Instituto Tecnológico De Costa Rica Departamento De Computación Programa De Maestría

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

Implementación De Un Foro
De Discusión Jerárquico,
Basado En XML, Con Consideraciones
De Plausibilidad

Tabla De Contenidos

- 1. Especificación.
- 2. Arquitectura.
- 3. Conclusiones.
 - 4. Demostración.

Especificación

- ✓ El problema que se resolverá,
- ✓ Las funciones del sistema,
- ✓ El Diagrama de casos de uso general,
- ✓ Algunos Diagramas de colaboración,
- ✓ El Modelo Conceptual,
- ✓ El Diagrama de Paquetes.

Descripción general del problema.

- ✓ El foro de discusión jerárquico —en adelante, FDJ—, permite, a sus usuarios, debatir libremente 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.

05/09/2018 5 de 55

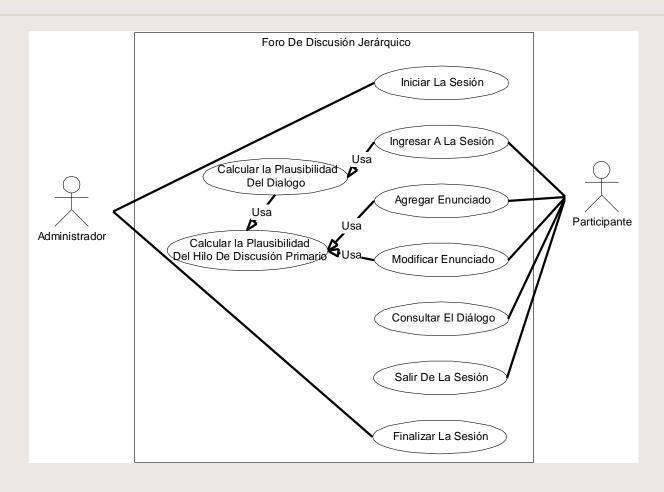
Funciones del Sistema

- ✓ Iniciar la sesión del Foro.
- ✓ Registrar el ingreso de cada usuario a la sesión.
- ✓ Registrar la salida de cada usuario de la sesión.
- ✓ Registrar los enunciados que se agregan.
- ✓ Registrar los enunciados que se modifican.
- ✓ Calcular el estado de Plausibilidad de los hilos de discusión primarios del diálogo, cada vez que algún usuario ingresa a la sesión.

(c.) Funciones del Sistema

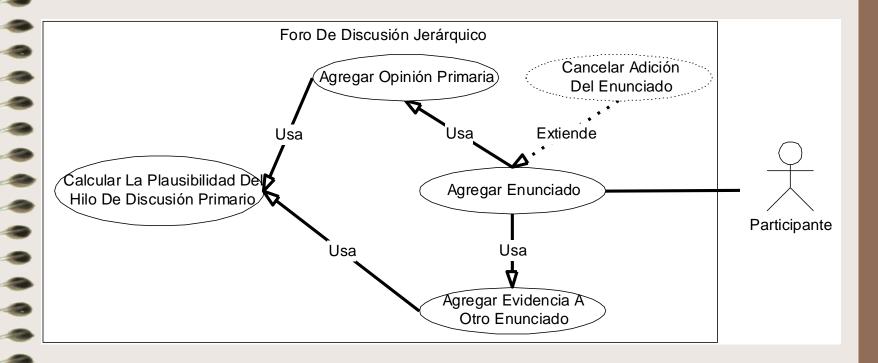
- ✓ Calcular el estado de Plausibilidad del hilo de discusión primario en que algún usuario agrega o modifica enunciados.
- ✓ Consultar el diálogo realizado, utilizando alguno de los modos de consulta válidos.
- ✓ Almacenar persistentemente el diálogo desarrollado en la sesión.
- ✓ Finalizar la sesión del Foro.

Casos de Uso Básicos del FDJ



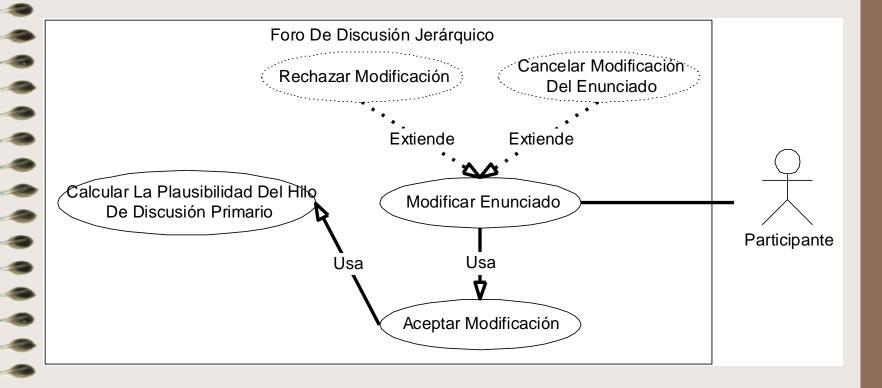
05/09/2018 8 de 55

Caso de Uso "Agregar Enunciado"



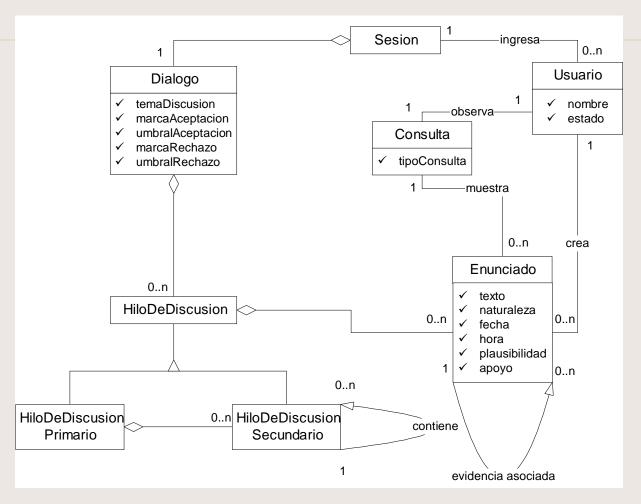
05/09/2018 9 de 55

Caso de Uso "Modificar Enunciado"



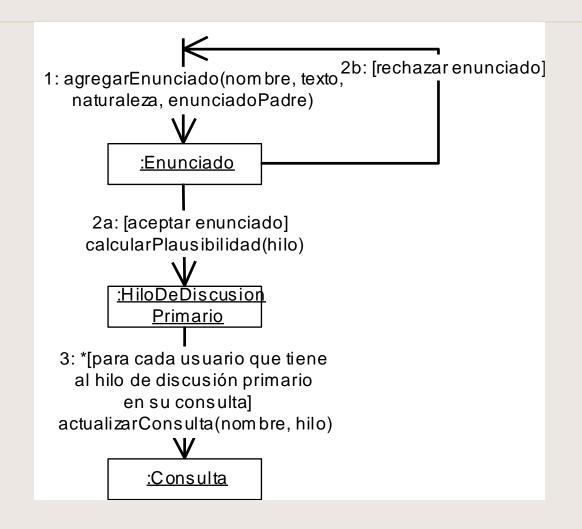
05/09/2018 10 de 55

Modelo Conceptual del FDJ

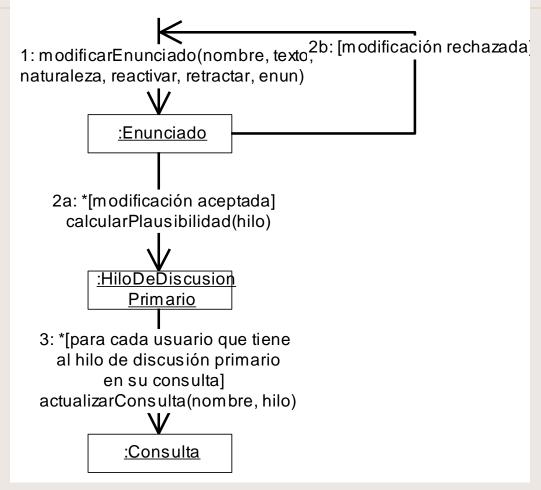


05/09/2018 11 de 55

Colaboración "Agregar Enunciado"



Colaboración "Modificar Enunciado"



05/09/2018 13 de 55

Adición de Métodos al Modelo Conceptual

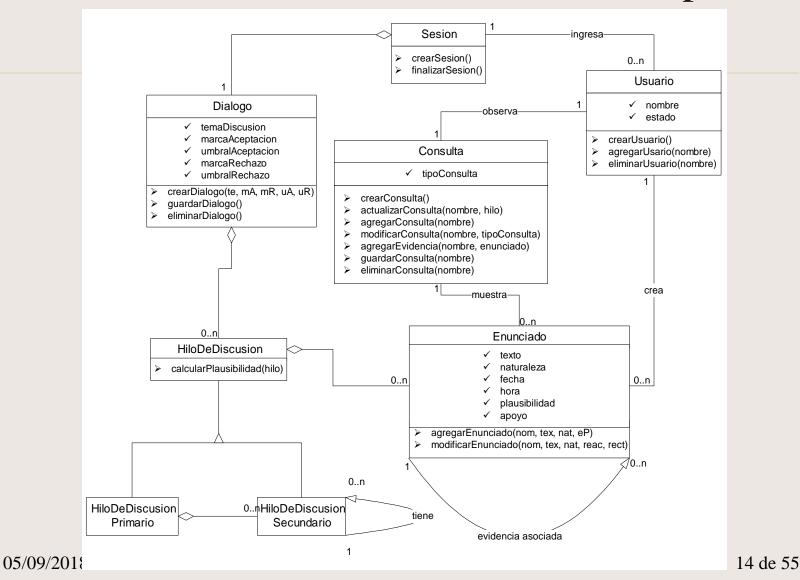
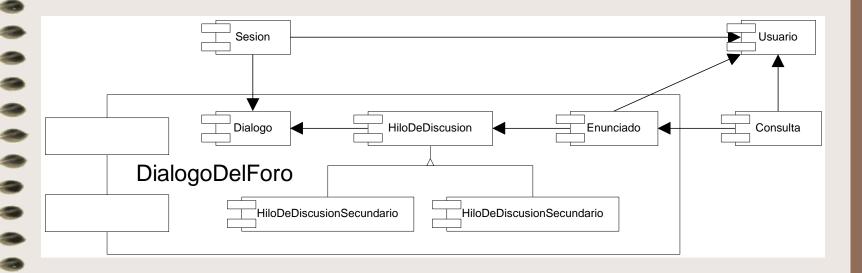


Diagrama de Paquetes del FDJ

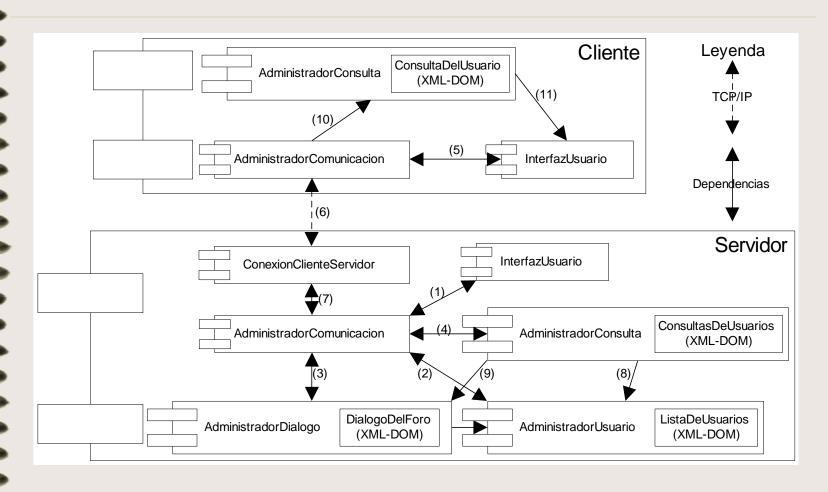


05/09/2018 15 de 55

Arquitectura y Diseño

- ✓ El diagrama de emplazamiento,
- ✓ El diagrama de clases estáticas,
- ✓ El modelo de identificación de enunciados,
- ✓ Las estructuras que datos XML que se utilizarán,
- ✓ La estructura de los mensajes que intercambiará el *FDJ*.

Diagrama de Emplazamiento del FDJ



05/09/2018 17 de 55

- ✓ Se considera el conjunto de hilos de discusión que integra el diálogo de una sesión del foro como un árbol n-ario.
- ✓ 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.

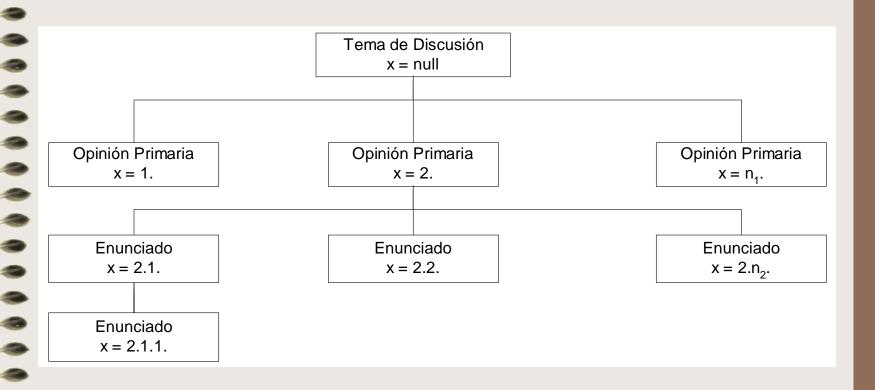
05/09/2018 18 de 55

- ✓ El "grado" de un nodo dentro de un árbol n-ario, se define como la cantidad de hijos que posee el nodo.
- ✓ Consideremos un nodo al que deseamos agregar un hijo. La llave que identifica al nodo está definida como:

05/09/2018 19 de 55

$$x = llave(n_p) \oplus \left\| grado(n_p) + 1 \right\| \oplus "."$$

$$x = i \oplus \left\| k + 1 \right\| \oplus "."$$



05/09/2018 21 de 55

$$2. = null \oplus ||Grad(null) + 1|| \oplus ||.|| = ||1 + 1|| \oplus ||.|| = 2.$$

$$2.n_2. = "2." \oplus \|Grad("2.") + 1\| \oplus "." = "2." \|(n_2 - 1) + 1\| \oplus "." = 2.n_2.$$

$$2.1.1. = "2.1." \oplus ||Grad("2.1.") + 1|| \oplus "." = "2.1." ||0 + 1|| \oplus "." = 2.1.1.$$

Estructura de los documentos XML utilizados

- ✓ La lista de usuarios de la sesión (*ListaDeUsuarios*),
- ✓ El Diálogo que desarrollan los usuarios (*DialogoDelForo*),
- ✓ Una Consulta general (ConsultasDeUsuarios),
- ✓ La consulta que realiza cada usuario sobre el diálogo (*ConsultaDelUsuario*),



DTD para la lista de usuarios

```
<?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)>
```

05/09/2018 24 de 55

DTD para el diálogo del foro

<?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, Evidencia Asociada. 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)>

DTD para la Consulta del Cliente

```
<?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,
                           Evidencia Asociada Vacia,
                           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 EvidenciaAsociadaVacia (#PCDATA)>
       <!ELEMENT NaturalezaAnterior (#PCDATA)>
```

DTD para la consulta general

05/09/2018 27 de 55

Estructura de paquetes intercambiados entre el cliente y el servidor.

✓ Problema: Se tiene la capacidad de enviar y recibir, de manera segura, exclusivamente cadenas de bytes.

05/09/2018 28 de 55

Formato de mensajes simples

{tipoMensaje} {cabecera} {datos}

- ✓ {*tipoMensaje*}: Carácter que identificará a la trama como un mensaje de un tipo previamente definido.
- √ {cabecera}: Un conjunto de caracteres numéricos y delimitados, permitirán identificar a un conjunto de campos, previamente definidos.
- ✓ {datos}: Cadena de caracteres que conforma el mensaje en sí.

Formato de Mensajes complejos

```
{tramaInicial}
{trama1}
{trama2}
\{trama(n)\}
{tramaFinal}
```

Mensajes Simples que envía el Servidor a los Clientes

Tema de Discusión	T/DATA
Inicio de Mensaje Complejo	I/InicioTrama
Reemplazar Enunciados	R/Reemplazar
Fin de Mensaje Complejo	I/FinTrama
Anexar Enunciados	A/Anexar
Sobrescribir Enunciados	S/Sobrescribir
Enunciado	E/1/2/3/4/5/6/7/8/9/10/11/DATA
Mensajes de la Línea de Estado	M/DATA
Tema de Discusión	T/DATA
Inicio de Mensaje Complejo	I/InicioTrama

05/09/2018 31 de 55

Mensajes Complejos que envía el Servidor a los Clientes

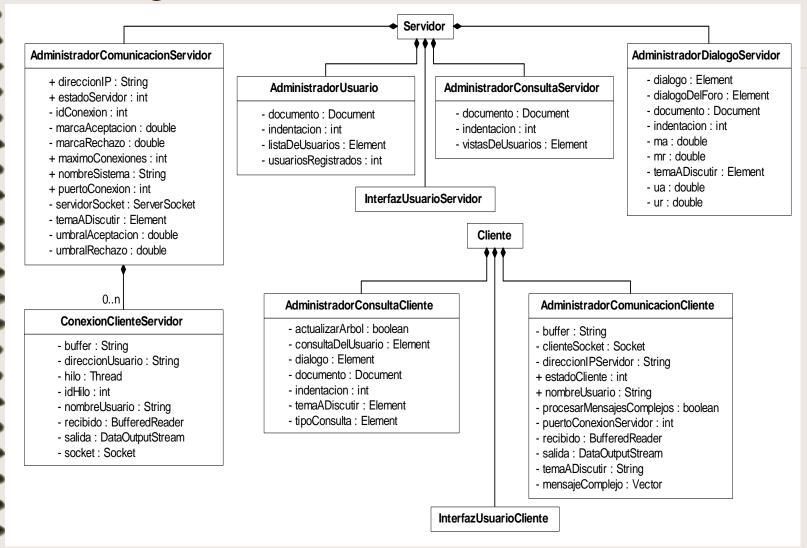
Reemplazar	Sobrescribir	Anexar
Enunciados	Enunciados	Enunciados
I/InicioTrama	I/InicioTrama	I/InicioTrama
R/Reemplazar	S/Sobrescribir	A/Anexar
E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA
E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA	E/1/2/3/4/5/6/7/8/9/10/11/DATA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

05/09/2018 32 de 55

Mensajes que envían los Clientes al Servidor

Mensaje	Cadena
Evidencia Asociada de un Enunciado	E/LLAVE
Cambiar de Consulta	V/DATA
Salir de la sesión	S/
Agregar Enunciado	A/1/2/3/4/DATA
Modificar Enunciado	M/1/2/3/4/5/6/DATA

Diagrama de Clases Estáticas del FDJ



05/09/2018 34 de 55

Clase AdministradorComunicacionServidor

AdministradorComunicacionServidor

- + Constructor AdministradorComunicacionServidor()
- + Constructor AdministradorComunicacionServidor(int puerto, String direccionIP)
- + boolean BajarServidor()
- + void cerrar_Server_sock()
- + String getDir(ServerSocket s)
- + String getName(ServerSocket s)
- + Bolean Levanta (int port)
- + Void mandaMensajeUno(String cad, int idUsuario)

05/09/2018

35 de 55

Clase ConexionClienteServidor

ConexionClienteServidor

- + void cerrarSocket()
- + void cerrarThread()
- + Constructor ConexionClienteServidor (int indice, Socket socket, String dir_conex, int act
- + void enviar_cad(String cad_usu)
- + void refresh(String S)

Clase Administrador Usuario

Administrador Usuario

- + Constructor Administrador Usuario (int tabulacion)
- + Constructor AdministradorUsuario()
- + Bolean agregarUsuario(String nombrePropuesto, String hostPropuesto)
- + Bolean borrarUsuario(String nombre)
- + Bolean esUsuarioActivo(String nombre)
- + Bolean esUsuarioRegistrado(String nombre)
- + Void guardarArchivo(String nombreArchivo)
- + Int obtenerCantidadDeUsuariosActivos()
- + Int obtenerCantidadDeUsuariosRegistrados()
- + Int obtenerIdUsuario(String nombreBuscado)
- + Jlist obtenerListaUsuariosSwing()

Clase Administrador Dialogo

AdministradorDialogoServidor

- + String actualizarDialogoTrama(String trama)
- + Constructor Administrador Dialogo (String tema, double ma, double mr, double ua, double ur, int tabulacion)
- + Constructor Administrador Dialogo()
- + int agregarEnunciado(String llavePadre, String textoEnunciado, String nombreUsuario, String naturalezaEnunciado)
- + void calcularPlausibilidadHilo(String llave)
- + boolean esEnunciadoRegistrado(String llaveEnunciado)
- + void guardarArchivo(String nombreArchivo)
- + int modificarEnunciado(String llaveEnunciado, String textoEnunciado, String nombreUsuario, String naturalezaEnunciado, boolean retractar, boolean reactivar)
- + int obtenerCantidadDeHijos(Node enunciado)
- + Vector obtenerEnunciadosAlterados(String llaveEnunciado)
- + String obtenerEvaluarAgregarEnunciado(int codigo)
- + String obtenerEvaluarModificarEnunciado(int codigo)
- + Element obtenerEvidenciaAsociada(Element enunciado)
- + String obtenerLlaveEvidenciaAgregada(String llavePadre)
- + String obtenerNodoBaseEnunciado(String llaveEnunciado)

Clase AdministradorConsultaServidor

AdministradorConsultaServidor

- + Constructor AdministradorConsultaServidor()
- + Constructor AdministradorConsultaServidor(int tabulacion)
- + boolean agregarConsultaDeUsuario(int idUsuario)
- void agregarEnunciadoAConsulta(int idUsuario, String llaveEnunciado)
- + void borrarConsultaDeUsuario(int idusuario)
- + boolean esConsultaDeUsuarioValida(int idusuario)
- + boolean esEnunciadoIncluido(int idUsuario, String llave)
- + void guardarArchivo(String nombreArchivo)
- + void modificarAtributoConsultaActualizada(int idUsuario, boolean estado)
- + void modificarConsultaUsuario(int idUsuario, String tipoConsulta
- + String obtenerTipoConsulta(int idUsuario)
- + String obtenerTramaEnunciado(String llaveEnunciado)
- + Vector obtenerVectorTramasClientes(Vector vectorEnunciados, int idUsuario, String llave)

Clase AdministradorComunicacionCliente

AdministradorComunicacionCliente

- + Constructor AdministradorComunicacionCliente()
- + Constructor AdministradorComunicacionCliente(String dir_server, int port_server, String nombre)
- void cerrar_socket()
- void conectar(String dir_ser, int port)
- + void desconectar()
- + boolean enviar(String cad)
- void refresh(String S)
- + JScrollPane repintarArbol(JScrollPane panel)

Clase AdministradorConsultaCliente

AdministradorConsultaCliente

- + Void actualizarDialogoMensaje(Vector vectorMensajes)
- + Constructor AdministradorConsultaCliente()
- + Constructor AdministradorConsultaCliente(String tema, int indentacion)
- + Int agregarEnunciado(String llavePadre, String naturalezaEnunciado, String textoEnunciado, String nombreUsuario, String fecha, String hora, String plausibilidad, String color, String apoyo, String evidenciaVacia, String naturalezaAnterior)
- + Void asignarTipoConsulta(String tipoConsulta)
- + Bolean esEnunciadoRegistrado(String Llave)
- + Void guardarArchivo(String nombreArchivo)
- + Int modificarEnunciado(String llaveEnunciado, String naturalezaEnunciado, String textoEnunciado, String nombreUsuario, String fechaEnunciado, String horaEnunciado, String plausibilidadEnunciado, String colorEnunciado, String apoyoEnunciado, String evidenciaVacia, String naturalezaAnterior)
- + Jtree obtenerArbolJTree(JTree arbol)
- + String obtenerllavePadre(String llaveHijo)
- + String obtenerMensajeAgregarEnunciado(String llavePadre, String textoEnunciado String nombreUsuario, String naturalezaEnunciado)
- + String obtenerMensajeModificarEnunciado(String llaveEnunciado, String textoEnunciado, String nombreUsuario, String naturalezaEnunciado, boolean retractar, boolean reactivar)
- + Node obtenerNodoBaseEnunciado(String llaveEnunciado)
- + String obtenerTipoConsulta()
- + Element obtenerEvidenciaAsociada(Element enunciado)

Validación del sistema contra su especificación

- ✓ Se realizó esencialmente, sobre la clase *AdministradorDialogo*.
- ✓ Metodología:
- ✓ Se plantearon algunos ejemplos.
- ✓ Para cada ejemplo se calculó, manualmente, el apoyo y el estado de plausibilidad
- ✓ Se creó una aplicación Java, sin interfaz que ejecuta instrucciones "directamente".

(c.) Validación del sistema contra su especificación

- ✓ Se comparó, manualmente, los datos contenidos en el archivo XML generado por la aplicación de prueba, contra los datos obtenidos manualmente.
- ✓ En todos los casos, los resultados obtenidos con el software coincidieron razonablemente bien con los resultados obtenidos manualmente.

Productos obtenidos en la investigación

- ✓ La especificación del sistema modelada con *UML*.
- ✓ La arquitectura del sistema modelada con *UML*.
- ✓ Un modelo alterno de identificación de enunciados, dentro del *FDJ*.
- ✓ El modelo de almacenamiento de información del *FDJ* basado en XML.
- ✓ Un modelo de comunicación de mensajes bajo TCP/IP, basado en caracteres.
- ✓ El código ejecutable del sistema con su manual de usuario.

Conclusiones

- ✓ Se implementó, un prototipo funcional del FDJ propuesto por [Núñez], que materializa las características más esenciales de esta formalización.
- ✓ Representa la implementación de uno de los modelos de votación propuestos por [Olivares].
- ✓ Se modeló utilizando el metalenguaje *UML*, como herramienta de especificación y diseño.
- ✓ Utiliza el metalenguaje *XML* como estructura principal de almacenamiento de información yolátil y persistente.

(c.) Conclusiones

- ✓ Se obtiene un prototipo de sistema razonablemente funcional, que constituye un medio alterno a la realización de reuniones físicas, permite que los usuarios voten implícitamente al dialogar.
- ✓ Este sistema posibilita la realización de reuniones electrónicas y de procesos de toma de decisiones en grupo, desde una perspectiva novedosa: la jerarquización del diálogo.

(c.) Conclusiones

- ✓ El *FDJ* permite controlar el problema representado por la distancia entre participantes, al ofrecer un "salón virtual".
- ✓ Facilita la compresión del diálogo entre participantes, al jerarquizarlo.
- ✓ Cuantifica la aceptabilidad de cada enunciado que se expresa en el diálogo, con base a la Teoría de la Plausibilidad.

05/09/2018 47 de 55

(c.) Conclusiones

✓ Adicionalmente, el prototipo del *FDJ* puede ser utilizado como una herramienta de investigación teórica.

05/09/2018 48 de 55

Investigaciones derivadas

- ✓ Incluir un módulo de votaciones fundamentado en la Teoría de Plausibilidad,
- ✓ Incluir el intercambio de datos multimedia,
- ✓ Ampliar los tipos de consultas disponibles,
- ✓ Ampliar la cobertura de las operaciones deshacer / rehacer del *FDJ*,

(c.) Investigaciones derivadas

- ✓ Evaluar la efectividad del prototipo al ser utilizado por los grupos de usuarios potenciales,
- ✓ Evaluar el desempeño y eficiencia computacional del prototipo y de ser posible mejorarlo en estos aspectos,
- ✓ Mejorar el interfaz de la aplicación, para que tenga el acabado de una aplicación más depurada,

05/09/2018

50 de 55

(c.) Investigaciones derivadas

- ✓ Mejorar el modelo de colaboración utilizado en el prototipo:
 - ➤ Otorgar al administrador del servidor la posibilidad de administrar a los usuarios,
 - ➤ Incluir la posibilidad de que los usuarios del sistema incluyan enunciados bajo la figura del anonimato.

(c.) Investigaciones derivadas

- ✓ Ampliar el modelo de comunicación de mensajes, para que incluya características de seguridad:
 - Encriptación de las tramas,
 - ➤ Inclusión de firmas digitales para los clientes y el servidor.

05/09/2018 52 de 55

Referencias Bibliográficas

- ✓ LARMAN, C. UML y patrones: Introducción al análisis y diseño orientado a objetos. Prentice-Hall. 1999.
- ✓ 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.

(c.) Referencias Bibliográficas

- ✓ 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.
- ✓ **POSADAS, M.** *El Modelo de objetos de documento (DOM)*. Grupo EIDOS, 4 de diciembre del 2000. (http://www.eidos.es/).

05/09/2018 54 de 55

Gracias a Todos

Por Su Atención....