIZADAS POR ESTA INVESTIGACIÓN.	124
4.3.	CONCLUSIONES OBTENIDAS DE ESTA INVESTIGACIÓN.	125
4.4.	TRABAJOS QUE SE PUEDEN DERIVAR DE ESTA INVESTIGACIÓN.	128
5.	REFERENCIAS BIBLIOGRÁFICAS.	131

6.	APÉNDICE 1: MANUAL DE USUARIO DEL FORO DE DISCUSIÓN JERÁRQUICO.....	136
6.1.	DESCRIPCIÓN GENERAL DE LA APLICACIÓN.	136
6.2.	TIPOS DE USUARIO.	136
6.3.	REQUERIMIENTOS.	137
6.4.	INSTALACIÓN.	138
6.5.	ACTIVACIÓN DE LAS APLICACIONES DEL <i>FDJ</i>	142
6.6.	FUNCIONAMIENTO DEL <i>FDJ</i>	143
6.6.1.	FUNCIONAMIENTO DEL SERVIDOR.	144
6.6.1.1.	INICIO DE LA SESIÓN.	144
6.6.1.2.	DIÁLOGO PRINCIPAL.....	146
6.6.1.3.	CAMBIAR EL MÁXIMO DE USUARIOS PERMITIDOS EN LA SESIÓN.	148
6.6.1.4.	FINALIZAR LA SESIÓN.....	150
6.6.2.	FUNCIONAMIENTO DEL CLIENTE.	150
6.6.2.1.	INICIO DE LA SESIÓN.	151
6.6.2.2.	DIÁLOGO PRINCIPAL.....	153

6.6.2.3.	AGREGAR ENUNCIADOS AL DIÁLOGO.....	156
6.6.2.4.	MODIFICAR ENUNCIADOS DEL DIÁLOGO.....	159
6.6.2.5.	CAMBIAR EL TIPO DE CONSULTA QUE REALIZA UN CLIENTE.....	162
6.6.2.6.	FINALIZAR LA SESIÓN.....	164
6.6.3.	CÁLCULO DEL ESTADO DE PLAUSIBILIDAD DE LOS HILOS DE DISCUSIÓN.	164
6.7.	MEJORAS Y AMPLIACIONES FUTURAS DEL <i>FDJ</i>	165
6.8.	ERRORES CONOCIDOS.	167
7.	APÉNDICE 2: GLOSARIO DE TÉRMINOS DEL DOMINIO DE LA APLICACIÓN “ <i>FORO DE DISCUSIÓN</i> <i>JERÁRQUICO</i> ”.	168
8.	ANEXO 1: FORMALIZACIÓN DEL FORO DE DISCUSIÓN JERÁRQUICO.....	171
8.1.	INTRODUCCIÓN	171
8.2.	ENUNCIADO.....	173
8.2.1.	IDENTIFICACIÓN DE UN ENUNCIADO.....	174
8.2.2.	NATURALEZA DE UN ENUNCIADO.....	175

8.2.3.	ESTADO DE PLAUSIBILIDAD.....	177
8.3.	HILOS DE DISCUSIÓN (THREADED DISCUSSIONS).....	179
8.3.1.	HILO DE DISCUSIÓN PRIMARIO	179
8.3.2.	HILO DE DISCUSIÓN SECUNDARIO	180
8.4.	FORO DE DISCUSIÓN.....	180
8.5.	EVIDENCIA.....	181
8.5.1.	EVIDENCIA ASOCIADA A UN ENUNCIADO	182
8.5.2.	EVIDENCIA DISTINGUIBLE	182
8.6.	EVOLUCIÓN DE LA NATURALEZA DE LOS ENUNCIADOS	183
8.7.	OPERACIONES SOBRE EL FORO DE DISCUSIÓN.....	185
8.7.1.	AÑADIR	186
8.7.2.	MODIFICAR	186
8.7.3.	CONSULTAR	188
8.8.	RETROALIMENTACIÓN.....	189
8.8.1.	PROPIEDADES DE PLAUSIBILIDAD.....	190
8.8.2.	POLÍTICA DE CONSENSO	190

8.8.3.	ALGORITMO DFS-PLAUSIBLE	193
8.9.	EJEMPLO DE ESTADOS DE PLAUSIBILIDAD EN UN FORO.....	197
8.10.	VOTACIÓN	198
8.10.1.	MÉTODO DE VOTACIÓN.....	199
8.10.1.1.	PRINCIPIOS DE VOTACIÓN.....	199
8.10.1.2.	NORMAS DE VOTACIÓN.....	200
8.10.1.3.	POLÍTICA DE CONSENSO PARA VOTACIÓN	203
8.10.2.	MOMENTOS DE VOTACIÓN	204
8.11.	EJEMPLO DE VOTACIÓN.....	205
8.12.	MÉTODOS DE DISCUSIÓN	206

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Matriz Tiempo-Espacio de Johansen	7
Ilustración 2: Casos de Uso Básicos del <i>FDJ</i>	36
Ilustración 3: Caso de Uso “Iniciar la Sesión”	37
Ilustración 4: Caso de Uso “Ingresar a la Sesión”	39
Ilustración 5: Caso de Uso “Agregar Enunciado”	40
Ilustración 6: Caso de Uso “Modificar Enunciado”	42
Ilustración 7: Caso de Uso “Consultar el Diálogo”	43
Ilustración 8: Caso de Uso “Salir de la Sesión”	44
Ilustración 9: Caso de Uso “Finalizar Sesión”	45
Ilustración 10: Modelo Conceptual del <i>FDJ</i>	46
Ilustración 11: Colaboración “Iniciar la Sesión”.....	51
Ilustración 12: Colaboración “Calcular la Plausibilidad del Hilo de Discusión Primario”	52
Ilustración 13: Colaboración “Calcular la Plausibilidad del Diálogo”	53
Ilustración 14: Colaboración “Ingresar a la Sesión”	55

Ilustración 15: Colaboración “Agregar Enunciado”	56
Ilustración 16: Colaboración “Modificar Enunciado”	58
Ilustración 17: Colaboración “Consultar el Diálogo”	59
Ilustración 18: Colaboración “Salir de la Sesión”	60
Ilustración 19: Colaboración “Finalizar Sesión”	61
Ilustración 20: Adición de Métodos al Modelo Conceptual	64
Ilustración 21: Diagrama de Paquetes del <i>FDJ</i>	65
Ilustración 22: Diagrama de Emplazamiento del <i>FDJ</i>	69
Ilustración 23: Función Generadora de Llaves para los Nodos de un Árbol n-ario	76
Ilustración 24: Diálogo Genérico con los Nodos Identificados Bajo el Modelo Decimal	78
Ilustración 25: Cálculo de Algunos Identificadores de Nodos de la Ilustración 24	78
Ilustración 26: DTD para la Lista de Usuarios del <i>FDJ</i>	80
Ilustración 27: Documento XML de la Lista de Usuarios del <i>FDJ</i>	82
Ilustración 28: DTD para la el Diálogo del <i>FDJ</i>	83

Ilustración 29: Documento XML del Diálogo del <i>FDJ</i>	87
Ilustración 30: DTD para la Consulta General del <i>FDJ</i>	88
Ilustración 31: Documento XML de la Consulta General del <i>FDJ</i>	91
Ilustración 32: DTD para la Consulta del Cliente del <i>FDJ</i>	92
Ilustración 33: Documento XML de la Consulta del Cliente del <i>FDJ</i>	96
Ilustración 34: Formato de Mensaje Simple del <i>FDJ</i>	98
Ilustración 35: Formato de Mensaje Complejo del <i>FDJ</i>	100
Ilustración 36: Diagrama de Clases Estáticas del <i>FDJ</i>	105
Ilustración 37: Diagrama de Clases Estáticas del <i>FDJ</i> con Atributos	112
Ilustración 38: Adición de Métodos al Diagrama de Clases Estáticas del <i>FDJ</i> (Cliente)	119
Ilustración 39: Adición de Métodos al Diagrama de Clases Estáticas del <i>FDJ</i> (Servidor)	120
Ilustración 40: Diálogo de Inicio de Sesión en el Servidor	144
Ilustración 41: Diálogo Principal del Servidor	146
Ilustración 42: Diálogo de Propiedades del Servidor	148

Ilustración 43: Diálogo para Alterar el Máximo de Usuarios Permitidos en la Sesión	149
Ilustración 44: Mensaje de Error en la Captura del Máximo de Usuarios Permitidos	149
Ilustración 45: Diálogo para Finalizar la Sesión del Servidor.....	150
Ilustración 46: Diálogo de Inicio de Sesión en el Cliente	151
Ilustración 47: Diálogo Principal del Cliente	153
Ilustración 48: Formato de los Enunciados del <i>FDJ</i>	155
Ilustración 49: Diálogo para Agregar Enunciados al <i>FDJ</i>	157
Ilustración 50: Diálogo para Modificar Enunciados.....	159
Ilustración 51: Mensaje de Error al Modificar un Enunciado que no Pertenece al Usuario.....	161
Ilustración 52: Lista de Consultas.....	163
Ilustración 53: Diálogo para Finalizar la Sesión del Cliente.....	164

ÍNDICE DE TABLAS

Tabla 1: Funciones Principales del <i>FDJ</i>	35
Tabla 2: Clases del Modelo Conceptual del <i>FDJ</i>	47
Tabla 3: Atributos del Modelo Conceptual del <i>FDJ</i>	48
Tabla 4: Métodos del Modelo Conceptual del <i>FDJ</i>	63
Tabla 5: Atributos y Métodos dentro del Diagrama de Paquetes del <i>FDJ</i>	67
Tabla 6: Relaciones entre las Clases de los Diagramas de Paquetes y Emplazamiento	71
Tabla 7: Interpretación del Diagrama de Emplazamiento del <i>FDJ</i>	74
Tabla 8: Mensajes Simples que Envía el Servidor a los Clientes del <i>FDJ</i>	102
Tabla 9: Mensajes Complejos que Envía el Servidor a los Clientes del <i>FDJ</i> ...	102
Tabla 10: Mensajes que Envían los Clientes al Servidor del <i>FDJ</i>	104
Tabla 11: Atributos de la Clase <i>AdministradorUsuario</i> del Servidor en el Diagrama de Clase del <i>FDJ</i>	106
Tabla 12: Atributos de la Clase <i>AdministradorDialogo</i> del Servidor en el Diagrama de Clase del <i>FDJ</i>	107
Tabla 13: Atributos de la Clase <i>AdministradorConsultaServidor</i> en el Diagrama de Clase del <i>FDJ</i>	108

Tabla 14: Atributos de la Clase <i>AdministradorComunicacionServidor</i> en el Diagrama de Clase del <i>FDJ</i>	109
Tabla 15: Atributos de la Clase <i>ConexionClienteServidor</i> del Servidor en el Diagrama de Clase del <i>FDJ</i>	109
Tabla 16: Atributos de la Clase <i>AdministradorConsultaCliente</i> del Servidor en el Diagrama de Clase del <i>FDJ</i>	110
Tabla 17: Atributos de la Clase <i>AdministradorComunicacionCliente</i> del Servidor en el Diagrama de Clase del <i>FDJ</i>	111
Tabla 18: Métodos de la Clase <i>AdministradorConsultaCliente</i> del <i>FDJ</i>	114
Tabla 19: Métodos de la Clase <i>AdministradorComunicacionCliente</i> del <i>FDJ</i> ...	114
Tabla 20: Métodos de la Clase <i>AdministradorUsuario</i> del <i>FDJ</i>	115
Tabla 21: Métodos de la Clase <i>AdministradorDialogoServidor</i> del <i>FDJ</i>	116
Tabla 22: Métodos de la Clase <i>AdministradorConsultaServidor</i> del <i>FDJ</i>	117
Tabla 23: Métodos de la Clase <i>ConexionClienteServidor</i> del <i>FDJ</i>	117
Tabla 24: Métodos de la Clase <i>AdministradorComunicacionServidor</i> del <i>FDJ</i>	117

1. INTRODUCCIÓN, MARCO TEÓRICO Y METAS.

En este capítulo se expone el marco teórico que sustenta a esta investigación y las metas que se alcanzarán.

En tal sentido, se presentan:

1. La introducción,
2. La descripción del contenido,
3. El marco teórico,
4. Las metas que se pretende alcanzar.

1.1. INTRODUCCIÓN.

Es universalmente sabido que los seres humanos nacen y viven en sociedad. Sin embargo, la sociedad los condiciona a desarrollarse desde un punto de vista individualista. Por ejemplo, los empleados de una empresa cualquiera, deben trabajar en equipo, pero en el momento de considerar su eficiencia, en muchas empresas se da mayor importancia a su desempeño individual, más que a su desempeño grupal [OLIV96].

La situación antes descrita tiene un gran arraigo en nuestra sociedad, porque dentro de los sistemas educativos formales es difícil encontrar situaciones en las que se *“enseñe a vivir en sociedad”*. Aunque se *“estudia en grupo”*, la mayoría de las actividades, incluyendo la evaluación, se fundamentan en el desempeño individual del estudiante [NÚÑE99].

Por otro lado, tan complejo como aprender a leer y escribir, es aprender a tomar decisiones en grupo. En tal sentido, el foro de discusión, en general, es una de las herramientas de uso más generalizado en el trabajo cooperativo apoyado por redes de computadores. Sin embargo, este tipo de herramienta no ha evolucionado acorde con las necesidades de nuestra sociedad, en términos de facilitar los procesos de toma de decisiones. Esta situación se agrava, porque a la fecha, ya existen las construcciones formales necesarias que sustentarían la implementación de herramientas de este tipo más eficientes, tales como las presentadas por [NÚÑE99].

En ese sentido, esta investigación tratará de cambiar dicha situación.

1.2. DESCRIPCIÓN DEL CONTENIDO.

En este documento se presentan cinco capítulos, dos apéndices y un anexo, que integran esta investigación.

El primer capítulo presenta el marco teórico que sustenta esta investigación, en términos de las teorías de sistemas colaborativos, que pueden servir de apoyo a los procesos de toma de decisiones en grupo. También se presenta un análisis de algunas herramientas computarizadas que facilitan los procesos de toma de decisiones en grupo. Se analizan algunas de las tecnologías que se pueden utilizar en la implementación de herramientas que apoyen los procesos de toma de decisiones. Finalmente, se presentan las metas que alcanzará esta investigación.

El segundo capítulo expone el análisis del Foro de Discusión Jerárquico, al nivel de especificación. En tal sentido, se presentan y analizan: El problema que se

resolverá, las funciones del sistema, los casos de uso con sus diagramas, el modelo conceptual, los diagramas de colaboración y el diagrama de paquetes.

En el tercer capítulo, se expone el diseño y la arquitectura interna del sistema. En tal sentido, se presentan y analizan: El diagrama de emplazamiento, el diagrama de clases estáticas, el esquema de identificación de enunciados, las estructuras que datos XML que se utilizarán, la estructura de los mensajes que intercambiará el *“Foro de Discusión Jerárquico”* -en adelante *FDJ*-, así como los atributos y métodos presentes en el diagrama de clases estáticas.

El cuarto capítulo presenta la validación del sistema contra la formalización, que representa el trabajo de [NÚÑE99], además de las conclusiones y trabajos derivados que culminan esta investigación.

El quinto capítulo presenta las referencias bibliográficas de esta investigación.

El primer apéndice presenta el manual de usuario, para el cliente y el servidor del foro de discusión.

El segundo apéndice expone el glosario de términos del dominio de la aplicación -Foro de Discusión Jerárquico-.

Finalmente, primer anexo presenta la formalización del foro de discusión, según [NÚÑE99].

1.3. MARCO TEÓRICO.

En esta sección se presenta el marco teórico que sustenta esta investigación. En tal sentido:

1. Se describen las principales características que deben acompañar a los sistemas de apoyo a la toma de decisiones,
2. Se presenta un breve resumen de la teoría de los sistemas colaborativos,
3. Se analizan las principales características de los sistemas que apoyan los procesos de toma de decisiones en grupo, así como de los foros de discusión; describiendo algunas aplicaciones implementadas comercialmente,
4. Finalmente, se analizan las principales tecnologías que se pueden utilizar en la implementación de herramientas que apoyen los procesos de toma de decisiones.

1.3.1. TOMA DE DECISIONES EN GRUPO.

El hombre, al desarrollar sus actividades vitales, esta constantemente obligado a realizar procesos de toma de decisiones grupales.

Según [OLIV96] se presenta un problema de toma de decisiones, cuando un grupo de personas requiere tomar una decisión. La complejidad de este problema aumenta a medida que aumenta el tamaño del grupo de individuos involucrados. Otros factores que también influyen en la complejidad de este problema son: la proximidad física de los miembros del grupo, y la dificultad intrínseca del problema abordado. Sin embargo, los últimos factores se consideran de menor relevancia dentro de la solución de este problema.

En tal sentido, el proceso de toma de decisiones se complica principalmente con el crecimiento del tamaño del grupo, por el crecimiento exponencial de la

posibilidad de comunicación simultánea entre los participantes. Esto, porque la capacidad humana para mantener comunicaciones simultáneas, es limitada.

Sin embargo, dentro de ciertas limitaciones, es factible plantear sistemas de apoyo computacional a la toma de decisiones. Las limitaciones que plantea **[OLIV96]** son:

1. Debe aplicarse un método perfectamente estructurado,
2. Debe diferenciarse perfectamente una etapa de generación de alternativas y una etapa de selección de opciones, la segunda posterior a la primera,
3. La transformación de alternativas a opciones la realiza un experto, no el grupo de decisores,
4. En la etapa de generación de alternativas, la comunicación directa entre los decisores está restringida,
5. En la etapa de selección de opciones no hay comunicación directa entre decisores.

Por otro lado, la teoría de plausibilidad tal y como la plantea **[OLIV96]** permite apoyar los procesos de toma de decisiones, al brindar un modelo orientado a la formación del consenso en el grupo y porque ofrece amplias posibilidades de expresar la cooperación.

A continuación se presenta un breve recuento de las teorías que sustentan los diseños e implementaciones de aplicaciones que apoyan los procesos de toma de decisiones grupales, para facilitar la comprensión de este documento a las

personas que no están relacionadas directamente con la temática que se analiza en él.

1.3.2. SISTEMAS COLABORATIVOS.

Los sistemas colaborativos se enmarcan dentro del área de investigación conocida como CSCW¹. El CSCW es una actividad coordinada, apoyada por computadores, que tiene como objetivo la resolución de problemas, y la comunicación efectiva entre grupos de personas que colaboran entre si, según [KOCH95].

El software multiusuario que apoya al CSCW, se conoce como “*Groupware*”. Una definición muy aceptada del término Groupware la presenta [ELGI91]; al enunciar que son: “*Sistemas de computadoras que apoyan a grupos de personas que trabajan en tareas o metas comunes y proveen una interfaz a un ambiente de trabajo compartido*”.

Para [GUER98], existen aplicaciones que facilitan la interacción a distancia entre los miembros de un equipo de trabajo; en tanto, otras apoyan la comunicación de personas reunidas físicamente en el mismo lugar. Además, existen aplicaciones para la confección de documentos compartidos por grupos de personas que no se encuentran reunidas físicamente en el mismo lugar.

¹ “Computer Supported Cooperative Work”, o Trabajo Cooperativo Soportado por Computador.

Los sistemas colaborativos pueden ser clasificados, de acuerdo a [JOHA88], basándose en la forma en que la gente interactúa cuando trabaja. Esta taxonomía es presentada en la siguiente Ilustración.

		<u>Espacio o Lugar</u>	
		Mismo	Diferente
<u>Tiempo</u>	Mismo	Interacción cara a cara	Interacción sincrónica distribuida
	Diferente	Interacción asincrónica	Interacción asincrónica distribuida

Ilustración 1: Matriz Tiempo-Espacio de Johansen

Para ilustrar la aplicación de esta taxonomía a un tipo de aplicación particular, consideremos el correo electrónico; esta es una actividad en la que normalmente se produce interacción de tipo asincrónica distribuida, porque los usuarios que se comunican por este medio, generalmente lo hacen desde localidades geográficas distantes, en horarios diferentes.

Según [GUER98], algunos de los conceptos que deben estar bien definidos al referirnos a las aplicaciones colaborativas, son:

1. **Usuario:** Miembro de un grupo o equipo de trabajo que utiliza una aplicación colaborativa como apoyo a sus actividades,

2. **Rol:** Patrón de comportamiento social esperado, determinado por el estatus de un individuo ante la sociedad. Los roles determinan lo que puede y debe hacer cada uno de los miembros del grupo,
3. **Sesión:** Un período de tiempo dedicado a la interacción sincrónica o asincrónica por parte de los usuarios de un grupo, apoyados por un sistema de Groupware,
4. **Memoria grupal:** Es el espacio de almacenamiento, común a todos los usuarios, donde se almacena, ordenadamente, la información relativa al desarrollo de la actividad realizada por el grupo de usuarios, a nivel del proceso de trabajo y del producto final de la actividad grupal,
5. **Vista o consulta:** Es una representación visual o multimedial de alguna porción del contexto o área de trabajo compartido. Diferentes vistas pueden contener la misma información, pero diferir en su presentación a los usuarios. Asimismo, diferentes vistas pueden ocultar cierta información, dependiendo del rol del usuario que la solicita,
6. **Protocolo de control de piso:** Es utilizado para proveer coordinación y control de concurrencia sobre los objetos compartidos en una sesión. Determina la forma de asignar el control a los usuarios que lo solicitan,
7. **Percepción:** Es la visión general, que tiene cada usuario, del ambiente de trabajo y la información que se recibe de los otros miembros del equipo – conocimiento de: quiénes están alrededor y que actividades han realizado o realizan—.

1.3.3. HERRAMIENTAS COMPUTARIZADAS QUE FACILITAN LA TOMA DE DECISIONES EN GRUPO.

Existen varios tipos de herramientas computarizadas que permiten apoyar la toma de decisiones en grupo, en mayor o menor grado. Para los efectos de la investigación que se plantea, resultan relevantes:

1. Los sistemas que apoyan los procesos de toma de decisiones en grupo,
2. Los foros de discusión.

1.3.3.1. SISTEMAS QUE APOYAN LOS PROCESOS DE TOMA DE DECISIONES EN GRUPO.

Estas herramientas brindan una área de trabajo compartida donde cada usuario aporta sus ideas al grupo. Además, implementan algún tipo de control de piso – generalmente FIFO–, en caso que varios usuarios traten de incluir simultáneamente sus ideas en el área de trabajo compartida. Finalmente, se fundamentan en alguna metodología que apoye los procesos de toma de decisiones en grupo.

En general, las herramientas computacionales que se analizarán, ofrecen facilidades a sus usuarios para la realización de diversas tareas colaborativas, tales como:

- ☒ Generar mapas de ideas, bajo esquemas jerárquicos o lineales,
- ☒ Evaluar decisiones dentro del grupo,

- ☑ Realizar votaciones dentro del grupo, siguiendo criterios diversos,

Para los efectos de esta investigación, se analizarán en orden alfabético a:

1. **Consensus @nyWARE® / QuestMap™:** Consiste en una suite de aplicaciones que apoya los procesos de toma de decisión y votaciones, dentro de empresas y organizaciones. Es producida por “The Soft Bicycle Company” [TSBC99]. Las herramientas computacionales que integran a este sistema permiten a sus usuarios la realización de tareas colaborativas, específicamente:

- ☑ Generación de ideas dentro del grupo, bajo esquemas jerárquicos o lineales,
- ☑ Evaluar y tomar decisiones dentro del grupo, a nivel de votaciones,
- ☑ Crear mapas de ideas.

Esta herramienta esta fundamentada en la metodología *IBIS*, según [CONK99] y [GDSS99]. A grandes rasgos, IBIS es una metodología que permite resolver “*problemas controversiales*”. De acuerdo con los documentos consultados, un problema se considera controversial si puede plantearse su solución a través de varias alternativas. Dichas alternativas se suponen como mutuamente contradictorias.

La metodología y el software en cuestión, se fundamentan en las premisas que planteamos a continuación:

- ☑ Toda discusión debe empezar en una pregunta,

- ☒ A dicha pregunta se le dan posibles soluciones a través de ideas,
- ☒ Dichas ideas son sustentadas con argumentos, a favor o en contra de la idea,
- ☒ Esta información puede ser representada y manipulada gráficamente.

2. **Group Support Systems®:** Es otra suite de aplicaciones que apoya los procesos de toma de decisión y votaciones, dentro de empresas y organizaciones. Es producida por “Ventana Corporation” [VENT99]. Group Support Systems® permite a sus usuarios realizar las siguientes tareas:

- ☒ Organización de ideas,
- ☒ Generación de tormentas de ideas,
- ☒ Crear y comentar listas de tópicos, bajo un esquema jerárquico,
- ☒ Generar y comentar tópicos bajo un esquema plano (contrario al esquema jerárquico, todos los tópicos y comentarios quedan al mismo nivel, como al redactar una carta personal),
- ☒ Evaluar y tomar decisiones dentro del grupo, al nivel de votaciones.

Según [WABR99], se puede fundamentar el uso de Group Support Systems®, en lo que denominan: *“Modelo del Aprendizaje Cognocitivista / Económico”* (Economic / Cognitive Model).

A grandes rasgos, este modelo consiste en enfocar las teorías cognocitivistas del aprendizaje desde la perspectiva que el objetivo del aprendizaje es: *“incrementar la probabilidad de que los aprendices y las personas a su alrededor sobrevivan y mejoren sus condiciones de vida, a través del incremento de su productividad”*.

El modelo que mencionamos es utilizado para fundamentar un estudio donde se concluye que: Group Support Systems® puede ser utilizado, con ventaja, como herramienta que facilita el proceso de enseñanza - aprendizaje, frente a las herramientas tradicionalmente utilizadas.

3. **TeamEC for Windows™:** Al igual que las anteriores, esta suite, apoya los procesos de toma de decisión y votaciones, dentro de empresas y organizaciones. Es producida por “Expert Choice, Inc” [EXCH99b]. TeamEC for Windows™, permite a los usuarios realizar las siguientes tareas:

- ☒ Recabar información acerca de enunciados que se evaluarán,
- ☒ Estructurar los enunciados que se evaluarán, asignándoles niveles de prioridad, definiendo las alternativas que se desprenden de cada enunciado,

- ☑ Asignar pesos (absolutos o relativos) a cada uno de los enunciados con sus correspondientes alternativas,
- ☑ Evaluar los enunciados con sus alternativas, para seleccionar la que sea más conveniente a los usuarios, dependiendo de los parámetros previamente definidos,
- ☑ Realizar análisis de sensibilidad y del tipo “*que pasaría sí...*”, bajo la modalidad de escenarios.

Según [EXCH99a], TeamEC for Windows™, se fundamenta en la metodología *AHP*. A grandes rasgos, AHP (Analytic Hierarchy Process) es una metodología que sistematiza el proceso de toma de decisiones y lo cuantifica. Para ello, plantea la ponderación relativa de todas las alternativas que se evalúan. Cada una de estas alternativas, es subdividida en otras que son ponderadas de manera similar, dentro de cada nodo. Con estos pesos, se calcula un estadígrafo de cada alternativa, que es utilizado para seleccionar la opción más acertada. Se considera que la decisión más adecuada es aquella que tiene el valor más grande en el estadígrafo calculado.

1.3.3.2. FOROS DE DISCUSIÓN.

En estas herramientas también se tiene una área de trabajo compartida donde cada usuario aporta al diálogo sus ideas u opiniones respecto al tema de conversación. El control de piso es de tipo FIFO, en caso que varios usuarios traten de intervenir en el diálogo simultáneamente. A la fecha, es normalmente aceptado que dentro de los foros de discusión, el diálogo se lleva a cabo siguiendo un patrón lineal en la conversación.

Al realizar esta investigación, resultó contrastante que, a diferencia de las herramientas de apoyo a los procesos de toma de decisión, los foros de discusión que se analizaron, carecen de un sustento teórico bien definido, en cuanto a teorías e investigaciones formales se refiere, a nivel de las compañías que los producen.

Esta situación podría explicarse en función de que: los productos en cuestión han sido concebidos como plataformas de trabajo de uso general, que pueden ser utilizadas de acuerdo a la conveniencia de los usuarios del sistema; en contraste con los sistemas de apoyo a los procesos de toma de decisiones en grupo, que tienen un objetivo bien definido.

Actualmente se encuentran muchas implementaciones de foros de discusión de este tipo, ya sea como programas separados, o como parte de sistemas colaborativos más grandes. Entre ellos tenemos:

1. **Imeet™, IServe™, Ishow™:** Con base a la información presentada en [SNEA99], se puede afirmar que iServe™, iMeet™ y iShow™, son aplicaciones de propósito general que permiten integrar a grupos de usuarios en un foro virtual. Es producida por “SneakerLabs, Inc”. iServe™, iMeet™ y iShow™, en conjunto, permiten a los usuarios realizar las siguientes tareas:

- ☒ Intercambio de ideas, preguntas y respuestas entre los participantes de una reunión virtual,
- ☒ Realizar conferencias y demostraciones de productos, en línea, a los participantes de una reunión virtual,

- ☒ Exponer presentaciones PowerPoint a los participantes de una reunión virtual.

Con base a la investigación que se realizó en la página Web de la empresa SneakerLabs, Inc., no existe una metodología definida que sustente el desarrollo de los productos iServe™, iMeet™ y iShow™, que pueda enmarcarse dentro de las teorías que sustentan los procesos de toma de decisiones (como es el caso de TeamEC for Windows con la metodología AHP).

2. Microsoft Exchange 2000 Conferencing Server: Microsoft Exchange 2000 Conferencing Server **[MICR00a]**: es un sistema colaborativo integrado por un conjunto de aplicaciones que permiten:

- ☒ Desarrollar reuniones basadas en datos, donde se puede compartir archivos, aplicaciones, se cuenta con una pizarra compartida, así como la capacidad de foro de discusión en línea,
- ☒ Desarrollar reuniones basadas en audio y video,
- ☒ Administrar los recursos asignados a cada reunión.

Al igual que en el caso de la empresa SneakerLabs, Inc., el sitio Web de Microsoft no presenta información acerca de alguna metodología definida que sustente el desarrollo del producto. La información disponible se ubica estrictamente a nivel de la arquitectura del producto, con el propósito de servir como base para el desarrollo de aplicaciones agregadas “Add-On’s” a este producto.

3. **Microsoft Windows NetMeeting:** Microsoft Windows NetMeeting, es otro sistema colaborativo que, representa un subconjunto del MS-Exchange 2000 Conferencing Server, ya que no posee herramientas para administración de reuniones.

La relevancia de Windows NetMeeting está en que es, probablemente, uno de los sistemas colaborativos más ampliamente distribuidos, puesto que se incluye dentro del paquete de distribución del sistema operativo MS-Windows, desde su versión 95. Este hecho lo coloca al alcance de todos los usuarios de este sistema operativo.

1.3.4. TECNOLOGÍAS QUE SE PUEDEN UTILIZAR PARA LA IMPLEMENTACIÓN DE HERRAMIENTAS QUE APOYEN LOS PROCESOS DE TOMA DE DECISIONES.

Actualmente existe toda una serie de tecnologías que permiten implementar herramientas computarizadas que apoyen a los procesos de toma de decisiones. A continuación, se presenta una breve descripción de algunas de estas, así como las características que pueden ser explotadas en dichos desarrollos.

1.3.4.1. INTERNET.

De acuerdo a lo que plantea [GUER98], el cada vez más explosivo fenómeno Internet, y su crecimiento casi exponencial, proveen un marco ideal para la explotación y desarrollo de aplicaciones colaborativas, dada su amplia difusión mundial

Uno de los aportes más grandes que puede hacer Internet a la sociedad está en el hecho de brindar la posibilidad para que pueden colaborar entre sí, grupos de usuarios distantes geográficamente, bajo los modelos de *“tele trabajo”* y *“tele enseñanza”*.

1. **La Tele Enseñanza:** De acuerdo a los conceptos que plantea [CIYD00], la tele enseñanza, como modelo, se puede caracterizar, básicamente en términos de que el alumno puede asistir a clases, conferencias, y demás actividades que se desarrollan dentro de las escuelas; sin necesidad de trasladarse al lugar físico donde se imparten las lecciones.

Bajo este modelo, el proceso de comunicación entre profesor y alumno — en su más amplio sentido— se logra mediante el empleo de herramientas de Groupware y telecomunicaciones, que facilitan el intercambio de información en formato de audio, vídeo y datos en general. Los sistemas se diseñan con el objetivo de permitir a los usuarios comunicarse entre ellos *“como si estuviesen en la misma habitación”*.

Para el profesor y sus alumnos la clase virtual es muy parecida a una clase ordinaria. Los profesores y los alumnos interactúan a través del lenguaje hablado o escrito, se *“ven”* unos a otros, usan pizarras etc. Una clase virtual no es más que una reproducción de una clase clásica, sin el inconveniente de tener que trasladarse físicamente a un punto geográfico predefinido.

2. **El Tele Trabajo:** De acuerdo a los conceptos que plantea [CIYD00], la primera vez que surge el término teletrabajo es en Estados Unidos en la década de los 70 durante la crisis del petróleo, con el objetivo de abaratar los costos de transporte. Es una forma de organizar el trabajo de manera que éste se realice fuera del lugar donde se ubica la empresa, con ayuda de las tecnologías de la información y las telecomunicaciones.

El teletrabajo introduce una ruptura de los antiguos esquemas de lugar y tiempo. Ahora el trabajador no tiene que acudir a un lugar determinado para llevar a cabo una actividad ni tiene que seguir un horario fijo. Impone una libertad mayor al trabajador y le brinda una serie de ventajas, entre las que se destacan:

- ☑ Más calidad de vida,
- ☑ Flexibilidad para organizar el tiempo libre,
- ☑ Menores costos de desplazamiento al lugar de trabajo, comidas, vestimenta necesaria para algunos trabajos, etc,
- ☑ Libertad para elegir donde vivir, independientemente del trabajo,
- ☑ Amplía el número de personas o empresas interesadas en su trabajo.

En contraste, el trabajador está obligado a tener un mayor compromiso con la empresa, pues su permanencia en el sistema esta condicionada únicamente por su productividad y eficiencia.

1.3.4.2. JAVA.

La aparición de Java como lenguaje de programación, comenzó en 1991, de forma contemporánea a la implementación de la WWW en Suiza por parte del CERN. En California, uno de los integrantes del equipo de desarrollo de Sun Microsystems, James Gosling, creó Java. Aunque no fue difundida la primera versión (alfa) hasta la primavera-verano de 1995. Este lenguaje de Sun ha

transformado sensiblemente la Web, y en poco tiempo ha dotado de inteligencia activa a las páginas estáticas que originalmente componían la Web **[FRAG00]**.

Dentro de las características fundamentales de diseño del lenguaje Java, destacan:

- ☑ Esta totalmente orientado a objetos,
- ☑ Tiene manejo automático de memoria,
- ☑ Es de ejecución multiplataforma,
- ☑ Soporta múltiples hilos de ejecución,
- ☑ Tiene características avanzadas de seguridad y confiabilidad.

En pocas palabras, es un lenguaje que es **independiente del “hardware”**, y rompe con los paradigmas de los lenguajes de programación tradicionales.

1.3.4.3. XML.

XML (eXtensible Markup Language) no es, como su nombre sugiere, un lenguaje de marcado. XML es un meta-lenguaje que permite definir lenguajes de marcado adecuados a usos determinados **[REIN00]**.

XML es un estándar internacionalmente reconocido, que no pertenece a ninguna compañía, y su utilización es libre. Sus ventajas fundamentales se ubican en términos de que permite separar, radicalmente, la información o el contenido de su presentación o formato. Además, fue diseñado para ser utilizado con cualquier lenguaje o alfabeto. Es decir, XML a nivel de manejo de información, se presenta

como una tecnología independiente del lenguaje de programación en que se implemente; de manera similar a como Java se presenta, como lenguaje de programación independiente del hardware.

Con XML se pueden desarrollar aplicaciones que intentan transferir una parte significativa de la carga de procesamiento de la información del servidor al cliente. Funcionará con un subprograma Java —por ejemplo—, que se ejecutará en el sistema del cliente, realizando algunas tareas que usualmente se encargarían al servidor del sistema. Esta característica hará que muchas de las funciones de consulta puedan desarrollarse desde el mismo navegador web o del sistema del cliente.

También se presta para desarrollar aplicaciones que precisen que el cliente web presente diferentes versiones de los mismos datos a diferentes usuarios. Por ejemplo:

1. Se puede implementar una aplicación colaborativa diseñada para que interactúen tanto profesores, como estudiantes y administrativos. Podría personalizarse con base al rol del usuario, presentado los datos bajo vistas distintas —el profesor puede modificar cualquier calificación de sus grupos, el estudiante solo conoce sus calificaciones personales sin derecho a modificarlas, etc.—.
2. Por otro lado, se podría aplicar en el diseño de un manual de diferentes grados (iniciación, intermedio y avanzado) con sus diferentes idiomas, y otras características personalizables. Esto hará que este manual se pueda personalizar según el tipo de usuario y permitirles, por ejemplo, extraer la información que necesitan de acuerdo a criterios de ordenación y formatos particulares.

En tal sentido, XML presenta características que pueden ser explotadas en el desarrollo de sistemas colaborativos y de apoyo a la toma de decisiones **[MICR00b]**.

1.3.4.4. DOM.

El Document Object Model (DOM) es una interfaz de programación de aplicaciones para documentos HTML y XML. El DOM define la estructura lógica de los documentos y la manera en que estos documentos son accedidos y manipulados **[W3C98]**.

El DOM no es un modelo específico de XML sino precisamente lo que convierte al HTML estático en dinámico. Se puede entender como la forma en la que los exploradores interpretan una página que por su naturaleza propia es estática (o desprovista de comportamientos programables), transformando sus elementos en objetos que, como tales, poseen propiedades, métodos y eventos, y que por lo tanto, se convierten en entidades programables **[POSA00]**.

Con el DOM, los programadores pueden crear documentos, navegar en su estructura y agregar, modificar o eliminar elementos y contenido. Cualquier cosa que se encuentre dentro de un documento HTML o XML, puede ser accedida, cambiada, eliminada o agregada utilizando el DOM, salvo unas pocas excepciones.

Lo más importante antes de trabajar con DOM es tener presente que todo el contenido de un documento se ve desde la jerarquía como un conjunto de nodos. ¿Por qué nodos en lugar de elementos? Por que una de las diferencias principales con la jerarquía DHTML es que allí encontrábamos colecciones que debían existir siempre, independientemente de que tengan o no elementos. Por ejemplo,

siempre encontramos una colección de imágenes, aunque la página no contenga una sola imagen.

En XML, la situación es más flexible: no existen colecciones predefinidas de objetos, por la naturaleza propia de la especificación. La única cosa que se conoce con certeza, es que habrá un objeto “*Element*”, que corresponderá con el nodo raíz de la jerarquía. Todo lo demás, dependerá del contenido del documento.

Ahora, dado que el DOM es el modelo que usualmente se asume, al desarrollar aplicaciones basadas en XML, se infiere que tiene algún grado de importancia al desarrollar este tipo de aplicaciones [MICR00b].

1.3.4.5. FRAMEWORKS QUE FACILITAN EL DESARROLLO DE APLICACIONES COLABORATIVAS.

Observando todos los recursos que hemos mencionado previamente, podría pensarse, erróneamente, que el desarrollo de sistemas colaborativos es usualmente una tarea fácil. Sin embargo, si se consideran solamente los recursos que se han mencionado hasta este momento, se debe pensar que la implementación de este tipo de aplicación se tiene que realizar desde cero, controlando todas las variables correspondientes a los problemas de seguridad, comunicación, control de concurrencia y diseño de interfaces, entre otros.

Es por esta razón que han surgido algunos intentos para desarrollar Frameworks o API's para el desarrollo especializado de este tipo de aplicación. A continuación, se mencionan algunas de ellas:

1. **La plataforma TOP:** La plataforma “*Ten Objects Platform*” -TOP- nace debido a la falta de herramientas para la rápida construcción y modificación de aplicaciones colaborativas docentes. Esta plataforma está compuesta

por diez objetos para la construcción de aplicaciones colaborativas, y un servidor que provee acceso a los métodos y atributos de estos objetos. Esta documentada en **[GUER98]**.

Esta plataforma se presenta como relativamente simple de utilizar, pues la interfaz de la aplicación se crea utilizando JavaScript y HTML, así como un Administrador de Plataforma; en tanto que utiliza un servidor predefinido. Es la que se presenta como más práctica, al momento de implementar aplicaciones.

2. **El proyecto Habanero 2.0:** El proyecto Habanero es desarrollado por la NCSA y la Universidad Urbana de Illinois. Habanero permite compartir objetos Java con otras aplicaciones distribuidos a través de Internet. Su enfoque principal esta puesto en extender aplicaciones Java monousuario a aplicaciones Java multiusuario, para lo que provee las facilidades de comunicación necesarias. Sin embargo, las aplicaciones Java deben ser construidas desde cero, utilizando la biblioteca de clases que ofrece el Habanero, así como las que suministra el Java que se esté utilizando **[NCSA00]**.
3. **El proyecto Infospheres:** Este proyecto es apoyado por un grupo de instituciones y empresas norteamericanas, entre las que destacan: la Oficina de Ciencias de la Fuerza Aérea, Novell, y la National Science Foundation, agrupadas bajo el Center for Research on Parallel Computation **[CRPC00]**.

De manera similar al proyecto Habanero, este proyecto desarrolla una API, acompañada de herramientas de desarrollo y depuración, con las que se pretende solventar el problema del desarrollo rápido de aplicaciones

colaborativas, utilizando el lenguaje de programación Java. Sin embargo, plantea los mismos problemas que adolece el proyecto Habanero.

1.4. METAS.

A continuación, se presenta una breve descripción de las metas que pretende alcanzar esta investigación.

1.4.1. ANTECEDENTES DEL PROBLEMA.

En [NÚÑE99] se propone un modelo de foro de discusión diferente a los comentados en esta investigación, pues esta basado en un modelo de hilos de discusión –es jerárquico– y, al mismo tiempo, permite apoyar los procesos de toma de decisiones en grupo, basándose en la teoría de plausibilidad de [AGÜE87]. Sin embargo, a la fecha, no se conocen implementaciones de foros de discusión en línea que se basen en estos conceptos simultáneamente.

Las metas que pretende alcanzar esta investigación se plasman en los objetivos que se desean alcanzar.

1.4.2. OBJETIVO GENERAL.

1. Implementar un prototipo de foro de discusión jerárquico, basado en XML y la teoría de plausibilidad, que facilite los procesos de toma de decisión grupales en Internet.

1.4.3. OBJETIVOS ESPECÍFICOS.

1. Analizar los requerimientos de plausibilidad que debe cumplir el prototipo de foro de discusión jerárquico, con base en la formalización planteada en **[NÚÑE99]**.
2. Diseñar el prototipo de foro de discusión jerárquico, de tal forma que el esquema de almacenamiento de información se implemente en XML.
3. Diseñar el prototipo de foro de discusión jerárquico, utilizando el paradigma de orientación a objetos.
4. Diseñar la arquitectura interna de las aplicaciones cliente y servidor del prototipo de foro de discusión jerárquico.
5. Programar el prototipo de foro de discusión jerárquico, implementándolo en el lenguaje de programación Java.

1.4.4. PRODUCTOS DE LA INVESTIGACIÓN.

El producto de esta investigación será una aplicación cliente / servidor, programada en Java, que implemente un prototipo de foro de discusión basado en consideraciones de la teoría de plausibilidad.

En tal sentido se debe recalcar que lo que se pretende es, fundamentalmente, implementar el foro de discusión que está formalizado en **[NÚÑE99]**.

1.4.5. JUSTIFICACIÓN.

Con base en la información recolectada al elaborar el marco teórico de este documento, se puede afirmar que no existen aplicaciones implementadas que realizan las tareas planteadas.

Se debe observar que, en las herramientas de apoyo a la toma de decisiones analizadas se pueden generar ideas bajo esquemas jerárquicos. Sin embargo, la votación se plantea como un proceso aparte.

Por otro lado, los foros de discusión analizados, tanto lineales como jerárquicos, no implementan esquemas de votación en línea, con respecto al desarrollo del diálogo.

En tal sentido, el foro que se propone implementar involucra al proceso de votación con el desenvolvimiento del diálogo jerárquico *—al dialogar bajo un esquema jerárquico, implícitamente se vota—*.

Actualmente, sólo se conoce la formalización matemática de esta herramienta, presentada por [NÚÑE99].

El producto final de esta investigación será un foro de discusión “operativo” que permita apoyar efectivamente a los procesos de toma de decisiones grupales, posibilitando la realización de un diálogo intergrupar jerárquico [HORN00].

1.4.6. ORIGINALIDAD E IMPACTO.

Esta investigación ofrece aportes que tienen un significativo impactante en las áreas de teletrabajo y teleenseñanza en términos de que posibilita la realización

de reuniones electrónicas y de procesos de toma de decisiones en grupo, desde una perspectiva novedosa: la jerarquización del diálogo, a la vez que se le realimenta con una medida cuantitativa de la aceptabilidad que tiene cada enunciado dentro del diálogo. Es importante recalcar que ambas áreas del conocimiento se encuentran enmarcadas dentro del denominador común que representan las tecnologías de telecomunicación modernas -léase Internet-.

Si se considera el impacto de esta investigación en el ámbito de la tele enseñanza, se ofrece una herramienta que, convenientemente utilizada, apoyará una de las actividades más importantes del ser humano, el aprender. La implementación de herramientas de este tipo, es un paso en dirección de hacer realidad el funcionamiento de las “*Escuelas Virtuales*”, tal y como las plantea [NEGR95].

En ellas, será posible salvar los problemas que representan las distancias y el tiempo, a través del uso atinado de las tecnologías de la información [NÚÑE99]. Bajo esta perspectiva, el aporte de la investigación se orienta específicamente a:

1. Controlar el problema representado por la distancia entre educandos y educador, al “reunirlos” en un ambiente donde pueden dialogar en tiempo real,
2. Facilitar la comprensión del diálogo entre educando y educadores, con la jerarquización de la conversación,
3. Cuantifica la aceptabilidad de cada enunciado que se expresa en el diálogo, con base a la Teoría de la Plausibilidad.

Por otro lado, si se considera el impacto de esta investigación en el ámbito del tele trabajo, se ofrece una herramienta que, convenientemente utilizada, apoyará significativamente los procesos de toma de decisiones en grupo y facilitará la

comprensión del diálogo entre colaboradores, sin importar distancias, al tiempo que ofrece una medida cuantitativa del grado de aceptación que presentan las ideas que son planteadas, al desarrollar el diálogo. Bajo esta perspectiva, el aporte de este trabajo se orienta específicamente a:

1. Controlar el problema representado por la distancia entre colaboradores, al “reunirlos” en un ambiente donde pueden dialogar en tiempo real,
2. Facilitar la comprensión del diálogo entre colaboradores, con la jerarquización de la conversación,
3. Cuantificar la aceptabilidad de cada enunciado que se expresa en el diálogo, con base a la Teoría de la Plausibilidad.

A la fecha, y de acuerdo al marco teórico de esta investigación, ninguna de las aplicaciones que se han analizado puede ofrecer, simultáneamente la posibilidad de satisfacer los tres aportes que se plantean, a nivel de la tele enseñanza y el tele trabajo.

1.4.7. DELIMITADORES.

Por la complejidad inherente a la implementación de este tipo de aplicación y dado que se pretende obtener un prototipo de aplicación:

1. El prototipo a desarrollar se centrará en implementar los aspectos fundamentales de la formalización planteada por [NÚÑE99]; no se hará énfasis en abarcar todos los detalles y variaciones que se pueden presentar dentro del problema implementado.

2. No se pretende realizar un proceso de evaluación formal de la efectividad de la herramienta implementada, en términos de los grupos de usuarios potenciales previamente enfocados.
3. No se realizarán pruebas de rendimiento formales para evaluar el desempeño y la eficiencia computacional de la herramienta.
4. Únicamente se validará el prototipo de aplicación contra su especificación, con el objeto de asegurar que ella satisface la especificación planteada.
5. A nivel de interfaz de usuario, no se pretende entregar una herramienta totalmente acabada. La interfaz de usuario del prototipo de aplicación proveerá al usuario de los controles básicos, que le permitan comunicarse con los motores del sistema.

Los procesos antes descritos, pueden ser planteados como algunos de los trabajos de investigación, que se pueden derivar de los resultados de esta investigación.

2. ESPECIFICACIÓN DEL FORO DE DISCUSIÓN JERÁRQUICO.

En este capítulo se presenta la especificación del *Foro de Discusión Jerárquico*. En tal sentido, se presentan y analizan:

1. El problema que se resolverá,
2. Las funciones del sistema,
3. Los casos de uso con sus diagramas,
4. El modelo conceptual,
5. Los diagramas de colaboración,
6. El diagrama de paquetes, finalmente.

El análisis que se plantea en este capítulo; está fundamentado, íntegramente, en la formalización del foro de discusión, presentada por **[NÚÑE99]**. Esta formalización puede ser consultada, dentro de esta investigación, en el “Anexo 1: formalización del foro de discusión jerárquico.”

En el análisis del Foro de Discusión Jerárquico se utilizará el lenguaje unificado de modelado *UML*, enfocado, en la mayoría de los casos, de acuerdo a como lo presenta **[LARM99]**. La decisión se fundamenta en que es la única referencia bibliográfica conocida en el medio local, que presenta una metodología de análisis y diseño de sistemas fundamentada, mayormente, en *UML*.

2.1. DESCRIPCIÓN GENERAL DEL PROBLEMA.

El “*Foro de Discusión Jerárquico*” o *FDJ*, es una herramienta que permite, a sus usuarios, debatir libremente sus ideas, siguiendo el esquema de hilos de discusión. Al mismo tiempo que los usuarios exponen sus ideas, el *FDJ* realiza el análisis del estado de plausibilidad de sus planteamientos, y lo presenta a los usuarios, en la medida que expresan sus opiniones.

2.2. TIPOS DE USUARIO.

El *FDJ* maneja dos roles para sus usuarios:

1. **Administrador:** Es la persona que debe iniciar y finalizar las sesiones del *FDJ*. Se encarga también de establecer el tema de discusión, las marcas de aceptación y rechazo, así como la fecha y la hora de las reuniones, y cualquier otra información adicional que los usuarios requieran para integrarse a la reunión. Finalmente, debe comunicar, a los usuarios del sistema, la información antes descrita. Durante el desarrollo de sesión, no tiene injerencia en su desenvolvimiento.
2. **Participante:** Es el grupo de usuarios que utiliza el *FDJ* para dialogar. En general, puede ser cualquier grupo de personas que requiere utilizar el diálogo como herramienta para encontrar solución a un problema cualquiera. El problema en cuestión, será el tema que discutirán en el transcurso de la sesión.

2.3. PROBLEMA QUE SE RESOLVERÁ.

En el *FDJ*, se brinda una área de trabajo compartida donde sus usuarios pueden intercambiar opiniones o enunciados. Estas opiniones se estructuran de manera jerárquica, de acuerdo al modelo de hilos de discusión [NÚÑE99].

Cada uno de los enunciados registrados dentro del diálogo, estará categorizado bajo una de las naturalezas permitidas². En consecuencia, el *FDJ* puede calcular el estado de plausibilidad de cualquiera de los enunciados que lo integran.

Las sesiones del *FDJ* se pueden describir, en términos generales, como relatamos a continuación:

1. El administrador del foro informa a los usuarios potenciales del foro, la realización de una reunión. Esencialmente les informa: la fecha, hora, tema a discutir, marcas y umbrales de aceptación y rechazo que se aplicarán en la reunión. Esta comunicación se realiza utilizando algún medio alternativo, por ejemplo, correo electrónico.
2. El administrador del foro inicia la sesión a la fecha y hora pre-establecidas, utilizando los parámetros comunicados a los usuarios potenciales.
3. Los usuarios se reúnen en el área de trabajo compartida a la hora previamente acordada.

² Las naturalezas que pueden asumir los enunciados dentro del foro son: *Aclaración (A)*, *Evidencia A Favor (F)*, *Evidencia En Contra (C)*, *Desactivado (D)* y *Opinión Primaria (O)*.

4. Un usuario cualquiera plantea alguna idea, opinión o problema a discutir. Esta idea es categorizada como *Opinión Primaria*. Esta *Opinión Primaria* es el origen de un *Hilo de discusión*.
5. Cualquiera de los participantes –inclusive el usuario que aportó la *Opinión Primaria*–, puede aportar elementos de juicio a dicho planteamiento. Estos elementos de juicio pueden categorizarse como *Evidencia A Favor*, *Evidencia En Contra*, o *Aclaración*; o
6. Plantear otras *Opiniones Primarias*, independientes de las que ya se hayan formulado; o
7. Modificar el texto o la naturaleza de cualquiera de los enunciados que ha incluido en el *FDJ*. La modificación de un enunciado significa que: se modifica el texto del enunciado, o se modifica la naturaleza del enunciado, o se desactiva / reactiva el enunciado, o el usuario se retracta del enunciado; o
8. Puede seleccionar una de las consultas permitidas³, que filtrará los enunciados del diálogo, mostrando solamente aquellos que corresponden a la consulta escogida.
9. Cada vez que un usuario agrega o modifica algún enunciado dentro del *FDJ*, se recalcula el estado de plausibilidad del hilo de discusión primario al que pertenece el enunciado modificado. Los valores resultantes de este

³ Estas consultas son: *Ver Todo*, *Ver Lo Leído*, *Ver Lo No Leído*, *Ver Opiniones Primarias*, *Ver Sólo enunciados F / C / A*, *Abrir / Cerrar Un Hilo De Discusión*.

cálculo son observados en cada enunciado que se altera, al momento en que se refresca la información de la consulta que realiza cada usuario del foro.

10. Cada vez que un nuevo usuario se integra al diálogo, el *FDJ* recalculará el estado de plausibilidad de todos los hilos de discusión primarios del diálogo registrado hasta ese momento. Los valores resultantes de este cálculo son observados en cada enunciado que se altera, al momento en que se refresca la información de la consulta que realiza cada usuario del foro.
11. Al final de la sesión de cada usuario, el *FDJ* almacenará en su sistema un documento, que contendrá la parte del diálogo que observaba al momento de abandonar el foro, de acuerdo a la consulta que realizaba en el momento que abandonó la sesión. Además, archivará una copia completa de todo el diálogo realizado, en el sistema donde se administra el *FDJ*.
12. El administrador del foro finaliza la sesión al concluir el debate, o al completarse el tiempo asignado para la reunión.

2.4. FUNCIONES.

Con base en la narración de las sesiones de trabajo, que se han descrito previamente, y en la formalización de [NÚÑE99], se pueden formular las funciones principales del *FDJ*, desde el punto de vista de sus usuarios potenciales. Se han redactado en una tabla, que se presenta a continuación.

#	Función	Visibilidad Frente Al Usuario
1.	Iniciar la sesión del Foro.	Evidente
2.	Registrar el ingreso de cada usuario a la sesión.	Evidente
3.	Registrar la salida de cada usuario de la sesión.	Evidente
4.	Registrar los enunciados que se agregan.	Evidente
5.	Registrar los enunciados que se modifican.	Evidente
6.	Calcular el estado de Plausibilidad de los hilos de discusión primarios del diálogo, cada vez que algún usuario ingresa a la sesión.	Evidente
7.	Calcular el estado de Plausibilidad del hilo de discusión primario en que algún usuario agrega o modifica enunciados.	Evidente
8.	Consultar el diálogo realizado, utilizando alguno de los modos de consulta válidos.	Evidente
9.	Almacenar persistentemente el diálogo desarrollado en la sesión.	Evidente
10.	Finalizar la sesión del Foro.	Evidente

Tabla 1: Funciones Principales del *FDJ*

2.5. CASOS DE USO.

En esta sección se presentan los casos de uso desarrollados para el *FDJ*; que están fundamentados en las funciones de la Tabla 1.

La ilustración que se presenta a continuación, contiene los casos de uso básicos que se han identificado.

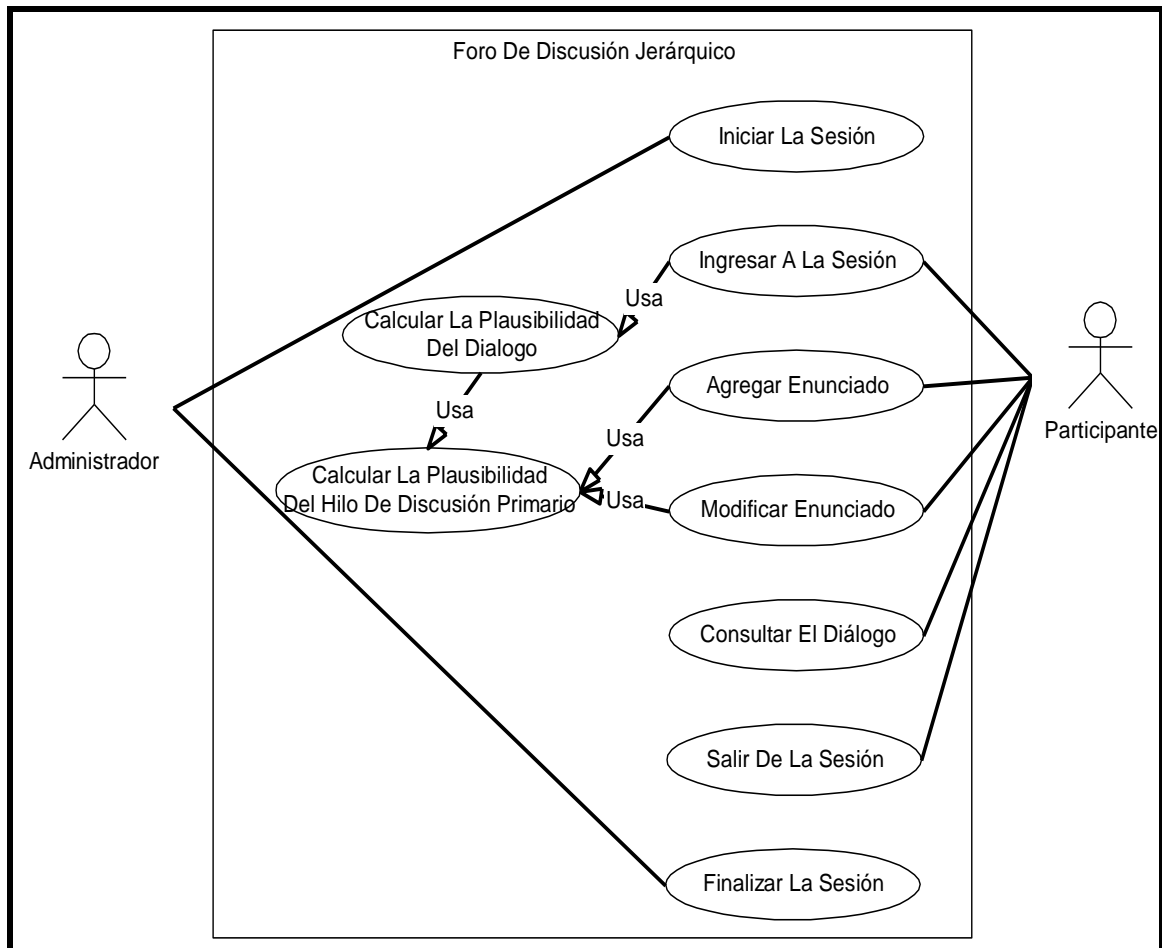


Ilustración 2: Casos de Uso Básicos del *FDJ*

A continuación, se analizan los casos de uso con más detalle.

2.5.1. CASO DE USO: INICIAR LA SESIÓN.

Secuencia Principal de Eventos: El caso de uso se inicia cuando el *FDJ* solicita al administrador del foro: el *tema de discusión* y las *marcas y umbrales de aceptación y rechazo* que se aplicarán a la sesión. Estos datos se almacenan en el sistema para su uso posterior; finalizando el caso de uso.

Secuencia Alternativa de Eventos: El administrador del foro cancela la captura del *tema de discusión y de las marcas y umbrales de aceptación y rechazo* que se aplicarán a la sesión.

La ilustración que se presenta a continuación, describe gráficamente al caso de uso en mención.

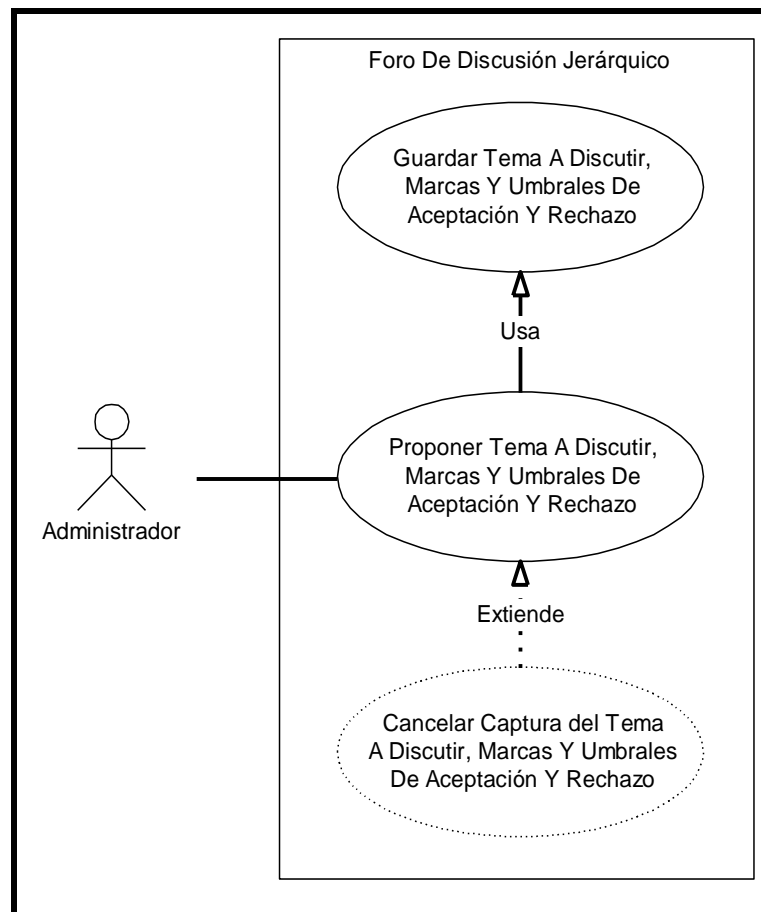


Ilustración 3: Caso de Uso "Iniciar la Sesión"

2.5.2. CASO DE USO: CALCULAR LA PLAUSIBILIDAD DEL HILO DE DISCUSIÓN PRIMARIO.

Secuencia Principal de Eventos: El caso de uso se inicia cuando se requiere calcular el estado de Plausibilidad de los enunciados que forman un hilo de discusión primario. Este cálculo se realiza de acuerdo a la secuencia de pasos señalada por el algoritmo DFS_Plausible. A continuación, el *FDJ* actualiza la consulta de cada uno de los participantes del foro, para que refleje la misma información que contiene el *FDJ*; finalizando el caso de uso.

2.5.3. CASO DE USO: CALCULAR LA PLAUSIBILIDAD DEL DIÁLOGO.

Secuencia Principal de Eventos: El caso de uso se inicia cuando el diálogo es alterado al ingresar algún participante. Inmediatamente se registra la alteración en el diálogo, el *FDJ* recalcula el estado de plausibilidad del diálogo, de acuerdo a lo señalado por el algoritmo DFS_Plausible. En consecuencia, aplica el caso de uso “*Calcular La Plausibilidad Del Hilo De Discusión Primario*”, a cada uno de los hilos de discusión primarios que integran el diálogo. A continuación, el *FDJ* actualiza la consulta de cada uno de los participantes del foro, para que refleje la misma información que contiene el *FDJ*; finalizando el caso de uso.

2.5.4. CASO DE USO: INGRESAR A LA SESIÓN.

Secuencia Principal de Eventos: El caso de uso se inicia cuando un participante intenta ingresar a la sesión del foro. Para tales efectos, proporciona el nombre de usuario con que se identificará. El *FDJ* comprueba que el nombre de usuario propuesto no es utilizado por ninguno de los participantes previamente admitidos

en la sesión, se reserva el nombre que propuso y le asigna la categoría “Activo”. A continuación se inicia el caso de uso “*Calcular La Plausibilidad Del Diálogo*”; finalizando el caso de uso.

Secuencia Alternativa de Eventos: El *FDJ* rechaza la adición del participante porque el nombre de usuario propuesto se asignó previamente. También se puede presentar que el participante cancele su inclusión en la sesión.

La ilustración que se presenta a continuación, describe gráficamente al caso de uso en mención.

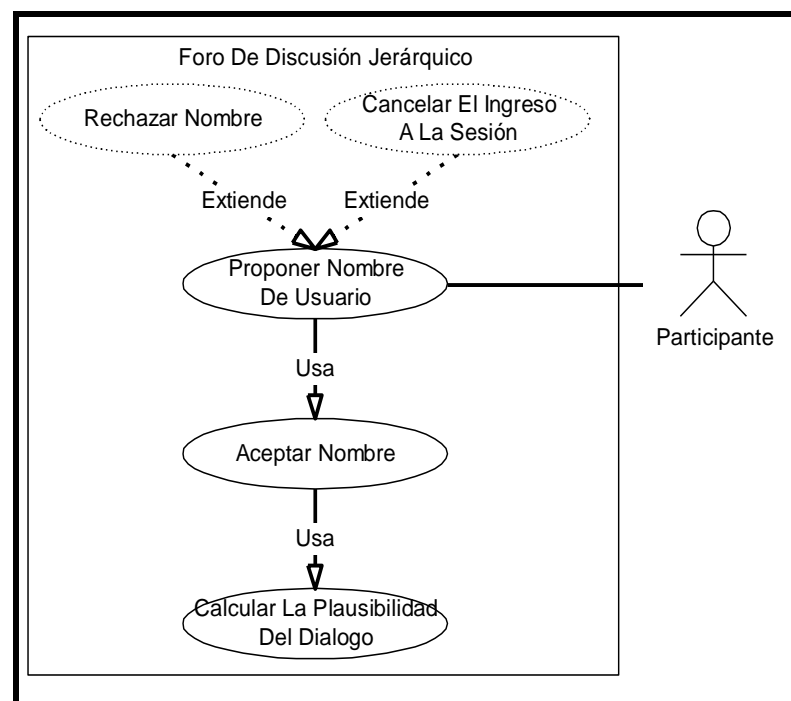


Ilustración 4: Caso de Uso “Ingresar a la Sesión”

2.5.5. CASO DE USO: AGREGAR ENUNCIADO.

Secuencia Principal de Eventos: El caso de uso se inicia cuando un participante intenta agregar un enunciado en el foro. En primer término, selecciona el

enunciado padre de su enunciado. A continuación, anota el texto y selecciona la naturaleza que se asignará a su enunciado. El *FDJ* agrega el enunciado al diálogo, como evidencia del enunciado padre. A continuación se inicia el caso de uso “*Calcular La Plausibilidad Del Hilo De Discusión Primario*”; finalizando el caso de uso.

Secuencia Alternativa de Eventos: El participante agrega una *Opinión Primaria*, en caso que el enunciado que agrega no tenga enunciado padre. En el caso que el participante señale el enunciado padre de su enunciado; podrá categorizarlo como: *Aclaración, Evidencia A Favor, o Evidencia en Contra*, del *enunciado padre*. También se puede presentar que el participante cancele la adición del enunciado.

La ilustración que se presenta a continuación, describe gráficamente al caso de uso en mención.

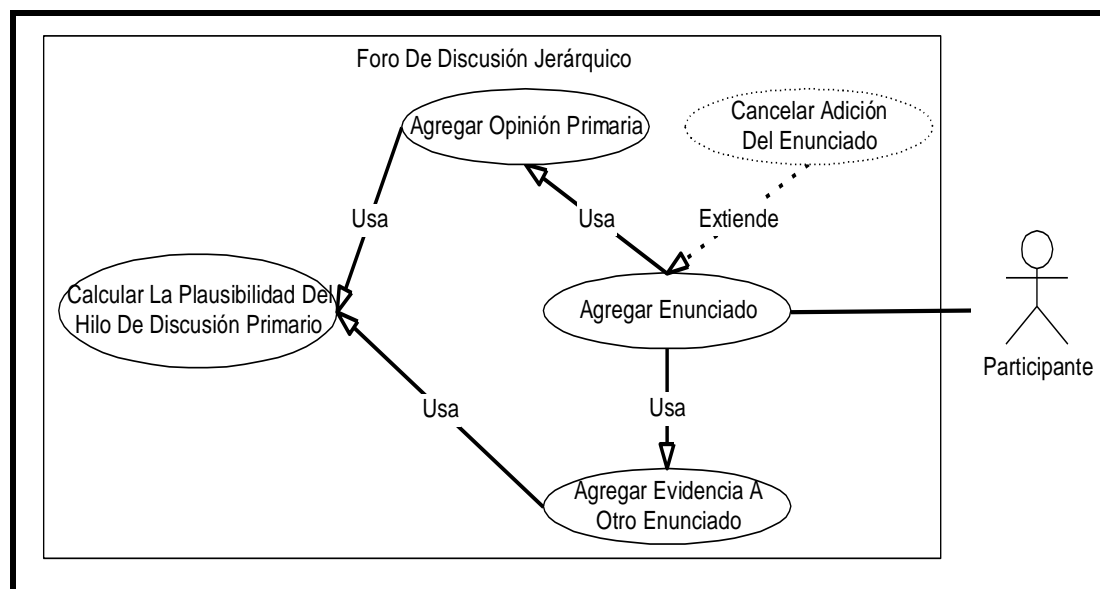


Ilustración 5: Caso de Uso “Agregar Enunciado”

2.5.6. CASO DE USO: MODIFICAR ENUNCIADO.

Secuencia Principal de Eventos: El caso de uso se inicia cuando un participante intenta modificar un enunciado en el foro. En primer término, selecciona el enunciado que desea modificar. En caso que el participante tenga autorización para cambiar el enunciado, se alterará el diálogo del foro de acuerdo al cambio que el usuario solicite; a saber: *Modificar el texto o la naturaleza del enunciado, desactivar / reactivar el enunciado; o retractarse del enunciado*. Salvo en el caso que se altera únicamente el texto del enunciado, la modificación afecta a los enunciados que forman parte de la evidencia asociada al enunciado alterado. A continuación se inicia el caso de uso “*Calcular La Plausibilidad Del Hilo De Discusión Primario*”; finalizando el caso de uso.

Secuencia Alternativa de Eventos: Si el participante no tiene autorización para alterar al enunciado, se le informa al participante que no tiene autorización para modificar ese enunciado. El diálogo no se altera. También se puede presentar que el participante cancele la adición del enunciado.

La ilustración que se presenta a continuación, describe gráficamente al caso de uso en mención.

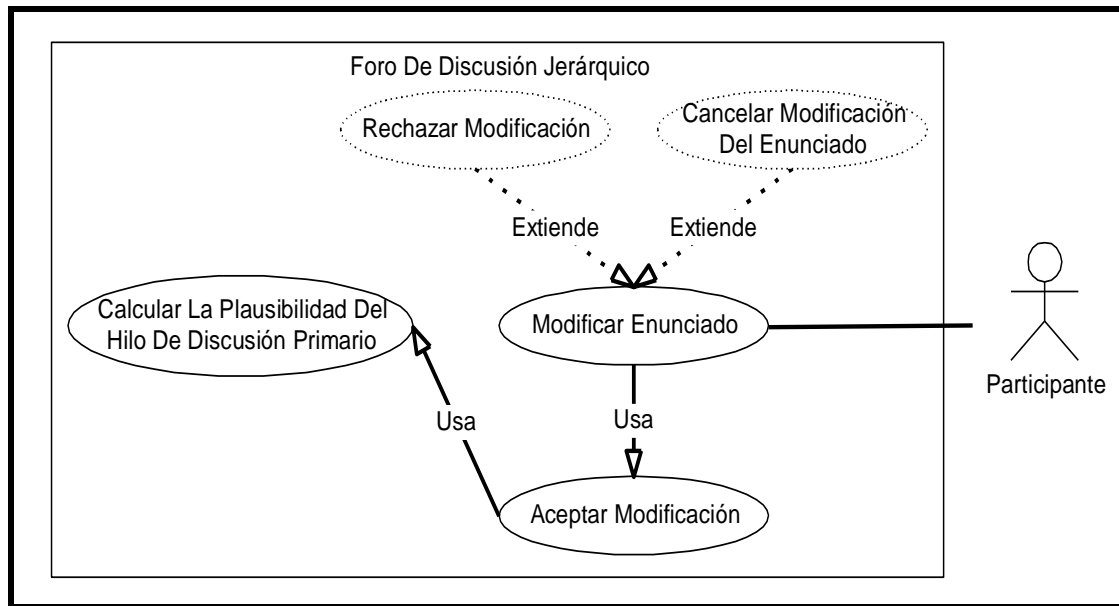


Ilustración 6: Caso de Uso "Modificar Enunciado"

2.5.7. CASO DE USO: CONSULTAR EL DIÁLOGO.

Secuencia Principal de Eventos: El caso de uso se inicia cuando un participante cambia el modo de consulta que utiliza para observar el diálogo. El participante selecciona el modo de consulta deseado. A continuación el *FDJ* le responde presentado los enunciados que responden a la consulta solicitada por él; finalizando el caso de uso.

La ilustración que se presenta a continuación, describe gráficamente al caso de uso en mención.

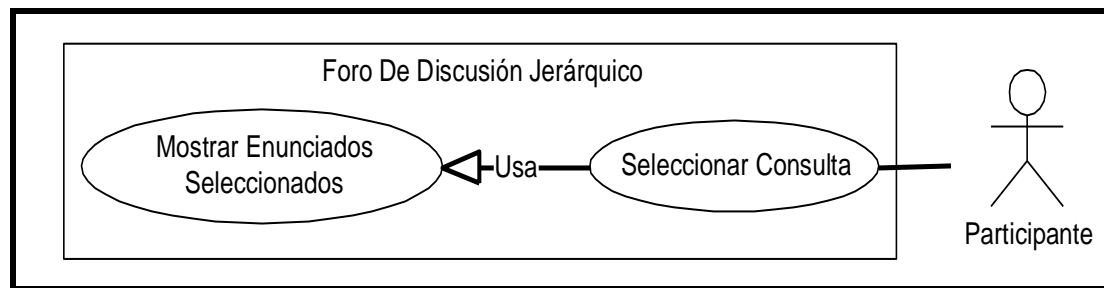


Ilustración 7: Caso de Uso “Consultar el Diálogo”

2.5.8. CASO DE USO: SALIR DE LA SESIÓN.

Secuencia Principal de Eventos: El caso de uso se inicia cuando uno de los participantes se retira de la sesión. El *FDJ* lo categoriza como usuario “*Inactivo*” y archiva la vista del diálogo que observa dicho usuario; finalizando el caso de uso.

Secuencia Alternativa de Eventos: Se puede presentar que el participante cancele su salida de la sesión.

La ilustración que se presenta a continuación, describe gráficamente al caso de uso en mención.

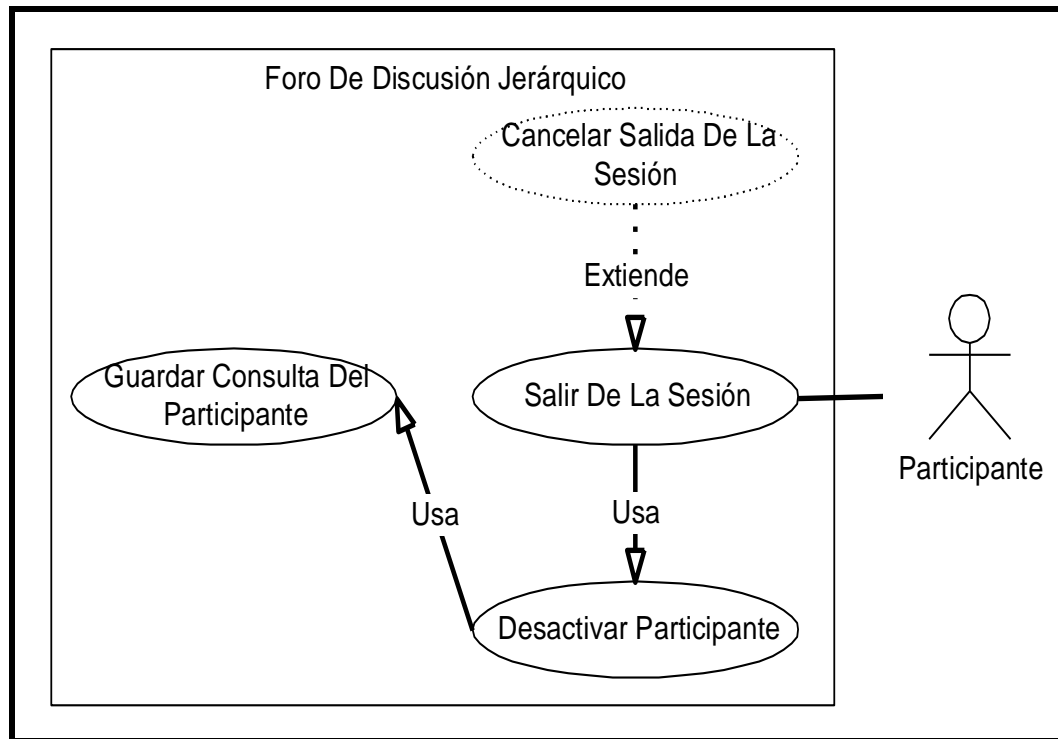


Ilustración 8: Caso de Uso “Salir de la Sesión”

2.5.9. CASO DE USO: FINALIZAR SESIÓN.

Secuencia Principal de Eventos: El caso de uso se inicia cuando el administrador del *FDJ* finaliza la sesión. El *FDJ* categoriza a los usuarios que permanezcan en el diálogo como “*Inactivo*” y archiva el diálogo que se ha realizado hasta ese momento; finalizando el caso de uso.

Secuencia Alternativa de Eventos: Se puede presentar que el administrador del *FDJ* cancele la finalización de la sesión.

La ilustración que se presenta a continuación, describe gráficamente al caso de uso en mención.

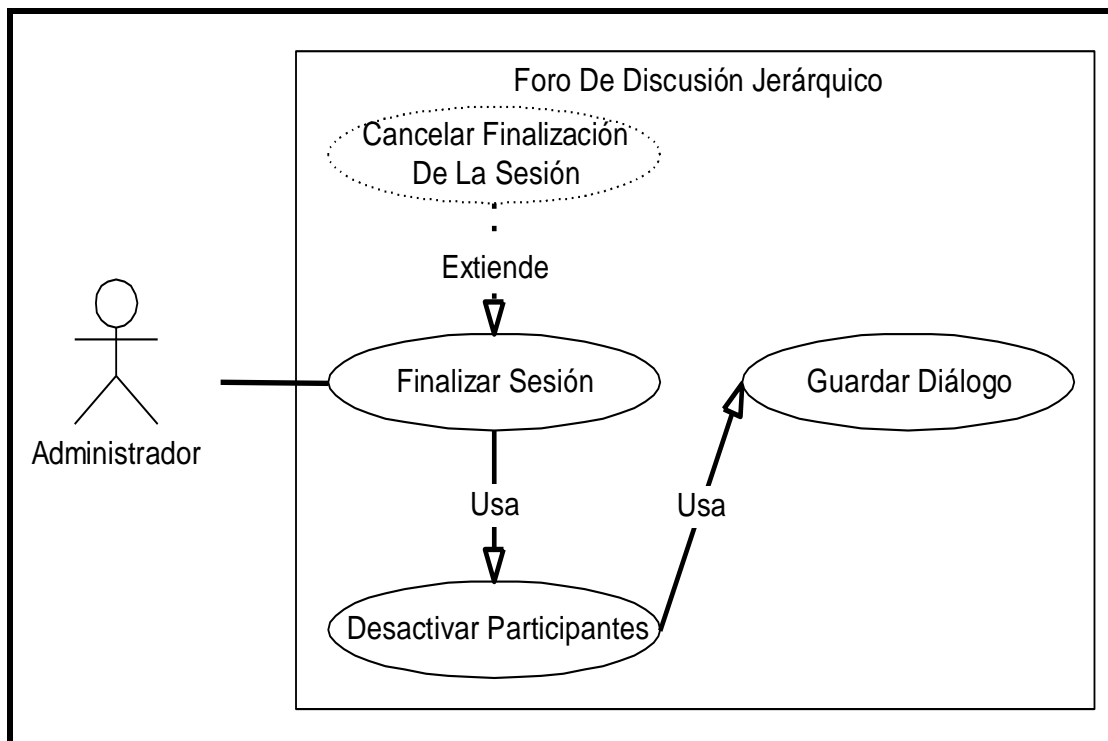


Ilustración 9: Caso de Uso “Finalizar Sesión”

2.6. MODELO CONCEPTUAL.

En esta sección se analizará el modelo conceptual del *FDJ*, que se presenta en la Ilustración que se muestra a continuación:

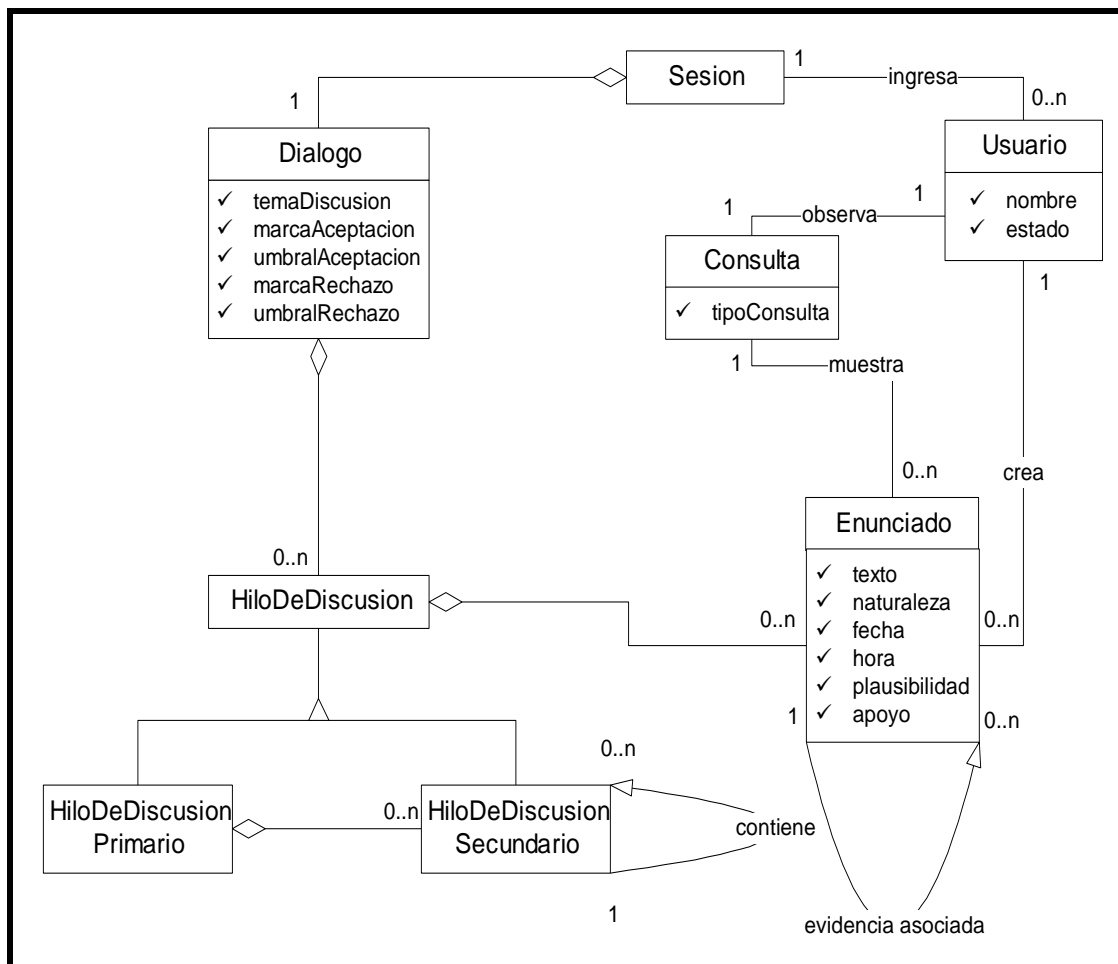


Ilustración 10: Modelo Conceptual del *FDJ*

Las clases del Modelo Conceptual, se explican a continuación, en una tabla.

Clase	Explicación
Usuario	Identifica a los participantes del diálogo.
Sesion	Representa la reunión que realizan los usuarios del foro. Está constituida por los usuarios que ingresan al foro para dialogar; así como por el diálogo que ellos realizan.
Consulta	Es un subconjunto de enunciados del diálogo, que observa un usuario particular.
Dialogo	Representa a los enunciados expresados por los usuarios, organizados de acuerdo al esquema de hilos de discusión.
Enunciado	Opiniones categorizadas que expresan los usuarios con respecto al tema de discusión.
Hilo De Discusion	Es la representación jerárquica del diálogo desarrollado con respecto a un enunciado, que se toma como raíz de la discusión.
Hilo De Discusion Primario	Es un hilo de discusión que tiene como raíz a una opinión primaria.
Hilo De Discusion Secundario	Es un hilo de discusión que tiene como raíz a un enunciado que no es opinión primaria.

Tabla 2: Clases del Modelo Conceptual del FDJ

Los atributos del Modelo Conceptual, se explican a continuación, en una tabla.

Clase	Atributo	Explicación
Usuario	Nombre	Es el nombre o pseudónimo que adopta el usuario durante la sesión.
	Estado	“Activo” si el usuario está presente en la sesión. “Inactivo” si el usuario se retiró de la sesión.
Consulta	tipoConsulta	Identifica cada una de las consultas definidas dentro del sistema: <i>Ver todo, Ver lo leído, Ver lo no leído, Ver Enunciados O, Ver Enunciados F, Ver Enunciado C, Ver Enunciados A, Abrir Un hilo de Discusión, Cerrar Un hilo de discusión.</i>

Clase	Atributo	Explicación
Diálogo	temaDiscusión	Es el tema que se discutirá a durante la sesión.
	marcaAceptacion	Utilizada para el Análisis de Plausibilidad. Es un número entre cero y uno, sin incluir los extremos.
	umbralAceptacion	Utilizada para el Análisis de Plausibilidad. Es un número entre cero y la marca de aceptación, sin incluir los extremos.
	marcaRechazo	Utilizada para el Análisis de Plausibilidad. Es un número entre cero y uno, sin incluir los extremos.
	umbralRechazo	Utilizada para el Análisis de Plausibilidad. Es un número entre cero y la marca de rechazo, sin incluir los extremos.
Enunciado	texto	Es la afirmación escrita que constituye al enunciado.
	naturaleza	Es una categorización subjetiva o producto del Análisis de Plausibilidad, sobre su enunciado padre (si tiene). Puede ser <i>Opinión Primaria (O)</i> , <i>Aclaración (A)</i> , <i>Evidencia A Favor (F)</i> , <i>Evidencia En Contra (C)</i> , <i>Desactivado (D)</i> .
	fecha	Es la fecha en que se redactó el enunciado.
	Hora	Es la hora en que se redactó el enunciado.
	plausibilidad	Es una medida del grado de aceptación que tiene un enunciado dentro de la sesión. Puede ser: <i>Aceptable</i> , <i>Satisfactorio</i> , <i>Indeterminado</i> , <i>Dudoso</i> , <i>Insatisfactorio</i> . Se calcula con base en el apoyo que recibe el enunciado, de su evidencia asociada.
	apoyo	Es una medida del grado de aceptación de la naturaleza de un enunciado dado, calculado con base en la evidencia a favor y la evidencia en contra de dicho enunciado. Es un número entre 0 y 1, sin incluir los extremos.

Tabla 3: Atributos del Modelo Conceptual del FDJ

Es importante recalcar que aquellos objetos que no tienen atributos dentro del modelo conceptual, no aparecen dentro de la tabla. Además, no se han planteado consideraciones acerca de los tipos de datos involucrados, así como del alcance

de los atributos. Estas decisiones serán tomadas en la etapa de diseño del sistema.

Existen algunos hechos que se deben destacar al comparar la formalización del *FDJ*, frente al modelo conceptual que se presenta.

1. La formalización del *FDJ* incluye al *identificador del usuario* dentro de los atributos del enunciado. Además, identifica al enunciado a través de una llave compuesta por el *nombre del usuario* más el *identificador del usuario* más la *fecha y hora* en que se escribió el enunciado. En términos del modelo conceptual que se presenta, este identificador se representa como:

Usuario.nombre + Usuario.idEnunciado + Enunciado.fecha +
Enunciado.hora

Este planteamiento, a nivel del modelo conceptual se presenta como prematuro, porque establece consideraciones de diseño, a nivel de la especificación.

En consecuencia se ha decidido que en este nivel el enunciado no tenga un atributo que lo identifique. Se pospone a la fase de diseño del sistema, la definición concreta de este identificador. Esta vinculación entre objetos se representará, en este nivel, a través de las asociaciones presentes en el Modelo Conceptual **[LARM99]**.

2. Las consideraciones anteriores provocan que el atributo *evidencia* no se presente dentro de la clase Enunciado. Igual situación se presenta con el atributo *idUsuario*, de la clase Usuario.

3. Interpretando la semántica de la formalización del *FDJ*, se entiende que *“los usuarios pueden ingresar y salir dinámicamente del foro, mientras el diálogo se desarrolla”*, y que *“los enunciados que redacta un usuario que se retira de la sesión, no desaparecen del diálogo”*. En consecuencia, se amplió el modelo, considerando un atributo de *estado* para cada usuario. En él se controla si el usuario está presente en el diálogo –en este caso se asigna el valor *“Activo”*–, o si se ha retirado del diálogo –en este caso se asigna el valor *“Inactivo”*. De este modo, se puede relacionar a los usuarios con los enunciados que han redactado, aunque el usuario se halla retirado de la sesión.
4. El administrador del foro no aparece representado dentro de las clases del modelo conceptual porque no tiene injerencia en el desarrollo de la sesión. Sólo se encarga de iniciar y finalizar la sesión.

2.7. DIAGRAMAS DE COLABORACIÓN.

En esta sección se presentan los diagramas de colaboración correspondientes a cada caso de uso desarrollado previamente.

2.7.1. COLABORACIÓN: INICIAR LA SESIÓN.

La colaboración se inicia cuando se suministra, a una instancia de la clase *Dialogo*, el tema de discusión y las marcas y umbrales de aceptación y rechazo que se aplicarán a la sesión. A continuación, se crean instancias para:

- ☒ La clase *Sesión*, que albergará a los clientes y el diálogo que desarrollarán,

- ☑ La clase *Usuario*, que albergará a los usuarios que se registren en la sesión,
- ☑ La clase *Consulta*, que albergará las consultas que realizarán los usuarios, al diálogo; finalizando la colaboración.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención.

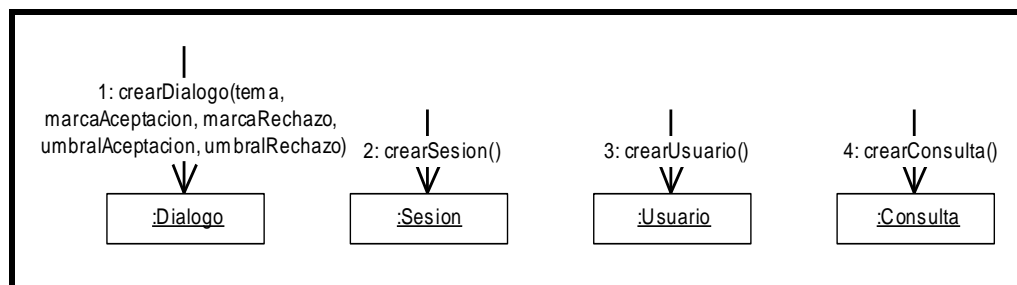


Ilustración 11: Colaboración “Iniciar la Sesión”

2.7.2. COLABORACIÓN: CALCULAR LA PLAUSIBILIDAD DEL HILO DE DISCUSIÓN PRIMARIO.

La colaboración se inicia cuando se suministra, a la instancia de la clase *HiloDeDiscusiónPrimario*, el hilo de discusión al que se desea calcular su estado de plausibilidad. Este cálculo se realiza de acuerdo a la secuencia de pasos señalada por el algoritmo DFS_Plausable. Luego, se actualiza a todas las consultas registradas en la instancia de la clase *Consulta*; finalizando la colaboración.

Es importante destacar que: al analizar la formalización del *FDJ* que plantea [NÚÑE99], en todas las consultas que pueden realizar los usuarios del foro, siempre están presentes, como mínimo, los enunciados que tienen naturaleza

“*Opinión Primaria*”; es decir, las raíces de los hilos de discusión. Otros enunciados pueden o no estar presentes, dependiendo de varios factores que serán analizados en su momento.

También se debe recalcar que la actualización de consultas se realizará únicamente a los usuarios que están “*activos*”. Cuando un usuario se retira del foro, se elimina su consulta.

Por esa razón, es que la colaboración plantea “tajantemente” que se deben actualizar las consultas de todos los usuarios, con los enunciados alterados o agregados al hilo de discusión primario, en mención; pues, al menos, se deberá actualizar la Opinión Primaria correspondiente al hilo de discusión suministrado.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención.

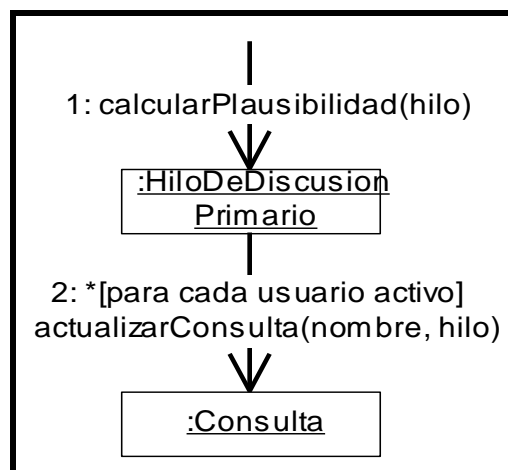


Ilustración 12: Colaboración “Calcular la Plausibilidad del Hilo de Discusión Primario”

2.7.3. COLABORACIÓN: CALCULAR LA PLAUSIBILIDAD DEL DIÁLOGO.

La colaboración se inicia cuando se suministra, a la instancia de la clase *HiloDeDiscusiónPrimario*, cada uno de los hilos de discusión que integran el diálogo, para calcular su estado de plausibilidad. Luego se inicia la colaboración “Calcular La Plausibilidad Del Hilo De Discusión Primario”; finalizando la colaboración.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención con más detalle.

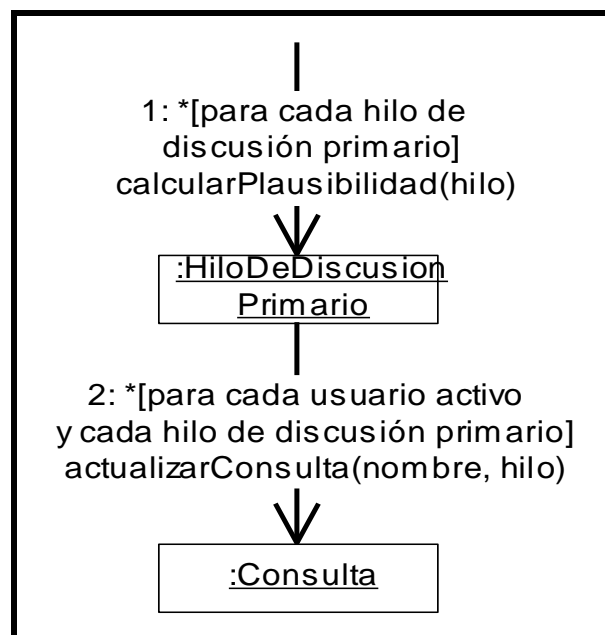


Ilustración 13: Colaboración “Calcular la Plausibilidad del Diálogo”

2.7.4. COLABORACIÓN: INGRESAR A LA SESIÓN

La colaboración se inicia cuando se suministra, a instancia de la clase *Usuario*, el nombre que propone un participante que desea ingresar a la sesión. En la instancia de la clase *Usuario* se comprueba que el nombre de usuario propuesto no es utilizado por ninguno de los participantes previamente admitidos en la sesión. En el caso que el nombre propuesto esté en uso, se rechaza la adición del usuario. En el caso que el nombre propuesto se acepte, se reserva el nombre propuesto y le asigna la categoría “Activo”.

A continuación, se inicia la colaboración “Calcular la Plausibilidad del Diálogo”; finalizando la colaboración.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención con más detalle.

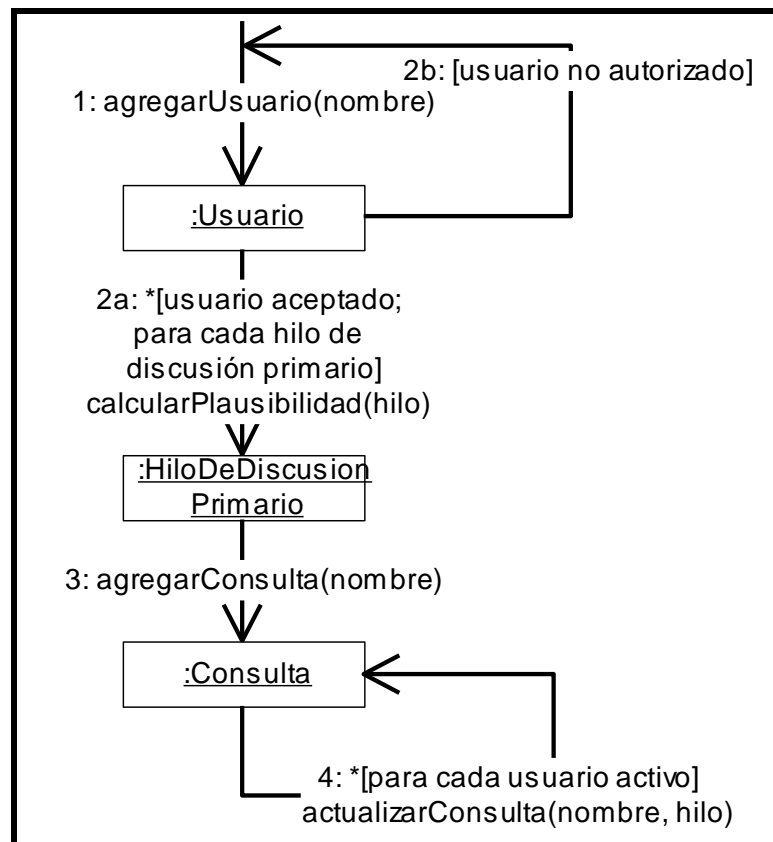


Ilustración 14: Colaboración “Ingresar a la Sesión”

2.7.5. COLABORACIÓN: AGREGAR ENUNCIADO.

La colaboración se inicia cuando se suministra, a la instancia de la clase *Enunciado*, el mensaje de que agregue un enunciado. En consecuencia se proporciona a la instancia de clase: el *nombre* del usuario que agrega el enunciado, el *texto* que integra el enunciado, la *naturaleza* que tendrá el enunciado, y el *enunciado Padre* del enunciado que agregamos.

En la instancia de la clase *Enunciado* se comprueba la consistencia de los datos proporcionados. En caso que exista alguna inconsistencia en los datos proporcionados, se rechaza la inclusión del enunciado; en tal caso se permite que se corrija la inconsistencia en los datos y se envíe el mensaje nuevamente. En el

caso que los datos proporcionados sean consistentes, se agrega el enunciado propuesto. El enunciado agregado formará parte de la evidencia asociada del enunciado padre.

A continuación, se inicia la colaboración “Calcular la Plausibilidad del Hilo de Discusión Primario”; finalizando la colaboración.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención con más detalle.

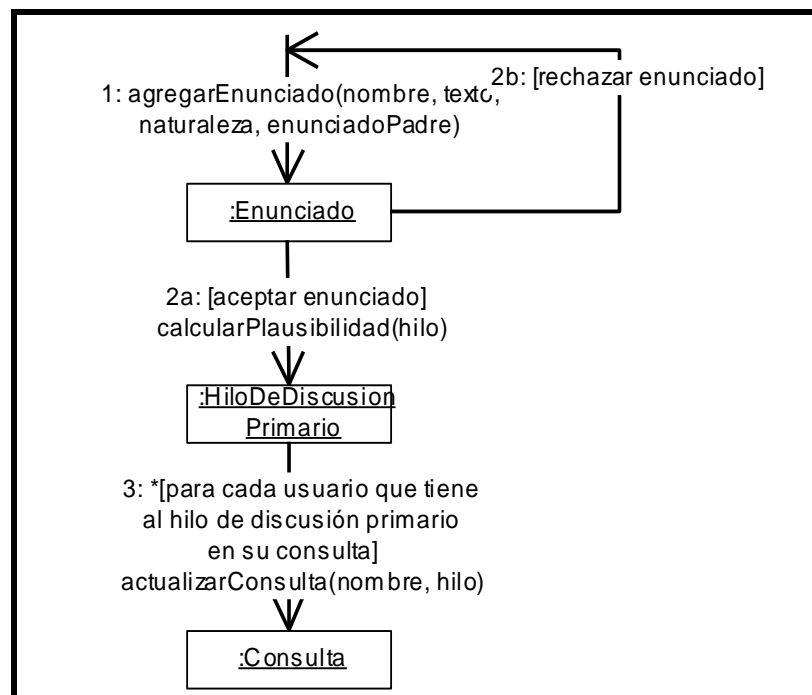


Ilustración 15: Colaboración “Agregar Enunciado”

2.7.6. COLABORACIÓN: MODIFICAR ENUNCIADO.

La colaboración se inicia cuando se suministra, a instancia de la clase *Enunciado*, el mensaje que modifique un enunciado. En consecuencia se proporciona a la instancia de clase: el *nombre* del usuario que modifica el enunciado, el *texto*

propuesto para el enunciado, la *naturaleza* propuesta para el enunciado, así como las variables *reactivar* y *retractar*, y el enunciado que se modifica *enun*. En las variables *reactivar* y *retractar* el usuario indica si se está retractando del enunciado o si lo está reactivando / desactivando.

En la instancia de la clase *Enunciado* se comprueba la consistencia de los datos proporcionados. En caso que exista alguna inconsistencia en los datos proporcionados, se rechaza la modificación del enunciado; en tal caso se permite que se corrija la inconsistencia en los datos y se envíe el mensaje nuevamente. En el caso que los datos proporcionados sean consistentes, se modifica el enunciado propuesto de acuerdo a los parámetros proporcionados.

A continuación, se inicia la colaboración “Calcular la Plausibilidad del Hilo de Discusión Primario”; finalizando la colaboración.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención con más detalle.

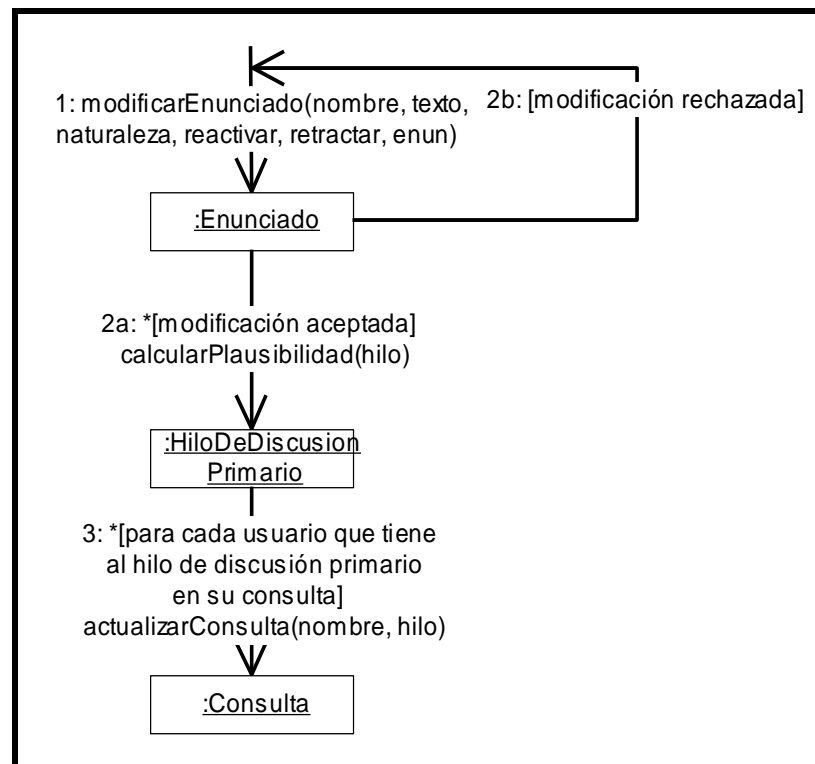


Ilustración 16: Colaboración “Modificar Enunciado”

2.7.7. COLABORACIÓN: CONSULTAR EL DIÁLOGO.

La colaboración se inicia cuando se suministra, a instancia de la clase *Consulta*, el mensaje que modifique el tipo de consulta que realiza un usuario. En consecuencia se proporciona a la instancia de clase: el *nombre* del usuario al que se le modificará su consulta, así como el tipo de consulta que se le asignará (*tipoConsulta*). La colaboración también se puede iniciar suministrando el mensaje que se agregue evidencia asociada a la consulta de un usuario. En este último caso, se proporciona el *nombre* del usuario al que se modificará su consulta, así como el *enunciado* al que se incluirá la evidencia asociada.

A continuación, la instancia de clase *Consulta* actualiza la consulta del usuario con los enunciados correspondientes a su solicitud; finalizando la colaboración.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención.

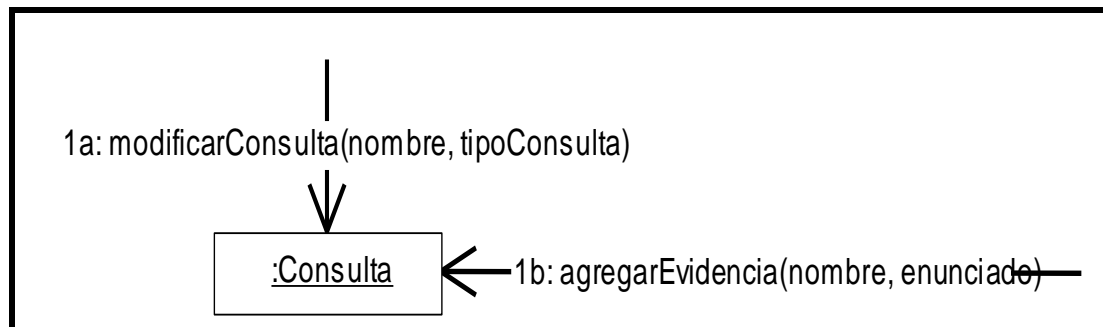


Ilustración 17: Colaboración “Consultar el Diálogo”

2.7.8. COLABORACIÓN: SALIR DE LA SESIÓN.

La colaboración se inicia cuando se suministra, a instancia de la clase *Usuario*, el mensaje que “elimine” a un usuario. En consecuencia se proporciona a la instancia de clase: el *nombre* del usuario que será eliminado (desea abandonar la sesión).

A continuación, la instancia de clase *Usuario* asigna el estado “Inactivo”, al usuario que se elimina. Luego, se envía a la instancia de clase *Consulta*, el mensaje que guarde la consulta del usuario desactivado, en un medio de almacenamiento persistente. Finalmente la instancia de la clase *Consulta* elimina la consulta que corresponde al usuario desactivado; finalizando la colaboración.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención.

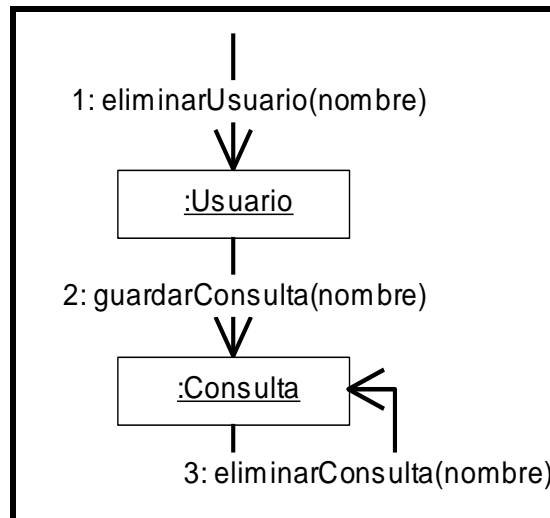


Ilustración 18: Colaboración "Salir de la Sesión"

2.7.9. COLABORACIÓN: FINALIZAR SESIÓN.

La colaboración se inicia cuando se suministra, a instancia de la clase *Dialogo*, el mensaje que guarde el diálogo desarrollado.

A continuación, se le envía un mensaje a la instancia de clase *Usuario* para que desactive a todos los usuarios activos en la sesión. Luego, se envía a la instancia de clase *Consulta*, el mensaje que elimine todas las consultas existentes. Después se envía un mensaje a la instancia de Clase *Sesión*, para que finalice la sesión. Finalmente, se envía un mensaje a la instancia de clase *Dialogo* para que elimine el diálogo existente; finalizando la colaboración.

La ilustración que se presenta a continuación, describe gráficamente la colaboración en mención.

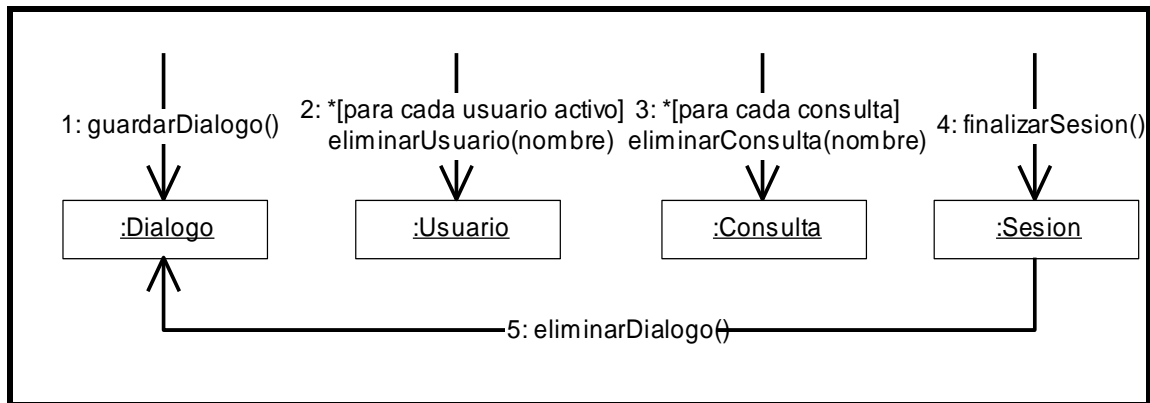


Ilustración 19: Colaboración "Finalizar Sesión"

2.8. ADICIÓN DE MÉTODOS AL MODELO CONCEPTUAL.

Con base en el modelo conceptual y los diagramas de colaboración previamente analizados, se puede ampliar el modelo conceptual del *FDJ*. Esta ampliación será discutida en esta sección.

Los métodos del Modelo Conceptual, se presentan a continuación, explicados en términos de una tabla.

Clase	Método	Acción
Consulta	crearConsulta()	Declara la instancia de la clase.
	actualizarConsulta(nombre, hilo)	Reemplaza, agrega o sobrescribe enunciados en un <i>hilo</i> de discusión de la consulta de un usuario <i>-nombre-</i> dado, para reflejar las alteraciones que se presentan en el desarrollo de la sesión.
	agregarConsulta(nombre)	Crea una consulta para el usuario <i>nombre</i> . Esta consulta inicialmente contiene los enunciados correspondientes a la vista predeterminada que debe observar un usuario al ingresar al foro ⁴ .
	modificarConsulta(nombre, tipoConsulta)	Cambia la modalidad de consulta que realiza el usuario <i>nombre</i> , a la señalada por <i>tipoConsulta</i> . En la misma acción se reemplazan los enunciados de la consulta del usuario, por los que corresponden al tipo de consulta recién asignado.
	agregarEvidencia(nombre, enunciado)	Agrega los enunciados que forman la evidencia asociada de <i>enunciado</i> a la consulta del usuario <i>nombre</i> .
	guardarConsulta(nombre)	Guarda los enunciados que integran la consulta del usuario <i>nombre</i> .
	eliminarConsulta(nombre)	Elimina la consulta del usuario <i>nombre</i> .
Usuario	crearUsuario()	Declara la instancia de la clase.
	agregarUsuario(nombre)	Incluye el <i>nombre</i> de un usuario nuevo dentro de la sesión; además, le asigna el estado “Activo”.
	eliminarUsuario(nombre)	Asigna el estado “Inactivo” al usuario <i>nombre</i> .
Sesion	crearSesion()	Declara la instancia de la clase.
	finalizarSesion()	Elimina la sesión del <i>FDJ</i> .
HiloDeDiscusión	calcularPlausibilidad(hilo)	Calcula el estado de plausibilidad de un <i>hilo</i> de discusión dado, del dialogo.

Clase	Método	Acción
Enunciado	agregarEnunciado(nom, tex, nat, eP)	Agrega el texto <i>-tex-</i> , de naturaleza <i>-nat-</i> , expresado por el usuario <i>-nom-</i> , como un enunciado que forma parte de la evidencia asociada del enunciado <i>-eP-</i> .
	modificarEnunciado(nom, tex, nat, reac, rect, enun)	Reemplaza el texto <i>-tex-</i> , y la naturaleza <i>-nat-</i> , expresado por el usuario <i>-nom-</i> , en el enunciado <i>-enun-</i> , señalando si el usuario reactiva / desactiva <i>-rec-</i> o retracta <i>-rect-</i> el enunciado.
Dialogo	crearDialogo(te, mA, mR, uA, uR)	Declara la instancia de la clase. Los parámetros <i>te</i> , <i>mA</i> , <i>mR</i> , <i>uA</i> , <i>uR</i> son, respectivamente: el tema a discutir, la marca de aceptación, la marca de rechazo, el umbral de aceptación, y el umbral de rechazo; que se utilizarán durante la sesión.
	guardarDialogo()	Guarda el dialogo de una sesión del <i>FDJ</i> .
	eliminarDialogo()	Elimina el dialogo desarrollado durante la sesión del <i>FDJ</i> .

Tabla 4: Métodos del Modelo Conceptual del *FDJ*

Es importante recalcar que algunos de los métodos planteados en la tabla anterior, están redactados de manera “*ambigua*” e “*imprecisa*”. Esta situación se presenta porque: En el nivel de especificación, se ha procurado evitar la introducción de elementos y conceptos que puedan establecer consideraciones de diseño prematuras. Así, por ejemplo, no se han hecho consideraciones acerca del alcance que tienen los métodos y los atributos dentro del modelo conceptual.

La ilustración que se presenta a continuación, describe gráficamente la ampliación que se propone al modelo conceptual del *FDJ*.

⁴ En [NÚÑE99] no se especifica cual de todas las consultas se considera como predeterminada.

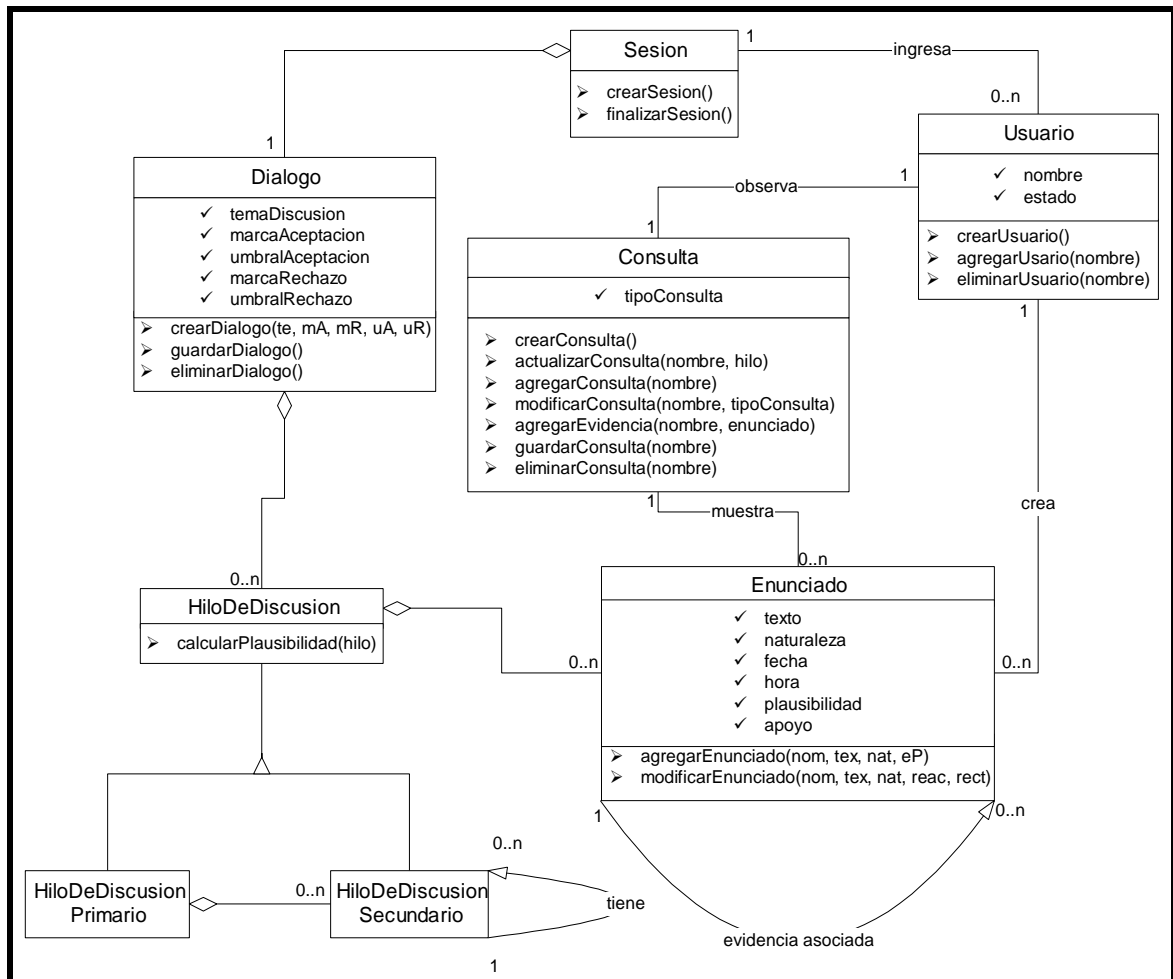


Ilustración 20: Adición de Métodos al Modelo Conceptual

Nuevamente es importante recalcar que en este diagrama se ha procurado modelar únicamente los conceptos del sistema, sin entrar en detalles de implementación.

Se puede observar que los atributos y los métodos de las clases se han calificado con los símbolos ✓ y ➤, respectivamente, que no aparecen dentro del *UML*.

Esta “*omisión*” aparente a las reglas del *UML* se “*justifica*” porque, en este nivel, no se desea entrar en detalles de implementación; por ejemplo, calificar a los

métodos y atributos con propiedades de visibilidad⁵ con respecto a las restantes clases; y tipificar los parámetros recibidos y devueltos por los métodos [BOOC96].

2.9. DIAGRAMA DE PAQUETES.

Con base en el modelo conceptual previamente desarrollado, se puede establecer el diagrama de paquetes para el *FDJ*. Este diagrama ilustra las relaciones de dependencia que existen entre las clases que integran al *FDJ* [FOSC99].

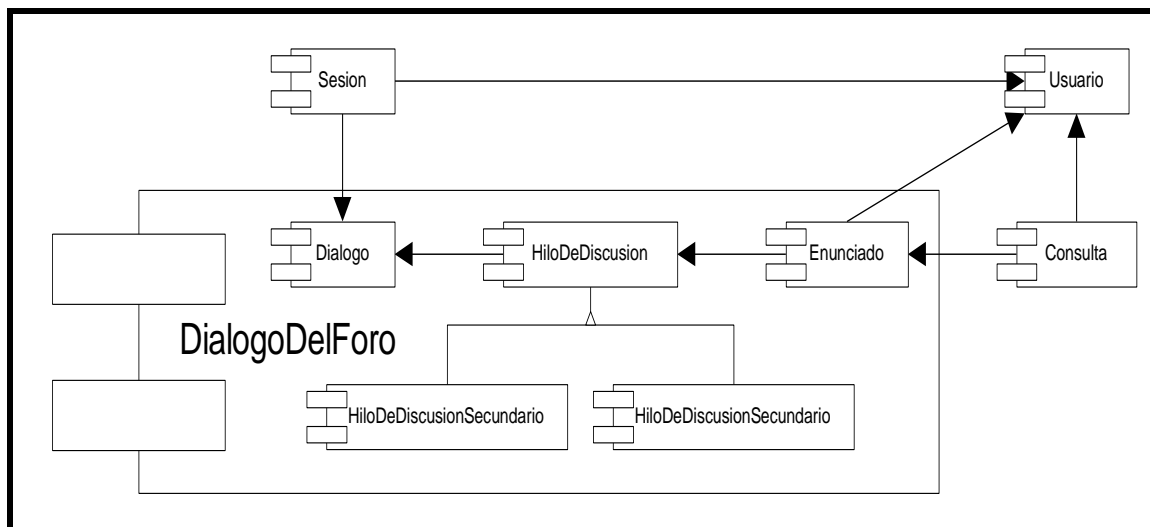


Ilustración 21: Diagrama de Paquetes del *FDJ*

La diferencia fundamental entre el diagrama de paquetes y el modelo conceptual, desarrollado en la fase de especificación, radica en la reagrupación de clases en la superclase *DialogoDelForo*. Esta reagrupación responde a los criterios que se plantean a continuación:

⁵ Según las convenciones del UML [FOSC99]: público (+), privado (-) y protegido (#).

1. De acuerdo al modelo conceptual, la clase *Dialogo* en sí misma, sólo contiene el tema de discusión, así como las marcas y umbrales de aceptación y rechazo.
2. Las marcas y umbrales se utilizan al momento de calcular el estado de plausibilidad de los hilos de discusión del foro.
3. Los hilos de discusión, tanto primarios como secundarios, en sí mismos, no son más que agrupaciones jerárquicas de enunciados.
4. Dentro del diálogo no existen enunciados libres. Los enunciados, para que tengan sentido, deben estar jerarquizados en algún hilo de discusión.
5. Dentro del modelo conceptual, algunas clases quedaron desprovistas de atributos y métodos -específicamente las clases *HiloDeDiscusionPrimario* e *HiloDeDiscusionSecundario*-.

En esencia, al agrupar en una superclase *DialogoDelForo* a las clases *Dialogo*, *HiloDeDiscusion*, *HiloDeDiscusionPrimario*, *HiloDeDiscusionSecundario* y *Enunciado*, se coloca dentro de un entorno común, a todos los elementos, que en conjunto forman el “*Dialogo del Foro*”. Esta interpretación es semánticamente acorde con la formalización del *FDJ* de [NÚÑE99].

La tabla que se muestra a continuación, presenta la distribución de los atributos y métodos, del modelo conceptual, dentro del diagrama de paquetes.

Clase	Atributo	Método
Usuario	* nombre * estado	crearUsuario() agregarUsuario(nombre) eliminarUsuario(nombre)

Clase	Atributo	Método
Sesion	-----	crearSesion() finalizarSesion()
Consulta	* tipoConsulta	crearConsulta() actualizarConsulta(nombre, hilo) agregarConsulta(nombre) modificarConsulta(nombre, tipoConsulta) agregarEvidencia(nombre, enunciado) guardarConsulta(nombre) eliminarConsulta(nombre)
DialogoDelForo	temaDiscusion marcaAceptación umbralAceptacion marcaRechazo umbralRechazo * texto * naturaleza * fecha * hora * plausibilidad * apoyo	crearDialogo(te, mA, mR, uA, uR) guardarDialogo() eliminarDialogo() calcularPlausibilidad(hilo) agregarEnunciado(nom, tex, nat, eP) modificarEnunciado(nom, tex, nat, reac, rect, enun)

Tabla 5: Atributos y Métodos dentro del Diagrama de Paquetes del *FDJ*

Dentro de la Tabla 5 se puede observar que los atributos y los métodos de la clase *Enunciado* se integraron a la clase *DialogoDelForo*. Esta alteración del modelo conceptual está acorde con lo que se plantea en el diagrama de paquetes.

Además, los atributos de la clase *Usuario* se calificaron con un “*”, al igual que los que pertenecían a las clases *Enunciado* y *Consulta*. Las alteraciones antes descritas responden, fundamentalmente, a que los elementos de datos que integran a las clases *Usuario*, *Consulta* y *Enunciado* tienen multiplicidad “cero o más” dentro del modelo conceptual; de manera directa o indirecta.

Con la discusión relativa al diagrama de paquetes, se finaliza el análisis de la especificación del *FDJ*. En el siguiente capítulo, se analiza el diseño y la arquitectura del *FDJ*, con base en la especificación que se ha planteado en este capítulo.

3. ARQUITECTURA Y DISEÑO DEL FORO DE DISCUSIÓN JERÁRQUICO.

En este capítulo se analiza la arquitectura y el diseño interno del foro de discusión jerárquico. En tal sentido, se estudia:

1. El diagrama de emplazamiento,
2. El diagrama de clases estáticas,
3. El modelo de identificación de enunciados,
4. Las estructuras que datos XML que se utilizarán,
5. La estructura de los mensajes que intercambiará el *FDJ*,

Todos los casos, se modelan desde la perspectiva de implementación del sistema. Al igual que en el capítulo anterior, en el diseño del *FDJ* se utilizará UML y el lector puede consultar, por los detalles de la formalización del *FDJ*, en el “Anexo 1: formalización del foro de discusión jerárquico.”

3.1. DIAGRAMA DE EMPLAZAMIENTO: ARQUITECTURA CLIENTE / SERVIDOR DEL *FDJ*.

Con base en el modelo conceptual y el diagrama de paquetes analizados en el capítulo anterior, se puede modelar la arquitectura Cliente / Servidor que se utilizará en la implementación del *FDJ*, a través de un diagrama de emplazamiento, que se presenta a continuación.

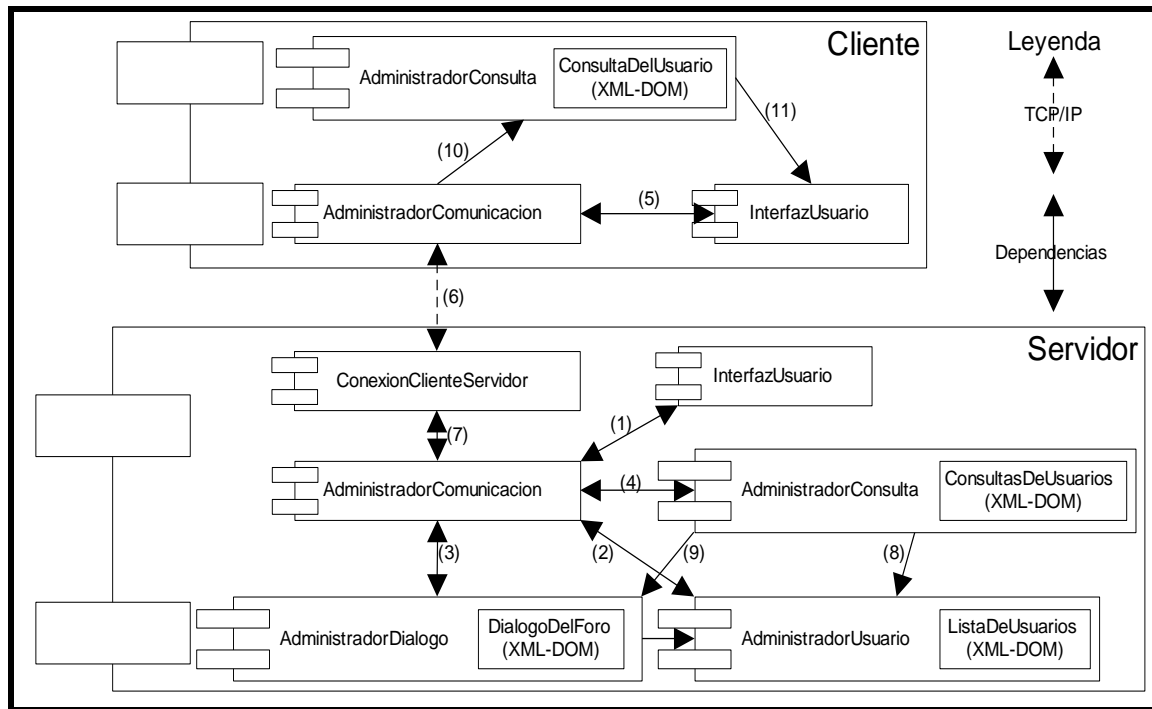


Ilustración 22: Diagrama de Emplazamiento del FDJ

A primera vista, parece que este diagrama no está muy relacionado con el diagrama de paquetes previamente expuesto, pero si lo están. A continuación, se presentan algunos argumentos del por qué ambos diagramas están relacionados.

1. El diagrama de emplazamiento incluye varios paquetes que no aparecen dentro del primer diagrama: Interfaces (Cliente / Servidor), Administrador de Comunicación (Cliente / Servidor), Administradores de Consultas (Cliente / Servidor) y Administradores de Diálogo (Servidor). Además, los elementos del diagrama de paquetes han desaparecido o al menos, cambiado de nombre. Sin embargo, todos los elementos del diagrama de paquetes están presentes en el diagrama de emplazamiento, solamente que bajo otra perspectiva.
2. En el diagrama de emplazamiento se dividió el sistema dos subsistemas: cliente y servidor. Esta división responde a la necesidad práctica de que los

usuarios del *FDJ* lo utilizarán desde lugares físicamente diferentes entre sí, pero al mismo tiempo -interacción sincrónica distribuida, ver Ilustración 1-. En el diagrama de paquetes este problema no se considera.

3. El diagrama de emplazamiento que se presenta en la Ilustración 22 está planteado bajo la perspectiva de la implementación del sistema; modela la arquitectura del *FDJ*. En cambio, el diagrama de paquetes, presentado en la Ilustración 21, modela a la especificación del sistema.

La tabla que se presenta a continuación, explica y complementa las relaciones existentes entre los elementos de los diagramas de paquetes y emplazamiento.

Clases		Relaciones entre diagramas
Diagrama de Paquetes	Diagrama de Emplazamiento	
Usuario	AdministradorUsuario	Ambas clases administran el ingreso y la salida de los usuarios a las sesiones del foro.
DialogoDelForo	AdministradorDialogo	Ambas clases administran el diálogo -los enunciados- que expresan los usuarios durante la sesión del foro.
Consulta	AdministradorConsulta (Cliente / Servidor)	Ambas clases administran las consultas que realizan los usuarios en el diálogo. El diagrama de emplazamiento especializa la clase <i>Consulta</i> en dos partes: una le permite al usuario observar su consulta (<i>AdministradorConsulta Cliente</i>); la otra, administra las consultas de todos los usuarios en el servidor; permite determinar cuando se debe actualizar la consulta de un usuario particular (<i>AdministradorConsulta Servidor</i>).
-----	AdministradorComunicacion (Cliente / Servidor)	Se ofrece un mecanismo a los Clientes del sistema para que se comuniquen con el servidor del <i>FDJ</i> . No aparecen en el diagrama de paquetes porque ese diagrama está planteado a nivel de especificación.

Clases		Relaciones entre diagramas
Diagrama de Paquetes	Diagrama de Emplazamiento	
Sesion	ConexionClienteServidor	Ambas clases administran la entrada / salida de los clientes a la sesión del diálogo. Cuando un Cliente se agrega a la sesión, se le crea una instancia de la clase <i>ConexiónClienteServidor</i> , que administrará la comunicación entre <i>AdministradorComunicación</i> del Servidor y del Cliente.
-----	InterfazUsuario (Cliente / Servidor)	Se ofrece un mecanismo a los usuarios del sistema (Usuario => Cliente / Administrador => Servidor), para puedan comunicarse con el sistema. No aparecen en el diagrama de paquetes porque está planteado a nivel de especificación.

Tabla 6: Relaciones entre las Clases de los Diagramas de Paquetes y Emplazamiento

En conclusión, de acuerdo a las consideraciones planteadas en la Tabla 6, se puede afirmar que el diagrama de paquetes y el diagrama de emplazamiento del *FDJ*, representan el mismo problema, observado desde dos perspectivas distintas.

3.2. INTERACCIÓN ENTRE LAS CLASES DEL DIAGRAMA DE EMPLAZAMIENTO.

En esta sección se explica la forma en que interactúan las clases del diagrama de emplazamiento que describe al *FDJ*. Los números que se colocan entre paréntesis hacen referencia al flujo de mensajes y datos entre las clases del diagrama.

Cuando se analice la interacción entre las clases, es importante que se tome en cuenta, en el contexto, las consideraciones planteadas en las secciones 3.3, 3.4 y 3.5, ya que aunque no son mencionadas directamente en todos los casos, siempre están presentes en el contexto de la interacción de las clases.

La explicación se modela a través de una tabla donde se presentan las clases presentes en el diagrama de emplazamiento, conjuntamente con las funciones que desempeñan y las estructuras de datos (XML-DOM) más importantes que se utilizan en el sistema.

Localización	Clase	Funciones que desempeña
Servidor	InterfazUsuario	<ol style="list-style-type: none"> 1. Recibe los parámetros que regirán el dialogo entre los clientes, a saber: <ol style="list-style-type: none"> 1.1. Puerto de Comunicaciones que utilizarán los clientes. 1.2. Tema a Discutir 1.3. Marcas y umbrales de aceptación y rechazo 2. Inicializa a la instancia de la clase <i>AdministradorComunicación</i> con los parámetros antes mencionados (1). 3. Permite que el administrador del foro altere la cantidad máxima de usuarios que pueden ser admitidos en el foro (1). 4. Presenta al administrador del foro una lista que contiene a los usuarios del foro (1) y (2).
Servidor	Administrador Comunicación	<ol style="list-style-type: none"> 1. Controla la transmisión y recepción de los paquetes de información que se intercambian entre el servidor y cada uno de los usuarios (7), (6). 2. Desensambla los paquetes que recibe de los clientes, y envía la instrucción y el usuario, al administrador correspondiente –Diálogos, Consulta o Usuarios-, para que realice la transacciones que amerite (4), (2), (3). 3. Ensambla los paquetes que se envían a los clientes del foro (7), (6). 4. Controla la cantidad máxima de usuarios que pueden ser admitidos en el foro. 5. Controla cuando se debe actualizar la información que se exhibe en la interfaz de Usuario del Servidor (1), (2) y en las interfaces de los clientes (2), (3), (4), (7), (6). 6. La estructura interna de los paquetes se detallará en la sección 3.5.
Servidor	Conexión Cliente Servidor	<ol style="list-style-type: none"> 1. Administra la conexión TCP/IP entre los AdministradorComunicacion del servidor y de un cliente en particular.
Servidor	Administrador Usuario	<ol style="list-style-type: none"> 1. Administra el ingreso y la salida de los usuarios en el foro (2), (7), (6). 2. Informa a los usuarios el resultado de sus intentos de ingreso a la sesión (2), (7), (6), (5).

Localización	Clase	Funciones que desempeña
Servidor	ListaDeUsuarios (XML-DOM)	<ol style="list-style-type: none"> 1. Contiene la representación XML-DOM de la información correspondiente a cada uno de los usuarios que se conecta al foro. 2. Su estructura interna se detallará en el momento que se analicen las estructuras XML-DOM, en la sección 3.4.
Servidor	Administrador Dialogo	<ol style="list-style-type: none"> 1. Acepta o rechaza las alteraciones del diálogo que solicitan los clientes - agregar, modificar, desactivar / reactivar, retractar enunciados-: Acepta: (3), (7), (6), (10), (11); Rechaza: (3), (7), (6), (5); 2. Recalcula el estado de plausibilidad de los hilos primarios del diálogo general, cuando un usuario ingresa al foro (3), o agrega / altera un enunciado específico.
Servidor	DialogoDel Foro (XML-DOM)	<ol style="list-style-type: none"> 1. Contiene la representación XML-DOM del diálogo que se desarrolla dentro del foro, a nivel de todos los clientes que se encuentran conectados. 2. Está estructurado jerárquicamente, de acuerdo al esquema de hilos de discusión. 3. Su estructura interna se detallará en el momento que se analicen las estructuras XML-DOM -sección 3.4- y el modelo de identificación de enunciados -sección 3.3-.
Servidor	Administrador Consulta	<ol style="list-style-type: none"> 1. Envía grupos de enunciados a cada uno de los usuarios que los soliciten, o que los requieran, de acuerdo a lo que resulte de cotejar el DialogoDelForo con ConsultasDeUsuarios (9), (4), (7), (6), (10), (11).
Servidor	ConsultaDe Usuarios (XML-DOM)	<ol style="list-style-type: none"> 1. Contiene la representación XML-DOM de una bitácora donde AdministradorConsulta registra los cambios que debe actualizar en la consulta de cada usuario. 2. Estos cambios se le enviarán al usuario, sin que él los solicite, porque consisten en las actualizaciones que requiere su consulta, para que sea consisten con del DialogoDelForo, por efecto de las alteraciones del diálogo realizadas por él u otros usuarios. 3. Su estructura interna se detallará en el momento que se analice las estructuras XML-DOM -sección 3.4-.

Localización	Clase	Funciones que desempeña
Cliente	InterfazUsuario	<ol style="list-style-type: none"> 1. Recibe los parámetros que utilizará el usuario en particular, para conectarse con el servidor del foro: <ol style="list-style-type: none"> 1.1. Puerto de Comunicación 1.2. Nombre o dirección IP del servidor 1.3. Nombre propuesto 2. Entrega a AdministradorComunicacion del cliente los parámetros antes mencionados para que tramite su envío al servidor (5). 3. Muestra al usuario la consulta que realiza del diálogo del foro, estructurada como un árbol, con el tema de discusión como su raíz (11), o los mensajes de error que se reciben del servidor (5).
Cliente	Administrador Comunicación	<ol style="list-style-type: none"> 1. Controla la transmisión y recepción de los paquetes de información que se intercambian entre el cliente y el servidor (6). 2. Ensambla los paquetes que se envían al servidor del foro. 3. Desensambla los paquetes que recibe del servidor, y envía las instrucciones al AdministradorConsulta, para que actualice la consulta del cliente (10). 4. Cuando el cliente se conecta por primera vez, recibe el identificador interno de usuario que utilizará, o un mensaje de error (6), (5). 5. La estructura interna de los paquetes se detallará en la sección 3.5.
Cliente	Administrador Consulta	<ol style="list-style-type: none"> 1. Administra la actualización de la consulta del cliente, ya sea por los cambios que el usuario aporta al diálogo, o por los cambios que realicen otros usuarios (10).
Cliente	Consultadle Usuario (XML-DOM)	<ol style="list-style-type: none"> 1. Contiene la representación XML-DOM de los enunciados que el usuario observa en un momento dado. Es un subconjunto de enunciados del DialogoDelForo. 2. Está estructurado jerárquicamente, de acuerdo al esquema de hilos de discusión. 3. Su estructura interna se detallará en el momento que se analice las estructuras XML-DOM -sección 3.4-.

Tabla 7: Interpretación del Diagrama de Emplazamiento del FDJ

Es importante recalcar que, de acuerdo al diagrama de emplazamiento y su interpretación, las clases *AdministradorComunicacion* (en el cliente y el servidor), controlan a nivel interno a la aplicación, pues administran la comunicación cliente / servidor, y también administran la comunicación entre las clases internas del cliente y del servidor, a nivel de dependencias.

A continuación, se analizarán algunos temas que hemos mencionado previamente y se deben estar bien definidos al momento de considerar los atributos y métodos que se presentan dentro de los diagramas de emplazamiento y de clases. Ellos son:

1. El modelo de identificación de los enunciados dentro del diálogo.
2. La estructura de los documentos XML utilizados.
3. La estructura de los mensajes que se intercambian entre el cliente y el servidor.

Luego, se finalizará el capítulo con el análisis de los atributos y métodos que se presentan dentro de los diagramas de emplazamiento y de clases.

3.3. MODELO DE IDENTIFICACIÓN DE LOS ENUNCIADOS DENTRO DEL DIÁLOGO.

En [NÚÑE99] se define un modelo de identificación de los enunciados del *FDJ*. Está basado en únicamente 4 parámetros, y representa una notación muy compacta y conveniente para realizar estudios teóricos (los detalles de esta notación están en la sección 8.4).

Sin embargo, para efectos de facilitar la implementación, se ha decidido utilizar un modelo de identificación más intuitivo y que facilite la localización de enunciados tanto como sea posible. En consecuencia, los enunciados del diálogo serán identificados de acuerdo al modelo que se plantea a continuación.

Se considera el conjunto de hilos de discusión $fd = \langle h_1, h_2, \dots, h_n \rangle$, que integra el diálogo de una sesión del foro, como un árbol n-ario. En dicho árbol, se asume que su *raíz* está definida por el “*Tema de Discusión*”. En consecuencia, todos los hilos de discusión serán *nodos hijos del nodo raíz*.

Por otro lado, se asume que los nodos -enunciados- serán identificados al momento de ser agregados al árbol. Además, la raíz del árbol tendrá identificación nula.

También se acepta que de acuerdo a [JOZA98], el “*grado*” de un nodo dentro de un árbol n-ario, se define como la cantidad de hijos que posee el nodo.

Tomando todos los elementos de juicio que se han mencionado, se puede modelar una función que genera los índices o llaves de los nodos, definida en los términos que se expresan a continuación: Consideremos un nodo n_p al que deseamos agregar un hijo n_h . La llave x que identifica a n_h está definida como:

$$\begin{aligned} x &= llave(n_p) \oplus \parallel grado(n_p) + 1 \parallel \oplus ". " \\ x &= i \oplus \parallel k + 1 \parallel \oplus ". " \end{aligned}$$

Ilustración 23: Función Generadora de Llaves para los Nodos de un Árbol n-ario

Donde:

- ☑ $llave(n_p) = i$ es una función que relaciona a cada nodo del árbol n_p con su identificador: i ,

- ☑ $grado(n_p) = k$ es una función que relaciona a cada nodo del árbol n_p con la cantidad de hijos que posee: k ,
- ☑ \oplus denota a la concatenación de cadenas de caracteres,
- ☑ $\|y\|$ denota una función que transforma enteros a cadenas,
- ☑ $"."$ denota una cadena constante.

Este modelo, aparentemente más complejo, resulta en la práctica muy simple, pues no es más que la formalización del esquema decimal de numeración jerárquico. Además, a nivel de implementación resulta ventajoso, pues cada llave que identifica a un nodo, contiene la información histórica de todos sus padres inmediatos, hasta llegar al nodo raíz, de manera directa.

La ilustración que se presenta a continuación, muestra un diálogo genérico dentro del foro, con sus enunciados organizados como los nodos del árbol, conjuntamente con el identificador de cada nodo, de acuerdo al modelo antes planteado:

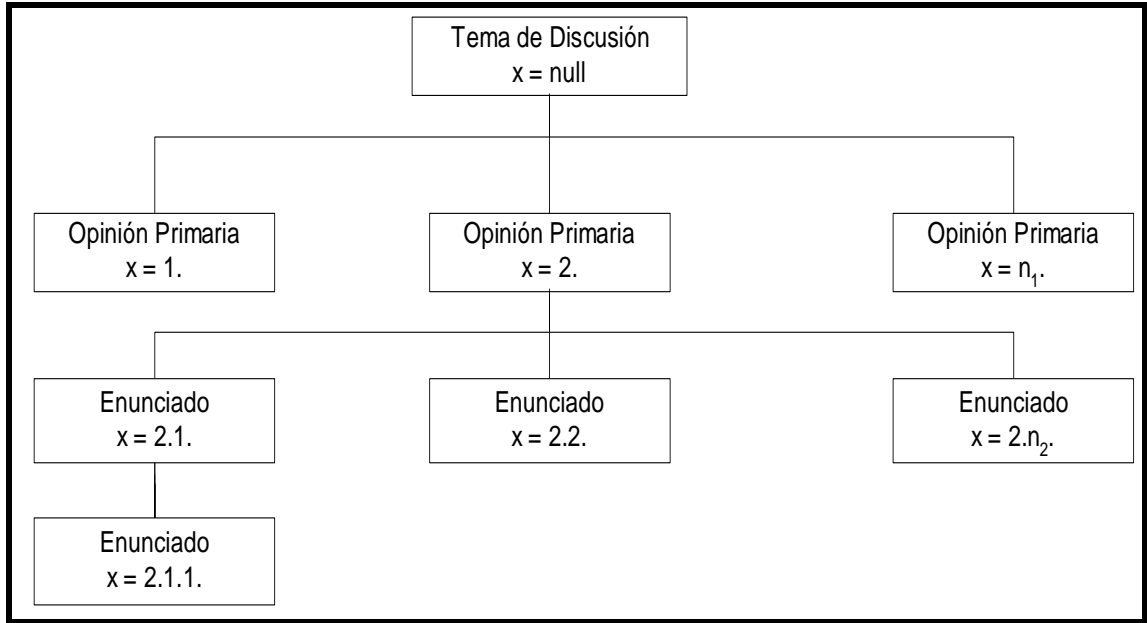


Ilustración 24: Diálogo Genérico con los Nodos Identificados Bajo el Modelo Decimal

En la Ilustración 24, no se debe olvidar que el identificador de cada nodo se calcula en el momento que éste es agregado al árbol. Además, para los efectos del cálculo del identificador del nodo, el árbol está compuesto -en ese momento-, por el nodo padre y los nodos hermanos a la izquierda del nodo que se desea agregar.

A continuación, se mostrará como se calcularon algunos de los identificadores de los nodos del árbol de la Ilustración 24:

$$\begin{aligned}
 2. &= null \oplus \parallel Grad(null) + 1 \parallel \oplus "." = \parallel 1 + 1 \parallel \oplus "." = 2. \\
 2.n_2. &= "2." \oplus \parallel Grad("2.") + 1 \parallel \oplus "." = "2." \parallel (n_2 - 1) + 1 \parallel \oplus "." = 2.n_2. \\
 2.1.1. &= "2.1." \oplus \parallel Grad("2.1.") + 1 \parallel \oplus "." = "2.1." \parallel 0 + 1 \parallel \oplus "." = 2.1.1.
 \end{aligned}$$

Ilustración 25: Cálculo de Algunos Identificadores de Nodos de la Ilustración 24

3.4. ESTRUCTURA DE LOS DOCUMENTOS XML UTILIZADOS.

En esta sección se analizará la estructura de los documentos XML que se utilizarán para almacenar la información dentro del *FDJ*.

Con base en la Ilustración 22 (Diagrama de Emplazamiento), se puede observar que se requieren 4 documentos para administrar la información dentro del *FDJ*. Ellos son:

- ☑ La lista de usuarios de la sesión (*ListaDeUsuarios*), donde se registran las activaciones / desactivaciones de los usuarios que ingresan al foro, así como la unicidad de los nombres que utilizan.
- ☑ El Diálogo que desarrollan los usuarios (*DialogoDelForo*), donde se almacena toda la estructura del diálogo que desarrollan los usuarios del foro en el transcurso de la sesión, organizado en hilos de discusión y enunciados, de acuerdo a la formalización del *FDJ*.
- ☑ Una Consulta general (*ConsultasDeUsuarios*), donde se administran, dinámicamente, los enunciados que forman parte de la consulta de cada uno de los clientes activos en la sesión.
- ☑ La consulta que realiza cada usuario sobre el diálogo (*ConsultaDelUsuario*), y que reside en el sistema del cliente.

A continuación, se procederá a analizar la estructura de cada uno de estos documentos, en términos de las “*Definiciones de Tipos de Datos*” -en adelante, DTD- que los definen.

3.4.1. LISTA DE USUARIOS.

La DTD que define a la lista de usuarios del *FDJ* se presenta a continuación:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT ListaDeUsuarios (Usuario*)>
  <!ELEMENT Usuario (IdUsuario, Nombre, Host, Estado)>
    <!ELEMENT IdUsuario (#PCDATA)>
    <!ELEMENT Nombre (#PCDATA)>
    <!ELEMENT Host (#PCDATA)>
    <!ELEMENT Estado (#PCDATA)>
```

Ilustración 26: DTD para la Lista de Usuarios del *FDJ*

De acuerdo a la interpretación de esta DTD, se tiene que:

1. *ListaDeUsuarios* es el nodo raíz del documento XML.
2. Está compuesto por 0 o más nodos de tipo *Usuario*.
3. Cada nodo de tipo *Usuario* está compuesto a su vez por cuatro nodos identificados como: *IdUsuario*, *Nombre*, *Host*, y *Estado*.
4. Los nodos que integran al nodo *Usuario* contienen texto plano.

A continuación, se expone la interpretación semántica de los nodos que componen a la DTD:

- ☒ *ListaDeUsuarios*: Representa al conjunto de usuarios activos e inactivos que intervienen en una sesión del *FDJ*.
- ☒ *Usuario*: Representa a un usuario cualquiera de los que intervienen en la sesión.

- ☑ *IdUsuario*: Representa al identificador interno del usuario dentro del servidor del *FDJ*. Adopta el valor de una cadena de caracteres que representa a un número entero positivo no nulo.
- ☑ *Nombre*: Representa el nombre o pseudónimo que adopta el usuario en el transcurso de la sesión.
- ☑ *Host*: Representa el nombre del sistema en que se encuentra el cuando se une a la sesión del *FDJ*.
- ☑ *Estado*: Permite diferenciar a los usuarios que se han retirado de la sesión, de los que se mantienen en ella. Respectivamente, puede adoptar el valor “Inactivo” o “Activo”, según el caso que se aplique al usuario en particular.

La ilustración que se puede observar a continuación, presenta el listado de un documento XML genérico, que responde a la DTD antes analizada:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<ListaDeUsuarios>
  <Usuario>
    <IdUsuario>1</IdUsuario>
    <Nombre>Usuario 1</Nombre>
    <Host>Sistema 1</Host>
    <Estado>Activo</Estado>
  </Usuario>
  <Usuario>
    <IdUsuario>2</IdUsuario>
    <Nombre>Usuario 2</Nombre>
    <Host>Sistema 2</Host>
    <Estado>Activo</Estado>
  </Usuario>
  <Usuario>
    <IdUsuario>3</IdUsuario>
    <Nombre>Usuario 3</Nombre>
    <Host>Sistema 3</Host>
    <Estado>Inactivo</Estado>
  </Usuario>
  <Usuario>
    <IdUsuario>4</IdUsuario>
    <Nombre>Usuario 4</Nombre>
    <Host>Sistema 4</Host>
    <Estado>Activo</Estado>
  </Usuario>
  <Usuario>
    <IdUsuario>5</IdUsuario>
    <Nombre>Usuario 5</Nombre>
    <Host>Sistema 3</Host>
    <Estado>Inactivo</Estado>
  </Usuario>
</ListaDeUsuarios>

```

Ilustración 27: Documento XML de la Lista de Usuarios del *FDJ*

3.4.2. DIÁLOGO DEL FORO.

La DTD que define al diálogo del foro se presenta a continuación:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT DialogoDelForo (TemaADiscutir,
                           MarcaDeAceptacion,
                           MarcaDeRechazo,
                           UmbralDeAceptacion,
                           UmbralDeRechazo,
                           Dialogo)>
  <!ELEMENT TemaADiscutir (#PCDATA)>
  <!ELEMENT MarcaDeAceptacion (#PCDATA)>
  <!ELEMENT MarcaDeRechazo (#PCDATA)>
  <!ELEMENT UmbralDeAceptacion (#PCDATA)>
  <!ELEMENT UmbralDeRechazo (#PCDATA)>
  <!ELEMENT Dialogo (Enunciado*)>
    <!ELEMENT Enunciado (Llave,
                        EvidenciaAsociada,
                        Naturaleza,
                        NaturalezaAnterior,
                        Texto,
                        Identificador,
                        Plausibilidad,
                        Apoyo)>
      <!ELEMENT Llave (#PCDATA)>
      <!ELEMENT EvidenciaAsociada (Enunciado*)>
      <!ELEMENT Naturaleza (#PCDATA)>
      <!ELEMENT NaturalezaAnterior (#PCDATA)>
      <!ELEMENT Texto (#PCDATA)>
      <!ELEMENT Identificador (IdUsuario, Fecha, Hora)>
        <!ELEMENT IdUsuario (#PCDATA)>
        <!ELEMENT Fecha (#PCDATA)>
        <!ELEMENT Hora (#PCDATA)>
      <!ELEMENT Plausibilidad (#PCDATA)>
      <!ELEMENT Apoyo (#PCDATA)>

```

Ilustración 28: DTD para la el Diálogo del FDJ

De acuerdo a la interpretación de esta DTD -bastante más complicada que la anterior-, se tiene que:

1. *DialogoDelForo* es el nodo raíz del documento XML.
2. Está compuesto por 6 nodos identificados como: *TemaADiscutir*, *MarcaDeAceptacion*, *MarcaDeRechazo*, *UmbralDeAceptacion*, *UmbralDeRechazo* y *Dialogo*.

3. Los nodos que integran al nodo *DialogoDelForo* contienen texto plano, salvo el nodo *Dialogo*, que está compuesto por 0 o más nodos de tipo *Enunciado*.
4. Los nodos de tipo *Enunciado* están compuestos por 8 nodos identificados como: *Llave*, *EvidenciaAsociada*, *Naturaleza*, *NaturalezaAnterior*, *Texto*, *Identificador*, *Plausibilidad*, y *Apoyo*.
5. Los nodos *Llave*, *Naturaleza*, *NaturalezaAnterior*, *Texto*, *Plausibilidad*, y *Apoyo* contienen texto plano.
6. Los nodos de tipo *Identificador* están compuestos por 3 nodos denotados como *IdUsuario*, *Fecha*, *Hora*.
7. Los nodos de tipo *EvidenciaAsociada* están compuestos por 0 o más nodos de tipo *Enunciado* (es una definición recursiva).

A continuación, se expone la interpretación semántica de todos los nodos que componen a la DTD del Diálogo:

- ☑ *DialogoDelForo*: Representa al Dialogo que se desarrolla dentro de la sesión, incluyendo al tema que se discute y las marcas y umbrales de aceptación y rechazo, necesarias para realizar el cálculo del Estado de Plausibilidad del diálogo. Se equipara directamente con la clase *DialogoDelForo* del Diagrama de Paquetes (Ilustración 21) y con la clase *DialogoDelForo* del Diagrama de Emplazamiento (Ilustración 22). Es el objeto principal en el documento.
- ☑ *TemaADiscutir*, *MarcaDeAceptacion*, *MarcaDeRechazo*, *UmbralDeAceptacion*, y *UmbralDeRechazo*: Los nombres de estos

nodos se explican por si mismos, al comparar la DTD contra la formalización del *FDJ*.

- ☑ *Dialogo*: Representa el conjunto de hilos de discusión que componen la discusión que desarrollan los participantes de la sesión.
- ☑ *Enunciado*: Es la idea o concepto que desea aportar un usuario cualquiera al diálogo que se desarrolla en la sesión.
- ☑ *Llave*: Representa al elemento que nos permite diferenciar cada uno de los nodos que integran el diálogo, obtenido de acuerdo al procedimiento señalado en 3.3.
- ☑ *EvidenciaAsociada*: Representa al conjunto de enunciados que permiten calcular el estado de plausibilidad del enunciado. Se refiere al conjunto de enunciados, hijos de un enunciado dado.
- ☑ *Naturaleza, NaturalezaAnterior*: Representa la categorización actual y previa, respectivamente, de un enunciado dado, con respecto a su padre.
- ☑ *Texto*: Representa la idea o concepto que presenta el enunciado.
- ☑ *Identificador*: Representa el componente que permite diferenciar a los enunciados del diálogo.
- ☑ *IdUsuario*: Representa al identificador interno de usuario, que a su vez, se corresponde con el nombre del usuario que agregó el

enunciado. Adopta del valor de una cadena de caracteres que representa a un número entero positivo no nulo.

- ☑ *Fecha*: Representa a la fecha, en formato aaaa-mmmm-dd, en que se creó el enunciado, según el servidor del foro.
- ☑ *Hora*: Representa a la hora, en formato 00:00:00 AM/PM (12 horas), en que se creó el enunciado, según el servidor del foro.
- ☑ *Plausibilidad*: Representa el estado de plausibilidad que tiene asignado un enunciado. Puede asumir los valores *Satisfactorio*, *Aceptable*, *Dudoso*, *Insatisfactorio*, *Indeterminado*.
- ☑ *Apoyo*: Es una medida del grado de aceptación de la naturaleza de un enunciado dado, calculado con base en la evidencia a favor y la evidencia en contra de dicho enunciado. Es un número entre 0 y 1, sin incluir los extremos.

Es importante señalar que el diseño de la DTD impone una limitación importante al sistema, cuando lo comparamos contra la formalización expuesta en [NÚÑE99].

La formalización, tal y como está planteada, involucra la capacidad de registrar, en un objeto convenientemente diseñado, la evolución histórica de todas las naturalezas de cada enunciado. Este objeto permite que los usuarios puedan realizar operaciones de reactivar / desactivar enunciados de tal manera que un enunciado cualquiera, a través de la operación modificar, puede recuperar cualquiera de las naturalezas que ha tenido previamente.

Por otro lado, como se está implementando un prototipo de aplicación -que pretende servir únicamente de instrumento de investigación del problema-, se ha

considerado razonable dotar al sistema de una capacidad limitada de deshacer y rehacer, a solamente un nivel, para simplificar el diseño. Esta limitación está reflejada en el diseño de la DTD que se analiza en esta sección.

La ilustración que se puede observar a continuación, presenta el listado de un documento XML genérico, que responde a la DTD antes analizada:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DialogoDelForo>
  <TemaADiscutir>Un Tema Cualquiera</TemaADiscutir>
  <MarcaDeAceptacion>0.5</MarcaDeAceptacion>
  <MarcaDeRechazo>0.5</MarcaDeRechazo>
  <UmbralDeAceptacion>0.25</UmbralDeAceptacion>
  <UmbralDeRechazo>0.25</UmbralDeRechazo>
  <Dialogo>
    <Enunciado>
      <Llave>2.</Llave>
      <EvidenciaAsociada>
        <Enunciado>
          <Llave>2.1.</Llave>
          <EvidenciaAsociada/>
          <Naturaleza>F</Naturaleza>
          <NaturalezaAnterior>F</NaturalezaAnterior>
          <Texto>Un Enunciado Cualquiera</Texto>
          <Identificador>
            <IdUsuario>1</IdUsuario>
            <Fecha>2001/Julio/20</Fecha>
            <Hora>02:42:46 AM</Hora>
          </Identificador>
          <Plausibilidad>Aceptable</Plausibilidad>
          <Apoyo>1.0000</Apoyo>
        </Enunciado>
      </EvidenciaAsociada>
      <Naturaleza>O</Naturaleza>
      <NaturalezaAnterior>O</NaturalezaAnterior>
      <Texto>Y Otro Enunciado Cualquiera</Texto>
      <Identificador>
        <IdUsuario>1</IdUsuario>
        <Fecha>2001/Julio/20</Fecha>
        <Hora>02:42:37 AM</Hora>
      </Identificador>
      <Plausibilidad>Satisfactorio</Plausibilidad>
      <Apoyo>1.0000</Apoyo>
    </Enunciado>
  </Dialogo>
</DialogoDelForo>
```

Ilustración 29: Documento XML del Diálogo del FDJ

3.4.3. CONSULTA GENERAL.

La DTD que define la consulta general del foro, se presenta a continuación:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT ConsultasDeUsuarios (ConsultaDeUsuario*)>
  <!ELEMENT ConsultaDeUsuario (IdUsuario,
                                TipoConsulta,
                                ConsultaActualizada,
                                Dialogo)>
    <!ELEMENT IdUsuario (#PCDATA)>
    <!ELEMENT TipoConsulta (#PCDATA)>
    <!ELEMENT ConsultaActualizada (#PCDATA)>
    <!ELEMENT Dialogo (Enunciado*)>
      <!ELEMENT Enunciado (Llave, EvidenciaAsociada)>
        <!ELEMENT Llave (#PCDATA)>
        <!ELEMENT EvidenciaAsociada (Enunciado*)>
```

Ilustración 30: DTD para la Consulta General del FDJ

De acuerdo a la interpretación de esta DTD, se tiene que:

1. *ConsultasDeUsuarios* es el nodo raíz del documento XML.
2. Está compuesto por 0 o más nodos de tipo *ConsultaDeUsuario*.
3. Cada nodo de tipo *ConsultaDeUsuario* está compuesto a su vez por cuatro nodos identificados como: *IdUsuario*, *TipoConsulta*, *ConsultaActualizada*, y *Dialogo*.
4. Los nodos que integran al nodo *ConsultaDeUsuario* contienen texto plano, a excepción del nodo *Dialogo*.
5. Cada nodo de tipo *Dialogo* está compuesto a su vez por 0 o más nodos de tipo *Enunciado*.

8. Los nodos de tipo *Enunciado* están compuestos por 2 nodos identificados como: *Llave*, y *EvidenciaAsociada*,
9. El nodo *Llave* contiene texto plano.
10. El nodo *EvidenciaAsociada* está compuesto por 0 o más nodos de tipo *Enunciado* (es una definición recursiva).

A continuación, se expone la interpretación semántica de los nodos que componen a un nodo *ConsultasDeUsuarios*:

- ☒ *ConsultaDeUsuario*: Representa a todos los enunciados que componen la consulta que realiza un usuario particular.
- ☒ *IdUsuario*: Representa al identificador interno de usuario, que a su vez, se corresponde con el nombre del usuario que agregó el enunciado. Adopta el valor de una cadena de caracteres que representa a un número entero positivo no nulo.
- ☒ *TipoConsulta*: Representa al tipo de consulta que realiza un usuario particular. Puede asumir los valores "*Ver Opiniones Primarias*", "*Ver Todo*", "*Ver Opiniones A Favor*", "*Ver Opiniones En Contra*", "*Ver Aclaraciones*", "*Ver Lo Leído*" y "*Ver Lo No Leído*".
- ☒ *ConsultaActualizada*: Es una variable de control que permite controlar si la consulta de un usuario en particular es consistente con el diálogo, puede asumir los valores "*true*" o "*false*" en los casos en que está actualiza o no, respectivamente.

- ☑ *Dialogo*: Representa el conjunto de hilos de discusión que componen la discusión que desarrollan los participantes de la sesión.
- ☑ *Enunciado*: Es la idea o concepto que desea aportar un usuario cualquiera al diálogo que se desarrolla en la sesión.
- ☑ *Llave*: Representa al elemento que nos permite diferenciar cada uno de los nodos que integran el diálogo, obtenido de acuerdo al procedimiento señalado en la sección 3.3.
- ☑ *EvidenciaAsociada*: Representa al conjunto de enunciados que permiten calcular el estado de plausibilidad del enunciado. Se refiere al conjunto de enunciados, hijos de un enunciado dado.

La ilustración que se puede observar a continuación, presenta el listado de un documento XML genérico, que responde a la DTD antes analizada:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<ConsultasDeUsuarios>
  <ConsultaDeUsuario>
    <IdUsuario>1</IdUsuario>
    <TipoConsulta>Ver Opiniones Primarias</TipoConsulta>
    <ConsultaActualizada>true</ConsultaActualizada>
    <Dialogo>
      <Enunciado>
        <Llave>1.</Llave>
        <EvidenciaAsociada/>
      </Enunciado>
      <Enunciado>
        <Llave>2.</Llave>
        <EvidenciaAsociada/>
      </Enunciado>
    </Dialogo>
  </ConsultaDeUsuario>
  <ConsultaDeUsuario>
    <IdUsuario>2</IdUsuario>
    <TipoConsulta>Ver Opiniones Primarias</TipoConsulta>
    <ConsultaActualizada>true</ConsultaActualizada>
    <Dialogo>
      <Enunciado>
        <Llave>1.</Llave>
        <EvidenciaAsociada/>
      </Enunciado>
      <Enunciado>
        <Llave>2.</Llave>
        <EvidenciaAsociada>
          <Enunciado>
            <Llave>2.1.</Llave>
            <EvidenciaAsociada/>
          </Enunciado>
        </EvidenciaAsociada>
      </Enunciado>
    </Dialogo>
  </ConsultaDeUsuario>
</ConsultasDeUsuarios>

```

Ilustración 31: Documento XML de la Consulta General del FDJ

3.4.4. CONSULTA DEL CLIENTE.

La DTD que define al diálogo del foro se presenta a continuación:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!--ELEMENT ConsultaDelUsuario (TemaADiscutir, TipoConsulta, Dialogo)-->
  <!--ELEMENT TemaADiscutir (#PCDATA)-->
  <!--ELEMENT TipoConsulta (#PCDATA)-->
  <!--ELEMENT Dialogo (Enunciado*)-->
    <!--ELEMENT Enunciado (Llave,
      EvidenciaAsociada,
      Naturaleza,
      Texto,
      Identificador,
      Plausibilidad,
      Color,
      Apoyo,
      EvidenciaAsociadaVacía,
      NaturalezaAnterior)-->
      <!--ELEMENT Llave (#PCDATA)-->
      <!--ELEMENT EvidenciaAsociada (Enunciado*)-->
      <!--ELEMENT Naturaleza (#PCDATA)-->
      <!--ELEMENT Texto (#PCDATA)-->
      <!--ELEMENT Identificador (Nombre, Fecha, Hora)-->
        <!--ELEMENT Nombre (#PCDATA)-->
        <!--ELEMENT Fecha (#PCDATA)-->
        <!--ELEMENT Hora (#PCDATA)-->
      <!--ELEMENT Plausibilidad (#PCDATA)-->
      <!--ELEMENT Color (#PCDATA)-->
      <!--ELEMENT Apoyo (#PCDATA)-->
      <!--ELEMENT EvidenciaAsociadaVacía (#PCDATA)-->
      <!--ELEMENT NaturalezaAnterior (#PCDATA)-->

```

Ilustración 32: DTD para la Consulta del Cliente del FDJ

De acuerdo a la interpretación de esta DTD, se tiene que:

1. *ConsultaDelUsuario* es el nodo raíz del documento XML.
2. Está compuesto por 3 nodos identificados como: *TemaADiscutir*, *TipoConsulta* y *Dialogo*.
3. Los nodos que integran al nodo *ConsultaDelUsuario* contienen texto plano, salvo el nodo *Dialogo*, que está compuesto por 0 o más nodos de tipo Enunciado.

4. Los nodos de tipo *Enunciado* están compuestos por 10 nodos identificados como: *Llave*, *EvidenciaAsociada*, *Naturaleza*, *NaturalezaAnterior*, *Texto*, *Identificador*, *Plausibilidad*, *Color*, *Apoyo* y *EvidenciaAsociadaVacía*.
5. Los nodos *Llave*, *Naturaleza*, *NaturalezaAnterior*, *Texto*, *Plausibilidad*, *Color*, *Apoyo* y *EvidenciaAsociadaVacía* contienen texto plano.
6. Los nodos de tipo *Identificador* están compuestos por 3 nodos denotados como *IdUsuario*, *Fecha*, *Hora*.
7. Los nodos de tipo *EvidenciaAsociada* están compuestos por 0 o más nodos de tipo *Enunciado* (es una definición recursiva).

A continuación, se expone la interpretación semántica de todos los nodos que componen a la DTD del Diálogo:

- ☑ *ConsultaDelUsuario*: Representa a todos los enunciados que componen la consulta que realiza un usuario en particular.
- ☑ *TemaADiscutir*: El nombre del nodo se explica por mismo, al comparar la DTD contra la formalización del *FDJ*.
- ☑ *TipoConsulta*: Representa al tipo de consulta que realiza un usuario particular. Puede asumir los valores "*Ver Opiniones Primarias*", "*Ver Todo*", "*Ver Opiniones A Favor*", "*Ver Opiniones En Contra*", "*Ver Aclaraciones*", "*Ver Lo Leído*" y "*Ver Lo No Leído*".
- ☑ *Dialogo*: Representa el conjunto de enunciados, organizados en hilos de discusión, que forman parte de la consulta que realiza un usuario en particular.

- ☑ *Enunciado*: Es la idea o concepto que desea aportar un usuario cualquiera al diálogo que se desarrolla en la sesión.
- ☑ *Llave*: Representa al elemento que nos permite diferenciar cada uno de los nodos que integran el diálogo, obtenido de acuerdo al procedimiento señalado en la sección 3.3.
- ☑ *EvidenciaAsociada*: Representa al conjunto de enunciados que permiten calcular el estado de plausibilidad del enunciado. Se refiere al conjunto de enunciados, hijos de un enunciado dado.
- ☑ *Naturaleza, NaturalezaAnterior*: Representa la categorización actual y previa, respectivamente, de un enunciado dado, con respecto a su padre.
- ☑ *Texto*: Representa la idea o concepto que presenta el enunciado.
- ☑ *Identificador*: Representa el componente que permite diferenciar a los enunciados del diálogo.
- ☑ *IdUsuario*: Representa al identificador interno de usuario, que a su vez, se corresponde con el nombre del usuario que agregó el enunciado. Adopta del valor de una cadena de caracteres que representa a un número entero positivo no nulo.
- ☑ *Fecha*: Representa a la fecha, en formato aaaa-mmmm-dd, en que se creó el enunciado, según el servidor del foro.
- ☑ *Hora*: Representa a la hora, en formato 00:00:00 AM/PM (12 horas), en que se creó el enunciado, según el servidor del foro.

- ☑ *Plausibilidad*: Representa el estado de plausibilidad que tiene asignado un enunciado. Puede asumir los valores *Satisfactorio*, *Aceptable Dudoso*, *Insatisfactorio*, *Indeterminado*.
- ☑ *Color*: Representa al color con que se debe pintar el fondo del texto del enunciado. Este color está definido en algunas consultas -ver lo leído y lo no leído, por ejemplo-.
- ☑ *Apoyo*: Es una medida del grado de aceptación de la naturaleza de un enunciado dado, calculado con base en la evidencia a favor y la evidencia en contra de dicho enunciado. Es un número entre 0 y 1, sin incluir los extremos.
- ☑ *EvidenciaAsociadaVacía*: Es una variable de control que nos informa si el enunciado analizado tiene o no evidencia asociada en el diálogo del foro. Adopta el valor de una cadena de caracteres que representa a un valor booleano: si la evidencia asociada del enunciado está vacía toma el valor “*true*”; en cambio si el enunciado tiene evidencia asociada no vacía, toma el valor “*false*”.

La ilustración que se puede observar a continuación, presenta el listado de un documento XML genérico, que responde a la DTD antes analizada:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<ConsultaDelUsuario>
  <TemaADiscutir>Un Tema Cualquiera</TemaADiscutir>
  <TipoConsulta>Ver Opiniones Primarias</TipoConsulta>
  <Dialogo>
    <Enunciado>
      <Llave>1.</Llave>
      <EvidenciaAsociada>
        <Enunciado>
          <Llave>1.1.</Llave>
          <EvidenciaAsociada/>
          <Naturaleza>F</Naturaleza>
          <Texto>Otro Enunciado Cualquiera</Texto>
          <Identificador>
            <Nombre>Usuario 2</Nombre>
            <Fecha>2001/Julio/21</Fecha>
            <Hora>03:35:01 AM</Hora>
          </Identificador>
          <Plausibilidad>Aceptable</Plausibilidad>
          <Color>Blanco</Color>
          <Apoyo>1.0000</Apoyo>
          <EvidenciaAsociadaVacía>true</EvidenciaAsociadaVacía>
          <NaturalezaAnterior>F</NaturalezaAnterior>
        </Enunciado>
      </EvidenciaAsociada>
      <Naturaleza>O</Naturaleza>
      <Texto>Un Enunciado Cualquiera</Texto>
      <Identificador>
        <Nombre>Usuario 1</Nombre>
        <Fecha>2001/Julio/21</Fecha>
        <Hora>03:34:04 AM</Hora>
      </Identificador>
      <Plausibilidad>Satisfactorio</Plausibilidad>
      <Color>Blanco</Color>
      <Apoyo>1.0000</Apoyo>
      <EvidenciaAsociadaVacía>false</EvidenciaAsociadaVacía>
      <NaturalezaAnterior>O</NaturalezaAnterior>
    </Enunciado>
  </Dialogo>
</ConsultaDelUsuario>

```

Ilustración 33: Documento XML de la Consulta del Cliente del FDJ

3.5. ESTRUCTURA DE LOS MENSAJES QUE SE INTERCAMBIAN ENTRE EL CLIENTE Y EL SERVIDOR.

En esta sección se analizará la estructura de los paquetes de información que intercambiarán, entre sí, los administradores de comunicaciones del servidor y de los clientes del *FDJ*.

Los supuestos: Se ha establecido que la comunicación entre el cliente y el servidor del *FDJ* se realizará utilizando una conexión TCP / IP que comunicará a cada cliente, con el servidor. Esta conexión será implementada utilizando Sockets TCP / IP. El flujo de datos entre el cliente y el servidor, bajo este modelo, se reduce entonces a transmitir cadenas de bytes entre las aplicaciones cliente y servidor que, por la implementación TCP en el dominio Internet, se considera confiable [SIGA99].

Teniendo presentes los supuestos antes mencionados, se tiene como entorno del problema: Se tiene la capacidad de enviar y recibir, de manera segura, exclusivamente cadenas de bytes. Estas limitantes obligan a desarrollar una solución al problema de comunicación utilizando estos recursos.

3.5.1. FORMATO DE LOS MENSAJES SIMPLES.

La solución propuesta consiste en estructurar los paquetes de información que se envían entre el cliente y el servidor del *FDJ* como “*grupos de cadenas*”, donde su significado semántico dependerá de unos pocos caracteres que estarán colocados al principio de cada cadena. La solución es, básicamente, utilizar una variante del concepto de “*célula*”, “*trama*”, o “*paquete*”, de longitud fija [TANE97]. La variación consiste en considerar que la trama es de longitud variable.

En general, se puede caracterizar la solución del problema en los siguientes términos:

Cada trama de texto estará formada por los elementos:

$$\boxed{\{tipoMensaje\}\{cabecera\}\{datos\}}$$

Ilustración 34: Formato de Mensaje Simple del FDJ

donde:

- ☑ $\{tipoMensaje\}$ se refiere a un carácter que identificará a la trama como un mensaje de un tipo previamente definido.
- ☑ $\{cabecera\}$ se refiere a un conjunto de caracteres numéricos y delimitados, que permitirán identificar a un conjunto de campos, previamente definidos, con base en $\{tipoMensaje\}$.
- ☑ $\{datos\}$ se refiere a la cadena de caracteres que conforma el mensaje en sí.

Se debe observar que con este modelo de tramas, se puede enviar y recibir mensajes simples, como vectores o n-úplas, porque:

1. $\{tipoMensaje\}$ se puede conocer con sólo analizar el primer carácter de la trama.
2. Conociendo $\{tipoMensaje\}$, se sabe que tiene asociada una n-úpla $\langle x_1, x_2, \dots, x_n \rangle$ es decir, un registro, formado por un número conocido de elementos, donde el i-ésimo elemento de la n-upla señala la longitud del

i-ésimo componente del mensaje, que es una subcadena de la trama que se ha recibido.

3. Conociendo la n-úpla particular que se ha recibido, se procede a analizar la cadena recibida, extrayendo los elementos que la integran, con base en los supuestos antes mencionados.

Si ocurriera que el mensaje consta de una única expresión de caracteres -con o sin parámetros-, podrían eliminarse los campos *{cabecera}* o *{datos}* y reducir el mensaje a *{tipoMensaje}{datos}* o *{tipoMensaje}* respectivamente, sin pérdida de generalidad.

Es importante recalcar que de acuerdo a este modelo, en teoría, se pueden enviar y recibir mensajes simples de longitud arbitraria. Esta solución puede ser utilizada para enviar y recibir mensajes de tipo atómico -un elemento-, como por ejemplo, mensajes de error del servidor a los clientes.

3.5.2. FORMATO DE LOS MENSAJES COMPLEJOS.

Si se necesita enviar o recibir un conjunto de datos más complejo que el analizado previamente, -por ejemplo, un grupo de enunciados que se deben agregar a la consulta de un usuario-, simplemente se definen dos mensajes especiales, que delimitarán a un conjunto arbitrario de tramas que serán enviadas o recibidas, donde cada trama está definida de acuerdo a la solución propuesta en la sección 3.5.1. Es decir, los mensajes complejos responderán al formato que se describe a continuación:

$\{tramaInicial\}$
$\{trama1\}$
$\{trama2\}$
...
$\{trama(n)\}$
$\{tramaFinal\}$

Ilustración 35: Formato de Mensaje Complejo del *FDJ*

Ahora, de acuerdo a este modelo, en teoría, se pueden enviar y recibir mensajes complejos de longitud arbitraria.

A continuación, se plantearán en detalle los formatos de mensaje que se manejarán entre el servidor y los clientes del *FDJ*.

3.5.3. MENSAJES QUE SE ENVÍAN DESDE EL SERVIDOR HACIA LOS CLIENTES.

En esta sección se detallarán los mensajes que se enviarán desde el servidor a los clientes del *FDJ*. En principio se explicará el formato de los mensajes simples, y luego el de los mensajes complejos, con base en los mensajes simples.

Los mensajes simples que se envían del servidor hacia los clientes del foro se definen y explican a continuación, a través de la tabla que se presenta a continuación:

Mensaje	Cadena	Componentes
Tema de Discusión	T/DATA	<ul style="list-style-type: none"> ✓ T: "T"ema de discusión ✓ /: delimitador ✓ DATA: texto del mensaje
Inicio de Mensaje Complejo	I/InicioTrama	<ul style="list-style-type: none"> ✓ I: "I"nicio de una trama ✓ /: delimitador ✓ InicioTrama: texto del mensaje
Reemplazar Enunciados	R/Reemplazar	<ul style="list-style-type: none"> ✓ R: "R"eemplazar los enunciados existentes en el cliente, por los enunciados contenidos en este mensaje, exclusivamente. Este mensaje se utiliza cuando el cliente cambia el tipo de consulta que realiza ✓ /: delimitador ✓ Reemplazar: texto del mensaje
Fin de Mensaje Complejo	I/FinTrama	<ul style="list-style-type: none"> ✓ F: "F"in de trama ✓ /: delimitador ✓ FinTrama: texto del mensaje
Mensajes de la Línea de Estado	M/DATA	<ul style="list-style-type: none"> ✓ M: "M"ensaje enviado ✓ /: delimitador ✓ DATA: Texto del mensaje
Anexar Enunciados	A/Anexar	<ul style="list-style-type: none"> ✓ A: "A"nexar los enunciados contenidos en el mensaje, a los enunciados existentes en el cliente. En caso que algún enunciado se encuentre dentro de la consulta, se sobrescribe. Este mensaje es utilizado cuando un cliente agrega un enunciado y se requiere dicho enunciado aparezca en la consulta del cliente –normalmente, el mismo usuario que lo agregó, o alguno otro que debe observarlo, con base en la consulta que realiza ✓ /: delimitador ✓ Anexar: texto del mensaje

Mensaje	Cadena	Componentes
Sobrescribir Enunciados	S/Sobrescribir	<ul style="list-style-type: none"> ✓ S: "S"obrescribir los enunciados existentes en el árbol, con los enunciados que contenidos en el mensaje. En caso que algún enunciado no se encuentre dentro de la consulta, se anexa. Este mensaje es utilizado cuando un cliente modifica un enunciado y se requiere actualizar las vistas de los usuarios del foro, con los enunciados que resultaron alterados. ✓ /: delimitador ✓ Sobrescribir: texto del mensaje
Enunciado	E/1/2/3/4/5/6/7/8/9/10/11/DATA	<ul style="list-style-type: none"> ✓ E: "E"nunciado ✓ /: delimitador ✓ 1/./11: representan las 11 componentes que integran a un enunciado que se envía a los clientes para integrarlos a sus consultas. ✓ DATA: mensaje que se envía al cliente, está compuesto por la concatenación de las 11 componentes que integran a un enunciado que se envía a los clientes.

Tabla 8: Mensajes Simples que Envía el Servidor a los Clientes del FDJ

Los mensajes complejos que se envían del servidor hacia los clientes del foro se definen en esta sección. Sencillamente consisten de secuencias finitas de mensajes simples, tal como se describe en la sección 3.5.2. El formato general de estos mensajes se describe a continuación:

Reemplazar Enunciados	Sobrescribir Enunciados	Anexar Enunciados
I/InicioTrama	I/InicioTrama	I/InicioTrama
R/Reemplazar	S/Sobrescribir	A/Anexar
E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA
E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA
....
E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA
I/FinTrama	I/FinTrama	I/FinTrama

Tabla 9: Mensajes Complejos que Envía el Servidor a los Clientes del FDJ

Es evidente que la aplicación cliente estará en capacidad de procesar estos tipos de mensajes complejos porque:

- ☑ El cliente recibe cada línea de texto de manera segura (TCP),
- ☑ El cliente es capaz de capturar fácilmente el primer carácter de cada trama,
- ☑ Este carácter puede ser analizado utilizando algún esquema de decisión estructurada tipo “case”,
- ☑ Se pueden almacenar los enunciados recibidos en alguna estructura de datos temporal, tal como un vector.

Al final del proceso, el cliente tiene los enunciados que se le enviaron y conoce la instrucción que se le envió; perfectamente puede realizar algún tipo de procesamiento con los enunciados recibidos.

Con esta exposición se puede considerar resuelto, a nivel de diseño, el problema que representa la comunicación desde Servidor hacia los clientes.

3.5.4. MENSAJES QUE SE ENVÍAN DESDE LOS CLIENTES HACIA EL SERVIDOR.

En esta sección se detallarán los mensajes que se enviarán desde los clientes hacia el servidor del *FDJ*.

Tomando en cuenta el análisis realizado a nivel de especificación, se puede deducir que los clientes del *FDJ* sólo necesitan enviar mensajes simples al

servidor. Los mensajes que se envían desde el cliente hacia el servidor del foro, se definen a través de la tabla que se presenta a continuación:

Mensaje	Cadena	Componentes
Evidencia Asociada de un Enunciado	E/LLAVE	<ul style="list-style-type: none"> ✓ E: “E”videncia asociada de un enunciado ✓ /: delimitador ✓ LLAVE: es la llave o identificador que distingue al enunciado dentro del diálogo.
Cambiar de Consulta	V/DATA	<ul style="list-style-type: none"> ✓ V: “V”ista solicitada ✓ /: delimitador ✓ DATA: nombre de la vista solicitada
Salir de la sesión	S/	<ul style="list-style-type: none"> ✓ S: “S”alir de la sesión ✓ /: delimitador
Agregar Enunciado	A/1/2/3/4/DATA	<ul style="list-style-type: none"> ✓ A: “A”gregar Enunciado ✓ /: delimitador ✓ 1/./4: representan las 4 componentes que integran a un enunciado que se envía desde un cliente hacia el servidor, para agregarlo al diálogo. ✓ DATA: mensaje que se envía al cliente, está compuesto por la concatenación de las 4 componentes que integran a un enunciado que se envía al servidor, para agregarlo al diálogo.
Modificar Enunciado	M/1/2/3/4/5/6/DATA	<ul style="list-style-type: none"> ✓ M: “M”odificar Enunciado ✓ /: delimitador ✓ 1/./6: representan las 6 componentes que integran a un enunciado que se envía desde un cliente hacia el servidor, para modificar al diálogo. ✓ DATA: mensaje que se envía al cliente, está compuesto por la concatenación de las 6 componentes que integran a un enunciado que se envía al servidor, para modificar al diálogo.

Tabla 10: Mensajes que Envían los Clientes al Servidor del FDJ

3.6. DIAGRAMA DE CLASES ESTÁTICAS.

El Diagrama de Emplazamiento, permite modelar el Diagrama de Clases Estáticas, del FDJ, que se presenta en la ilustración siguiente:

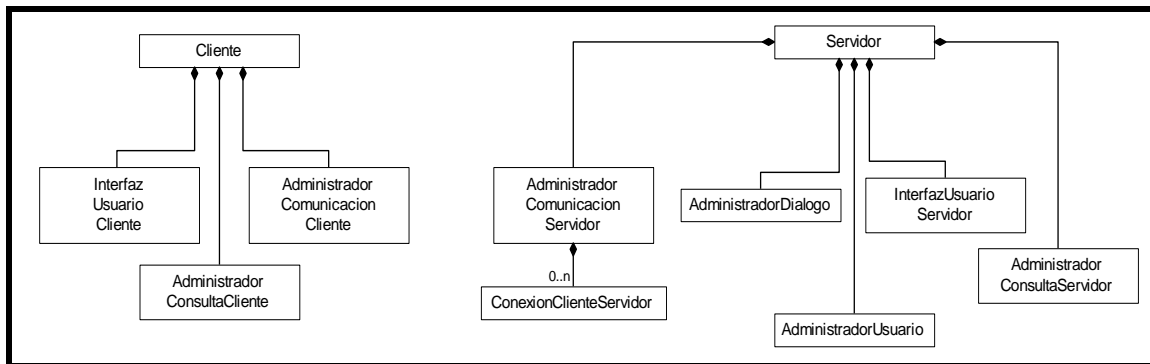


Ilustración 36: Diagrama de Clases Estáticas del FDJ

En lo que resta del capítulo, se procederá a incluir los atributos y métodos, a nivel de implementación, en las clases de este diagrama, para concluir el diseño del *FDJ*.

En tal sentido, es importante que se tenga presente una advertencia acerca de las clases *InterfazUsuarioServidor* e *InterfazUsuarioCliente*: Estas clases no recibirán mayor mención dentro de diseño que se modela en lo que resta del capítulo.

La razón de esta “*omisión*”: Tomando en cuenta los requerimientos particulares a la aplicación que se implementa, observa que estas clases solamente ofrecen mecanismos de entrada / salida de información, entre los usuarios del sistema y las clases que se están modelando -los motores de la aplicación-, entonces:

1. Los atributos que tienen las interfaces representan a los atributos de los motores de la aplicación.
2. Tanto los atributos como los métodos que poseen las interfaces son implementados, en su mayor parte, por las herramientas de “programación visual”, que se utilizan en la actualidad. La intervención humana normalmente se reduce a seleccionar el o los controles que faciliten más la

interacción hombre - máquina y redefinir algunos de sus atributos predeterminados.

3.7. DISTRIBUCIÓN DE ATRIBUTOS EN EL DIAGRAMA DE CLASES.

En esta sección se redistribuyen los atributos presentes del diagrama de paquetes, y en el diagrama de emplazamiento, dentro del diagrama de clases estáticas.

Además, se incluyen atributos que responden a las necesidades de la implementación, en especial, los que tienen relación con los objetos XML. Esta redistribución se modela a través de las tablas que se presenta a continuación:

Atributo	Alcance	Tipo	Explicación
documento	privado	Document	Representa una instancia del documento XML en memoria, que a su vez contiene a los elementos estructurales del documento -nodos-. Utilizada al guardar e imprimir el documento XML-DOM.
indentacion	privado	Int	Representa la cantidad de espacios en blanco que se dejará entre cada nivel de indentación, al imprimir o guardar el documento XML-DOM, para facilitar la comprensión de la jerarquía.
listaDeUsuarios	privado	Element	Representa al elemento raíz de la jerarquía del documento XML-DOM. Utilizándolo como base, se puede localizar a cualquier usuario y sus atributos.
usuariosRegistrados	privado	int	Representa la cantidad de usuarios que se encuentran registrados en la lista -activos e inactivos-.

Tabla 11: Atributos de la Clase *AdministradorUsuario* del Servidor en el Diagrama de Clase del FDJ

Atributo	Alcance	Tipo	Explicación
dialogo	privado	Element	Representa al elemento que contiene el dialogo que se desarrolla dentro del foro. Utilizándolo como base, se puede localizar cualquier enunciado dentro de la jerarquía.
dialogoDelForo	privado	Element	Representa al elemento raíz de la jerarquía del documento XML-DOM. Utilizándolo como base, se puede localizar cualquier otro elemento dentro de la jerarquía, como enunciados, marcas, umbrales o el tema de discusión.
documento	privado	Document	Representa una instancia del documento XML en memoria, que a su vez contiene a los elementos estructurales del documento -nodos-. Utilizada al guardar e imprimir el documento XML-DOM.
indentacion	privado	int	Representa la cantidad de espacios en blanco que se dejará entre cada nivel de indentación, al imprimir o guardar el documento XML-DOM, para facilitar la comprensión de la jerarquía.
ma	privado	double	Representa el valor que asumirá la marca de aceptación para una sesión particular del foro.
mr	privado	double	Representa el valor que asumirá la marca de rechazo para una sesión particular del foro.
temaADiscutir	privado	Element	Representa al elemento que contiene el tema que se discute dentro del foro.
ua	privado	double	Representa el valor que asumirá el umbral de aceptación para una sesión particular del foro.
ur	privado	double	Representa el valor que asumirá el umbral de rechazo para una sesión particular del foro.

Tabla 12: Atributos de la Clase *AdministradorDialogo* del Servidor en el Diagrama de Clase del FDJ

Atributo	Alcance	Tipo	Explicación
documento	privado	Document	Representa una instancia del documento XML en memoria, que a su vez contiene a los elementos estructurales del documento -nodos-. Utilizada al guardar e imprimir el documento XML-DOM.
indentacion	privado	int	Representa la cantidad de espacios en blanco que se dejará entre cada nivel de indentación, al imprimir o guardar el documento XML-DOM, para facilitar la comprensión de la jerarquía.

Atributo	Alcance	Tipo	Explicación
vistasDeUsuarios	privado	Element	Representa al elemento raíz de la jerarquía del documento XML-DOM. Utilizándolo como base, se puede localizar cualquier otro elemento dentro de la jerarquía, como la vista de un usuario particular, y cualquiera de sus atributos.

Tabla 13: Atributos de la Clase *AdministradorConsultaServidor* en el Diagrama de Clase del FDJ

Atributo	Alcance	Tipo	Explicación
direccionIP	público	String	Representa la dirección DNS o IP del servidoresistema que está conectado al socket
estadoServidor	público	int	Representa la disponibilidad del servidor a recibir conexiones (0 = servidor desactivado, 1 = servidor activado)
idConexion	privado	int	Representa el identificador del hilo de ejecución asignado a un cliente particular
marcaAceptacion	privado	double	Representa el valor que asumirá la marca de aceptación para una sesión particular del foro.
marcaRechazo	privado	double	Representa el valor que asumirá la marca de rechazo para una sesión particular del foro.
maximoConexiones	público	int	Representa la cantidad máxima de usuarios activos que pueden presentarse concurrentemente
nombreSistema	público	String	Representa el nombre del sistema donde se ejecuta el servidor
puertoConexion	público	int	Representa el puerto de comunicaciones que se utilizará al momento de establecer la conexión entre el cliente y el servidor
servidorSocket	privado	ServerSocket	Representa el servidor de conexiones (Socket de Flujo, orientado a conexión)
temaADiscutir	privado	Element	Representa al elemento que contiene el tema que se discute dentro del foro.
umbralAceptacion	privado	double	Representa el valor que asumirá el umbral de aceptación para una sesión particular del foro.

Atributo	Alcance	Tipo	Explicación
umbralRechazo	privado	double	Representa el valor que asumirá el umbral de rechazo para una sesión particular del foro.

Tabla 14: Atributos de la Clase *AdministradorComunicacionServidor* en el Diagrama de Clase del *FDJ*

Atributo	Alcance	Tipo	Explicación
buffer	privado	String	Representa la cadena de texto que se recibe del cliente y debe ser procesada
direccionUsuario	privado	String	Representa la dirección DNS o IP del sistema que está conectado al socket
hilo	privado	Thread	Representa el hilo de ejecución asignado a un cliente particular
idHilo	privado	int	Representa el identificador del hilo de ejecución asignado a un cliente particular
nombreUsuario	privado	String	Representa el nombre del usuario que está conectado al socket
recibido	privado	BufferedReader	Representa el flujo de datos que entra al servidor, desde el sistema del cliente
salida	privado	DataOutputStream	Representa el flujo de datos que sale del servidor, y se dirige hacia el sistema del cliente
socket	privado	Socket	Representa la conexión asignada a un cliente particular (Socket de Flujo, orientado a conexión)

Tabla 15: Atributos de la Clase *ConexionClienteServidor* del Servidor en el Diagrama de Clase del *FDJ*

Atributo	Alcance	Tipo	Explicación
actualizarArbol	privado	boolean	Representa una bandera, que indica cuando se debe actualizar el árbol que observa el usuario
consultaDelUsuario	privado	Element	Representa al elemento raíz de la jerarquía del documento XML-DOM. Utilizándolo como base, se puede localizar cualquier otro elemento dentro de la jerarquía, como enunciados, Tema a discutir, tipo de consulta.
dialogo	privado	Element	Representa al elemento que contiene la consulta que se realiza al dialogo del foro. Utilizándolo como base, se puede localizar cualquier enunciado dentro de la jerarquía.
documento	privado	Document	Representa una instancia del documento XML en memoria, que a su vez contiene a los elementos estructurales del documento -nodos-. Utilizada al guardar e imprimir el documento XML-DOM.

Atributo	Alcance	Tipo	Explicación
indentacion	privado	int	Representa la cantidad de espacios en blanco que se dejará entre cada nivel de indentación, al imprimir o guardar el documento XML-DOM, para facilitar la comprensión de la jerarquía.
temaADiscutir	privado	Element	Representa al elemento que contiene el tema que se discute dentro del foro.
tipoConsulta	privado	Element	Representa al elemento que contiene el tipo de consulta que realiza el usuario

Tabla 16: Atributos de la Clase *AdministradorConsultaCliente* del Servidor en el Diagrama de Clase del FDJ

Atributo	Alcance	Tipo	Explicación
buffer	privado	String	Representa la cadena de texto que se recibe del cliente y debe ser procesada
clienteSocket	privado	Socket	Representa la conexión del cliente con el servidor (Socket de Flujo, orientado a conexión)
direccionIPServidor	privado	String	Representa la dirección DNS o IP del servidor al que se conecta el cliente
estadoCliente	público	int	Representa la disponibilidad de la conexión que ofrece el servidor (0 = servidor desactivado, 1 = servidor activado)
nombreUsuario	público	String	Representa el nombre del usuario se une a la sesión
procesarMensajesComplejos	privado	boolean	Representa una bandera que señala cuando se ha terminado de recibir los mensajes complejos
puertoConexionServidor	Privado	int	Representa el puerto de comunicaciones que se utilizará al momento de establecer la conexión con el servidor
recibido	Privado	BufferedReader	Representa el flujo de datos que entra desde servidor
salida	Privado	DataOutputStream	Representa el flujo de datos que sale del cliente, y se dirige hacia el servidor del foro
temaADiscutir	Privado	String	Representa al tema que se discute dentro del foro.

Atributo	Alcance	Tipo	Explicación
mensajeComplejo	Privado	Vector	Representa a los mensajes complejos que se reciben del servidor

Tabla 17: Atributos de la Clase *AdministradorComunicacionCliente* del Servidor en el Diagrama de Clase del *FDJ*

Ahora, reuniendo los atributos modelados en las tablas previamente mostradas, con el diagrama de clases estáticas de la Ilustración 36, se obtiene un diagrama de clases estáticas que incluye los atributos del *FDJ* y se muestra en la siguiente ilustración:

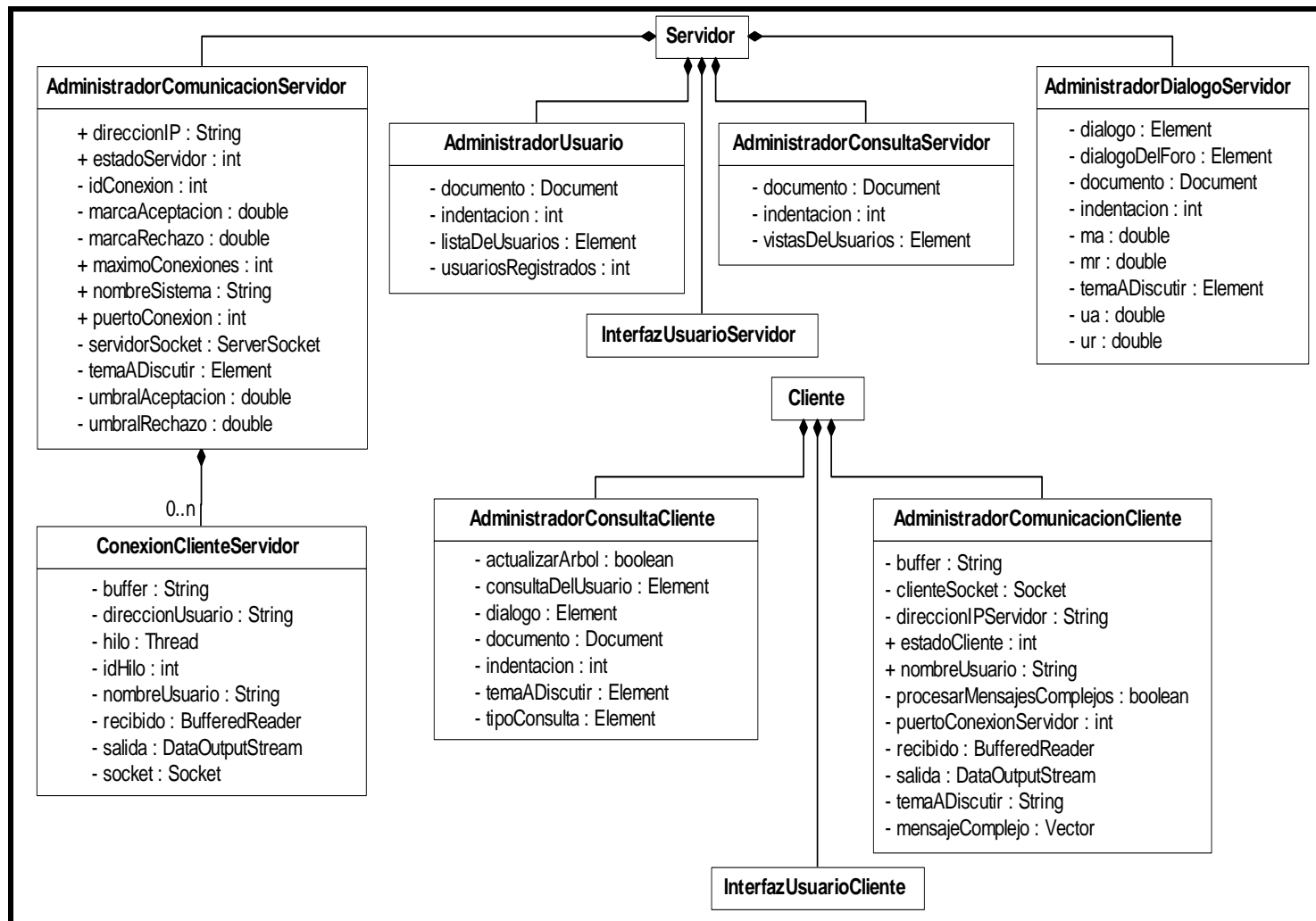


Ilustración 37: Diagrama de Clases Estáticas del *FDJ* con Atributos

3.8. ADICIÓN DE MÉTODOS AL DIAGRAMA DE CLASES ESTÁTICAS.

Los métodos más importantes de cada una de las clases del diagrama de clases estáticas, se explican, a continuación, a través de tablas.

Método	Alcance	Tipo	Explicación
actualizarDialogoMensaje(Vector vectorMensajes)	público	Void	Transforma los mensajes que se reciben del servidor, en órdenes que alterarán la estructura de la consulta del cliente, agregando, modificando o reemplazando enunciados.
AdministradorConsultaCliente()	público	Constructor	Crea instancias de la clase, sin alterar su estado.
AdministradorConsultaCliente(String tema, int indentacion)	público	Constructor	Crea instancias de la clase, inicializando sus parámetros.
agregarEnunciado(String llavePadre, String naturalezaEnunciado, String textoEnunciado, String nombreUsuario, String fecha, String hora, String plausibilidad, String color, String apoyo, String evidenciaVacía, String naturalezaAnterior)	público	Int	Agrega enunciados a la consulta del cliente y retorna un código de error, dependiendo si se acepta o no la adición del enunciado.
asignarEstadoActualizarArbol(boolean estadoConsulta)	público	Void	Altera el estado de la bandera de actualización del árbol en el cliente.
asignarTipoConsulta(String tipoConsulta)	público	Void	Cambia el tipo de consulta que realiza el cliente
esEnunciadoRegistrado(String llave)	público	Boolean	Comprueba si un enunciado dado está incluido en la consulta del cliente.
guardarArchivo(String nombreArchivo)	público	Void	Guarda la consulta del usuario como un archivo XML, con el nombre propuesto como parámetro.
modificarEnunciado(String llaveEnunciado, String naturalezaEnunciado, String textoEnunciado, String nombreUsuario, String fechaEnunciado, String horaEnunciado, String plausibilidadEnunciado, String colorEnunciado, String apoyoEnunciado, String evidenciaVacía, String naturalezaAnterior)	público	Int	Modifica enunciados en la consulta del cliente y retorna un código de error, dependiendo si se acepta o no la modificación del enunciado.
obtenerArbolJTree(JTree arbol)	público	Jtree	Genera un árbol JTree que representa a la consulta del cliente
obtenerCantidadDeHijos(Node enunciado)	público	int	Permite obtener la cantidad de hijos que forman la evidencia asociada de un enunciado.
obtenerEstadoActualizarArbol()	público	boolean	Permite determinar si se debe actualizar el árbol de la consulta del cliente

Método	Alcance	Tipo	Explicación
obtenerllavePadre(String llaveHijo)	público	String	Permite obtener la llave del padre de un enunciado, dada la llave del enunciado hijo.
obtenerMensajeAgregarEnunciado(String llavePadre, String textoEnunciado, String nombreUsuario, String naturalezaEnunciado)	público	String	Permite recuperar una cadena correspondiente al mensaje de una orden para agregar un enunciado al dialogo, lista para ser enviada al servidor del foro
obtenerMensajeModificarEnunciado(String llaveEnunciado, String textoEnunciado, String nombreUsuario, String naturalezaEnunciado, boolean retractar, boolean reactivar)	público	String	Permite recuperar una cadena correspondiente al mensaje de una orden para modificar un enunciado del dialogo, lista para ser enviada al servidor del foro
obtenerNodoBaseEnunciado(String llaveEnunciado)	público	Node	Permite recuperar el nodo base, dentro de la estructura XML-DOM, de un enunciado, dada la llave que lo identifica.
obtenerTipoConsulta()	público	String	Permite recuperar el tipo de consulta que efectúa el cliente.
obtenerEvidenciaAsociada(Element enunciado)	público	Element	Permite recuperar la evidencia asociada de un enunciado, como un elemento, que contiene a todos los enunciados hijos, recursivamente.

Tabla 18: Métodos de la Clase *AdministradorConsultaCliente* del *FDJ*

Método	Alcance	Tipo	Explicación
AdministradorComunicacionCliente()	público	Constructor	Permite que se declaren instancias de la clase sin alterar sus atributos.
AdministradorComunicacionCliente(String dir_server, int port_server, String nombre)	público	Constructor	Permite declarar instancias de la clase inicializando la comunicación con el servidor.
cerrar_socket()	privado	void	Cierra la conexión socket con el servidor.
conectar(String dir_ser, int port)	privado	void	Establece la comunicación vía socket con el servidor
desconectar()	público	void	Elimina la comunicación vía socket con el servidor.
enviar(String cad)	público	boolean	Envía las cadenas al servidor, que contienen las órdenes que anota el cliente.
refresh(String S)	privado	void	Escucha al servidor, esperando recibir sus mensajes, los analiza, y envía las órdenes recibidas a las clases que lo ameriten.
repintarArbol(JScrollPane panel)	público	JScrollPane	Retorna un JScrollPane que muestra al usuario la consulta que realiza.

Tabla 19: Métodos de la Clase *AdministradorComunicacionCliente* del *FDJ*

Método	Alcance	Tipo	Explicación
AdministradorUsuario(int tabulacion)	público	Constructor	Permite declarar instancias de la clase, inicializando sus atributos.
AdministradorUsuario()	público	Constructor	Permite declarar instancias de la clase, sin alterar sus atributos.

Método	Alcance	Tipo	Explicación
agregarUsuario(String nombrePropuesto, String hostPropuesto)	público	boolean	Permite agregar usuarios a la lista, retorna true si el usuario se agregó exitosamente, o false, en caso que no se pueda agregar.
borrarUsuario(String nombre)	público	boolean	Permite desactivar usuarios de la lista, retorna true en caso que se pueda desactivar exitosamente al usuario, o false en caso que no se pueda desactivar
esUsuarioActivo(String nombre)	público	boolean	Permite determinar si existe algún usuario activo en la sesión, registrado bajo el nombre propuesto
esUsuarioRegistrado(String nombre)	público	boolean	Permite determinar si existe algún usuario registrado en la sesión, bajo el nombre propuesto, la búsqueda incluye a los usuarios desactivados.
guardarArchivo(String nombreArchivo)	público	void	Permite guardar la lista de usuarios en un archivo con formato XML.
obtenerCantidadDeUsuariosActivos()	público	int	El nombre del método se explica por si solo.
obtenerCantidadDeUsuariosRegistrados()	público	int	El nombre del método se explica por si solo.
obtenerIdUsuario(String nombreBuscado)	público	int	Permite recuperar el identificador de usuario, dado el nombre de un usuario, en caso que no exista el usuario dentro de la sesión, retorna 0.
obtenerListaUsuariosSwing()	público	JList	Permite recuperar un lista JList que contiene a los usuarios registrados en la sesión, activos y desactivados.

Tabla 20: Métodos de la Clase *AdministradorUsuario* del *FDJ*

Método	Alcance	Tipo	Explicación
actualizarDialogoTrama(String trama)	público	String	Permite transformar los mensajes que se reciben de los clientes, en órdenes que alterarán la estructura del diálogo del servidor, agregando o modificando enunciados.
AdministradorDialogo(String tema, double ma, double mr, double ua, double ur, int tabulacion)	público	Constructor	Permite declarar instancias de la clase inicializando sus atributos internos.
AdministradorDialogo()	público	Constructor	Permite declarar instancias de la clase sin inicializar sus atributos internos.
agregarEnunciado(String llavePadre, String textoEnunciado, String nombreUsuario, String naturalezaEnunciado)	público	int	Permite agregar enunciados al diálogo del servidor. El enunciado se agrega basándose en la llave del enunciado padre.
calcularPlausibilidadHilo(String llave)	público	void	Permite calcular, en cascada, el estado de plausibilidad de un hilo de discusión.
esEnunciadoRegistrado(String llaveEnunciado)	público	boolean	Determina si un enunciado, definido por su llave, se encuentra registrado o no en el diálogo.
guardarArchivo(String nombreArchivo)	público	void	Permite guardar el diálogo realizado en la sesión, en un archivo XML.

Método	Alcance	Tipo	Explicación
modificarEnunciado(String llaveEnunciado, String textoEnunciado, String nombreUsuario, String naturalezaEnunciado, boolean retractar, boolean reactivar)	público	int	Permite modificar un enunciado del diálogo, con base en su llave.
obtenerCantidadDeHijos(Node enunciado)	público	int	Permite obtener la cantidad de enunciados que forman la evidencia asociada de otro enunciado.
obtenerEnunciadosAlterados(String llaveEnunciado)	público	Vector	Permite obtener un vector que contiene los enunciados que resultan alterados al realizarse la adición o modificación de un enunciado definido por llaveEnunciado.
obtenerEvaluarAgregarEnunciado(int codigo)	público	String	Permite evaluar los códigos numéricos que se recuperan al agregar enunciados y recuperar el mensaje correspondiente.
obtenerEvaluarModificarEnunciado(int codigo)	público	String	Permite evaluar los códigos numéricos que se recuperan al modificar enunciados y recuperar el mensaje correspondiente.
obtenerEvidenciaAsociada(Element enunciado)	público	Element	Permite recuperar la evidencia asociada de un enunciado, como un elemento del documento XML, que incluye a todos los enunciados hijos, en cascada.
obtenerLlaveEvidenciaAgregada(String llavePadre)	público	String	Permite recuperar el valor que se asignará a la llave de un enunciado que se agrega al foro, como evidencia asociada de otro enunciado.
obtenerNodoBaseEnunciado(String llaveEnunciado)	público	String	Permite recuperar el nodo base de la estructura XML, correspondiente a la llave de un enunciado previamente registrado.

Tabla 21: Métodos de la Clase *AdministradorDialogoServidor* del *FDJ*

Método	Alcance	Tipo	Explicación
AdministradorConsultaServidor()	público	Constructor	Permite declarar instancias de la clase, sin inicializar sus atributos.
AdministradorConsultaServidor(int tabulacion)	público	Constructor	Permite declarar instancias de la clase, inicializando sus atributos.
agregarConsultaDeUsuario(int idUsuario)	público	boolean	Agrega la consulta del usuario identificado por idUsuario.
agregarEnunciadoAConsulta(int idUsuario, String llaveEnunciado)	privado	void	Permite agregar un enunciado a la consulta de un usuario.
borrarConsultaDeUsuario(int idusuario)	público	void	Elimina la consulta del usuario identificado por idusuario.
esConsultaDeUsuarioValida(int idusuario)	público	boolean	Permite determinar si el usuario idusuario posee una consulta.
esEnunciadoIncluido(int idUsuario, String llave)	público	boolean	Permite determinar si el enunciado definido por llave se encuentra incluido en la consulta del usuario identificado por idUsuario.
guardarArchivo(String nombreArchivo)	público	void	Permite guardar las consultas en un archivo XML.
modificarAtributoConsultaActualizada(int idUsuario, boolean estado)	público	void	Permite establecer que la consulta del usuario idUsuario esta actualizada o no.

Método	Alcance	Tipo	Explicación
modificarConsultaUsuario(int idUsuario, String tipoConsulta)	público	void	Permite modificar el tipo de consulta que observa un usuario dado.
obtenerTipoConsulta(int idUsuario)	público	String	Permite conocer el tipo de consulta que realiza un usuario dado.
obtenerTramaEnunciado(String llaveEnunciado)	público	String	Permite recuperar la trama que se enviará a un usuario, para que incluya en su consulta local, a un enunciado.
obtenerVectorTramasClientes(Vector vectorEnunciados, int idUsuario, String llave)	público	Vector	Permite actualizar las vistas de los usuarios dentro del servidor, y retorna un vector con los mensajes que se debe enviar a cada usuario, para que se actualice su consulta, con los datos de refresco del servidor.

Tabla 22: Métodos de la Clase *AdministradorConsultaServidor* del FDJ

Método	Alcance	Tipo	Explicación
cerrarSocket()	público	void	Cierra el socket de un cliente.
cerrarThread()	público	void	Cierra el hilo en que se ejecuta una conexión de un cliente.
ConexionClienteServidor (int indice, Socket socket, String dir_conex, int act)	público	Constructor	Crea una conexión que escuchará a un cliente dado, a través de un socket, y una dirección de conexión.
enviar_cad(String cad_usu)	público	void	Envía un mensaje al cliente que tiene asignada la conexión.
refresh(String S)	público	void	Escucha los mensajes que envía el cliente que tiene asignada la conexión.

Tabla 23: Métodos de la Clase *ConexionClienteServidor* del FDJ

Método	Alcance	Tipo	Explicación
AdministradorComunicacionServidor()	público	Constructor	Crea una instancia de la clase sin inicializar los atributos.
AdministradorComunicacionServidor(int puerto, String direccionIP)	público	Constructor	Crea una instancia de la clase inicializando sus atributos.
BajarServidor()	público	boolean	Desactiva el servidor.
cerrar_Server_socket()	público	void	Cierra el servidor de socket.
getDir(ServerSocket s)	público	String	Recupera la dirección IP del sistema donde se aloja el servidor.
getName(ServerSocket s)	público	String	Recupera el nombre del sistema donde se aloja el servidor del foro.
Levanta (int port)	público	boolean	Activa el servidor, en el puerto indicado.
mandaMensajeUno(String cad, int idUsuario)	público	void	Envía un mensaje al usuario señalado por idUsuario.

Tabla 24: Métodos de la Clase *AdministradorComunicacionServidor* del FDJ

Dada la cantidad de métodos presentes en cada una de las clases que componen el diagrama de clases estáticas, se ha segmentado en dos partes, Cliente y Servidor, que se presentan en las siguientes ilustraciones que se muestran a continuación:

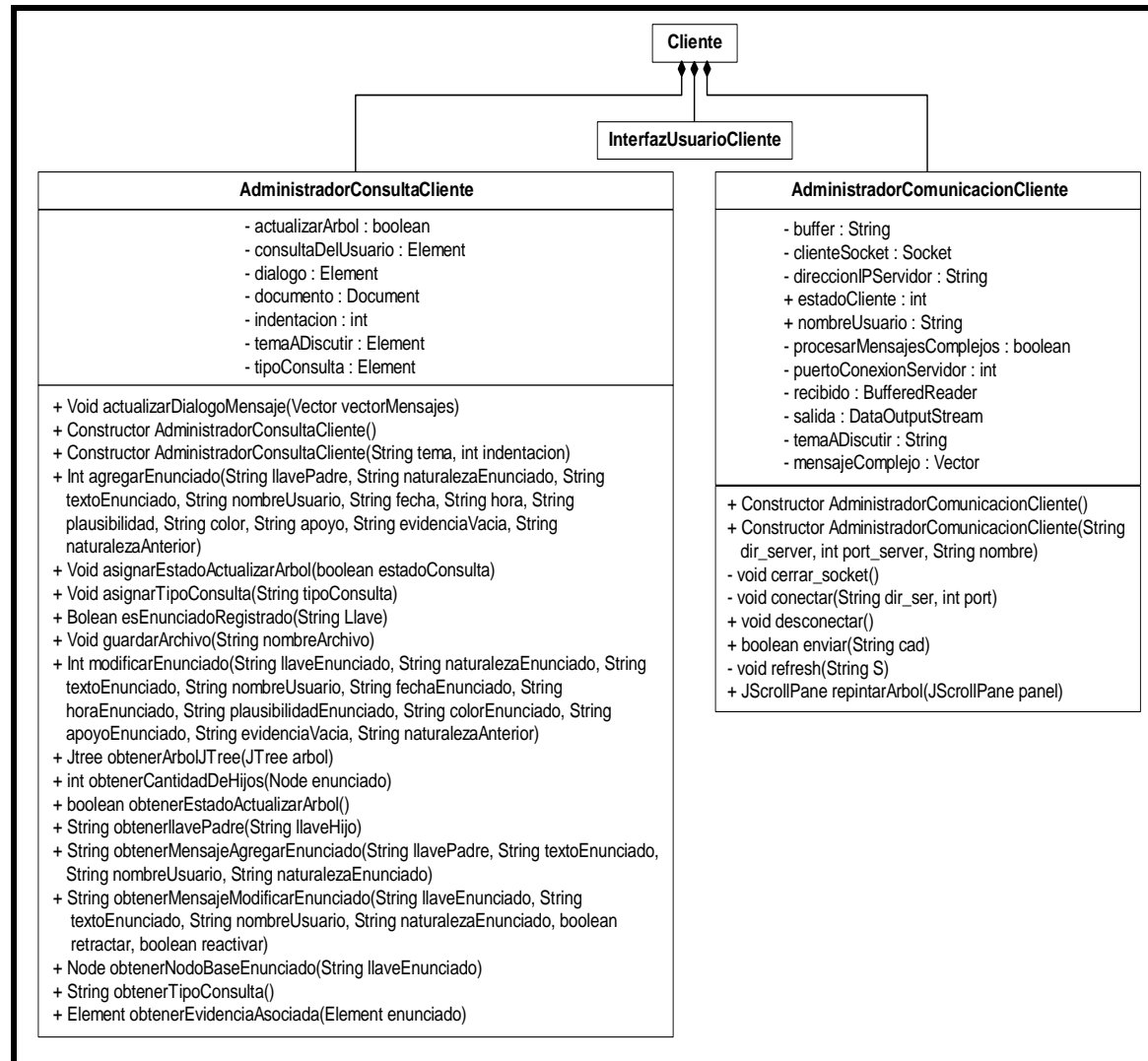


Ilustración 38: Adición de Métodos al Diagrama de Clases Estáticas del *FDJ* (Cliente)

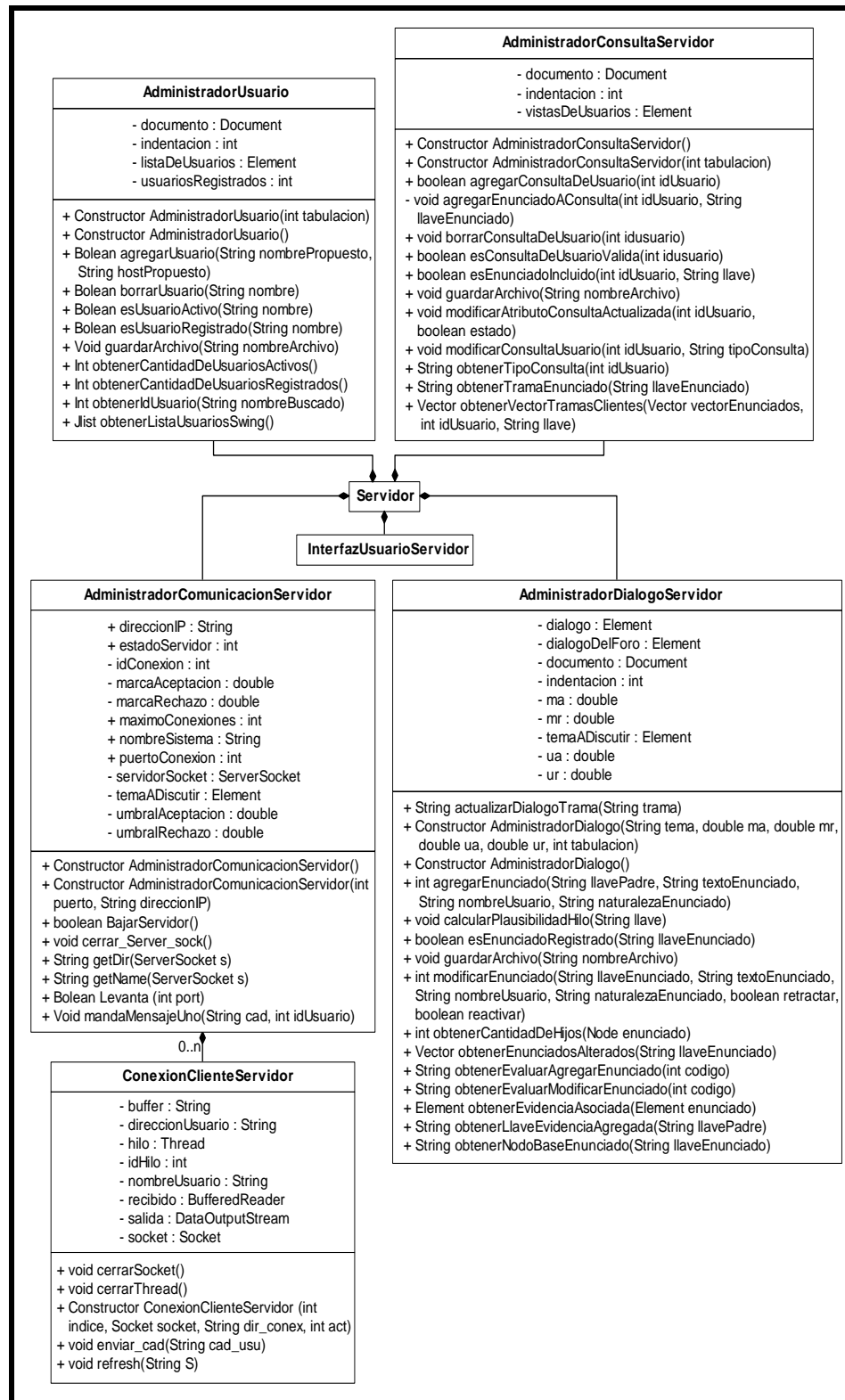


Ilustración 39: Adición de Métodos al Diagrama de Clases Estáticas del *FDJ* (Servidor)

4. VALIDACIÓN DEL SISTEMA, CONTRIBUCIONES, CONCLUSIONES Y TRABAJOS DERIVADOS.

Este capítulo presenta la culminación de esta investigación. En tal sentido:

1. Se presenta la validación del sistema contra la formalización planteada en [NÚÑE99],
2. Se exponen las contribuciones realizadas por esta investigación,
3. Se formulan las conclusiones principales que sintetizan a esta investigación,
4. Se sugieren algunos trabajos adicionales que pueden realizarse como productos derivados de esta investigación.

4.1. VALIDACIÓN DEL SISTEMA CONTRA LA ESPECIFICACIÓN DEL *FDJ*.

La validación del sistema implementado frente a la formalización de [NÚÑE99], se realizó esencialmente, sobre la clase *AdministradorDialogo*. La validación del sistema se modeló sobre la *AdministradorDialogo* porque:

1. Es en la clase *AdministradorDialogo* donde se almacena el diálogo que desarrollan los usuarios del foro y donde se calcula el estado de plausibilidad de los hilos de discusión del diálogo.

2. Por la forma en que actúa la combinación de algoritmos que calcula el estado de plausibilidad de los hilos de discusión, existen situaciones que se pueden presentar en la clase *AdministradorDialogo* que no se pueden presentar dentro del software real. Por ejemplo: cuando el análisis del estado de plausibilidad de un enunciado lo califica como *Dudoso*, el enunciado es desactivado. En consecuencia, no se le puede agregar evidencia adicional, es decir, no podrá llegar al estado *Insatisfactorio*. Sin embargo, un enunciado puede “mejorar” su estado de plausibilidad (pasar de aceptable a satisfactorio, por ejemplo), por efecto de la inclusión de nuevos usuarios en la sesión.

Para realizar la validación antes mencionada se procedió de acuerdo a la metodología que se describe a continuación:

- 1 Se plantearon algunos ejemplos ficticios del diálogo basados en los ejemplos presentados por **[NÚÑE99]**,
- 2 Para cada ejemplo se calculó, manualmente, el apoyo y el estado de plausibilidad de cada enunciado dentro de los hilos de discusión del diálogo.
- 3 Se procedió a crear una aplicación Java, sin interfaz. En esta aplicación se establecen una serie de instrucciones que realizan “directamente”⁶ las acciones que se explican a continuación:

⁶ *Directamente* se debe entender en el contexto de que son instrucciones que se ejecutan sin la intervención del usuario, de manera similar a como

- 3.1 Se declararon instancias de las clases *AdministradorUsuario* y *AdministradorDialogo* del servidor del foro.
 - 3.2 A la clase *AdministradorUsuario* se le agregaron los usuarios que participarían en el diálogo.
 - 3.3 A la clase *AdministradorDialogo* se le agregaron los enunciados que forman los hilos de discusión del diálogo, siguiendo la secuencia natural de adición; es decir, se agregó a los enunciados padres, antes que los enunciados hijos.
 - 3.4 Cuando se terminó de incluir los enunciados que forman el ejemplo, se ordenó a la clase *AdministradorDialogo* que calculara el estado de plausibilidad de cada uno de los hilos de discusión que integran el diálogo en ese momento.
 - 3.5 Luego se ordenó a la clase *AdministradorDialogo* que guardara el diálogo desarrollado en un archivo XML.
- 4 Finalmente, se comparó, manualmente, los datos contenidos en el archivo XML generado por la aplicación de prueba, contra los datos obtenidos manualmente.

actúan los “scripts” de comandos bajo Unix o los archivos “.bat” bajo DOS / Windows.

En todos los casos, los resultados obtenidos con el software coincidieron “razonablemente bien”⁷ con los resultados obtenidos manualmente.

4.2. CONTRIBUCIONES REALIZADAS POR ESTA INVESTIGACIÓN.

Haciendo una comparación de los objetivos propuestos en el anteproyecto de esta investigación, frente a los resultados alcanzados al culminar el proyecto, se obtienen las contribuciones que se mencionan a continuación:

1. **La especificación del sistema modelada con *UML*:** Porque se analizaron los requerimientos de plausibilidad que debe cumplir el foro, en el capítulo 2 de esta investigación.
2. **La arquitectura del sistema modelada con *UML*:** En efecto, el sistema se modeló utilizando *UML* y el paradigma de orientación a objetos, en el desarrollo de los capítulos 2 y 3.
3. **Un modelo alternativo de identificación de enunciados, dentro del *FDJ*:** En esta investigación se propone, diseña e implementa un mecanismo de identificación de enunciados dentro del diálogo desarrollado en los hilos de discusión, alternativo al utilizado por [NÚÑE99], en la sección 3.3.

⁷ *Razonablemente bien* se debe entender en el contexto de que las diferencias que se presentaron en la comparación no son significativas. Para los efectos de la investigación, por ejemplo, se considera que 0.6665 es una buena aproximación de $\frac{2}{3}$.

4. **El modelo de almacenamiento de información del *FDJ* basado en XML:**
Porque la información más importante dentro del foro -usuarios registrados, diálogo desarrollado y consultas de los clientes-, se almacena utilizando estructuras que están diseñadas e implementadas utilizando XML, en la sección 3.4.
5. **Un modelo de comunicación de mensajes complejos bajo TCP, basado en caracteres:** En efecto, se diseñó e implementó el esquema de comunicación que utilizan las aplicaciones cliente / servidor, en la sección 3.5.
6. **El código ejecutable del sistema con su manual de usuario:**
Finalmente, se obtiene el código ejecutable, correspondiente a las aplicaciones cliente / servidor que implementan al foro de discusión jerárquico, adjunta a este documento, en el CD que lo acompaña; con el manual de usuario correspondiente.

4.3. **CONCLUSIONES OBTENIDAS DE ESTA INVESTIGACIÓN.**

Al completar este proyecto de investigación, se obtienen las conclusiones que se plantean a continuación:

3. Se implementó, un prototipo funcional del *FDJ* propuesto por [NÚÑE99], que materializa las características más esenciales de esta formalización. Esta implementación se modeló utilizando el metalenguaje UML, como herramienta de especificación y diseño. Además, utiliza el metalenguaje XML como estructura principal de almacenamiento de información volátil y persistente. El *FDJ* también representa la implementación de uno de los

modelos de votación propuestos por [OLIV96]. Se observa, entonces, como una serie de trabajos teóricos, previos a esta investigación, se han implementado al culminar este proyecto. En consecuencia, este proyecto se coloca como la culminación de una etapa en el proceso de investigación teórico de la *Teoría de Plausibilidad*.

4. Esta investigación tiene especial relevancia, porque el prototipo de sistema obtenido, es razonablemente funcional, además de ofrecer un medio alternativo a la realización de reuniones físicas. Es una herramienta que involucra al proceso de votación con el desenvolvimiento del diálogo jerárquico –al dialogar bajo un esquema jerárquico, implícitamente se vota–.
5. El sistema implementado representa un aporte importante en las áreas de tele trabajo y tele enseñanza en términos de que posibilita la realización de reuniones electrónicas y de procesos de toma de decisiones en grupo, desde una perspectiva novedosa: la jerarquización de la discusión, a la vez que ésta se realimenta con una medida cuantitativa de la aceptabilidad que tiene cada enunciado dentro del diálogo. Específicamente se ha logrado:
 - 5.1. Controlar el problema representado por la distancia entre participantes, al “reunirlos” en un ambiente donde pueden dialogar en tiempo real,
 - 5.2. Facilitar la comprensión del diálogo entre participantes, con la jerarquización de la conversación,
 - 5.3. Cuantificar la aceptabilidad de cada enunciado que se expresa en el diálogo, con base a la Teoría de la Plausibilidad.

6. La documentación incluida dentro del informe final de esta investigación modela los componentes fundamentales de una aplicación Cliente / Servidor típica. Esta documentación puede ser utilizada como base de la especificación, diseño e implementación de otras aplicaciones Cliente / Servidor, realizando las adecuaciones pertinentes de acuerdo al problema particular que se desea resolver.
7. Resulta relevante observar como se puede modelar el proceso de especificación y diseño de aplicaciones utilizando UML. Es importante percibir que *“sólo se deben crear los diagramas que aportan información significativa al modelo”*. En el caso particular de esta implementación, se logró comprender el comportamiento y la estructura del sistema, utilizando:
 - 7.1. Los diagramas de casos de uso,
 - 7.2. Los diagramas de colaboración,
 - 7.3. El modelo conceptual,
 - 7.4. El diagrama de paquetes,
 - 7.5. El diagrama de emplazamiento,
 - 7.6. El diagrama de clases estáticas.

Estos diagramas, aunados a la comprensión cabal de la formalización que fundamenta esta investigación, han permitido realizar el proceso de codificación del sistema de una forma relativamente directa.

8. Adicionalmente, el prototipo del *FDJ* ofrece una herramienta de investigación teórica; porque la formalización planteada por [NÚÑE99] no ha sido aplicada a nivel práctico. En consecuencia, la comparación del modelo teórico contra una primera implementación práctica, puede aportar nuevos elementos de juicio y experiencias valiosas que podrían realimentar el proceso de investigación teórica.

4.4. TRABAJOS QUE SE PUEDEN DERIVAR DE ESTA INVESTIGACIÓN.

1. El *FDJ* implementa un subconjunto de la formalización planteada en [NÚÑE99]. En consecuencia, puede ser objeto de una serie de mejoras que se describen a continuación:
 - 1.1. **Incluir un módulo de votaciones fundamentado en la Teoría de Plausibilidad:** Este módulo de votaciones está considerado en la formalización de [NÚÑE99].
 - 1.2. **Incluir el intercambio de datos multimedia:** Adicionalmente al intercambio de mensajes basados en texto, la formalización de [NÚÑE99] contempla que el foro de discusión puede permitir el intercambio de mensajes que contengan tanto texto plano como información multimedia -imágenes, sonidos, animaciones, entre otras posibilidades-.
 - 1.3. **Ampliar los tipos de consultas disponibles:** La implementación actual del *FDJ* permite que los usuarios consulten el diálogo desarrollado bajo dos modalidades básicas: “*Ver Opiniones Primarias*”, y “*Ver Todo*”. Sin embargo, la formalización de

[NÚÑE99] contempla que el foro de discusión puede permitir otros tipos de consultas, tales como: “*Ver Lo Leído*”, “*Ver Lo No Leído*”, “*Ver Enunciados Aclaración*”, “*Ver Enunciados A Favor*”, “*Ver Enunciados En Contra*”.

- 1.4. **Ampliar la cobertura de las operaciones deshacer / rehacer del *FDJ*:** El prototipo del *FDJ* considera un único nivel para cada una de estas operaciones.
- 1.5. Se puede mejorar la interfaz de la aplicación, para que tenga el acabado de una aplicación más depurada.
- 1.6. Se puede mejorar el modelo de colaboración utilizado en el prototipo, al otorgar al administrador del servidor la posibilidad de administrar a los usuarios registrados en el *FDJ* -desactivar a los usuarios problemáticos, por ejemplo-. Además, se puede incluir la posibilidad de que los usuarios del sistema incluyan enunciados bajo la figura del anonimato.
- 1.7. Se puede ampliar el modelo de comunicación de mensajes complejos, para que incluya características de seguridad, tales como la encriptación de las tramas, y la inclusión de firmas digitales para los clientes y el servidor. También se puede generalizar el modelo de comunicación, para que maneje mensajes que representan instrucciones que no han sido definidas previamente.
2. El prototipo se puede someter a una serie de pruebas para evaluar su efectividad, al ser utilizado por los grupos de usuarios potenciales señalados en el marco teórico de este informe.

3. Se puede someter al prototipo a series de pruebas de rendimiento formales para evaluar su desempeño y eficiencia computacional y, de ser posible, mejorarlo en estos aspectos.

Las mejoras antes descritas, pueden ser tomadas como punto de partida para algunos de los trabajos de investigación, que se pueden derivar de los resultados de esta investigación.

5. REFERENCIAS BIBLIOGRÁFICAS.

- [AGÜE87] **AGÜERO, U.** *A theory of plausibility for computer architecture design.* Ph.D. thesis, Center for Advanced Computer Studies, University of Southwestern Louisiana, 1987.
- [BOOC96] **BOOCH, G.** *Análisis y Diseño Orientado A Objetos con Aplicaciones.* Addison-Wesley, 1996.
- [CIYD00] **CENTRO INFORMÁTICO Y DE COMUNICACIONES DEL EDIFICIO DE INGENIERÍAS.** *Groupware.* Universidad de las Palmas de la Gran Canaria, 16 de noviembre del 2000. (http://www4.ulpgc.es/gsi/video_conf/GroupWare/GroupWare.html).
- [CONK99] **CONKLIN, E. J., Ph.D.** *Capturing Organizational Memory.* Group Decision Support Systems Working Papers, sin paginación, 19 de noviembre de 1999. (<http://www.gdss.com/COM.htm>).
- [CRPC00] **THE CENTER FOR RESEARCH ON PARALLEL COMPUTATION.** *Caltech Infospheres Project.* <http://www.infospheres.caltech.edu/> noviembre del 2000.
- [DUTA99] **DUTARI DUTARI R. E.** Evaluación de aplicaciones que pueden ser utilizadas como sistemas de apoyo a la toma de decisiones, en grupos grandes, comunicados con Internet. Informe final del Seminario de Investigación 1, Instituto Tecnológico de Costa Rica. 1999.

- [ELGI91] **ELLIS, C. A.; GIBBS, S. J. and REIN, G. L.** *Groupware – Some Issues and Experiences*. Communications of the ACM, 34(1):38–58, January 1991.
- [EXCH99a] **EXPERT CHOICE, INC.** *Methodology*. Expert Choice, Inc., sin paginación, 5 de noviembre de 1999. (<http://www.expertchoice.com/methodology>).
- [EXCH99b] **EXPERT CHOICE, INC.** *Team Expert Choice Brochure*. Expert Choice, Inc., sin paginación, 19 de noviembre de 1999. (http://www.expertchoice.com/software/teamec/teamec_brochure.htm).
- [FOSC99] **FOWLER, M y SCOTT, K.** *UML Gota a Gota*. Addison-Wesley, 1999.
- [FRAG00] **FRAGUAS, A.** *Java y la OOP*. Revista Algoritmo. © Grupo EIDOS <http://www.eidos.es> 2000.
- [GDSS99] **GROUP DECISION SUPORT SYSTEMS.** *The IBIS Manual*. Group Decision Suport Systems Working Papers, sin paginación, 19 de noviembre de 1999. (<http://www.gdss.com/IBIS.htm>).
- [GUER98] **GUERRERO B., L. A.** *Un framework para el desarrollo de aplicaciones colaborativas en Web*. Tesis para optar al grado de Magíster Scientiae en computación en el Instituto Tecnológico de Costa Rica, 1998.
- [HORN00] **DEE HORNE, D.** *The Use of the Discussion Forum in On-line Courses*. University of Northern British Columbia, 3 páginas, 4 de

diciembre del 2000.
<http://vaughan.fac.unbc.ca/ctl/tow/structure/dee.html>).

- [JOHA88] **JOHANSEN, R.** Groupware: computer Support for Bussiness Teams. Free Press, 1988.
- [JOZA98] **JOYANES AGUILAR, L.; Y ZAHONERO MARTÍNEZ, I.** *Estructura de Datos: Algoritmos, abstracción y objetos*. McGraw-Hill, 1998.
- [KOCH95] **KOCH, M.** The Collaborative Multi-User Editor Project IRIS. Technische Universität München. 1995.
- [LARM99] **LARMAN, C.** UML y patrones: Introducción al análisis y diseño orientado a objetos. Prentice-Hall. 1999.
- [MICR00a] **MICROSOFT CORPORATION.** *Microsoft Exchange 2000 Conferencing Server Datasheet*.
<http://www.microsoft.com/exchange/> noviembre del 2000.
- [MICR00b] **MICROSOFT CORPORATION.** *Por qué XML*.
<http://www.microsoft.com/spain/msdn/xml/default.asp> enero del 2000.
- [NCSA00] **NCSA.** *NCSA HABANERO 2.0*
<http://havefun.ncsa.uiuc.edu/habanero/>. Noviembre del 2000.
- [NEGR95] **NEGROPONTE, N.** *Ser digital*. Atlántida, 1995.

- [NÚÑE99] **NÚÑEZ MARÍN, G.** *Toma de decisiones en grupo para juegos educativos colaborativos*. Tesis para optar al grado de Magíster Scientiae en computación en el Instituto Tecnológico de Costa Rica, 1999.
- [OLIV96] **OLIVARES, C.** *Fundamentos lógicos y políticas para un modelo de votación en grupos grandes basado en la teoría de plausibilidad*. Tesis para optar al grado de Magíster Scientiae en computación en el Instituto Tecnológico de Costa Rica, 1996.
- [POSA00] **POSADAS, M.** *El Modelo de objetos de documento (DOM)*. Grupo EIDOS, 4 de diciembre del 2000. (<http://www.eidos.es/>).
- [REIN00] **REINO R., A.** *Introducción al XML en Castellano*. <http://www.ibium.com/alf/xml/index.asp> 26 de enero del 2000.
- [ROCA99] **RODRÍGUEZ DOMÍNGUEZ, M. Y CASTRO ORDÓÑEZ, L. A.** *Formalización de la teoría de plausibilidad y del grafo de plausibilidad*. Cartago, Tesis para optar al grado de Magíster Scientiae en computación en el Instituto Tecnológico de Costa Rica, 1999.
- [RUJA99] **RUMBAUGH, J.; JACOBSON, I.; y BOOCH, G.** *The unified modeling language reference manual*. Addison-Wesley, 1999.
- [SIGA99] **SILBERSCHATZ, A. Y GALVIN, P. B.** *Sistemas Operativos*. Quinta Edición, Addison-Wesley, 1999.
- [SNEA99] **SNEAKERLABS, INC.** *Real-Time Remote Web-Conferencing, Live Application Sharing, Interactive Customer Service*. SneakerLabs,

Inc., sin paginación, 19 de noviembre de 1999.
(<http://www.sneakerlabs.com/home.html>).

[TANE97] TANENBAUM, A. S. *Redes de Computadoras*. Tercera Edición, Prentice-Hall, 1997.

[TSBC99] THE SOFT BICYCLE COMPANY. *Consensus @nyWARE Feature*. The Soft Bicycle Company, sin paginación, 19 de noviembre de 1999. (<http://www.softbicycle.com/C@Wfeat.html>).

[VENT99] VENTANA CORPORATION. *GroupSystems® Hardware and Software Requirements*. Ventana Corporation, sin paginación, 1 de diciembre de 1999.
(<http://www.ventana.com/html/requirements.html>).

[W3C98] W3C. Document Object Model (DOM) Level 1 Specification Versión 1.0. W3C, 1998. <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/DOM.pdf>

[WABR99] WALSH, K.R; BRIGGS, R.O; & others. *Learning with GSS: A case of study*. University of Arizona, 10 páginas, 19 de noviembre de 1999. (<http://www.ventana.com/library/nv3.pdf>).

6. APÉNDICE 1: MANUAL DE USUARIO DEL FORO DE DISCUSIÓN JERÁRQUICO.

En este apéndice se presenta el manual de usuario del Foro de Discusión Jerárquico -*FDJ*.

6.1. DESCRIPCIÓN GENERAL DE LA APLICACIÓN.

El foro de discusión jerárquico –en adelante, *FDJ*–, es una herramienta que permite, a sus usuarios, debatir libremente sus ideas, siguiendo el esquema de hilos de discusión. Al mismo tiempo que los usuarios debaten sus ideas, el *FDJ* realiza un análisis de plausibilidad de su planteamientos, y lo presenta a los usuarios, en la medida que expresan sus opiniones.

6.2. TIPOS DE USUARIO.

El *FDJ* maneja dos roles para sus usuarios.

1. **Administrador del Foro:** Es la persona que debe iniciar y finalizar las sesiones del *FDJ*. Se encarga también de establecer el tema de discusión, la fecha y la hora de las reuniones, así como cualquier otra información adicional que los usuarios requieran para integrarse a la reunión. Finalmente, debe comunicar, a los usuarios del sistema, la información antes descrita. Durante el desarrollo de sesión, no tiene injerencia en su desenvolvimiento.

2. **Participante:** Es el grupo de usuarios que utiliza el *FDJ* para dialogar. En general, puede ser cualquier grupo de personas que requiere utilizar el diálogo como herramienta para encontrar solución a un problema cualquiera. El problema en cuestión, será el tema que discutirán en el transcurso de la sesión.

6.3. REQUERIMIENTOS.

El *FDJ* está diseñado e implementado para operar en los sistemas que satisfacen los requisitos planteados a continuación:

- ☑ **Software de sistema:** Máquina Virtual Java -*MVJ*- compatible con el “*Java Development Kit*” -JDK- de Sun versión 1.3.X o superior. Se recomienda la *MVJ* 1.3.1.
- ☑ **Red de Computadoras:** Red de área local o de área extensa, basada en el protocolo TCP / IP. Se debe disponer de, al menos, un sistema que tenga asignado un IP visible desde todos los sistemas que se desea comunicar a través del foro.
- ☑ **Hardware:** Todos los sistemas conectados a la red de computadoras antes descrita, deben tener un procesador con rendimiento similar o superior al de cualquier procesador Intel Pentium o compatible. Al menos, 32Mb de memoria de acceso aleatorio -*MAA*-. Se debe disponer de 2Mb para la instalación de los archivos del Servidor y del Cliente del *FDJ*. Se recomiendan procesadores con rendimiento similar o superior al de un procesador Intel Pentium MMX o compatible, equipado con 64Mb de *MAA*.

- ☑ **Observación Importante:** En última instancia, los requerimientos de hardware están condicionados a la existencia de la *MVJ*, en consecuencia, el modelo de procesador es secundario, y se enuncia únicamente como referencia para catalogar el rendimiento del sistema en que se desea instalar el *FDJ*.

6.4. INSTALACIÓN.

La distribución del *FDJ* consta de dos archivos comprimidos con formato .zip: Cliente.zip y Servidor.zip, que contienen los archivos necesarios para la instalación del cliente y el servidor del *FDJ*, respectivamente. El proceso de instalación consta de los pasos que se enuncian a continuación:

1. **Instalación de la *MVJ*:** El disco de instalación del *FDJ* proporciona el “*Software Development Kit*” -*SDK*- 1.3 de Sun, para ambientes Windows y Linux, que pueden ser utilizados para completar este paso. Además, esta etapa se debe complementar con la adición de la máquina *MVJ* a la variable de ambiente del sistema *PATH*.

- 1.1. **Ambiente Windows:** Este paso se puede completar al incluir en el archivo autoexec.bat, como última instrucción a:

```
SET PATH=<PATH_MVJ>\BIN;$PATH$;
```

Donde se asume que <PATH_MVJ> se refiere al directorio donde se instaló la *MVJ*.

- 1.2. **Ambiente Linux:** Este paso se puede completar al incluir en el archivo /etc/profile, la instrucción:

```
export PATH="<PATH_MVJ>\BIN:.$PATH$:"
```

Donde se asume que <PATH_MVJ> se refiere al directorio donde se instaló la *MVJ*.

2. **Instalación de una herramienta de descompresión de archivos .zip:**

Los archivos necesarios para la instalación de la herramienta de descompresión de archivos .zip no se proporcionan en la distribución del FDJ. El usuario puede utilizar cualquier herramienta compatible con su plataforma de trabajo.

3. **Descompresión de los archivos del Servidor y del Cliente en algún directorio del sistema:**

El Servidor y el Cliente del *FDJ* están formados por archivos .class y .jar, independientes de la plataforma. Simplemente deben ser descomprimidos de los archivos .zip y colocados en algún directorio del sistema.

3.1. **Ambiente Windows:** Este paso se puede completar con la orden:

```
C:\>unzip Cliente.zip
```

O

```
C:\>unzip Servidor.zip
```

Que descomprimirá los archivos del cliente o del servidor, respectivamente, en el directorio raíz del disco C, dentro de un directorio que tiene el mismo nombre que el archivo .zip. Se asume que unzip es una herramienta genérica utilizada para realizar la descompresión de los archivos.

3.2. **Ambiente Linux:** Este paso se puede completar con la orden:

```
/usr/bin#unzip Cliente.zip
```

O

```
/usr/bin#unzip Servidor.zip
```

Que descomprimirá los archivos del cliente o del servidor, respectivamente, en el directorio `/usr/bin`, dentro de un directorio que tiene el mismo nombre que el archivo `.zip`. Se asume que `unzip` es una herramienta genérica utilizada para realizar la descompresión de los archivos.

4. **Integración de la librería XML a la MVJ:** Se debe copiar el archivo `xml4j_1_1_9.jar` -presente en el directorio de los archivos del Cliente o del Servidor, en el directorio de extensiones de la MVJ. Además, se debe incluir la librería XML en la variable de ambiente `CLASSPATH`.

4.1. **Ambiente Windows:** Este paso se puede completar con la orden:

```
C:\>copy \Cliente\xml4j_1_1_9.jar <PATH_MVJ>\jre\lib\ext\xml4j_1_1_9.jar
```

O

```
C:\>copy \Servidor\xml4j_1_1_9.jar <PATH_MVJ>\jre\lib\ext\xml4j_1_1_9.jar
```

Que copiará el archivo `xml4j_1_1_9.jar` al directorio de extensiones de la *MVJ*. Se asume que el Cliente y el Servidor han sido instalados en sus respectivas carpetas en el directorio raíz, y que la *MVJ* se instaló en `<PATH_MVJ>`.

Además, se debe agregar como última línea del archivo `autoexec.bat` la instrucción:

```
SET CLASSPATH=.;C:\Cliente\xml4j_n_n_n.jar;%CLASSPATH%;
```

O

```
SET CLASSPATH=.;C:\Servidor\xml4j_n_n_n.jar;%CLASSPATH%;
```

Según se instale el Cliente o el Servidor del *FDJ*, respectivamente.

4.2. **Ambiente Linux:** Este paso se puede completar con la orden:

```
#cp /usr/bin/Cliente/xml4j_1_1_9.jar <PATH_MVJ>/jre/lib/ext/xml4j_1_1_9.jar
```

O

```
#cp /usr/bin/Servidor/xml4j_1_1_9.jar <PATH_MVJ>/jre/lib/ext/xml4j_1_1_9.jar
```

Que copiará el archivo `xml4j_1_1_9.jar` al directorio de extensiones de la *MVJ*. Se asume que el Cliente y el Servidor han sido instalados en sus respectivas carpetas en el directorio raíz, y que la *MVJ* se instaló en `<PATH_MVJ>`.

Además, se debe agregar al archivo `/etc/profile`, la instrucción:

```
export CLASSPATH=".:usr/bin/Cliente/xml4j_1_1_9.jar:$CLASSPATH"
```

O

```
export CLASSPATH=".:usr/bin/Servidor/xml4j_1_1_9.jar:$CLASSPATH"
```

Según se instale el Cliente o el Servidor del *FDJ*, respectivamente.

6.5. ACTIVACIÓN DE LAS APLICACIONES DEL *FDJ*.

Para activar al *FDJ* debemos realizar los pasos que se describen a continuación:

1. **Ambiente Windows:** Desde una ventana MS-DOS, se ejecuta la instrucción:

```
C:\Cliente>Cliente9X
```

O

```
C:\Servidor>Servidor9X
```

Según se desee activar al cliente o al servidor del *FDJ*, respectivamente. Se asume que el Cliente y el Servidor han sido instalados de acuerdo al procedimiento descrito en este manual.

2. **Ambiente Linux:** Desde una consola del sistema, se ejecuta la instrucción:

```
/usr/bin/Cliente$ClienteLinux
```

O

```
/usr/bin/Servidor$ServidorLinux
```

Según se desee activar al cliente o al servidor del *FDJ*, respectivamente. Se asume que el Cliente y el Servidor han sido instalados de acuerdo al procedimiento descrito en este manual.

6.6. FUNCIONAMIENTO DEL *FDJ*.

En primer término, se debe mantener en mente la idea de que el *FDJ* es una aplicación basada en *IGU*⁸, en consecuencia, presenta los controles que normalmente se espera encontrar en dichas aplicaciones, tales como: botones de comandos, cajas de texto, listas de selección, árboles, etc. Es decir, funciona como se espera que funcione cualquier aplicación común, desarrollada para el Sistema Operativo MS-Windows 9X; o para un entorno de ventanas X-Windows, como los que se presentan en las distribuciones más populares del Sistema Operativo Linux (Gnome, KDE, para mencionar los más conocidos en el medio local).

Una observación importante acerca del interfaz: Ciertos botones son constantes dentro del sistema, presentando comportamientos regulares y bien conocidos. Sólo se documentarán en este punto, ellos son:

1. **Aceptar:** El usuario está indicando al *FDJ* que procese los datos introducidos en el diálogo donde se encuentra.
2. **Terminar:** El usuario está indicando al *FDJ* que desea finalizar la sesión.
3. **Cancelar:** El usuario está indicando al *FDJ* que no procese los datos introducidos en el diálogo donde se encuentra.
4. **Si:** El usuario está indicando al *FDJ* que está de acuerdo con la pregunta que se le está haciendo en ese momento.

⁸ *IGU*: Interfaz Gráfico de Usuario; en inglés *GUI*.

5. **No:** El usuario está indicando al *FDJ* que no está de acuerdo con la pregunta que se le está haciendo en ese momento.

6.6.1. FUNCIONAMIENTO DEL SERVIDOR.

El *FDJ* es una aplicación Cliente / Servidor, en consecuencia, para funcione correctamente, se debe activar el componente del servidor, en primer término. La activación del servidor se debe realizar en uno de los sistemas conectado a la red, que tenga un IP visible por todos los sistemas que se deben conectar al foro.

6.6.1.1. INICIO DE LA SESIÓN.

Al activar el servidor de *FDJ*, se presenta al administrador del sistema un diálogo de inicio de sesión, tal y como se muestra en la Ilustración 40.

Ilustración 40: Diálogo de Inicio de Sesión en el Servidor

Los campos que se han señalado con números son muy importantes al momento de inicializar la sesión, porque representan parámetros críticos para la reunión.

1. **Puerto Asignado:** Es un número entero comprendido entre 1024 y 9999.

2. **Tema de Discusión:** En el caso más general, puede ser una cadena de caracteres que no consista únicamente de espacios en blanco.
3. **Marcas de Aceptación / Rechazo:** Son números con fracción decimal, que pueden estar comprendidos en el intervalo $0 < marcaAceptacion < 1$ y $0 < marcaRechazo < 1$.
4. **Umbrales de Aceptación / Rechazo:** Son números con fracción decimal, que pueden estar comprendidos en el intervalo $0 < umbralAceptacion < marcaAceptacion$, y $0 < umbralRechazo < marcaRechazo$.
5. **Línea de mensajes:** Es utilizada por el *FDJ* para informar al administrador del sistema, que se presenta algún error en la captura de los datos.

Cuando el administrador del *FDJ* completa la captura de estos parámetros, debe pulsar el botón aceptar, para que el sistema compruebe la consistencia de los datos. Si el administrador del foro no desea iniciar la sesión del foro, puede pulsar el botón cancelar, para finalizar la aplicación.

En caso que se presente alguna inconsistencia entre ellos -puerto menor que 1024, por ejemplo-, el sistema presentará un mensaje en la línea de mensajes. El *FDJ* acepta los parámetros dados solamente en el caso que los ellos sean consistentes entre sí. En caso de duda, el administrador puede pulsar el botón "Limpiar", que acompaña a cada campo, para restaurarlo a su valor predeterminado.

6.6.1.2. DIÁLOGO PRINCIPAL.

En el caso que el *FDJ* acepte los parámetros suministrados, desaparece el diálogo antes mostrado y se presenta un nuevo diálogo, que es la ventana principal del servidor del foro, que se presenta en la Ilustración 41.

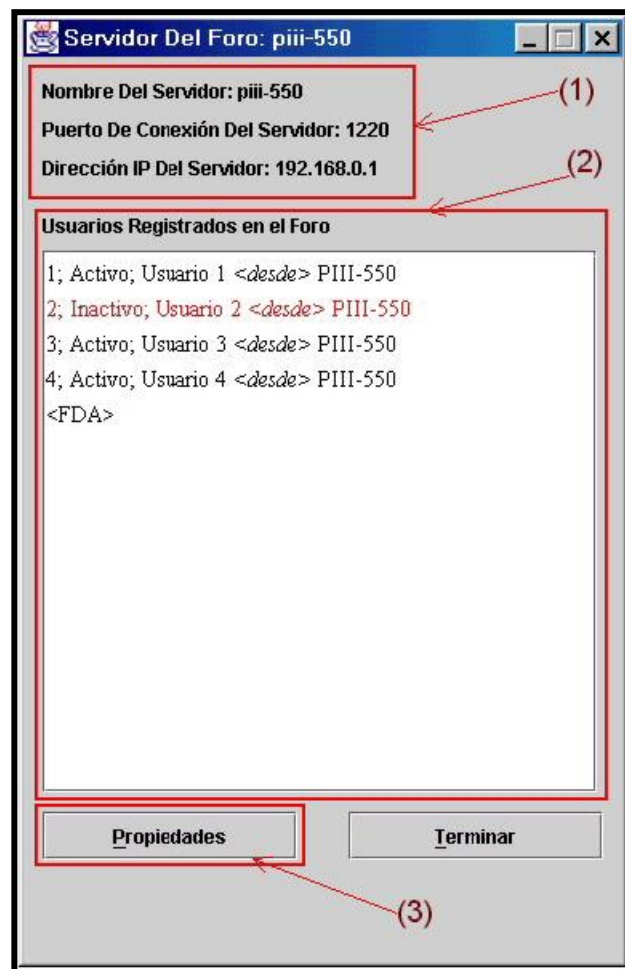


Ilustración 41: Diálogo Principal del Servidor

En este diálogo aparecen, fundamentalmente, tres áreas:

1. **Información de Acceso al Servidor:** Allí aparecen anotados: El *nombre del servidor -Host* en la jerga TCP/IP-; el *puerto de conexión* -que

previamente definió el administrador-; y la *dirección IP*, que corresponde al servidor. Estos parámetros los deben conocer los clientes potenciales del foro, para poderse conectar a la sesión.

2. **Lista de Usuarios Conectados:** A medida que los usuarios del foro se unen a la sesión, aparecen registrados en la Lista de Usuarios. Para cada usuario, aparece anotado: El *identificador* de usuario, el *estado* del usuario -"Activo" si está en la sesión, "Inactivo" si se retiró-; el *nombre* que utilizó para registrarse en la sesión y; finalmente; el nombre del sistema -*Host*- donde se localiza el usuario. Los usuarios que están inactivos en la sesión, se pueden identificar fácilmente porque sus datos aparecen pintados de color rojo; en tanto, los datos de los usuarios que permanecen activos están pintados de color negro.
3. **Botón Propiedades:** Este botón permite que el administrador del foro ingrese en un diálogo donde puede consultar todos los parámetros que se asignaron al inicio de la sesión. Además, le permite alterar la cantidad máxima de usuarios conectados a la sesión. Este diálogo se presenta en la Ilustración 42.

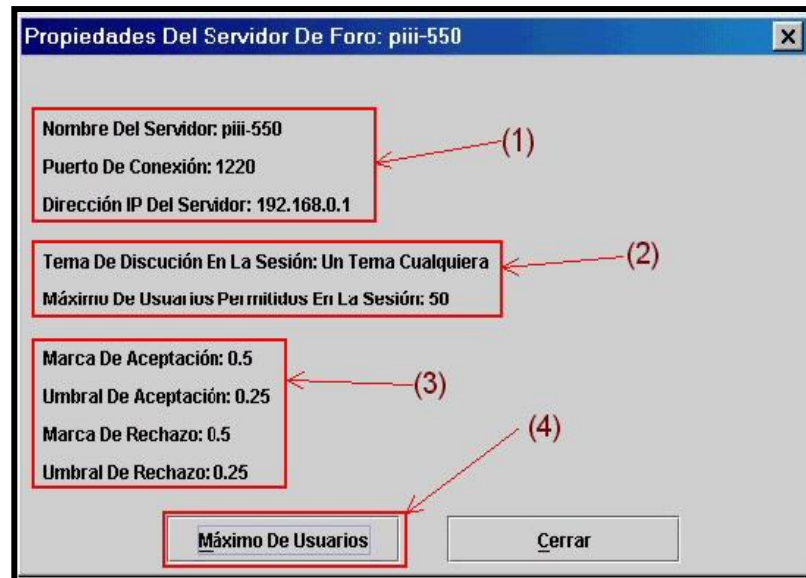


Ilustración 42: Diálogo de Propiedades del Servidor

Las áreas que aparecen en este diálogo son bastante evidentes entre si: La información de acceso al servidor, el tema de discusión y el máximo de usuarios permitidos en la sesión; finalmente, las marcas y umbrales de aceptación y rechazo que se aplican en la sesión.

6.6.1.3. CAMBIAR EL MÁXIMO DE USUARIOS PERMITIDOS EN LA SESIÓN.

En la Ilustración 42 es relevante el botón “*Máximo De Usuarios*” -área (4)-; que nos permite alterar la cantidad máxima de usuarios conectados a la sesión. Este diálogo se presenta en la Ilustración 43.

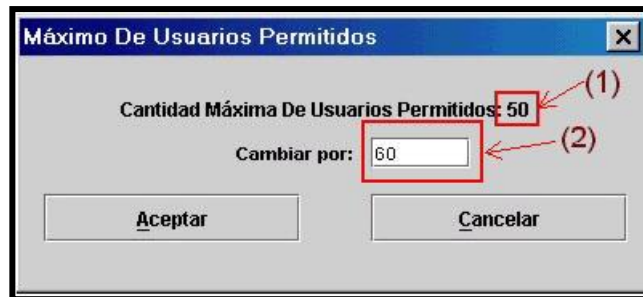


Ilustración 43: Diálogo para Alterar el Máximo de Usuarios Permitidos en la Sesión

Se presentan dos áreas relevantes dentro de este diálogo:

1. **Cantidad Máxima De Usuarios Permitidos:** Indica cuantos usuarios se pueden conectar a la sesión del *FDJ* simultáneamente, en el momento en que aparece el diálogo.
2. **Cambiar por:** Este cuadro de texto, permite que el *FDJ* capture un nuevo valor para la cantidad máxima de usuarios permitidos en la sesión. En general, el cuadro de texto espera un valor entero mayor que cero. En caso que se no se suministre un valor con estas características, aparecerá una caja de mensaje, indicando el error y el tipo de valor que se espera capturar en el cuadro de texto -en este caso, no se altera el valor actual, ver Ilustración 44-. Tanto en el caso que se captura un valor correcto o se trata de capturar un valor incorrecto, el diálogo se cierra al finalizar la acción.

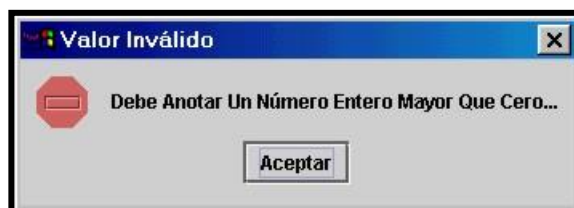


Ilustración 44: Mensaje de Error en la Captura del Máximo de Usuarios Permitidos

6.6.1.4. FINALIZAR LA SESIÓN.

Finalmente, cuando termina la sesión, y el administrador trata de finalizar el funcionamiento del servidor, el *FDJ*, debe pulsar el botón “*Terminar*” que se encuentra dentro de su diálogo principal. A continuación se presenta otro diálogo donde se debe confirmar esta acción. Este diálogo se presenta a continuación:

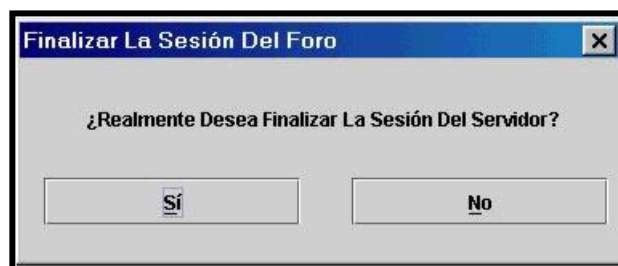


Ilustración 45: Diálogo para Finalizar la Sesión del Servidor

En el caso que se confirme la finalización de la sesión, el servidor del *FDJ* archivará el diálogo realizado en el directorio donde se encuentra instalado, así como la el listado de los usuarios que se conectaron a la sesión. Estos archivos se llaman “*DialogoDelForo.xml*” y “*ListaDeUsuarios.xml*”, respectivamente. Ambos son archivos de texto ASCII, en formato XML, que pueden ser observados en cualquier navegador http, compatible con XML, o en última instancia, en un editor ASCII.

6.6.2. FUNCIONAMIENTO DEL CLIENTE.

Esta sección describe el funcionamiento del cliente del *FDJ*.

6.6.2.1. INICIO DE LA SESIÓN.

Para que un usuario inicie su sesión en el *FDJ* como cliente, se requiere conocer, indispensablemente, los parámetros que solicita el diálogo que se presenta en la Ilustración 46:

Ilustración 46: Diálogo de Inicio de Sesión en el Cliente

Los parámetros antes citados son:

- ☒ Fecha y hora de la reunión,
- ☒ Puerto de conexión asignado a la reunión (1),
- ☒ Nombre -Host- o la dirección IP del computador donde se aloja el servidor (2),
- ☒ Un nombre que identificará al usuario dentro de la sesión (3). Este nombre no puede repetirse dentro de la sesión.

Otros tópicos que debe conocer el usuario, pero que no resultan relevantes al momento de realizar la conexión a la sesión son:

- ☒ El tema a debatir en la sesión,
- ☒ Marcas y umbrales de aceptación y rechazo que se aplicarán.

Todos los parámetros necesarios para realizar la conexión al servidor del *FDJ* se deben comunicar a los usuarios potenciales, con suficiente anterioridad al momento de realizar la discusión.

Generalmente, se utilizará un medio de comunicación alternativo, tal como el correo electrónico, para concretar dicha notificación.

Al momento en que el usuario pulsa el botón aceptar, el cliente del *FDJ* tratará de iniciar una sesión en el servidor identificado con los parámetros antes señalados.

En la línea de mensajes del diálogo de inicio de sesión(4), aparecerán una serie de mensajes que indicarán el avance del proceso de conexión, así como los mensajes de error que se pueden originar en el proceso

Los errores más comunes que se presentan en el proceso de conexión son:

1. **El usuario trata de iniciar la sesión utilizando un nombre previamente asignado:** Este error se corrige simplemente alterando el nombre que propone el usuario, hasta que proponga uno que no está en uso.
2. **El servidor del foro al que se trata de conectar no existe:** Este error se manifiesta cuando el cliente trata de conectarse:

- ☒ En una fecha u hora diferentes a las asignadas para la reunión,
- ☒ Utilizando un puerto, dirección IP o Host que no corresponden a los asignados a la reunión.

En ambas situaciones, la solución está en utilizar los parámetros correspondientes a la reunión real.

6.6.2.2. DIÁLOGO PRINCIPAL.

Si el usuario logra conectarse al servidor del foro, aparecerá un diálogo como el que se presenta en la Ilustración 47:

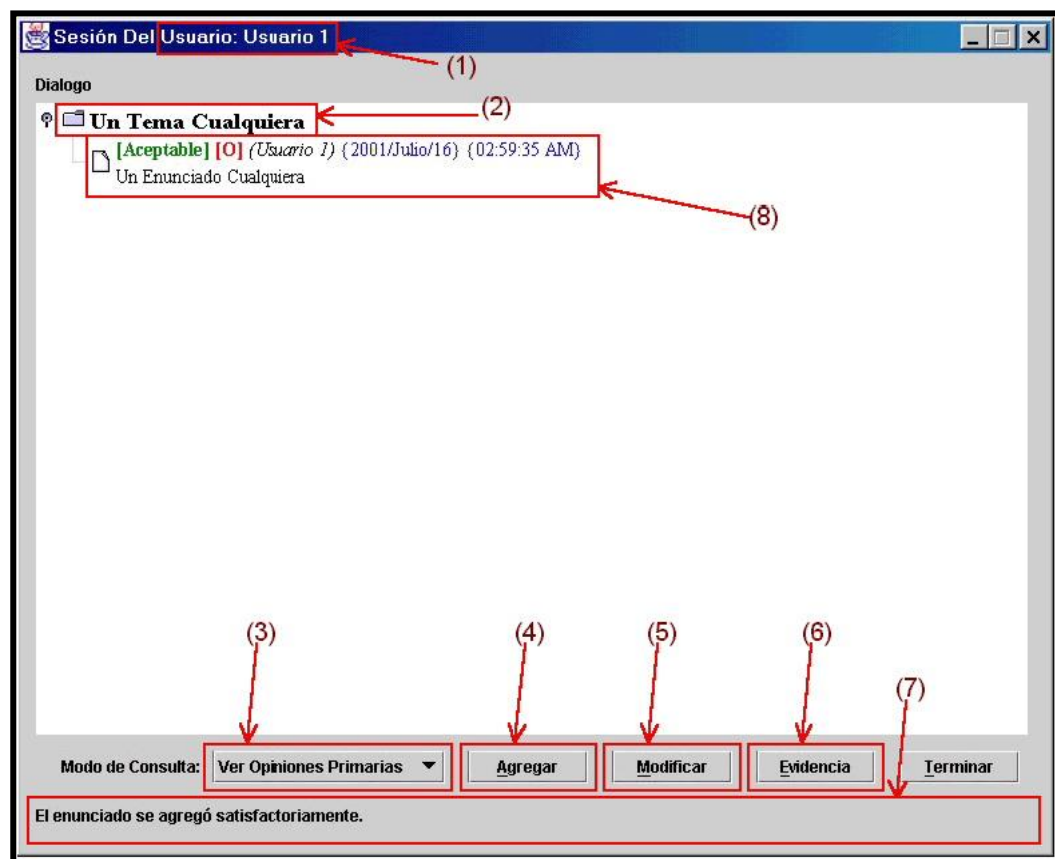


Ilustración 47: Diálogo Principal del Cliente

Este diálogo presenta varias áreas relevantes. Ellas son:

1. **Nombre que el usuario utiliza dentro de la sesión:** En esta zona aparece el nombre que identifica a éste usuario, de los demás usuarios conectados

en la sesión. El servidor del foro controla que no existan dos usuarios que tengan el mismo nombre de usuario, dentro de la misma sesión.

2. **Tema de Discusión:** En esta zona aparece el Tema de Discusión de la sesión. Cuando cada usuario ingresa a la sesión, el servidor le envía, entre otras cosas, el Tema de Discusión que se debatirá en la sesión, que se muestra como la raíz del diálogo que se desarrollará en la sesión. Dicho diálogo se presenta como un árbol, similar al árbol de directorios de un sistema de archivos.
3. **Consultas:** En esta zona se muestra una lista que contiene las modalidades de consulta permitidas. Estas consultas permiten que el usuario “filtre” los enunciados que integran el diálogo, mostrando únicamente aquellos que responden al criterio de selección escogido. Las consultas que se han implementado son:
 - ☒ *Ver Opiniones Primarias:* que muestra únicamente a los enunciados que son origen de un hilo de discusión.
 - ☒ *Ver Todo:* que muestra a todos los enunciados que se han incluido en el diálogo, en el transcurso de la sesión.
4. **Agregar:** Este botón permite que el usuario agregue enunciados al diálogo que se desarrolla en la sesión. Cuando un usuario agrega un enunciado, éste es agregado como “hijo” del enunciado que está seleccionado en ese momento.
5. **Modificar:** Este botón permite que el usuario modifique enunciados que previamente agregó. Cuando un usuario modifica un enunciado, se intenta modificar al enunciado que está seleccionado en ese momento. En caso

no exista un enunciado seleccionado, el botón no realiza ninguna acción. El *FDJ* rechaza las acciones de los usuarios que intentan modificar enunciados de otros usuarios. Cuando se presenta esta situación -intentos de modificar enunciados de otros usuarios-, aparece en la Línea de Mensajes (7) una advertencia de que esa acción no está permitida.

6. **Evidencia:** Este botón permite que el usuario incluya en su consulta a la evidencia asociada del enunciado que está seleccionado en ese momento. En caso no exista un enunciado seleccionado, el botón no realiza ninguna acción.
7. **Línea de mensajes:** Es utilizada por el *FDJ* para informar al usuario, que se presenta algún error en la captura de los datos. En ella aparecen también mensajes informando cuando un usuario ingresa o se retira de la sesión.
8. **Enunciado:** Los enunciados que agregan los usuarios del *FDJ* a medida que se desarrolla el diálogo, aparecerán estructurados según la jerarquía que les corresponda por el hilo de la discusión en que se ubican. Cada enunciado posee una estructura bien definida, de acuerdo al formato presentado en la Ilustración 48:

[EstadoDePlausibilidad] (Apoyo) **[Naturaleza]** (Nombre) {fecha} {hora}
 Texto

Ilustración 48: Formato de los Enunciados del *FDJ*

A continuación, se explicará el significado y los valores que puede asumir cada una de las partes del enunciado:

- ☒ **EstadoDePlausibilidad:** Es un valor que se calcula de acuerdo al algoritmo DFS_Plausible, al aplicarlo sobre todos los enunciados que forman un hilo de discusión. Los valores que puede asumir este elemento son: *“Satisfactorio”, “Aceptable”, “Dudoso”, “Insatisfactorio”, “Indeterminado”*.
- ☒ **Apoyo:** Es el apoyo a favor que recibe el enunciado, por parte de su evidencia asociada, producto del análisis de plausibilidad realizado sobre el hilo de discusión en que se encuentra.
- ☒ **Naturaleza:** Este valor es asignado por el usuario, al momento de agregar o modificar el enunciado. Los valores que puede asumir este elemento son: *“Opinión Primaria”, “Evidencia A Favor”, “Evidencia En Contra”, “Aclaración”, “Desactivado”*. Es importante recalcar que dentro del enunciado, sólo aparece una inicial que identifica a la naturaleza en mención. Estas iniciales son, respectivamente: *“O”, “F”, “C”, “A”, “D”*.
- ☒ **Nombre:** Es el nombre del usuario que agregó el enunciado.
- ☒ **Fecha Hora:** Es la fecha y la hora en que se agregó el enunciado en el servidor del *FDJ*.
- ☒ **Texto:** Es la idea escrita que expresa el usuario.

6.6.2.3. AGREGAR ENUNCIADOS AL DIÁLOGO.

Cuando un usuario, dentro de su diálogo principal, pulsa el botón “Agregar”, se presenta otro dialogo similar al que se observa en la Ilustración 49:

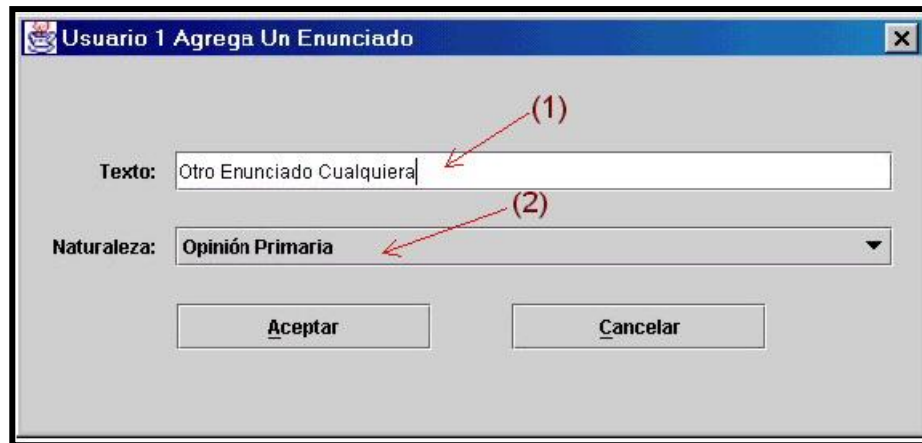


Ilustración 49: Diálogo para Agregar Enunciados al *FDJ*

Este diálogo presenta dos áreas relevantes. Ellas son:

1. **Texto del Enunciado:** En este campo, el usuario escribirá el texto que integra al enunciado que desea agregar al diálogo.
2. **Naturaleza del Enunciado:** En esta lista el usuario debe seleccionar, de las naturalezas posibles que puede asumir su enunciado, la que le desea asignar. Las naturalezas que podría asumir un enunciado cualquiera dependen de la naturaleza de su enunciado padre. Así, se presentan varios casos, de acuerdo a la teoría del *FDJ*:
 - ☑ **El usuario no seleccionó un enunciado al momento de pulsar el botón aceptar:** En este caso, el enunciado agregado sólo puede tener naturaleza “*Opinión Primaria*”.
 - ☑ **El usuario seleccionó el tema de discusión al momento de pulsar el botón aceptar:** Nuevamente, el enunciado agregado sólo puede tener naturaleza “*Opinión Primaria*”.

- ☑ **El usuario seleccionó algún enunciado al momento de pulsar el botón aceptar:** En este caso, el enunciado agregado puede asumir cualquiera de las naturalezas que se indican a continuación: *“Aclaración”, “Evidencia A Favor”, “Evidencia En Contra”*.

En general, el procedimiento para agregar un enunciado al diálogo es bastante sencillo:

1. Se selecciona, con el ratón, el enunciado padre del enunciado que se agregará.
2. Se pulsa el botón *Agregar*. Si el enunciado seleccionado esta desactivado, el botón no ejercerá acción alguna.
3. En el campo *Texto* se escribe el enunciado en sí.
4. En la lista *Naturaleza* se selecciona, de las naturalezas disponibles, aquella que se desea asignar al enunciado.
5. Se pulsa el botón *Aceptar* para que el *FDJ* procese los datos escritos, o *Cancelar* en caso que se desee abandonar la adición del enunciado.

El único mensaje que puede presentarse en la línea de mensajes, normalmente, cuando se agrega un enunciado es:

1. **El enunciado se agregó satisfactoriamente:** Cuando la inclusión del enunciado se realizó sin ningún tipo de contratiempo.

Existen otros mensajes adicionales, pero en la práctica, es imposible que aparezcan, por las comprobaciones que realiza el sistema a lo interno de su estructura, por eso no son documentados en este manual.

6.6.2.4. MODIFICAR ENUNCIADOS DEL DIÁLOGO.

Cuando un usuario, dentro de su diálogo principal, pulsa el botón “Modificar”, se presenta otro dialogo similar al que se observa en la Ilustración 50:

Ilustración 50: Diálogo para Modificar Enunciados

Este diálogo presenta varias áreas relevantes. Ellas son:

1. **Texto del Enunciado:** En este campo, aparecerá al usuario el texto que integra al enunciado que desea modificar. Utilizando las capacidades usuales de su teclado y ratón, debe realizar los cambios que considere pertinentes, o puede dejarlo sin alteraciones.
2. **Naturaleza del Enunciado:** En esta lista el usuario debe seleccionar, de las naturalezas posibles que puede asumir su enunciado, la que le desea reasignar. El cambiar la naturaleza de un enunciado es una acción que tiene consecuencias sobre los enunciados que forman la evidencia el

enunciado modificado -automáticamente toman la naturaleza “*Aclaración*”-

.

3. En esta zona aparecen: el estado de plausibilidad del enunciado, el apoyo que recibe el enunciado -de acuerdo al algoritmo *DFS_Plausible*-, y la fecha / hora, en que se agregó el enunciado en el servidor del *FDJ*. Estos valores son calculados por el servidor del foro y no pueden ser alterados por los usuarios, únicamente tienen carácter informativo.
4. Los botones de radio que aparecen en esta zona, posibilitan que el usuario modifique al enunciado de varias formas.
 - ☒ *Retractar*: Cuando el usuario se retracta del enunciado, toma la naturaleza “*Desactivado*” y toda su evidencia asociada toma la naturaleza “*Aclaración*”.
 - ☒ *Desactivar*: Cuando el usuario desactiva un enunciado, obviamente tomará la naturaleza “*Desactivado*”, al igual que toda su evidencia asociada.
 - ☒ *Activar / Reactivar*: Un enunciado que ha sido Desactivado, puede ser reactivado por el usuario que lo creó, al modificarlo y aplicar esta acción.

En general, el procedimiento para modificar un enunciado al diálogo es bastante sencillo:

1. Se selecciona con el ratón el enunciado que se desea modificar.

2. Se pulsa el botón *Modificar*. En caso que el usuario trate de modificar un enunciado que no le pertenece, aparecerá una caja de mensajes indicando que “El *UsuarioXXX* no puede modificar los mensajes del *UsuarioYYY*” como se puede observar en la . En este caso, el enunciado no sufre ninguna alteración.



Ilustración 51: Mensaje de Error al Modificar un Enunciado que no Pertenece al Usuario

3. Si se requiere, se altera el texto del enunciado.
4. Si se requiere, se selecciona, de las naturalezas disponibles, aquella que se desea asignar al enunciado.
5. Si se requiere, se selecciona el botón de radio que corresponda a la acción que se desea realizar.
6. Se pulsa el botón Aceptar para que el FDJ procese los datos escritos, o Cancelar en caso que se desee abandonar la adición del enunciado.

El único mensaje que puede presentarse en la línea de mensajes, normalmente, cuando se modifica un enunciado es:

1. **El enunciado se modificó satisfactoriamente:** Cuando la alteración del enunciado se realizó sin ningún tipo de contratiempo.

Existen otros mensajes adicionales, pero en la práctica, es imposible que aparezcan, por las comprobaciones que realiza el sistema a lo interno de su estructura, por eso no son documentados en este manual.

6.6.2.5. CAMBIAR EL TIPO DE CONSULTA QUE REALIZA UN CLIENTE.

Cuando un usuario, dentro de su diálogo principal, pulsa la lista de consultas, se presenta una lista que contiene las consultas que se pueden realizar para filtrar los mensajes del *FDJ*. Esta lista se observa en la Ilustración 52, dentro de la elipse roja:

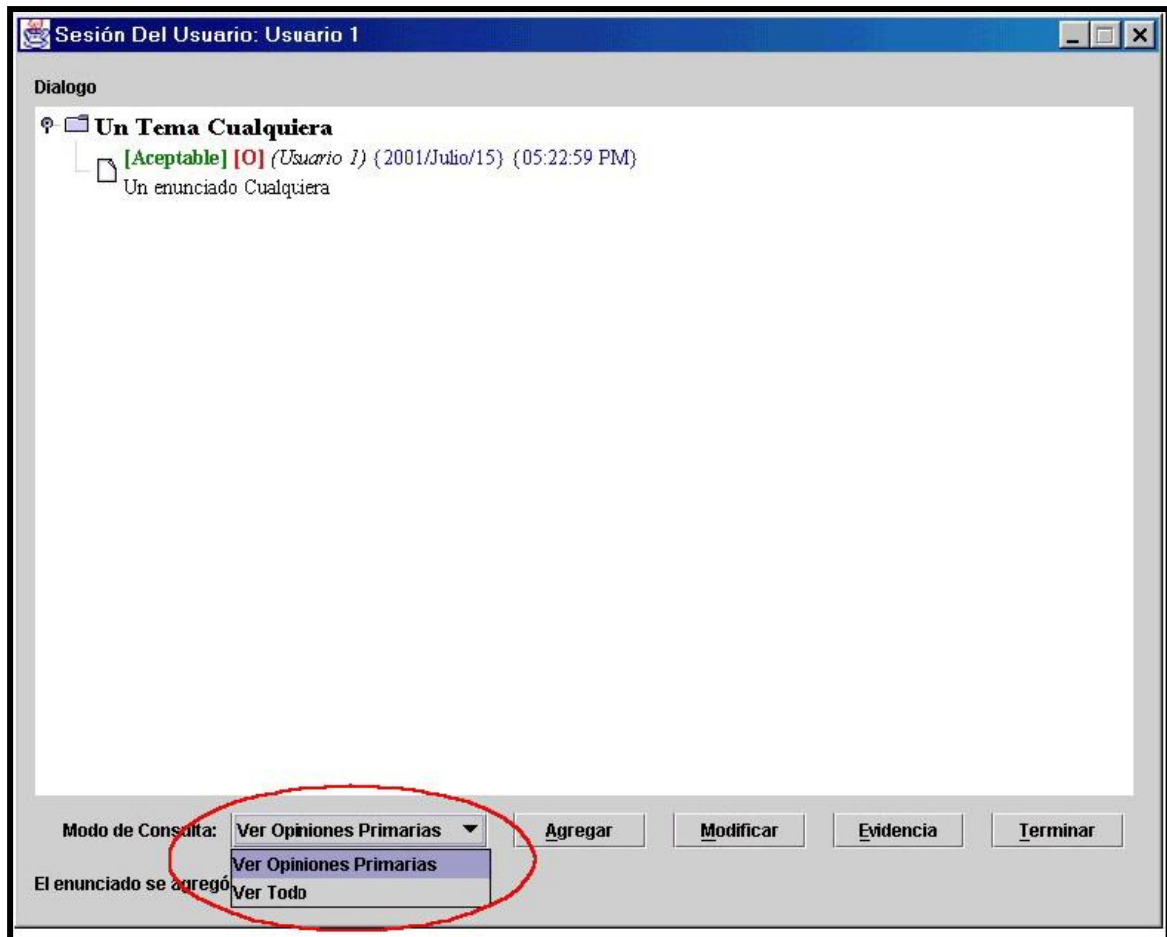


Ilustración 52: Lista de Consultas

En general, el procedimiento para cambiar el tipo de consulta que utiliza un cliente es muy simple:

1. Se selecciona con el ratón el tipo de consulta que se desea utilizar.
2. El *FDJ* procesa los enunciado que integran el diálogo realizado en la sesión, y muestra al cliente sólo aquellos que corresponden a la consulta que ha solicitado.

6.6.2.6. FINALIZAR LA SESIÓN.

Cuando un usuario desea finalizar su participación en la sesión del foro, debe pulsar el botón “*Terminar*” que se encuentra dentro de su diálogo principal. A continuación se presenta otro diálogo donde se debe confirmar esta acción. Este diálogo se presenta a continuación:

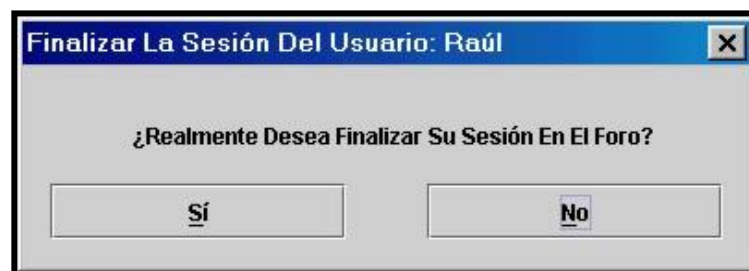


Ilustración 53: Diálogo para Finalizar la Sesión del Cliente

En el caso que se confirme la finalización de la sesión, el servidor del *FDJ* cambiará su atributo de estado, de “Activo” a “Inactivo” y eliminará su consulta. Por otro lado, el cliente del usuario archivará la consulta que realizaba el usuario en el directorio donde se encuentra instalado. El archivo donde se guarda la consulta tendrá el nombre “ConsultaDelUsuario.xml”. Al igual que en el caso del servidor, es un archivo de texto ASCII, en formato XML, que puede ser observados en cualquier navegador http, compatible con XML, o en última instancia, en un editor ASCII.

6.6.3. CÁLCULO DEL ESTADO DE PLAUSIBILIDAD DE LOS HILOS DE DISCUSIÓN.

El cálculo del estado de plausibilidad de los hilos de discusión del *FDJ* es un proceso que realiza el servidor del foro, cuando un usuario ingresa a la sesión, o cuando algún usuario agrega o modifica enunciados.

El servidor del foro recalcula el estado de plausibilidad de todos los enunciados que lo requieran. Además, envía a cada uno de los usuarios incluidos en la sesión, la información que sea necesaria para que su consulta sea consistente con el diálogo que se almacena en el servidor.

Estos procesos los realiza el servidor del foro, sin intervención alguna de los usuarios o del administrador del sistema.

6.7. MEJORAS Y AMPLIACIONES FUTURAS DEL *FDJ*.

El fundamento teórico del *FDJ* que se ofrece, está cimentado en la formalización de un Foro de Discusión basado en Hilos de Discusión. Esta formalización aparece documentada en la Tesis de Maestría: “*Toma de Decisiones en Grupo para Juegos Educativos Colaborativos*” de Giannina Núñez Marín, del Instituto Tecnológico de Costa Rica, del año 1999.

El *FDJ* implementa un subconjunto de la formalización antes mencionada. En consecuencia, puede ser objeto de una serie de mejoras que se describen a continuación:

1. **Incluir un módulo de votaciones fundamentado en la Teoría de Plausibilidad:** Este módulo de votaciones está considerado en la formalización de Núñez Marín.
2. **Incluir el intercambio de datos multimedia:** Adicionalmente al intercambio de mensajes basados en texto, la formalización de Núñez Marín contempla que el foro de discusión puede permitir el intercambio de mensajes que contengan tanto texto plano como información multimedia - imágenes, sonidos, animaciones, entre otras posibilidades-.

3. **Ampliar los tipos de consultas disponibles:** La implementación actual del *FDJ* permite que los usuarios consulten el diálogo desarrollado bajo dos modalidades básicas: “*Ver Opiniones Primarias*”, y “*Ver Todo*”. Sin embargo, la formalización de Núñez Marín contempla que el foro de discusión puede permitir otros tipos de consultas, tales como: “*Ver Lo Leído*”, “*Ver Lo No Leído*”, “*Ver Enunciados Aclaración*”, “*Ver Enunciados A Favor*”, “*Ver Enunciados En Contra*”.
4. **Ampliar la cobertura de las operaciones deshacer / rehacer del *FDJ*:** El prototipo del *FDJ* considera un único nivel para cada una de estas operaciones.
5. Se puede mejorar la interfaz de la aplicación, para que tenga el acabado de una aplicación más depurada.
6. Se puede mejorar el modelo de colaboración utilizado en el prototipo, al otorgar al administrador del servidor la posibilidad de administrar a los usuarios registrados en el *FDJ* -desactivar a los usuarios problemáticos, por ejemplo-. Además, se puede incluir la posibilidad de que los usuarios del sistema incluyan enunciados bajo la figura del anonimato.
7. Se puede ampliar el modelo de comunicación de mensajes complejos, para que incluya características de seguridad, tales como la encriptación de las tramas, y la inclusión de firmas digitales para los clientes y el servidor. También se puede generalizar el modelo de comunicación, para que maneje mensajes que representan instrucciones que no han sido definidas previamente.

8. El prototipo se puede someter a una serie de pruebas para evaluar su efectividad, al ser utilizado por los grupos de usuarios potenciales señalados en el marco teórico de este informe.
9. Se puede someter al prototipo a series de pruebas de rendimiento formales para evaluar su desempeño y eficiencia computacional y, de ser posible, mejorarlo en estos aspectos.

6.8. ERRORES CONOCIDOS.

Se conocen algunos errores menores en el funcionamiento del cliente del *FDJ*, todos relacionados con la interfaz principal de la aplicación. Ellos son:

1. Cuando el árbol del diálogo crece más allá de los límites de la pantalla visible, la función de autoscroll, no funciona.
2. Cada vez que el servidor del foro actualiza la consulta del cliente, el árbol del diálogo de dicha consulta se contrae completamente.

A la fecha, estos errores no se han podido corregir.

7. APÉNDICE 2: GLOSARIO DE TÉRMINOS DEL DOMINIO DE LA APLICACIÓN “*FORO DE DISCUSIÓN JERÁRQUICO*”.

En este apéndice se presenta un glosario que contiene las definiciones de términos más importantes dentro del dominio del sistema.

1. **Apoyo:** Es una medida del grado de aceptación de la naturaleza de un enunciado dado, calculado con base en la evidencia a favor y la evidencia en contra de dicho enunciado. Es un número entre 0 y 1, sin incluir los extremos.
2. **Consulta:** Es un subconjunto de Enunciados del diálogo, que observa un usuario, en un momento dado de la sesión.
3. **Diálogo:** Es un conjunto de Enunciados que se agrupan de manera jerárquica, de acuerdo al modelo de hilos de discusión.
4. **Enunciado:** Es una opinión categorizada que emite un usuario, acerca del tema de discusión de una sesión del *FDJ*.
5. **Estado:** es un atributo que indica si el usuario está o no presente en la sesión.
6. **Evidencia Asociada:** Se refiere al conjunto de enunciados, hijos de un enunciado dado.
7. **Fecha:** Se refiere a la fecha calendario en que se redactó un enunciado.

8. **Hilo De Discusión Primario:** Es un hilo de discusión que tiene como raíz a una opinión primaria.
9. **Hilo De Discusión Secundario:** Es un hilo de discusión que tiene como raíz a un enunciado que no es opinión primaria.
10. **Hilo De Discusión:** Es la representación jerárquica del diálogo desarrollado con respecto a un enunciado, que se toma como raíz de la discusión.
11. **Hora:** Se refiere a la hora calendario en que se redactó un enunciado.
12. **Marca De Aceptación:** Es un número entre cero y uno, sin incluir los extremos. Se utiliza en el Análisis De Plausibilidad.
13. **Marca De Rechazo:** Es un número entre cero y uno. Se utiliza en el Análisis De Plausibilidad.
14. **Naturaleza:** Es una categorización subjetiva o producto del Análisis de Plausibilidad, sobre su enunciado padre (si tiene). Puede ser Opinión Primaria (O), Aclaración (A), Evidencia A Favor (F), Evidencia En Contra (C), Desactivado (D).
15. **Nombre:** Es el nombre o pseudónimo que adopta el usuario durante la sesión.
16. **Plausibilidad:** Es una medida del grado de aceptación que tiene un enunciado dentro de la sesión. Puede ser: Aceptable, Satisfactorio, Indeterminado, Dudoso, Insatisfactorio.

17. **Sesión:** Es el período de tiempo en el que el *FDJ* admite usuarios y administra el diálogo que ellos desarrollan.
18. **Tema De Discusión:** Es el tema que se discutirá a durante la sesión.
19. **Texto:** Es la afirmación escrita que constituye al enunciado.
20. **Tipo De Consulta:** Identifica cada una de las consultas definidas dentro del sistema: Ver todo, Ver lo leído, Ver lo no leído, Ver Enunciados O, Ver Enunciados F, Ver Enunciado C, Ver Enunciados A, Abrir o Cerrar Un hilo de discusión.
21. **Umbral De Aceptación:** Es un número entre cero y la Marca De Aceptación, sin incluir los extremos. Se utiliza en el Análisis De Plausibilidad.
22. **Umbral De Rechazo:** Es un número entre cero y la Marca De Rechazo, sin incluir los extremos. Se utiliza en el Análisis De Plausibilidad.
23. **Usuario:** Persona que utiliza el *FDJ* como herramienta para debatir un problema, con otros usuarios.

8. ANEXO 1: FORMALIZACIÓN DEL FORO DE DISCUSIÓN JERÁRQUICO.

Los sistemas colaborativos se desarrollan con el fin de soportar las diversas tareas de grupos. Considerando las diferentes características de un grupo, entre ellas, la diversidad de pensamiento de sus miembros, se vislumbra la necesidad de poseer herramientas que propicien la expresión de las ideas de los miembros del grupo y que permitan mostrar el criterio del mismo.

En términos generales, es deseable contar con herramientas que faciliten el proceso de toma de decisiones del grupo, desde la formulación del problema de decisión hasta la selección de la mejor opción.

El método de toma de decisiones que se propone está basado fundamentalmente en tres aspectos: un foro de discusión basado en hilos de discusión (del inglés *threaded discussions*) con enunciados que representan evidencia a favor y evidencia en contra, un método de votación basado en el modelo de votación en grupos grandes [Oliv96] y la toma de decisiones en grupo basada en la teoría de plausibilidad [Agüe87]. Este método de toma de decisiones busca primordialmente apoyar el proceso de toma de decisiones en dos aspectos: la búsqueda del consenso y la realización de votaciones.

8.1. INTRODUCCIÓN

El foro de discusión es una herramienta, básicamente de comunicación, que se pone a disposición de los jugadores para que puedan debatir libremente sus ideas.

Mediante el foro, los jugadores pueden someter a discusión cualquier asunto relacionado con el juego. De la discusión debería obtenerse como resultado el grado de aprobación y de rechazo de los jugadores respecto al objeto de la discusión.

En este trabajo el foro de discusión se concibe como un conjunto de hilos de discusión, cada uno de los cuales está constituido por expresiones denominadas enunciados, que describen opiniones de los jugadores.

Con la finalidad de introducir de manera intuitiva esta herramienta, suponga el siguiente foro de discusión (Ejemplo 1).

O <Pedro,12-11-98,12:30>
 La fracción $\frac{1}{2}$ es propia.
 F <Juan, 12-11-98,12:35>
 Es cierto, porque $1 < 2$.
 A <Marta,12-11-98,12:45>
 Quiere decir que el numerador es menor que el denominador.

O <Rosa, 12-11-98,12:33>
 La forma correcta de representar la solución de $\frac{2}{3} + \frac{3}{5}$ es $\frac{19}{15}$.
 F <José,12-11-98, 12:45>
 Si, porque estás representando el resultado como una fracción impropia.
 C <María, 12-11-98,12:50>
 Considero que debes representarlo como el número mixto $1\frac{4}{15}$, porque es la forma más simplificada de la expresión.

Ejemplo 1: Foro de discusión formado por hilos de discusión.

En este foro se han generado dos hilos de discusión, el primero iniciado por Pedro y el segundo iniciado por Rosa.

La opinión de Pedro que generó el primer hilo de discusión, recibió el apoyo de Juan y Marta realizó una aclaración acerca de lo expresado por Juan. En este

hilo, obsérvese el anidamiento de los enunciados escritos por los jugadores y el orden cronológico en que suceden.

Por su parte, el enunciado que dio inicio al hilo que generó Rosa, recibió un apoyo y un rechazo. El apoyo se lo brindó José y el rechazo María. Se dice que estos enunciados son evidencia acerca del enunciado que escribió Rosa, y que muestran el grado de aprobación y de rechazo de algunos jugadores con respecto a la opinión de Rosa.

En los hilos de discusión se puede apreciar que el orden cronológico entre un hilo y el otro sólo está limitado por la cronología de sus enunciados iniciales.

Ahora que se cuenta con la noción intuitiva del foro de discusión, es posible presentar la formalización del foro, empezando por sus elementos más simples.

8.2. ENUNCIADO

Un enunciado es el conjunto de palabras con que se expresa breve y sencillamente una idea, problema que se va a resolver, etc.

En el foro, un enunciado representa una opinión primaria, evidencia a favor, evidencia en contra o una aclaración. Cada enunciado tiene las siguientes propiedades: texto, identificación, naturaleza, estado de plausibilidad y evidencia asociada.

Formalmente, un enunciado e es una 5-upla $e = \langle tx, id, n, p, V \rangle$ donde:

tx Es el texto del enunciado que además puede contener gráficos, sonido, etc.

id Es la identificación del enunciado por medio del nombre del jugador que lo escribe y, la fecha y hora en que lo hace.

n Es la naturaleza del enunciado, su clase. Es decir, es una clasificación subjetiva del enunciado respecto a su enunciado padre (si es que lo tiene) en un foro.

p Es el estado de plausibilidad del enunciado en una etapa dada de la discusión. El estado de plausibilidad de un enunciado se deduce a partir de la evaluación de las naturalezas de sus enunciados hijos.

V Es el conjunto de enunciados hijos de e , cuyas naturalezas permiten determinar su estado de plausibilidad. Al conjunto V se le denomina evidencia asociada a e .

Por notación, para un enunciado e , su texto, identificación, naturaleza, estado de plausibilidad, evidencia asociada se identifica con e_{tx} , e_{id} , e_n , e_p , e_v respectivamente.

El conjunto de todos los enunciados de un foro de discusión es E .

Con la finalidad de ser más explícitos, estas propiedades de un enunciado se definen formalmente a continuación.

8.2.1. IDENTIFICACIÓN DE UN ENUNCIADO

La identificación es una tríada que hace identificable un enunciado $e \in E$. La tríada de identificación es $e_{id} = \langle nn, ii, ff \rangle$, donde:

nn es el identificador visible o externo del enunciado, que puede ser el nombre o un seudónimo del jugador que escribió el enunciado, en una discusión sin anonimato; si hay anonimato, *nn* toma el valor de ?? para indicar que se desconoce quien escribió el enunciado.

ii es el identificador interno (protegido y encriptado) del jugador que escribió el enunciado. Es único para cada jugador.

ff es la fecha y hora en que se escribió el enunciado. El campo *ff.dd* permite identificar la fecha propiamente (dd-mm-aaaa) y el campo *ff.hh*, la hora.

Es importante destacar la relevancia de estos tres identificadores. El identificador *ii* permite que la persona que escribió un enunciado, aún en anonimato, sea la única que pueda modificarlo (esta operación se define posteriormente). El identificador *ff* facilita la conservación del orden cronológico de los enunciados. Y el identificador *nn* beneficia la creación de la conciencia de grupo en una discusión sin anonimato.

8.2.2. NATURALEZA DE UN ENUNCIADO

La naturaleza de un enunciado e , denotada por e_n , es una connotación subjetiva que el creador del enunciado le da a éste respecto a su enunciado padre (si es que lo tiene) en un foro. No obstante, la naturaleza adopta un valor de suma importancia para determinar el estado de plausibilidad del nodo padre.

Este valor se asigna según el tipo de opinión reflejada en e de la siguiente forma:

- i. *Opinión primaria*: Es una opinión principal que se desea someter a discusión; es el enunciado que origina un hilo de discusión. Se denota por el valor 0. El

conjunto de todas las opiniones primarias de un foro de discusión es denotado por O , y se define de la siguiente manera $O = \{e \in E / e_n = \mathbf{O}\}$. Las opiniones primarias no tienen enunciado padre en un foro.

- ii. *Aclaración*: Este es un comentario sobre otro enunciado (el padre) y está sujeto a discusión. Está identificado por el valor \mathbf{A} . El conjunto de todas las aclaraciones de un foro de discusión es denotado por A , y se define como sigue:

$$A = \{e \in E / e_n = \mathbf{A}\}$$

- iii. *Evidencia a favor*: Manifestación que favorece a otro enunciado de la discusión. Se denota por el valor \mathbf{F} . El conjunto de todas las evidencias a favor de un foro de discusión es denotado por F , y se define por $F = \{e \in E / e_n = \mathbf{F}\}$

- iv. *Evidencia en contra*: Manifestación que desfavorece a otro enunciado de la discusión. Se identifica por el valor \mathbf{C} . El conjunto de todas las evidencias en contra de un foro de discusión es denotado por C , y se define mediante el conjunto $C = \{e \in E / e_n = \mathbf{C}\}$

- v. *Desactivado*: Es producto de la acción de un jugador o del análisis de la plausibilidad de un hilo de discusión; y no tiene validez para el análisis y discusión en el foro. Se denota por el valor \mathbf{D} . El conjunto de todos los enunciados desactivados de un foro de discusión es denotado por D , y se define como sigue: $D = \{e \in E / e_n = \mathbf{D}\}$

El conjunto de las posibles naturalezas de los enunciados de un foro de discusión se denota con N y se define de la siguiente manera: $N = \{\mathbf{O}, \mathbf{A}, \mathbf{F}, \mathbf{C}, \mathbf{D}\}$.

Nótese que, los conjuntos O , A , F , C , y D forman una partición⁹ de E :

$$1. \forall X, Y \in \{O, A, F, C, D\}, \text{ implica que } X \neq Y, X \cap Y = \emptyset$$

$$2. E = O \cup A \cup F \cup C \cup D$$

8.2.3. ESTADO DE PLAUSIBILIDAD

El estado de plausibilidad de un enunciado se refiere a su grado de aceptabilidad por parte de los jugadores.

El estado de plausibilidad, e_p , de un enunciado $e \in E$, puede ser

- i. *Acceptable*: la evidencia en contra de la plausibilidad del enunciado e no existe o es despreciable, pudiendo existir evidencia a favor.
- ii. *Satisfactorio*: existe evidencia significativa a favor del enunciado e y la evidencia en contra de la plausibilidad de e no existe o es despreciable.

⁹ Una colección $Z = \{S_i\}$ de conjuntos no vacíos forman una partición de un conjunto S si:

- los conjuntos son disjuntos entre sí, es decir, $S_i, S_j \in Z$ y $i \neq j$ implica que $S_i \cap S_j = \emptyset$, y
- su unión es S , es decir, $S = \bigcup_{S_i \in Z} S_i$.

- iii. *Indeterminado*: no se sabe si existe o no evidencia en contra de la plausibilidad del enunciado e . Este es el estado inicial de cada nuevo enunciado.
- iv. *Dudoso*: la evidencia en contra de la plausibilidad del enunciado e no es despreciable.
- v. *Insatisfactorio*: existe evidencia significativa en contra de la plausibilidad de e .

El conjunto de estados de plausibilidad es $EP = \{ Ace, Sat, Ind, Dud, Ins \}$.

La correspondencia entre un enunciado y su estado de plausibilidad está dada por la siguiente función:

$$fp : E \rightarrow EP \text{ tal que } fp(e) = e_p$$

o bien, $fp = \{ (e, e_p) / e \in E \text{ y } e_p \in EP \}$

donde e_p es el resultado de considerar la naturaleza de todos los enunciados que conforman la evidencia de e .

Por otra parte, si $e_n = D$, entonces $e_p = nil$ para indicar que el enunciado no tiene estado de plausibilidad.

Para todos los propósitos prácticos de esta tesis, se usará la expresión “no existe evidencia en contra” como equivalente a “la evidencia en contra no existe o es despreciable”.

La evidencia asociada a un enunciado se definirá posterior al foro de discusión para facilitar su comprensión, mientras a continuación se define formalmente un hilo de discusión.

8.3. HILOS DE DISCUSIÓN (THREADED DISCUSSIONS)

Un hilo de discusión, respecto a un enunciado $e \in E$, es un árbol n -ario¹⁰ con e como raíz y cuyos otros nodos son enunciados que se refieren a e (forman la evidencia de e). Para $e \in E$, h_e representa el hilo de discusión que empieza o nace en e . El nivel de la raíz de h_e es uno y el de sus hijos es dos.

Si x es un nodo de nivel $k-1$ y y un nodo de nivel k , conectado a x , entonces y representa un enunciado acerca de x . Se dice que y pertenece al hilo que empieza en x .

8.3.1. HILO DE DISCUSIÓN PRIMARIO

Un hilo de discusión primario es un hilo de discusión h_e donde $e \in O$. (Figura 1)

¹⁰ Un árbol n -ario es un árbol posicional en el cual para cada nodo los hijos etiquetados mayores que n no existen. En un árbol posicional los hijos de un nodo son etiquetados con enteros positivos distintos. Es un árbol libre (grafo no dirigido acíclico y conectado) en el cual uno de los nodos es distinguido de los otros. El nodo distinguido es llamado la raíz del árbol. Los nodos de este árbol se denominan nodos. [CoLe89]

8.3.2. HILO DE DISCUSIÓN SECUNDARIO

Un hilo de discusión secundario es un hilo de discusión h_e donde $e \in F \cup C$.

(Figura 1)

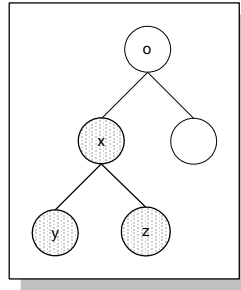


Figura 1: Hilo de discusión primario que empieza en O e hilo de discusión secundario que empieza en X.

8.4. FORO DE DISCUSIÓN

Un foro de discusión es una secuencia $fd = \langle h_1, h_2, h_3, \dots, h_n \rangle$ de hilos de discusión primarios, ordenada con respecto a la fecha y la hora de la raíz de los hilos de discusión primarios.

Un nodo e que pertenece a h_l se representa por e_{ij}^{lk} donde:

l es el l -ésimo hilo de discusión primario al cual pertenece el enunciado e , con $l=1,2,3,\dots,n$. Es decir, es el número secuencial del árbol en fd .

k es el nivel del nodo e en h_l , con $k = 1,2,3,\dots,m_l$. Si e es la raíz de h_l , entonces $k=1$. Si e es una hoja de h_l , entonces $k = m_l$.

i es la posición de e en el nivel k de h_l , $i = 1, 2, 3, \dots, z_k$. Dado un nivel k de h_l , sus nodos están ordenados cronológicamente, es decir, si $a < b$ con $a = 1, 2, 3, \dots, z_k$, $b = 1, 2, 3, \dots, z_k$, entonces e_{aj}^{lk} tiene fecha anterior o igual a la fecha de e_{bj}^{lk} .

j es la posición del padre de e en el nivel $k-1$, con $j = 1, 2, 3, \dots, z_{k-1}$, $j = 0$ para indicar que e es la raíz.

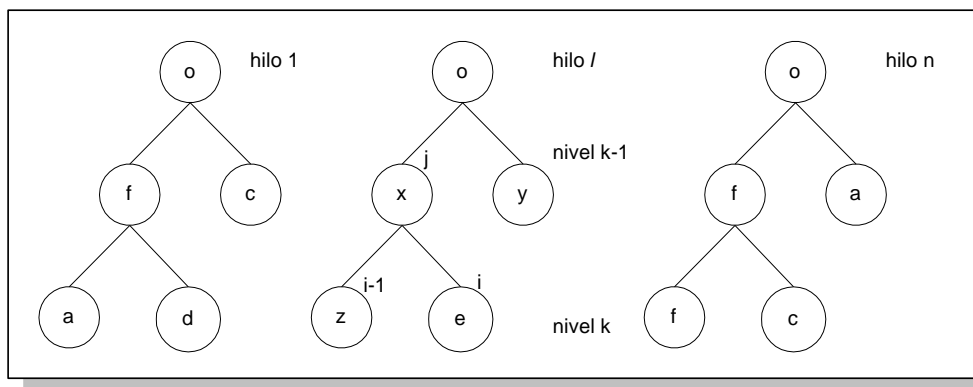


Figura 2: Foro de discusión.

8.5. EVIDENCIA

La evidencia de un enunciado $e \in E$ es el conjunto de enunciados que se escriben con el fin principal de favorecer o desfavorecer a e .

En esta sección se define formalmente el conjunto que representa la evidencia asociada a un enunciado.

También se define lo que se denominará la evidencia distinguible a favor (en contra) de un enunciado, que en términos generales se refiere al conjunto formado por la evidencia a favor (en contra) de jugadores distintos que apoya (rechaza) a un enunciado.

8.5.1. EVIDENCIA ASOCIADA A UN ENUNCIADO

La evidencia x_v asociada a un enunciado $x \in E$ es el conjunto de nodos que pertenecen al hilo de discusión h_x , excluyendo a x .

8.5.2. EVIDENCIA DISTINGUIBLE

Antes de definir propiamente la evidencia distinguible es oportuno describir el apoyo a la naturaleza de un enunciado.

El **apoyo a la naturaleza de un enunciado** $e \in E$, denotado por $apoyo_e$ es una medida del grado de aceptabilidad de la naturaleza de dicho enunciado, basado en la evidencia a favor y la evidencia en contra, tal que $0 \leq apoyo_e \leq 1$.

A continuación se define propiamente la evidencia distinguible, la evidencia distinguible a favor y la evidencia distinguible en contra.

Sea $e_v = \{f_1, f_2, f_3, \dots, f_m\}$ el conjunto de la evidencia asociada a un enunciado $e \in E$ de nivel k ; y sea $e'_v = \{f \in e_v / (f_n = \mathbf{F} \vee f_n = \mathbf{C}) \wedge (\text{nivel de } f = k + 1)\}$, es decir, el conjunto de la evidencia a favor y de la evidencia en contra del enunciado $e \in E$.

Considere una llave que distingue a un enunciado de cualquier otro formada por la naturaleza del enunciado e_n y el identificador interno ii del jugador que lo escribió, denotada por $llave_e = e_n + ii$.

Se define la evidencia distinguible de un enunciado $e \in E$ como el conjunto

$$e''_v = \left\{ f \in e'_v / \exists g \in e'_v : (llave_g = llave_f) \wedge (apoyo_g > apoyo_f) \right\}.$$

Por su parte, la evidencia distinguible a favor se define como el conjunto

$$e_v f = \{f \in e_v'' / f_n = F\}$$

y la evidencia distinguible en contra como el conjunto

$$e_v c = \{f \in e_v'' / f_n = C\}.$$

8.6. EVOLUCIÓN DE LA NATURALEZA DE LOS ENUNCIADOS

La naturaleza de los enunciados en un foro puede ser cambiada por los jugadores al presentarse las diversas alternativas o situaciones del juego. A estos cambios se les denomina evolución de la naturaleza de los enunciados y se describe formalmente a continuación.

La evolución $evol$ de la naturaleza de un enunciado $e \in E$ contiene todas la naturalezas que adquiere un enunciado en la discusión y se denota por la secuencia

$$evol_e = \langle n_1, n_2, \dots, n_{i-1}, n_i, n_{i+1}, \dots, n_m \rangle \text{ con } n_i \in N \text{ } i = 1, 2, 3, \dots, m.$$

La relación de orden de esta secuencia se denota por $<$ significando “precede cronológicamente a”, así por ejemplo, $n_i < n_j$ si y sólo si el enunciado e toma la naturaleza n_i antes que la naturaleza n_j .

Sobre la secuencia $evol$ es necesario definir tres funciones: una que agregue a la secuencia la nueva naturaleza de un enunciado, denominada adi ; la otra que determine la naturaleza de un enunciado a partir de una previa, que se denomina

suc, y una última que determine la naturaleza anterior de un enunciado a partir de la que tiene, denominada *ant*.

La función *adi* que agrega la nueva naturaleza n_{m+1} de un enunciado $e \in E$ en su secuencia $evol_e$ se define de la siguiente manera:

$$adi : N \times evol_e \rightarrow evol_e \cup \langle n_{m+1} \rangle \text{ tal que } adi(n_{m+1}, evol_e) = \langle n_1, n_2, \dots, n_m, n_{m+1} \rangle$$

o bien,

$$adi = \left\{ \left(\left(n_{m+1}, \langle n_1, n_2, \dots, n_m \rangle \right), \langle n_1, n_2, \dots, n_m, n_{m+1} \rangle \right) / n_{m+1} \in N \right\}$$

Esta función es de gran utilidad para las operaciones sobre un hilo de discusión, en especial para añadir enunciados de naturaleza \mathbb{F} , \mathbb{C} o \mathbb{A} , así como también cuando se modifica la naturaleza de un enunciado, en la retractación y desactivación, como se verá en la siguiente sección.

La función que calcula el sucesor sobre la secuencia *evol* se denota *suc* y se define en los siguientes términos:

$$suc : evol_e \rightarrow evol_e \text{ tal que } suc(n_i) = \begin{cases} n_{i+1} & \text{para } i = 1, 2, \dots, m-1 \\ nil & \text{para } i = m \end{cases}$$

o bien,

$$suc = \left\{ (n_i, n_{i+1}) / n_i, n_{i+1} \in evol \text{ para } i = 1, 2, \dots, m-1 \right\} \cup \left\{ (n_m, nil) / n_m \in evol \right\}$$

nil se utiliza para indicar que no existe el sucesor.

Esta función es útil para recorrer la secuencia $evol_e$.

La función que calcula el antecesor sobre la secuencia $evol$ se denota ant y se define en los siguientes términos:

$$ant : evol_e \rightarrow evol_e \text{ tal que } ant(n_i) = \begin{cases} nil & \text{para } i = 1 \\ n_{i-1} & \text{para } i = 2, 3, \dots, m \end{cases}$$

o bien,

$$ant = \{(n_i, n_{i-1}) / n_i, n_{i-1} \in evol \text{ para } i = 2, 3, \dots, m\} \cup \{(n_m, nil) / n_m \in evol\}$$

al igual que en la función anterior, nil se utiliza para indicar que no tiene antecesor.

Esta función se utiliza en la operación de reactivar un enunciado, como se discute en la siguiente sección.

En la siguiente sección se definen las operaciones que se pueden realizar sobre el foro de discusión.

8.7. OPERACIONES SOBRE EL FORO DE DISCUSIÓN

En un foro de discusión se insertan enunciados y los jugadores cambian naturalezas de los enunciados, entre otras actividades. Por lo tanto, se vuelve importante definir un conjunto de operaciones que faciliten la interacción de los jugadores mediante el foro de discusión.

Las operaciones sobre el foro de discusión son las siguientes: añadir, modificar y consultar.

8.7.1. AÑADIR

La operación añadir permite agregar un enunciado $e \in E$ al foro de discusión. Dependiendo de la naturaleza del enunciado, ocurren los siguientes casos.

- i. *Añadir una opinión primaria O*: Se crea la raíz de un nuevo árbol n-ario al final de la secuencia del foro de discusión. Si $fd = \langle h_1, h_2, h_3, \dots, h_n \rangle$ es el estado del foro antes de añadir el enunciado e , entonces $fd' = \langle h_1, h_2, h_3, \dots, h_n, h_{n+1} \rangle$ es el estado del foro después de añadir el enunciado e donde h_{n+1} es un árbol (de nivel 0) donde la raíz es la opinión primaria de naturaleza O.
- ii. *Añadir una aclaración A, una evidencia a favor F o una evidencia en contra C*: Este enunciado que se agrega es evidencia asociada de un enunciado $e' \in E$. El enunciado $e \in E$ es el hijo más reciente de e' , es decir, se agrega un nuevo hijo a la derecha del último hijo del nodo padre e' en el hilo de discusión al cual pertenece e' .

8.7.2. MODIFICAR

Durante una discusión los jugadores, en algunas ocasiones, querrán cambiar ciertos enunciados que han escrito o sus características; razón por la cual se vuelve necesario definir operaciones que permitan estas tareas.

La operación modificar admite la alteración del contenido del enunciado, además de su naturaleza. También permite desactivar un enunciado dentro del foro de discusión y volverlo a activar. En particular, estas modificaciones producen los cambios que se describen a continuación en cada una de ellas.

- i. *Modificar el texto*: Esta modificación se refiere al deseo del jugador de cambiar el texto del enunciado. En este caso el foro de discusión no es alterado, a no ser el contenido del enunciado. Si la modificación es de la semántica del enunciado, el jugador debería tener el cuidado de considerar la naturaleza, si fuera necesario.
- ii. *Modificar la naturaleza*: Permite al jugador modificar la naturaleza de un enunciado $e \in E$. En el hilo de discusión primario al cual pertenece e ocurre lo siguiente: a) el enunciado e cambia de naturaleza; b) los enunciados de h_e excepto e toman la naturaleza \mathbb{A} para permitir que se les siga agregando evidencia; c) a la secuencia *evol* de cada uno de los nodos del árbol h_e se le agrega la nueva naturaleza, mediante la función de evolución *adi*. Por otro lado, el enunciado e con su nueva naturaleza queda dispuesto para que se le agregue nueva evidencia.
- iii. *Retractación*: La retractación permite renunciar a un enunciado, es la eliminación del nodo para la discusión y el análisis de plausibilidad. En el hilo de discusión primario al cual pertenece $e \in E$ ocurre lo siguiente: a) el enunciado e toma la naturaleza \mathbb{D} ; b) los enunciados de h_e , excepto e , toman la naturaleza \mathbb{A} para permitir que se les siga agregando evidencia; c) a la secuencia *evol* de cada uno de los nodos del árbol h_e se le agrega la nueva naturaleza, por medio de la función de evolución *adi*; d) el estado de plausibilidad del enunciado e es *nil*. Por otra parte, el enunciado e con su nueva naturaleza no queda disponible para que se le agregue nueva evidencia.
- iv. *Desactivar/reactivar*: Cualquier nodo del árbol puede ser desactivado (para el análisis de plausibilidad). Si existe un árbol que nace en el nodo también éste se desactiva. En el hilo de discusión primario al que pertenece $e \in E$ ocurre lo siguiente: a) el enunciado e que se desea desactivar toma la naturaleza \mathbb{D} ; b)

los enunciados de h_e también toman la naturaleza D ; c) a la secuencia *evol* de cada uno de los nodos del árbol h_e se le agrega la nueva naturaleza D , empleando la función de evolución *adi*; d) el estado de plausibilidad de h_e es *nil*. Por otra parte, los nodos del árbol h_e con su nueva naturaleza no quedan disponibles para que se le agregue nueva evidencia. Un nodo que ha sido desactivado previamente puede ser reactivado, con la consiguiente reactivación del árbol que nace en él.

En cualquiera de los casos de modificación, es necesario comprobar que el jugador que desea realizarla sea el autor, o bien que esté autorizado. Esta verificación se lleva a cabo mediante el identificador interno *ii*.

8.7.3. CONSULTAR

En un momento dado de la discusión cualquiera de los jugadores puede desear ver sólo algunos de los enunciados del foro, o simplemente ver todo el foro de discusión. Las siguientes operaciones permiten realizar algunas consultas en el foro de discusión.

- i. *Ver todo*: Permite ver toda la información que se encuentra en el foro de discusión. Es decir, se muestran todos los hilos de discusión y sus respectivos nodos.
- ii. *Ver lo leído*: La información que el jugador ya leyó se denota mediante un color *x*. En esta operación se muestran los nodos cuya información ya se leyó y los padres de dichos nodos, aún cuando no se hayan leído, hasta llegar a la raíz del hilo de discusión primario de cada árbol *n*-ario. Si en los nodos mostrados hay información que no se ha leído se distingue con un color *y*.

- iii. *Ver lo no leído*: La información que el jugador aún no ha leído se denota mediante un color y . En esta operación se muestran los nodos cuya información no ha sido leída y los padres de dichos nodos, aún cuando ya se hayan leído, hasta llegar a la raíz del hilo de discusión primario de cada árbol n -ario. Si en los nodos mostrados hay información que se ha leído se distingue con su color x .
- iv. *Ver sólo enunciados O*: Se muestran sólo las raíces de los hilos de discusión primarios.
- v. *Ver sólo enunciados F, C o A*: En esta operación se muestran los nodos cuya naturaleza es F , C o A . Para ello, en cada árbol n -ario, se muestran dichos nodos y sus padres, aún cuando no sean de la naturaleza en cuestión, hasta llegar a la raíz del hilo de discusión primario. Si en los nodos mostrados hay información que no es de la naturaleza en cuestión se distingue con otro color z .
- vi. *Abrir y cerrar un hilo de discusión h_e* : En esta operación se muestra o no todos los enunciados del árbol que empieza en el nodo e . Para ello en el árbol n -ario se muestra el padre e y todos sus hijos.

8.8. RETROALIMENTACIÓN

Es importante para los jugadores conocer cuál es el criterio del grupo acerca de lo que se discute en cada hilo de discusión en el foro. Este concepto se denomina *retroalimentación*.

Con la finalidad de ofrecer a los jugadores una retroalimentación de la discusión, se realiza un análisis sobre cada hilo de discusión primario basado en la Teoría

de Plausibilidad y el Modelo de Votación en Grupos Grandes. Este análisis ofrece a los jugadores el grado de aprobación y el grado de rechazo que tiene el grupo con respecto a cada hilo de discusión.

En general, el grado de aprobación y el grado de rechazo se obtiene recorriendo cada hilo de discusión primario h_l desde las hojas hasta la raíz, es decir, del enunciado $e_{\max(i) \max(j)}^{l \max(k)}$ al enunciado e_{10}^{l0} con $l = 1, 2, 3, \dots, n$; de acuerdo con las propiedades de plausibilidad y el algoritmo DFS-Plausible que se describen a continuación.

8.8.1. PROPIEDADES DE PLAUSIBILIDAD

Las siguientes propiedades del estado de plausibilidad de los enunciados dentro del foro de discusión son útiles para el algoritmo DFS-Plausible.

1. El estado de plausibilidad de las hojas con naturaleza F o C de cualquier árbol es *acceptable*. Las demás hojas tienen estado de plausibilidad *nil*. Es decir, los enunciados $e_{ij}^{l \max(k)}$ con $e_n = F$ o $e_n = C$ tienen $e_p = \text{acceptable}$.
2. La naturaleza e_n de un enunciado $e \in E$ no debe tomarse en cuenta para el análisis de plausibilidad cuando su estado de plausibilidad $e_p \in \{Dud, Ins, Ind\}$.

8.8.2. POLÍTICA DE CONSENSO

Las políticas de consenso son necesarias para la búsqueda del consenso, pues ellas marcan las pautas para determinar el grado de aprobación y el grado de rechazo del grupo respecto a un objeto de discusión. En esta sección se define la política de consenso adoptada.

El grado de aprobación y el grado de rechazo de un problema reflejan la aceptabilidad del objeto de discusión; con el propósito de establecer estos grados se definen unos parámetros para la determinación de la aceptabilidad del objeto de discusión. Estos parámetros se denominan marca de aprobación (m_a), umbral de aprobación (u_a), marca de rechazo (m_r) y umbral de rechazo (u_r).[Oliv96]

La marca de aprobación (m_a) y el umbral de aprobación (u_a) se utilizan para expresar el grado de aprobación; se encuentran en los siguientes rangos del conjunto de los números racionales: $0 < m_a < 1$ y $0 < u_a < m_a$.

Por su parte, la marca de rechazo y el umbral de rechazo se utilizan para expresar el grado de rechazo; estas marcas están definidas en los siguientes rangos del conjunto de los números racionales: $0 < m_r < 1$ y $0 < u_r < 1$.

La política de consenso está compuesta por la política de consenso de aprobación y la política de consenso de rechazo, y se puede definir con base en las marcas y umbrales antes descritos; así la política de consenso de aprobación para un nodo e con evidencia e_v asociada está dada por la relación

$$m_a \geq c_1 \wedge u_a \geq c_2 \text{ donde } c_1, c_2 \in [0,1] \subset Q \text{ con } c_2 \leq c_1$$

y la política de consenso de rechazo para dicho nodo está determinada por

$$m_r \geq c_3 \wedge u_r \geq c_4 \text{ donde } c_3, c_4 \in [0,1] \subset Q \text{ con } c_4 \leq c_3$$

Las constantes c_1 , c_2 , c_3 y c_4 existen paralelamente para determinar la política de consenso.

Los valores que serán utilizados para determinar la plausibilidad de un enunciado e respecto a las marcas y umbrales de referencia se denominan *apoyo de la*

evidencia a favor, denotado $\text{apoyo}_e F$, y *apoyo de la evidencia en contra*, en cuyo caso se denota $\text{apoyo}_e C$. Estos valores representan una medida de la forma en que la evidencia a favor (en contra) favorece (desfavorece) al enunciado e .

Sea J el conjunto de jugadores y $e \in E$. El apoyo de la evidencia a favor de e , $\text{apoyo}_e F$, y de la evidencia en contra de e , $\text{apoyo}_e C$, se definen de la siguiente manera:

$$\text{apoyo}_e C = \frac{\sum_{x \in e_v C} \text{apoyo}_x}{|J|}$$

$$\text{apoyo}_e F = 1 - \text{apoyo}_e C$$

El cálculo del apoyo se basa en el principio de “el que calla otorga”, en el sentido de que la inexistencia de oposición a un enunciado puede asumirse como aprobación--al menos hasta que la intención se haga explícita. A su vez, eso implica que la existencia de argumentos a favor explícitos no son más que una reafirmación de la inexistencia de oposición de parte de un jugador.

Los apoyos calculados con las fórmulas anteriores serán confrontados con las marcas y umbrales establecidos para obtener los estados de plausibilidad y el apoyo a la naturaleza de un enunciado de la siguiente manera:

- i. si $\text{apoyo}_e F \geq m_a \wedge \text{apoyo}_e C < u_r$, entonces $fp(e) = Sat$ y $\text{apoyo}_e = \text{apoyo}_e F$
- ii. si $\text{apoyo}_e F \geq u_a \wedge \text{apoyo}_e C < u_r$, entonces $fp(e) = Ace$ y $\text{apoyo}_e = \text{apoyo}_e F$
- iii. si $\text{apoyo}_e C \geq u_r$, entonces $fp(e) = Dud$

iv. si $\text{apoyo}_e C \geq m_r$, entonces $fp(e) = \text{Ins}$

v. si $\text{apoyo}_e F < u_a \wedge \text{apoyo}_e C < u_r$, entonces $fp(e) = \text{Ind}$

8.8.3. ALGORITMO DFS-PLAUSIBLE

Este algoritmo, basado en el algoritmo de búsqueda en profundidad (depth-first search) [CoLe89], permite determinar el estado de plausibilidad y el apoyo de un hilo de discusión primario en un foro de discusión.

DFS-Plausible se puede describir en los siguientes términos.

1. El objetivo es propagar el estado de plausibilidad de cada hilo de discusión en el foro desde las hojas hasta la raíz. En la búsqueda en profundidad, las aristas son exploradas a partir del último nodo v descubierto y que aún tiene aristas que salen de él y que no han sido visitadas. Cuando todas las aristas de v han sido exploradas, se retrocede para visitar las aristas que salen del nodo a partir del cual v fue descubierto, determinando el estado de plausibilidad de v y su apoyo. Este proceso continúa hasta que se han descubierto todos los nodos que pertenecen a un árbol (hilo de discusión primario). Si aún quedan árboles que no se han descubierto, entonces la raíz de uno de ellos es seleccionada como nueva fuente y la búsqueda se repite a partir de ella. Todo este proceso se repite hasta que cada árbol sea recorrido. Los nodos se marcan durante la búsqueda de la siguiente forma; cada nodo es inicialmente blanco, es gris cuando es descubierto en la búsqueda, y es negro cuando ha finalizado, es decir, cuando su lista de adyacencia ha sido examinada completamente.

2. Para calcular el estado de plausibilidad de los nodos internos hasta la raíz se realizan las siguientes actividades:

- 2.1. *Inicialización.* Se inicializa en uno el apoyo a la naturaleza de cada nodo, es decir, $\forall e \in E, \text{apoyo}_e = 1$.
- 2.2. *Selección.* Si existe más de un enunciado con naturaleza F o C como evidencia asociada a un nodo x de un mismo jugador, se selecciona el nodo con mayor apoyo a su naturaleza, por tener más respaldo del resto de los jugadores. Esto se hace con el fin de que el apoyo a favor o el apoyo en contra represente el sentir de la mayoría de los jugadores. La identificación de estos enunciados se realiza mediante la llave del nodo.
- 2.3. *Cálculo $\text{apoyo}(F)$ y $\text{apoyo}(C)$.* Para determinar el grado de aprobación y el grado de rechazo por parte de los jugadores con respecto a un nodo x , se calcula el estado de plausibilidad de dicho nodo con base en su evidencia. El grado de rechazo se calcula utilizando la fórmula $\text{apoyo}_e C$ y el principio de selección descrito en el párrafo anterior, por su parte del grado de aprobación se obtiene mediante la fórmula $\text{apoyo}_e F$. La fórmula $\text{apoyo}_e C$ determina la razón entre la evidencia en contra distinguible $e_v C$ y del total de jugadores que participan del juego.
- 2.4. *Determinación del estado de plausibilidad de un nodo x .* De acuerdo con la política de consenso utilizada y con los resultados del paso anterior se obtiene el grado de aprobación y el grado de rechazo del nodo x por parte de los jugadores y el consiguiente estado de plausibilidad del nodo.


```

5.      do if  $color[u] = \text{BLANCO}$                                 ** El nodo  $u$  no se ha visitado
6.      then DFSP-VISITA( $u$ )

DFSP-VISITA( $u$ )
1.       $color[u] \leftarrow \text{GRIS}$                                 ** El nodo  $u$  se ha descubierto
2.      for cada  $v \in \text{Ady}[u]$                                 ** Se explora la arista  $(u, v)$ 
3.      do if  $color[v] = \text{BLANCO}$ 
4.      then DFSP-VISITA( $v$ )
5.      if  $llave[v] = llave[v_{i-1}]$                                 **  $v_{i-1}$  es visitado antes que  $v_i$ 
6.      then  $sel[v] \leftarrow \max(\text{apoyo}[v], \text{apoyo}[v_{i-1}])$ 
7.      else  $sel[v] \leftarrow \text{apoyo}[v]$  ** Considerar en  $sel$   $e_n = C$ 
8.       $color[u] \leftarrow \text{NEGRO}$                                 ** El árbol que nace en  $u$  fue visitado
9.      if  $u$  no ES HOJA
10.     then  $\text{sumaC} = \sum sel[u]$                                 ** Sumar sólo los  $e \in E$  con  $e_n = F$ 
11.      $\text{apoyoC} = \text{sumaC} / n_{\text{Jugadores}}$ 
12.      $\text{apoyoF} = 1 - \text{apoyoC}$ 
13.     if  $\text{apoyoF} \geq m_a$  and  $\text{apoyoC} < u_r$ 
14.     then  $e_p[u] = \text{SATISFACTORIO}$ 
15.      $\text{apoyo}[u] = \text{apoyoF}$ 
16.     else if  $\text{apoyoF} \geq u_a$  and  $\text{apoyoC} < u_r$ 
17.     then  $e_p[u] = \text{ACEPTABLE}$ 
18.      $\text{apoyo}[u] = \text{apoyoF}$ 
19.     else if  $\text{apoyoC} > m_r$ 
20.     then  $e_p[u] = \text{INSATISFACTORIO}$ 
21.      $e_n[u] = D$ 
22.     else if  $\text{apoyoC} \geq u_r$ 
23.     then  $e_p[u] = \text{DUDOSO}$ 
24.      $e_n[u] = D$ 
25.     else  $e_p[u] = \text{INDETERMINADO}$ 
26.      $e_n[u] = D$ 

```

El procedimiento DFS-PLAUSIBLE trabaja como sigue. Las líneas de 1-3 pintan todos los nodos en blanco e inicializan su *apoyo* en 1. Las líneas 4-6 verifican el color de cada nodo en E en el ciclo y, cuando un nodo blanco es encontrado, se visita usando DFSP-VISITA. Cada vez que DFSP-VISITA(u) es llamado en la línea 6, el nodo u se convierte en la raíz de un nuevo árbol.

En cada llamada DFSP-VISITA(u), el nodo u es inicialmente blanco. La línea 1 pinta a u en gris. Las líneas 2-7 examinan cada nodo v adyacente a u y recursivamente visitan a v si es blanco; además verifican si los nodos v_{i-1} y v_i (que son hermanos) son distinguibles. Como cada nodo $v \in \text{Ady}[u]$ es considerado en

la línea 2, se dice que la arista (u, v) es explorada por la búsqueda en profundidad. Finalmente, después que cada arista que sale de u ha sido explorada, las líneas 8-26 pintan a u en negro y si u no es hoja calculan su estado de plausibilidad y su apoyo.

El tiempo de ejecución de DFS-PLAUSIBLE es el siguiente. Los ciclos en las líneas 1-3 y las líneas 4-6 de DFS-PLAUSIBLE toman tiempo $\Theta(E)$, el cual es el tiempo exclusivo para ejecutar las llamadas a DFSP-VISITA. El procedimiento DFSP-VISITA es llamado exactamente una vez por cada nodo $v \in E$, puesto que DFSP-VISITA es invocado sólo en los nodos blancos y la primera cosa que éste hace es pintar el nodo en gris. Durante la ejecución de DFS-VISITA(v), el ciclo en las líneas 2-7 es ejecutado $|Adj[v]|$ veces. Puesto que

$$\sum_{v \in E} |Adj[v]| = \Theta(S),$$

el costo total de ejecutar las líneas 2-3 de DFS-VISITA es $\Theta(S)$. El tiempo de ejecución de DFS-PLAUSIBLE es por lo tanto $\Theta(E+S)$.

8.9. EJEMPLO DE ESTADOS DE PLAUSIBILIDAD EN UN FORO

El foro de discusión del Ejemplo 2 está compuesto por tres hilos de discusión primarios, en el mismo se pueden notar diferentes niveles de anidamiento.

Suponga que son tres jugadores. Considere que la política de consenso establece las siguientes marcas y umbrales:

$$m_r = m_a = \frac{1}{2} \text{ y } u_r = u_a = \frac{1}{4}$$

Contiguo a cada hilo de discusión se encuentra su estado de plausibilidad y su aceptabilidad. Así, el primer hilo de discusión primario tiene $e_p = \text{satisfactorio}$ y $\text{apoyo} = 1$. El segundo hilo de discusión primario tiene $e_p = \text{satisfactorio}$ y $\text{apoyo} = 1$. El último tiene $e_p = \text{satisfactorio}$ y $\text{apoyo} = 2/3$.

□0 <Nombre1, Hora-Fecha> -----	e_{10}^{11}	$e_p = \text{satisfactorio}$ $\text{apoyo} = 1$
F <Nombre2, Hora-Fecha> -----	e_{11}^{12}	
A <Nombre3, Hora-Fecha> -----	e_{21}^{12}	
□0 <Nombre3, Hora-Fecha> -----	e_{10}^{21}	$e_p = \text{satisfactorio}$ $\text{apoyo} = 1$
□0 <Nombre2, Hora-Fecha> -----	e_{10}^{31}	$e_p = \text{satisfactorio}$ $\text{apoyo} = 2/3$
F <Nombre1, Hora-Fecha> -----	e_{11}^{32}	$e_p = \text{insatisfactorio}$
C <??, Hora-Fecha> -----	e_{11}^{33}	
C <Nombre3, Hora-Fecha> -----	e_{21}^{33}	
C <Nombre3, Hora-Fecha> -----	e_{21}^{32}	$e_p = \text{satisfactorio}$ $\text{apoyo} = 1$
F <??, Hora-Fecha> -----	e_{12}^{33}	
F <Nombre2, Hora-Fecha> -----	e_{22}^{33}	
F <Nombre1, Hora-Fecha> -----	e_{32}^{33}	

Ejemplo 2: Ejemplo de un foro de discusión al que se le ha aplicado el algoritmo DFS-Plausible.

En el Ejemplo 2 se observa que a los enunciados de tipo 0 les precede un cuadro donde se reflejará el estado de plausibilidad de dicho enunciado. En todos los enunciados se observa que contiguo a su símbolo se encuentra la identificación del jugador que lo escribió.

8.10. VOTACIÓN

La votación es una actividad de selección que está presente en todo grupo. El método de votación está basado en el Modelo de Votación en Grupos Grandes

[Oliv96] y tiene dos formas de llevarse a cabo: votar por un enunciado en el foro de discusión y votar por la solución de algún problema en el juego.

8.10.1. MÉTODO DE VOTACIÓN

El método de votación es una adaptación en el foro del Modelo de Votación en Grupos Grandes basado en la Teoría de Plausibilidad.[Oliv96] .

En una situación de toma de decisiones en grupo existen decisores y problemas de decisión. En una actividad de votación, los problemas de decisión toman frecuentemente la forma de opciones. La evaluación del grupo está basada en la evaluación de cada individuo y en la influencia o peso que el individuo tiene en toda la decisión.

Cuando se considera una opción, un decisor debería prestar atención a la evidencia positiva y a la negativa. La evidencia a favor o la evidencia en contra de una opción está representada en términos del grado de aprobación y del grado de rechazo. La plausibilidad (o aceptabilidad) de una opción es calculada combinando la evidencia expresada por cada individuo.

8.10.1.1. PRINCIPIOS DE VOTACIÓN

El esquema de votación está basado en cinco principios que representan especificaciones de alto nivel o restricciones de la Teoría de Plausibilidad:

1. Cada decisor puede expresar aprobación y rechazo, al mismo tiempo.
2. La aprobación y el rechazo pueden ser expresados en diferentes grados.

3. Los decisores pueden tener diferentes niveles de influencia en la decisión.
4. El grado de aprobación del grupo es calculado combinando los grados de aprobación de todos los miembros, ajustados con respecto a sus niveles de influencia. Las consideraciones correspondientes se aplican al grado de rechazo.
5. Las opciones son clasificadas en estados de plausibilidad de acuerdo con sus grados de aprobación y rechazo asignados.

8.10.1.2. NORMAS DE VOTACIÓN

Derivado de los principios, el esquema de votación incluye normas más específicas para estructurar el proceso de votación basado en la teoría de plausibilidad, como sigue:

1. En los juegos a cada jugador se le asigna la misma cantidad de bolas blancas y bolas azules. Formalmente, sea \mathbb{N} el conjunto de los números naturales y J el conjunto de los jugadores, se define la función a_b que asigna a cada jugador una cantidad de bolas blancas, de la siguiente forma:

$$a_b : J \rightarrow \mathbb{N} \text{ tal que } a_b(j_i) = k_b$$

por su parte, a_z es la función que determina la cantidad de bolas azules asignadas a cada decisor (jugador):

$$a_z : J \rightarrow \mathbb{N} \text{ tal que } a_z(j_i) = k_z$$

2. El número de bolas blancas y azules asignadas a un decisor representa el nivel de influencia del decisor en el grupo. No obstante, para los efectos de esta investigación el nivel de influencia es igual para todos los jugadores.
3. El grado de aprobación de una opción o por un decisor j es el número de bolas blancas asignadas por el decisor j a la opción o . El grado de rechazo de una opción o por un decisor j es el número de bolas azules que j asigna a o .

Formalmente, c_b es una función que determina la cantidad de bolas blancas que el decisor j_i asigna a la opción o_j , que debe ser menor o igual que $a_b(j_i)$:

$$c_b : J \times O \rightarrow \mathbb{N} \text{ tal que } c_b(j_i, o_j) = b_{ij} \text{ donde } Op = \{o_1, o_2, \dots, o_p / o \text{ es una opción}\}$$

por su parte, c_z es una función que determina la cantidad de bolas azules que el decisor j_i asigna a la opción o_j , que debe ser menor o igual que $a_z(j_i)$:

$$c_z : J \times O \rightarrow \mathbb{N} \text{ tal que } c_z(j_i, o_j) = z_{ij} \text{ donde } Op = \{o_1, o_2, \dots, o_p / o \text{ es una opción}\}$$

4. El grado de aprobación grupal de o_j es la suma algebraica de los grados de aprobación individuales de o_j , y el grado de rechazo grupal de o_j es la suma algebraica de los grados de rechazo individuales de o_j .

Formalmente, se define el grado de aprobación grupal como $b_j = \sum_{i=1}^n c_b(j_i, o_j)$ y se

$$\text{define el grado de rechazo grupal como } z_j = \sum_{i=1}^n c_z(j_i, o_j).$$

5. El esquema de votación incluye la definición de una marca de aprobación (m_a), un umbral de aprobación (u_a), marca de rechazo (m_r) y un umbral de rechazo (u_r). m_a debe estar entre uno y el número de bolas blancas asignadas al total

de decisores. u_a debe estar entre uno y m_a . m_r debe estar entre uno y el número de bolas azules asignadas al total de decisores. u_r debe estar entre uno y m_r .

Formalmente, $1 \leq m_a \leq \sum_{i=1}^n a_b(j_i); 1 \leq u_a \leq m_a;$

$$1 \leq m_r \leq \sum_{i=1}^n a_z(j_i); 1 \leq u_r \leq m_r$$

6. Un problema de decisión que ha sido sometido a un proceso de votación califica para un estado de plausibilidad de acuerdo con las siguientes definiciones.

- *Satisfactorio*: grado de aprobación mayor o igual que la marca de aprobación y el grado de rechazo menor que el umbral de rechazo. Es decir, $b_j \geq m_a \wedge z_j < u_r$.
- *Aceptable*: grado de aprobación mayor o igual que el umbral de aprobación y grado de rechazo menor que el umbral de rechazo. Es decir, $b_j \geq u_a \wedge z_j < u_r$.
- *Dudoso*: grado de rechazo mayor o igual que el umbral de rechazo. Es decir, $z_j \geq u_r$.
- *Insatisfactorio*: grado de rechazo mayor que la marca de rechazo. Es decir, $z_j \geq m_r$.
- *Indeterminado*: grado de aprobación menor que el umbral de aprobación y grado de rechazo menor que el umbral de rechazo. Es decir, $b_j < u_a \wedge z_j < u_r$.

8.10.1.3. POLÍTICA DE CONSENSO PARA VOTACIÓN

El consenso se refiere al grado de aprobación y al grado de rechazo requeridos para la determinación de la aceptabilidad de una opción. Existe un consenso de aprobación y un consenso de rechazo.

Las políticas de consenso definen la ubicación de las marcas de aprobación y de rechazo y de los umbrales de aprobación y de rechazo.

La marca y el umbral de aprobación son fracciones de la unidad del total de bolas blancas asignadas al total de decisores. La marca y el umbral de rechazo son fracciones de la unidad del total de bolas azules asignadas al total de decisores

Se denomina b_{comp} al total de bolas blancas que se toma como base para la definición de la marca y el umbral de aprobación y z_{comp} al total de bolas azules que se toma como base para la definición de la marca y el umbral de rechazo. Entonces,

$$m_a = k_1 b_{comp}, 0 < k_1 \leq 1$$

$$u_a = k_2 b_{comp}, 0 < k_2 \leq k_1$$

$$m_r = k_3 b_{comp}, 0 < k_3 \leq 1$$

$$u_r = k_4 b_{comp}, 0 < k_4 \leq k_3$$

donde,

$$b_{comp} = \sum_{i=1}^n a_b(j_i) \text{ y } z_{comp} = \sum_{i=1}^n a_z(j_i).$$

Para que el apoyo a una opción como resultado de la votación permanezca en el intervalo de 0 a 1 como en la política de consenso para los enunciados del foro de discusión, es necesario establecer una proporción de la siguiente manera

$$\frac{\sum_{i=1}^n a_b(j_i)}{b_j} = \frac{1}{apoyo_{oj}}.$$

8.10.2. MOMENTOS DE VOTACIÓN

Los momentos de votación se refieren a las actividades dentro del juego donde se puede llevar a cabo una votación.

Estas actividades se llevan a cabo dentro del foro de discusión, cómo y cuándo se indica a continuación:

1. Durante una discusión es posible que los jugadores deseen votar por un enunciado del foro de discusión. Para ello, cada jugador indica su grado de aprobación y de rechazo asignando la cantidad de bolas blancas y azules que desee. Se contabiliza la cantidad de bolas blancas y azules asignadas por los jugadores al enunciado; finalmente, se determina el estado de plausibilidad y la aceptabilidad del enunciado de acuerdo con las normas de votación y política de consenso para votación escogidas.
2. Otra alternativa de la votación es cuando los jugadores desean buscar la mejor solución a un problema. Los jugadores someten a votación un conjunto de

soluciones de un problema para seleccionar la mejor de acuerdo con el criterio del grupo. En este caso, los jugadores llevan al foro de discusión las soluciones (opciones); cada jugador indica su grado de aprobación y de rechazo asignando a cada opción la cantidad de bolas blancas y azules que desee. Luego, se contabiliza la cantidad de bolas blancas y azules asignadas por los jugadores a cada opción; y se determina el estado de plausibilidad y el apoyo a las soluciones de acuerdo con las normas de votación y política de consenso para votación escogidas.

En este caso cada solución (opción) se convierte en un hilo de discusión primario que además de someterse a votación puede someterse a discusión.

En los momentos de votación descritos el foro de discusión se convierte en una herramienta que soporta la votación.

8.11. EJEMPLO DE VOTACIÓN

El foro de discusión del Ejemplo 3 está compuesto por tres hilos de discusión primarios, los cuales representan soluciones de un mismo problema que los jugadores desean someter a votación.

Suponga que son tres jugadores. Considere que se le ha asignado a cada jugador un total de 10 bolas blancas y 10 bolas azules, y que la política de consenso establece las siguientes marcas y umbrales: $m_r = m_a = \frac{1}{2} * 30 = 15$ y $u_r = u_a = \frac{1}{4} * 30 = 7,5$.

El primer hilo recibió un total de 12 bolas blancas (3,3,6) y 7 bolas azules (3,2,2), el segundo hilo obtuvo un total de 9 bolas blancas (3,5,1) y 12 bolas azules (5,1,6) y el tercer hilo ganó 9 bolas blancas (4,2,3) y 11 bolas azules (2,7,2). Contiguo a

cada hilo de discusión se encuentra su estado de plausibilidad y su apoyo. Así el primer hilo de discusión primario tiene $e_p = \text{aceptable}$ y $\text{apoyo} = 12/30 = 0,40$. El segundo hilo de discusión primario tiene $e_p = \text{dudoso}$ y el último también. Estos resultados indican que el primer hilo de discusión primario es el que goza de la aceptación por parte de los jugadores.

<input type="checkbox"/> 0 <Nombre1, Hora-Fecha> -----	e_{10}^{11}	$e_p = \text{aceptable}$ $\text{apoyo} = 0,30$
<input type="checkbox"/> 0 <Nombre3, Hora-Fecha> -----	e_{10}^{21}	$e_p = \text{dudoso}$
<input type="checkbox"/> 0 <Nombre2, Hora-Fecha> -----	e_{10}^{31}	$e_p = \text{dudoso}$

Ejemplo 3: Ejemplo de un foro de discusión con votación.

8.12. MÉTODOS DE DISCUSIÓN

Los métodos de toma de decisiones usualmente constan de tres fases: la discusión del problema, la búsqueda de alternativas y la generación de opciones de solución.

Este esquema ha sido tomado en cuenta en esta investigación de la siguiente manera:

- **Discusión del problema:** El foro de discusión es la herramienta que se pone a disposición de los jugadores para que discutan sus puntos de vista; ya sean éstos hechos que ocurren en el juego o soluciones a los problemas.
- **Búsqueda de alternativas:** La búsqueda de alternativas también es una actividad que soporta el foro de discusión al permitir la discusión de temas abiertos.

- Generación de opciones de solución: Esta actividad no es propiamente apoyada por la herramienta de toma de decisiones. Estas alternativas se van generando en el desarrollo del juego y pueden ser corroboradas mediante una discusión en el foro. Estas opciones de solución pueden ser sometidas a votación.

Dentro de todo el proceso de toma de decisiones uno de los aspectos relevantes en esta tesis es la evaluación de las soluciones en el foro de discusión. Esta fase del proceso de toma de decisiones se realiza de la siguiente manera:

- Todas las soluciones se convierten en opiniones de hilo de discusión primarios. Es decir, por cada solución se genera automáticamente un hilo de discusión primario en el foro.
- Sobre cada solución se puede llevar a cabo una discusión en el foro. Esto es, se puede desarrollar el hilo de discusión secundario que nace en la solución.
- Sobre una solución se pueden realizar discusiones y votación, puesto que el foro permite efectuar votaciones sobre un enunciado.
- En el caso de una votación las bolas blancas de un jugador se manifiestan en el hilo de discusión como enunciados con naturaleza F , o sea evidencia a favor; y las bolas azules, como enunciados con naturaleza C , o sea evidencia en contra.