

**INTELIGENCIA ARTIFICIAL EN LA GENERACIÓN DE ESTRATEGIAS DE
APRENDIZAJE PARA UN SISTEMA TUTOR INTELIGENTE**



**ASTRID JOHANA VALENCIA
CESAR AUGUSTO ESCOBAR**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE SISTEMAS
POPAYÁN
2006**

TABLA DE CONTENIDO

INTRODUCCIÓN.....	9
CAPITULO 1. MARCO TEÓRICO	13
1.1. Los Modelos Pedagógicos y Las estrategias de Aprendizaje.....	13
1.1.1. La Pedagogía Tradicional.....	13
1.1.2. El Modelo Instruccional: Respuestas Definidas Previamente.....	14
1.1.3. La Nueva Escuela.....	14
1.1.4. El Modelo Activista – Se aprende haciendo.....	15
1.1.5. Los Modelos Pedagógicos Contemporáneos	15
1.1.6. La Pedagogía Conceptual	16
1.1.7. El Currículo y sus preguntas.....	17
1.1.8. Estrategias de Aprendizaje	18
1.2. El Aprendizaje y los Estilos de Aprendizaje.....	21
1.2.1. El Aprendizaje	21
1.2.2. Tipos de Aprendizaje	23
1.2.3. Estilos De Aprendizaje.....	24
1.3. Educación en Línea.....	27
1.3.1. Concepto de Educación	28
1.3.2. Concepto de Educación en Línea.....	29
1.3.3. El Método Educativo, Los Medios Y La Tecnología.....	31
1.3.4. Modelos para el diseño instruccional de cursos en línea.....	32
1.4. El Estándar SCORM (Sharable Content Object Reference Model) 33	
1.4.1. Importancia de SCORM en este proyecto.....	34
1.4.2. Componentes del Estándar SCORM.....	34
1.5. Los Sistemas Tutores Inteligentes	37
1.6. Enfoques De Inteligencia Artificial.....	40
1.6.1. Representación del conocimiento.....	40
1.6.2. Técnicas de Representación del Conocimiento	41
1.7. Inteligencia Artificial en el Proyecto	43
1.7.1. Modelos Simbólicos.....	43
1.7.2. Modelos Conexionistas.....	44
1.7.3. Los Sistemas Expertos	45
1.8. Conceptos de IA utilizados en el Modelo de Toma de decisiones.	46

1.8.1. Introducción	46
1.8.2. Conocimiento Heredable y Sistema de Ranuras	47
1.8.3. Conocimiento Relacional.....	48
1.8.4. Mecanismo de Inferencia	50
1.8.5. Árboles de Decisión.....	50
1.8.6. Reglas de División	52
1.8.7. Ganancia de Información - Entropía.....	52
1.8.8. Algoritmo ID3	53
CAPITULO 2. EL MODELO DE TOMA DE DECISIONES DEL MODULO INTELIGENTE.....	56
2.1. Generalidades.....	56
2.2. Mecanismo de Representación de conocimiento	56
2.3. Mecanismo de Inferencia.....	58
2.3.1. El Árbol de Decisión	59
2.3.2. El algoritmo Deliberar.....	59
2.4. Una Ilustración del modelo de toma de decisiones	60
2.5. Modelo Final vs Tecnologías existentes en el ambito de la IA y los Sistemas de Toma de decisiones - DSS.....	65
2.5.1. Bodegas de Datos - DataWarehouse	65
2.5.2. Sistemas de Razonamiento Basado en Casos - CBR.....	67
CAPITULO 3. EL MÓDULO INTELIGENTE DEL STI.	68
3.1. Entradas	68
3.1.1. Conjunto de Atributos y sus valores.....	69
3.1.2. Conjunto de ejemplos o experiencias	70
3.1.3. Modelo de Datos del Modulo Inteligente.....	72
3.2. Procesos	73
3.2.1. Proceso de construcción del árbol de decisión	73
3.2.2. Proceso de Evaluación y Estructuración.....	74
3.3. Salidas	75
CAPITULO 4. ETAPA DE ANÁLISIS	77
4.1. Requerimientos.....	77
4.2. Descripción del Sistema	78
4.3. Diagrama General de Casos de Uso	80
4.3.1. Caso de Uso: Gestionar Ítems Contenido.....	81
4.3.2. Caso de Uso: Gestionar Contenido Programático.....	81
4.3.3. Caso de Uso: Gestionar Agregaciones	82
4.3.4. Caso de Uso: Gestionar Actividades	82
4.3.5. Caso de Uso: Crear Estructura de Contenido.....	83
4.4. Modelo Conceptual Preliminar	83
4.5. Modelo Lógico de Datos.....	86

4.5.1. Modelo de Datos: Gestionar Ítems Contenido	86
4.5.2. Modelo de Datos: Gestionar Contenido Programático.....	87
4.5.3. Modelo de Datos: Gestionar Agregaciones	88
4.5.4. Modelo de Datos: Gestionar Actividades	88
4.5.5. Modelo de Datos: Crear Estructura de Contenido.....	89
CAPITULO 5. ETAPA DE DISEÑO	91
5.1. Descripción de la Arquitectura.....	91
5.1.1. En el Cliente.....	93
5.1.2. En el servidor	96
5.1.3. Motivación para la definición de la Arquitectura	97
5.1.4. Los Patrones Utilizados	97
5.2. Casos de Uso Reales.....	98
5.2.1. Casos De Uso Real: Gestión Ítems Contenido	99
5.2.2. Casos De Uso Real: Gestión Contenidos Programáticos	100
5.2.3. Casos De Uso Real: Gestión Agregaciones.....	101
5.2.4. Casos De Uso Real: Gestión Actividades	103
5.2.5. Caso De Uso Real: Crear Estructura de Contenidos	104
CAPITULO 6. RESULTADOS OBTENIDOS	106
6.1. Objetivos Planteados	106
6.2. Tabla de Resultados	108
6.3. Lineamientos de Conformación e Indicadores	112
6.4. Productos.....	112
CAPITULO 7. CONCLUSIONES Y RECOMENDACIONES	114
7.1. CONCLUSIONES.....	114
7.2. RECOMENDACIONES	115
CAPITULO 8. REFERENCIAS BIBLIOGRAFICAS.....	116
ANEXOS.....	121

TABLA DE FIGURAS

Figura 1. Librería de SCORM (Tomado de SCORM sequencing and Navigation Version 3.1.3).....	34
Figura 2. Conceptos usados del estandar SCORM	36
Figura 3. Estructura de un Sistema Tutor Inteligente	39
Figura 4. Diagrama de Bloques de una aplicación Inteligente	47
Figura 5. Representación de Conocimiento Relacional	49
Figura 6. Modelo general de toma de decisiones	56
Figura 7. Representación del perfil del estudiante.....	57
Figura 8. Mecanismo de Inferencia	59
Figura 9. Ejemplos de Atributos.....	60
Figura 10. Conjunto de experiencias	61
Figura 11. Árbol generado después de la primera recursión	62
Figura 12. Modelo de creación de un árbol de decisión	64
Figura 13. Modulo Inteligente.....	68
Figura 14. Modelo de Datos del Modulo Inteligente.....	72
Figura 15. Estructuras del Árbol de Decisión.....	74
Figura 16. Modelo de Datos de las Salidas.....	76
Figura 17. Modelo Gestión Ítems Contenido.....	86
Figura 18. Modelo Gestión Contenido Programático	87
Figura 19. Modelo Gestión Agregaciones	88
Figura 20. Modelo Gestión Actividad	89
Figura 21. Modelo Crear Estructura de Contenido	90
Figura 22. Arquitectura General del Sistema	92
Figura 23. Diagrama Caso de Uso Real Gestión Ítems	99
Figura 24. Diagrama Caso de Uso Real Gestión Contenidos	101
Figura 25. Diagrama Caso de Uso Real Gestión Agregaciones	102
Figura 26. Diagrama Caso de Uso Real Gestión Actividades	103
Figura 27. Diseño Instruccional Lineal	128
Figura 28. Diseño instruccional Ramificado	129
Figura 29. Diseño Instruccional con Hipercontenidos.....	130
Figura 30. Diseño Instruccional Directo del Estudiante	131
Figura 31. Ejemplos de Recursos (Asset's)	132
Figura 32. Organización de Contenidos.....	133
Figura 33. Ejemplo de un PIF	134

Figura 34. Estructura de contenidos Vs. Árbol de actividades (tomado de SCORM Sequencing and Navigation Version 1.3.1)	137
Figura 35. Pantalla Inicial con las Opciones	206
Figura 36. Gestión de Atributos.....	206
Figura 37. Árbol de decisión generado y su conjunto de experiencias – Caso 1.....	207
Figura 38. Árbol de decisión generado y su conjunto de experiencias – Caso 2.....	207
Figura 39. Pantalla de diagnostico	208
Figura 40. Pantalla de generación de resultados	208

INDICE DE TABLAS

Tabla 1.	Conjunto de experiencias inicial	62
Tabla 2.	Modificaciones al conjunto de experiencias después de la primera recursión	63
Tabla 3.	Conjunto de experiencias para iniciar la segunda recursión	63
Tabla 4.	Atributos Estudiante	69
Tabla 5.	Atributos Sistema	70
Tabla 6.	Ejemplo de una experiencia	71
Tabla 7.	Clasificación de las funciones del sistema	78
Tabla 8.	Funciones Básicas del Sistema.....	78
Tabla 9.	Funciones Gestión Agregación	79
Tabla 10.	Funciones Gestión Estructura de Contenidos	79
Tabla 11.	Funciones Gestión Contenido Programático.....	79
Tabla 12.	Funciones Gestión Ítems Contenido.....	79
Tabla 13.	Funciones Gestión Actividad	79
Tabla 14.	Atributos del Sistema	80
Tabla 15.	Diagrama de Casos de Uso.....	81
Tabla 16.	Modelo Conceptual Preliminar	85
Tabla 17.	Caso de Uso Real Crear Estructura de Contenido ..	105
Tabla 18.	Clasificación Estilos de Aprendizaje CHAEA	121
Tabla 19.	Clasificación Organización de la Información	124
Tabla 20.	Clasificación Inteligencias Múltiples	125
Tabla 21.	Concepción según el sistema de representación ...	126
Tabla 22.	Concepción según la categoría bipolar.....	127
Tabla 23.	Caso de Uso Insertar Ítems – Alto Nivel	139
Tabla 24.	Caso de Uso Consultar Ítems – Alto Nivel	139
Tabla 25.	Caso de Uso Modificar Ítems – Alto Nivel.....	139
Tabla 26.	Caso de Uso Eliminar Ítem – Alto Nivel.....	140
Tabla 27.	Caso de Uso Definir Contenido – Alto Nivel.....	140
Tabla 28.	Caso de Uso Consultar Contenido – Alto Nivel	141
Tabla 29.	Caso de Uso Activar Contenido – Alto Nivel	141
Tabla 30.	Caso de Uso Modificar Contenido – Alto Nivel	141
Tabla 31.	Caso de Uso Insertar Agregación – Alto Nivel.....	142
Tabla 32.	Caso de Uso Consultar Agregación – Alto Nivel.....	142
Tabla 33.	Caso de Uso Modificar Agregación – Alto Nivel	142
Tabla 34.	Caso de Uso Eliminar Agregación – Alto Nivel.....	143

Tabla 35.	Caso de Uso Insertar Actividad – Alto Nivel.....	143
Tabla 36.	Caso de Uso Consultar Actividad – Alto Nivel.....	143
Tabla 37.	Caso de Uso Modificar Actividad – Alto Nivel	144
Tabla 38.	Caso de Uso Eliminar Actividad – Alto Nivel.....	144
Tabla 39.	Caso de Uso CrearEstructuraContenido - Alto Nivel	144
Tabla 40.	Caso de Uso Insertar Ítems – Expandido	145
Tabla 41.	Caso de Uso Consultar Ítem - Expandido	146
Tabla 42.	Caso de Uso Modificar Ítems – Expandido	146
Tabla 43.	Caso de Uso Eliminar Ítem – Expandido.....	147
Tabla 44.	Caso de Uso Definir Contenido – Expandido.....	148
Tabla 45.	Caso de Uso Consultar Contenido – Expandido	149
Tabla 46.	Caso de Uso Activar Contenido – Expandido	149
Tabla 47.	Caso de Uso Modificar Contenido – Expandido	150
Tabla 48.	Caso de uso Insertar Agregación - Expandido	151
Tabla 49.	Caso de Uso Consultar Agregación – Expandido ...	152
Tabla 50.	Caso de Uso Modificar Agregación – Expandido	152
Tabla 51.	Caso de Uso Eliminar Agregación – Expandido.....	153
Tabla 52.	Caso de uso Insertar Actividad - Expandido.....	154
Tabla 53.	Caso de Uso Consultar Actividad – Expandido.....	154
Tabla 54.	Caso de Uso Modificar Actividad – Expandido	155
Tabla 55.	Caso de Uso Eliminar Actividad – Expandido.....	156
Tabla 56.	Caso de Uso CrearEstructuraContenido – Expandido	157
Tabla 57.	Caso de Uso Real Insertar Ítems	158
Tabla 58.	Caso de Uso Real Consultar Ítem.....	159
Tabla 59.	Caso de Uso Real Modificar Ítem	160
Tabla 60.	Caso de Uso Real Eliminar Ítem.....	161
Tabla 61.	Caso de Uso Real Definir Contenido	163
Tabla 62.	Caso de Uso Real Consultar Contenido.....	164
Tabla 63.	Caso de Uso Real Activar Contenido.....	165
Tabla 64.	Caso de Uso Real Modificar Contenido	166
Tabla 65.	Caso de uso Real Insertar Agregación	167
Tabla 66.	Caso de Uso Real Consultar Agregación	168
Tabla 67.	Caso de Uso Real Modificar Agregación	169
Tabla 68.	Caso de Uso Real Eliminar Agregación	170
Tabla 69.	Caso de uso Real Insertar Actividad	171
Tabla 70.	Caso de Uso Real Consultar Actividad	172
Tabla 71.	Caso de Uso Real Modificar Actividad	173
Tabla 72.	Caso de Uso Real Eliminar Actividad	174
Tabla 73.	Diagrama de Secuencia Insertar Ítem	175
Tabla 74.	Diagrama de Secuencia Consultar Ítem	176

Tabla 75.	Diagrama de Secuencia Modificar Ítem.....	177
Tabla 76.	Diagrama de Secuencia Eliminar Ítem	178
Tabla 77.	Diagrama de Secuencia Definir Contenido	179
Tabla 78.	Diagrama de Secuencia Consultar Contenido	180
Tabla 79.	Diagrama de Secuencia Activar Contenido.....	181
Tabla 80.	Diagrama de Secuencia Modificar Contenido	182
Tabla 81.	Diagrama de Secuencia Insertar Agregación	183
Tabla 82.	Diagrama de Secuencia Consultar Agregación	184
Tabla 83.	Diagrama de Secuencia Modificar Agregación.....	185
Tabla 84.	Diagrama de Secuencia Eliminar Agregación	186
Tabla 85.	Diagrama de Secuencia Insertar Actividad	187
Tabla 86.	Diagrama de Secuencia Consultar Actividad	188
Tabla 87.	Diagrama de Secuencia Modificar Actividad.....	189
Tabla 88.	Diagrama de Secuencia Eliminar Actividad	190
Tabla 89.	Diagrama de Secuencia Crear Estructura	192

INTRODUCCIÓN

En países poco desarrollados la educación en general (primaria, secundaria y universitaria) no está garantizada para todos los sectores de la sociedad, lo que genera problemas sociales que afectan en gran medida dicho desarrollo. Por tal razón, nacen propuestas que van en pro de aumentar la cobertura de la educación sin desmejorar su calidad. Este es el caso de la educación virtual (o también llamada e-learning) y más específicamente las universidades virtuales que buscan complementar la educación presencial, agregándole nuevas características que esta última no está en capacidad de soportar, sin entrar a competir con ella.

Sin embargo siempre que se habla de Educación en Línea (e-learning) se piensa en todas las barreras mentales que ese nuevo concepto debe superar; pero pocas veces se habla de los nuevos adelantos que a nivel pedagógico se pueden lograr a través del uso de las tecnologías de la información. En este proyecto, se plantea un proceso de personalización del aprendizaje a las características de los estudiantes de los cursos en línea por medio del eficiente uso de herramientas informáticas y pedagógicas que acerquen a la educación en línea a niveles que la educación presencial difícilmente alcanzaría.

Ahora, los modelos de educación virtual carecen de interacción directa Estudiante – Profesor o esta se limita a unos pocos encuentros, lo que debilita el papel que desempeña dicha relación en el proceso de aprendizaje, fundamentalmente porque cualquiera que sea el tipo de educación, ella misma es un acto comunicativo en el cual se intercambian conocimientos y experiencias, permitiendo a cada interlocutor redefinir sus redes conceptuales constantemente. Es por esta razón que en ambientes de Educación en Línea, la figura del tutor toma especial interés dentro del contexto del proceso de aprendizaje, desvirtuando la idea que dicha relación no es necesaria en estos ambientes educativos. Entre los principales roles del tutor en los ambientes de educación a distancia se encuentran[1]: Formador, Profesor, Investigador, Tecnólogo Educativo, Tutor, Recurso Didáctico y Monitor.

Estas tareas resultarían casi imposibles de realizar por una sola persona, que además de tener estos diferentes roles, puede llegar a tener muchos más estudiantes de los que tendría en un ambiente presencial. Esto hace que su labor requiera de la ayuda de herramientas tecnológicas que le apoyen en procesos tutoriales con los estudiantes en los momentos en los que por diversas dificultades no pueda comunicarse directamente con los estudiantes para resolver sus dudas o reorientar su trabajo, en conclusión los Sistemas Tutores Inteligentes son un complemento para la actividad enseñanza/aprendizaje, particularmente en

el aspecto que intenta conjugar el estilo de enseñanza del profesor y las preferencias de aprendizaje de los estudiantes.¹

Los tutores inteligentes se han convertido en el caballo de batalla de los cursos masivos, funcionando como un puente entre el diseño instruccional realizado por un alejado profesor y el alumno que necesita de dicho profesor para su proceso de aprendizaje[2]. Por lo tanto, los tutores inteligentes se muestran como una solución a la falta de interacción entre los estudiantes y la figura del profesor. Los Tutores Inteligentes en estos ambientes deben tener ciertas características especiales para lograr sus objetivos, una de las más deseables es el soporte a las diferentes estrategias educativas. Esto se da gracias a que la nueva educación debe dejar de someter al estudiante a un modelo que está predefinido solo por el tutor y empezar a cambiar el orden cuando se pretende adaptar el modelo educativo a las características propias de cada estudiante.

Numerosos trabajos se han realizado en esta área en distintos países. En su mayoría resultan ser países en donde el e-learning es un medio validado y aceptado socialmente como forma de educación y que por tanto solo buscan una mayor calidad en un proceso que ya vienen usando hace algún tiempo. En países menos desarrollados los trabajos en estas áreas son escasos, lo que otorga un valor importante a este proyecto. Además, los esfuerzos hechos en el área de personalización de la educación pocas veces incluyen tecnologías tan importantes como la Inteligencia Artificial. Aunque se persiga el mismo objetivo, siempre se busca una dependencia de parte del sistema hacia el Tutor humano, sin tomar en cuenta que es posible dar al Tutor Software cierta autonomía que le permita decidir sobre una estrategia de aprendizaje apropiada, soportada en el análisis del perfil del estudiante que se genera a partir de su interacción con el sistema.

El presente proyecto hace parte de un macroproyecto denominado Unicauca Virtual Fase II el cual es auspiciado por la Universidad del Cauca y COLCIENCIAS, Los objetivos del presente proyecto son:

Objetivo General:

Integrar técnicas de Inteligencia Artificial, Conceptos de Estilos de aprendizaje y Sistemas Tutores Inteligentes de forma tal que permita generar un módulo inteligente apoyado en computador para la gestión de estrategias de Enseñanza Aprendizaje en un ambiente de educación en línea.

Objetivos Específicos:

¹ Estos sistemas computacionales se caracterizan por poseer el material instruccional separado de la especificación de cómo y cuando puede ser usado y re-usado de manera múltiple. Tomado de <http://cs.uns.edu.ar/jeitics2005/Trabajos/pdf/22.pdf>

- Desarrollar un modulo inteligente que integre técnicas de Inteligencia Artificial con conceptos de educación virtual y Estrategias de Aprendizaje para un Sistema Tutor Inteligente.
- Generar estrategias de enseñanza y aprendizaje a partir de reglas de conocimiento para la presentación personalizada de los contenidos en un ambiente de educación en línea.
- Utilizar las técnicas de Inteligencia Artificial para representar el conocimiento de los estudiantes de forma tal que permita el proceso de toma de decisiones que involucran dicho conocimiento.
- Desarrollar un prototipo que permita realizar la validación de los resultados del proyecto.
- Integrar la herramienta Software a la fase II del macroproyecto Unicauca Virtual.

En resumen, el presente proyecto pretende desarrollar un sistema software para la educación en línea, que tenga en cuenta el uso de sistemas tutoriales inteligentes para la solución de problemas instruccionales y pedagógicos. Centrado específicamente en desarrollar el módulo tutor con inteligencia artificial, el cual retome los datos obtenidos del monitoreo del estudiante y desarrolle estrategias instruccionales y de aprendizaje a cada estudiante, teniendo en cuenta su estilo de aprendizaje.

Todo el desarrollo del proyecto mencionado, se verá evidenciado a lo largo de esta monografía de la siguiente manera:

En el Capítulo 1 se mostrará la fundamentación teórica del proyecto en el que se hará una breve introducción de cada uno de los temas tenidos en cuenta para su desarrollo; los temas a tratar son: Modelos Pedagógicos, El Aprendizaje, Los Estilos de Aprendizaje, La educación en Línea, el estándar SCORM y el área de la Inteligencia Artificial (IA).

En el Capítulo 2 se hará una profundización del estudio en el área de la IA, se mencionarán los aspectos tenidos en cuenta para el desarrollo del modulo inteligente del tutor y la fundamentación teórica del desarrollo hecho en el modulo de toma de decisiones del Sistema Tutor.

En el Capítulo 3 se mostrará la aplicación de toda la teoría estudiada en el capítulo 2 y se evidenciará el trabajo hecho para la construcción del modulo tutor inteligente usando Inteligencia Artificial.

En el Capítulo 4 y 5 se mostrarán los artefactos del análisis y el diseño hechos en el proceso de desarrollo del presente proyecto, así como la arquitectura establecida para la implementación del prototipo.

Para finalizar en el capítulo 6 se especifican los resultados obtenidos al finalizar el trabajo de investigación y desarrollo del presente proyecto, teniendo en cuenta los objetivos planteados en el anteproyecto, los indicadores de evaluación de resultados y las herramientas que evidencian dichos resultados, además de las observaciones pertinentes y las conclusiones que se presentarán en el capítulo 7.

CAPITULO 1. MARCO TEÓRICO

1.1. Los Modelos Pedagógicos y Las estrategias de Aprendizaje

Cuando se piensa en diseñar un programa de educación ya sea presencial o no presencial, entre las que se encuentra la educación en línea, es importante definir primero el enfoque o modelo pedagógico que servirá de marco de referencia para el diseño de la estrategia con la que se abordará el curso.[3] Todo esto se consigue respondiendo a la necesidad de explicar el tipo de hombre y de sociedad que se desea formar.

El modelo de enseñanza y aprendizaje de los sistemas de educación en línea debe tener unas bases conceptuales fuertes que permitan definir diferentes condiciones bajo las cuales los estudiantes aprenderán, así como las condiciones bajo las cuales los maestros intervendrán en ese proceso de aprendizaje de los estudiantes, teniendo en cuenta los diversos componentes curriculares, los propósitos, los contenidos y la secuenciación (Ver apartado 1.1.7 El currículo). Un modelo pedagógico brinda los lineamientos generales básicos para establecer los componentes del currículo además de las estrategias metodológicas, los recursos didácticos y los criterios de evaluación.

Teniendo en cuenta lo anterior, se dará una breve descripción de los modelos pedagógicos aplicables a entornos en línea tenidos en cuenta para el desarrollo de este proyecto con sus principales características, ellos son:

1.1.1. La Pedagogía Tradicional

Propósito: Enseñar conocimientos y normas. El maestro cumple la función de transmisor, su propósito es enseñar conocimientos y normas.

Tiene una importante característica, parte del dicho que “la letra con sangre entra”, pero buscando, además de enseñar conocimientos, respeto a los adultos, colocando el aprendizaje como un acto de autoridad.

Su sustentación teórica más aproximada la describe Alain[5], cuando relaciona el aprendizaje con las etapas del hombre, el cual siendo niño desea ser adulto, y que ve en el maestro una persona que le brinda esa posibilidad pero a costa de abandonar el mundo de niño, según Alain el niño agradecerá al maestro el haberlo obligado a favorecer su deseo, porque el autor propone que los facilismos y el placer, no estimulan a las personas como lo hacen los retos y las dificultades. Por esto el acto de instruir no es mas que lograr que el niño acepte el esfuerzo

que ello causa y que encuentre en ello el camino hacia el ser hombre. Esta pedagogía sustenta la rudeza en el trato y justifica la imitación sucesiva como única posibilidad de que el estudiante aprenda.

1.1.2. El Modelo Instruccional: Respuestas Definidas Previamente.

Propósitos: Transmitir los saberes específicos y las valoraciones aceptadas socialmente. El método de exposición del maestro es oral y visual, dicha exposición se hará de manera reiterada y severa.

Según los diversos paradigmas pedagógicos el conocimiento puede ser transmitido, creado o reconstruido. Este modelo es netamente transmisionista, es decir, el conocimiento se crea fuera de la institución y llega a esta a través de las lecciones del profesor. El estudiante solo es un receptor que apropia conocimientos por medio de la imitación y reiteración. No se tiene en cuenta si se enseña a niños, adolescentes o adultos, el estudiante es un elemento pasivo del proceso y si atiende como es debido y repite un buen número de veces se garantiza el aprendizaje.

Los recursos didácticos son lo más parecidos a la realidad, ya que se deben repetir muchas veces para crear modelos mentales que generen el conocimiento en el estudiante. La evaluación busca determinar hasta cual punto la repetición logró imprimir los conocimientos en el estudiante[4].

1.1.3. La Nueva Escuela

Nace como una corriente en contra de la educación noble y autoritaria. Bajo el lema de la defensa de los derechos humanos, la libertad y el individuo; y con la fundamentación teórica del darwinismo que proponía a la acción como elemento central de la selección natural, condenando la pasividad con la desaparición y a la adaptación proveniente de la acción como única forma de sobrevivir.

Aunque parezcan situaciones muy filosóficas fueron estas que junto con algunos adelantos en la caracterización de las etapas evolutivas del hombre las que dan partida a la nueva corriente pedagógica conocida como la pedagogía de la acción o el activismo, en donde el hombre pasó de ser el objeto de la práctica educativa al sujeto de la misma.

Se defiende la acción como condición y garantía del aprendizaje. Manipular es aprender, ya que la manipulación directa de los objetos es la que permite el conocimiento de los mismos[4].

1.1.4. El Modelo Activista – Se aprende haciendo

Propósitos: No limitarse al aprendizaje, preparar estudiantes para la vida. Los contenidos sirven para preparar para la vida al estudiante, se debe estudiar la naturaleza y la vida misma.

El conocimiento será efectivo en la medida que repose en la experiencia. El estudiante pasa a ser elemento fundamental de los procesos educativos. El conocimiento es creado.

La escuela debe cambiar la concepción de solo transmitir conocimientos a los estudiantes hacia una formación más integral, donde, partiendo de las experiencias con el mundo real se prepare al estudiante para afrontar la vida. La experimentación permite la creación de conceptos simples y concretos y su aplicación a la vida debe generar un aprendizaje más complejo y abstracto. Es importante dar al estudiante libertad de palabra y acompañarla de acción.

Los recursos didácticos son cualquier cosa que al permitir su manipulación y experimentación permita educar los sentidos. Materiales diseñados para los estudiantes, no para el profesor y que no sean caminos hacia el conocimiento, si no el conocimiento mismo; la evaluación pretende hacer una valoración de los conocimientos creados.

1.1.5. Los Modelos Pedagógicos Contemporáneos

Propósito: El estudiante tiene que reconstruir activamente los conceptos de la ciencia e incorporarlo a sus estructuras de pensamiento.

La estructura curricular se organiza a partir de las motivaciones esporádicas y circunstanciales de los alumnos.

Según Reich (1993) los lineamientos para la escuela del futuro son:

- ❖ El favorecimiento de las operaciones de análisis.
- ❖ La formación de un pensamiento sistemático y global.
- ❖ El desarrollo de la habilidad para trabajar cooperativamente con los compañeros.
- ❖ Y la exigencia de formar individuos más creativos.

“Las evaluaciones que hacemos, la metodología y los recursos didácticos que utilizamos, no son ajenos a nuestra concepción pedagógica”[4].

El alumno se forma no sólo como resultado en su participación activa en la sociedad y en la ciencia, sino porque duda, porque puede privilegiar lo estético, en tanto imaginación y sensibilidad para influir en la lógica, como ciencia y para regresar a los valores sociales. Es el encuentro dialéctico de lo individual y lo

social. Lo social no absorbe al individuo sino que lo engrandece, por ello se realiza como individuo y como ser social, la escuela no puede estar ajena a la sociedad.

Si en el primer momento el centro es el maestro y el alumno aprendía en la instrucción; si en el segundo momento el alumno es el centro y se educa mediante su participación, desarrollándose intelectualmente; en el tercer momento se forma porque siente, valora y crea y, en consecuencia, es libre en sociedad[6].

1.1.6. La Pedagogía Conceptual

Los postulados principales de la pedagogía conceptual son:

- ❖ La escuela tiene que jugar un papel central en la promoción del pensamiento, las habilidades y los valores. Su propósito debe ser que cada día le presente nuevos, creativos y más difíciles retos a los alumnos.
- ❖ La escuela debe propiciar condiciones para lograr que se garantice al alumno la aprehensión de los conceptos básicos de la ciencia y las relaciones entre ellos.
- ❖ Los principales conceptos de la ciencia deben ser aprehendidos desde los primeros años de escolaridad para que luego estos puedan ser organizados en estructuras.
- ❖ Propiciar espacios y tiempos para desarrollar las operaciones intelectuales.
- ❖ Que los contenidos sean interesantes para el alumno, los motive.
- ❖ Desarrollar el hábito continuo de leer e investigar.
- ❖ La escuela futura deberá diferenciar la pedagogía de la enseñanza y el aprendizaje. Los contenidos deben comprender el conjunto de los conceptos básicos, las habilidades y destrezas básicas de cada área y los valores relacionados con ellos.
- ❖ Se deberán diferenciar los instrumentos del conocimiento (matemática, física) de las operaciones intelectuales (análisis, deducción) para promover cada una de ellas.
- ❖ Se tendrán que reconocer las diferencias cualitativas que existen entre los alumnos y actuar conforme a ello. La escuela tendrá que adecuar los espacios, los contenidos, los referentes y los tiempos a las características propias de los alumnos.
- ❖ Para asimilar los instrumentos de conocimiento es necesario que se desequilibren los instrumentos formados de manera espontánea: El desequilibrio es una fase necesaria en la asimilación conceptual.

Los elementos de un modelo pedagógico son la evaluación, los recursos, los propósitos, el contenido, el enfoque curricular, el método y la secuenciación, los modelos pedagógicos resuelven las mismas preguntas del currículo, solo que a un nivel mayor de generalidad y abstracción[7].

1.1.7. El Currículo y sus preguntas.

Un currículo es la caracterización de los propósitos, los contenidos, la secuenciación, el método, los recursos didácticos y la evaluación. Cada uno de estos elementos resuelve una pregunta pedagógica diferente pero interrelacionada con las demás, estas son[4]:

- ❖ **¿PARA QUÉ ENSEÑAR?:** Este constituye el fin, se trata de encontrar la finalidad y los propósitos adecuados para los cursos, resulta ser una condición necesaria para la consecución de la calidad educativa.
- ❖ **¿QUÉ ENSEÑAR?:** Constituye los contenidos curriculares; en esta pregunta se plasma el propósito del curso; al definir los contenidos curriculares es necesario tomar postura ante su carácter y sus jerarquías.
- ❖ **¿CUÁNDO, CÓMO Y CON QUÉ ENSEÑAR?:** La pregunta “¿cuándo enseñar?” se refiere a la estructura y secuenciación de los contenidos, al tiempo que el problema metodológico vinculado con la relación y el papel del maestro, el alumno y el saber conduce a la pregunta “¿cómo enseñar?”. El carácter y la finalidad de los medios, la ayuda y los recursos didácticos aparecen para resolver el interrogante “¿con qué enseñar?”.
- ❖ **¿CÓMO EVALUAR?:** Debe responder por el cumplimiento parcial o total de los propósitos, por el diagnóstico del proceso y por las consecuencias que de ello se deriven.

Dando respuesta a cada una de estas preguntas se logrará resolver muchos de los problemas que presenta la educación formal, convirtiendo este paso en un ciclo retroalimentado que ayudara a que las nuevas tendencias en la educación a ofrecer un aprendizaje efectivo y de calidad.

Este proyecto se basa en el currículo educativo definido por la institución educativa que haga uso de la herramienta, para este caso la Universidad del Cauca, pues como se menciona con anterioridad, el proyecto propone una herramienta de educación personalizada enfatizada en la educación en línea, pero teniendo siempre en cuenta la educación presencial y su fundamentación pedagógica, por ende el proyecto no entrará a replantear las preguntas, sino a apoyar la consecución de una respuesta satisfactoria a cada una de ellas con cierta especialidad en la pregunta referente al como y con que enseñar, ya que la herramienta busca dar nuevos matices a la relación entre los actores, diferentes medios para su comunicación sin que tiempo y espacio converjan, y diversas opciones para la estructuración y secuenciación de los contenidos basados en criterios de personalización como el perfil del estudiante.

1.1.8. Estrategias de Aprendizaje

“La estrategia de aprendizaje es el proceso de toma de decisiones, consciente e intencional, en el que el estudiante elige y recupera los conocimientos que necesita para hacer su trabajo.” [8]

Hoy en día la individualidad del estudiante como ser humano es un factor determinante para la toma de decisiones en el proceso de aprendizaje, dichas individualidades hacen del aprendizaje un proceso estratégico que busca que cada estudiante con sus características propias logre las habilidades deseadas para la aprehensión de los conocimientos transmitidos por la escuela, sea cual sea su modalidad.

Algunos de los factores a tener en cuenta para la toma de decisiones (para la generación de las estrategias de aprendizaje) son: Personales, de Trabajo y de Contexto; así como las variables obtenidas de la regulación o monitorización del proceso de aprendizaje desarrollado por cada estudiante y las habilidades cognitivas necesarias y adquiridas en el mismo proceso.

VARIABLES PERSONALES: Entre las variables personales se encuentran:

- ❖ **LOS OBJETIVOS:** Propósitos y expectativas que se pretenden conseguir con relación al trabajo que se hace
- ❖ **LOS CONOCIMIENTOS PREVIOS:** ¿Qué se sabe sobre el tema? ¿Qué se ignora? ¿Qué se puede hacer para obtener la información?
- ❖ **RECURSOS PERSONALES:** Consciencia de la disponibilidad individual de las capacidades que requiere el trabajo, especialmente la autonomía y responsabilidad del estudiante.
- ❖ **INTERÉS:** Manifestación de las motivaciones personales, intereses y desinterés sobre el tema que se está tratando
- ❖ **AUTO-CONCEPTO Y EFICACIA:** Análisis sincera de la propia imagen de uno mismo. Explicación de la visión de las propias capacidades y dificultades para resolver el trabajo.

VARIABLES DEL TRABAJO: Entre las variables del trabajo se encuentran:

- ❖ **LOS OBJETIVOS DEL TRABAJO:** Consciencia de la comprensión o no, de la demanda del trabajo. Comprender las instrucciones del profesor, las preguntas de un ejercicio o examen y también la finalidad del mismo.

- ❖ **CARACTERÍSTICAS DEL CONTENIDO:** Reflexión sobre los diferentes tipos de contenido (hechos, conceptos, procedimientos, valores...) y su estructura interna, amplitud y nivel de dificultad.

VARIABLES DEL CONTEXTO: Entre los factores del contexto se tienen:

- ❖ **EL TIEMPO:** Adecuación de las actividades al tiempo disponible.
- ❖ **EL LUGAR:** Incidencia del contexto físico en las acciones a tomar.
- ❖ **LOS MATERIALES:** Los recursos que se utilizan en función del trabajo.

Otra de las definiciones de estrategias de aprendizaje propone que las estrategias de enseñanza o *estrategias docentes* son procedimientos o arreglos que los profesores o agentes de enseñanza utilizan de forma flexible y estratégica, para promover la mayor cantidad y calidad de aprendizajes significativos en los alumnos. Debe hacerse un uso inteligente, adaptativo e intencional de ellas, con la finalidad de prestar la ayuda pedagógica adecuada a la actividad constructiva de los estudiantes.[9]

Ambas definiciones coinciden en afirmar que las estrategias de aprendizajes son los procesos llevados a cabo para conseguir un objetivo, en este caso el aprendizaje individualizado del estudiante y que esto implica un plan de acción sobre dicho aprendizaje teniendo en cuenta factores intrínsecos (de los estudiantes) y extrínsecos (del entorno) del proceso

El plan de acción a desarrollar y ejecutar requiere de la selección de una técnica de estudio o estrategia de aprendizaje y de la identificación de las habilidades cognitivas presentes en cada estudiante, para ello se dará una breve definición de la clasificación de las estrategias según la habilidad que se desea desarrollar en el estudiante y que lo conducen al conocimiento.

LAS ESTRATEGIAS DE APRENDIZAJE o técnicas de estudio que conducen al conocimiento son:

TECNICA PARA OBERVAR → Observar es dar una dirección intencional a la percepción. Están la *AUTO-OBSERVACIÓN* (El sujeto y el objeto se centran en uno mismo), la *OBSERVACIÓN DIRECTA* (Se Observa el hecho o el elemento en su lugar natural de acción), la *OBSERVACIÓN INDIRECTA* (Se Aprovechan las observaciones de otras personas o registros) y la *BÚSQUEDA DE DATOS*.

TECNICA PARA ANALIZAR → Analizar es destacar los elementos básicos de una unidad de información. Según la manera de percibir la información que nos llega se pueden resaltar diferentes tipos de Análisis: *ANÁLISIS ORAL, ANÁLISIS TEXTUALES, ANÁLISIS VISUAL*.

TECNICA PARA ORDENAR → Ordenar es disponer de forma sistemática un conjunto de datos, a partir de un atributo determinado Según los requisitos de la demanda o propios, podemos establecer diferentes criterios y formas de ordenar los datos: *ORDEN ALFABETICO, ORDEN NUMERICO, ORDEN SERIAL, ORDEN TEMPORAL, ORDEN ESPACIAL, ORDEN PROCEDIMENTAL* (instrucciones)

TECNICA PARA CLASIFICAR → clasificar es disponer un conjunto de datos por clases o categorías. Según el trabajo que hacemos con los datos y su correspondencia y apariencia gráfica-visual podemos establecer diferentes formas de Clasificación: *RESUMIR, RELACIONAR, CATEGORIZAR,*

TECNICA PARA REPRESENTAR → representar es la creación de nuevo o recreación personal, de unos hechos, fenómenos, situaciones. Según la forma de expresión escogida para el tratamiento de la información, podemos hablar de diferentes maneras de Representación: *GRÁFICA, ICÓNICA, VERBAL, CINÉTICO-GESTUAL (Mímica).*

TECNICA PARA MEMORIZAR → memorizar es el proceso de codificación, almacenamiento y reintegro de un conjunto de datos. Según la dirección del proceso (input-output) y la intención del mismo, se pueden distinguir diferentes formas de trabajo de memorización: *CODIFICAR, RECONOCER, RECONSTRUIR, MANTENER Y ACTUALIZAR.*

TECNICA PARA INTERPRETAR → interpretar es la atribución de un significado personal a los datos contenidos en la información que se recibe. Según los objetivos del trabajo, o personales. Se puede optar por diferentes formas de interpretación: *JUSTIFICAR, INFERIR, TRANSFERIR.*

TECNICA EVALUAR → evaluar es valorar la comparación entre un producto, unos objetivos y un proceso. En función del alcance y los propósitos encomendados, se pueden concretar diferentes formas y técnicas de evaluación: *DIAGNOSTICAR, VERIFICAR, REGULAR, DEMOSTRAR, VALORAR.*[10]

El prototipo desarrollado para el presente proyecto se enfatizó en la construcción de un modulo inteligente que permite tomar decisiones con respecto al proceso de aprendizaje del estudiante. Teniendo en cuenta los factores personales, del trabajo y del contexto mencionados con anterioridad se genera la estrategia de aprendizaje que aporte efectivamente a la actividad constructiva del estudiante, la funcionalidad de dicho modulo se evidenciará claramente en la estrategia de aprendizaje que este dé cómo resultado del proceso de personalización de los contenidos del curso; Esta estrategia estará representada básicamente con las técnicas para organizar el contenido, sugiriendo la secuenciación adecuada de los contenidos del curso para cada estudiante en particular, la técnica para

representar, ofreciendo mediante la secuenciación, una navegación distinta a cada estudiante por la estructura de contenido del curso a través de iconos habilitados o deshabilitados según sea el caso y la técnica de observación que se da en el proceso de la toma de decisiones pues el conjunto de experiencias previas de los estudiantes logrará una estrategia más eficaz en un nuevo estudiante con características similares a las presentadas en las experiencias. Las estrategias de aprendizaje planteadas por este proyecto pueden ser tan simples como aplicar el estilo de aprendizaje para un estudiante, o tan compleja que aborde desde las características psicológicas del estudiante hasta los aspectos involucrados en los ambientes de su aprendizaje como se ve en los Sistemas de Hipermmedia Adaptativa[11] (Ver CAPITULO 2).

1.2. El Aprendizaje y los Estilos de Aprendizaje

1.2.1. El Aprendizaje

“El aprendizaje es la actividad humana caracterizada por la producción de cambios duraderos y transferibles a nuevas situaciones como consecuencia directa de las prácticas realizadas” [7].

La producción de cambios duraderos tiene que ver con la mutación de conocimientos y conductas anteriores. Esta idea de cambio implica que para lograr el aprendizaje hay que desaprender. Ahora esos cambios pueden ser de sustitución o de reorganización.

- *Sustitución:* Los cambios de sustitución son puntuales, reversibles y su duración depende de la práctica.
- *Reorganización:* Los cambios de reorganización son generales e irreversibles y su duración es estable en el tiempo.

Esto indica que la educación virtual tendrá que articular los dos tipos de cambios pues habrá estudiantes que unas veces necesiten aprender por sustitución y otras por reorganización; Por tal razón los maestros deben saber las respuestas a las siguientes preguntas[7]:

❖ ¿Qué se aprende? → Resultados

Lo que las personas aprenden puede ser: sucesos o conductas, habilidades sociales, actitudes o representaciones sociales, información verbal o conceptos, técnicas o estrategias.

El aprendizaje de sucesos hace referencia a que una persona pueda establecer relaciones entre acontecimientos que ocurren a su alrededor. El aprendizaje de

conductas tiene que ver con la respuesta que el sujeto elabora para asumir o modificar sucesos.

El aprendizaje de habilidades sociales consiste en el manejo de esos elementos que nos permiten la interacción con los otros. Mientras que el aprendizaje de actitudes le dan al individuo la identidad social. Las representaciones sociales son un refinamiento de las dos anteriores y consiste en aquello que es compartido por la sociedad y que permite tomar una posición frente a la realidad.

El aprendizaje de información verbal consiste en el almacenamiento de datos (no información) en la memoria. El aprendizaje de conceptos es más complejo pues se le atribuye significado a lo que se aprende no son solo datos, es información.

El aprendizaje de técnicas hace referencia a las acciones secuenciales que permiten lograr un propósito, el aprendizaje de estrategias consiste en la planificación de las técnicas para alcanzar un objetivo.

❖ ¿Cómo se aprende? → Proceso

Los procesos que se siguen para el aprendizaje son básicamente cuatro: la motivación, la atención, la recuperación y transferencia y la conciencia.

La motivación es la que mueve al aprendiz, conocer lo que motiva al alumno, lo que desea conseguir y lo que está buscando debe ser el punto de partida para el aprendizaje en la educación virtual.

La atención es lo que permite fijar la mirada en el horizonte trazado. Permite que el estudiante se enfoque en cumplir su meta y hacer el proceso de aprendizaje responsablemente evitando los desvíos por los múltiples elementos que posee el ambiente virtual y que pueden distraer al estudiante de su objetivo.

La recuperación y transferencia se refieren a la capacidad de recordar conocimientos previos y poder relacionarlo con cosas nuevas.

La conciencia está ligada a la responsabilidad con la que el estudiante asuma su aprendizaje, pues en ambientes virtuales es él quien maneja su propio proceso

❖ ¿En qué se aprende? → Condiciones

Las condiciones de aprendizaje son aquellos factores en los cuales tiene incidencia el maestro quien debe propiciarlas para que esos procesos se den, las condiciones son fundamentalmente dos: la práctica y la interacción y cooperación entre aprendices.

La práctica por ser el factor de aprendizaje, debe ser el punto de mayor preocupación para el profesor, éste debe ser el principal promotor de dicha práctica y conseguir no solo relacionar esas prácticas con la teoría estudiada, sino que debe enfocarla a favorecer el aprendizaje del estudiante.

La interacción y cooperación entre los aprendices convierte al proceso de aprendizaje en una actividad social que enriquece no solo al grupo sino al

individuo, en este aspecto el profesor debe propiciar los espacios adecuados para que los alumnos puedan realizar esta actividad, como foros, Chat, etc.

1.2.2. Tipos de Aprendizaje

Los tipos de aprendizaje son parte del proceso de aprendizaje, mientras que las capacidades adquiridas son el resultado del aprendizaje. Los tipos describen las formas en que el aprendizaje se puede dar en el individuo, estas pueden ser a través de signos y señales, de asociaciones verbales, encadenamiento de hechos, respuestas operantes, de conceptos o principios[11].

Asumiendo los postulados de la conceptualización del aprendizaje, se distinguen dos tipos generales de aprendizaje, el aprendizaje cerrado y el aprendizaje abierto.

Aprendizaje Cerrado: Es aquel en el que el aprendizaje está estructurado previamente por los supuestos que maneja el maestro. Así es el maestro quien determina prácticamente todo el proceso y sus estudiantes se limitan a seguir las reglas de juego establecidas. La consecuencia de esto es el éxito de quienes por sus características personales se acomodan a la estructura planteada y el rotundo fracaso de quienes no la manejan. El aprendizaje cerrado solo tiene en cuenta un estilo de aprendizaje, el del maestro que lo propone. En resumen consiste en determinar patrones comunes.

Aprendizaje Abierto: Significa que hace flexibles algunos de los determinantes de aprendizaje. Por eso algunos autores prefieren llamarlo "*flexible*"; las decisiones sobre el aprendizaje no son el resultado de la arbitrariedad del docente. Por el contrario esas decisiones las toma el estudiante o el grupo de estudiantes

Pero no por esto se debe especular que en la educación virtual no existen reglas de juego claras, que no hay controles externos y que el maestro es solamente una figura decorativa que nada aporta, pues todo esto lo realiza el estudiante según su voluntad. Nada de esto es cierto, pues la autonomía del estudiante no implica ausencia de compromisos, de responsabilidades, fechas límites o plazos concretos. Tampoco implica que el maestro sea una figura pasiva, por el contrario el maestro realiza una tarea fundamental que es la de problematizador, de interlocutor válido que plantea las reglas de juego que deben seguirse para el éxito del proceso.[13]

Las principales características del aprendizaje abierto son:

- ❖ Reconoce al estudiante como interlocutor válido. No es un receptor pasivo, puede aportar a las decisiones.

- ❖ Se acomoda a las formas en que la gente aprende naturalmente. Teniendo en cuenta las diferencias entre los individuos (alumnos).
- ❖ Se abren múltiples opciones para el estudiante y se crean diversas alternativas de control.
- ❖ Se basa en materiales de aprendizaje centrados en el estudiante, sus ritmos, sus necesidades, sus intereses y sus posibilidades con las TIC's.
- ❖ Cuestiona los contenidos y produce nuevos conocimientos por reconstrucción, reconociendo los presaberes y enriqueciéndolos con otros aportes.
- ❖ Promueve la interacción entre los alumnos y la socialización del aprendizaje a través de múltiples alternativas de diálogo, multidireccionales y participativas como los chats, los grupos de discusión y el trabajo colaborativo.
- ❖ Ayuda a que los usuarios se atribuyan el mérito de su aprendizaje y a que desarrollen entusiasmo frente a su proceso.
- ❖ Resalta las competencias personales: Abstracción, pensamiento sistémico, experimentación, colaboración y cooperación.
- ❖ Implementa el aprendizaje descentralizado. Independiente de lugar, tiempo y espacio.
- ❖ Fortalecimiento de la autonomía del estudiante en su proceso de aprendizaje.[7]

Se considera que la herramienta desarrollada será de gran utilidad pues promueve en todos los sentidos el aprendizaje abierto; Pone al estudiante como un actor dinámico, un agente receptivo, que aporta para su aprendizaje resaltando su individualidad y por ende su papel en el aprendizaje. Todo esto convierte al presente proyecto en una opción para lograr un aprendizaje personalizado y efectivo. Según Unigarro en su libro de educación virtual "el aprendizaje abierto es una muy buena alternativa para lograr que los estudiantes desarrollen las competencias deseadas". [7]

1.2.3. Estilos De Aprendizaje.

El término 'estilo de aprendizaje' se refiere al hecho que, cuando se quiere aprender algo, cada individuo utiliza su propio método o conjunto de estrategias. Aunque las estrategias concretas que se utilizan varían según lo que se quiera aprender, cada individuo tiende a desarrollar unas preferencias globales. Esas preferencias o tendencias a utilizar más unas determinadas maneras de aprender que otras constituyen el propio estilo de aprendizaje[14].

Las diferencias en el aprendizaje son el resultado de muchos factores, como por ejemplo la motivación, el bagaje cultural previo y la edad. Pero esos factores no explican por qué con frecuencia se encuentran alumnos con la misma motivación y de la misma edad y bagaje cultural que, sin embargo, aprenden de distinta manera, de tal forma que, mientras a uno se le da muy bien redactar, al otro le

resultan mucho más fáciles los ejercicios de gramática. Esas diferencias sí podrían deberse, sin embargo, a su distinta manera de aprender.

El concepto de los estilos de aprendizaje está directamente relacionado con la concepción del aprendizaje como un proceso activo. Si se considera que el aprendizaje equivale a recibir información de manera pasiva lo que el alumno haga o piense no es muy importante, pero si se entiende el aprendizaje como la elaboración por parte del receptor de la información recibida parece bastante evidente que cada individuo elaborará y relacionará los datos recibidos en función de sus propias características.

Utilizando este concepto se puede caracterizar a los estudiantes de forma tal que se identifique su mejor estrategia de aprendizaje. Esto se logra a través de la clasificación de las actividades que el estudiante realice sobre el sistema, que harán parte de una gran base de conocimiento que servirá de soporte para la toma de decisiones del sistema.

Pero la realidad siempre es mucho más compleja que cualquier teoría. Por lo tanto es importante no utilizar los estilos de aprendizaje como una herramienta para clasificar a los alumnos en categorías cerradas. La manera de aprender de cada individuo evoluciona y cambia constantemente, como ellos mismos.

Existe una diversidad de concepciones teóricas que han abordado, explícitamente o implícitamente, los diferentes 'estilos de aprendizaje'. Todas ellas tienen su atractivo, y en todo caso cada cual la seleccionará según el aspecto del proceso de aprendizaje que le interese[15]. Para no dejar de lado esta diversidad, a continuación se mostrarán algunas de las clasificaciones más conocidas, de las cuales la primera [14] (Estilos de Aprendizaje – CHAEA) será la que se tendrá en cuenta para el desarrollo inicial del sistema, sin embargo, el prototipo desarrollado, fue diseñado para que a futuro se puedan adicionar otras clasificaciones de estilos de aprendizaje, enriqueciendo no solo el aspecto de personalización de los estudiantes, sino también las experiencias recogidas de dichos estudiante, ampliando de esta manera el área de impacto del módulo inteligente al dar mayor precisión en la toma de decisiones (ver CAPITULO 2).

Inicialmente CHAEA ha sido la clasificación escogida para el desarrollo del proyecto, no solo por que es la más reconocida en el medio educativo presencial y virtual, sino por que de los estudios de investigación realizados en el área, CHAEA es la clasificación que más experimentación real ha tenido en diferentes áreas del conocimiento entre las cuales se pueden nombrar: el estudio de la Universidad Autónoma de Manizales llamado "*Estilo de aprendizaje y rendimiento académico en un entorno virtual.*"², el estudio llevado a cabo por la Universidad de

² Este estudio pretende determinar la relación entre los estilos de aprendizaje (usando CHAEA) y el rendimiento académico en el curso virtual de Lectoescritura de la UAM. Disponible en línea en:

Madrid llamado “Evaluación de Estilos de Aprendizaje en estudiantes de enfermería”³ y el análisis hecho por a Universidad de Valencia llamado “Estilos De Aprendizaje Y Utilización De Las Tic En La Enseñanza Superior”⁴ entre otros.

❖ Estilos de Aprendizaje - CHAEA.

Catalina Alonso define los Estilos de Aprendizaje como “los rasgos cognitivos, afectivos y fisiológicos, que sirven como indicadores relativamente estables, de cómo los alumnos perciben, interaccionan y responden a sus ambientes de aprendizaje” [16].

Para P. Honey y A. Mumford hay cuatro estilos de aprendizaje, que a su vez son las cuatro fases de un proceso cíclico de aprendizaje: *Activo, Reflexivo, Teórico y Pragmático*. Para mayor información Ver ANEXO 1

❖ Organización de la información.

Se refieren a los estudios científicos, psicológicos, físicos y biológicos que se han realizado sobre el funcionamiento del cerebro humano. Dichos estudios y experimentos arrojaron resultados de la división interna del cerebro en lo que se refiere a la forma en que organizamos la información que aprendemos del medio y como relacionamos dicha información con tres partes visibles del cerebro: Reptilino o Central, Derecho e Izquierdo. Para mayor información Ver ANEXO 2

❖ Inteligencias Múltiples.

Las inteligencias múltiples promueven la idea de que la mayoría de los individuos tenemos un espectro de inteligencias. Cada una desarrollada de modo y a un nivel particular, producto de la dotación biológica de cada uno, de su interacción con el entorno y de la cultura imperante en su momento histórico. Las combinamos y las usamos en diferentes grados, de manera personal ya sea para el proceso de aprendizaje o para cualquier proceso en general. Esta teoría define 8 inteligencias: Musical, Corporal, Lingüística, Matemática, Espacial, Intrapersonal, Interpersonal y Naturalista [17]. Para mayor información Ver ANEXO 3

<http://www.cibersociedad.net/congres2004/foros/conversa.php?idioma=es&tematica=&grup=&id=&pagina=23>

³ El objetivo de este estudio fue la identificación de las preferencias personales del estilo de aprendizaje de la población española de estudiantes de Enfermería, Disponible en línea en: <http://www.um.es/eglobal/3/pdf/03d03.pdf>

⁴ Este estudio en el que se pretende ver si existen relaciones entre los distintos estilos de aprendizaje de los sujetos y la utilización que hacen de las TIC, Así como, si existen diferencias entre las diversas formaciones recibidas, preferencias profesionales, curso o en relación al género. Dirigida a Maestros y futuros Pedagogos, personas que están o estarán vinculadas a la formación formal y no formal. Disponible en línea en: <http://www.virtualeduca.org/virtualeduca/virtual/actas2002/actas02/117.pdf>

❖ **Según el sistema de representación**

El Sistema de representación propone que todos los seres humanos tenemos tres grandes sistemas para representar mentalmente la información, el visual, el auditivo y el kinestésico. Utilizamos el sistema de representación visual siempre que recordamos imágenes abstractas (como letras y números) y concretas. El sistema de representación auditivo es el que nos permite oír en nuestra mente voces, sonidos, música. Por último, cuando recordamos el sabor de nuestra comida favorita, o lo que sentimos al escuchar una canción estamos utilizando el sistema de representación kinestésico [17]. Para mayor información Ver ANEXO 4

❖ **Según la categoría bipolar (Felder y Silverman)**

Silverman clasifica los estilos de aprendizaje de los estudiantes en cinco dimensiones, relacionadas a: qué tipo de información percibe mejor ¿Sensorial o intuitiva?, En qué modalidad percibe más efectivamente la información sensorial ¿Visual o verbal?, Cómo prefiere procesar la información que percibe ¿Activamente o reflexivamente? Cómo logra entender el estudiante ¿Secuencialmente o globalmente?, El estilo de aprendizaje de un estudiante vendrá dado por la combinación de las respuestas obtenidas en las cinco dimensiones [18]. Para mayor información Ver ANEXO 5

1.3. Educación en Línea

“La sociedad de la información tiene dos conceptos que están articulados armónicamente a ella: el aprendizaje y la información” [19].

La educación ha sido en la historia, la puerta de entrada al aprendizaje. “Los seres humanos somos auténticos informívoros, necesitamos información para sobrevivir igual que alimento, calor o contacto social. Todos nos alimentamos de información que nos permite no solo predecir, sino también controlar los acontecimientos de nuestro entorno. Predicción y control son dos de las funciones fundamentales del aprendizaje, incluso en los organismos más simples” [20]. La evolución del conocimiento va de la mano con la evolución de las especies.

La clave del éxito de las instituciones educativas, entonces será la articulación entre la eficacia de la transmisión de información considerada como valiosa debido a su utilidad y por supuesto las posibilidades de información.

Algunos de los cambios educativos que se proponen son:[7]

- Cambio de la estructura docente actual. Los maestros deben apoyarse en las redes de la información, manejar las TIC's, el maestro ya no se comporta como un guardián transmisor, sino como un proponente.
- La labor educativa ya no es exclusividad de los docentes de profesión, ahora es una labor multidisciplinaria.
- Revolución en contenidos y programas, se debe enseñar lo que lleve a solucionar los problemas inmediatos de la vida cotidiana.
- Enfoque del conocimiento hacia el futuro inmediato, enfocar en preparar los jóvenes de hoy, no a las generaciones del mañana.

La posibilidad que este estilo educativo (educación virtual) nos brinda, para propiciar el desarrollo humano científico, económico y técnico es francamente incuestionable, siempre y cuando logremos articular 3 elementos claves: la pedagogía, lo comunicativo y lo tecnológico, lo pedagógico como el horizonte de sentido; lo comunicativo como el ámbito del proceso educativo y la tecnología como la herramienta que favorece el enfoque de los 2 primeros. Así, y solo así podremos beneficiarnos de las bondades de esta nueva manera de hacer educación en el marco de la sociedad del conocimiento.

La educación virtual, no es simplemente una singular manera de hacer llegar la información a lugares distantes y a muchísimas personas sino que es toda una perspectiva pedagógica.

Para llegar a su definición se hablará primero de lo que es la educación partiendo desde su concepción filosófica, luego se hablará del concepto de educación virtual (o en línea) y por último de su planeación instruccional.

1.3.1. Concepto de Educación

La educación es una acción, no un suceso; existe porque sus protagonistas saben lo que hacen y desean hacerlo, la educación no es un suceso, un acontecimiento arbitrario o inconsciente. La educación consiste en la influencia que un sujeto recibe desde el exterior. La información es el horizonte de sentido de la educación, es resultado de lo que cada sujeto elige para su propia vida a partir de una serie de opciones y espacios que se le presentan en el transcurrir de su existencia.

Según HABERMAS - Filósofo y sociólogo alemán, representante más conocido de la teoría crítica en la actualidad en su escrito "CONOCIMIENTO E INTERÉS" dice que: ***"el conocimiento siempre resulta de algún interés"***, para él hay tres intereses fundamentales que llevan al hombre a conocer algo:[21]

INTERÉS TÉCNICO: También conocido como Empírico-Analítico, es el interés de control de las cosas, es lo que nos lleva a explicar los fenómenos a partir, del establecimiento de leyes.

INTERÉS PRÁCTICO: También conocido como Histórico-Hermenéutico, es el interés de comprender las cuestiones centrales de la vida, relacionar la existencia con su alrededor, trata de comprender las relaciones entre las personas.

INTERÉS EMANCIPATORIO: También conocido como Critico-Social, es el interés por la autonomía y la libertad vividas racionalmente.

Habermas propone además que: **“La educación es una acción comunicativa”** donde se pueden encontrar 3 acciones.[22]

ACCIONES INSTRUMENTALES: Son las que se realizan entre las personas y los objetos, para conseguir un objetivo concreto, un ejemplo es lo que hace el carpintero con el martillo, construir una silla.

ACCIONES ESTRATÉGICAS: Son las que se produce entre las personas y tienen como fin organizar un grupo para conseguir una meta.

ACCIONES COMUNICATIVAS: Se dan entre personas y su finalidad es lograr el entendimiento (individualidad y autonomía).

Después de haber dado una visión general del concepto de educación serán mucho más reconocibles los beneficios que la tecnología como herramienta proporciona a la educación tradicional, dando paso a la educación virtual o en línea.

1.3.2. Concepto de Educación en Línea.

Cabe anotar que hay diversas definiciones de la educación virtual, incluso algunos autores prefieren llamarla educación en línea a continuación daremos algunas de ellas:

Educación a Distancia: La educación a distancia fue definida como una institución basada en la educación formal donde el grupo de enseñanza está separado, y donde los sistemas de telecomunicaciones son usados para conectar aprendices, recursos e instructores[23].

E-Learning: Se puede definir como el uso de las tecnologías multimedia para desarrollar y mejorar nuevas estrategias de aprendizaje, Se define como la enseñanza entregada a través de un computador por via CD-ROM, Internet, Intranet que tiene las siguientes características:

- ❖ Incluir contenidos relevantes para el objetivo de aprendizaje.
- ❖ Usar métodos instruccionales tales como ejemplos y prácticas para ayudar en el aprendizaje.
- ❖ Usar elementos comunicativos como palabras y figuras para entregar el contenido y los métodos.
- ❖ Construir nuevos conocimientos y habilidades conectadas con el objetivo del aprendizaje individual o para un funcionamiento organizacional mejorado[24].

Educación Virtual: Es un proceso educativo con todo el sentido que requiere la educación formal, pero donde la relación comunicativa entre maestro-discípulo es una relación mediada por tecnologías de la información y la comunicación, que hace posible el encuentro sin la necesidad que cuerpos, tiempos y espacio confluyan.⁵

Características destacables:

- Está disponible en el lugar donde se encuentren los alumnos.
- Se acomoda a los tiempos del estudiante.
- Mayor responsabilidad del alumno en su propio aprendizaje.
- Brinda tanto al maestro como a los alumnos la posibilidad de invertir más tiempo en actividades educativas.
- Ofrece alternativas a los alumnos sobre ritmos, metodologías, formatos, profundización de contenidos, etc., favoreciendo así los procesos particulares.
- Incluye la tecnología apropiada y efectiva.
- Permite generar verdaderos procesos de autoevaluación, coevaluación y heteroevaluación[7].

Como es posible evidenciar, las definiciones anteriores coinciden en afirmar que son las condiciones del ambiente la principal diferencia con la educación presencial, permitiendo que el proceso de aprendizaje se realice independientemente del tiempo y el espacio, ambas definiciones constituyen una forma de educación a distancia. Por otro lado, todas las definiciones que convergen alrededor del tema de la educación en línea han sido planteadas por autores con un vasto conocimiento en el área y varios trabajos de investigación que permiten fundamentar acertadamente sus posiciones.

Este proyecto no pretende entrar a debatir dichas definiciones ni proponer otras nuevas, lo que se pretende es hacer un aporte significativo en el área de la educación Virtual o en línea para que esta logre llenar las expectativas de los protagonistas del proceso de aprendizaje⁶ especialmente al estudiante; todo esto

⁵ Manuel Antonio Unigarro, “cuando hablamos de educación virtual no hacemos referencia a una educación simulada o ficticia, por el contrario queremos expresar que es un proceso educativo con todo el sentido que posee el concepto de educación y que se diferencia de esta en el medio en el que trabajan los actores del aprendizaje”.

⁶ Estudiante o alumno, Profesor o Tutor.

a través del correcto diseño instruccional de los cursos que, para todas las modalidades de educación es el aspecto más relevante en la consecución de un proceso de aprendizaje de calidad.

Por esta razón a lo largo del desarrollo del presente proyecto se hablará de educación virtual y educación en línea haciendo referencia a la educación no presencial donde los medios telemáticos y las TIC's más que una opción son una condición de ella, sin ánimo de restarle importancia a sus definiciones.

1.3.3. El Método Educativo, Los Medios Y La Tecnología

El método del modelo de educación virtual tiene como eje el proceso de comunicación educativa. Para que la comunicación educativa pueda darse desde la perspectiva de método, es necesario que el proceso educativo virtual garantice los siguientes aspectos:

- Interactividad: Es la relación hombre – máquina y no la relación hombre – máquina – hombre, es una acción instrumental no comunicativa, es decir, en la interactividad solo se tienen en cuenta las acciones que el ser humano realiza con la maquina, no como responde esta ante dichas acciones.
- Interacción: Es la relación hombre – hombre, en el caso de la educación es la relación estudiante – estudiante, estudiante – maestro y maestro – maestro, constituyen los intercambios comunicativos entre los actores humanos del aprendizaje.
- Movilidad: El Ciberespacio constituye el medio principal de transferencia de información y la principal herramienta de movilidad pues provee al hombre acceder a ella a través de cualquier tipo de dispositivos ya sea fijos o móviles.
- Convertibilidad: Esta es la propiedad que permite al usuario hacer una libre transferencia de información a través de medios diferentes.
- Conectabilidad: Es la posibilidad de acceder a cualquier tipo de información, en cualquier momento.

Los medios de la educación virtual pueden ser:

Expositivos: El profesor expone a los alumnos sus trabajos en el tema. Solo se busca transmitir información, es un medio unidireccional.

Activos: El estudiante hace la investigación y el profesor aclara dudas.

Interactivos: Comunicación entre los actores profesor – estudiante y viceversa, permite la interlocución igual que el medio anterior, son considerados bidireccionales.

1.3.4. Modelos para el diseño instruccional de cursos en línea

Al igual que la educación presencial, la educación a distancia y todos los modelos educativos, la educación virtual se centran en propiciar un ambiente adecuado que permitan al estudiante asimilar nuevos conocimientos y enriquecer los ya existentes, por esta razón cualquier modelo para el diseño instruccional de un curso, será útil para cualquiera de los modelos educativos mencionados, teniendo en cuenta que dicho diseño tendrá bases pedagógicas y que en cualquiera de los casos la pedagogía es un factor importante.

Teniendo en cuenta lo anterior se hablará de cuatro modelos que actualmente se utilizan para el diseño de los cursos, especialmente para los cursos de educación a distancia y por razones ya comentadas se pueden utilizar para el diseño de los cursos en línea. Ninguno de los cuatro modelos es nuevo ni novedoso, dos de ellos están basados en modelos pedagógicos de los años 50's y 60's. Estos modelos son:

- ❖ **Diseño Instruccional Lineal:** Se basa en instrucciones programadas linealmente. Primero, un área de contenido como *fundamentos de la educación a distancia* se puede dividir en ideas importantes llamadas módulos, los módulos de instrucción pueden dividirse en temas y cada tema es un evento instruccional o experiencia de aprendizaje seguida por algún tipo de valoración como un test que debe ser aprobado por el estudiante, para pasar a los módulos siguientes a si continuar el proceso. Para mayor información Ver ANEXO 6
- ❖ **Diseño instruccional Ramificado (Branched):** Es similar al lineal con dos diferencias importantes. La primera es que su forma de evaluación es más sofisticada es el sentido que diagnostica el progreso de los estudiantes y su entendimiento de los conceptos y temas. La segunda diferencia importante es el uso de eventos instruccionales o experiencias de aprendizaje alternativos. Para mayor información Ver ANEXO 7
- ❖ **Diseño instruccional de Hipercontenidos (Hypercontent):** Al igual que todos los demás, tiene módulos, temas y conceptos. las experiencias de aprendizaje son diseñadas y producidas y los temas son presentados usando textos, audio, graficas, dibujos y videos. el control lo tiene el estudiante y los temas pueden ser estudiados de forma aleatoria, de manera no secuencial o en orden de hipercontenidos. Ver ANEXO 8
- ❖ **Diseño directo del estudiante (Learner):** En este modelo, el diseñador instruccional identifica los módulos o capítulos y los temas, incluyendo las experiencias de aprendizaje, pero no coloca ningún tipo de orden o secuencia

dentro del capítulo o entre los módulos. El estudiante decide en que orden va a estudiar los temas e incluso que temas va a estudiar. El estudiante construye su propia estrategia instruccional y su propio diseño instruccional, los estudiantes se mueven a través de los módulos en el orden que ellos escojan. Cualquier requisito que se tenga se pasa al estudiante que hace las veces de diseñador instruccional. Ver ANEXO 9

A pesar de las diferencias en metodología y uso, estos modelos tienen grandes similitudes. Primero, la instrucción es dividida en módulos, aunque diferentes personas en el medio utilizan términos como unidades o bloques en lugar de módulos, la verdad es que se trata de la división de los contenidos del curso en partes más pequeñas[23]. Teniendo en cuenta la anterior teoría el presente proyecto al igual que algunos profesores mezclan los cuatro modelos dentro de sus procesos de diseño instruccional según sea el caso. Sin embargo por el mismo enfoque del proyecto (educación en línea) se hace más uso del diseño instruccional de hipercontenidos.

1.4. El Estándar SCORM (Sharable Content Object Reference Model)

El modelo de referencia para compartir objetos de contenidos - SCORM (Sharable Content Object Reference Model) [31] es una iniciativa del departamento de defensa de los Estados Unidos que forma parte del programa de Aprendizaje Avanzado Distribuido - ADL (Advanced Distributed Learning) que fue presentado a finales del año 1997, con unos objetivos bastante ambiciosos en lo que se refiere a la educación, especialmente a la educación en línea, como la generación de educación de calidad, a la medida de las necesidades de cada estudiante y puesta en el lugar que quiera y al momento que quiera; todo ello apoyado en una plataforma tecnológica que soporte todas las características antes mencionadas.

Esta estandarización fue quizás la primera en alcanzar interés mundial y en aumentar la preocupación de los profesionales del medio por realizar aportes que enriquecieran la labor emprendida en este estándar. Esto trajo consigo una serie de mejoras y de versiones de SCORM que fueron el fruto de las experiencias adquiridas por medio del uso de éste en áreas prácticas y reales, es decir, en aulas virtuales. Actualmente la versión más reciente es la de Julio 22 del año 2004 esta es la versión 1.3.1 y contiene una serie de documentos en donde se muestran cada parte del estándar de forma individual.

Un primer documento contiene una breve descripción general del estándar (Overview), un segundo documento describe los modelos de secuenciación y navegación de los cursos (Sequencing and Navigation), un tercer documento

muestra todo el modelo de agregación de contenidos que constituye el eje central del estándar (Content Aggregation Model) y por ultimo tenemos el documento que describe la parte tecnológica del estándar, es decir, como lograr introducir todas las descripciones de los documentos anteriores a una plataforma tecnológica que lo soporte (Run-Time Environment). Interiormente, ellos tienen algunos subtemas como lo muestra la Figura 1.

1.4.1. Importancia de SCORM en este proyecto

La importancia del estándar SCORM en este proyecto, y en general en el proyecto Unicauca Virtual Fase II, radica en el aporte conceptual que realiza en lo que se refiere a la estructuración y manejo de los cursos virtuales. En él, se encuentran definidos y estandarizados algunos de los conceptos más relevantes del área de educación en línea, el presente proyecto usará como base (por lo menos en la parte conceptual) este estándar ya que se encuentra internacionalmente reconocido y es de una amplia aceptación. Por ello, se presentarán en este capítulo algunas de las definiciones conceptuales del estándar y su relación con el trabajo que se ha realizado en el presente proyecto.

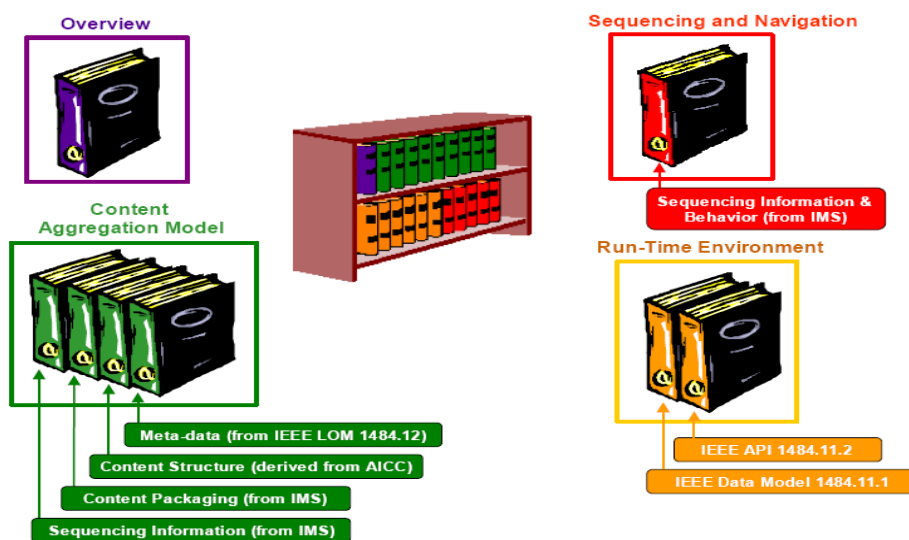


Figura 1. Librería de SCORM (Tomado de SCORM sequencing and Navigation Version 3.1.3)

1.4.2. Componentes del Estándar SCORM

A continuación se mostrara el funcionamiento y la interoperabilidad de algunos de los componentes del estándar que resultan importantes para el presente proyecto.

En esta descripción se omite el *Overview* dado que no hace parte de uno de los componentes funcionales del estándar, solo busca introducir al lector en el contexto del E-learning y describir en alto nivel la base conceptual del estándar, lo

cual ya se ha expuesto a través de este documento. Igualmente se dejará de lado el libro correspondiente a la plataforma tecnológica, ya que el objetivo de este proyecto no es implementar el modelo propuesto por SCORM, sino más bien hacer una adaptación de algunas de sus características a las necesidades propias del proyecto. Estos componentes son:

⇒ **El modelo de agregación de contenidos (Content Aggregation Model – CAM):**

Provee la información general de la composición, estructuración y secuenciación de los contenidos dentro del estándar independientemente del modelo pedagógico, esto hace que de una u otra forma pueda representar cualquier modelo pedagógico que se desee implantar en el curso, lo cual favorece mucho las características del proyecto, ya que precisamente lo que se busca es dar soporte a modelos pedagógicos diferentes que satisfagan las características propias de los estudiantes que realizan los cursos. Las reglas que establecen este comportamiento se dividen en cuatro partes:

- El Modelo de Contenidos proporciona una terminología común para ser usada en todo el CAM.
- El Modelo de Empaquetado ofrece la descripción y los requerimientos para componer objetos de aprendizaje y formar unidades más complejas.
- Metadatos para la descripción de los propios componentes de SCORM
- Normas de Secuenciación y Presentación que luego serán analizadas más a fondo en el libro de navegación y secuenciación. Ver ANEXO 10

⇒ **Secuenciación y Navegación - SN**

Aunque es el último libro del estándar decidimos nombrarlo en este momento dada su importancia para nuestro proyecto, ya que es esta parte de SCORM la que nos brinda las herramientas necesarias para generar las estrategias de aprendizaje personalizadas a cada tipo de estudiante; y este es uno de los objetivos que se persiguen en este proyecto. Además, este libro proporciona la fundamentación teórica que utilizamos en la Fase II del proyecto Unicauca Virtual para la estructuración y organización de los cursos virtuales, desde el manejo de los árboles instruccionales hasta los metadatos de los recursos. En este libro se define que recursos se van a mostrar en el curso, en que orden deben mostrarse para que sean efectivos en el proceso de aprendizaje del estudiante y como el estudiante debe hacer la navegación a través del curso virtual.

Está formada por cinco partes:

- ✓ Terminología y conceptos básicos sobre secuenciación.
- ✓ Modelo de definición de secuenciación.
- ✓ Modelo de comportamiento de secuenciación.
- ✓ Requerimientos y control de la navegación.

✓ Modelo de datos de navegación.

Para ver la información acerca de cada una de las partes antes mencionadas ver ANEXO 11

Es importante aclarar que el presente proyecto propuso ciertas modificaciones sobre el estandar para satisfacer algunos de los requerimientos del proyecto. Algunas de esas modificaciones hacen una extensión del estándar y otras simplemente implementan ciertos conceptos ya definidos por el modelo. A continuación se nombrarán los conceptos utilizados:

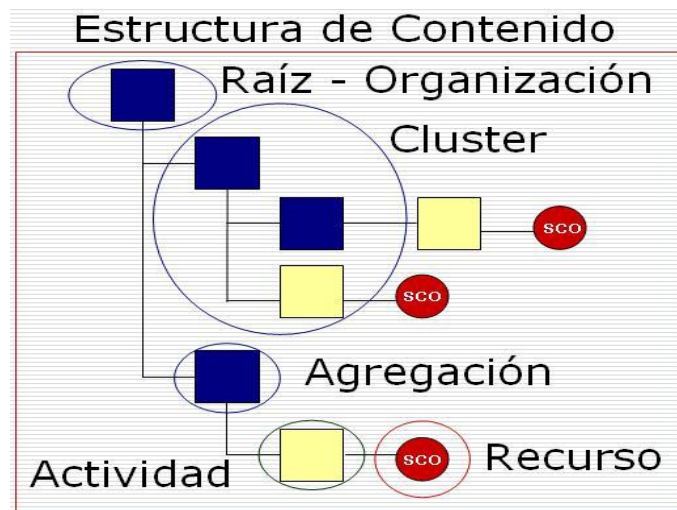


Figura 2. Conceptos usados del estandar SCORM

- **Estructura de Contenidos:** Que en términos del estandar se define como *Paquete de Contenidos* y representa la organización del contenido de un curso en línea con modulos, temas y capitulos, traducidos en agregaciones, actividades y recursos. Sobre dicha organización se aplican los criterios de navegación y secuenciación definidos por SCORM.
- **Reglas de Secuenciación:** Define las reglas de navegación a través de la estructura de contenidos y permite personalizar dicha navegación de acuerdo a los criterios establecidos por el proyecto, en esta parte el presente desarrollo aporta al estandar una ampliación en el rango de acción de dichas reglas, pues la secuenciación definida por SCORM permite que un nodo padre del arbol (que para el contexto del presente proyecto se tratará de ahora en adelante como AGREGACION) posea comportamientos dependiendo del cumplimiento de las condiciones de los nodos hijos (tratado como ACTIVIDADES). Mientras que el aporte permite que dichas condiciones se pueden ejercer no solo sobre los nodos hijos del padre actual, sino sobre los nodos hijos de otros contenidos padres. Para la secuenciación de contenidos, SCORM tiene en cuenta tres factores importantes que tambien son tenidos en cuenta para este proyecto:

- El árbol de actividades o Estructura de Contenidos
 - La estrategia de secuenciación definida
 - El comportamiento definido ante cualquier tipo de actividad del estudiante que inicie uno de los eventos. Ver ANEXO 11
- **Reglas Rollup:** Al igual que las reglas de secuenciación, las reglas rollup pertenecen a las agregaciones (contenido padre), pero para este caso, las rollup definen el estado en lugar de la acción que tomara dicha agregación con respecto al cumplimiento de las condiciones por parte de sus actividades (nodos hijas). Esta si aplica solo a las actividades hijas.
 - **Objetivos:** Al igual que en el estandar, los objetivos para este proyecto se definen para las agregaciones y se satisfacen a partir de las actividades ya sea propias o hijas de otras agregaciones.
 - **Cluster:** Al igual que en el estandar SCORM, se define como el bloque de datos conformado por un contenido padre y sus descendientes primarias unicamente, involucran agregaciones y actividades de acuerdo a la estructura de contenidos sobre la que se este trabajando.

Dichas modificaciones surgieron como solución a ciertas limitaciones o falencias que según el contexto del presente proyecto posee el modelo de referencia SCORM, dichas falencias son: [33]

- ✓ Ausencia de una secuenciación dirigida por técnicas de inteligencia artificial.
- ✓ Ausencia de una secuenciación basada en la planificación.
- ✓ Ausencia de una secuenciación que requiera información de sistemas externos.
- ✓ Ausencia de aspectos que promuevan el Aprendizaje colaborativo.
- ✓ Ausencia de estrategias de sincronización entre actividades realizadas en paralelo.

1.5. Los Sistemas Tutores Inteligentes

La más notable contribución de la ciencia cognitiva a la tecnología educacional es lo que ha sido conocida como Sistemas Tutores Inteligentes (STI). Estos sistemas, también, están basados principalmente en los desarrollos de la Inteligencia Artificial (IA) y pueden definirse como programas de enseñanza-aprendizaje basados en el computador cuya finalidad última es facilitar procesos de aprendizaje altamente personalizados[25].

Las diferencias fundamentales con respecto a los Tutoriales residen en la forma en la que se conciben ambos diseños [25]. En un programa tutorial tradicional se trata de inducir al estudiante la respuesta correcta mediante una serie de estímulos que han sido cuidadosamente planificados, a modo de discurso socrático. En cambio en un STI se intenta simular alguna de las capacidades cognitivas del estudiante y se utilizan los resultados de tal simulación para con base en ellos tomar decisiones instruccionales.

Algunas de las limitaciones que poseen los STI son: la dificultad de comunicación con el estudiante, la elaboración de conclusiones sobre su conocimiento y sus características individuales basandose en su conducta [26].

Los STI apoyan al profesor durante el proceso de ejercitación del alumno. Para lograr esto, el STI deberá contar con la capacidad inteligente, de observar qué está intentando realizar el usuario, para determinar su comportamiento, encontrar sus fallas, aciertos y retroalimentar al usuario además de almacenar con dicha información una base de conocimiento, la cual será usada tanto por el profesor como por el alumno para guiar el proceso de aprendizaje. Esta parte inteligente es el objetivo principal de este proyecto. Finalmente se deben llevar todos estos procesos con un correcto modelo de planificación estratégica, para después evaluar los resultados y verificar si las actividades planteadas en el curso obtuvieron resultados significativos en el desarrollo personal y de conocimiento de los estudiantes[25].

Parece existir cierto acuerdo respecto a la estructura y los nombres de los componentes de los Sistemas Tutoriales Inteligentes, la dimensión instruccional; la dimensión comunicación hombre-máquina; y la dimensión de conocimiento experto. La estructura de un STI se muestra en la Figura 3.

Cada modulo del STI tiene relación con por lo menos alguno de los otros componentes, pero a la vez se encuentran con cierto grado de independencia de los demás. El presente proyecto se centra en el Modulo Tutor pues es el modulo encargado de hacer la toma de decisión respecto al perfil del estudiante, pero a su vez interactúa con los tres módulos restantes

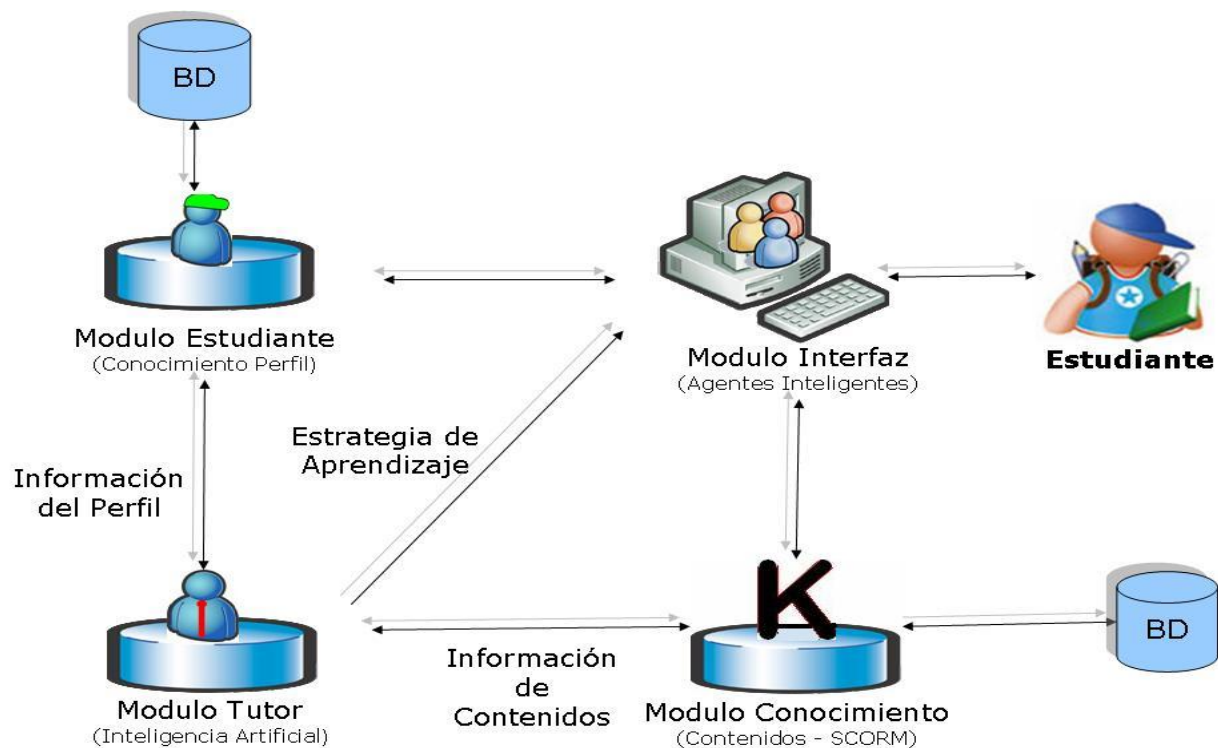


Figura 3. Estructura de un Sistema Tutor Inteligente

A continuación se muestran las funciones de cada módulo:

El módulo del tutor se caracteriza por controlar el sistema. Es el encargado de determinar qué hacer, utilizando conocimientos, tanto pedagógicos como del dominio de la aplicación. El tutor contiene las estrategias y procesos que gobiernan la comunicación entre el sistema y el estudiante. El tutor crea para cada estudiante un fichero a partir de los datos que lo identifican. El módulo Tutor utilizará toda la información contenida en el módulo del estudiante y a través del uso de IA generará la estrategia de aprendizaje que sea mas conveniente para el estudiante. Este proceso es fruto de todo un proceso de investigación que comienza con la elección de un enfoque de IA para este caso.

El módulo del Conocimiento es también conocido como módulo experto y contiene el conocimiento acerca del área de estudio. Manipulando este conocimiento el sistema debe ser capaz de responder las preguntas del estudiante y proporcionarle una guía. El Módulo de Conocimiento representa todos los contenidos de las materias que se encuentran en el sistema, módulo que esta desarrollado por medio del LMS y SCORM.

El modulo Interfaz proporciona el ambiente con el que el estudiante va a interactuar, sirve para conocer y controlar la actuación del estudiante durante el proceso de aprendizaje, el modulo de Interfaz se compone de un sistema multiagentes que realiza dos tareas fundamentales; la primera es monitorear las actividades que el estudiante realice sobre el sistema y almacenar un grupo de datos en el modulo del Estudiante que son analizados por el módulo tutor para decidir la estrategia a seguir. La segunda se refiere a la ejecución de la estrategia de aprendizaje que se establezca para cada estudiante por parte del modulo Tutor.

El modulo del Estudiante: tiene, además de la información personal del estudiante y su información académica, una serie de estructuras de almacenamiento que permitan guardar su estrategia de aprendizaje elegida, su estilo de aprendizaje, las actividades que realiza sobre el sistema durante la ejercitación del estudiante en el proceso de aprendizaje y otra serie de información cuyo único objetivo es servir de base de conocimiento sobre la cual el Modulo Tutor tomará la decisión además, basado en esta información, el sistema determina el aprendizaje del estudiante en función de las habilidades adquiridas durante las sesiones de trabajo.

1.6. Enfoques De Inteligencia Artificial

Los problemas abordados por la Inteligencia Artificial (IA) configuran un amplio espectro. Tienen muy poco en común excepto dos aspectos fundamentales. El primero, y el más sencillo, es que todos son muy complicados. El segundo nace tres décadas después de que se comenzaran los estudios e investigaciones en lo que respecta a la IA, y es que *“la inteligencia necesita de conocimiento”*. Al llegar a esta deducción, inevitablemente, se explicó un poco más el primer aspecto, ya que el conocimiento como tal, posee características poco deseables si se habla de sencillez. Tales características son [27]:

- Es voluminoso
- Es difícil de caracterizar con exactitud
- Cambia constantemente
- Se distingue de los datos en que se organiza de forma tal que se corresponde con la forma en que va a ser usado

1.6.1. Representación del conocimiento

Un buen sistema de representación del conocimiento debe tener cuatro características fundamentales[28]:

- **Suficiencia de la representación:** La capacidad de representar todos los tipos de conocimiento necesarios en el dominio.
- **Suficiencia deductiva:** Capacidad para manipular las estructuras de la representación con el fin de obtener nuevas estructuras que se correspondan con un nuevo conocimiento deducido a partir del antiguo.
- **Eficiencia deductiva:** Capacidad de incorporar información adicional en las estructuras de conocimiento con el fin de que los mecanismos de inferencia puedan seguir las direcciones más prometedoras.
- **Eficiencia en la adquisición:** Capacidad de adquirir nueva información con facilidad. En el caso más simple es aquél en el que una persona inserta directamente el conocimiento en la base de datos. Idealmente, el programa sería capaz de controlar la adquisición de conocimiento por sí mismo[28].

1.6.2. Técnicas de Representación del Conocimiento

Existen múltiples técnicas para la representación del conocimiento, se pueden enumerar las siguientes:

❖ Conocimiento Relacional Simple:

Es el más simple, pero igualmente es el más pobre a la hora de obtener información. Se compone de un conjunto de atributos que junto con unos valores asociados permite describir los objetos de la base de conocimiento. Es el que obtenemos en las bases de datos. Sin embargo, cabe anotar que brinda una excelente plataforma para mecanismos de inferencia más avanzados que utilicen dichas relaciones, por ejemplo, *Datamining*. Es ampliamente utilizado y para el presente proyecto ofreció las características necesarias para representar la información[28].

❖ Conocimiento Heredable:

Este tiene sus bases en las mismas relaciones del ítem anterior, pero aumenta su riqueza semántica con la inclusión de mecanismos de inferencia que operen sobre la estructura de la representación. Dicha estructura debe estar diseñada de acuerdo con el mecanismo de inferencia que se desee utilizar. Uno de los mecanismos de inferencia más ampliamente utilizados y divulgados en la actualidad es el de la *herencia de propiedades*. En este mecanismo la estructura de representación se basa en clases contenedoras que heredan a clases más pequeñas sus propiedades. Esto pone de manifiesto dos características fundamentales: organización por clases y jerarquías. Este tipo de estructuras de

representación normalmente son llamadas *estructuras de ranuras* o *redes semánticas*. En el ítem 1.8.2 se explicará con más detalle este modelo.

❖ **Conocimiento Deductivo:**

Este se fundamenta en toda la potencia de la lógica que ha sido desarrollada durante muchos años, se puede presentar en forma de *Lógica de Predicados*, *Silogismos*, entre otras. Esta forma de conocimiento esta dotada de uno de los mecanismos de inferencia más ampliamente usado y refinado con los cuales cuenta el hombre, *las reglas lógicas de inferencia*. Estas pueden ser de formas distintas, razonamiento hacia adelante, razonamiento hacia atrás, la resolución, etc., todos ellos con los mismos fundamentos lógicos y algunas características diferentes aplicables en diferentes dominios de problemas.

El hecho de que la representación del conocimiento y la inferencia esten basados en la lógica, determina que el conocimiento debe ser completo, consistente e inalterable, los caminos de inferencia deben ser determinísticos y el proceso de adquisición del conocimiento es complejo por las características propias de la Lógica.

El contexto del presente proyecto posee características tales como: Inconsistencia del conocimiento y cambios constantes del mismo de acuerdo a las condiciones en las que se dé el proceso de aprendizaje. Dichas características junto con la poca flexibilidad del conocimiento deductivo frente a ellas, hizo que se descartará este enfoque como solución a nuestro problema.

❖ **Conocimiento Procedimental u Operacional:**

Este tiene un lenguaje propio de representación, LISP. Este tipo de conocimiento se centra en el “que hacer” cuando se da una determinada situación. Desafortunadamente no aplica muy efectivamente con algunas de las características fundamentales de la representación del conocimiento como la suficiencia deductiva y eficiencia en la adquisición. Además las representaciones obtenidas en lenguajes como LISP son muy poco comprensibles, a pesar que sus resultados sean correctos, lo que lo pone en desventaja ya que en otras formas de representación las personas pueden manipular el conocimiento, no solo la maquina.

Ahora, no es difícil saber que un sistema inteligente debe estar compuesto de dos partes fundamentales: la primera, un sistema de representación de conocimiento (que, idealmente debe tener las características de conocimiento anteriormente citadas); el segundo, es un mecanismo de inferencia capaz de obtener la información de la base de conocimiento.

1.7. Inteligencia Artificial en el Proyecto

A continuación se dará una muestra de los mecanismos de representación de conocimiento concreto y motores de inferencia tenidos en cuenta para la implementación del modulo inteligente objeto de este proyecto, además de los criterios que llevaron a la escogencia o descarte de cada una de ellas para el desarrollo del proyecto[30]:[29]

1.7.1. Modelos Simbólicos

Es un mecanismo muy potente de representar conocimiento, proporciona una forma muy eficiente de obtener nuevo conocimiento a partir de existente (suficiencia deductiva) por medio de la deducción matemática. Sin extendernos mucho en la definición de la lógica de predicados y en sus múltiples usos trataremos los problemas que tiene a la hora de utilizarse en ambientes como el que nos presenta el proyecto "IA en la generación de estrategias de aprendizaje de un Sistema Tutor Inteligente- STI".[30]

Sabemos que un sistema de representación de conocimiento debe tener cuatro características importantes. La lógica de predicados es fuerte en las dos primeras gracias a la representación lógica (que es quizás una de las mas fuertes y usadas) y a la deducción matemática (que brinda la suficiencia deductiva) e incluso incorpora la tercera (eficiencia deductiva) ya que la deducción lógica parte de un hecho indiscutible y es que el conocimiento del mundo es completo, consistente e inalterable. Esta es la razón por la cual los sistemas lógicos como la lógica de predicados, la representación mediante reglas y el razonamiento simbólico incorporan las tres primeras características de un buen sistema de representación, ya que en sus bases de conocimiento solo existe información coherente, cada aserción o silogismo es verdadero por que no existen dentro de la base de conocimiento ningún otro que lo contradiga, esto hace que no existan diferentes caminos para la deducción y que siempre se tome el correcto. El problema empieza cuando dicha información del mundo no es del todo completa, consistente e inalterable. Para este proyecto es deseable que la información que se tiene de los estudiantes cambie de acuerdo a las características que estos presenten a medida que interactúen con el sistema, esto presupone dos problemas graves, en primer lugar, los sistemas lógicos no proporcionan mecanismos de adquisición de conocimiento concretos (suficiencia de la adquisición, cuarta característica de la representación del conocimiento), esta es una labor que debe hacerse casi manual y sus cambios suponen cambios en todos los procesos de inferencia, lo cual no es deseable en nuestro ambiente. La segunda es mas grave aun, la inclusión de ese nuevo conocimiento puede provocar inconsistencias en la representación del mundo que se tiene y generaría un sin fin de problemas en la inferencia ya que, como se dijo anteriormente, la

inferencia en este mecanismo parte de que el conocimiento es completo, consistente e inalterable[27].

Ahora, se podría pensar en mecanismos que al igual que la lógica de predicados tienen como base la lógica, pero que cuyos procesos de inferencia están preparados para tomar decisiones bajo incertidumbre, es decir, no necesitan del conocimiento completo, consistente e inalterable del que se hablaba anteriormente.

Tales mecanismos existen, razonamiento simbólico bajo incertidumbre, razonamiento no monótono e incluso el mismo razonamiento estadístico (que utiliza las teorías de probabilidad en su inferencia); pero al solucionar los problemas del conocimiento incompleto, inconsistente y alterable, agrega otros como la dificultad de generar indicadores que decidan los caminos de respuesta cuando exista la incertidumbre dado que no existen experiencias previas que puedan ser tomadas para realizar dichos indicadores ni tampoco se cuenta con expertos en la materia que asistan en dicha labor; además los tiempos de respuesta en la inferencia, la potencia de procesamiento en la inferencia y otros de carácter hardware al igual que su respuesta ya no tiene la certeza que tenían las anteriores. Todo esto sin nombrar el hecho que, al igual que los anteriores, no tienen mecanismos concretos de adquisición de nuevo conocimiento.

Cabe agregar que en este tipo de representación de conocimiento se incluye la Lógica Difusa como un mecanismo de razonamiento estadístico, con las mismas características de este, pero igualmente con los mismos problemas, la única distinción es la agregación de conjuntos difusos en lugar de conjuntos lógicos.

1.7.2. Modelos Conexionistas

Surgen de la idea obvia de representar directamente el modelo de la mente humana en un sistema computacional. Intentan, más que copiar el modelo de decisión del cerebro humano, utilizar algunas de las características mas conocidas de este modelo. Se caracterizan por que tienen:

- Un gran numero de elementos muy simples que procesan de forma similar a las neuronas
- Un gran numero de conexiones con “pesos” entre los elementos. Estos pesos en las conexiones codifican el conocimiento de una red.
- Control distribuido altamente paralelo
- Énfasis en el aprendizaje automático de representaciones internas.

Una diferencia importante en estos modelos radica en la representación del conocimiento, ya que el conocimiento en estos modelos se encuentra en el gran número de conexiones con valores reales que existen entre los elementos que los componen. Es precisamente esta diferencia la que genera los problemas para su

utilización en nuestro caso, ya que la representación de nuestro conocimiento en esta forma presupone un grado mayor de complejidad al incluir un proceso de aprendizaje al problema. Los modelos basados en este esquema representan el conocimiento después de un previo proceso de aprendizaje, dichos procesos son guiados por algoritmos de aprendizaje ya establecidos (aprendizaje hacia atrás, maquinas de Boltzman, aprendizaje por refuerzo, aprendizaje no supervisado, entre otros). Estos algoritmos se especializan en entrenar redes con fines muy distintos a los que se plantean en nuestro caso, por lo tanto intentar utilizar este enfoque implicaría un riesgo grande en la etapa de aprendizaje, riesgo que se acrecentaría si tomamos en cuenta que dichos procesos de aprendizaje toman mucho tiempo en realizarse, además de los grandes recursos computacionales que acarrearán y el esfuerzo que utilizan. En esos casos el conocimiento suele ser muy consistente y la información muy completa, lo que facilita el proceso, pero en nuestro caso estas características no se tienen, por lo tanto, en el peor de los casos, utilizar este enfoque podría derivar en la construcción de un nuevo algoritmo de aprendizaje que creemos sería un trabajo de otro nivel.

Otro punto en contra de estos modelos radica en que un buen sistema inteligente deben tener, según McKeyley [1993], una forma de explicar a las personas el conocimiento empleado, para que estos puedan saber como fue el mecanismo que se utiliza para tomar la decisión, en estos enfoques dicho conocimiento no sería mas que una gran cantidad de números reales que una persona no entendería, únicamente la maquina

1.7.3. Los Sistemas Expertos

Aunque los sistemas expertos hacen parte de los modelos simbólicos, tienen características diferentes que nos llevaron a tratarlos de forma individual.

Los sistemas expertos resuelven problemas que generalmente son resueltos por expertos humanos, por lo tanto necesitan de bases de conocimiento muy eficientes y que, nuevamente, deben ser consistentes, completas e inalterables (o por lo menos no de forma dinámica), y acá comienzan los problemas.

Al representar el conocimiento como un conjunto de reglas, nuevamente estamos utilizando la lógica como punto de partida. Pero el hecho de utilizar la lógica como punto de partida implica tener también sus problemas, discutidos anteriormente. Los sistemas lógicos en general no tienen suficiencia en la adquisición y los sistemas expertos no son la excepción, y esta es una característica importante dentro de nuestro problema

Además, los sistemas inteligentes de este tipo deben proveer un mecanismo para explicar a los usuarios que confían en ellos de que forma se tomo la decisión. Este proceso en el ámbito de los sistemas expertos se conoce como "Explicación" y tiene una característica importante: Nadie confiara en los resultados de un sistema

experto en el cual no se entienda el razonamiento utilizado para llegar a una respuesta. Imaginen un sistema experto en el área de la medicina que diagnostique una enfermedad terminal, el paciente no creería a menos que su procedimiento de razonamiento este validado por un doctor especialista en el área, por lo tanto el procedimiento de la toma de decisiones debe estar claro y debe ser entendido por el usuario.

Este punto sería realmente difícil de lograr en nuestro ambiente ya que trataríamos de explicar que se tomó una decisión de forma experta sin un experto, es decir, tenemos un sistema experto en una área en donde no tenemos un experto humano que pueda, además de crear la base de conocimiento, validar los resultados del sistema, y es bien sabido que la validación de los resultados constituye una importante etapa de los procesos de Ingeniería de Software y de Ingeniería de Conocimiento. Además, las características de suficiencia en la adquisición de nuevo estarían ligadas a un experto que las provea, con el cual no contamos y la construcción de una base de conocimiento sería una tarea bastante “difusa” (por emplear la jerga) ya que sin un experto tendríamos dicha tarea nosotros y cabe anotar la importancia que tiene esta en los sistemas expertos, ya que si la base de conocimiento no es lo suficientemente completa el sistema no sería eficaz.

1.8. Conceptos de IA utilizados en el Modelo de Toma de decisiones.

A continuación se presentan los conceptos de IA que hacen parte del modelo de toma de decisiones desarrollado en este proyecto.

1.8.1. Introducción

Una vez recorridos y estudiados en su parte teórica algunos de los enfoques de IA más importantes y de mayor utilización práctica, el proceso investigativo que buscaba este proyecto se centró en un pequeño grupo de ellos que dadas las características del proyecto eran los que más se acercaban a satisfacerlas.

Se tenía la idea clara que una aplicación inteligente debe contar con dos partes fundamentales, una base de conocimiento (que represente alguno de los tipos de conocimiento anteriormente nombrados) y un mecanismo de inferencia que obtenga información de dicha base de conocimiento de forma efectiva y que está sea válida y corresponda al conocimiento almacenado. Ver Figura 4



Figura 4. Diagrama de Bloques de una aplicación Inteligente

En ese orden de ideas y teniendo en cuenta el proceso de estudio que permitió descartar varios de los enfoques de IA, se centraron todos los esfuerzos en un pequeño grupo de mecanismos de representación de conocimiento que a su vez presentan sus propios mecanismos de inferencia.

1.8.2. Conocimiento Heredable y Sistema de Ranuras

Fue el primer enfoque que dio luz al proyecto. Fruto de la investigación se dedujo que era este enfoque la forma más sencilla y efectiva de representar el conocimiento acerca del estudiante que debe tener el sistema para que posteriormente, basándose en ese conocimiento, tomar una decisión acerca de la estrategia de aprendizaje más adecuada para dicho estudiante.

Para entender un poco este enfoque, es necesario hablar de su predecesor, un enfoque más sencillo llamado *Conocimiento Relacional*. Este tipo de conocimiento, basa su representación en atributos y sus respectivos valores acerca de un objeto. Este tipo de conocimiento es el que se encuentra en las bases de datos relacionales. Los problemas que éste tiene no se enfocan específicamente en la representación del conocimiento ya que ella en sí ofrece excelentes ventajas en proyectos de estas características como se verá más adelante. Sus problemas se relacionan con los mecanismos de inferencia que se utilizan, es decir, estos tienen una escasa capacidad deductiva razón por la cual no se tomó en cuenta en principio.

La representación de conocimiento heredable se basa en la teoría del conocimiento relacional adicionando una característica importante y es la extensión de su riqueza semántica agregando una estructura a sus atributos y generando la posibilidad de utilizar mecanismos que operen sobre dicha estructura para mejorar la inferencia sobre la información. Estos mecanismos se encuentran basados en la *herencia* clásica que conocemos del paradigma de orientación a objetos.

En principio, se pensó que representando la información del estudiante por medio de estructuras de ranuras (más exactamente *frames* [34]) que tienen características como estructuración en clases y jerarquías, dando así soporte a la herencia que en principio sería de atributos; se lograría generar estrategias de aprendizaje partiendo del perfil del estudiante representado en forma de jerarquías de clases. Pero pronto surgirían algunos problemas relacionados especialmente con la inferencia de información a través de la herencia y con la suficiencia en la adquisición de los modelos de ranuras.

La herencia es un mecanismo de inferencia muy potente cuando la información es estable, por ejemplo, cuando se define una clase *ave* con un atributo que indica que el *ave* *tiene_plumas* y por medio de relaciones del tipo *es_un* se define otra clase llamada *loro* que herede de la clase *ave*; fácilmente y por medio de herencia de atributos inferimos que el loro tiene plumas y ese razonamiento acerca del conocimiento del loro va a permanecer inmutable, es decir, siempre se sabrá que el loro tiene plumas. Cuando se habla de modelar a personas no existe información estable o exacta. Así como un estudiante con un perfil determinado por el conjunto de atributos heredados de varias clases “padre” se le debe aplicar una estrategia de aprendizaje particular (asumiendo que esta daría los mejores resultados) es bastante probable que otro estudiante con el mismo perfil utilice otra estrategia que obtenga los mismos resultados o incluso mejores; o lo que sería peor, que utilizando la misma estrategia se obtengan resultados negativos.

En lo que se refiere a la suficiencia en la adquisición se descubrió que dada esa misma inestabilidad en la información resultaría bastante difícil incluir nueva información de los estudiantes a la base de conocimiento, primero, por que el problema de ubicarla dentro de la estructura de red sería bastante dispendioso y segundo por que podría fácilmente contradecir la información que ya se tiene dentro del sistema. Este problema es conocido a nivel general como “el problema del frame” (The Frame Problem) y ha sido tratado por diversos investigadores durante muchos años.

Encontrados estos problemas se pensó en un mecanismo que diera mejores resultados. Se retomo el proceso dejado en la etapa anterior y se dedujo que si se utiliza el conocimiento relacional se eliminaría este último problema, es decir, el de la suficiencia en la adquisición; pero aun se tendría que solucionar el de la inferencia de la información. Ahora, para entender más a fondo como se solucionaron estos problemas se debe mirar más específicamente este tipo de representación.

1.8.3. Conocimiento Relacional

Es una forma de conocimiento declarativo [28] lo que hace que sea sencillo y practico a la hora de aplicarlo a problemas reales como este. Como ya se había

hablado anteriormente, tiene problemas de capacidad deductiva, es decir que los procesos de inferencia que se realizan sobre dicho conocimiento son muy débiles a la hora de obtener información. Este era el panorama que se tenía acerca del conocimiento relacional; pero diversas investigaciones sobre éste se estaban llevando a cabo, y estas abrirían nuevos caminos en pro de encontrar una solución.

Un mecanismo de análisis de información muy importante se ha venido desarrollando en los últimos años, el *DataMining*[35], con la aparición de este, se aumentaron características de inferencia de información a estos tipos de sistemas de representación de conocimiento ya que actúan sobre los datos que se encuentran en las tablas relacionales e infieren diferentes tipos de información de acuerdo a la naturaleza de la aplicación.

Este desarrollo dejó al descubierto que la representación del conocimiento en forma relacional es más potente de lo que se creía, pues dio cabida a nuevos y diversos mecanismos de inferencia que anteriormente no se tenían a disposición.

Representar la información relacionalmente resulta bastante bueno para los objetivos de este proyecto, ya que se cuenta con la información necesaria en la base de datos que almacena toda la información del proyecto, por lo tanto es solo cuestión de organizarla de forma tal que se de soporte a algún tipo de inferencia sobre dicha información y además se proporcionan ventajas que contrastan con las características de los sistemas de representación:

- ✓ Operaciones bien definidas de la teoría relacional (proyección, selección y unión)
- ✓ Facilidad en la construcción de un lenguaje de consulta (SQL)
- ✓ Facilidad en la inclusión de nuevos hechos (suficiencia en la adquisición).

La representación de la información, entonces, se presentaría de la siguiente forma:

Jugador	Edad	Altura	Peso	Goles
Pablo Aimar	22	1,75	75	7
Mendieta	27	1,70	65	5
Cañizares	29	1,85	80	0
John Carew	20	1,90	84	5

Figura 5. Representación de Conocimiento Relacional

Como se puede notar, no es más que un conjunto de atributos (Jugador, Edad, Altura, Peso, Goles) y los respectivos valores para cada atributo (Pablo Aimar, 22, 1.75, 75, 7), y aunque en una primera mirada no parezca existir mucho conocimiento en tablas simples como las de la Figura 5 y mucho menos parece que exista mucho que inferir de dicha información resulta cierto que hay grandes cantidades de información que se pueden obtener de la representación relacional con aplicar mecanismos de inferencia y algoritmos inteligentes que operen sobre esta información.

1.8.4. Mecanismo de Inferencia

Los árboles de decisión fueron la respuesta al problema de la inferencia sobre la información que se tenía al representar el conocimiento de forma relacional. Aunque se debe aclarar que no son propiamente mecanismos de inferencia dentro del ámbito de la IA si no que son medios de clasificación, acá se utilizaran como la base del proceso de inferencia de lo cual no se encontró ningún antecedente lo que da un valor agregado al proyecto. Primero, se verán las características generales de los Árboles de Decisión y posteriormente se muestra la forma como se integraron con el conocimiento relacional para generar información sobre la cual se puedan tomar decisiones.

1.8.5. Árboles de Decisión

Los árboles de decisión constituyen probablemente el modelo de clasificación más popular y utilizado. El conocimiento obtenido durante el proceso de aprendizaje inductivo, es decir, conjuntos de aprendizaje, se representa mediante un árbol en el cual cada nodo interno contiene una pregunta acerca de un atributo particular (con un nodo hijo para cada posible respuesta) y en el que cada hoja se refiere a una decisión (etiquetada con una de las respuestas al problema).

Un árbol de decisión puede utilizarse para clasificar un ejemplo concreto comenzando en su raíz y siguiendo el camino determinado por las respuestas a las preguntas de los nodos internos hasta que se llega a una hoja del árbol. Su funcionamiento es análogo a los controles de los rieles de un ferrocarril: cada caso es dirigido hacia una u otra rama de acuerdo con los valores de sus atributos al igual que los trenes cambian de vía según su destino (las hojas del árbol) en función de la posición de las agujas de la red de rieles de los ferrocarriles (los nodos internos).

Los árboles de clasificación son útiles siempre que los ejemplos a partir de los que se desea aprender se puedan representar mediante un conjunto prefijado de atributos y valores, ya sean estos discretos o continuos. Sin embargo, no resultan adecuados cuando la estructura de los ejemplos es variable, es decir, que existen

diferentes tipos de conjuntos de ejemplos de los cuales se debe generar el árbol. Tampoco están especialmente indicados para tratar con información incompleta (cuando aparecen valores desconocidos en algunos atributos de los casos de entrenamiento) y pueden resultar problemáticos cuando existen dependencias funcionales en los datos del conjunto de entrenamiento (cuando unos atributos son función de otros).

La construcción de un árbol de decisión a partir del conjunto de datos de entrada (ejemplos o experiencias) se suele realizar de forma descendente mediante algoritmos greedy[36] de una eficiencia del orden $O(n \log n)$, siendo n el número de ejemplos incluidos en el conjunto de entrenamiento.

Los árboles de decisión se construyen recursivamente siguiendo una estrategia descendente, desde conceptos generales hasta ejemplos particulares. Esa es la razón por la cual el acrónimo TDIDT, que proviene de “Top-Down Induction on Decision Trees”, se emplea para hacer referencia a la familia de algoritmos de construcción de árboles de decisión.

Una vez que se han reunido los datos que se utilizarán como base del conjunto de entrenamiento, se descartan a priori aquellos atributos que sean irrelevantes utilizando algún método de selección de características y, finalmente, se construye el árbol de decisión recursivamente. El método de construcción de árboles de decisión mediante particionamiento recursivo del conjunto de casos de entrenamiento tiene su origen en el trabajo de Hunt[37] a finales de los años 50.

La información que se obtiene de los árboles de decisión es el fruto de la combinación de varias teorías en diversos campos de la ciencia. La idea que tienen todos los algoritmos de la generación TDIDT es la vieja premisa de la informática, y en general, de la ingeniería: “Divide y Vencerás”. Es más, el primer grupo de algoritmos fueron llamados así, y esto se debe a la filosofía que tienen. El proceso que realizan estos algoritmos es, a grandes rasgos, el siguiente. Parten de un conjunto de ejemplos (que llamaremos experiencias) y un conjunto de atributos los cuales tienen un conjunto discreto de valores que pueden tomar, por ejemplo, el atributo *color_de_piel* va tomar únicamente valores como *blanco*, *negro*, *mestizo*, *amarillo*; con esta información y apoyados en unas reglas de división que generalmente son heurísticas adaptadas al contexto y que se estudiarán más adelante, se va eligiendo dentro de los atributos el que mejor divide la información de las experiencias. Cuando se tiene dicho atributo se generan nodos hijos para cada posible valor que pueda tomar este atributo y recursivamente se van repitiendo el proceso hasta que finalmente en las hojas del árbol van quedando los valores resultado que se desean predecir.

Es importante anotar que gran parte del éxito del algoritmo depende de la capacidad de la regla de división que se utilice, por ello se pueden observar

algunas de las reglas de división mas importantes y su sustentación matemática en [38]. Solo se presentara la utilizada en este proyecto.

1.8.6. Reglas de División

Cualquier pregunta que divida el conjunto de experiencias en al menos dos subconjuntos no vacíos conducirá a la construcción de un árbol de decisión. No obstante, el objetivo del proceso de construcción de árboles de decisión es obtener un árbol que revele información interesante a la hora de realizar predicciones o clasificaciones. Cada posible pregunta ha de evaluarse mediante alguna heurística y, dado que los algoritmos suelen ser Greedy, ésta desempeña un papel esencial en la construcción del árbol.

Las heurísticas estadísticas usadas intentan favorecer las divisiones que mejor discriminan unos atributos de otros. Ejemplos muy conocidos de estas heurísticas son la ganancia de información usada por ID3 (que se muestra a continuación), el criterio de proporción de ganancia de C4.5 [39] o el índice de diversidad de *Gini* empleado en CART [37]

1.8.7. Ganancia de Información - Entropía

El algoritmo ID3 intenta maximizar la ganancia de información conseguida por el uso del atributo A_i para ramificar el árbol de decisión mediante la minimización de la función I:

$$I(A_i) = \sum_{j=1}^{M_i} p(A_{ij})H(C|A_{ij})$$

donde A_i es el atributo utilizado para ramificar el árbol, M_i es el número de valores diferentes del atributo A_i , $p(A_{ij})$ es la probabilidad de que el atributo A_i tome su j -ésimo valor y $H(C|A_{ij})$ es la entropía de clasificación del conjunto de ejemplos en los que el atributo A_i toma su j -ésimo valor. Esta entropía de clasificación se define como:

$$H(C|A_{ij}) = - \sum_{k=1}^J p(C_k|A_{ij}) \log_2 p(C_k|A_{ij})$$

siendo J el número de atributos del problema y $p(C_k|A_{ij})$ una estimación de la probabilidad de que un ejemplo pertenezca a la clase C_k cuando su atributo A_i toma su j -ésimo valor. En realidad, la estimación de la probabilidad $p(C_k|A_{ij})$ no es más que la frecuencia relativa $f(C_k|A_{ij})$ en el conjunto de entrenamiento utilizado.

La probabilidad de que un caso escogido aleatoriamente pertenezca a la clase C_k es $p(C_k)$ y la información que se obtiene es $\log_2 p$. La información que esperamos obtener al clasificar un caso cualquiera del conjunto de datos de entrenamiento sería igual a:

$$\sum p(C_k) \log_2 p(C_k)$$

cantidad a la que se denomina entropía del conjunto.

La información necesaria para transmitir la división del conjunto de casos de entrenamiento T en M_i subconjuntos T_j es igual a:

$$\sum p(T_j) H(T_j)$$

donde $p(T_j)$ es la probabilidad de que un ejemplo pertenezca al subconjunto T_j y $H(T_j)$ es la entropía de clasificación del conjunto T_j .

La ganancia de información que se produce al dividir T en los subconjuntos T_j es igual a:

$$H(T) - \sum p(T_j) H(T_j)$$

donde $H(T)$ es la entropía de T . Para comparar las posibles particiones del conjunto T se evalúa la ganancia de información obtenida por cada una de ellas. Al ser $H(T)$ constante, nos basta con comparar el valor de la expresión:

$$-\sum p(T_j) H(T_j)$$

Esta heurística suele favorecer la construcción de árboles de decisión con un grado de ramificación elevado, hecho que resulta interesante en el desarrollo de este proyecto.

1.8.8. Algoritmo ID3

Fue desarrollado por J. Ross Quinlan y recibe su nombre como acrónimo de Iterative Dicotomiser 3 – ID3. Es uno de los más utilizados y se encuentra ampliamente implementado en software comercial y de investigación.

Aunque el algoritmo se desarrolló inicialmente para clasificar movimientos en ajedrez, con atributos categóricos (que podían tomar sólo valores discretos de un conjunto finito) y dos clases, la mayoría de las implementaciones actuales permiten atributos con valores enteros o reales, y clasificación en más de dos clases.

La entrada del algoritmo, un conjunto de casos ejemplo (experiencias) en el cual debe haber, para aumentar la precisión del algoritmo, representantes de todos los atributos manejados por el algoritmo. El conjunto es analizado, el cual entrega las reglas inducidas representadas en forma de un árbol n-ario.

Como se ha insinuado antes, la representatividad del conjunto de entrenamiento afecta la calidad de las reglas inducidas, es decir, el algoritmo es incapaz de descubrir algo que no esté allí. Por esto, es indispensable la buena selección de atributos y ejemplos. La selección de atributos debe basarse en el conocimiento acumulado por la experiencia y los ejemplos deben reflejar la experiencia adquirida por el sistema.

Los atributos son los factores que influyen la clasificación o decisión. Como se dijo anteriormente, en este algoritmo cada atributo forma un nodo intermedio en un árbol cuyas hojas o nodos terminales son los atributos de decisión. Dado el conjunto de ejemplos, el ID3 selecciona el atributo que subdivide los ejemplos "de la mejor manera", utilizando para ello la heurística de ganancia de información o entropía. Un atributo con N posibles valores subdividiría los ejemplos en N subconjuntos, y será "*el mejor atributo*" para la próxima regla si cada uno de los N subconjuntos contiene elementos de sólo una clase de atributo; sería un mal atributo a seleccionar si los subconjuntos poseen elementos de muchas clases. En el caso de atributos con valores numéricos se requiere escoger valores de corte (umbrales) de manera óptima para la subdivisión, por ejemplo "altura mayor de 1.50" divide los ejemplos en 2 subconjuntos. Como criterio para medir el grado de mezcla de las clases de atributos en los subconjuntos, el ID3 utiliza una medida de información, basándose en la famosa teoría estadística de la información de Claude Shannon [39]: "selecciona el atributo que produce la mayor disminución de la información (en el sentido de Shannon) en los subconjuntos, es decir, escoge el atributo que al partir el conjunto de ejemplos genere la mayor riqueza semántica en los conjuntos resultantes".

Si los atributos se terminan antes de completar el árbol, entonces hay ejemplos contradictorios en el conjunto de entrenamiento, o sea, ejemplos con iguales valores de atributos, pero de resultados diferentes. El algoritmo básico original no está diseñado para esto, y es necesario introducir un atributo extra para distinguir entre ejemplos contradictorios. En otras situaciones el proceso puede revelar carencias en el conjunto de entrenamiento, al identificar un nodo terminal en el árbol inducido, para el cual no hay una experiencia o ejemplo, o sea que no existe un conjunto de valores de atributos con ese resultado. La estructura del algoritmo ID3 y su pseudocódigo se pueden ver en el ANEXO 31.

La escogencia de este atributo se basó fundamentalmente en dos aspectos principales, el algoritmo ID3 y su criterio de ganancia de información favorecen enormemente la generación de árboles de decisión muy ramificados, esto quiere decir que este criterio de división ayuda a que en los árboles de decisión que se crean a partir de él exista mayor cantidad de valores de atributos y esta era un

buen escenario para este proyecto. Además, la implementación práctica y experimentación que se encontró de este algoritmo lo mostraban como una excelente opción, ya que sus evoluciones no presentaban esa misma cantidad de experimentación en el ámbito mundial y las ventajas que se presentaban no compensaban el grado de dificultad adicional que representaba su implementación dado que se planteaba un modelo acorde a las características del algoritmo ID3

CAPITULO 2. EL MODELO DE TOMA DE DECISIONES DEL MODULO INTELIGENTE.

Una vez estudiados los componentes del modelo por separado, se mostrara como convergen para generar un modelo que permite realizar el proceso de toma de decisiones del presente proyecto.

2.1. Generalidades

En el capítulo 1 se dijo que una aplicación inteligente se compone de dos partes fundamentales, el mecanismo de representación de conocimiento y un mecanismo de inferencia. Se dijo además que el mecanismo de representación de conocimiento era el Relacional y que el mecanismo de inferencia se basa en los árboles de decisiones. En este capítulo mostraremos paso a paso como se generaran las estrategias de aprendizaje del modelo de toma de decisiones y el recorrido que se debe hacer para que partiendo del perfil del estudiante se genere una estrategia de aprendizaje adecuada. Ver Figura 6

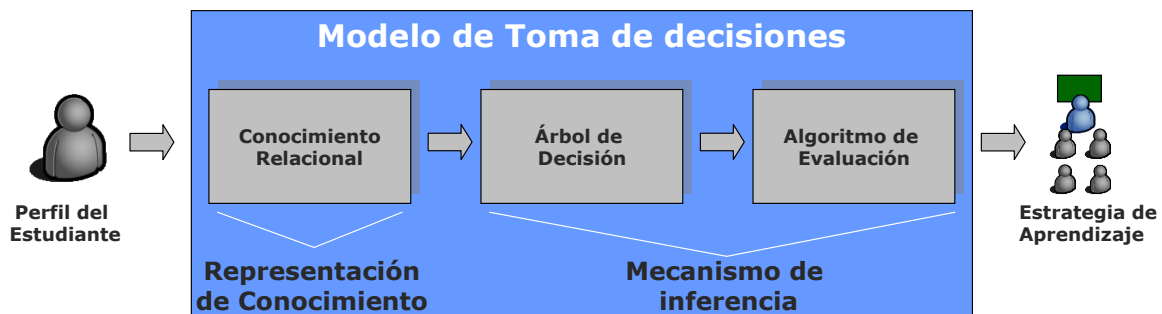


Figura 6. Modelo general de toma de decisiones

2.2. Mecanismo de Representación de conocimiento

El modelo de toma de decisiones parte de un conocimiento esencial que es la materia prima de todo el modelo, el perfil del estudiante. El perfil del estudiante contiene diversa información que hace parte del conocimiento que el sistema tiene del estudiante. El modelo es totalmente abierto a recibir como entrada cualquier tipo de información que el sistema (en su modulo interfaz) pueda obtener del estudiante, desde atributos que definan el estudiante dentro de algún sistema de clasificación psicológico como CHAEA y su respectivo estilo de aprendizaje hasta atributos que definan los pasatiempos del estudiante. Esto quiere decir que el

modelo es completamente dinámico en el sentido que se puede utilizar gran variedad de atributos con el fin de definir con mayor exactitud cada experiencia del estudiante dentro del sistema permitiendo que la evaluación se realice con mayor variedad y generando resultados más significativos y personalizados a cada estudiante en particular.

La representación de la información del estudiante se encuentra dividida en dos tipos de atributos diferentes, los *Atributos Estudiante* y los *Atributos Sistema*.

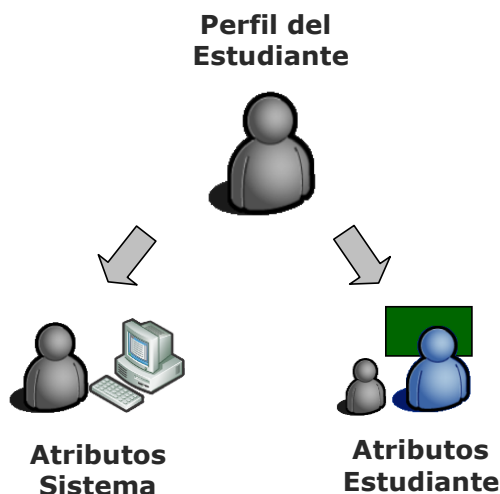


Figura 7. Representación del perfil del estudiante

Los *Atributos Sistema* se relacionan con las características que el sistema ha presentado al estudiante durante el desarrollo del curso o determinada parte de él (no se puede olvidar que la estrategia de aprendizaje puede cambiar durante el curso si el estudiante o el profesor creen que la actual no es la correcta). Estas características son opciones que tiene el sistema de presentar al estudiante de formas diferentes los contenidos, siempre con la premisa de buscar la opción que más favorezca al perfil del estudiante. En este grupo encontramos atributos como: la estructura del árbol de contenidos, los tipos de actividades que se le presentan, el diseño instruccional del curso y un atributo creado a partir de los estilos de aprendizaje llamado *estrategia*, esta estrategia puede ser apoyar el estilo (como lo plantea la teoría de *styles learning*[14]), ir en contra del estilo (como dice el modelo de Felder) o buscar un balance entre el estilo propio y el contrario a la cual se llamara *nivelar*.

Los *Atributos Estudiante* son los atributos que describen la información propia del estudiante y se relacionan con el perfil inicial que tiene el sistema de él; son atributos que describen sus características propias de aprendizaje, no solo en ambientes virtuales si no también presénciales. En este grupo se encuentran atributos como: Modelo de clasificación, estilo de aprendizaje en el cual fue clasificado, nivel de aceptación del estilo en el que se encuentra clasificado,

detalles de sus pasatiempos o áreas de interés, entre otros; los cuales corresponden a las características propias que tiene el estudiante y que, por tanto, no pueden ser cambiadas por el sistema (solo cambian por iniciativa intencional del mismo estudiante) y que deben ser tenidas en cuenta al momento de tomar una decisión acerca de la estrategia de aprendizaje mas apropiada para él.

La representación de conocimiento relacional proporcionará además la fuente de representación de las experiencias que servirán de entrenamiento al modulo inteligente, estas serán tuplas con uno de los valores posibles de cada atributo definido y su atributo resultado, que tomaremos como un atributo que resuma si la experiencia, o sea el conjunto de valores de los atributos en el sistema, fue satisfactoria o no (un valor booleano).

El conocimiento relacional será la fuente de información de los atributos, los valores que pueden tomar cada atributo y el conjunto de ejemplos o experiencias que harán las veces de entrada al algoritmo ID3 para que forme un árbol de decisión. Este se convertirá en la materia prima sobre la cual se puedan tomar decisiones acerca de que valores de los atributos son mas convenientes para obtener los mejores resultados en una estrategia de aprendizaje, basando la decisión en el atributo a calcular (o atributo resultado)

2.3. Mecanismo de Inferencia

El proceso de inferencia sobre el conocimiento representado de forma relacional se realiza en base a un árbol de decisión y un algoritmo de evaluación que llamaremos *deliberar*⁷. Para ver el codigo del algoritmo ver ANEXO 35

⁷ El nombre resulta representativo ya que el sistema penal acusatorio basa su veredictos en la comparación del caso a resolver con casos iguales o parecidos que se tengan en la experiencia, y esto representa la filosofía del modulo de toma de decisiones

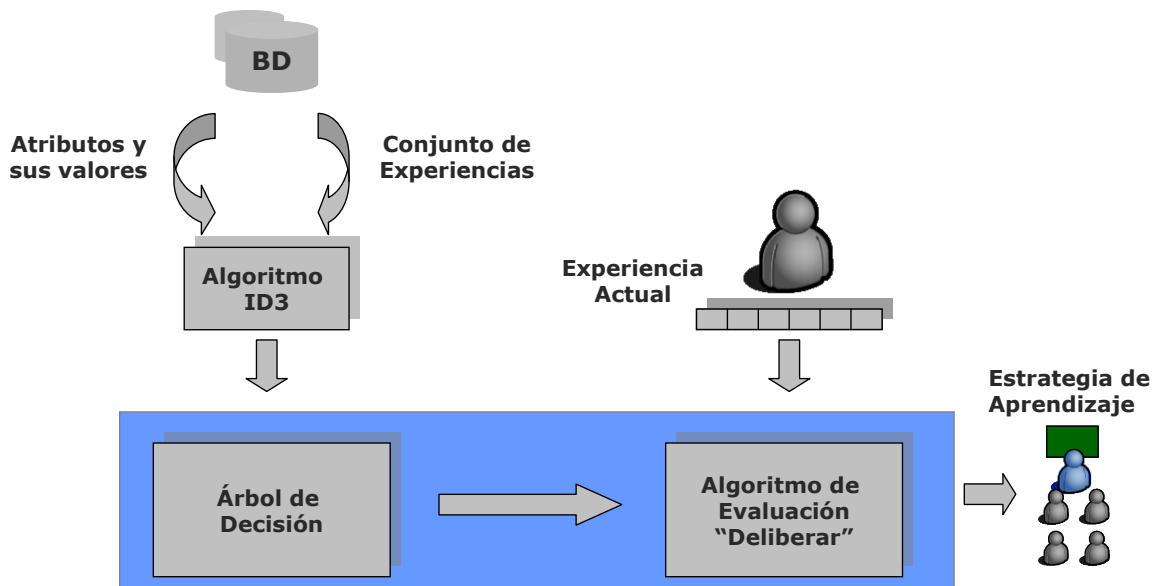


Figura 8. Mecanismo de Inferencia

2.3.1. El Árbol de Decisión

Anteriormente se ha explicado el proceso de construcción de árbol de decisión, por lo tanto solo se mostrara su importancia dentro del proceso de inferencia.

A partir de la base de datos se extrae en conjunto de atributos y sus respectivos valores al igual que el conjunto de experiencias que sirven como "entrenamiento" para que el algoritmo ID3 genere el árbol de decisión que es el eje fundamental del proceso de inferencia, ya que es el encargado de analizar a partir de la regla de división incorporada en su algoritmo de construcción las experiencias ingresadas para generar un árbol que agrupe la información mas significativa de dichas experiencias y que este pueda ser usado para evaluar una nueva experiencia y tomar una decisión. Este ultimo proceso es el que desarrolla el algoritmo Deliberar, que recibe el árbol de decisión (que no es mas que la experiencia sobre la cual se va a tomar una decisión) y lo que llamamos la experiencia actual que recoge los valores de los atributos que el estudiante tiene actualmente y que son evaluados sobre las rutas del árbol con el fin de encontrar los valores de los Atributos Sistema que mejores resultados generan al acompañar los valores de los Atributos Estudiante que propone la experiencia actual.

2.3.2. El algoritmo Deliberar

A esta altura se encuentran definidos y explicados todos los componentes del modelo de toma de decisiones excepto este algoritmo que realiza el último paso y que define los valores de los Atributos Sistema sobre los cuales se debe aplicar la

estrategia de aprendizaje al estudiante. Por este motivo se presenta una explicación de este y se mostrara su pseudocódigo en los anexos.

Una vez se tiene el árbol de decisión y al estudiante al cual se desea generar su estrategia de aprendizaje el algoritmo “deliberar” lo que hace es ubicar las rutas que generaron experiencias positivas dentro del árbol de decisión y tomar el conjunto de valores de los atributos que dieron tal resultado, para posteriormente separar los valores en atributos Estudiante y atributos Sistema. Con los atributos del estudiante realiza una comparación con los valores de estos atributos en el estudiante a procesar con el fin de encontrar la misma combinación⁸, cuando esto se logre se devolverá el conjunto de valores de los atributos del sistema que acompañaban a esa experiencia en la base de conocimiento, entendiendo que si la experiencia con ese perfil del estudiante (representado por los valores de los atributos del estudiante) fue satisfactoria anteriormente con esos valores de atributos del sistema tiene gran probabilidad de tenerla nuevamente.

2.4. Una Ilustración del modelo de toma de decisiones

A manera de ejemplo se muestra el primer prototipo que se realizo del modelo aplicado, entendiendo que sirvió para dar claridad y corregir algunos detalles del modelo general que se presenta en este capítulo.



Figura 9. Ejemplos de Atributos

Para tal fin, se tomará un ejemplo de predicción climática. Definiremos los atributos mostrados en la Figura 9 con sus respectivos valores:

El conjunto de ejemplos o experiencias quedaría definido como un grupo con $n + 1$ valores de atributos, donde n es el numero de atributos totales a tener en cuenta en la experiencia y el campo agregado corresponde a un resultado que para el ejemplo se tomará como un valor booleano que represente si el día fue bueno o malo.

⁸ En el caso de no encontrar la misma combinación, se buscará el caso en el que el mayor numero de atributos del estudiante coincidan

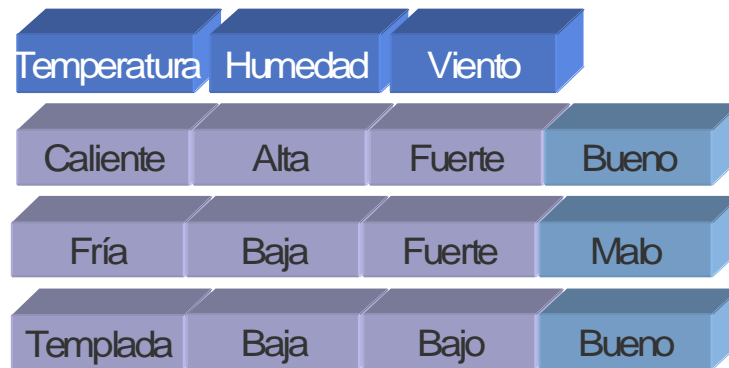


Figura 10. Conjunto de experiencias

Lo que un grupo de ejemplo representa es la idea general obtenida de una experiencia anterior que, en palabras, dice que una temperatura *Caliente* sumada a una humedad *alta* y un viento *fuerte* resultan en un día *bueno*. Es obvio que para el ejemplo estas serían casi impredecibles pero es posible que en la mayoría de los casos en donde los atributos tengan estos valores el día resulte bueno, y eso es precisamente lo que el algoritmo obtiene – un árbol donde las rutas constituyen el camino que más comúnmente se encuentre representado en el conjunto de entrenamiento. Esta característica es la que hace posible utilizarlo en el ámbito de este proyecto.

Nótese que los datos se siguen representando en forma relacional y es de ahí que obtendremos la información que resulte de las experiencias que anteriormente se hayan tenido. Es decir, en el ejemplo supongamos que existe un sistema que monitorea los valores correspondientes a cada atributo y los guarda en una tabla relacional junto con la experiencia que se obtuvo ese día, o sea, si el día fue bueno o malo.

Con esa información y el conjunto de experiencias se podría generar por medio del algoritmo ID3 un árbol de decisión que para días futuros pronostique como va a presentarse el día, así:

Comienza con el conjunto de experiencias organizado de la siguiente forma:

	Temperatura	Humedad	Viento	Resultado
1	Caliente	Alta	Fuerte	Bueno
2	Fría	Baja	Fuerte	Malo
3	Templada	Baja	Bajo	Bueno
4	Fría	Media	Moderado	Bueno
5	Fría	Baja	Bajo	Malo
6	Templada	Media	Moderado	Bueno
7	Caliente	Baja	Fuerte	Malo
8	Fría	Baja	Fuerte	Malo
9	Caliente	Alta	Moderado	Bueno
10	Fría	Media	Bajo	Bueno

Tabla 1. Conjunto de experiencias inicial

Elige uno por uno cada atributo y calcula la entropía general, la entropía por cada posible valor del atributo y la ganancia de información para el atributo, así:

$$\begin{aligned} \text{Entropía (Temperatura)} &= 0.8631 \\ \text{Entropía (Temperatura Fría)} &= 0.6365 \\ \text{Entropía (Temperatura Caliente)} &= 0 \\ \text{Entropía (Temperatura Templada)} &= 0.5711 \\ \text{Ganancia de Información (Temperatura)} &= 0.456895 \end{aligned}$$

$$\begin{aligned} \text{Entropía (Humedad)} &= 0.8631 \\ \text{Entropía (Humedad Alta)} &= 0.636514 \\ \text{Entropía (Humedad Media)} &= 0.2 \\ \text{Entropía (Humedad Baja)} &= 0.24561 \\ \text{Ganancia de Información (Humedad)} &= 0.26547 \end{aligned}$$

$$\begin{aligned} \text{Entropía (Viento)} &= 0.8631 \\ \text{Entropía (Viento Fuerte)} &= 0.5485 \\ \text{Entropía (Viento Moderado)} &= 0 \\ \text{Entropía (Viento Bajo)} &= 0.21254 \\ \text{Ganancia de Información (Viento)} &= 0.36985 \end{aligned}$$

Como la mayor ganancia de información se obtiene con el atributo *Temperatura*, y la mayor entropía se presenta cuando el valor es *frió* el algoritmo escoge este atributo como raíz del árbol de decisión

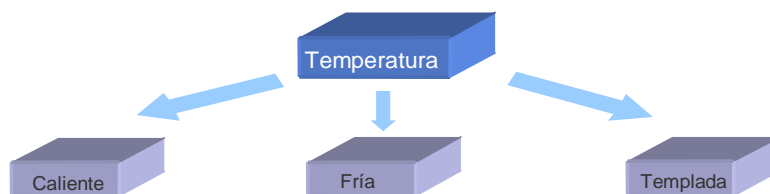


Figura 11. Árbol generado después de la primera recursión

Posteriormente elimina del conjunto de experiencias la columna de la temperatura y las filas correspondientes a los valores de menor entropía, así:

	Temperatura	Humedad	Viento	Resultado
1	Caliente	Alta	Fuerte	Bueno
2	Fría	Baja	Fuerte	Malo
3	Templada	Baja	Bajo	Bueno
4	Fría	Media	Moderado	Bueno
5	Fría	Baja	Bajo	Malo
6	Templada	Media	Moderado	Bueno
7	Caliente	Baja	Fuerte	Malo
8	Fría	Baja	Fuerte	Malo
9	Caliente	Alta	Moderado	Bueno
10	Fría	Media	Bajo	Bueno

Tabla 2. Modificaciones al conjunto de experiencias después de la primera recursión

Dejando el conjunto de experiencias de la siguiente manera:

	Humedad	Viento	Resultado
2	Baja	Fuerte	Malo
4	Media	Moderado	Bueno
5	Baja	Bajo	Malo
8	Baja	Fuerte	Malo
10	Media	Bajo	Bueno

Tabla 3. Conjunto de experiencias para iniciar la segunda recursión

Con este nuevo conjunto de experiencias y eliminando del conjunto de atributos el atributo temperatura genera una segunda recursión en el algoritmo utilizando el mismo procedimiento descrito hasta alcanzar los nodos hojas con los valores del atributo resultado.

De esta forma y después de $n \text{ Log}(n)$ (donde n es el número de experiencias) cantidad de recursiones el algoritmo genera un árbol de decisión como el que se muestra en la Figura 12

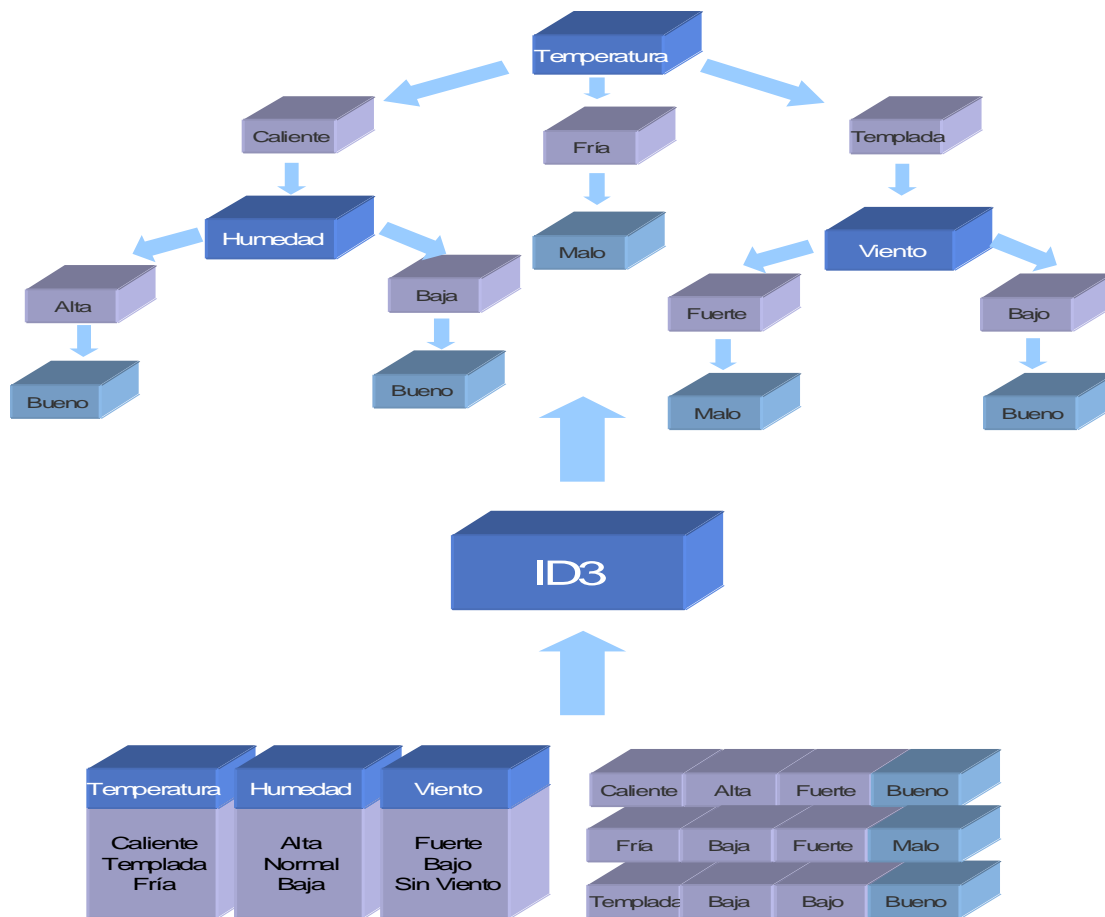


Figura 12. Modelo de creación de un árbol de decisión

Ahora, teniendo el árbol tomar una decisión sería cuestión de evaluar un nuevo conjunto de atributos con las características de un día cualquiera a través del árbol para predecir como va a presentarse ese día.

El modulo inteligente del Sistema Tutor Inteligente que se utilizara para generar estrategias de aprendizaje para los estudiantes parte del modelo descrito en este capitulo, con algunas variaciones obvias que se mostraran en el siguiente capitulo, donde se adaptan todos los mecanismos al contexto propio del proyecto, es decir, los Sistemas Tutoriales Inteligentes.

2.5. Modelo Final vs Tecnologías existentes en el ambito de la IA y los Sistemas de Toma de decisiones - DSS

En el desarrollo de este proyecto se partió del contexto en el que esta inmerso, es decir el aprendizaje en línea (más específicamente los Sistemas Tutores Inteligentes) y se tenía la idea clara de utilizar IA para generar estrategias de aprendizaje partiendo del conocimiento almacenado de los estudiantes. En el desarrollo del proyecto, y fruto de la investigación se fueron desarrollando ideas que fueran en pro de lograr ese objetivo general y se tomaron decisiones para facilitar el desarrollo.

Cuando el modelo se desarrollo surgieron preguntas acerca del contexto en el cual se podría aplicar y si tecnológicamente existe algo parecido que ya haya sido desarrollado. Con el objeto de aclarar un poco las diferencias y similitudes con algunas de las tecnologías y desarrollos que se han hecho en estas áreas presentaremos las características de algunos de ellos y como se relacionan con el modelo desarrollado.

2.5.1. Bodegas de Datos - DataWarehouse

Las bodegas de datos, según W. H. Inmon, son una colección de datos integrados, orientados a temas, que dan soporte a las funcionalidades del DSS, donde cada unidad de dato es relevante en algún momento en el tiempo. El diseño y construcción de una Bodega de Datos o DataWarehouse, es de gran importancia en el análisis de datos manejados en un ambiente determinado [51]. Este mismo autor define cuatro características mínimas que debe tener una bodega de datos. Estas son: **Orientadas a temas**, significa que para la construcción de una bodega de datos se seleccionan áreas importantes del negocio; **Integradas**, Porque los datos que contiene, a pesar de que vienen de diferentes fuentes de datos operacionales, están en un formato consistente; **No Volátiles**, significa que los datos que contiene una Bodega de datos no son regularmente accedidos o modificados. Esto es lógico porque el propósito de la bodega es permitir que se analice lo que ha ocurrido; y por ultimo **De tiempo variante**, una bodega de datos basa su funcionalidad en la recolección de datos en horizontes de tiempo muy grandes para realizar análisis o pronósticos a través de acontecimientos pasados en la organización

Si se mira con detalle la definición y las características generales del Autor se podría pensar que existe una relación muy estrecha entre el modelo desarrollado en el presente proyecto y la teoría expuesta en las bodegas de datos, o por lo menos en lo que respecta a la representación del conocimiento. Pero existen muchas diferencias de fondo derivadas de la forma como se desarrollo este proyecto.

En su definición el autor pone de manifiesto la importancia de los procesos de diseño y construcción de una bodega de datos, estos procesos no fueron tomados en cuenta en este proyecto dado que en principio no se tenía la idea clara del camino que se iba a tomar. El no utilizar la metodología propuesta para el desarrollo de las bodegas de datos hace que no se apliquen de forma estricta algunas de las características más importantes de estas. Por ejemplo la característica del tiempo variante que propone Inmon, aunque se desarrollo de una forma parecida no permite realizar análisis de tiempos anteriores, es decir, no podríamos obtener historiales de las decisiones que se toman aunque exista la información histórica. Además, en la misma forma como fue concebida la representación de conocimiento en este proyecto hay diferencias importantes con respecto a la forma como se conciben las bodegas de datos ya que estas ultimas parten de la necesidad de permitir a un conjunto de usuarios externos al sistema tomar las decisiones sobre la información organizada en la bodega mientras que en este proyecto se pensó únicamente con el objetivo de soportar el proceso de toma de decisiones pero automático, es decir dentro del mismo sistema.

Otra diferencia importante radica en el manejo de las dimensiones y las arquitecturas que existen dentro de la teoría de las bodegas de datos. Aunque observando el modelo de datos del mecanismo de representación del conocimiento presentado en la sección 3.1.3, se observa una arquitectura estrella con tres dimensiones (Estudiante o usuario, Atributos y Perfil) no se podría incluir los valores de los atributos como dimensiones adicionales, es decir, el pensar que los valores de los atributos modelo, estilo, entre otros, puedan ser dimensiones adicionales constituyen un error en la teoría de las bodegas de datos por que en este proyecto estos valores no referencia ninguna tabla por medio de una clave única y solo representan el valor de ese atributo en la experiencia del estudiante, lo que en el ambiente de las bodegas de datos representaría el valor del hecho relacionado con cada dimensión sobre el cual se desarrollo la bodega. Este problema se vuelve mas critico cuando se encuentra que en la teoría de las bodegas de datos se plantea la necesidad de que en cada dimensión exista una jerarquía que sirva como eje en el soporte del proceso de la toma de decisiones, jerarquías que en este proyecto no se tienen.

Este es el panorama a nivel teórico, a nivel de la implementación las diferencias son mucho más notables, ya que por el hecho de no pensarse como una bodega de datos no se utilizo ninguna de las herramientas de desarrollo que estas presentan, lo que haría complicada la conexión de estas con desarrollos en OLAP o DataMining.

Es claro que existen grandes relaciones entre el modelo desarrollado en este proyecto y las bodegas de datos pero igualmente es claro que no se puede afirmar que el mecanismo de representación del conocimiento desarrollado en este proyecto constituye por si solo una bodega de datos.

2.5.2. Sistemas de Razonamiento Basado en Casos - CBR

El aprendizaje basado en casos consiste en adquirir conocimiento a partir de experiencias precedentes o casos. También se conoce como razonamiento basado en casos por el hecho de que este tipo de aprendizaje no se concibe sin el proceso de razonamiento que conlleva la obtención de una nueva experiencia. Es un método para resolver problemas recordando situaciones similares, reutilizando la información y el conocimiento sobre dichas situaciones [13]. La idea original es simple: **“Un sistema basado en casos resuelve nuevos problemas adaptando soluciones que fueron usadas para resolver problemas anteriores.”**

Esta teoría resume de forma muy sencilla la filosofía del modelo desarrollado en este proyecto, pero existen algunas otras características de los CBR que no se desarrollan en nuestro proyecto. Por ejemplo, la memoria del sistema CBR almacena un cierto número de problemas junto a sus correspondientes soluciones. La solución de un nuevo problema se obtiene recuperando los casos similares almacenados en la memoria del sistema CBR. Por tanto, un caso engloba un problema y la solución dada a dicho problema. En este proyecto los casos o experiencias son agrupadas todas y el árbol de decisión se genera con todo el conjunto, por lo tanto la decisión no se toma con la información únicamente de un caso anterior, si no que utiliza todo el conjunto de experiencias para generar la respuesta. Esta se constituye en la principal diferencia entre los CBR y el modelo desarrollado.

CAPITULO 3. EL MÓDULO INTELIGENTE DEL STI.

El modelo de toma de decisiones que se describió en el capítulo anterior es la base del módulo inteligente que se implementó en este proyecto con el fin de verificar los resultados de la investigación. Pero es necesario realizar algunas adecuaciones para que pueda integrarse correctamente a los demás módulos del STI que se están desarrollando en el macroproyecto Unicauca Virtual.

Por tal motivo, se tratará este módulo como una caja negra (tal como se trata un módulo software), es decir, un sistema con entradas, procesos y salidas; donde se mostrará la forma que tendrá cada componente del módulo.

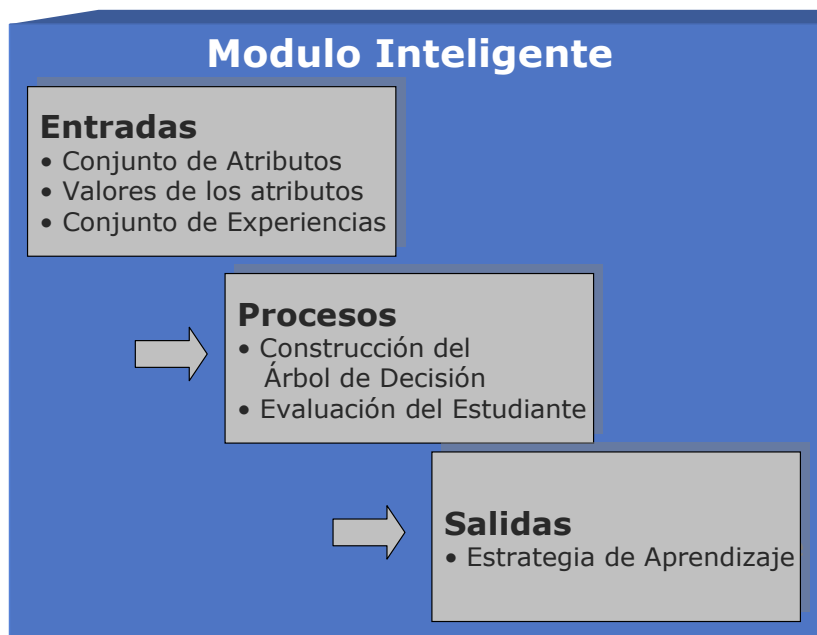


Figura 13. Modulo Inteligente

3.1. Entradas

Para entender de que forma se generan las entradas, es necesario recordar que en el capítulo anterior se dijo que el algoritmo inteligente ID3 recibe tres entradas diferentes (Atributos y sus valores, Conjunto de ejemplos, Atributo resultado o respuesta). De estas tres entradas es necesario obtener dos de ellas desde componentes fuera del módulo inteligente o tutor, estas son: El conjunto de Atributos y valores, y el conjunto de ejemplos o experiencias del sistema. La razón por la cual el atributo de respuesta o resultado no va a ser tomado como una entrada al igual que los dos anteriores es por que este va a ser realimentado

dentro del mismo conjunto de experiencias, es decir que hará parte de cada experiencia y será invariante por las mismas decisiones tomadas al desarrollar el modelo.

3.1.1. Conjunto de Atributos y sus valores

Es una de las entradas del algoritmo y tiene especial importancia ya que son ellos los que proporcionaran al algoritmo las herramientas suficientes para tomar la mejor decisión acerca de cual estrategia de aprendizaje es la mas adecuada para implementar al estudiante. Cabe anotar que cuanto mayor sea el número de atributos que se le definan al sistema y cuanto mayor sea su poder descriptivo⁹ mejor será la respuesta que se obtenga del módulo tutor.

Para desarrollar el prototipo que servirá para la validación de resultados en este proyecto se definió el siguiente conjunto de atributos y los valores a tomar en cuenta Ver Tabla 4 y Tabla 5

Atributo	Valores
<i>MODELO</i>	Felder
	Chaea
<i>ESTILO</i>	Activo
	Reflexivo
	Teórico
	Pragmático
	Visual
	Verbal
	Intuitivo
	Sensitivo
	Global
	Secuencial
	<i>NIVEL_ESTILO</i>
Medio	
Bajo	

Tabla 4. Atributos Estudiante

Atributo	Valores
<i>ESTRATEGIA</i>	ApoyarEstilo
	ContraEstilo
	Nivelar
<i>TIPOACTIVIDAD</i>	Análisis y Reflexión
	Demostración

⁹ Este se refiere a la capacidad de distinción que tiene el atributo de las aptitudes del estudiante.

	Explicación
	Refuerzo
	Motivación
	Evaluación
	Repaso
	Lectura
	Simulación
	Juego
<i>ESTRUCTURA_ARBOL</i>	Chioce
	ChioceExit
	Flow
	FowardOnly

Tabla 5. Atributos Sistema

La razón por la cual el conjunto de atributos fue dividido de esta forma ya se explico anteriormente. La razón por la cual fueron tomados estos atributos fue resultado del hecho de favorecer la utilización de información y conceptos que se definieron para todo el macroproyecto Unicauca Virtual y que por tanto hacen parte no solo de este proyecto si no también de los demás proyectos que desarrollan algunos otros módulos del sistema general.

3.1.2. Conjunto de ejemplos o experiencias

El conjunto de ejemplos o experiencias se generara como un esfuerzo conjunto del proyecto “AGENTES INTELIGENTES PARA ADAPTAR LA PRESENTACION DE CONTENIDOS PARA CURSOS DE EDUCACIÓN EN LÍNEA”¹⁰ y este proyecto. El primero, utilizara un sistema multiagente que se encargara de monitorear al estudiante y una vez que una experiencia sea generada por este, el sistema la guardara para que pueda ser usada por el modulo inteligente. En conjunto, ambos proyectos acordaron tres eventos importantes que dan lugar a generar una experiencia. El primero tiene lugar cuando el estudiante que ingresa a su sesión y comienza un capitulo de un curso representado por una agregación, una vez que observa y realiza todas las actividades de la agregación se dispone a presentar la evaluación, en ese momento se genera la experiencia con los valores de los atributos que el estudiante tenga en ese momento y el resultado de la evaluación (un valor boleano indicando si paso el capitulo o no). El segundo evento se presenta cuando el estudiante que inicia su estudio de una agregación decide salirse de sistema sin terminar de ver todas las actividades de la agregación y sin presentar la evaluación, en este caso se debe reportar los valores de los atributos que tenga el estudiante y una valoración negativa (es decir, como si hubiese perdido la evaluación) dado que la estructura actual del sistema no fue

¹⁰ Proyecto encargado de desarrollar el Modulo Interfaz del STI

de agrado para él. El tercer evento que da lugar a generar una experiencia es cuando el mismo estudiante, una vez que ha comenzado el estudio de una agregación solicita al sistema que cambie las actividades que le esta presentando por otras o que por medio del modulo inteligente le genere una nueva estrategia de aprendizaje; en este caso el reporte de la experiencia es igual que el anterior, es decir, negativo.

Con el conjunto de experiencias almacenadas en la base de datos como conocimiento relacional, el trabajo consiste en que por medio de SQL se estructure correctamente esta información para que pueda ser pasada al algoritmo.

Para dar un poco más de claridad se muestra un ejemplo de lo que seria experiencia real igual a las que ingresan al modulo inteligente, para ello se debe tener en cuenta los seis atributos y sus respectivos valores definidos en el ítem anterior:

Modelo	Estilo	Nivel de Estilo	Estrategia	Estructura Árbol	Tipo Actividad	Resultado
Felder	Global	Medio	ApoyarEstilo	Choice	Explicación	Positivo

Tabla 6. Ejemplo de una experiencia

La riqueza semántica del conocimiento relacional se puede ver fácilmente cuando se interpreta lo que esta experiencia significa. Para el modulo inteligente esta experiencia propone que los estudiantes que tengan el modelo de clasificación de *Felder*, y que hayan sido clasificados con el estilo *Global* en un nivel *Medio* (nótese que estos datos no pueden ser tomados como elementos mutables a la hora de realizar la sugerencia) el sistema debe propender a utilizar como estrategia *ApoyarEstilo*, fijar la estructura del árbol en *Choice* y utilizar actividades del tipo *Explicación* dado que esta experiencia resulto positiva, es decir, que existe gran posibilidad de éxito si un nuevo estudiante que tenga los mismos tres valores citados (*Felder*, *Global*, *Medio*) en los atributos Estudiante, el sistema le coloca como estrategia *ApoyarEstilo*, le fija el valor de la estructura del árbol en *Choice* y le presenta actividades del tipo *Explicación*.

Ahora, es un poco mas entendible el por que de la distinción de los atributos en los dos tipos y como se van a tomar las decisiones en el modulo inteligente.

3.1.3. Modelo de Datos del Modulo Inteligente

La Figura 14 muestra el modelo de datos desarrollado para almacenar en tablas relacionales la información referente a las entradas del modulo inteligente. Consta, básicamente y sencillamente, de tres tablas relacionadas entre ellas y con dependencia hacia la tabla usuario que registra la información del estudiante. La tabla “tblPerfil” almacena las diferentes experiencias de los estudiantes, ella guarda el identificador del estudiante, una secuencia de las diferentes experiencias de este con el modulo inteligente y el resultado de cada una. Vale la pena aclarar que el mismo estudiante puede tener una o muchas experiencias dentro del modulo, ya que se deben considerar casos de estudiantes con experiencias negativas recurrentes, o varias experiencias positivas en diferentes materias o cursos; y dado que el modelo no utiliza estas experiencias particularmente y solo las utiliza como parte de su entrenamiento, no existe problema para el modelo en que estas sean numerosas.

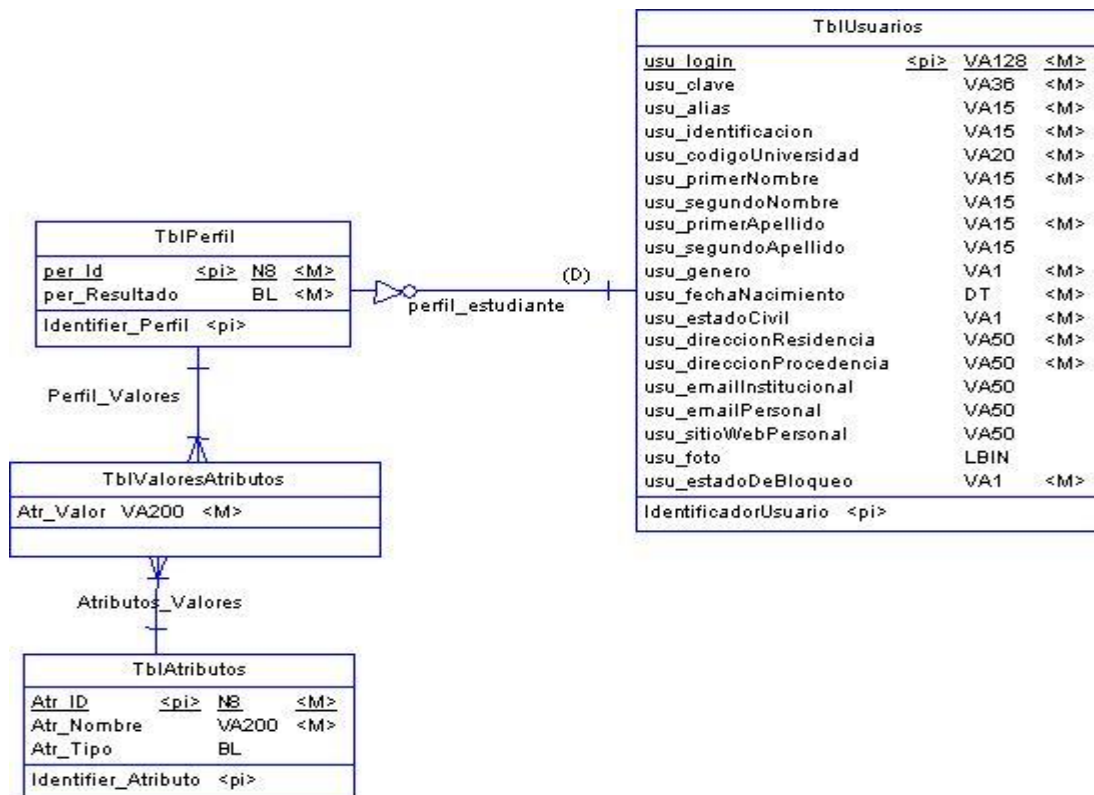


Figura 14. Modelo de Datos del Modulo Inteligente

La tabla “tblAtributos” solo contiene la información de los atributos definidos para el modulo y su clasificación, es decir, si son Atributos Sistema o Atributos Estudiante.

La tabla “tblValoresAtributos” es la tabla más importante del modelo, ella contiene las entradas del modulo: el conjunto de experiencias y los diferentes valores de los atributos. El conjunto de experiencias se almacena al guardar para cada estudiante un valor de atributo para cada uno de los atributos definidos y los diferentes valores de los atributos se obtienen al realizar consultas del tipo *Distinct* para cada identificador de atributo.

Como se ve a simple vista el modelo de datos resulta bastante sencillo y practico, además, es completamente dinámico y extensible a la cantidad de atributos que se deseen al igual que los diferentes valores que estos pueden tomar.

3.2. Procesos

El modulo inteligente tiene su parte más importante en el contexto de los procesos. Es acá donde se analiza toda la información con el objeto de tomar la decisión más adecuada para la estrategia de aprendizaje del estudiante.

Este sector del modulo inteligente se encuentra dividido en dos partes fundamentales, una que se encarga de recibir las entradas y generar el árbol de decisión y una segunda que tiene el objetivo fundamental de evaluar el árbol de decisión, elegir los atributos del sistema a sugerir y posteriormente realizar la evaluación de las reglas asociadas a dichos atributos elegidos con el fin de generar las salidas en forma de estrategias de aprendizaje.

3.2.1. Proceso de construcción del árbol de decisión

No se explicara a fondo este proceso, ya que esa labor fue realizada en capítulos anteriores. Solo se completara diciendo que para su construcción se utilizaron los atributos que se presentaron en la sección 3.1.1 con sus respectivos valores y que las experiencias sometidas al modulo inteligente se generaron con valores aleatorios en la primera experiencia¹¹, para posteriormente realizar una segunda donde el conjunto de experiencias fue estructurado por los integrantes del proyecto buscando darle consistencia lógica a los resultados, basados en el conocimiento adquirido durante todo el desarrollo del proyecto en pedagogía, estilos de aprendizaje y diseño instruccional.

Una parte importante del proceso de construcción del árbol de decisión consiste en mostrar las estructuras que son utilizadas para su creación. Estas son básicamente tres y se muestran en la Figura 15

¹¹ Con el objeto de observar el comportamiento del árbol de decisión

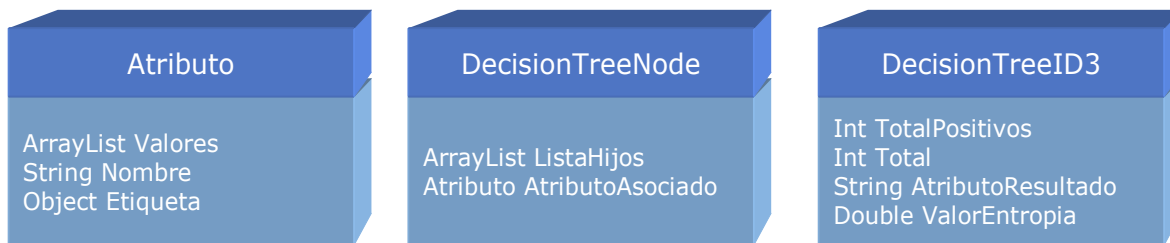


Figura 15. Estructuras del Árbol de Decisión

La clase Atributo representa un atributo como tal, contiene dentro de sus atributos la lista de los diferentes valores que este puede tomar, el nombre del atributo o una etiqueta que se utiliza para determinar si pertenece a un nodo hoja, es decir, a un nodo de resultado. Dentro de sus principales funciones están su constructor sobrecargado para determinar si se inicializa con un atributo (en caso de pertenecer a un nodo intermedio) o con la etiqueta (en caso de pertenecer a un nodo hoja) y una función que retorna su lista de posibles valores.

La clase DecisionTreeNode representa un nodo dentro del árbol de decisión, como son árboles del tipo n-ario estos tiene un atributo que contiene la lista de los hijos, tal como lo sugiere la representación “lista de hijos”, y un atributo llamado AtributoAsociado que representa una instancia de la clase Atributo que se asocia a este nodo del árbol. Tiene además gran variedad de funciones para el manejo de la lista de hijos, como insertar hijo, obtener hijo, contar hijos y una función que retorna el atributo asociado a ese nodo.

La clase DecisionTreeID3 representa el árbol de decisión creado a partir de un conjunto de atributos y un conjunto de experiencias. Los atributos y métodos asociados a esta clase se relacionan con los valores que se calculan dentro de algoritmo de creación del árbol de decisión, como la entropía, la ganancia, los valores positivos, los valores negativos, entre otros; pero además tiene el método que recursivamente crea el árbol de decisión basado en el algoritmo ID3.

3.2.2. Proceso de Evaluación y Estructuración

La forma de operación de este algoritmo se encuentra descrita en el capítulo 2, por tal razón no se profundizara al respecto. De igual forma el pseudo código del algoritmo se puede encontrar en los anexos.

Cuando los valores de los atributos del sistema se eligen con los criterios definidos anteriormente, ellos se evalúan dentro de unas reglas lógicas definidas por nosotros de acuerdo a las características que tengan cada uno de los valores de los atributos, de esta forma:

Atributo Estrategia	
Apoyar Estilo	Si se tiene este valor se buscará el estilo de aprendizaje del estudiante, y al momento de generar las reglas de secuenciación para las actividades del contenido se buscaran recursos que tengan este tipo de estilo definido dentro de su metadato.
Contra Estilo	Igualmente se consulta el estilo del estudiante como en el caso anterior, solo que al momento de generar las reglas de secuenciación se buscaran recursos con el estilo contrario al que se consulto.
Nivelar	Se incluirán en las reglas de secuenciación recursos de los dos tipos de estilos de aprendizaje, el definido y el contrario

Atributo Estructura	
Choice	Se le fijara la estructura de contenidos del estudiante en el valor de Choice = true
ChoiceExit	Se le fijara la estructura de contenidos del estudiante en el valor de ChoiceExit = true
Flow	Se le fijara la estructura de contenidos del estudiante en el valor de Flow=true.
FowardOnly	Se le fijara la estructura de contenidos del estudiante en el valor de ForwardOnly = true.

Atributo TipoActividad	
Any	Dado que para cualquier valor del atributo el procedimiento es el mismo vamos a resumir la regla diciendo que dependiendo de cual sea el valor del atributo que se tenga se buscaran las actividades de este tipo para incluirlas en las reglas de secuenciación que se le presentaran en la estructura de contenidos del estudiante. Por ejemplo, si el valor es <i>Explicación</i> se incluirán el mayor número de actividades de este tipo dentro de la estructura de contenidos del estudiante en cuestión usando para ello las reglas de secuenciación y los metadatos de los recursos.

3.3. Salidas

El sector de salidas del modulo inteligente tiene la tarea de que luego de evaluar las reglas definidas anteriormente se realicen las consultas SQL a la base de datos para estructurar las reglas de secuenciación que serán almacenadas y los

nuevos valores de los atributos, dejando su implementación en manos del proyecto “AGENTES INTELIGENTES PARA ADAPTAR LA PRESENTACIÓN DE CONTENIDOS PARA CURSOS DE EDUCACIÓN EN LÍNEA” el cual tiene esta tarea como uno de sus objetivos.

Las estructuras de almacenamiento que se utilizaran para esta labor son las descritas en la Figura 16, donde se genera la información necesaria para que el modulo Interfaz del STI pueda aplicarlas correctamente.

La tabla “tblEstrategiaEstilo” recoge la información de la estrategia que esta utilizando el sistema para un estudiante en particular, en ella se almacenan la información de la asignatura, el curso, el valor de la estrategia (ApoyarEstilo, ContraEstilo o Nivelar) y la fecha en que se le dio ese valor para un estudiante en particular. La tabla “tblEstrucContenidosEstudiante” contiene la información personalizada de la estructura de contenidos del estudiante en particular, que tiene los resultados del proceso de toma de decisiones que se realizo sobre el perfil del estudiante. Esta ultima es una copia del la tabla “tblEstrucContenidosPersonal” sumándole el campo que identifica al estudiante, *usu_login*.

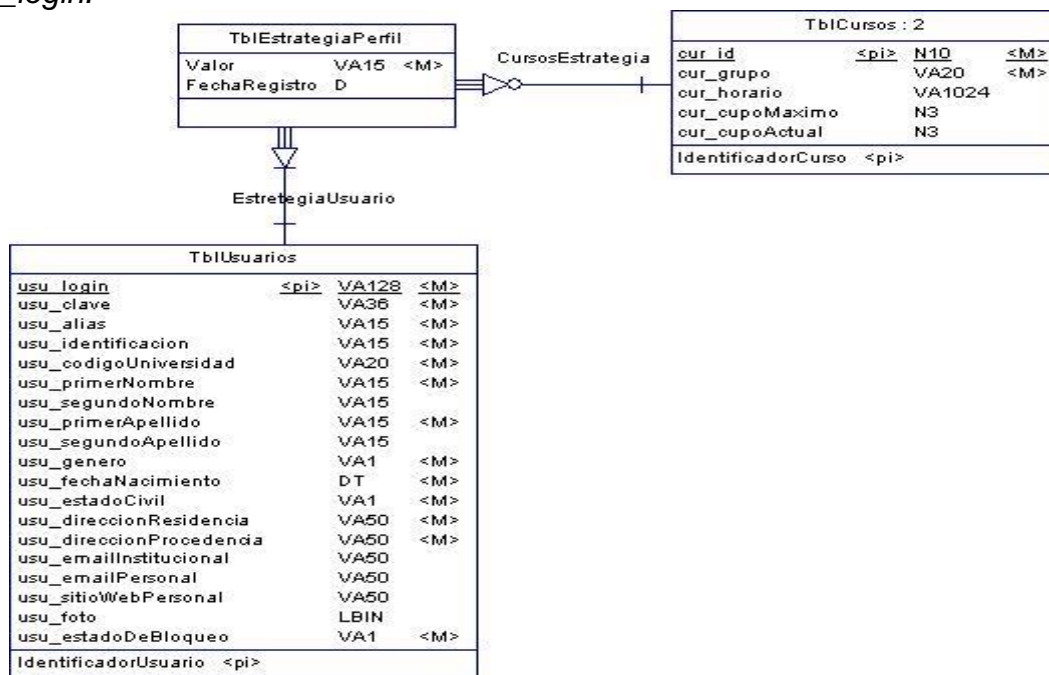


Figura 16. Modelo de Datos de las Salidas

CAPITULO 4. ETAPA DE ANÁLISIS

La esencia del análisis y el diseño orientado a objetos consiste en situar el dominio de un problema y su solución lógica dentro de la perspectiva de los objetos (cosas, conceptos o entidades)[41].

El desarrollo de este proyecto es guiado por el Proceso Unificado de Desarrollo (RUP), utilizando como lenguaje de modelamiento el UML (Lenguaje Unificado para la Construcción de Modelos). El RUP se caracteriza por ser iterativo e incremental, y conducido por las siguientes fases a nivel macro: Planeación y Elaboración, Construcción, y Aplicación.

4.1. Requerimientos

Son una descripción de las necesidades o deseos del producto a desarrollar, A continuación se mostrarán los artefactos generados en esta etapa:

❖ Panorama general

“INTELIGENCIA ARTIFICIAL EN LA GENERACIÓN DE ESTRATEGIAS DE APRENDIZAJE DE UN SISTEMA TUTOR INTELIGENTE”, es un proyecto que pretende generar a través del modulo tutor inteligente de un STI, estrategias de aprendizaje personalizadas que se ajusten al perfil del estudiante de un curso en línea, utilizando algún enfoque de inteligencia artificial, que le permita a dicho modulo tomar decisiones simples en el proceso de aprendizaje de los estudiantes y proporcionando una educación más personalizada y por ende más efectiva.

❖ Clientes

Antes de especificar el cliente directo se hará claridad que el sistema que se desarrolló tiene 3 actores fundamentales, un directivo quien es el que proporciona las herramientas necesarias para el profesor (que es el segundo actor) quien es el que construirá un ambiente adecuado inicial para el estudiante (que vendría a ser nuestro cliente directo) pueda interactuar directamente con la herramienta, y esta le proporcione la personalización deseada y le lleve a un estado de asimilación del conocimiento ideal a través de un proceso educativo de calidad.

❖ Metas

En términos generales, la meta del presente proyecto es ayudar a construir una educación en línea más efectiva en el proceso de aprendizaje del alumno y más

personalizada a las necesidades de los mismos, teniendo en cuenta su perfil y su estilo de aprendizaje, generando estrategias de enseñanza -aprendizaje basadas en las características antes mencionadas y en los datos recogidos durante el monitoreo del estudiante en el ambiente virtual.

4.2. Descripción del Sistema

❖ Funciones del Sistema

Como es bien sabido las funciones del sistema son las acciones que el sistema debe hacer y pueden priorizarse de acuerdo a su grado de importancia o visibilidad en el sistema, es decir, se puede establecer cuales de ellas son esenciales para el funcionamiento adecuado del sistema y cuales por el contrario pasarían inadvertidas pero que consumen tiempo y recurso. Las categorías se muestran en la Tabla 7: [41]

Categoría de la Función	Significado
Evidente	Debe realizarse y el usuario debería saber que se ha realizado
Oculto	Debe realizarse, aunque no es visible para los usuarios
Superflua	Opcionales; su inclusión no repercute significativamente en el costo ni en otras funciones.

Tabla 7. Clasificación de las funciones del sistema

De acuerdo a la anterior clasificación se tienen entonces las siguientes funciones para el presente proyecto.

Funciones Básicas del Sistema

Ref. #	Función	Categoría
R1.1.	Entrada al Sistema	Evidente
R1.2.	Validación del Usuario	Oculto
R1.3.	Salida del Sistema	Evidente

Tabla 8. Funciones Básicas del Sistema

Funciones de la Gestión de las Agregaciones

Ref. #	Función	Categoría
R2.1.	Definir Agregaciones	Evidente
R2.2.	Modificar Agregaciones	Evidente

R2.3.	Consultar Agregaciones	Ocultas
R2.4.	Eliminar Agregaciones	Evidente

Tabla 9. Funciones Gestión Agregación

Funciones de la Gestión de la Estructura de Contenidos

Ref. #	Función	Categoría
R3.1.	Crear Estructura de Contenidos	Evidente
R3.2.	Consultar Asignaturas	Ocultas
R3.3.	Consultar Actividades	Ocultas

Tabla 10. Funciones Gestión Estructura de Contenidos

Funciones de la Gestión del Contenido Programático

Ref. #	Función	Categoría
R4.1.	Definir Contenido Programático	Evidente
R4.2.	Activar Contenido Programático	Evidente
R4.3.	Modificar Contenido Programático	Evidente
R4.4.	Consultar Contenido Programático	Ocultas

Tabla 11. Funciones Gestión Contenido Programático

Funciones de la Gestión de los Ítems del Contenido Programático

Ref. #	Función	Categoría
R5.1.	Definir Ítems Contenido	Evidente
R5.2.	Modificar Ítems Contenido	Evidente
R5.3.	Consultar Ítems Contenido	Ocultas
R5.4.	Eliminar Ítems Contenido	Evidente

Tabla 12. Funciones Gestión Ítems Contenido

Funciones de la Gestión de las Actividades

Ref. #	Función	Categoría
R6.1.	Definir Actividades	Evidente
R6.2.	Modificar Actividades	Evidente
R6.3.	Consultar Actividades	Ocultas
R6.4.	Eliminar Actividades	Evidente

Tabla 13. Funciones Gestión Actividad

❖ Atributos del Sistema

Los atributos o características del sistema son:

Atributo	Detalles y restricciones de frontera
Tiempo de respuesta	(Restricción de Frontera) En cuanto a la toma de decisiones del modulo tutor, el tiempo de respuesta de este estará ligado no solo a la cantidad de experiencias registradas en la base de datos, sino de la cantidad de atributos y la cantidad de valores que dichos atributos podrán tomar, este tiempo podrá ser desde unos cuantos segundo hasta máximo 1 minuto. En cuanto a las demás consultas de la base de datos el tiempo de respuesta será mucho menor.
Metáfora de Interfaz	(Detalle) Las interfaces están diseñadas haciendo uso de la metáfora de ventanas y botones del Sistema Operativo Windows, cuyos mensajes emergentes usan la metáfora de cuadros de dialogo.
Tolerancia a Fallas	(Restricción de Frontera) El sistema esta diseñado para validar los errores de sucesos inesperados, tales como caída del servidor o de la bases de datos.
Facilidad de Uso	(Detalle) El sistema esta diseñado de manera que la navegación por el mismo sea intuitivo, claro y entendible, El estudiante tendrá total transparencia en la toma de decisiones del modulo tutor, haciendo que la interfaz este menos cargada.
Plataformas del sistema operativo	Microsoft Visual Studio .Net 2003 Enterprise, Microsoft Windows XP, Microsoft Office 2003, Oracle 10g.

Tabla 14. Atributos del Sistema

4.3. Diagrama General de Casos de Uso

Los casos de uso que se encuentran resaltados son los casos de uso más importantes para el funcionamiento del sistema y por ende tienen mucha repercusión en los objetivos del proyecto, por esta razón los artefactos de los demás casos de uso se verán con claridad en los anexos.

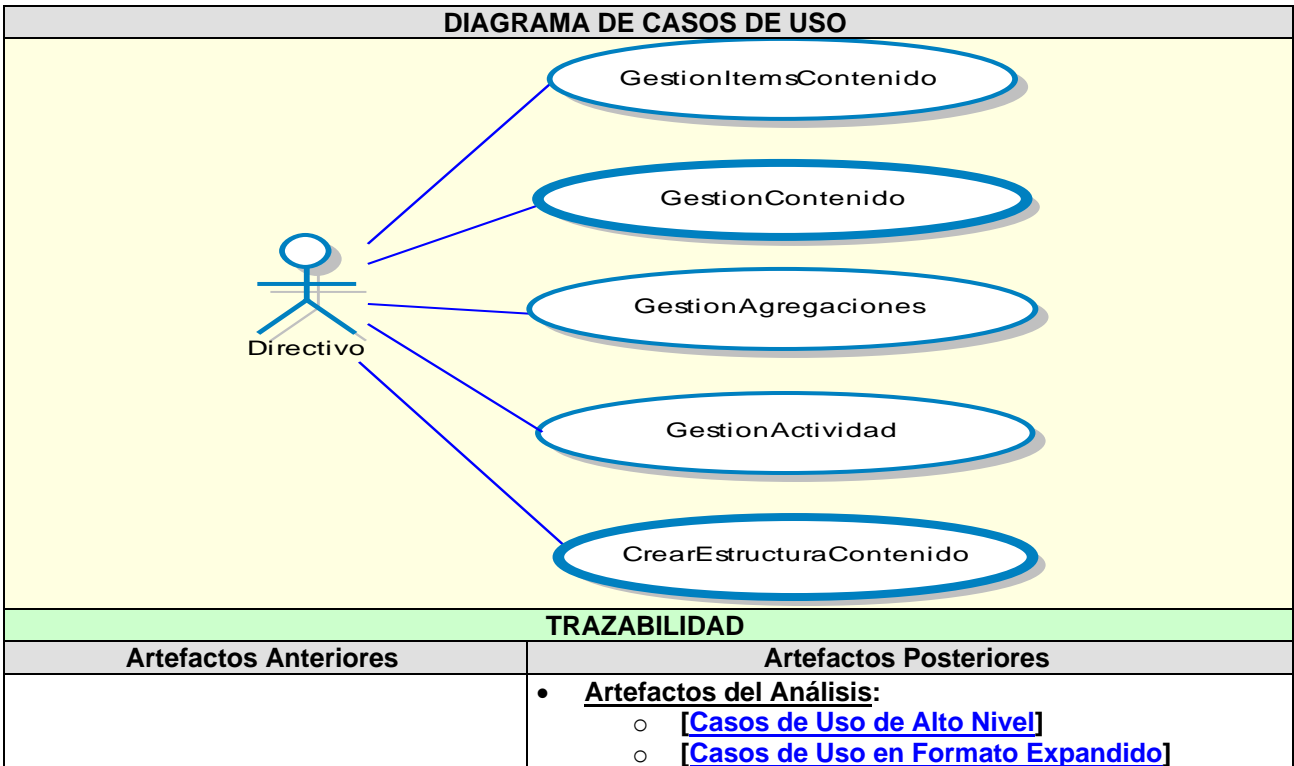


Tabla 15. Diagrama de Casos de Uso

4.3.1. Caso de Uso: Gestionar Ítems Contenido

La Gestión de Los ítems de Contenidos lleva a cabo 4 casos de uso Insertar, Consultar, Modificar y Eliminar Ítems de Contenidos.

Teniendo en cuenta que el presente proyecto no se va a ligar a una institución educativa en particular, se proporciona la flexibilidad a dichas instituciones de agregar o eliminar los ítems que de acuerdo al proyecto educativo institucional sean los necesarios y relevantes. Dichos ítems serán utilizados como plantilla para la creación del contenido programático de cualquier asignatura y deberán ser llenados por los tutores de cada asignatura con su respectiva información, los ítems podrán ser consultados modificados únicamente por el usuario Directivo. Ver ANEXO 12

4.3.2. Caso de Uso: Gestionar Contenido Programático

La Gestión de los Contenidos Programáticos lleva a cabo 5 casos de uso:

- *DefinirContenidoProgramático* Permite al usuario consultar los ítems de contenido definidos con anterioridad y llenar cada uno de ellos con la

información correspondiente de acuerdo a la asignatura a la que se deberá asignar dicho Contenido Programático.

- *ConsultarContenidoProgramático* Como su nombre indica, permite al usuario consultar todos los contenidos programáticos asociados a una asignatura en particular así como la información consignada en cada uno de los ítems del contenido.
- *ActivarContenidoProgramático* Permite al usuario colocar uno de los contenidos asociados a una asignatura en particular en un estado de vigencia, que indica que dicho contenido será el actualmente utilizado para esa asignatura en cada uno de los cursos que sea necesario crear. Cuando el usuario coloca el contenido programático elegido en estado activo, inmediatamente se coloca la fecha actual como fecha de inicio de activación para ese contenido y como fecha de finalización para el contenido que haya estado activado con anterioridad.
- *ModificarContenidoProgramático* Permite al usuario hacer las modificaciones deseadas a la información consignada en cada uno de los ítems de dicho contenido programático, no se podrá asociar dicho contenido a otra asignatura y mucho menos se podrán cambiar los datos de identificación del mismo; simplemente la información de los ítems asociados a él. Ver ANEXO 13

4.3.3. Caso de Uso: Gestionar Agregaciones

La Gestión de las agregaciones lleva a cabo 4 casos de uso Insertar, Consultar, Modificar y Eliminar Agregaciones.

Las agregaciones son los nodos del árbol instruccional o estructura de contenidos que representa cada uno de los temas que deben ser dados durante el desarrollo de un curso virtual, estos temas se crearan de acuerdo al temario de la asignatura a la que pertenezcan. Las agregaciones pueden ser nodos padres (de otras agregaciones o actividades) o nodos hijos (nodos finales del árbol - hojas). El usuario puede insertar las agregaciones que desea, consultar y modificar las agregaciones existentes si lo desea y eliminar aquellas que no considere de utilidad y que no estén siendo utilizadas en alguna estructura de contenidos existente. Ver ANEXO 14

4.3.4. Caso de Uso: Gestionar Actividades

La Gestión de las actividades lleva a cabo 4 casos de uso Insertar, Consultar, Modificar y Eliminar Actividades.

Las actividades son los nodos del árbol instruccional o estructura de contenidos que representan las acciones planeadas por el usuario profesor para impartir los temas de la asignatura (agregaciones) dichas actividades asocian los recursos (SCO's) que se les mostrarán a los estudiantes para que interactúen directamente con ellos. Dichos recursos serán asociados a las actividades de acuerdo a las características propias del recurso y del estudiante por ejemplo un recurso que apoya un estilo de aprendizaje Z, será útil para un estudiante cuyo estilo predominante sea Z. Ver ANEXO 15

4.3.5. Caso de Uso: Crear Estructura de Contenido

El caso de Uso Crear Estructura de Contenidos permite al usuario Directivo armar un árbol instruccional para una asignatura en particular y su contenido programático ACTIVO, con los objetivos, las agregaciones y las actividades existentes en la base de datos y que se adecuen más al temario de dicha asignatura. Este caso de uso utiliza los casos de uso referentes a la consulta de las asignaturas, consulta del contenido programático activo para esa asignatura, gestión de las agregaciones y gestión de las actividades entre otros, en este caso de uso se hacen validaciones básicas como que la asignatura realmente tenga un contenido programático activo, que las actividades sean las hojas del árbol, etc. Ver ANEXO 16

4.4. Modelo Conceptual Preliminar

A continuación se dará una breve definición de cada uno de los conceptos involucrados en el modelo conceptual preliminar y luego se dará una explicación de las relaciones existentes entre ellos.

Usuario: Estudiante, Profesor, Directivo, Administrador todo aquel que pueda hacer login en el sistema, que tenga login y clave, además de los datos generales.

Asignatura: Materia que se dictara a un curso en particular

Periodo: Espacio de tiempo en el que se lleva a cabo el curso virtual, tiene una fecha de inicio y de fin. Generalmente es semestral.

Curso: Grupo de Usuarios Estudiantes que llevarán a cabo el proceso de aprendizaje de una asignatura en particular.

Contenido Programático: Presentación general de la asignatura, plantilla de ítems que define el usuario profesor para desarrollar un curso virtual, ej. Metodología, Objetivos, Horas Practicas, etc.

EstructuraDeContenido: Lista de Temas (agregaciones) que constituyen el temario general que constituye la esencia de la asignatura.

Agregación: Nodo de la estructura de contenidos o árbol instruccional que representa cada uno de los temas de la lista de temas que se deben impartir en una asignatura en particular.

Objetivo: Nodo de la estructura de contenidos o árbol instruccional que puede ser asociado solo a las agregaciones para ofrecer mayor claridad de la funcionalidad de dicha agregación (tema) en el árbol o estructura de contenidos.

Actividad: Nodo de la estructura de contenidos o árbol instruccional que puede ser asociado a una agregación (tema) y constituye la acción planeada del usuario profesor para impartir dicho tema.

TipoActividad: Clasificación de las actividades del árbol, ej. Actividad de Juego, Actividad de Lectura, Actividad de Refuerzo, etc.

Recurso: Objeto físico (lectura, video, simulación, etc.) que se asocia a la actividad para que el estudiante comprenda el tema padre (agregación) mostrada en el proceso de aprendizaje del mismo.

ReglaSecuenciación: Forma en que se debe presentar una estructura de contenidos, el orden de las agregaciones y actividades y el tipo de recurso que debe ir asociado a dicha actividad, manera como el usuario estudiante podrá navegar a través del árbol instruccional definido por el usuario Directivo o Profesor.

RollUp: Similar a la regla de secuenciación que define el estado de comprensión de un tema (agregación) de acuerdo al estado de las actividades hijas y sus recursos.

TipoRegla: Clasificación de las reglas de secuenciación, ej. Regla de Precondición o de Poscondición, para aplicar antes o después de un evento provocado por la acción del estudiante.

Acción: Es la parte real de las reglas, indica que debe hacer el sistema frente a dicho evento causado por la acción del estudiante.

Las relaciones más importantes están evidenciadas en el modelo conceptual preliminar. A continuación se dará una breve descripción del universo del discurso.

La estructura de contenido en sí misma es la entidad más importante para el dominio de este problema debido a que a través de ella el módulo inteligente desarrollado para el proyecto puede sugerir los modos de secuenciación, la estructura está conformada por nodos padres que serían las agregaciones y nodos hijos que son las actividades, las actividades pueden tener uno o varios recursos asociados dependiendo del estilo de aprendizaje que favorezca cada recurso. Una estructura de contenidos o árbol instruccional es armada para asociarla a un contenido programático de una asignatura que a su vez puede tener varios cursos a los cuales se le da una estructura personalizada para que el usuario profesor pueda hacer las modificaciones que considere pertinente o introduzca las actividades que desee para el proceso de aprendizaje del usuario estudiante; Por último las actividades que forman la estructura de contenidos pueden ser de diferentes tipos, por ejemplo, actividad de juego, de reflexión, de simulación entre otros tipos y pueden ser usadas como y cuando el usuario profesor lo desee, cabe aclarar que toda la planificación inicial del curso y de la estructura de contenido la hace el usuario directivo.

Tanto para el modelo conceptual preliminar, como para el modelo lógico y físico de datos y los artefactos del diseño se utilizó la herramienta de modelado Sybase PowerDesigner 11.0. A continuación se puede visualizar el modelo conceptual preliminar

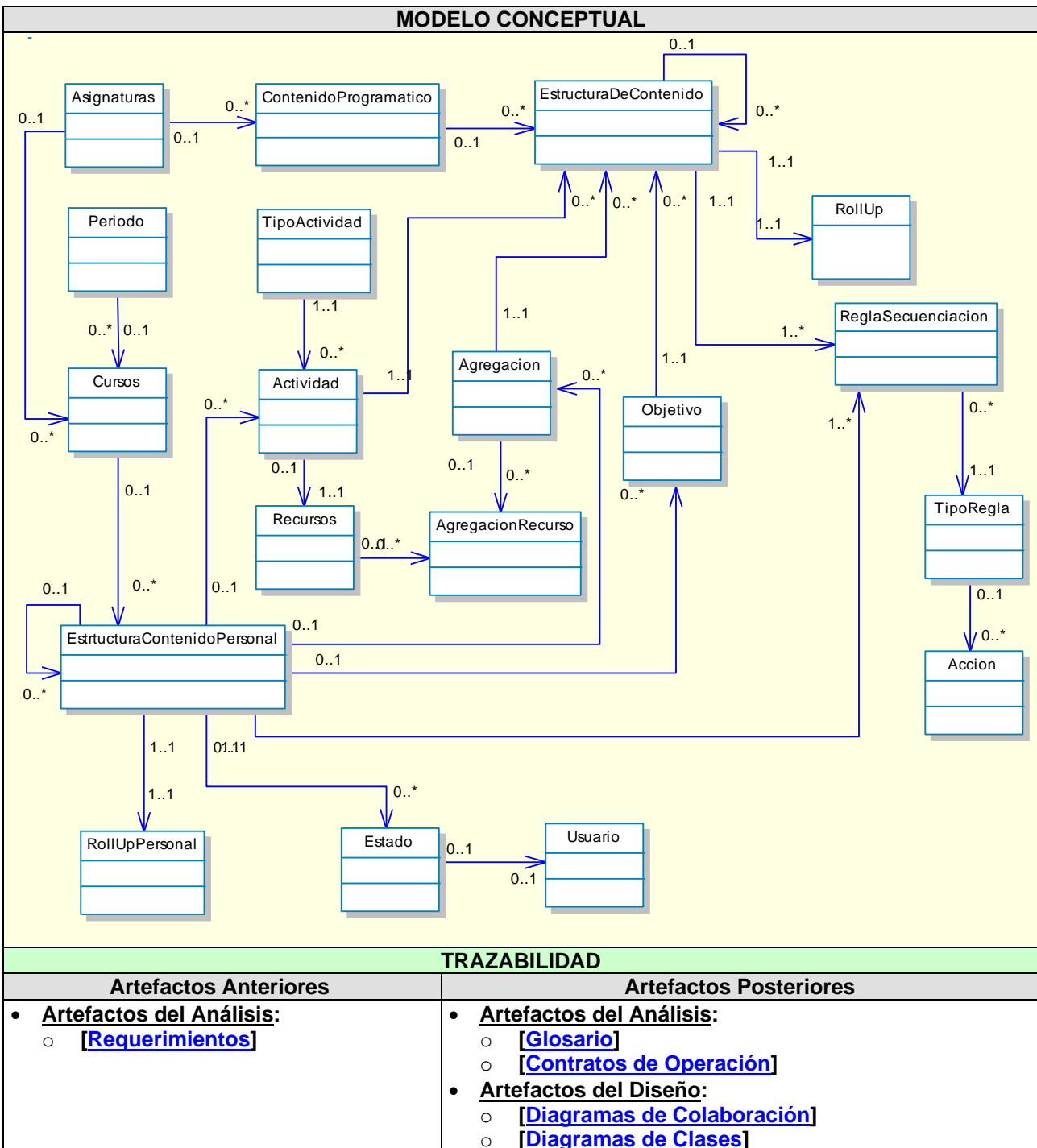


Tabla 16. Modelo Conceptual Preliminar

4.5. Modelo Lógico de Datos

El modelo lógico de Datos representa el Modelo Entidad – Relación obtenido del análisis realizado para la persistencia de los datos que maneja la herramienta desarrollada para este proyecto. A continuación se mostrará dicho modelo dividido en sub modelos teniendo en cuenta el papel que juega en los casos de uso descritos en el apartado anterior Ver apartado 4.3

4.5.1. Modelo de Datos: Gestionar Ítems Contenido

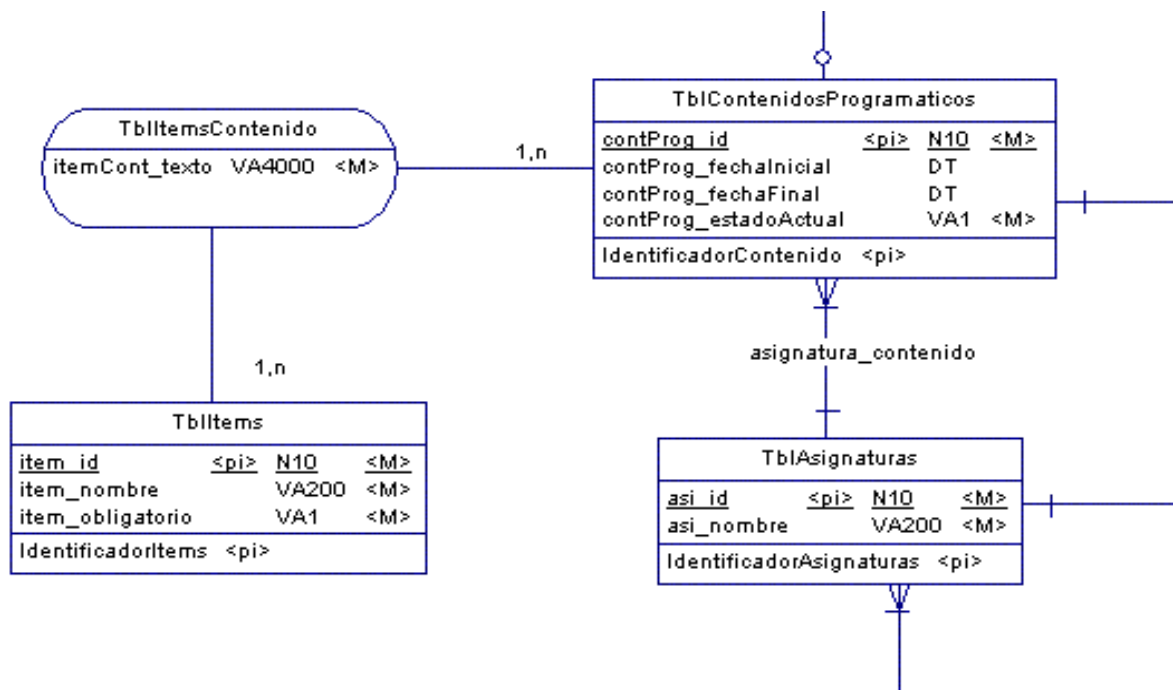


Figura 17. Modelo Gestión Ítems Contenido

El modelo de Datos para Gestionar los Ítems de Contenido involucra 4 tablas, de las cuales las más importantes son: TblItems, TblItemsContenido y TblContenidoProgramático

Los ítems de contenidos son creados con su identificador, su nombre y un valor adicional que indica si el ítem debe ser obligatoriamente llenado en todas las plantillas para la creación de contenidos, cada ítem es guardado en la tabla TblItems a la orden del usuario Directivo; El directivo a su vez es el encargado de crear los contenidos programáticos, para ello el sistema debe proveer una plantilla de creación donde se cargan los ítems de contenido recientemente creados para que el usuario digite la información que desee y la guarde en la tabla

TBLItemsContenido, cuando el usuario ha digitado dicha información para cada ítem, envía a la base de datos dicho contenido no sin antes asociarlo a una asignatura de las que existan en la TblAsignaturas, este contenido se guarda en la TblContenidosProgramaticos y estará disponible para realizar cualquier tipo de gestión sobre dicho contenido.

4.5.2. Modelo de Datos: Gestionar Contenido Programático

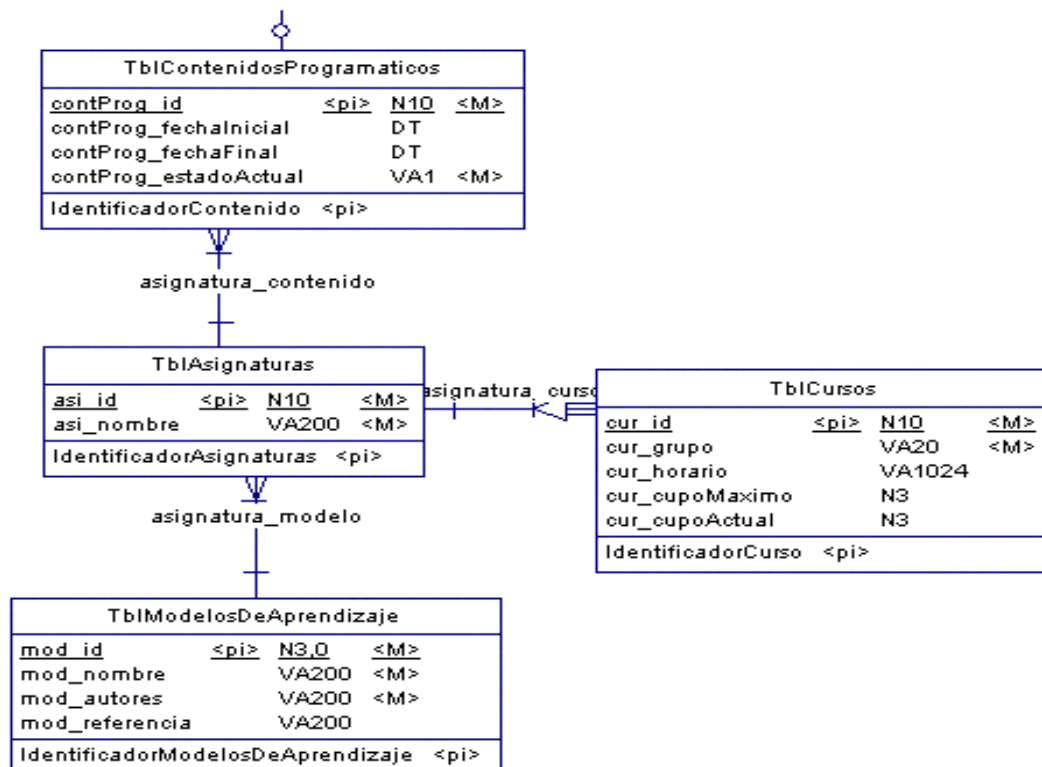


Figura 18. Modelo Gestión Contenido Programático

El modelo de Gestión de Contenidos Programáticos tiene las siguientes tablas: El contenido guardado en la TBLContenidosProgramáticos tiene un identificador único por el cual se podrá consultar, modificar, eliminar y activar, tiene dos campos fecha que corresponden a una fecha de inicio y una fecha final representando las fechas en que ese contenido programático se activa y se desactiva respectivamente. Como se dijo con anterioridad cada contenido programático está asociado a una asignatura en particular de la TBLAsignaturas, cada asignatura tiene de uno a varios cursos (TBLCursos) y estos pueden a su vez tener copias de la estructura de contenidos original.

4.5.3. Modelo de Datos: Gestionar Agregaciones

El modelo para la gestión de agregaciones tiene varias tablas cada una con funciones diferentes:

La TBLAgregaciones es la tabla más importante en este modelo puesto que es en esta tabla donde se ve reflejado toda acción del usuario, se guardan las agregaciones creadas que luego pueden ser consultadas, modificadas y si el usuario lo desea pueden ser eliminadas, siempre y cuando dicha agregación no esta relacionada con alguna estructura de contenidos existente

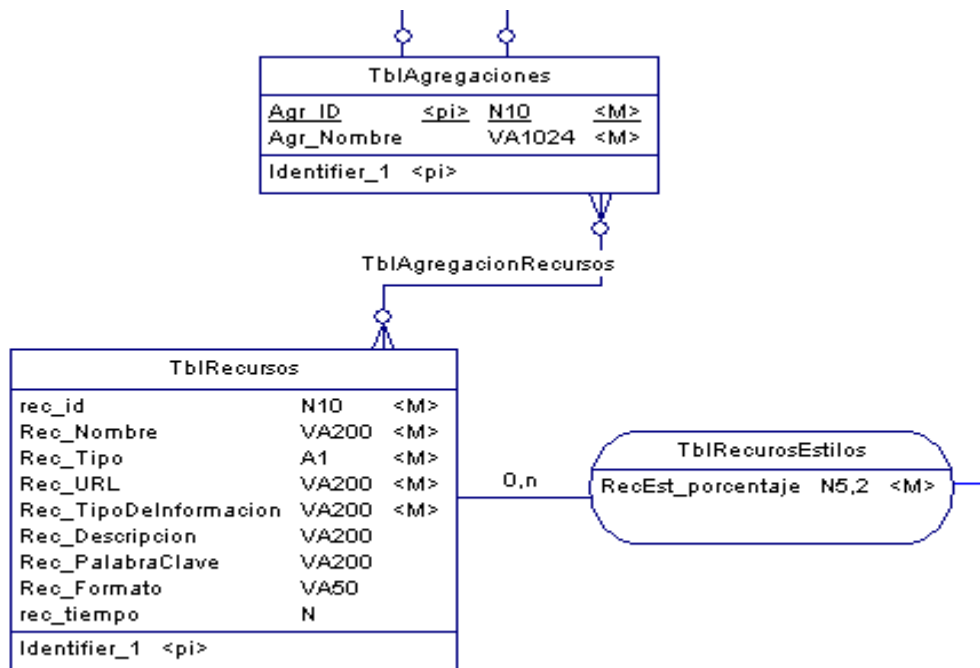


Figura 19. Modelo Gestión Agregaciones

4.5.4. Modelo de Datos: Gestionar Actividades

La gestión de Actividades es muy similar a la gestión de agregaciones con la diferencia de que el usuario Directivo puede consultar las actividades insertadas con anterioridad teniendo en cuenta dos criterios, el tipo de actividad al que pertenece que se ve en la tabla TBLTiposDeActividad y el nombre de la Actividad que se ve en la tabla TBLActividades, a su vez cada actividad puede tener asociados varios recursos dependiendo del estilo que apoye cada recurso.

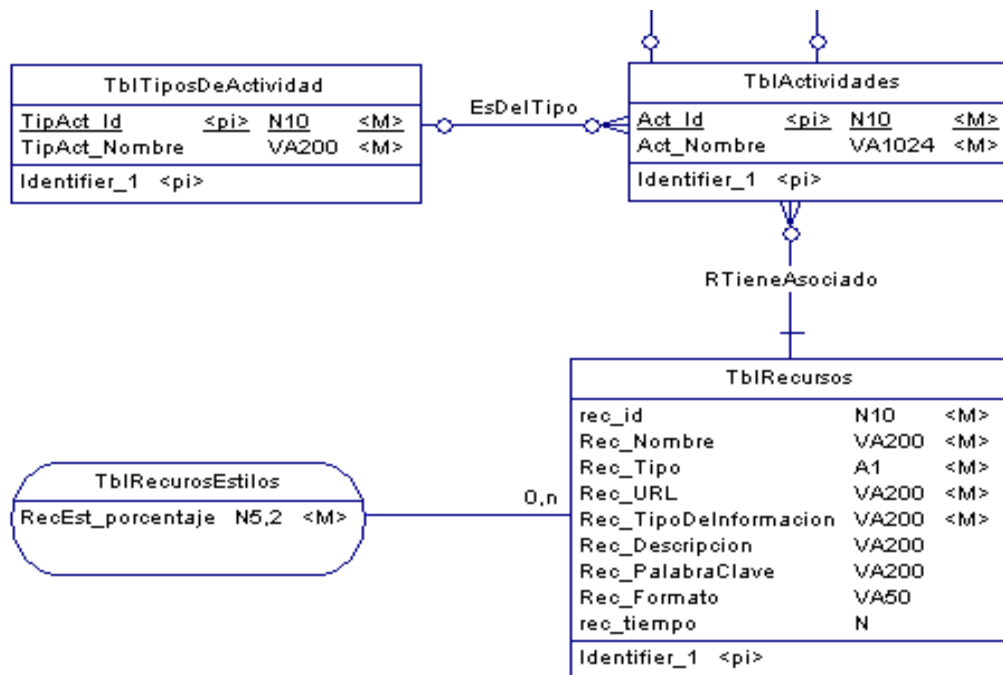


Figura 20. Modelo Gestión Actividad

4.5.5. Modelo de Datos: Crear Estructura de Contenido

El modelo de Datos Crear Estructura de Contenidos tiene 8 tablas involucradas, que cumplen un papel fundamental cada una de ellas.

El caso de uso crear estructura de contenidos incluye varios casos de uso que tienen que ver con las agregaciones, las actividades, los objetivos, las reglas de secuenciación y rollup y los objetivos asociados a cada nodo. El árbol instruccional que resulta de la unión de todos los elementos anteriores es guardada en la TblEstructuraDeContenidos nodo por nodo. Definiendo el orden de visualización inicial y los modos de secuenciación Choice, Choice Exit, Flow, Forward Only (Ver ANEXO 11).

Cada nodo del árbol es de un tipo en especial y dependiendo del tipo de nodo que sea, debe ser extraído de la tabla correspondiente, ya sea la TBLAgregaciones, la TblActividades, o la TBLObjetivos. Las reglas son creadas inicialmente por el usuario Directivo, pueden ser modificadas y actualizadas por el usuario Profesor, si el usuario no desea hacer ninguna clase remodificaciones a los modos de secuenciación inicial, la estructura personal tendrá la configuración por defecto de estos valores (*True* para Choice, *True* para Choice Exit, *False* para Flow, *False* para Forward Only) esta secuenciación y navegación propuesta al estudiante se guarda en las tablas TBLReglaRollUp y TBLReglasDeSecuenciacion.

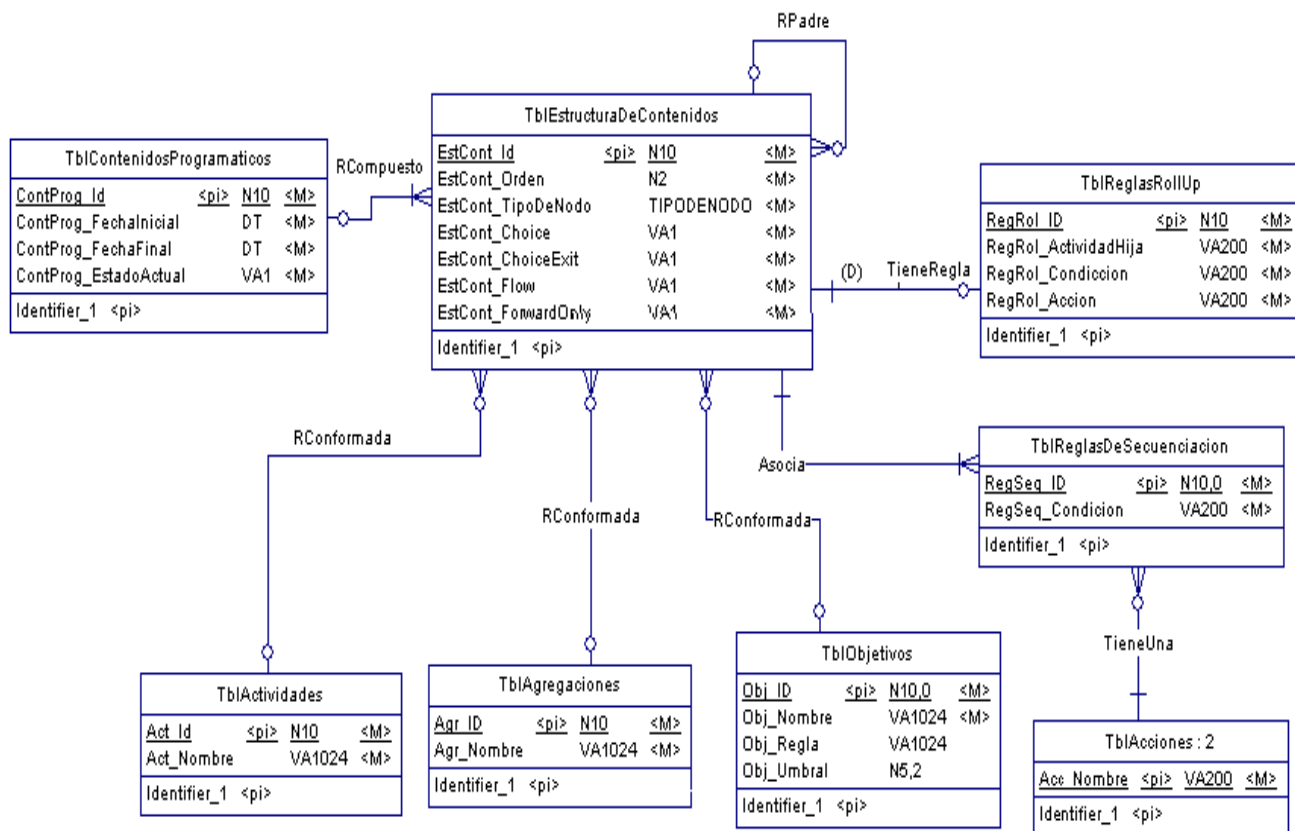


Figura 21. Modelo Crear Estructura de Contenido

La parte de los objetivos se estructura de la misma manera como se presentan las reglas de secuenciación, hay ciertas restricciones para la construcción del árbol instruccional o estructura de contenidos, entre las cuales están: Una actividad no puede ser un nodo padre ni de una agregación ni de otra actividad, Una agregación y una actividad no pueden ser hermanas, es decir, si una agregación padre tiene como hija una actividad, no se le pueden adicionar otras agregaciones a ese padre pero si todas las actividades que desee y si por el contrario una agregación padre tiene una agregación hija, no se le pueden adicionar otras actividades a ese padre pero si todas las agregaciones que desee así estas tengan nodos hijos.

CAPITULO 5. ETAPA DE DISEÑO

Durante el ciclo de desarrollo se pasa de la fase de análisis a la fase de diseño, una vez terminados los artefactos anteriormente mostrados. A continuación se mostraran algunos de los artefactos del diseño.

5.1. Descripción de la Arquitectura

La arquitectura definida para el presente proyecto es una arquitectura cliente – servidor¹² multicapas.

La arquitectura multicapas hace la división de las responsabilidades de la aplicación en más de 3 capas separando la presentación (ventanas, reportes, etc.) de la lógica de aplicaciones (tareas, reglas del negocio) y esta a su vez del almacenamiento (persistente, base de datos).

La importancia de la arquitectura multicapas radica en separar la capa de presentación haciendo que esta realice relativamente poco procesamiento de la aplicación, las ventanas enviarán a la capa intermedia peticiones de trabajo bien definidas, solicitadas por el usuario.

La capa de la lógica de aplicaciones se convierte en una capa intermedia bien definida que a su vez se divide en otras menos densas, estas capas albergan clases que representan los conceptos del dominio y los objetos servicio que son los encargados de la conexión con la capa de almacenamiento a través de las funciones como la interacción de bases de datos. Esto hace de la capa de aplicaciones un componente software reutilizable.

La capa de servicios es una capa que se encarga de toda la parte del almacenamiento permanente y también puede dividirse en subcapas con funciones específica. Esta arquitectura esta conformada específicamente por los siguientes componentes: (Ver Figura 22).

¹² La arquitectura cliente-servidor llamado modelo cliente-servidor es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia y permita simplificar las actualizaciones y mantenimiento del sistema. En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes.

En la parte del cliente se tienen:

- ✓ Los usuarios finales.
- ✓ La interfase de usuario.
- ✓ El controlador de eventos.
- ✓ La lógica de negocios local.

En la parte del servidor se tienen:

- ✓ La clase fachada.
- ✓ La lógica de negocios del servidor.
- ✓ La lógica de servicios

Cada uno de estos componentes cumple una funcionalidad clara, especializada y bien definida, permitiendo que cada componente se pueda desarrollar independientemente y acoplar sin mayores inconvenientes al finalizar el proceso de desarrollo, siendo este el mayor beneficio de tener una arquitectura multicapas bien estructurada.

A continuación se dará una breve explicación de cada uno de los componentes de la arquitectura, mencionados con anterioridad y posteriormente se hablará de los motivos por los cuales se planteo esta arquitectura para el desarrollo del presente proyecto.

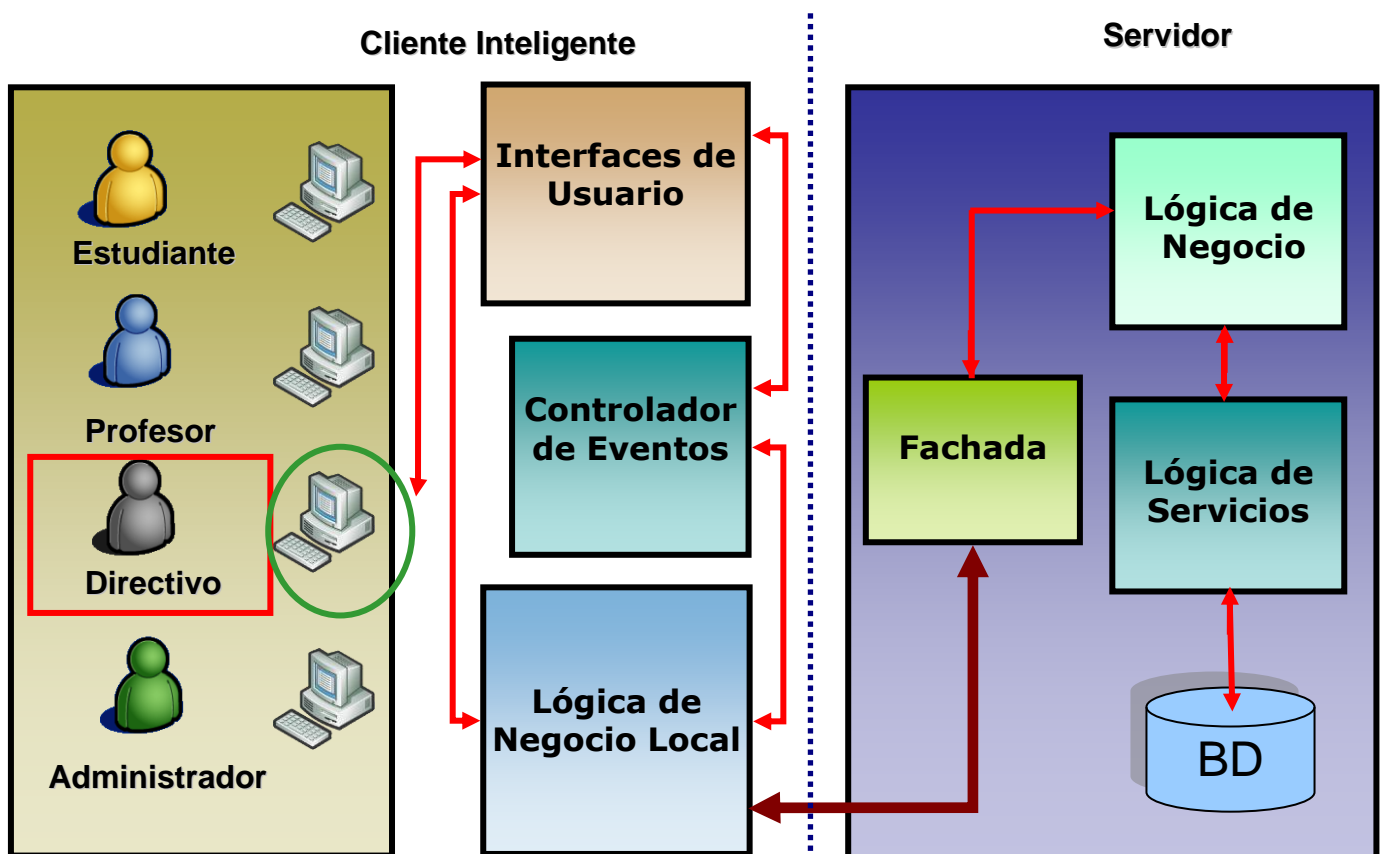


Figura 22. Arquitectura General del Sistema

5.1.1. En el Cliente

Los usuarios Finales:

- ✓ *El Administrador:* El Administrador es un actor importante en el manejo, gestión y mantenimiento adecuado de la herramienta desarrollada, pero no es un personaje relevante para el dominio del problema. Su labor es hacer todas las tareas coordinadoras para el buen funcionamiento del sistema. No pertenece al ambiente que se pretende modelar con este proyecto pues no hace parte del proceso de aprendizaje de un curso en línea.
- ✓ *El Directivo:* El directivo es un actor que pertenece al dominio del problema modelado para este proyecto, pero a diferencia del profesor y el alumno su labor es un poco más pasiva, debido a que su funcionalidad es ofrecer al profesor las herramientas adecuadas para el establecimiento de los parámetros que este último desea establecer en un curso determinado, además juega como el vocero de todos los profesores, es el encargado de establecer las reglas de juego generales y de dar a los profesores los puntos exactos donde estos puedan entrar a trabajar. El directivo se comporta como un administrador pero no de la herramienta sino del proceso pedagógico llevado a cabo por el estudiante y el profesor.
- ✓ *El Profesor:* El profesor es uno de los personajes activos en el proceso de aprendizaje del estudiante, pues es éste quien establece las condiciones iniciales que según su experiencia sean las adecuadas para dar inicio a dicho proceso, es el profesor quien tendrá la potestad de establecer la estructura que todo estudiante deberá seguir para lograr los objetivos del curso, pues aunque el aprendizaje en línea dependa casi exclusivamente de las condiciones bajo las cuales quiera trabajar el estudiante, no significa que éste no deba cumplir ciertos parámetros que como se dijo anteriormente son establecidos por el profesor.
- ✓ *El estudiante:* El estudiante es el actor principal en el desarrollo del Sistema Tutor Inteligente no solo por que es el protagonista del proceso de aprendizaje que se pretende abordar en este trabajo de investigación, sino por que además es el personaje con mayor relevancia para el dominio del problema, pues es el estudiante a quien se desea monitorear, es el estudiante el que mayor información proveerá al sistema y es el personaje sobre el cual, el sistema se basa no solo para generar las estrategias de aprendizaje, sino para evaluar el resultado de la toma de decisiones.

La interfase de usuario:

En la interfase de usuario se distinguen claramente tantas aplicaciones como usuarios finales existentes, es decir, cada usuario final (explicado en el aparte anterior) posee una aplicación independiente que le permite realizar las labores exclusivas de su rol y a la vez las labores que guardan cierta relación con las demás aplicaciones (labores definidas en conjunto). Esta capa es con la que el usuario interactúa directamente y a la que le hace las peticiones de funcionalidad.

- ✓ *La Aplicación del Administrador:* Esta aplicación permite al administrador hacer toda la labor de gestión y manejo de los usuarios, entre otras funcionalidades, cabe aclarar que esta aplicación es un componente funcional de otro de los proyectos que conforman el macroproyecto “Unicauca Virtual fase II”, por esta razón no se entrará en detalle en la descripción de la misma.
- ✓ *La Aplicación del Directivo:* La aplicación del directivo es la principal herramienta desarrollada por este proyecto, por ende para las capas subsecuentes se hablará explícitamente de las funcionalidades desarrolladas por el directivo.
La aplicación del director le permite a este llevar a cabo funciones como: Establecer los parámetros para la construcción de contenidos programáticos siguiendo la estructura que se haya definido no solo por la universidad sino por todos los profesores que se encargan de dicha labor, La creación de la estructura de contenido que se le dará al profesor para que este le agregué las actividades que considere conveniente o que según su experiencia sean las más adecuadas para iniciar su curso, hacer la gestión de información relevante como las agregaciones (temas del contenido) y la gestión de los ítems que según el proyecto educativo debe tener la plantilla de construcción de contenidos, Entre otras funcionalidades.
- ✓ *La aplicación del Profesor:* La aplicación del profesor tampoco es una herramienta objetivo del presente proyecto por eso no se entrará en detalle en su explicación. La aplicación del profesor le permite a este realizar funciones como la de establecer la estructura inicial con la cual dará inicio al curso utilizando y modificando la estructura general establecida por el directivo, el profesor también podrá escoger cuales serán las actividades que desea monitorear entre otras funcionalidades.
- ✓ *El estudiante:* En la aplicación del estudiante se llevan a cabo una gran cantidad de funcionalidades, entre las cuales se encuentran la monitorización detallada del estudiante antes, durante y después del desarrollo del curso, siguiendo de principio a fin el proceso de aprendizaje del estudiante y consignado cada cambio o característica relevante para la generación de las estrategias de aprendizaje y la toma de decisiones de la parte inteligente del tutor, todo esto se hace a través de la herramienta de agentes (otro de los

proyectos de Unicauca virtual fase II) con el que se hará acople para mostrar la total funcionalidad del Sistema Tutor Inteligente.

Controlador:

El controlador de eventos es una clase general que en principio se planteo como una clase intermedia que permitiría evidenciar el patrón de diseño “modelo vista controlador (MVC)”, sin embargo, a lo largo del desarrollo de los proyectos que conforman Unicauca Virtual, se descubrió que esta separación se convertiría en un cuello de botella, al contener código que involucrará todas las interfaces de la aplicación. Por esta razón esta clase quedo con un solo proposito, realizar las tareas que eran inevitablemente transversales a todas las clases y aplicaciones del proyecto, resumiendose a la funcionalidad de login y logout que hace el usuario desde la aplicación correspondiente. Esto se da debido a que cada aplicación hace la petición de entrada y salida del sistema y por ende cada usuario de la aplicación debe ser un usuario valido de la base de datos.

Lógica de Negocio Local

La lógica de negocios local cuenta con un conjunto de clases (una por cada elemento del dominio) que representa el modelo de negocio de la aplicación, implementando métodos para acceder a los elementos del dominio, cada una de estas clases ha sido creada con el objetivo de dar organización a todas y cada una de las funcionalidades que provee el sistema, haciendo una clasificación de las mismas de acuerdo o al usuario que las implementa o al elemento del dominio que dicha funcionalidad afecta.

Este componente de la arquitectura tiene una clase general importante llamada lógica local y es una clase que representa a través de las instancias, a todas y cada una de las demás clases que conforman la lógica de negocio local, como se explicará más adelante esta clase se comporta como fachada (patrón fachada), pues permite a los desarrolladores, acceder al conjunto de clases del dominio de una manera mucho más fácil y rápida.

Como función adicional de la clase lógica local, ésta implementa el patrón singleton permitiendo una y solo una instancia de esta clase que ejercerá el control sobre los elementos del negocio. Esta clase es instanciada por el componente interfaz cuyo objetivo es trasladar las peticiones del usuario a la capa de lógica de negocios quien a su vez hará uso de las funcionalidades de la capa de servicio a través del componente fachada cuando sea necesario.

5.1.2. En el servidor

La clase Fachada

Como se dijo en el ítem anterior, básicamente la funcionalidad de esta clase es servir como intermediaria entre la instancia de la lógica local de negocio y la lógica de servicio, se comporta de una manera muy similar a la clase lógica local del componente de la lógica de negocio de la parte del cliente, pues la clase fachada instancia cada uno de los elementos del dominio que se encuentran en la lógica de negocio de la parte del servidor, la fachada es una clase de la lógica de servicios con una interfaz común al conjunto heterogéneo de interfaces, o como lo es para este caso, clases de un paquete (Obedece al patrón fachada que se explicará más adelante). La clase fachada, permite acceder a clases del dominio que también están clasificadas de acuerdo al usuario que las necesita o al elemento del dominio que esta funcionalidad afecte, alguno de esos elementos son por ejemplo: asignaturas, usuario, etc.

La lógica de negocios del servidor

El componente de la lógica de negocios del servidor tiene una correspondencia casi total con las clases que están en la lógica de negocios local del cliente, pues no se trata de una lógica de negocios distinta, simplemente se trata de la correspondencia de la estructura de desarrollo de la lógica en cada capa de la arquitectura. En este componente el desarrollo de las funcionalidades se hace con más orientación a la capa de servicios, implementando los métodos provistos por la última capa (lógica de servicios) para el almacenamiento permanente de la información. Funcionalidad que será independiente del motor de bases de datos que se utilice pues dicha flexibilidad es provista por la lógica de servicios que es la que la implementa.

La lógica de servicios

La lógica de servicios cuenta con una clase que implementa todas las funcionalidades que la aplicación necesitará para el almacenamiento permanente de la información, esta clase ofrece la flexibilidad y la independencia del resto de la arquitectura con el motor de bases de datos que se desee utilizar, pues si se deseará implementar en un motor diferente al definido inicialmente, simplemente en este componente se hacen las adecuaciones correspondientes y el resto de la aplicación no se verá afectada.

Este componente ofrece los servicios de apertura y cerrado de la conexión a la base de datos, soporta la implementación de transacciones y la ejecución de sentencias sql para el manejo apropiado de la persistencia.

5.1.3. Motivación para la definición de la Arquitectura

Algunos de los motivos por los cuales se escogió la arquitectura multicapas son:

- Presenta aislamiento de la lógica de aplicaciones en componentes independientes susceptibles de reutilizarse después en otros sistemas.
 - Permite la distribución de las capas en varios nodos físicos (computadores) y en varios procesos, esto puede mejorar el desempeño, la coordinación y el compartir la información entre diferentes sistemas.
 - Soporta cambio en la capa de presentación sin tener fuertes repercusiones en las demás capas, permitiendo que el trabajo sea distribuido por los desarrolladores del proyecto.
 - Permite utilizar para el mismo sistema diferentes motores de bases de datos que soporten los servicios de almacenamiento permanente tan solo con adecuar la capa de servicios para tal fin.
 - Contribuye a un organizado y consistente trabajo de integración con las demás herramientas software desarrolladas para el macroproyecto “Unicauca Virtual Fase II” facilitando la consecución de uno de los objetivos específicos del presente proyecto que es la integración final al macroproyecto.[41]
- Una de las desventajas más notables de la arquitectura y que vale la pena nombrar aquí es la gran cantidad de pasos que se deben realizar en la etapa de programación, tanto para ir desde la interfaz a la capa de servicios como viceversa, esto hace de la programación una tarea tediosa, redundante y lenta.

5.1.4. Los Patrones Utilizados

A continuación se dará una breve explicación de algunos de los patrones que fueron tenidos en cuenta para el desarrollo del proyecto.

PATRON SINGLETON: El singleton es, posiblemente, el patrón más sencillo que existe. Trata las situaciones en las que solo se permite una instancia de una clase dada. Es como una visibilidad global de una clase aunque esto implique acoplamiento de paquetes. El patrón de diseño singleton (patrón unitario) está diseñado para restringir la instanciación de una clase o valor de un tipo a un solo o eventualmente muy pocos objetos. Resulta muy útil cuando por ejemplo se necesita un único objeto que coordina acciones en un sistema.

El patrón singleton se implementa creando una clase con un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado) para que no sea instanciable directamente. Como el propósito es garantizar que dicha clase solo tendrá una instancia y proporcionara un punto de acceso global a ella la propia clase es responsable de crear la única instancia de ella misma. Algunas de las ventajas que ofrece el uso

de este patrón es que puede tener un control estricto acerca de cómo y cuándo los clientes la acceden, debido a que encapsula su única instancia, además ofrece flexibilidad para realizar cambios en el proceso de instanciación, ya que es controlado por la propia clase. Algunas de las desventajas es que este patrón no contempla la eliminación de la instancia [42].

PATRON FACHADA: El patrón fachada (Facade) sirve para proveer de una interfaz unificada sencilla que haga de intermediaria entre un cliente y una interfaz o grupo de interfaces más complejas que para este caso se encuentran en el servidor.

Es un patrón estructural, estos son utilizados para formar grandes estructuras. Los patrones estructurales utilizan la herencia para componer interfases o implementaciones, su propósito es provee una interfaz a un conjunto de interfaces en un subsistema. Este patrón define un nivel más alto de interfaz que hace al subsistema más fácil de usar pues ayuda a minimizar la comunicación y las dependencias entre subsistemas ofreciendo el objeto fachada que unifica y simplifica la interfaz.

Algunas de las ventajas y desventajas son: Reducir la compilación de dependencias con Fachada puede limitar la recompilación necesitada para un pequeño cambio en un importante subsistema. Un objeto Fachada puede también simplificar la portación de sistemas a otras plataformas. No le impide a las aplicaciones el uso de las clases del subsistema si éstas las necesitan. De esta manera, se puede elegir entre facilidad de uso y generalidad [43].

PATRON MODELO VISTA CONTROLADOR: Es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El Modelo: Es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio.

Vista: Este presenta el modelo en un formato adecuado para que el usuario pueda interactuar con el, usualmente un elemento de interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista. [44]

5.2. Casos de Uso Reales

Un caso de uso real describe el diseño concreto del caso de uso, a continuación se mostrará el diseño de los casos de uso más importantes, la trazabilidad permitirá observar el curso normal del caso de uso referenciado.

5.2.1. Casos De Uso Real: Gestión Ítems Contenido

En el formulario que se muestra en la Figura 23 se pueden evidenciar algunas de las funcionalidades que la aplicación le ofrece al usuario Directivo. Este formulario permite al usuario Directivo hacer todas las operaciones respectivas a los casos de uso detallados a continuación, el formulario tiene una grilla **A** en la que se cargan todos los ítems consultados por el usuario, si la cantidad de registros es suficiente, se muestra un control de paginación de registros solo con cambiarle en **D** la cantidad de registros que se desean visualizar por pagina, para luego navegar por ellas con los botones **B** hacia atrás y **C** hacia delante, para cada registro de **A**, se puede visualizar la información completa en **F**. A través de la barra de navegación **E** se pueden hacer las operaciones de inserción, modificación, eliminación y consulta. (Ver ANEXO 22)

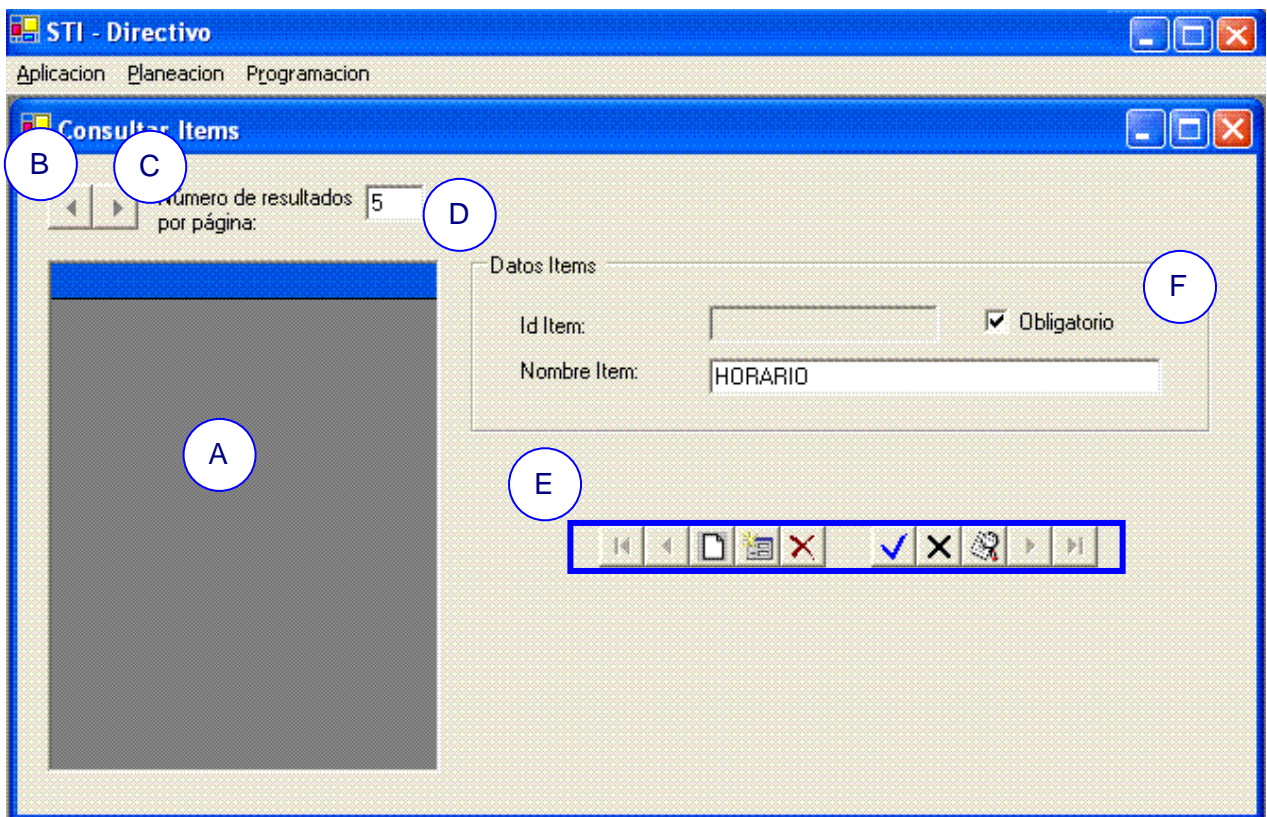


Figura 23. Diagrama Caso de Uso Real Gestión Ítems

- ☑ Caso de uso Real: Insertar Ítem Ver Tabla 57
- ☑ Caso de uso Real: Consultar Ítem Ver Tabla 58
- ☑ Caso de uso Real: Modificar Ítem Ver Tabla 59
- ☑ Caso de uso Real: Eliminar Ítem Ver Tabla 60

5.2.2. Casos De Uso Real: Gestión Contenidos Programáticos

El Formulario de la Figura 24 permite al usuario Directivo hacer todas las operaciones respectivas a los casos de uso detallados a continuación, el formulario tiene un campo **A** que es donde se cargan las asignaturas existentes e la base de datos, en el campo **B** se muestra la lista de contenidos programáticos que tiene asociada cada asignatura de **A**, para cada registro de **B**, se puede visualizar la información completa de los ítems en **C** y el estado actual del contenido **D**, es decir, si esta *ACTIVO* o *por ACTIVAR* en el primer caso se muestran también las fechas inicial y final de activación. Si los ítems consultados para la plantilla de creación de los contenidos programáticos son numeroso, se ofrece un control de paginación y a la vez un control de navegación a través de dichas paginas pulsando los botones **E** hacia atrás y **F** hacia adelante. A través de la barra de navegación **G** se pueden hacer las operaciones de Definición de Contenido, Consulta, Modificación y Eliminación. (Ver ANEXO 23)

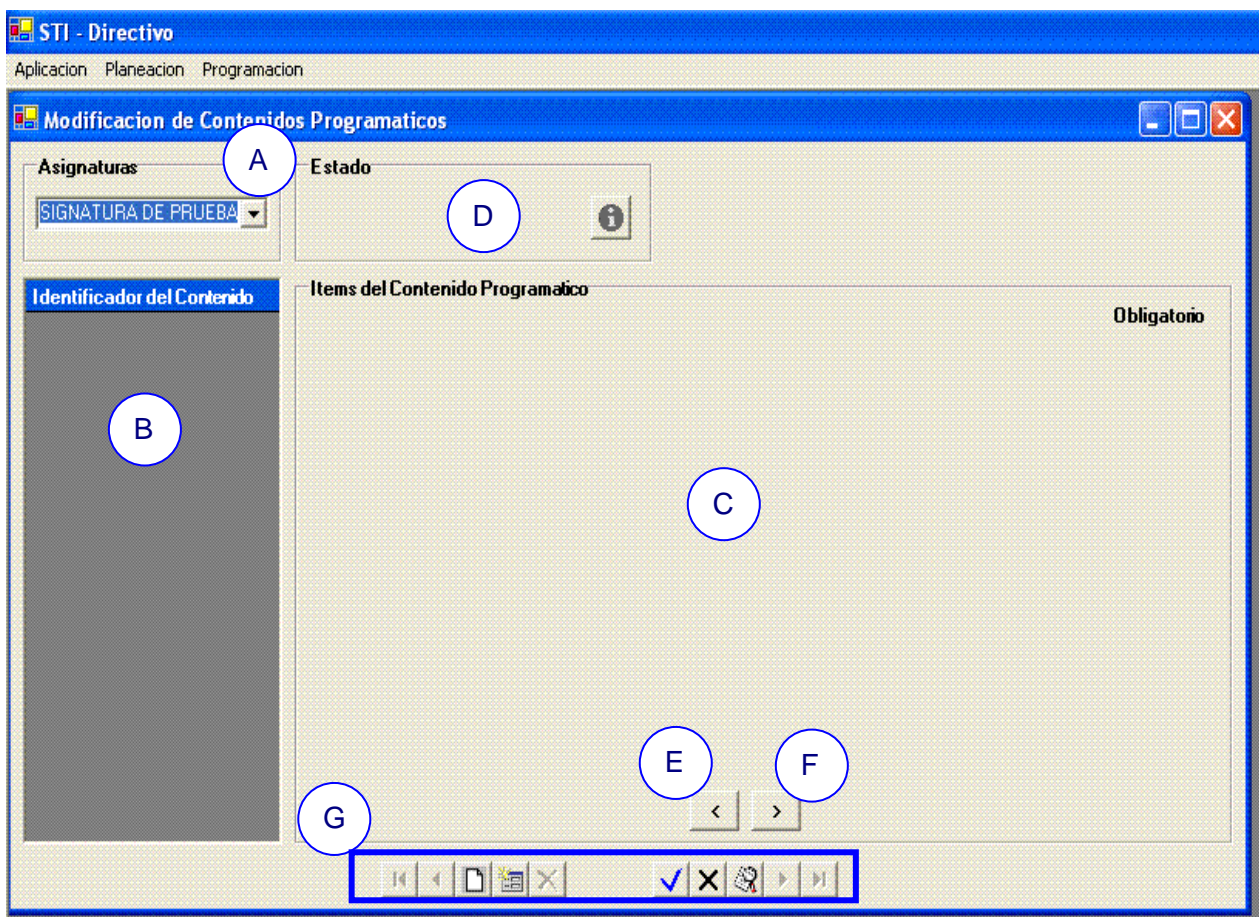


Figura 24. Diagrama Caso de Uso Real Gestión Contenidos

- Caso de Uso Real: Definir Contenido Programático → Ver Tabla 61
- Caso de Uso Real: Consultar Contenido Programático → Ver Tabla 62
- Caso de Uso Real: Activar Contenido Programático → Ver Tabla 63
- Caso de Uso Real: Modificar Contenido Programático → Ver Tabla 64

5.2.3. Casos De Uso Real: Gestión Agregaciones

Este Formulario permite al usuario Directivo hacer todas las operaciones respectivas a los casos de uso detallados a continuación, el formulario tiene un control **A** que permite al usuario Directivo realizar todas las operaciones de inserción, modificación, eliminación y consulta , en el campo **B** se muestra la lista de agregaciones consultadas por el usuario, si las agregaciones son numerosas, se ofrece un control de paginación colocando en **C** la cantidad de registros que se desean visualizar por pagina y a la vez un control de navegación a través de dichas paginas pulsando los botones **D** hacia atrás y **E** hacia adelante. En **F** se pueden visualizar los datos completos de cada registro (agregación) de **B** simplemente con seleccionarlo (Ver ANEXO 24)

- Caso de Uso Real: Insertar Agregación → Ver Tabla 65
- Caso de Uso Real: Consultar Agregación → Ver Tabla 66
- Caso de Uso Real: Modificar Agregación → Ver Tabla 67
- Caso de Uso Real: Eliminar Agregación → Ver Tabla 68

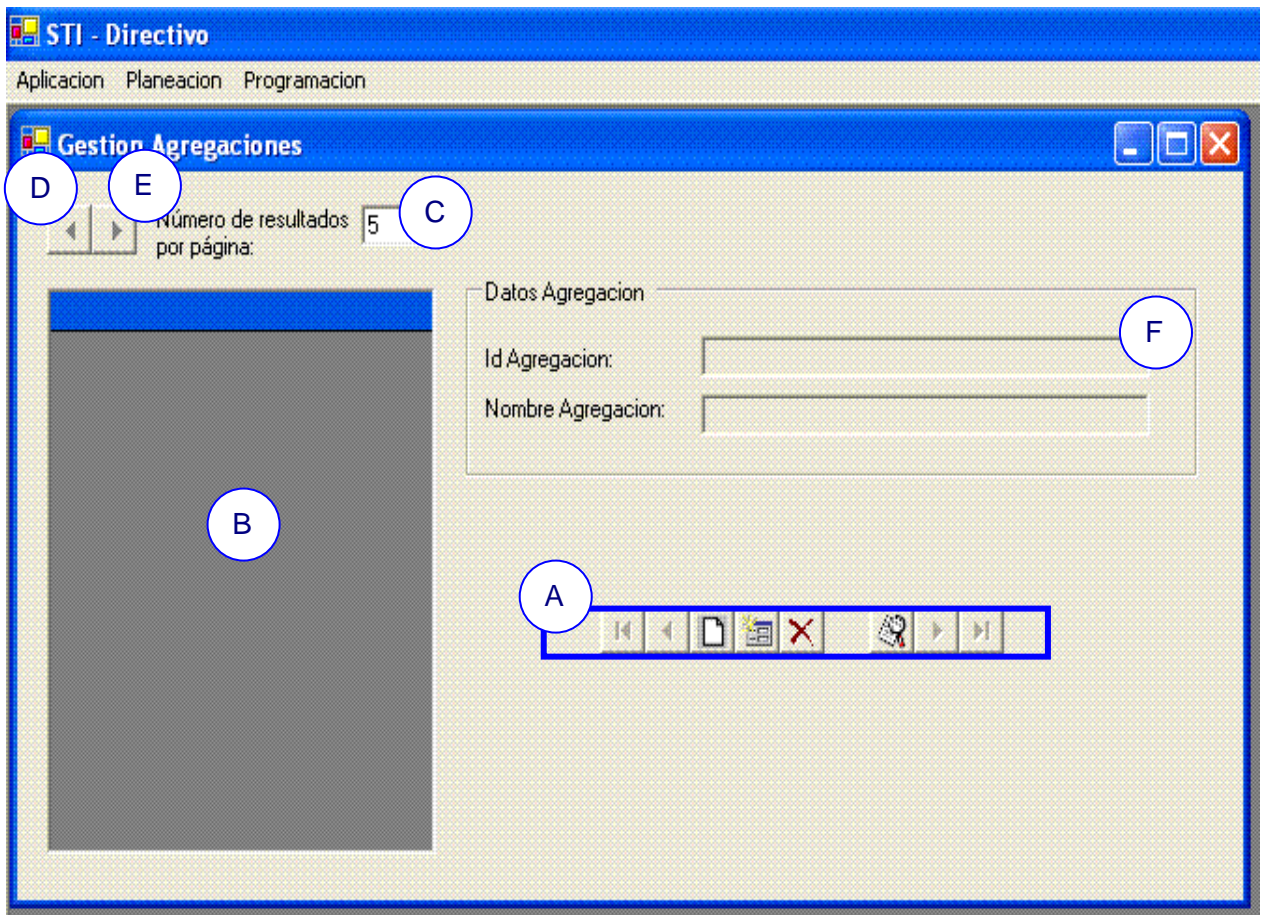


Figura 25. Diagrama Caso de Uso Real Gestión Agregaciones

5.2.4. Casos De Uso Real: Gestión Actividades

Este Formulario permite al usuario Directivo hacer todas las operaciones respectivas a los casos de uso detallados a continuación, el formulario tiene un control **A** que permite al usuario Directivo realizar todas las operaciones de inserción, modificación, eliminación y consulta, en el campo **B** se muestra la lista de actividades consultadas por el usuario, si las actividades son numerosas, se ofrece un control de paginación colocando en **C** la cantidad de registros que se desean visualizar por pagina y a la vez un control de navegación a través de dichas paginas pulsando los botones **D** hacia atrás y **E** hacia adelante. En **F** se pueden visualizar los datos completos de cada registro (actividad) de **B** simplemente con seleccionarlo (Ver ANEXO 25)

- ☑ Caso de Uso Real: Insertar Agregación → Ver Tabla 69
- ☑ Caso de Uso Real: Consultar Agregación → Ver Tabla 70
- ☑ Caso de Uso Real: Modificar Agregación → Ver Tabla 71
- ☑ Caso de Uso Real: Eliminar Agregación → Ver Tabla 72

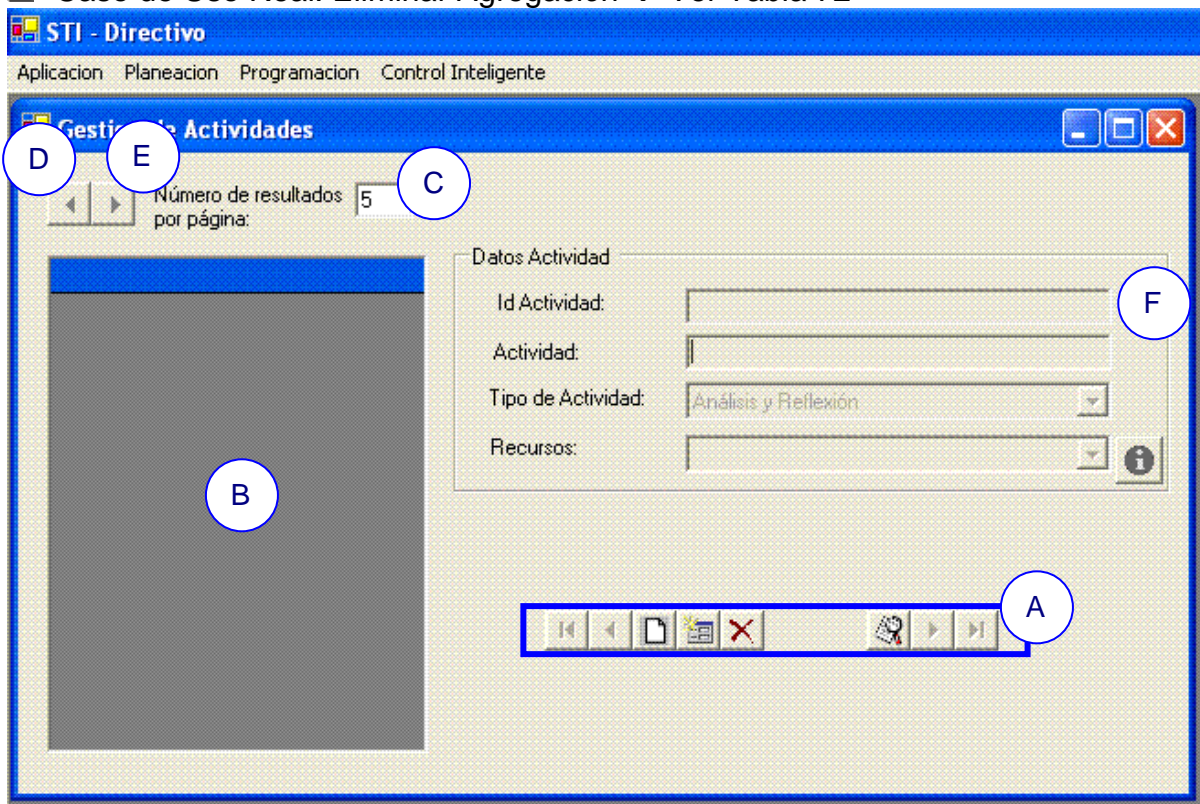
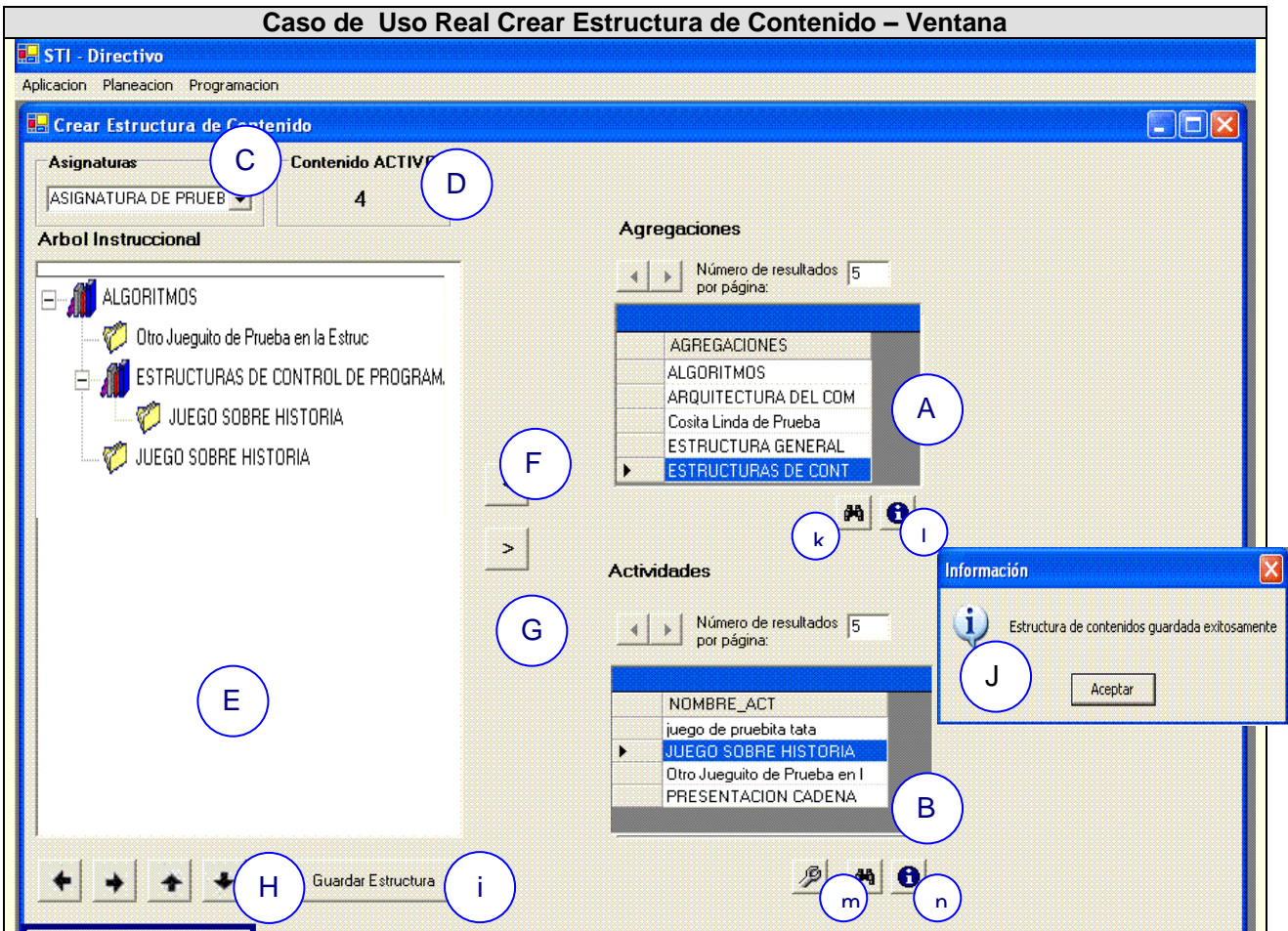


Figura 26. Diagrama Caso de Uso Real Gestión Actividades

5.2.5. Caso De Uso Real: Crear Estructura de Contenidos



CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción crear estructura de contenidos del menú	2. Se carga un formulario con el campo A para las agregaciones que pueden ser consultadas o gestionadas con los botones adicionales K y L respectivamente. El campo B para las agregaciones que también pueden ser consultadas o gestionadas con M y N respectivamente; y el campo C con todas las asignaturas de la base de datos.
3. El usuario escoge de la lista de asignaturas mostradas en C , la asignatura a la que desea vincular la estructura de contenidos a crear.	4. El sistema inmediatamente consulta el contenido programático que está activo para la asignatura escogida y lo muestra en D

<p>5. El usuario empieza a armar la estructura en forma de árbol en E usando F y G ya sea para escoger las agregaciones y las actividades disponibles o para retirarlas del árbol si lo desea, la organización de la estructura podrá manejarla el usuario utilizando cualquiera del conjunto de elementos de H, finalmente manda a guardar la estructura creada pulsando I.</p>	<p>6. Guarda en la base de datos la nueva estructura de contenido y retorna un mensaje de éxito como J</p>
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> ▪ Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Caso de Uso CrearEstructuraContenido – Expandido] 	

Tabla 17. Caso de Uso Real Crear Estructura de Contenido

CAPITULO 6. RESULTADOS OBTENIDOS

Este capítulo muestra un reporte de los resultados obtenidos del proceso de desarrollo partiendo de los objetivos planteados al inicio por los desarrolladores, incluyendo algunos de los problemas tenidos en la consecución de dichos objetivos. Así como los productos palpables del trabajo realizado.

6.1. Objetivos Planteados

✓ *Desarrollar un modulo inteligente que integre técnicas de Inteligencia Artificial con conceptos de educación virtual y Estrategias de Aprendizaje para un Sistema Tutor Inteligente.*

Los resultados obtenidos del estudio del estado del arte en el área de Inteligencia Artificial arrojaron el desarrollo de un modelo de diagnóstico y generación de estrategias de aprendizaje basado en los conceptos de representación de conocimiento relacional, árboles de decisión y reglas lógicas de primer orden. Se trata de un modelo descrito en los capítulos 1 y 2 de este documento. Este, a grandes rasgos, permite diagnosticar a partir de las experiencias obtenidas por el propio sistema nuevas estrategias de aprendizaje para obtener la mejor de acuerdo a los perfiles de los estudiantes.

✓ *Generar estrategias de enseñanza y aprendizaje a partir de reglas de conocimiento para la presentación personalizada de los contenidos en un ambiente de educación en línea.*

En este objetivo las reglas de conocimiento que se utilizaron como materia prima para la generación de las estrategias de aprendizaje se cambiaron por experiencias de aprendizaje generadas dinámicamente por el sistema, es decir, utilizamos los conceptos de los sistemas expertos pero con la suficiencia en la adquisición que estos no tienen lo que hace que el sistema se “auto-entrene” a partir de las propias experiencias que va obteniendo de la interacción con los estudiantes. Fruto de este entrenamiento y de algunas reglas lógicas de primer orden que se definieron se generan las estrategias de aprendizaje personalizadas a los perfiles de los estudiantes. Este objetivo tiene un valor agregado gracias al desarrollo del modulo capaz de auto-entrenarse, esto da características mucho mas dinámicas que si se definen las estrategias a partir de reglas de conocimiento, ya que estas estarían estáticas y su creación y mantenimiento seria bastante complicado y solo podría (o debería) ser hecho por un experto, el cual no se tiene en la realización de este proyecto. Por lo tanto, los cambios realizados en este objetivo en particular en lugar de ir de detrimento de los resultados aumentan su importancia.

✓ *Utilizar las técnicas de Inteligencia Artificial para representar el conocimiento de los estudiantes de forma tal que permita el proceso de toma de decisiones que involucran dicho conocimiento.*

Este objetivo se logra gracias a la representación de conocimiento relacional, uno de los enfoques de la ingeniería de conocimiento más usados en aplicaciones prácticas en la actualidad y que gracias a desarrollos en áreas como el Data Mining y OLAP han logrado grandes avances en los últimos años, ya que se han aumentado las potencialidades de este en gran medida. Además, se puede decir que se logró una representación que desde cualquier punto de vista es satisfactoria, ya que es sumamente sencilla, favorece enormemente el proceso de toma de decisiones y facilita los cuatro principios básicos de la representación de conocimiento, características que eran indispensables en el desarrollo de este proyecto.

✓ *Desarrollar un prototipo que permita realizar la validación de los resultados del proyecto*

El prototipo desarrollado para la validación de los resultados consiste específicamente en una aplicación software cuya funcionalidad representa el proceso de planeación y elaboración de un curso virtual, así como el proceso de toma de decisiones frente a las características de los estudiantes, la herramienta desarrollada está dirigida a ayudar al usuario directivo a realizar sus principales tareas, como hacer la gestión de las estructuras de contenidos que se presentarán al estudiante para que este interactúe con ella a lo largo del desarrollo del curso virtual, así como la gestión de las agregaciones (temas), actividades y recursos asociados a dicha estructura; también muestra como a través del módulo inteligente se puede hacer la toma de decisiones. Para observar pantallazos de la aplicación y el módulo inteligente ver ANEXO 36

✓ *Integrar la herramienta Software a la fase II del macroproyecto Unicauca Virtual*

Para la consecución de este objetivo fue definida, como primera medida, una arquitectura básica explicada detalladamente en la sección 5.1, esta arquitectura fue establecida en conjunto con todos los proyectos que conforman Unicauca Virtual Fase II.

El proceso de integración de la herramienta al macroproyecto ha sido una tarea ardua y de constante retroalimentación y trabajo en equipo, por esta razón vale la pena aclarar que se establecieron estándares de programación para que cualquier modificación de las clases que se comparte sean claras, además de la pertinente documentación de la misma, para que cualquier equipo de trabajo pueda reutilizar el código generado por otro equipo. La consecución de este objetivo se evidenciará con la entrega de la aplicación con las funcionalidades objetivo de cada proyecto.

6.2. Tabla de Resultados

OBJETIVOS PLANTEADOS	RESULTADOS	INDICADORES Escala (* 100)	OBSERVACIONES
<p>1. Desarrollar un modulo inteligente que integre técnicas de Inteligencia Artificial con conceptos de educación virtual y Estrategias de Aprendizaje para un Sistema Tutor Inteligente.</p>	<p>Los resultados obtenidos del estudio del estado del arte en el área de Inteligencia Artificial arrojó el desarrollo de un modelo de diagnostico y generación de estrategias de aprendizaje basado en los conceptos de representación de conocimiento relacional, árboles de decisión y reglas lógicas de primer orden. Se trata de un metamodelo descrito en la monografía del proyecto "Inteligencia Artificial en la generación de estrategias de aprendizaje para un Sistema Tutor Inteligente" en su capítulo 6. Esté, a grandes rasgos, permite diagnosticar a partir de las experiencias obtenidas por el propio sistema nuevas estrategias de aprendizaje para obtener la mejor de acuerdo a los perfiles de los estudiantes y a reglas lógicas definidas a partir de la experiencia lograda por los integrantes del proyecto en lo que se refiere a pedagogía en general, y específicamente a la pedagogía de la educación virtual.</p>	$IE = \frac{NoTeoriasRelacionadas}{NoTeoriasARelacionar} * 100$ $IE = \frac{3}{3} * 100 = 100\%$ <p>*<i>Conocimiento Relacional</i> *<i>Arboles de Decisión</i> *<i>Lógica de primer Orden</i></p>	<p>Para poder encontrar un modelo genérico y utilizable en el contexto de Unicauca Virtual se hizo necesario realizar una intensa búsqueda dentro de los diferentes enfoques de Inteligencia Artificial (IA) para encontrar el apropiado. Esta labor fue realmente complicada por que los enfoques de IA son bastante diversos y no existe dentro de todo el estudio e investigación en la materia ningún tipo de manual que exponga a un enfoque en particular como el mas apropiado para utilizar en un aplicación con características particulares, por lo tanto se hizo necesario realizar una primera investigación de las características propias de cada enfoque buscando confrontarlas con las características de la aplicación que deseábamos desarrollar. Esto dejo como fruto una serie de informes que se presentan en el sitio Web del proyecto y el desarrollo del artículo "Modelos de Ranuras en la generación de Estrategias de Aprendizaje de un Sistema Tutor Inteligente" que se encuentra en el mismo sitio.</p>
<p>2. Generar estrategias de enseñanza y aprendizaje</p>	<p>En esta objetivo las reglas de conocimiento que se utilizaron como materia prima para la</p>	$IE = \frac{NoTeoriasRelacionadas}{NoTeoriasARelacionar} * 100$	<p>Este es el objetivo que mas cambios tuvo a lo largo del desarrollo del</p>

<p>a partir de reglas de conocimiento para la presentación personalizada de los contenidos en un ambiente de educación en línea.</p>	<p>generación de las estrategias de aprendizaje se cambiaron por experiencias de aprendizaje generadas dinámicamente por el sistema, es decir, utilizamos los conceptos de los sistemas expertos pero con la suficiencia en la adquisición que estos no tienen lo que hace que el sistema se “auto-entrene” a partir de las propias experiencias que va obteniendo de la interacción con los estudiantes. Fruto de este entrenamiento y de algunas reglas lógicas de primer orden que se definieron se generan las estrategias de aprendizaje personalizadas a los perfiles de los estudiantes</p>	$IE = \frac{1}{1} * 100 = 100\%$ <p><i>Como se explicó en la sección 1.2.3 de la presente monografía la teoría o modelo de estilos de aprendizaje a Relacionar en el proyecto fue CHAEA</i></p> <p><i>Sin embargo se deja abierta la posibilidad para que a futuro se haga la inclusión de nuevos modelos a la aplicación.</i></p>	<p>proyecto, ya que en este punto es que de debía integrar la aplicación con el otro proyecto vinculado al Sistema Tutor Inteligente (STI). Los problemas surgen al integrar las salidas de las tareas de monitoreo del proyecto “AGENTES INTELIGENTES PARA ADAPTAR LA PRESENTACION DE CONTENIDOS PARA CURSOS DE EDUCACIÓN EN LÍNEA” con el modulo inteligente que se desarrolló en este proyecto. Este fue la razón por la cual se decidió cambiar algunas de las ideas que en principio se tenían para generar las estrategias de aprendizaje buscando facilitar la comunicación entre los dos proyectos. Se cambio la generación de un perfil de estudiante por medio de estructuras de ranuras a sistemas de representación de conocimiento relacional para facilitar las tareas tanto de generación de las estrategias de aprendizaje (tarea de este proyecto), como las de monitoreo del estudiante (tarea del proyecto mencionado anteriormente)</p>
<p>3. Utilizar las técnicas de Inteligencia Artificial para representar el conocimiento de los estudiantes de forma tal que permita el proceso de toma de decisiones que involucran dicho conocimiento.</p>	<p>Este objetivo se logro gracias a la representación de conocimiento relacional, uno de los enfoques de la ingeniería de conocimiento mas usados en aplicaciones practicas en la actualidad y que gracias a desarrollos en áreas como el Datamining y OLAP han logrado grandes avances en los últimos años, ya que se han aumentado las potencialidades de este en gran medida. Además, podemos decir que logramos una representación que desde cualquier punto</p>	$IE = \frac{NoTeoriasRelacionadas}{NoTeoriasARelacionar} * 100$ $IE = \frac{2}{4} * 100 = 50\%$ <p><i>*Modelos de Ranuras</i> <i>*Modelos Relacionales</i></p>	<p>En la investigación realizada de las características de las aplicaciones inteligentes rápidamente salieron a la luz dos muy importantes, un mecanismo de representación de conocimiento y un mecanismo de inferencia. Para el mecanismo de inferencia era importante que la representación del conocimiento tuviera cuatro características importantes, por la tanto decidimos</p>

	<p>de vista es satisfactoria, ya que es sumamente sencilla, favorece enormemente el proceso de toma de decisiones y facilita los cuatro principios básicos de la representación de conocimiento, características que eran indispensables para el desarrollo de este proyecto.</p>		<p>que la mejor forma de representar el conocimiento era a través de estructuras de ranuras que en ese momento brindaba los mejores acercamientos a lo que se necesitaba, pero al momento de integrar con el proyecto que debía monitorear al estudiante para ampliar su perfil nos dimos cuenta que esta representación traería muchos problemas para el desarrollo de los dos proyectos. Dado este problema se decidió buscar una representación que fuera más flexible para los dos proyectos y fue entonces donde encontramos la representación relacional.</p>
<p>4. Desarrollar un prototipo que permita realizar la validación de los resultados del proyecto</p>	<p>El prototipo desarrollado para la validación de los resultados consiste específicamente en una aplicación software cuya funcionalidad representa el proceso de planeación y elaboración de un curso virtual, así como el proceso de toma de decisiones frente a las características de los estudiantes, la herramienta desarrollada esta dirigida a ayudar al usuario directivo a realizar sus principales tareas, como hacer la gestión de las estructuras de contenidos que se presentarán al estudiante para que este interactúe con ella a lo largo del desarrollo del curso virtual, así como la gestión de las agregaciones (temas), actividades y recursos asociados a dicha estructura; también muestra como a través del modulo inteligente se puede hacer la toma de decisiones.</p>	$IC = \frac{NoUsuariosBeneficiados}{NoUsuariosABeneficiar} * 100$ $IC = \frac{3}{3} * 100 \quad IC = 100$ <p><i>Usuarios Beneficiados:</i></p> <ul style="list-style-type: none"> - <i>Directivo</i> - <i>Directamente</i> - <i>Profesor</i> - <i>Indirectamente</i> - <i>Estudiante</i> - <i>Indirectamente</i> $IE = \frac{NoRe cursosEjercidos}{NoRe cursosAsignados} * 100$ $IE = \frac{6}{6} * 100 = 100\%$ <p><i>IQ = ¿Cumple la herramienta con los requisitos mínimos de</i></p>	<p>El prototipo desarrollado implementa el modulo de toma de decisiones haciendo uso de características pertenecientes al modelo de aprendizaje CHAEA, pero para no limitar el trabajo desarrollado la herramienta permite extender la funcionalidad y hacer uso de nuevos modelos de aprendizaje.</p> <p>El desarrollo de la interfaz del prototipo ha sido desarrollado siguiendo estándares establecidos para facilitar la labor de integración del mismo con la herramienta del macroproyecto Unicauca Virtual Fase II.</p>

<p>5. Integrar la herramienta Software a la fase II del macroproyecto Unicauca Virtual</p>	<p>Para la consecución de este objetivo se definió una arquitectura básica definida para todos los proyectos de Unicauca Virtual Fase II, sino por que la integración ha sido una tarea constante de retroalimentación y trabajo en equipo, sin embargo aun no hay una integración total, debido a que aun el prototipo se encuentra en construcción.</p>	<p><i>funcionalidad? 3 = 75%</i></p> $IF = \frac{MetasAlcanzadas}{Re cursosConsumidos} * 100$ $IF = \frac{4}{5} * 100 \quad IF = 80$ $IC = \frac{NoUsuariosBeneficiados}{NoUsuarioqueEsperaServir} * 100$ $IC = \frac{3}{3} * 100 = 100\%$ <p><i>Usuarios Beneficiados:</i></p> <p><i>Directivo</i> -</p> <p><i>Directamente</i></p> <p><i>Profesor</i> -</p> <p><i>Indirectamente</i></p> <p><i>Estudiante</i> -</p> <p><i>Indirectamente</i></p> <p><i>IQ = ¿Está la herramienta integrada en su totalidad al macroproyecto Unicauca Virtual Fase II? 3 = 70%</i></p>	
--	---	---	--

6.3. Lineamientos de Conformación e Indicadores

Los indicadores de desempeño que se evalúan básicamente adoptan la forma de un cociente, en el cual, el denominador es un valor numérico que ayuda a efectuar la comparación con el logro obtenido así:

$$\text{Indicador} = \left(\frac{\text{Numerador}}{\text{Denominador}} \right) * \text{Factor Escala}$$

De esta forma se definen los siguientes modelos de indicadores que se deben personalizar y aplicar a los actores, dependiendo de su contexto y funciones definidas:

1. **Indicador de Cobertura (IC):** Determina la cantidad de elementos cobijados por un producto o estrategia.

$$\text{Cobertura} = \left(\frac{\text{No. de nodos Beneficiados con el Servicio}}{\text{Número de nodos que se esperaba Servir}} \right) * 100$$

2. **Indicador de Eficacia (IE):** Cumplir con los Requisitos definidos.

$$\text{Eficacia} = \left(\frac{\text{Recursos Ejercidos}}{\text{Recursos Asignados}} \right) * 100$$

3. **Indicador de Eficiencia (IF):** Permite identificar la relación que existe entre las metas alcanzadas, tiempo y recursos consumidos con respecto a un estándar. Buen uso de los recursos.

$$\text{Eficiencia} = \left(\frac{\text{Metas Alcanzadas}}{\text{Recursos Consumidos}} \right) * 100$$

4. **Indicador de Calidad (IQ):** Están orientados a medir la satisfacción de los beneficiarios.

Indicador = Calificación entre (1: Mala (0%), 2: Regular (50%), 3: Buena (75%), 4: Excelente (100%))

6.4. Productos

Como resultado del proceso de desarrollo, se presenta en primera instancia la presente monografía con la recopilación de la información recogida en el proceso

investigativo y el análisis la misma, además de las bases conceptuales y el seguimiento del proceso de desarrollo del proyecto, referente a los artefactos del análisis, diseño y construcción de la herramienta propuesta.

Se logró definir un modelo de toma de decisiones, involucrando conceptos de Inteligencia Artificial, Educación en Línea y conceptos de clasificación de perfiles o estilos de aprendizaje, para que los aspectos propios del estudiante sean los factores más relevantes en la toma de decisiones en cuanto a la estrategia de aprendizaje a presentar.

Se desarrollo un modulo inteligente software basado en el modelo de toma de decisiones desarrollado en este proyecto, éste modulo inteligente realiza un proceso de personalización que se ajusta más a las características propias del estudiante para sugerir una estrategia de aprendizaje según los conceptos tratados en este proyecto.

Se construyo un prototipo cuya funcionalidad cumple con los requerimientos establecidos por el macroproyecto de Unicauca Virtual en el que se encuentra inmerso este trabajo de grado. Permitiendo al usuario directivo realizar algunas de las tareas de planificación y elaboración de un curso en línea, además del control inteligente sobre el proceso de aprendizaje del estudiante a través de los dos productos anteriores.

Dos artículos redactados por los desarrolladores en las etapas iniciales del proceso de desarrollo que aportaron una visión un poco más clara del camino a tomar en la investigación, presentados para su publicación en los siguientes eventos:

- CELDA 2004 (Cognition and Exploratory Learning in Digital Age), Lisbon – Portugal, 15-17 Dic/04. <http://www.iadis.org/celda2004/>
- CLEI 2005 (Centro Latinoamericano de Estudios en Informática) evento anual, Cali – Colombia, 10-14 Oct/05 <http://aniversario60.univalle.edu.co/bitacora-grafica/octubre/clei/foto1.html>

CAPITULO 7. CONCLUSIONES Y RECOMENDACIONES

7.1. CONCLUSIONES

A continuación se darán a conocer las conclusiones a las que se llegaron al culminar el proceso de desarrollo del proyecto.

El modelo de toma de decisiones desarrollado hace algunos aportes significativos en cuanto al uso de conceptos de inteligencia artificial como los árboles de decisión como mecanismos de inferencia cuando son más comúnmente usados como mecanismo de clasificación.

El módulo inteligente desarrollado mejora su desempeño al poseer características de auto-entrenamiento, el cual se logra a través de las experiencias de aprendizaje de los estudiantes durante el desarrollo de los cursos, además, es completamente dinámico al permitir ingresar más atributos que pueden ser tenidos en cuenta para la toma de decisiones.

La herramienta desarrollada permite al usuario Directivo realizar ciertas tareas básicas de planeación y programación de un curso virtual, además de visualizar las acciones del módulo inteligente en su proceso de toma de decisiones.

En la fase de integración se encontraron varios problemas de pequeña magnitud, lo que nos permite corroborar que a pesar de lo tediosa que puede ser la tarea de programación con la arquitectura implementada, está ofreció una mayor estandarización y por ende mayor facilidad a la hora de la integración con la herramienta de Unicauca Virtual Fase II.

La formación de los desarrolladores ha sido enormemente enriquecida con todo el proceso de investigación realizado a lo largo del desarrollo de este proyecto.

7.2. RECOMENDACIONES

Las recomendaciones dadas de acuerdo a la experiencia obtenida durante el desarrollo del presente proyecto son:

La herramienta se desarrollo teniendo en cuenta el modelo de estilos de aprendizaje CHAEA por la experimentación y efectividad que según varios estudios posee dicho modelo, Por tal motivo se pueden hacer adecuaciones al modulo inteligente de la herramienta desarrollada, incluyendo muchos más atributos que tengan que ver no solo con el sistema, sino en si con el estudiante a través de la inclusión de otros tipos de clasificación o modelos de aprendizaje.

Debido a que el modelo desarrollado para la toma de decisiones del modulo inteligente hace referencia a conceptos de un área poco explorada en la educación. Se propone una segunda versión de la herramienta que tenga como objetivo hacer todas las validaciones pertinentes con cursos, experiencias y estudiantes reales durante un largo tiempo que permita validar que la herramienta efectivamente soluciona el problema de la calidad y la cobertura de la educación, soportando un programa de estudio completo y aprovechando toda la funcionalidad que el modulo ofrece.

En la escogencia del algoritmo inteligente para la construcción del árbol de decisión se utilizo el ID3 por las diferentes características del proyecto y las características propias del algoritmo. Sabemos que existen algunas evoluciones con respecto al ID3 actualmente, por lo tanto resultaría interesante evaluar las posibilidades de estas evoluciones para agregar nuevas características al proyecto, como la inclusión de valores de atributos en conjuntos continuos y no discretos, o la clasificación de las experiencias en más de dos clases

Igualmente , cuando seleccionamos la regla de división que utilizaría el algoritmo ID3 (Ganancia de Información - Entropía) se descarto la posibilidad de utilizar alguna otra regla de división que pueda generar árboles de decisión mas aproximados a las características de este ambiente. Aconsejamos realizar un poco mas de investigación en este sentido, utilizando otras reglas de división dentro de la generación del árbol de decisión.

Existen muchos temas dentro del contexto del E-learning en donde se podría agregar de forma muy fácil y efectiva el modelo desarrollado, tales áreas incluyen los Sistemas de Hipermedia Adaptativa, los sistemas de razonamiento basado en casos CBR y los mismos Sistemas Tutores Inteligentes. Seria muy interesante explorar más a fondo las posibilidades de este modelo en estas áreas, a través de proyectos de investigación que incluyan más tiempo de experimentación puesto que las bases conceptuales ya se encuentran planteadas.

CAPITULO 8. REFERENCIAS BIBLIOGRAFICAS

- [1]. Marcus, A, 2000. Caracterización de los servicios de la educación a distancia desde la óptica de una plataforma distribuida orientada a objetos. *Tesis de Maestría, Instituto Tecnológico y de Estudios Superiores de Monterrey*, Monterrey, México, pp 2-5
- [2]. Marcus, A., 2002. Sobre la Interoperabilidad de Herramientas de Educación a Distancia a través de Internet. *CITA2002, II Congreso Iberoamericano de Telemática*, Mérida, Venezuela.
- [3]. *EAN Facultad de Estudios a Distancia. Modelo Pedagógico EAN, [en línea]. Bogotá D.C. Colombia. Disponible en: <http://fed.ean.edu.co/www/section-606.jsp#224> [2006, 23 de Enero]*
- [4]. De Zubiría, J., (1994). El Currículo y los Modelos Pedagógicos, Los Modelos Pedagógicos Contemporáneos y la pedagogía Conceptual, Que son los Modelos Pedagógicos, El Currículo Educativo. *En Tratado de pedagogía conceptual Los Modelos Pedagógicos* (Pág. 38 - 125). Bogota DC, Colombia
- [5]. “*Proposiciones sobre la Educación*”, Alain. citado por Louis Not (1983).
- [6]. Quintero, P.X. y CULCHAC, W.E., *Aproximación a las prácticas pedagógicas de la Universidad del Valle Sede Zarza*. Trabajo de Grado, Ciencias de la Administración, Universidad del Valle, Zarzal – Colombia.
- [7]. Unigarro, M. A., (2001). El Concepto de Aprendizaje. En *Educación Virtual Encuentro formativo en el Ciberespacio* (Pág. 95). Bucaramanga, Colombia.
- [8]. Recopilación Marqués, P., *Glosario de Tecnología Educativa* [en línea]. Disponible en <http://dewey.uab.es/pmarques/glosario.htm> [2006, 14 de Marzo]
- [9]. Díaz Barriga, F. y Hernández, G., *Estrategias docentes para un aprendizaje significativo*. Una interpretación constructivista. 2ª. ed. México: McGraw Hill, 2002.
- [10]. Renteria Avila M. J., Estrategias de aprendizaje. [en línea] Disponible en: <http://www.monografias.com/trabajos14/decisiones-aprendizaje/decisiones-aprendizaje.shtml#conjunt> [2006, 15 de Marzo]
- [11]. Gaudioso, E., 2002. *Contribuciones al Modelado del Usuario en Entornos Adaptativos de Aprendizaje y Colaboración a través de Internet mediante*

técnicas de Aprendizaje Automático. Tesis Doctoral, Universidad Complutense de Madrid, Madrid - España.

- [12]. Gagné, R., (1970). Fundamentos y Enfoques de la Educación y del Aprendizaje. [en línea]. Disponible en: <http://cdu.cnc.una.py/docs/cnc/grupos/gagne/body.html> [2006, 20 de Febrero]
- [13]. Salinas, J., (1997). Enseñanza Flexible, Aprendizaje Abierto. Las redes como herramienta para la formación. [en línea]. Disponible en: http://www.ieev.uma.es/edutec97/edu97_c1/2-1-00.htm [2006, 26 de Enero]
- [14]. Alonso, (1994). *Los estilos de aprendizaje: procedimientos de diagnóstico y mejora*, Ediciones Mensajero, Bilbao, España, pp. 104-116.
- [15]. Cazau, P., Estilos de aprendizaje: Generalidades [en línea]. Disponible en: <http://www.itnl.edu.mx/docs/material21/EstilosAprendizaje/Lecturas/Estilos%20de%20aprendizaje%20Generalidades.pdf> [2006, 30 de Enero]
- [16]. Paule, M., De La Puente, A., Perez, J.R. y González, M., *Una manera de adaptación a los estilos de aprendizaje teóricos y activos con preferencia alta y muy alta con una herramienta informática*[en línea].Disponible en: <http://www.tecnoneet.org/docs/2002/3-62002.pdf> [2006, 25 de Enero]
- [17]. Robles, A., Aprender a Aprender [en línea]. Disponible en: <http://www.galeon.com/aprenderaaprender/general/indice.html#estilos> [2006, 30 de Enero]
- [18]. Baus Roset, T., Los Estilos de Aprendizaje [en línea]. Disponible en: <http://www.monografias.com/trabajos12/loestils/loestils.shtml#ambiental> [2006, 30 de Enero]
- [19]. Unigarro, M. A., 2001, *Educación Virtual Encuentro formativo en el Ciberespacio*, Bucaramanga, Colombia.
- [20]. Pozo, I., (1996). *Aprendices y Maestros*. La Nueva Cultura del Aprendizaje. Madrid: Alianza.
- [21]. Habermas, J., Conocimiento e Interés, *Ciencia y técnica como ideología*. [en línea]. Disponible en: <http://usuarios.lycos.es/Cantemar/Conocimiento.html> [2006, 30 de Enero]
- [22]. Habermas, J., (1988). *Teoría de la Acción Comunicativa*. Madrid: Taurus.

- [23]. Simonson, M., Smaldino, S., Albright, M. Y Zvacek, S., (2003). Definitions, History, and theories of Distance Education, *Teaching and learning at a Distance (second edition)*. Merrill Prentice Hall, Columbus, Ohio.
- [24]. Clark R. y Mayer R. E., (2003). E-Learning: Promise and Pitfalls, *e-Learning and the science of instruction*. Pfeiffer a Wiley imprint, United States.
- [25]. Urretavizcaya, M., (2001). "Sistemas Inteligentes en el ámbito de la Educación", [http://tcs.uned.es:8080/aepia/articulosMismoAutorVer.do?autor=Asociación Española para la Inteligencia Artificial \(AEPIA\)](http://tcs.uned.es:8080/aepia/articulosMismoAutorVer.do?autor=Asociación%20Española%20para%20la%20Inteligencia%20Artificial%20(AEPIA)) [En línea]. Edición Electrónica: Dpt. Lenguajes y Sistemas Informáticos U.N.E.D. Universidad Politécnica de Valencia. Disponible en: http://tcs.uned.es:8080/aepia/articulosPorNumeroVer.do?id_numero=12 [2005, 13 de Agosto]
- [26]. Gutiérrez, J.; Pérez, T. A.; Lopistéguy, P. Y Usandizaga, I. (1995): *Sistemas Tutores Inteligentes: una forma de conseguir Sistemas Hipermedia Educativos*, CAEPIA'95.
- [27]. Rich, E. y Knight, K., (1994). *Inteligencia Artificial (segunda edición)*. Mc Graw Hill, Madrid, España.
- [28]. *Representación del conocimiento en sistemas inteligentes* (2004), [en línea]. Madrid, España: Universidad Politécnica de Madrid, Gregorio F.F. Disponible en: <http://turing.gsi.dit.upm.es/~gfer/ssii/rcsi/index.html> [2005, 22 de Octubre].
- [29]. Winston, P.H., (1994), *Inteligencia Artificial*. Addison-Wesley Iberoamericana, Buenos Aires, Argentina
- [30]. Sontag, E.D., (1972), *Temas de Inteligencia Artificial*. Prolam S.R.I., Buenos Aires, Argentina
- [31]. *Advanced Distributed Learning* [en línea]. Washington, DC: U.S. Government. Disponible en: www.adlnet.org [2005, 15 de Octubre].
- [32]. *IMS Global Learning Consortium: Simple Sequencing Specification* [en línea]. Burlington, USA: IMS Global Learning Consortium. Disponible en: <http://www.imsglobal.org/simplesequencing/index.html> [2006, 15 de Enero].
- [33]. Rebollo, M. (2004). *El estándar SCORM para EaD*. Tesina del Master en Enseñanza y Aprendizaje Abiertos y a Distancia. Universidad Nacional de Educación a Distancia. Valencia, España.
- [34]. MINSKY, M.: *A Framework for Representing Knowledge* [en línea]. MIT AI Lab, June 1974. Publicado como Capítulo 6 en P. H. Winston (ed.): *The*

- Psychology of Computer Vision*, McGraw-Hill, 1975. Disponible en <http://web.media.mit.edu/~minsky/papers/Frames/frames.html>. [2005, 10 de Agosto]
- [35]. *Data Mining Technologies* [en línea]. Melville, New York: Data Mining Technologies. Disponible en: <http://www.data-mine.com/bin/site/templates/splash.asp> [2006, 18 de Enero].
- [36]. *TodoSobre.org* [en línea]. Mexico DF, Mexico: Wikipedia. Disponible en: <http://www.todosobre.org/Algoritmo-greedy> [2005, 13 de Noviembre].
- [37]. BREIMAN, L., FRIEDMAN, J. H., OLSHEN, R. A. & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, California, USA, 1984.
- [38]. BERZAL GALIANO F. (2003). *ART: Un método alternativo para la construcción de árboles de decisión*. Universidad de Granada, 2003.
- [39]. QUINLAN, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [40]. SHANNON C. E. (1949). *A mathematical Theory of Communication*. Bell System Technical Journal. Disponible en: <http://cm.bell-labs.com/cm/ms/what/shannonday/> [2005, 15 de Diciembre]
- [41]. Craig, L., 1999. *UML y Patrones Introducción al Análisis y Diseño Orientado a Objetos (Primera Edición)*. Prentice Hall, Naucalpan de Juárez, México.
- [42]. Wales, J. y Sanger, L., Singleton Patrón de Diseño, *Wikipedia la enciclopedia libre* [en línea]. Disponible en: http://es.wikipedia.org/wiki/Singleton_%28patr%C3%B3n_de_dise%C3%B1o%29 [2006, 16 de Febrero]
- [43]. Wales, J. y Sanger, L., Facade Patrón de Diseño, *Wikipedia la enciclopedia libre* [en línea]. Disponible en: http://es.wikipedia.org/wiki/Facade_%28patr%C3%B3n_de_dise%C3%B1o%29 [2006, 16 de Febrero]
- [44]. Wales, J. y Sanger, L., MVC Patrón de Diseño, *Wikipedia la enciclopedia libre* [en línea]. Disponible en: http://es.wikipedia.org/wiki/Modelo_Vista_Controlador [2006, 19 de Marzo]
- [45]. *IEEE Learning Technology Standards Committee* [en línea]. Chicago, EUA: IEEE. Disponible en: <http://ieeeltsc.org/wg11CMI/1484.11.2/> [2006, 8 de Febrero]

- [46]. *IMS Global Learning Consortium: Content Packaging Specification* [en línea]. Burlington, USA: IMS Global Learning Consortium. Disponible en: <http://www.imsglobal.org/content/packaging/> [2006, 15 de Enero]
- [47]. *IEEE Learning Technology Standards Committee* [en línea]. Chicago, EUA: IEEE. Disponible en: <http://ieeeltsc.org/wg12LOM/1484.12.1/> [2006, 8 de Febrero].
- [48]. Booch, G., Jacobson, I. y Rumbaugh, J. (1997). The UML specification documents. Santa Clara, CA,: Rational Software Corp. Véanse los documentos en www.rational.com
- [49]. Wales, J. y Sanger, L., *Wikipedia la enciclopedia libre* [en línea]. Disponible en <http://es.wikipedia.org/wiki/Cliente-servidor> [2006, 16 de Febrero]
- [50]. Marina, J. A., (2000). *Crónicas de la Ultramodernidad*. Barcelona: Anagrama.
- [51]. Inmon, W.H. (1996). *Building the Data Warehouse* (2da ed.). Wiley Computer Publishing.
- [52]. Kimball, R. (1998). *The Data Warehouse Toolkit: Practical techniques for building dimensional Data Warehouse*. Wiley Computer Publishing.

ANEXOS

ANEXO 1. Estilos de Aprendizaje – CHAEA.

Catalina M. Alonso junto con Peter Honey proponen el cuestionario –CHAEA- que está compuesto por ochenta ítems. El individuo ha de responder para cada pregunta si está de acuerdo o en desacuerdo y a partir de la ponderación de las respuestas se podrá determinar el estilo de aprendizaje del individuo.

ESTILOS DE APRENDIZAJE - CHAEA	
Activos	Se refiere a alumnos que se involucran totalmente y sin perjuicios a las experiencias nuevas y disfrutan el momento presente. Son personas abiertas, entusiastas, sin prejuicios ante las nuevas experiencias, incluso aumenta su motivación ante los retos.
Reflexivos	Son alumnos que tienden a adoptar una postura de observador que analiza sus experiencias desde muchas perspectivas distintas. Recogen datos y los analizan detalladamente antes de llegar a una conclusión. Les gusta escuchar, se muestran cautos, discretos e incluso a veces quizá distantes.
Teóricos	Los alumnos teóricos adaptan e integran las observaciones que realizan en teorías complejas y bien fundamentadas lógicamente. Piensan de forma secuencial. Buscan la racionalidad, la objetividad, la precisión y la exactitud.
Pragmáticos	Son alumnos que les gusta probar ideas, teorías y técnicas nuevas y comprobar si funcionan en la práctica. Buscan la rapidez y eficacia en sus acciones y decisiones. Se muestran seguros cuando se enfrentan a los proyectos que les ilusionan.

Tabla 18. Clasificación Estilos de Aprendizaje CHAEA

El cuestionario CHAEA.

Universidad de Deusto. Instituto de Ciencias de la Educación (ICE). CHAEA. Cuestionario Honey-Alonso de Estilos de Aprendizaje. Autores: Catalina M. Alonso, Domingo J. Gallego y Peter Honey

Instrucciones para responder al cuestionario

- Este cuestionario ha sido diseñado para identificar su Estilo preferido de Aprendizaje. No es un test de inteligencia , ni de personalidad
- No hay límite de tiempo para contestar al Cuestionario. No le ocupará más de 15 minutos.
- No hay respuestas correctas o erróneas. Será útil en la medida que sea sincero/a en sus respuestas.

- Si está más de acuerdo que en desacuerdo con el ítem seleccione 'Mas (+)'. Si, por el contrario, está más en desacuerdo que de acuerdo, seleccione 'Menos (-)'.
- Por favor conteste a todos los ítems.
- El Cuestionario es anónimo.

Muchas gracias.

Cuestionario HONEY-ALONSO de Estilos de Aprendizaje: CHAEA

A continuación mostraremos algunas de las preguntas del cuestionario

Cuestiones	+	-
1. Tengo fama de decir lo que pienso claramente y sin rodeos.		
2. Estoy seguro/a de lo que es bueno y lo que es malo, lo que está bien y lo que está mal.		
3. Muchas veces actúo sin mirar las consecuencias.		
4. Normalmente trato de resolver los problemas metódicamente y paso a paso.		
5. Creo que los formalismos coartan y limitan la actuación libre de las personas.		
6. Me interesa saber cuáles son los sistemas de valores de los demás y con qué criterios actúan.		
7. Pienso que el actuar intuitivamente puede ser siempre tan válido como actuar reflexivamente.		
8. Creo que lo más importante es que las cosas funcionen.		
9. Procuero estar al tanto de lo que ocurre aquí y ahora.		
10. Disfruto cuando tengo tiempo para preparar mi trabajo y realizarlo a conciencia.		
11. Estoy a gusto siguiendo un orden, en las comidas, en el estudio, haciendo ejercicio regularmente.		
12. Cuando escucho una nueva idea enseguida comienzo a pensar cómo ponerla en práctica.		
13. Prefiero las ideas originales y novedosas aunque no sean prácticas.		
14. Admito y me ajusto a las normas sólo si me sirven para lograr mis objetivos.		
15. Normalmente encajo bien con personas reflexivas, y me cuesta sintonizar con personas demasiado espontáneas, imprevisibles.		
16. Escucho con más frecuencia que hablo.		
17. Prefiero las cosas estructuradas a las desordenadas.		
18. Cuando poseo cualquier información, trato de interpretarla bien antes de manifestar alguna conclusión.		
19. Antes de hacer algo estudio con cuidado sus ventajas e inconvenientes.		

20. Me crezco con el reto de hacer algo nuevo y diferente.		
21. Casi siempre procuro ser coherente con mis criterios y sistemas de valores. Tengo principios y los sigo.		
22. Cuando hay una discusión no me gusta ir con rodeos.		
23. Me disgusta implicarme afectivamente en mi ambiente de trabajo prefiero mantener relaciones distantes.		
24. Me gustan más las personas realistas y concretas que las teóricas.		
25. Me cuesta ser creativo/a, romper estructuras.		
26. Me siento a gusto con personas espontáneas y divertidas.		
27. La mayoría de las veces expreso abiertamente cómo me siento.		
28. Me gusta analizar y dar vueltas a las cosas.		
29. Me molesta que la gente no se tome en serio las cosas.		
30. Me atrae experimentar y practicar las últimas técnicas y novedades		
31. Soy cauteloso/a a la hora de sacar conclusiones.		
32. Prefiero contar con el mayor número de fuentes de información. Cuantos más datos reúna para reflexionar, mejor.		
33. Tiendo a ser perfeccionista.		
34. Prefiero oír las opiniones de los demás antes de exponer la mía.		
35. Me gusta afrontar la vida espontáneamente y no tener que planificar todo previamente.		
36. En las discusiones me gusta observar cómo actúan los demás participantes.		
37. Me siento incómodo/a con las personas calladas y demasiado analíticas.		
38. Juzgo con frecuencia las ideas de los demás por su valor práctico.		
39. Me agobio si me obligan a acelerar mucho el trabajo para cumplir un plazo.		
40. En las reuniones apoyo las ideas prácticas y realistas.		
41.....		

ANEXO 2. Organización de la Información

Aprender no consiste en almacenar datos aislados. El cerebro humano se caracteriza por su capacidad de relacionar y asociar la gran cantidad de información que recibe continuamente y buscar pautas y crear esquemas que nos permitan entender el mundo que nos rodea. Pero no todos seguimos el mismo procedimiento, y la manera en que organicemos esa información afectará a nuestro estilo de aprendizaje.

Cada hemisferio procesa la información que recibe de distinta manera, es decir, hay distintas formas de pensamiento asociadas con cada hemisferio. Según como organicemos la información recibida, podemos distinguir entre[18]:

ORGANIZACIÓN DE LA INFORMACIÓN	
Reptilito o Central	Funcionamiento automático, lado de la motricidad, sexualidad, comida, conservación. Característica principal el pragmatismo.
Derecho	Lado creativo, emocional y sensible. Se conoce como el hemisferio holístico, procesa la información de manera global, partiendo del todo para entender las distintas partes que componen ese todo. Es intuitivo en vez de lógico, piensa en imágenes y sentimientos. Emplea un estilo de pensamiento divergente, creando una variedad y cantidad de ideas nuevas, más allá de los patrones convencionales.
Izquierdo	Procesos racionales, lado analítico, crítico y lógico. Es verbal y maneja símbolos y representaciones virtuales de la realidad. Es el hemisferio lógico, procesa la información de manera secuencial y lineal. Forma la imagen del todo a partir de las partes y es el que se ocupa de analizar los detalles. El hemisferio lógico piensa en palabras y en números, es decir contiene la capacidad para la matemática y para leer y escribir. Emplea un estilo de pensamiento convergente obteniendo nueva información al usar datos ya disponibles, formando nuevas ideas o datos convencionalmente aceptables.

Tabla 19. Clasificación Organización de la Información

ANEXO 3. Inteligencias Múltiples

La inteligencia es la capacidad para resolver problemas cotidianos, para generar nuevos problemas y para crear productos o para ofrecer servicios dentro del propio ámbito cultural.

La teoría de las inteligencias múltiples propone que La mayoría de los individuos tenemos la totalidad de este espectro de inteligencias. Cada una desarrollada de modo y a un nivel particular, producto de la dotación biológica de cada uno, de su interacción con el entorno y de la cultura imperante en su momento histórico. Las combinamos y las usamos en diferentes grados, de manera personal y única. Estas Inteligencias son[17]:

INTELIGENCIAS MÚLTIPLES	
Musical	Es la capacidad de percibir, discriminar, transformar y expresar las formas musicales. Incluye la sensibilidad al ritmo, al tono y al timbre.
Corporal	O cinestésica es la capacidad para usar todo el cuerpo en la expresión de ideas y sentimientos, y la facilidad en el uso de las manos para transformar elementos. Incluye habilidades de

	coordinación, destreza, equilibrio, flexibilidad, fuerza y velocidad, así como la percepción de medidas y volúmenes.
Lingüística	Es la capacidad de usar las palabras de manera efectiva, en forma oral o escrita. Incluye la habilidad en el uso de la sintáxis, la fonética, la semántica y los usos pragmáticos del lenguaje.
Lógico Matemática	Es la capacidad para usar los números de manera efectiva y de razonar adecuadamente. Incluye la sensibilidad a los esquemas y relaciones lógicas, las afirmaciones y las proposiciones, las funciones y otras abstracciones relacionadas.
Espacial	Es la capacidad de pensar en tres dimensiones. Permite percibir imágenes externas e internas, recrearlas, transformarlas o modificarlas, recorrer el espacio o hacer que los objetos lo recorran y producir o decodificar información gráfica.
Interpersonal	Es la capacidad de entender a los demás e interactuar eficazmente con ellos. Incluye la sensibilidad a expresiones faciales, la voz, los gestos y posturas y la habilidad para responder.
Intrapersonal	Es la capacidad de construir una percepción precisa respecto de sí mismo y de organizar y dirigir su propia vida. Incluye la autodisciplina, la autocomprensión y la autoestima.
Naturalista	Es la capacidad de distinguir, clasificar y utilizar elementos del medio ambiente, objetos, animales o plantas. Tanto del ambiente urbano como suburbano o rural. Incluye las habilidades de observación, experimentación, reflexión y cuestionamiento de nuestro entorno.

Tabla 20. Clasificación Inteligencias Múltiples

ANEXO 4. Según el Sistema de Representación

Clasifica a las personas según las preferencias sensoriales (o sistemas representativos dominantes) La mayoría de nosotros utilizamos los sistemas de representación de forma desigual, potenciando unos e infra-utilizando otros. Los sistemas de representación se desarrollan más cuanto más los utilizamos. La persona acostumbrada a seleccionar un tipo de información absorberá con mayor facilidad la información de ese tipo o, planteándolo al revés, la persona acostumbrada a ignorar la información que recibe por un canal determinado no aprenderá la información que reciba por ese canal, no porque no le interese, sino porque no está acostumbrada a prestarle atención a esa fuente de información. Utilizar más un sistema implica que hay sistemas que se utilizan menos y, por lo tanto, que distintos sistemas de representación tendrán distinto grado de desarrollo [17]

Los sistemas de representación no son buenos o malos, pero si más o menos eficaces para realizar determinados procesos mentales. A continuación se

especifican las características de cada uno de estos tres sistemas. Pérez Jiménez J (2001)

SISTEMA DE REPRESENTACION	
Visual	Recuerdan información visual, Aprende lo que ve. Necesita una visión detallada y saber a donde va. Le cuesta recordar lo que oye, detallista, posee una excelente escritura, le gustan las descripciones, En la ortografía No tiene faltas. "Ve" las palabras antes de escribirlas, Piensa en imágenes. Visualiza de manera detallada se imagina la escena., producción oral limitada, Almacena información rápidamente en cualquier orden.
Auditivo	Obra oral brillante, sabe escuchar, es conversador, poseen espíritu de lideres, Aprende lo que oye, a base de repetirse a si mismo paso a paso todo el proceso. Si se olvida de un solo paso se pierde. No tiene una visión global, evita las descripciones largas, mueve los labios y no se fija en las ilustraciones, en la ortografía Comete faltas. "Dice" las palabras y las escribe según el sonido, almacena información De manera secuencial y por bloques enteros (por lo que se pierde si le preguntas por un elemento aislado o si le cambias el orden de las preguntas.
Kinestésico	Hacen para aprender, sensibles, imaginativos y vivenciales, integra bondades de estilos táctiles, cinestésico, olfativo y gustativo [17]. Aprende con lo que toca y lo que hace. Necesita estar involucrado personalmente en alguna actividad. No es un gran lector. En la ortografía Comete faltas. Escribe las palabras y comprueba si "le dan buena espina". Almacena información mediante la "memoria muscular".

Tabla 21. Concepción según el sistema de representación

ANEXO 5. Según la Categoría Bipolar (Felder-Silverman)

Clasifica los estilos de aprendizaje de los estudiantes a partir de cinco dimensiones, las cuales están relacionadas con las respuestas que se puedan obtener a las siguientes preguntas:

Qué tipo de información percibe mejor el estudiante: ¿Sensorial o intuitiva?

A través de qué modalidad percibe más efectivamente la información sensorial: ¿Visual o verbal?

Cómo prefiere el estudiante procesar la información que percibe: ¿Activamente o reflexivamente?

Cómo logra entender el estudiante: ¿Secuencialmente o globalmente?

El estilo de aprendizaje de un estudiante vendrá dado por la combinación de las respuestas obtenidas en las cinco dimensiones. A continuación se exploran las características de aprendizaje de los estudiantes en las cinco dimensiones del modelo.

CATEGORIA BIPOLAR	
Activo/Reflexivo	<i>Activos:</i> Retienen y comprenden mejor nueva información cuando hacen algo activo con ella (discutiéndola, aplicándola, explicándosela). Prefieren aprender ensayando y trabajando con otros. <i>Reflexivos:</i> Entienden información pensando y reflexionando sobre ella, gusta del trabajo individual.
Sensorial/Intuitivo	<i>Sensoriales:</i> Son concretos, prácticos, orientados hacia hechos y procedimientos, de buena memoria. <i>Intuitivos:</i> Son conceptuales, innovadores, orientados hacia las teorías y los significados, odian las repeticiones, trabajan bien con abstracciones.
Visual/Verbal	<i>Visuales:</i> en la obtención de información prefieren representaciones visuales, diagramas de flujo. <i>Auditivos:</i> prefieren obtener la información en forma escrita o hablada; recuerdan lo que leen u oyen.
Secuencial/Global	<i>Secuenciales:</i> Aprenden en pequeños pasos incrementales paso a paso, ordenados y lineales. <i>Globales:</i> Aprenden en grandes saltos, al azar y «de pronto» visualizando la totalidad

Tabla 22. Concepción según la categoría bipolar

ANEXO 6. El diseño de Instrucción Lineal

Se basa en instrucciones programadas linealmente. Primero, un área de contenido como *fundamentos de la educación a distancia* se puede dividir en 12 ideas importantes. Estas ideas pueden ser llamadas módulos, los módulos de instrucción pueden dividirse en temas y cada tema es un evento instruccional o experiencia de aprendizaje seguida por algún tipo de valoración. Antes a que los estudiantes se les permita continuar con el siguiente tema deben tener una valoración satisfactoria para el modulo actual, si la valoración se realiza por medio de un test de objetivos ellos deben ganar el test. La secuencia de los eventos instruccionales relacionados por temas va seguida de continuas valoraciones hasta que todos los temas de un modulo sean estudiados y aprobados por el estudiante. Normalmente los estudiantes no podrán acceder al siguiente módulo hasta que las valoraciones del modulo actual no estén terminadas. Igualmente podrán ser valorados a mitad del curso o cuando este termine para evaluar la

capacidad de los estudiantes para sintetizar toda la información recibida a través de todos los temas del módulo.

El diseño instruccional lineal es completamente secuencial. Los estudiantes siempre se mueven por el mismo camino a través de conceptos, temas y módulos y cumplen las mismas valoraciones y test's. Ver Figura 27.

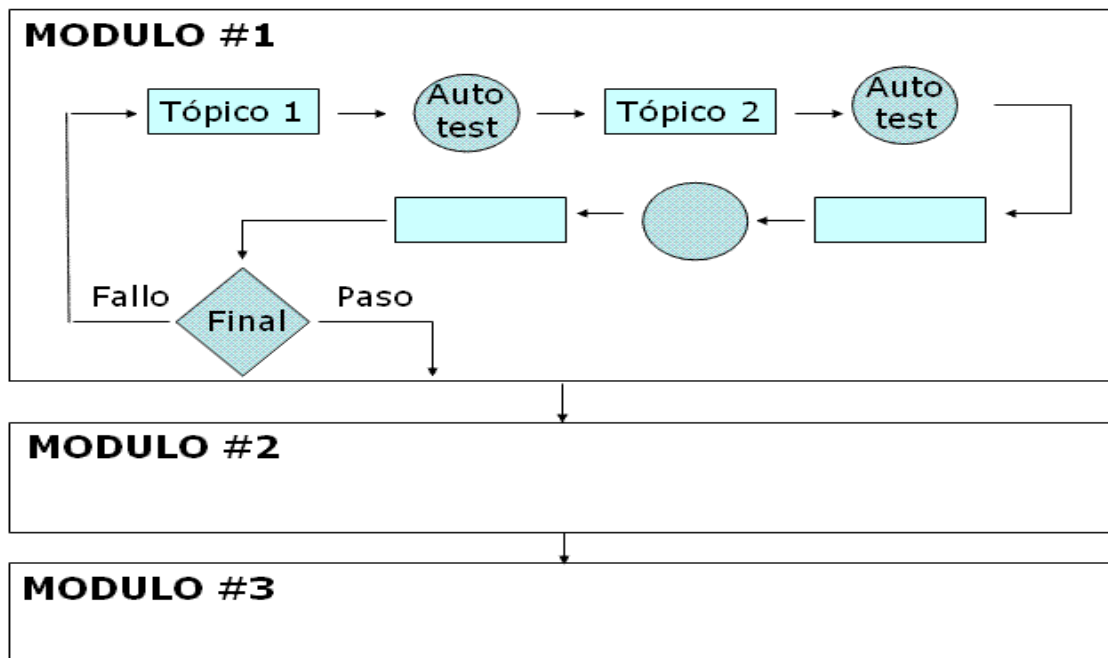


Figura 27. Diseño Instruccional Lineal

ANEXO 7. Diseño instruccional Ramificado (Branched)

Es similar al lineal con dos diferencias importantes. La primera es que su forma de evaluación es más sofisticada es el sentido que diagnostica el progreso de los estudiantes y su entendimiento de los conceptos y temas. Si un estudiante muestra más interés en algunos temas es posible saltar hacia temas mas adelante o “ramificar” (Branched, de ahí su nombre) a través de los demás módulos de la materia. Igualmente, si un estudiante tiene algún problema con el proceso de evaluación requerirá que él ramifique hacia atrás, es decir, se devuelva hasta los temas en donde presente problemas o que se diseñe desde la planeación una instrucción “terapéutica” o de nivelación antes que el estudiante pueda pasar a la siguiente lección.

La segunda diferencia importante de esta forma de diseño instruccional es el uso de eventos instruccionales o experiencias de aprendizaje alternativos. En otras palabras, los estudiantes podrán navegar los módulos a través de diferentes

actividades instruccionales de acuerdo a los resultados de sus evaluaciones. Justo como un tutor humano puede decidir que un estudiante de algebra necesita mas practica con matemáticas, una lección diseñada con este modelo puede sugerir a un estudiante que realice por un tiempo una lección de entrenamiento y practica. El diseño instruccional Branched es complicado de plantear y desarrollar pero su efectividad en tiempos de ejecución o de utilización es muy alto, ver Figura 28.

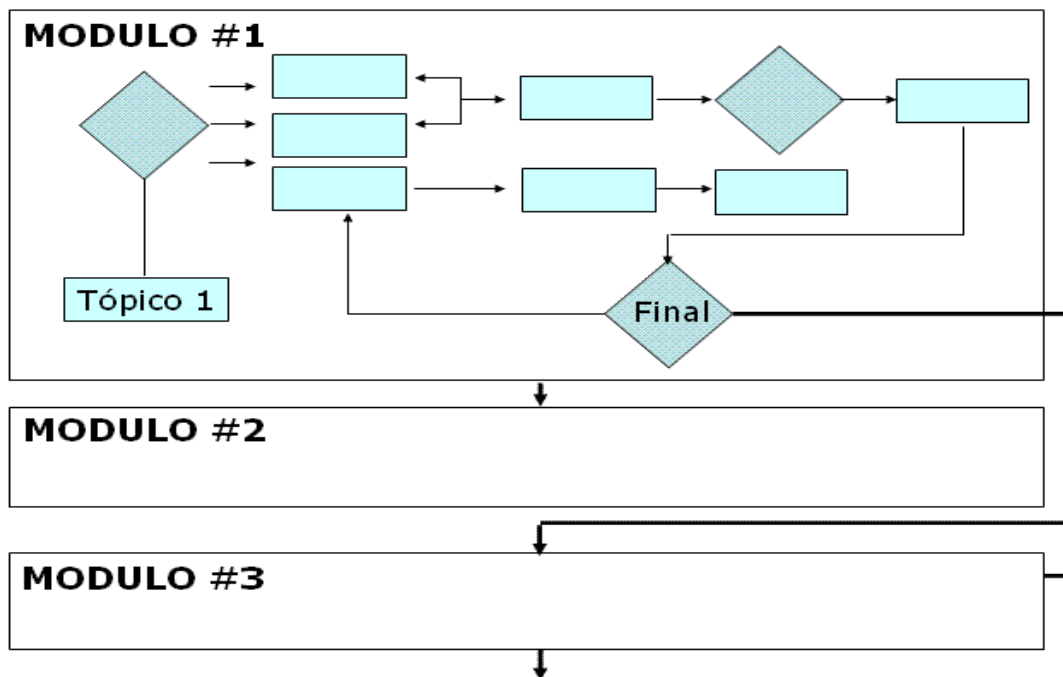


Figura 28. Diseño instruccional Ramificado

ANEXO 8. Diseño instruccional de Hypercontenidos (Hypercontent)

Al igual que todos los demás tiene módulos, temas y conceptos. Primero, los módulos son identificados; posteriormente son identificados los temas relacionados con el modulo y las experiencias de aprendizaje son diseñadas y producidas. Estos temas son presentados usando textos, audio, graficas, dibujos y videos. Finalmente, las actividades de valoración del modulo son desarrolladas, estas se diseñan pensando en determinar si el estudiante ha completado y entendido satisfactoriamente todo el modulo. De ser así, el estudiante puede moverse en el siguiente modulo de la secuencia previamente definida.

Dentro del modulo la secuencia de temas no es definida por el instructor, por el contrario, los temas y experiencias de aprendizaje correspondientes al modulo son vistas por los estudiantes en la secuencia que ellos prefieran; en otras

palabras, el control lo tiene el estudiante y los temas pueden ser estudiados de forma aleatoria, de manera no secuencial o en orden de hipercontenidos. La evaluación es al final del curso y se presenta como algo de mayor importancia, ver Figura 29.

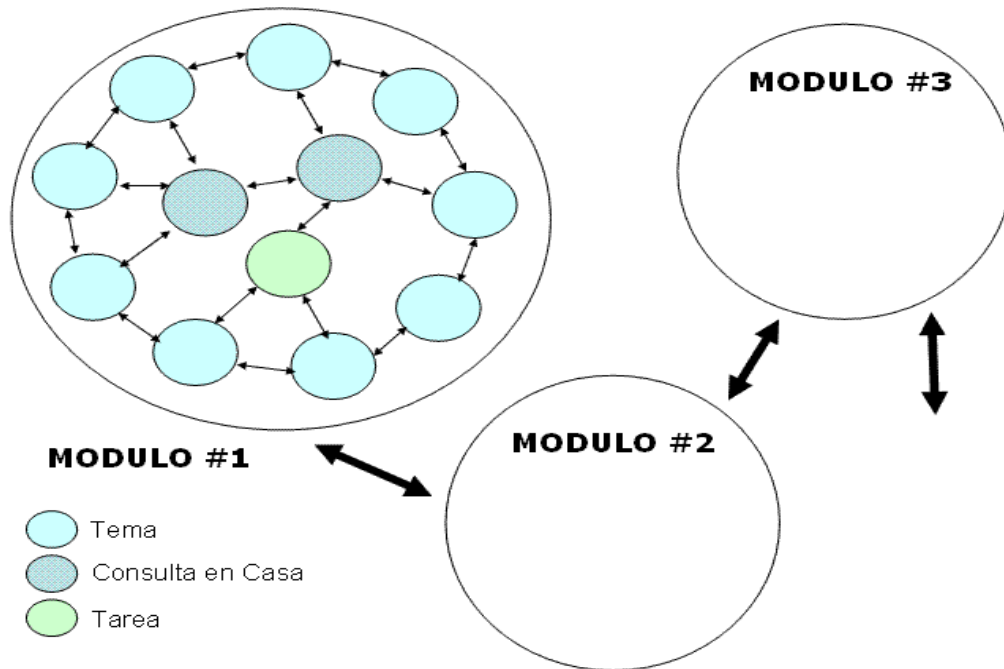


Figura 29. Diseño Instruccional con Hipercontenidos

ANEXO 9. Diseño directo del estudiante

En este modelo, el diseñador instruccional identifica los módulos o capítulos y los temas, incluyendo las experiencias de aprendizaje, pero no coloca ningún tipo de orden o secuencia dentro del capítulo o entre los módulos. El estudiante decide en que orden va a estudiar los temas e incluso que temas va a estudiar. El estudiante construye su propia estrategia instruccional y su propio diseño instruccional, los estudiantes se mueven a través de los módulos en el orden que ellos escojan. Cualquier requisito que se tenga se pasa al estudiante que hace las veces de diseñador instruccional.

Este modelo requiere de un considerable talento y esfuerzo de parte del estudiante para lograr el aprendizaje. La dirección da a los estudiantes los objetivos por modulo y las actividades de evaluación. Algunos constructivistas aconsejan que los procedimientos del “diseño directo del estudiante” deben dejar incluso que el estudiante realice sus propias actividades de valoración y evaluación.

En general, los modelos de diseño instruccional están evolucionando. Estos cuatro acercamientos son resultado de los esfuerzos y desarrollos de las experiencias realizadas en el pasado, es decir, con una programación a prueba y error, ver Figura 30.

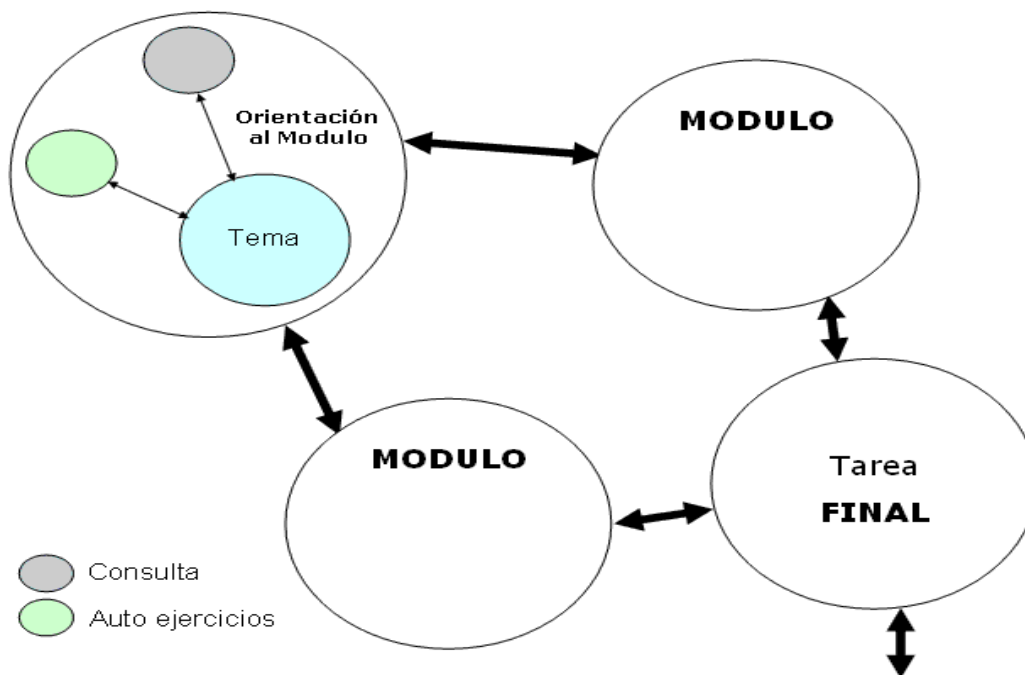


Figura 30. Diseño Instruccional Directo del Estudiante

ANEXO 10. Modelo de Agregación de Contenidos - CAM

El modelo de agregación de contenidos (CAM) proporciona una manera consistente para describir la estructura de la forma como se estructuran los contenidos dentro del estándar, es decir, es la metainformación sobre los distintos componentes y su estructura y una serie de reglas que determinan los recorridos permitidos sobre los contenidos.

Mantiene una posición neutral en lo que se refiere a un modelo pedagógico que sirva como base al estándar. El CAM da soporte al proceso de creación y agregación de recursos de aprendizaje para formar componentes más complejos que puede asociarse a actividades de aprendizaje para su realización por parte de los estudiantes.

Las reglas que establecen este comportamiento se dividen en cuatro partes:

- El Modelo de Contenidos proporciona una terminología común para ser usada en todo el CAM.
- El Modelo de Empaquetado ofrece la descripción y los requerimientos para componer objetos de aprendizaje y formar unidades más complejas.

- Metadatos para la descripción de los propios componentes de SCORM
- Normas de Secuenciación y Presentación que luego serán analizadas más a fondo en el libro de navegación y secuenciación.

❖ **Modelo de Contenidos**

El Modelo de Contenidos de SCORM describe los componentes que define esta inciativa para construir recursos de aprendizaje, y cómo estos recursos pueden combinarse en unidades de más alto nivel para formar unidades de instrucción que cumplan con las características instruccionales de la educación presencial y los modelos pedagógicos que alrededor de ella se han definido.

La unidad más básica definida por el estándar es el *Recurso (Asset)*. Estos no son mas que los archivos básicos que representan un información dentro del curso, es decir, archivos de texto, imágenes, animaciones, sonidos, objetos de evaluación, etc., los cuales tienen en común una características importante y es que todos tienen la capacidad de poderse mostrar en un navegador.

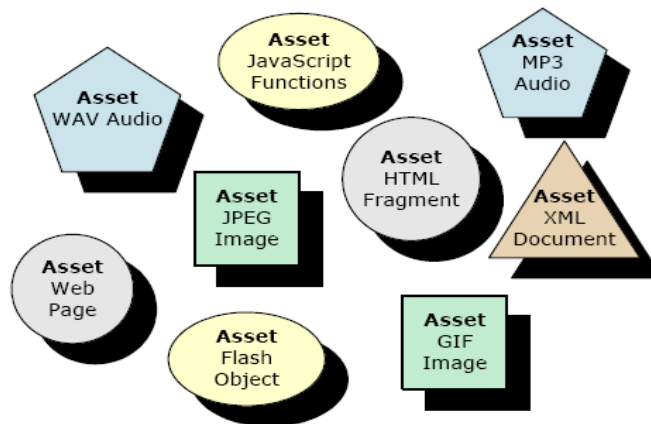


Figura 31. Ejemplos de Recursos (Asset's)

A partir de Asset's se generan unidades un poco mas complejas y que reúnen nuevas características que son llamadas SCO's (Objeto de contenido intercambiable - Shareable Content Object). Estos reúnen uno o mas recursos en una sola unidad que representa una entidad ejecutable capaz de comunicar información de si misma a una plataforma tecnológica que analiza dicha información. Para realizar la comunicación (SCO-Plataforma) el estándar define una interfaz de aplicación programada – API, desarrollada por la IEEE llamada IEEE ECMAScript API [44]. Esta describe todo el proceso de comunicación entre los SCO y la respectiva plataforma con la limitante de la dependencia tecnológica que existe. En este proyecto, se unificaron los recursos (asset's) y los SCO's en un solo termino que llamaremos *recurso*; esto gracias a que la característica

principal que los diferencia depende directamente de la capacidad que tienen los SCO's de comunicarse con la plataforma tecnológica y dado que en este proyecto dicha plataforma difiere bastante de la propuesta en el estándar esa comunicación no se realizará y por tanto no existen razones para diferenciar los dos términos.

Un nivel más arriba encontramos las *Organizaciones de Contenidos* que no son más que mapas que representan el uso de los distintos contenidos (Recursos o SCO) a través de un conjunto de unidades de instrucción denominadas *Actividades*. Estas actividades pueden estar a su vez formadas por más actividades, sin que haya límite en el nivel de anidamiento ni ninguna relación preestablecida con otras taxonomías (cursos, capítulos, lecciones, etc.). En este proyecto retomamos este concepto totalmente, ya que satisface las necesidades operacionales de nuestros objetivos. Tanto las Organizaciones de Contenidos como las Actividades tienen asociadas Metadatos que permiten reutilizarlas y localizarlas en repositorios.

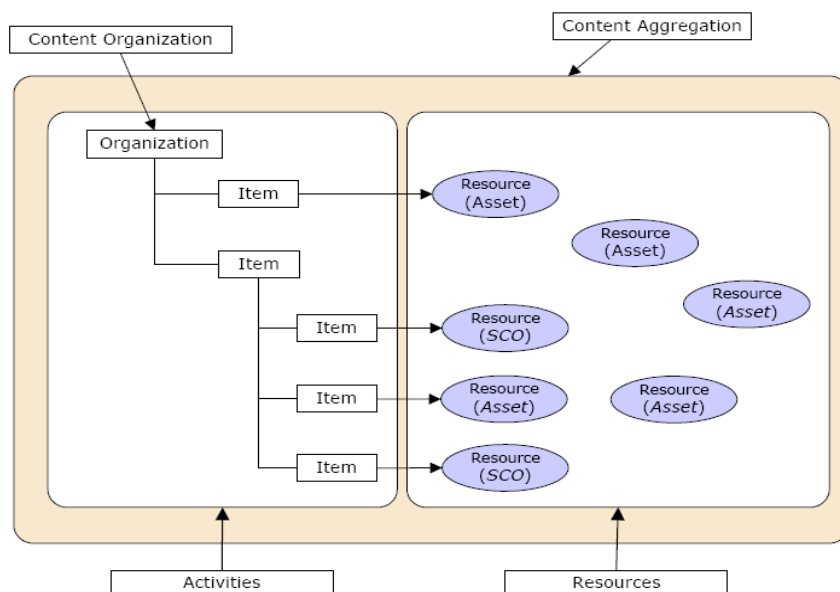


Figura 32. Organización de Contenidos

❖ *Modelo de empaquetado*

Una vez que el contenido de aprendizaje está construido y organizado para cumplir con las características instruccionales y pedagógicas del diseñador del curso, es necesario ponerlo a disposición de los alumnos, herramientas de autor, repositorios o plataformas de contenidos. Para ello, SCORM utiliza de forma estricta la especificación IMS Content Packaging Specification [46]. De esta manera se dispone de una forma estandarizada para intercambiar contenidos entre distintas plataformas y una descripción de la estructura y del comportamiento

de una colección de contenidos de aprendizaje, dando soporte a uno de los objetivos mas importantes en el desarrollo de este estándar, el cual es la capacidad de compartir las unidades instruccionales en diferentes cursos. Un paquete de contenidos se compone de dos partes fundamentales, un *manifiesto* en formato XML que describe la estructura de los contenidos y los recursos asociados a cada uno; y los recursos físicos como tal, es decir, los archivos con los recursos o las URL donde se estos se encuentran. Estos se agrupan en un solo archivo comprimido en formato PKZip que para el estándar SCORM recibe el nombre de PIF (Archivo de Paquete intercambiable – Package Interchange File). Este es el archivo que puede ser intercambiable entre varios cursos.

❖ **Manifiesto**

Este se compone de varias partes: Ver Figura 33

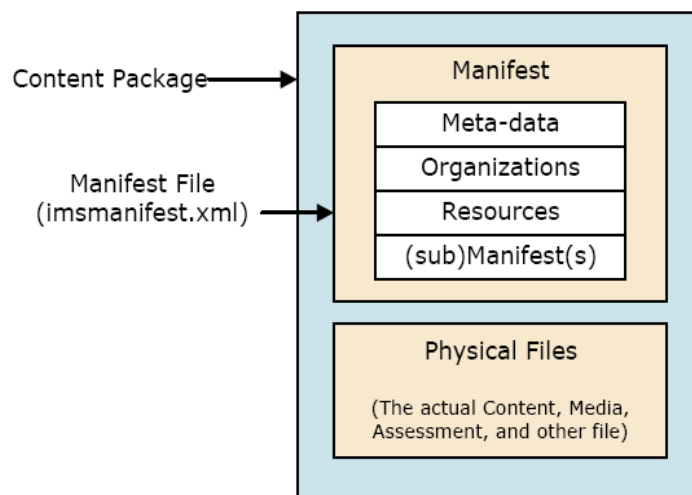


Figura 33. Ejemplo de un PIF

Metadatos: Describe el paquete como un todo. Acá encontramos información general como el esquema de los datos, el lenguaje de paquete, entre otros.

Organizaciones: Acá se encuentra la información acerca del paquete en el desarrollo del curso, es decir, información como la estructura de contenidos, los recursos asociados a cada actividad instruccional y la información acerca de la secuenciación y la navegación.

Recursos: Describe los recursos externos (representados por URL's) y locales que se encuentran comprimidos dentro del mismo paquete.

Submanifiestos: Se utilizan cuando los recursos son complejos y tienen su propia jerarquía interna.

❖ **Metadatos**

Los metadatos corresponden a la información acerca de la información que hay dentro del estándar SCORM. Es la herramienta que se utiliza para dar semántica a todos los componentes del curso. Cualquier clase de elemento dentro del paquete debe tener un metadato que informe de su contenido usando para ello etiquetas y atributos que agreguen semántica a la información contenida dentro del paquete y que se describe completamente en el estándar *IEEE 1484.12 Learning Object Meta-data* (LOM) [47].

❖ **Normas de secuenciación y presentación**

Describe como especificar determinadas estrategias de aprendizaje para la secuenciación de los contenidos de un paquete y para la presentación de los mismos en un navegador. En esta información se incluyen el manifiesto del PIF correspondiente, la definición de las actividades o su asociación directa al manifiesto como una colección de reglas y estrategias de secuenciación que pueden emplearse en varios manifiestos.

En el CAM simplemente se describe el vocabulario empleado para establecer reglas de secuenciación y guiar el desarrollo de la actividad instruccional.

ANEXO 11. Secuenciación y Navegación - SN

En este libro del estándar se recogen las características de los cursos de cara a la implementación práctica del curso en la plataforma tecnológica. Es decir, se define el orden en que se presentan los recursos, que recursos se presentan y la características de navegación que tendrá el estudiante en el curso.

Está formada por cinco partes:

- ✓ Terminología y conceptos básicos sobre secuenciación.
- ✓ Modelo de definición de secuenciación.
- ✓ Modelo de comportamiento de secuenciación.
- ✓ Requerimientos y control de la navegación.
- ✓ Modelo de datos de navegación.

Este componente está basado en la especificación de Secuencia Simple de IMS (IMS SS) [32], que define un método para representar el comportamiento deseado de una experiencia de aprendizaje. El término *simple* se refiere a que sólo se permiten un pequeño número de comportamientos.

Algunas de las limitaciones del estándar son [33]:

- ✓ Secuenciación dirigida por técnicas de inteligencia artificial.
- ✓ Secuenciación basada en la planificación.
- ✓ Secuenciación que requiera información de sistemas externos.
- ✓ Aprendizaje colaborativo.
- ✓ Sincronización entre actividades realizadas en paralelo.

Las reglas de secuenciación se aplican sobre las actividades definidas en una Organización de Contenido, forman parte del manifiesto de un PIF y pueden establecerse de forma individual sobre una actividad compleja (que esté formada a su vez por otras actividades) o ser unas reglas genéricas que pueden utilizarse en distintos manifiestos y paquetes de contenido.

Para la secuenciación de contenidos SCORM tiene en cuenta tres factores importantes: 1. del Árbol de Actividades, 2. de la estrategia de secuenciación definida; 3. del comportamiento definido ante cualquier tipo de actividad del estudiante que inicie uno de los eventos.

❖ **Conceptos para la Secuenciación**

La estructura que utiliza SCORM como base de la secuenciación de los contenidos es el Árbol de Actividades o árbol instruccional. La construcción de un Árbol de Actividades parte de una *Organización de Contenido* y está se conforma de una jerarquía de actividades que representan unidades de aprendizaje relevantes y que pueden anidarse sin límite para adecuarse a cualquier nivel de una taxonomía de niveles de aprendizaje (curso, módulo, lección, etc.). La relación existente entre una Organización de Contenido y un Árbol de Actividades es la que muestra la Figura 34.

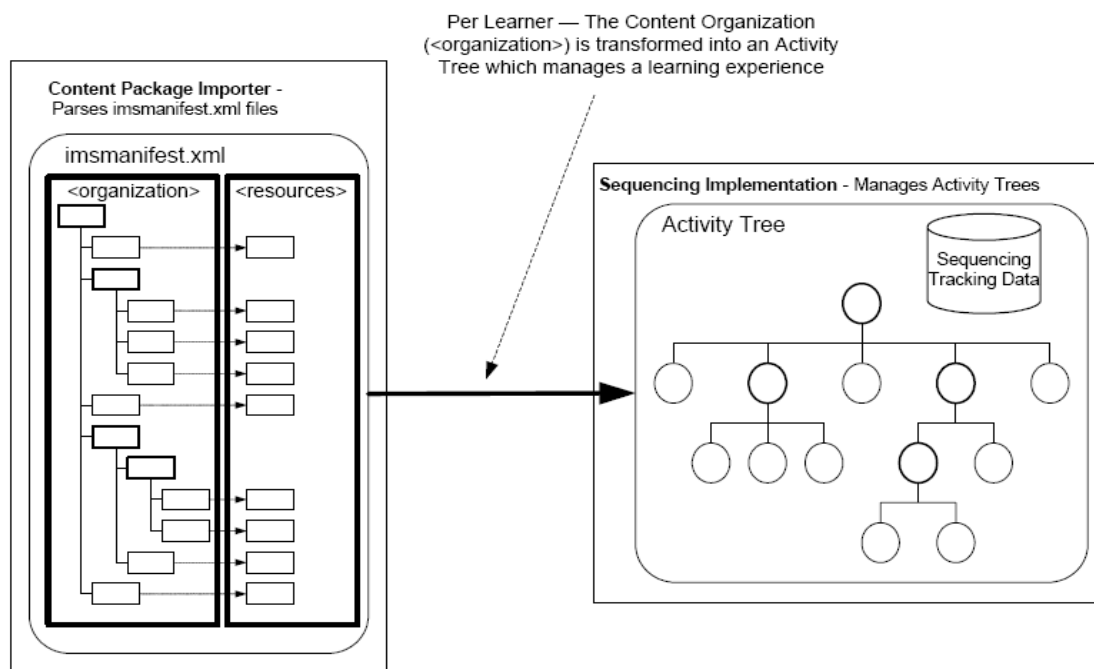


Figura 34. Estructura de contenidos Vs. Árbol de actividades (tomado de SCORM Sequencing and Navigation Version 1.3.1)

1. Un árbol representa la estructura de los contenidos como resultado del proceso de diseño, autoría y agregación de los contenidos. Para que cualquier plataforma pueda utilizarla, el árbol se convierte en una Organización de Contenido y se comprime en un paquete de contenido - PIF.
2. Una plataforma tecnológica traduce la Organización de Contenido en un Árbol de Actividades que representa la jerarquía de los contenidos, incluyendo información de seguimiento de cada actividad para cada estudiante.
3. Cuando el alumno escoge interactuar con el contenido representado en el Árbol de Actividades, la plataforma tecnológica evalúa la información de secuenciación y de su perfil personal para determinar qué actividad debe suministrarle o proporcionarle de un conjunto de actividades entre las que puede elegir.

❖ Modelo de definición de la secuenciación

Es un modelo derivado de IMS SS, en el que se definen los elementos necesarios para describir los comportamientos de secuenciación permitidos dentro del ambiente de las aulas virtuales. Estos elementos se aplican a las actividades dentro de los Árboles de Actividades o Instruccionales.

Para definir las posibles secuencias de actividades, SCORM establece una serie de modos de control de la secuencia. Se trata de un conjunto de elementos que describen cómo las peticiones de navegación afectan a un *cluster* y cómo las actividades correspondientes se toman en consideración. Este conjunto forma parte importante de los aportes del estándar a este proyecto, ya que favorece la aplicación de distintos modelos instruccionales en el desarrollo de estrategias de aprendizaje.

Los elementos de control de la secuenciación son:

Sequencing Control Choice: indica que el alumno es libre de elegir cualquier actividad dentro de un *cluster* en cualquier orden y sin ninguna restricción. La plataforma tendrá que proporcionar algún mecanismo (menú, lista desplegable, mapa) para seleccionar la actividad a la que se desea saltar. Contrasta con diseños instruccionales como el Constructivismo.

Sequencing Control Choice Exit: impide el salto entre ramas del Árbol de Actividades y fuerza a que los movimientos (especialmente hacia atrás) se realicen con las opciones de navegación de continuar y retroceder.

Sequencing Control Flow: indica que se permite la navegación a través de las actividades hija dentro de un *cluster*, es decir, el paso de una actividad a otra del mismo nivel. El orden entre las actividades debe ser declarado de forma explícita.

Sequencing Control Forward Only: como el anterior, con la restricción que solo se puede ir hacia adelante. Es un poco la filosofía del diseño instruccional Lineal.

❖ Modelo de Navegación

Las experiencias de aprendizaje se ofrecen al estudiante como una serie de actividades instruccionales a desarrollar. Los procesos de secuenciación que se implementan en la plataforma tecnológica son componentes estáticos, que actúan como respuesta a un evento de navegación.

La navegación es el modo que tienen los usuarios finales y la plataforma de formación de identificar unas peticiones de navegación para realizar una experiencia de aprendizaje. Los dispositivos que se emplean para que el usuario introduzca las órdenes de navegación pueden pertenecer a la plataforma, a los objetos de contenido o a ambos. No nos detendremos a explicar más a fondo este ítem dada la dependencia tecnológica que puede notarse del modelo además que hace parte del alcance de otro de los proyectos de la fase II de Unicauca Virtual.

ANEXO 12. Casos de Uso para la Gestión de Ítems – Formato Alto Nivel

Caso de Uso Insertar Ítems – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Insertar Ítems
Actores:	Directivo
Tipo:	Primario
Descripción	
Después de que el actor escoge la opción Definir Ítems, se muestra una ventana con un cuadro de texto, donde el actor digitará el nombre del ítem que desea adicionar a la base de datos. Después de guardada se le muestra un mensaje éxito o error según sea el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Insertar Ítems – Expandido]

Tabla 23. Caso de Uso Insertar Ítems – Alto Nivel

Caso de Uso Consultar Ítems – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Consultar Ítems
Actores:	Directivo
Tipo:	Primario
Descripción	
Después de que el actor escoge la opción Consultar Ítems, se muestra una ventana con un cuadro de texto, donde el actor digitará el nombre del ítem que desea consultar de la base de datos. Después de muestra la información del ítem si existe, sino le muestra un mensaje.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Consultar Ítem - Expandido]

Tabla 24. Caso de Uso Consultar Ítems – Alto Nivel

Caso de Uso Modificar Ítems – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Modificar Ítems
Actores:	Directivo
Tipo:	Primario
Descripción	
El usuario escoge el ítem que desea modificar. Luego se escoge la opción Modificar Ítem y se activan los campos que pueden ser modificados por el usuario, el actor digitará los nuevos datos del ítem a modificar y se guardan los cambios en la base de datos, mostrando un mensaje de éxito o error según el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Modificar Ítems – Expandido]

Tabla 25. Caso de Uso Modificar Ítems – Alto Nivel

Caso de Uso Eliminar Ítems – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Eliminar Ítems
Actores:	Directivo
Tipo:	Primario
Descripción	
El usuario escoge el ítem que desea eliminar. Luego se escoge la opción Eliminar Ítem y se muestra un cuadro de confirmación, si el usuario confirma el ítem es eliminado de la base de datos, mostrando un mensaje de éxito o error según sea el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Eliminar Ítem – Expandido]

Tabla 26. Caso de Uso Eliminar Ítem – Alto Nivel

ANEXO 13. Casos de Uso para la Gestión de Contenidos Programáticos – Formato Alto Nivel

Caso de Uso Definir Contenido Programático – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Definir Contenido Programático
Actores:	Directivo
Tipo:	Primario
Descripción	
Después de que el actor escoge la opción Definir Contenido Programático, se muestra una ventana con todos las asignaturas y los ítems asociados al contenido, a continuación el actor llenara los ítems obligatorios y los opcionales si lo desea, asociando dicha plantilla a una de las asignaturas consultadas, luego esta información será guardada en la base de datos y finalmente se mostrara un mensaje de éxito o error según sea el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Definir Contenido – Expandido]

Tabla 27. Caso de Uso Definir Contenido – Alto Nivel

Caso de Uso Consultar Contenido Programático – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Consultar Contenido Programático
Actores:	Directivo
Tipo:	Primario
Descripción	
El actor escoge la asignatura que desee y luego la opción Consultar Contenido Programático. Inmediatamente se muestran todos los contenidos programáticos asociados a dicha asignatura con la información consignada en cada uno de los ítems del contenido.	
TRAZABILIDAD	

Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Consultar Contenido – Expandido]

Tabla 28.Caso de Uso Consultar Contenido – Alto Nivel

Caso de Uso Activar Contenido Programático – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Activar Contenido Programático
Actores:	Directivo
Tipo:	Primario
Descripción	
Después de que el actor escoge la opción Activar Contenido Programático, se muestra una ventana con todas las asignaturas y los contenidos con sus ítems asociados a cada asignatura, al activarlo, inmediatamente se modificara el campo fecha de activación de dicho contenido finalmente se mostrara un mensaje de éxito o error según sea el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Activar Contenido – Expandido]

Tabla 29.Caso de Uso Activar Contenido – Alto Nivel

Caso de Uso Modificar Contenido Programático – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Modificar Contenido Programático
Actores:	Directivo
Tipo:	Primario
Descripción	
Después de que el actor escoge la opción Modificar Contenido Programático, se muestra una ventana con todas las asignaturas y los ítems asociados al contenido, a continuación el actor modificará los ítems que desee para luego registrarlos en la base de datos y finalmente se mostrara un mensaje de éxito o error según sea el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Modificar Contenido – Expandido]

Tabla 30.Caso de Uso Modificar Contenido – Alto Nivel

ANEXO 14. Casos de Uso para la Gestión de Agregaciones – Formato Alto Nivel

Caso de Uso Insertar Agregación – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Insertar Agregación
Actores:	Directivo
Tipo:	Primario
Descripción	

Después de que el actor escoge la opción Insertar Agregación, se muestra una ventana con un cuadro de texto, donde el actor digitará el nombre de la agregación que desea adicionar a la base de datos. Después de guardada se le muestra un mensaje éxito o error según sea el caso.

TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de uso Insertar Agregación - Expandido]

Tabla 31. Caso de Uso Insertar Agregación – Alto Nivel

Caso de Uso Consultar Agregación – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Consultar Agregación
Actores:	Directivo
Tipo:	Primario
Descripción	
Después de que el actor escoge la opción Consultar Agregación del menú, se muestra una ventana con un cuadro de texto, donde el actor digitará el nombre de la agregación que desea consultar. Después se muestran los datos de la agregación.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Consultar Agregación – Expandido]

Tabla 32. Caso de Uso Consultar Agregación – Alto Nivel

Caso de Uso Modificar Agregación – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Modificar Agregación
Actores:	Directivo
Tipo:	Primario
Descripción	
El actor escoge la agregación a modificar luego el actor escoge la opción Modificar Agregación, se muestra una ventana con un cuadro de texto, donde el actor digitará el nuevo nombre de la agregación a modificar y serán guardados los cambios en la base de datos, mostrando un mensaje de éxito o error según sea el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Modificar Agregación – Expandido]

Tabla 33. Caso de Uso Modificar Agregación – Alto Nivel

Caso de Uso Eliminar Agregación – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Eliminar Agregación
Actores:	Directivo
Tipo:	Primario
Descripción	
El actor escoge la agregación a eliminar luego el actor escoge la opción Eliminar Agregación, se muestra una ventana pidiendo confirmación, si el actor desea continuar se elimina la agregación de la base de datos, mostrando un mensaje de éxito o error según sea el caso.	

TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Eliminar Agregación – Expandido]

Tabla 34. Caso de Uso Eliminar Agregación – Alto Nivel

ANEXO 15. Casos de Uso para la Gestión de Actividades

Caso de Uso Insertar Actividad – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Insertar Actividad
Actores:	Directivo
Tipo:	Primario
Descripción	
El actor escoge la opción Insertar Actividad, se muestra una ventana con un cuadro de texto, donde el actor digitará el nombre de la actividad que desea adicionar a la base de datos y escoge el tipo de actividad que será dicha actividad. Luego da Éxito o Error según sea el caso	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de uso Insertar Actividad - Expandido]

Tabla 35. Caso de Uso Insertar Actividad – Alto Nivel

Caso de Uso Consultar Actividad – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Consultar Actividad
Actores:	Directivo
Tipo:	Primario
Descripción	
Después de que el actor escoge la opción Consultar Actividad, se muestra una ventana con un cuadro de texto, donde el actor podrá colocar el criterio de consulta ya sea con el nombre de la actividad o con el tipo de actividad que desea consultar. Después se muestran los datos de la actividad.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Consultar Actividad – Expandido]

Tabla 36. Caso de Uso Consultar Actividad – Alto Nivel

Caso de Uso Modificar Actividad – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Modificar Actividad
Actores:	Directivo
Tipo:	Primario
Descripción	
El actor escoge la actividad a modificar luego el actor escoge la opción Modificar Actividad, se muestra una ventana con un cuadro de texto, donde el actor digitará el nuevo nombre de la agregación a modificar o podrá cambiarle el tipo de actividad y serán guardados los cambios en la base de datos, mostrando un mensaje de éxito o error según sea el caso.	

TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Modificar Actividad – Expandido]

Tabla 37.Caso de Uso Modificar Actividad – Alto Nivel

Caso de Uso Eliminar Actividad – Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Eliminar Actividad
Actores:	Directivo
Tipo:	Primario
Descripción	
El actor escoge la actividad a eliminar luego el actor escoge la opción Eliminar Actividad, se muestra una ventana pidiendo confirmación, si el actor desea continuar se elimina la actividad de la base de datos, mostrando un mensaje de éxito o error según sea el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso Eliminar Actividad – Expandido]

Tabla 38.Caso de Uso Eliminar Actividad – Alto Nivel

ANEXO 16. Caso de Uso Crear Estructura de Contenido

Caso de Uso Crear Estructura de Contenido - Alto Nivel

CASO DE USO FORMATO DE ALTO NIVEL	
Nombre del Caso de Uso:	Crear Estructura de Contenido
Actores:	Directivo
Tipo:	Primario
Descripción	
Después de que el actor escoge la opción Crear Estructura de Contenido, se muestra una ventana con todas las agregaciones , las actividades, las asignaturas y los contenidos asociados a cada asignatura, a continuación el actor podrá empezar a armar el árbol instruccional con las agregaciones y actividades como lo desee, asociándolo a una asignatura y a un contenido programático de esa asignatura, este árbol será guardado en la base de datos y luego se mostrara un mensaje de éxito o error según sea el caso.	
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
[Diagrama de Casos de Uso]	[Caso de Uso CrearEstructuraContenido – Expandido]

Tabla 39.Caso de Uso CrearEstructuraContenido - Alto Nivel

ANEXO 17. Casos de Uso para la Gestión de Ítems - Formato Expandido

Caso de Uso Insertar Ítems - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Insertar Ítems.
Actores:	Directivo.
Propósito:	Insertar un ítem que se desea tener en los contenidos programáticos a crear.
Resumen:	El actor digita el nombre del nuevo ítem que va a estar asociado a todos los contenidos programáticos que se van a crear en adelante y lo guarda en la base de datos.
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R5.1
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción insertar ítem	2. se habilitan los campos del formulario para que el usuario digite la información
3. El usuario digita el nombre del nuevo ítem y lo manda a guardar en la base de datos	4. Guarda en la base de datos el nuevo ítem y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo guardar el ítem en la base de datos, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Insertar Ítems – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Insertar Ítem] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Insertar Ítems]

Tabla 40. Caso de Uso Insertar Ítems – Expandido

Caso de Uso Consultar Ítems – Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Consultar Ítems
Actores:	Directivo.
Propósito:	Consultar un ítem de la base de datos.
Resumen:	El actor digita el nombre del ítem que va a consultar y se muestra toda su información.
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R5.1
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción consultar ítem	2. se carga un formulario para que el usuario digite el ítem a consultar

3. El usuario digita el nombre del ítem y lo manda a consultar a la base de datos	4. Se retornan todos los ítems que cumplan con el criterio de búsqueda.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. El ítem no existe no se puede mostrar información.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Diagrama de Casos de Uso] [Caso de Uso Consultar Ítems – Alto Nivel] 	<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Diagrama de Secuencia Consultar Ítem] [Modelo Conceptual] Artefactos del Diseño: <ul style="list-style-type: none"> [Caso de Uso Real Consultar Ítem]

Tabla 41. Caso de Uso Consultar Ítem - Expandido

Caso de Uso Modificar Ítems - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Modificar Ítems
Actores:	Directivo
Propósito:	Modificar un ítem existente en la base de datos.
Resumen:	el actor consulta el Ítem a modificar, luego introduce el nuevo nombre y guarda el nuevo Ítem en la base de datos, se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R5.2, R5.3
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de la lista de ítems, el ítem que desea modificar	2. se cargan los datos del ítem en el formulario y se habilitan los campos modificables del ítem para que el usuario digite la nueva información
3. El usuario puede cambiar el nombre del ítem o el chequeo (obligatorio o no en la plantilla de creación de contenidos) y envía a guardar las modificaciones.	4. Se guarda en la base de datos el nuevo ítem y retorna un mensaje de éxito
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. Se le informa al usuario que no se pudo guardar en la base de datos el nuevo Ítem y retorna un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Diagrama de Casos de Uso] [Caso de Uso Modificar Ítems – Alto Nivel] 	<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Diagrama de Secuencia Modificar Ítem] [Modelo Conceptual] Artefactos del Diseño: <ul style="list-style-type: none"> [Caso de Uso Real Modificar Ítem]

Tabla 42. Caso de Uso Modificar Ítems – Expandido

Caso de Uso Eliminar Ítems - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Eliminar Ítems
Actores:	Directivo
Propósito:	Eliminar un ítem existente en la base de datos.
Resumen:	el actor escoge el ítem a Eliminar y confirma que lo desea eliminar, el ítem se borra de la base de datos y se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R5.2, R5.4
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge el ítem a eliminar	2. Se carga en el formulario toda la información del ítem seleccionado
3. El usuario manada a eliminar el ítem escogido	4. Se muestra el cuadro de confirmación de la eliminación del ítem
5. El usuario confirma la eliminación	6. Se elimina el ítem escogido y se le muestra un mensaje de Éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. Se le informa al usuario que no se pudo Eliminar el ítem retornando un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Eliminar Ítem – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Eliminar Ítem] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Eliminar Ítem]

Tabla 43. Caso de Uso Eliminar Ítem – Expandido

ANEXO 18. Casos de Uso para la Gestión de Contenidos Programáticos – Formato Expandido

Caso de Uso Definir Contenido Programático - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Definir Contenido Programático
Actores:	Directivo
Propósito:	Crear un contenido programático asociándolo a una asignatura.
Resumen:	el actor escoge una asignatura de las consultadas y asocia a ella la información digitada en la plantilla de contenidos programáticos y guarda el nuevo contenido en la base de datos, se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R3.2, R4.1, R5.3

CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción definir contenido programático.	2. se carga un formulario con las asignaturas y los ítems consultados de la base de datos para que los llene.
3. El usuario escoge la asignatura a la cual va a asociar el contenido programático, llena los ítems obligatorios y los opcionales si lo desea y manda a guardar el contenido a la base de datos.	4. Guarda en la base de datos el nuevo contenido programático y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo guardar el contenido programático en la base de datos, se retorna un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • <u>Artefactos del Análisis:</u> <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Definir Contenido – Alto Nivel] 	<ul style="list-style-type: none"> • <u>Artefactos del Análisis:</u> <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Definir Contenido] ○ [Modelo Conceptual] • <u>Artefactos del Diseño:</u> <ul style="list-style-type: none"> ○ [Caso de Uso Real Definir Contenido]

Tabla 44. Caso de Uso Definir Contenido – Expandido

Caso de Uso Consultar Contenido Programático - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Consultar Contenido Programático
Actores:	Directivo
Propósito:	Consultar contenido programático asociándolo a una asignatura.
Resumen:	El actor escoge una asignatura de las existentes y se consultan los contenidos programáticos asociados a dicha asignatura.
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R3.2, R4.1, R5.3
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la asignatura que tiene el contenido programático a consultar	2. Se muestran todos los contenidos programáticos asociados a la asignatura escogida.
3. El usuario escoge el contenido programático que desea consultar.	4. Se muestra la información de los ítems de contenido correspondiente al contenido seleccionado por el usuario.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo mostrar la información del contenido programático asociado a la asignatura se retorna un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores

<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Consultar Contenido – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Consultar Contenido] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Consultar Contenido]
---	--

Tabla 45. Caso de Uso Consultar Contenido – Expandido

Caso de Uso Activar Contenido Programático - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Activar Contenido Programático
Actores:	Directivo
Propósito:	Activar un contenido programático que ya se encuentra asociando a una asignatura.
Resumen:	el actor escoge una asignatura de las consultadas, escoge uno de los contenidos asociados a dicha asignatura y activa el que desee, se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R3.2, R4.4
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de la asignatura que tiene el contenido programático que se desea activar	2. Se muestran todos los contenidos programáticos asociados a la asignatura escogida.
3. El usuario escoge el contenido asociado que desee activar y lo manda a activar.	4. Se guarda en la base de datos la fecha de activación inicial con la fecha actual o se le da al usuario la opción de escoger la fecha en que desea que sea activo el contenido y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	2. Se le informa al usuario que no hay contenidos programáticos asociados a la asignatura y se deshabilita la opción de habilitar contenido.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Activar Contenido – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Activar Contenido] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Activar Contenido]

Tabla 46. Caso de Uso Activar Contenido – Expandido

Caso de Uso Modificar Contenido Programático - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Modificar Contenido Programático
Actores:	Directivo

Propósito:	Modificar los ítems de un contenido programático existente.
Resumen:	Se muestra una ventana con todas las asignaturas y los ítems asociados al contenido, el actor escoge el contenido a modificar y modificará los ítems que desee, luego los registra en la base de datos y finalmente se mostrara un mensaje de éxito o error según sea el caso.
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R3.2, R4.4
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de la asignatura que tiene el contenido programático que se desea activar	2. Se muestran todos los contenidos programáticos asociados a la asignatura escogida.
3. El usuario escoge el contenido programático asociado a la asignatura escogida. que desea modificar	4. Se carga el estado de dicho contenido en y los datos existentes en cada uno de los ítems del contenido escogido.
5. El da la orden de modificación	6. Se habilitan los campos que son modificables.
3. El usuario modifica la información consignada en los ítems del contenido y manda a guardar los cambios a la base de datos.	4. Guarda en la base de datos los datos de los ítems modificados y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
2.	2. se carga un formulario con las asignaturas y en caso de no tener contenidos asociados se muestra un mensaje al usuario informando lo ocurrido
CURSO ALTERNO 2	
Acción del Actor	Respuesta del Sistema
8.	8. Se envía un mensaje de error al guardar los datos modificados
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Modificar Contenido – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Modificar Contenido] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Modificar Contenido]

Tabla 47. Caso de Uso Modificar Contenido – Expandido

ANEXO 19. Casos de Uso para la Gestión de Agregaciones – Formato Expandido

Caso de Uso Insertar Agregación - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Insertar Agregación
Actores:	Directivo

Propósito:	Ingresar una nueva agregación a la Base de Datos
Resumen:	el actor digita el nombre de la nueva agregación y esta es guardada en la base de datos, se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R2.1
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción insertar una nueva agregación	2. se habilitan los campos modificables de la agregación.
3. El usuario digita la nueva información y envía la agregación modificada a guardar en la base de datos	4. Guarda en la base de datos las modificaciones y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo guardar la nueva agregación, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Insertar Agregación – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Insertar Agregación] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de uso Real Insertar Agregación]

Tabla 48. Caso de uso Insertar Agregación - Expandido

Caso de Uso Consultar Agregación – Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Consultar Agregación
Actores:	Directivo
Propósito:	Consultar una agregación de la base de datos
Resumen:	El actor consulta la agregación que desea introduciendo el nombre de la misma.
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R2.2, R2.3
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción consultar agregación	2. se carga un formulario para que el usuario digite la información
3. El usuario digita el nombre de la agregación que desea consultar	4. Se consulta la agregación digitada y se le muestran los demás datos en el formulario.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. Se le informa al usuario que la agregación que se desea consultar no existe en la base de datos.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores

<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Consultar Agregación – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Consultar Agregación] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Consultar Agregación]
--	--

Tabla 49. Caso de Uso Consultar Agregación – Expandido

Caso de Uso Modificar Agregación - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Modificar Agregación
Actores:	Directivo
Propósito:	Modificar el nombre de una agregación de la base de datos
Resumen:	el actor consulta la agregación a modificar, luego introduce el nuevo nombre y guarda la nueva agregación en la base de datos, se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R2.2, R2.3
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la agregación a modificar	2. Se carga la información de la agregación para que el usuario compruebe que es la agregación que desea
3. El usuario pide hacer actualización sobre los datos mostrados de la agregación	4. Se habilitan los campos modificables de la agregación para que el usuario digite la nueva información
5. El usuario digita la nueva información de la agregación y envía las modificaciones a la base de datos.	6. Guarda en la base de datos las modificaciones de la agregación y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudieron guardar las modificaciones, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Modificar Agregación – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Modificar Agregación] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Modificar Agregación]

Tabla 50. Caso de Uso Modificar Agregación – Expandido

Caso de Uso Eliminar Agregación – Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Eliminar Agregación
Actores:	Directivo

Propósito:	Eliminar una agregación de la base de datos	
Resumen:	el actor escoge la agregación a eliminar, luego confirma y la agregación es eliminada siempre y cuando no se use en alguna estructura de contenidos creada anteriormente	
Tipo:	Primario	
Referencias Cruzadas:	R1.1, R1.2, R1.3, R2.2, R2.4	
CURSO NORMAL DE LOS EVENTOS		
Acción del Actor		Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la agregación a eliminar		2. se muestran los datos de la agregación
3. El usuario escoge la opción de eliminar la agregación escogida		4. Se carga el cuadro de confirmación para la eliminación
5. El usuario confirma la eliminación de la agregación.		6. Se elimina la agregación de la base de datos y retorna un mensaje de éxito.
CURSO ALTERNO 1		
Acción del Actor		Respuesta del Sistema
4.		4. No se pudo eliminar la agregación, se retorna un mensaje de error
TRAZABILIDAD		
Artefactos Anteriores		Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Eliminar Agregación – Alto Nivel] 		<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Eliminar Agregación] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Eliminar Agregación]

Tabla 51. Caso de Uso Eliminar Agregación – Expandido

ANEXO 20. Casos de Uso para la Gestión de Actividades – Formato Expandido

Caso de Uso Insertar Actividad - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Insertar Actividad
Actores:	Directivo
Propósito:	Ingresar una nueva actividad a la Base de Datos
Resumen:	el actor digita el nombre de la nueva actividad con el tipo de actividad que es y se guarda en la base de datos, se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R6.1
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción insertar una nueva actividad	2. se habilitan los campos del formulario para que el usuario digite la información

3. El usuario digita el nombre de la nueva actividad y el tipo de actividad y la envía a guardar a la base de datos	4. Guarda en la base de datos la nueva actividad y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo guardar la nueva actividad, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Insertar Actividad – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Insertar Actividad] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de uso Real Insertar Actividad]

Tabla 52. Caso de uso Insertar Actividad - Expandido

Caso de Uso Consultar Actividad – Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Consultar Actividad
Actores:	Directivo
Propósito:	Consultar una actividad de la base de datos
Resumen:	El actor consulta la actividad que desea introduciendo el nombre de la misma o el tipo de la actividad
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R2.2, R6.2
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción consultar actividad	2. se carga un formulario para que el usuario digite la información
3. El usuario digita el nombre de la actividad o el tipo de actividad que desea consultar	4. Se consulta la actividad digitada y se le muestran los demás datos en el formulario.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. Se le informa al usuario que la actividad que se desea consultar no existe en la base de datos.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Consultar Actividad – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Consultar Actividad] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Consultar Actividad]

Tabla 53. Caso de Uso Consultar Actividad – Expandido

Caso de Uso Modificar Actividad - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Modificar Actividad
Actores:	Directivo
Propósito:	Modificar el nombre o el tipo de actividad de la base de datos
Resumen:	el actor consulta la actividad a modificar, luego introduce el nuevo nombre o cambia el tipo de actividad y guarda la actividad en la base de datos con las modificaciones, se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R2.2, R6.3
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la actividad a modificar y da la opción modificar	2. se habilitan los campos modificables de la actividad en el formulario para que el usuario digite la información
3. El usuario digita el nuevo nombre de la actividad a modificar o puede cambiar el tipo de actividad de la misma	4. Guarda en la base de datos las modificaciones de la actividad y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudieron guardar las modificaciones, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Modificar Actividad – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Modificar Actividad] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Modificar Actividad]

Tabla 54. Caso de Uso Modificar Actividad – Expandido

Caso de Uso Eliminar Actividad - Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Eliminar Actividad
Actores:	Directivo
Propósito:	Eliminar una actividad de la base de datos
Resumen:	el actor escoge la actividad a eliminar, luego confirma y la actividad es eliminada siempre y cuando no se use en alguna estructura de contenidos creada anteriormente
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R2.2, R6.4
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la actividad a eliminar	2. se muestran los datos de la actividad y un cuadro de dialogo de confirmación

3. El usuario confirma la eliminación de la actividad	4. Elimina la actividad de la base de datos y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo eliminar la actividad, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso Eliminar Actividad – Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Eliminar Actividad] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Eliminar Actividad]

Tabla 55. Caso de Uso Eliminar Actividad – Expandido

ANEXO 21. Caso de Uso Crear Estructura de Contenido – Formato Expandido

CASO DE USO FORMATO EXPANDIDO	
Nombre del Caso de Uso:	Crear Estructura de Contenido
Actores:	Directivo
Propósito:	Crear una estructura de contenidos con agregaciones y actividades para una asignatura y su contenido relacionado.
Resumen:	el actor crea la estructura de contenidos con las agregaciones y las actividades existentes, la vincula con una asignatura y un contenido asociado a esa asignatura, luego se guarda en la base de datos y se muestra un mensaje de error o de éxito según sea el caso
Tipo:	Primario
Referencias Cruzadas:	R1.1, R1.2, R1.3, R2.3, R3.1, R3.2, R3.3, R4.4
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge la opción crear estructura de contenidos	2. se carga un formulario con la todas las asignaturas consultadas de la base de datos
3. El usuario escoge de la lista de asignaturas consultadas, la asignatura a la que desea vincular la estructura de contenidos a crear	4. El sistema inmediatamente consulta el contenido programático que esta activo para la asignatura escogida
5. El usuario consulta las agregaciones o actividades que desee agregar a la estructura	6. Se muestra la información de las agregaciones y las actividades consultadas
7. El usuario empieza a armar la estructura en forma de árbol escogiendo de las agregaciones y las actividades consultadas las que desee y organizándolas de la forma como lo considere conveniente, finalmente manda a guardar la estructura creada.	8. Guarda en la base de datos la nueva estructura de contenido y retorna un mensaje de éxito.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema

2.	2. Se le informa al usuario que no hay asignaturas en la base de datos y deshabilita la opción de guardar la estructura.
CURSO ALTERNO 2	
Acción del Actor	Respuesta del Sistema
4.	4. La asignatura no tiene un contenido programático activo, se le informa al usuario.
CURSO ALTERNO 3	
Acción del Actor	Respuesta del Sistema
5.	6. Si no hay actividades se muestra la información solo de las agregaciones
7. El usuario empieza a armar la estructura de contenido solo con las agregaciones luego manda a guardar la estructura en la base de datos.	8. Guarda en la base de datos la nueva estructura de contenido y retorna un mensaje de éxito.
CURSO ALTERNO 4	
Acción del Actor	Respuesta del Sistema
6.	6. Si no hay agregaciones se carga el formulario con los demás datos, pero no se puede crear un árbol de estructura por que las actividades no pueden ser padre
CURSO ALTERNO 5	
Acción del Actor	Respuesta del Sistema
8.	8. No se pudo guardar la estructura de contenidos recientemente creada, se le envía al usuario un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Casos de Uso] ○ [Caso de Uso de Crear Estructura Contenido - Alto Nivel] 	<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Diagrama de Secuencia Crear Estructura] ○ [Modelo Conceptual] • Artefactos del Diseño: <ul style="list-style-type: none"> ○ [Caso de Uso Real Crear Estructura de Contenido]

Tabla 56. Caso de Uso Crear Estructura Contenido – Expandido

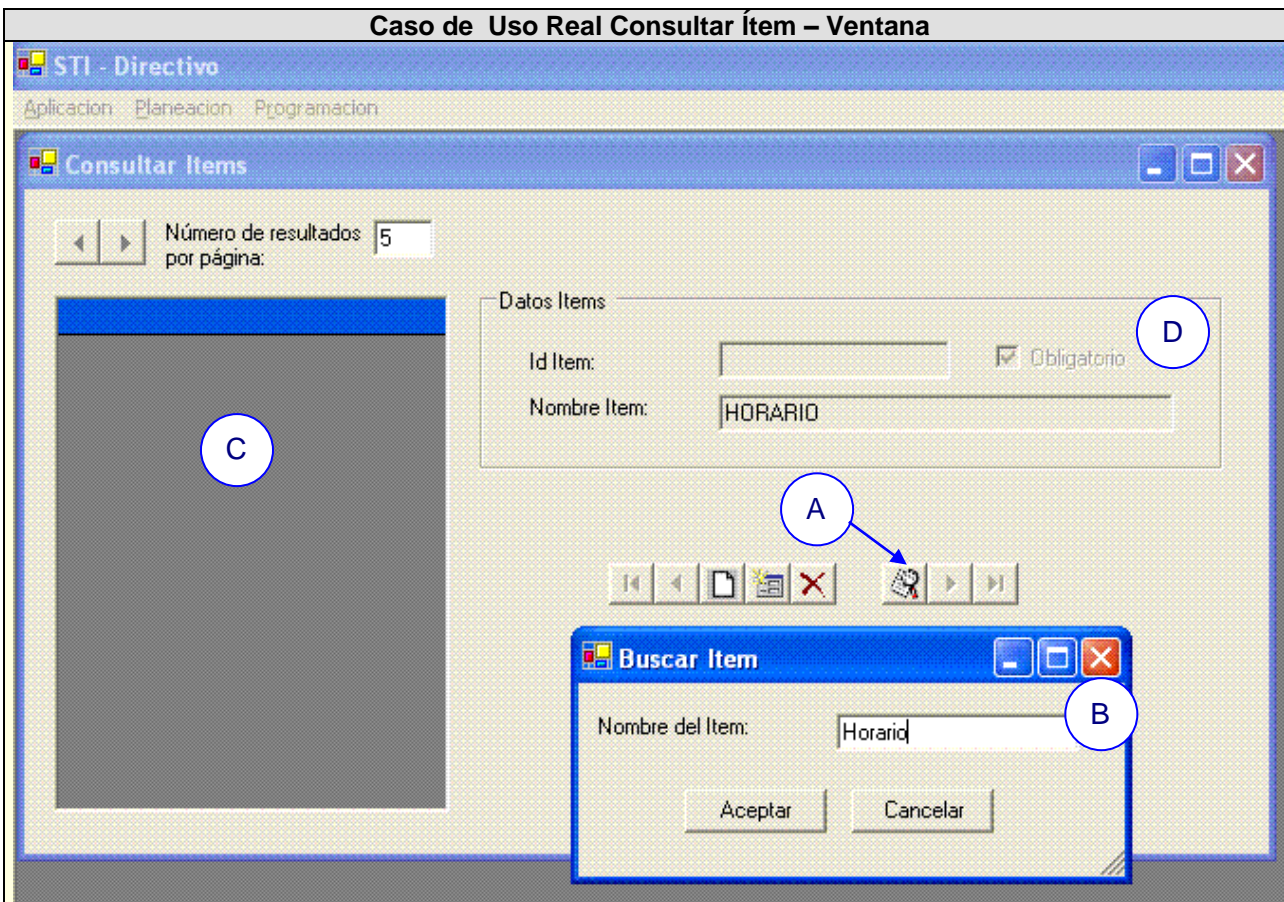
ANEXO 22. Casos de Uso Reales para la Gestión de Ítems Contenido

☑ Caso de Uso Real Insertar Ítems Contenido

CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario pulsa A para insertar el Ítem	2. Se habilitan los campos de B para que el usuario digite la información
3. El usuario digita el nombre del nuevo ítem y chequea el campo si desea que el ítem sea obligatorio o no y lo manda a guardar en la base de datos pulsando C	4. Guarda en la base de datos el nuevo ítem y retorna el mensaje de éxito como D y se deshabilita nuevamente C .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo guardar el ítem en la base de datos, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Caso de Uso Insertar Ítems – Expandido] 	

Tabla 57. Caso de Uso Real Insertar Ítems

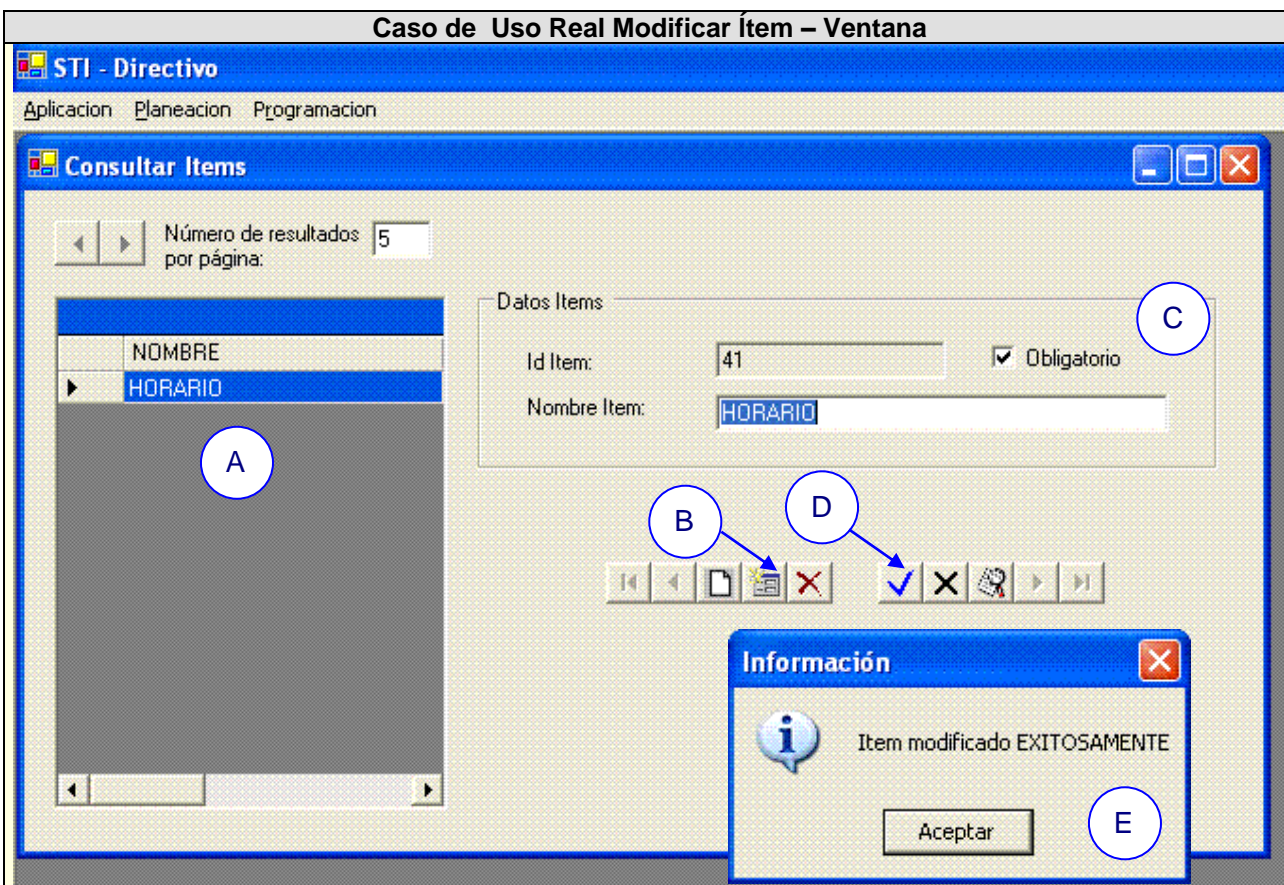
☑ **Caso de Uso Real Consultar Ítems Contenido**



CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario pulsa A que es la opción de consultar ítem	2. se carga el formulario B para que el usuario digite el ítem a consultar
3. El usuario digita el nombre del ítem en B y lo manda a consultar a la base de datos	4. Se retornan todos los ítems que cumplan con la condición de búsqueda en C y se carga la información del primer Registro de la lista en D .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. El ítem no existe no se puede mostrar información.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • <u>Artefactos del Análisis:</u> <ul style="list-style-type: none"> ○ [Caso de Uso Consultar Ítem - Expandido] 	

Tabla 58. Caso de Uso Real Consultar Ítem

☑ **Caso de Uso Real Modificar Ítems Contenido**



CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A el ítem que desea modificar, luego pulsa B que es la opción modificar ítem	2. se cargan los datos en C y se habilitan los campos modificables del ítem para que el usuario digite la nueva información
3. El usuario puede cambiar el nombre del ítem o el chequeo (obligatorio o no en la plantilla de creación de contenidos) y envía a guardar las modificaciones pulsando D .	4. Se guarda en la base de datos el nuevo ítem y retorna un mensaje de éxito como E y se deshabilita D .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. Se le informa al usuario que no se pudo guardar en la base de datos el nuevo ítem y retorna un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> ▪ <u>Artefactos del Análisis:</u> <ul style="list-style-type: none"> ○ [Caso de Uso Modificar Ítems – Expandido] 	

Tabla 59. Caso de Uso Real Modificar Ítem

☑ **Caso de Uso Real Eliminar Ítems Contenido**

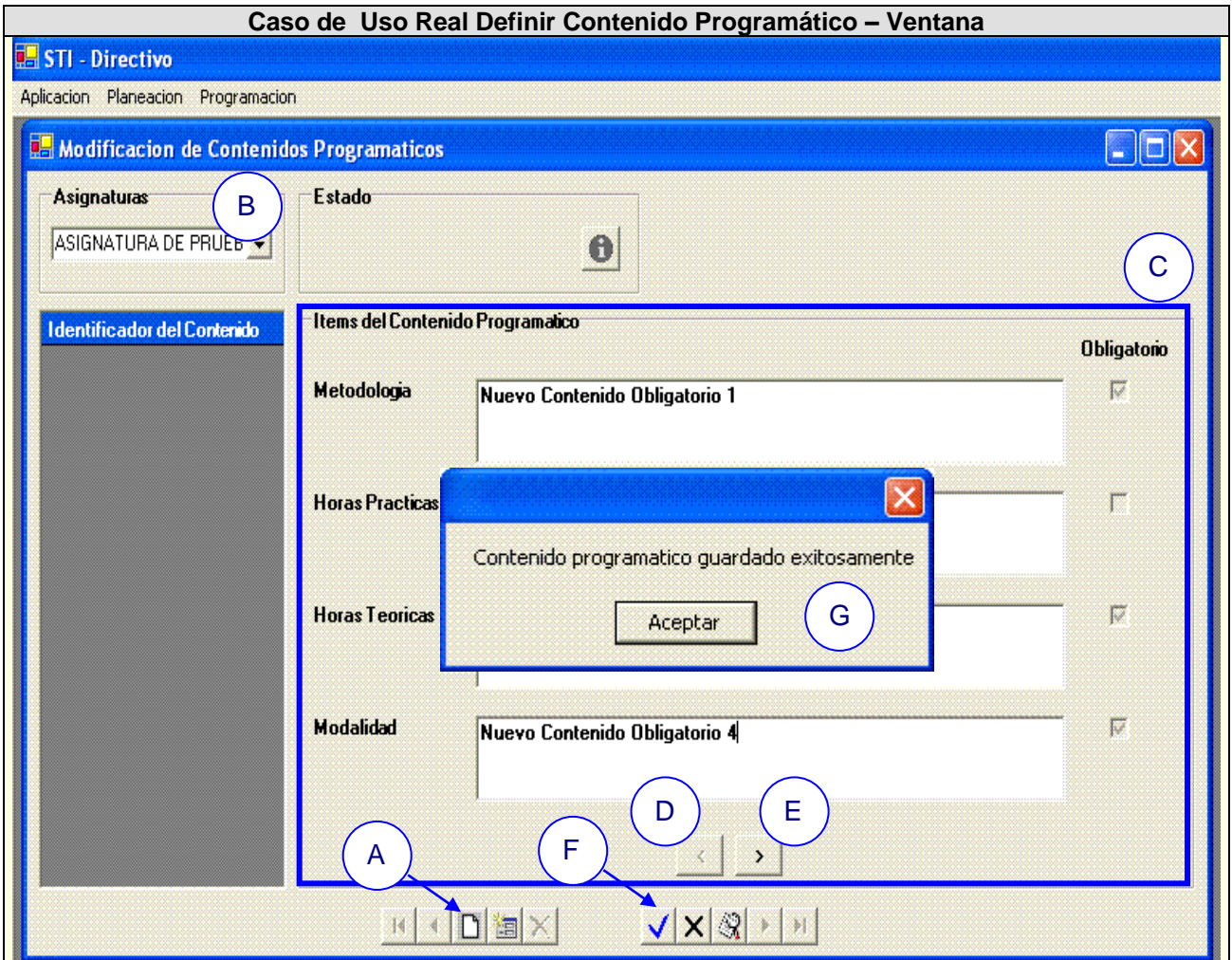
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A el ítem a eliminar	2. Se carga en B toda la información del ítem seleccionado
3. El usuario pulsa C que es la opción eliminar ítem	4. Se muestra el formulario D para que el usuario confirme si desea eliminar el ítem
5. El usuario confirma la eliminación pulsando Aceptar en el formulario D	6. Se elimina el ítem escogido y se le muestra un mensaje de Éxito como E .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. Se le informa al usuario que no se pudo Eliminar el ítem retornando un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Caso de Uso Eliminar Ítem – Expandido] 	

Tabla 60. Caso de Uso Real Eliminar Ítem

ANEXO 23. Casos de Uso Reales para la Gestión de Contenidos Programáticos

☑ Caso de Uso Real Definir Contenido Programático

CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario pulsa A que es la opción de definir contenido programático.	2. se cargan B las asignaturas existentes y en C los ítems consultados de la base de datos para que el usuario los llene.
3. El usuario escoge de B la asignatura a la cual va a asociar el contenido programático, llena los ítems obligatorios de C y los opcionales si lo desea y navega a través de la paginación con los botones D hacia atrás y E hacia delante, luego envía a guardar el contenido pulsando F .	4. Guarda en la base de datos el nuevo contenido programático y retorna un mensaje de éxito como G y luego se deshabilita nuevamente F .
CURSO ALTERNO 1	



Acción del Actor	Respuesta del Sistema
4.	4. No se pudo guardar el contenido programático en la base de datos, se retorna un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Caso de Uso Definir Contenido – Expandido] 	

Tabla 61. Caso de Uso Real Definir Contenido

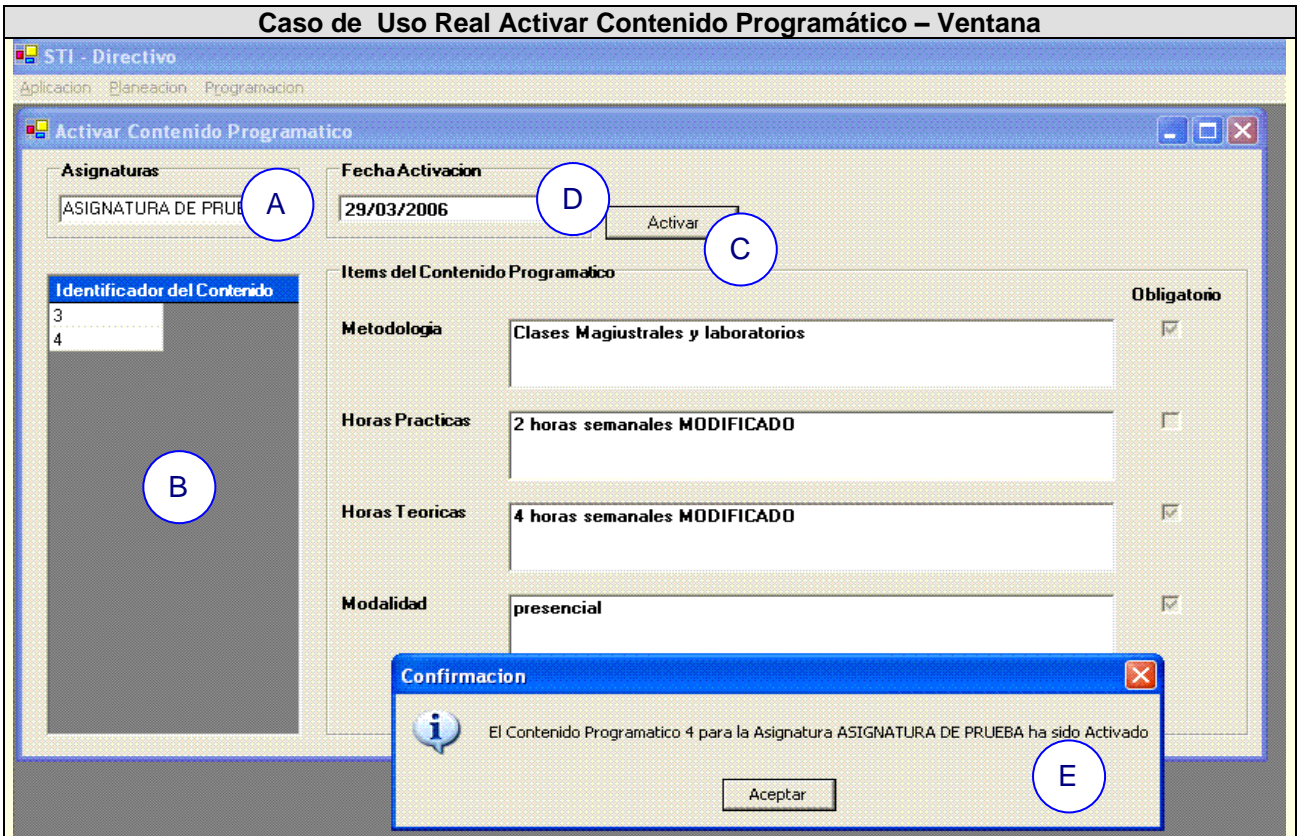
Caso de Uso Real Consultar Contenido Programático

Caso de Uso Real Consultar Contenido Programático – Ventana	
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A la asignatura que tiene el contenido programático a consultar	2. Se muestran en B todos los contenidos programáticos asociados a la asignatura escogida.

3. El usuario escoge de B el contenido programático que desea consultar. Que es equivalente a dar clic en C , debido a que el criterio de consulta es el identificador del contenido y el usuario no tiene por que saberlo.	4. Se muestra en D la información de los ítems de contenido correspondiente al contenido seleccionado por el usuario, se puede navegar por la paginación con E y F .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
	4. No se pudo mostrar la información del contenido programático asociado a la asignatura se retorna un mensaje de error.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Caso de Uso Consultar Contenido – Expandido] 	

Tabla 62. Caso de Uso Real Consultar Contenido

Caso de Uso Real Activar Contenido Programático

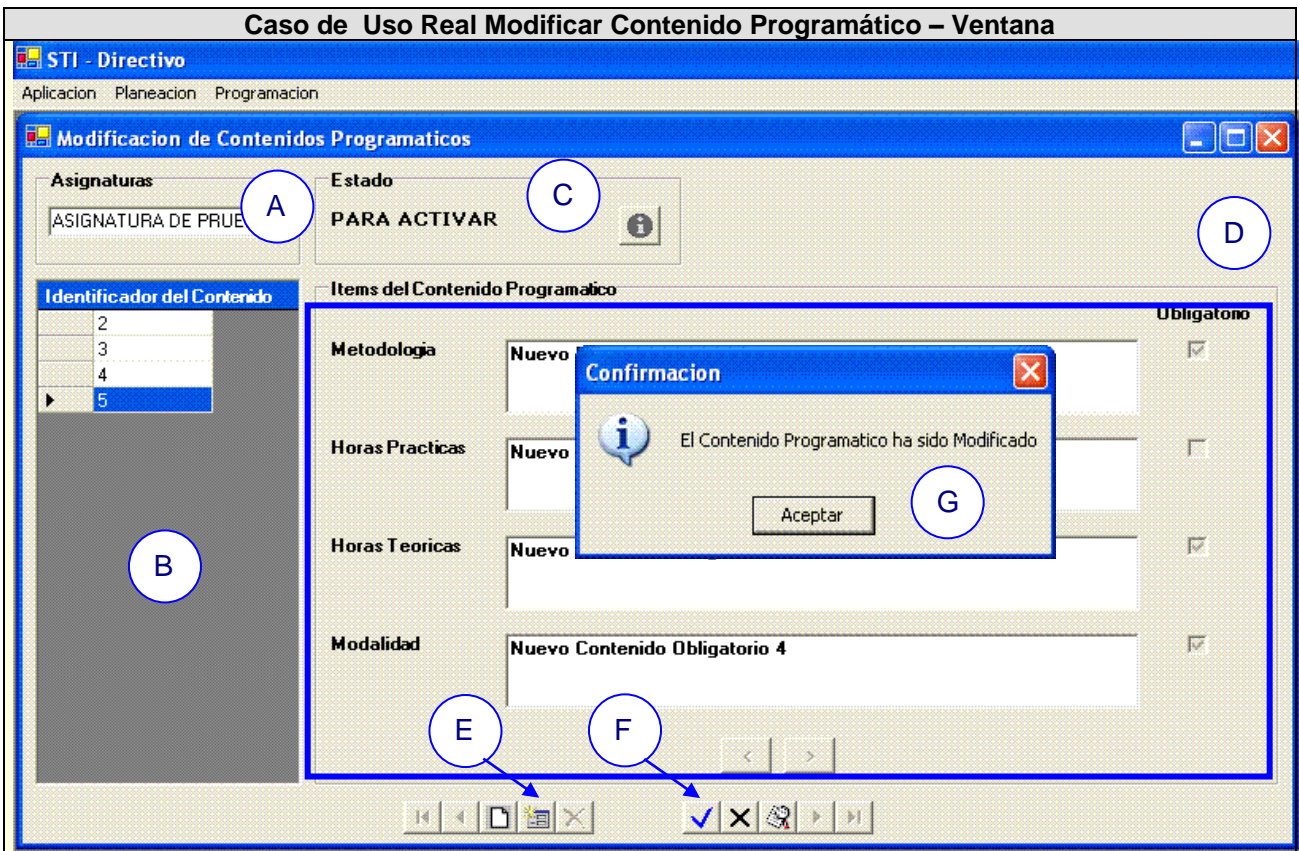


CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A la asignatura que tiene el contenido programático que se desea activar	2. Se muestran en B todos los contenidos programáticos asociados a la asignatura escogida.

3. El usuario escoge de B el contenido asociado que desee activar y pulsa C .	4. Se guarda en la base de datos la fecha de activación inicial con la fecha actual o se le da al usuario la opción de escoger la fecha en que desea que sea activo el contenido en el campo D y retorna un mensaje de éxito como E
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	2. Se le informa al usuario que no hay contenidos programáticos asociados a la asignatura y se deshabilita la opción de habilitar contenido.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Caso de Uso Activar Contenido - Expandido] 	

Tabla 63. Caso de Uso Real Activar Contenido

Caso de Uso Real Modificar Contenido Programático



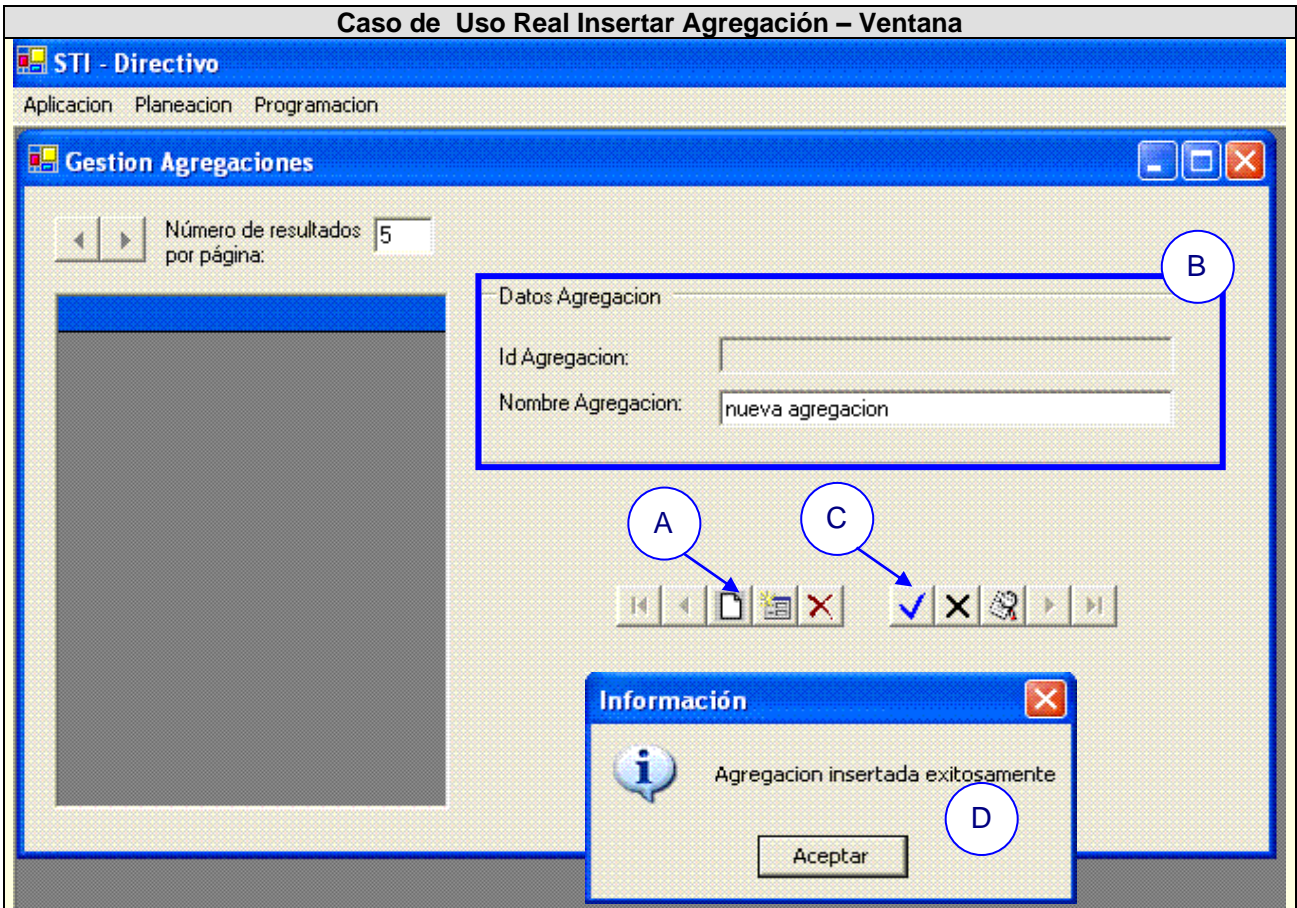
CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A la asignatura que tiene el contenido programático que se desea activar	2. Se muestran en B todos los contenidos programáticos asociados a la asignatura escogida.

3. El usuario escoge el contenido programático asociado a la asignatura escogida. que desea modificar de la lista de B	4. Se carga el estado de dicho contenido en C si es activo o no con la fecha correspondiente y se cargan en D los datos existentes en cada uno de los ítems del contenido escogido.
5. El usuario pulsa E para hacer las modificaciones que desee en los campos permitidos.	6. Se habilitan los campos de D que son modificables, en este caso se habilitan para edición todos los ítems del contenido.
7. El usuario modifica la información consignada en D de los ítems del contenido y manda a guardar los cambios a la base de datos dando clic en F	8. Guarda en la base de datos los datos de los ítems modificados y retorna un mensaje de éxito como G . y se deshabilita F nuevamente.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
2.	2. se carga un formulario con las asignaturas y en caso de no tener contenidos asociados se muestra un mensaje al usuario informando lo ocurrido
CURSO ALTERNO 2	
Acción del Actor	Respuesta del Sistema
8.	8. Se envía un mensaje de error al guardar los datos modificados
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • <u>Artefactos del Análisis:</u> <ul style="list-style-type: none"> ○ [Caso de Uso Modificar Contenido – Expandido] 	

Tabla 64. Caso de Uso Real Modificar Contenido

ANEXO 24. Casos de Uso Reales para la Gestión de Agregaciones

☑ Caso de Uso Real Insertar Agregación



CURSO NORMAL DE LOS EVENTOS

Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario da clic en A que es la opción insertar una nueva agregación	2. se habilitan los campos de B que pueden ser modificados por el usuario.
3. El usuario digita la nueva información en B y lo manda a guardar en la base de datos dando clic en C	4. Guarda en la base de datos la nueva agregación y retorna un mensaje de éxito como D . y se deshabilita C nuevamente

CURSO ALTERNO 1

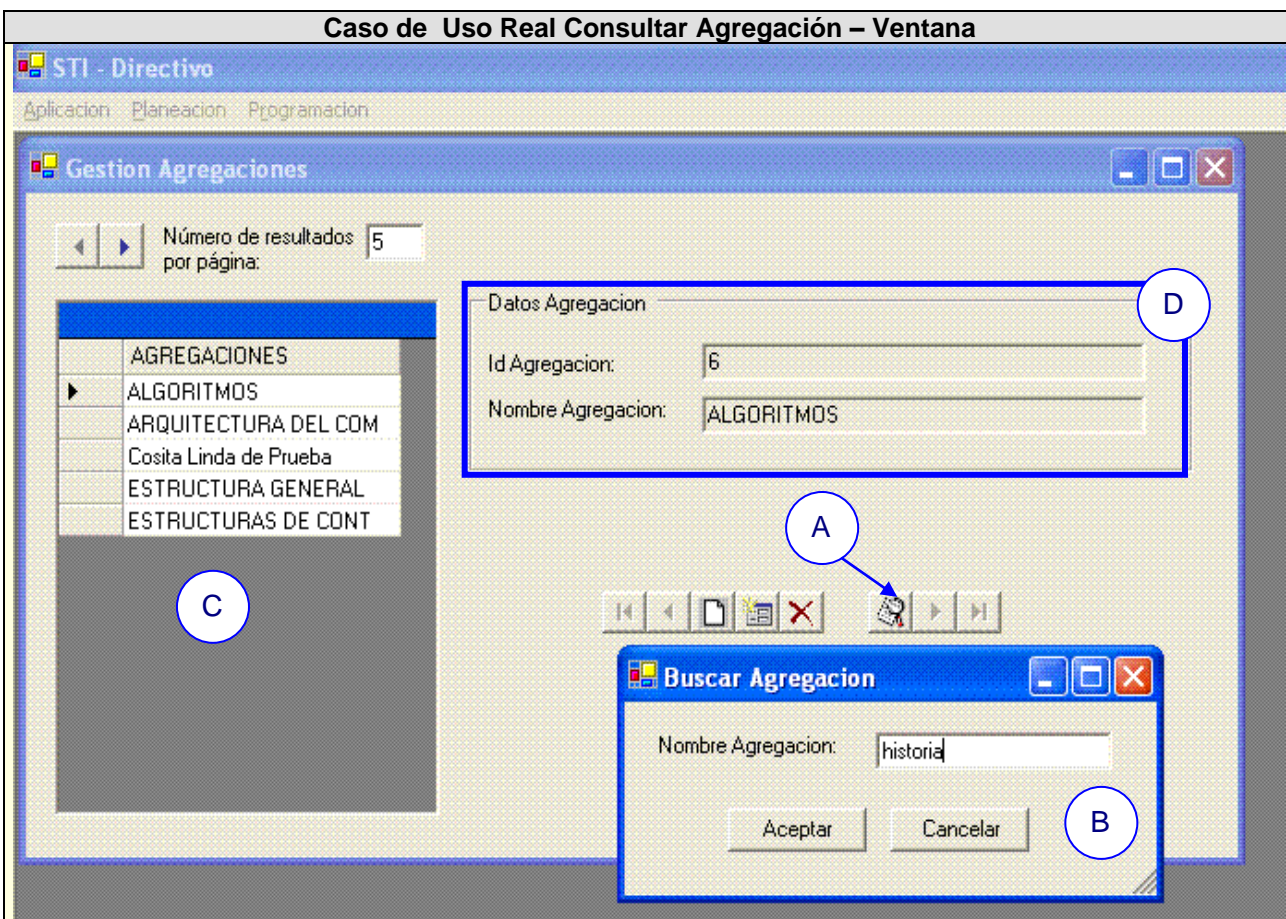
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo guardar la nueva agregación, se retorna un mensaje de error

TRAZABILIDAD

Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Caso de uso Insertar Agregación - Expandido] 	

Tabla 65. Caso de uso Real Insertar Agregación

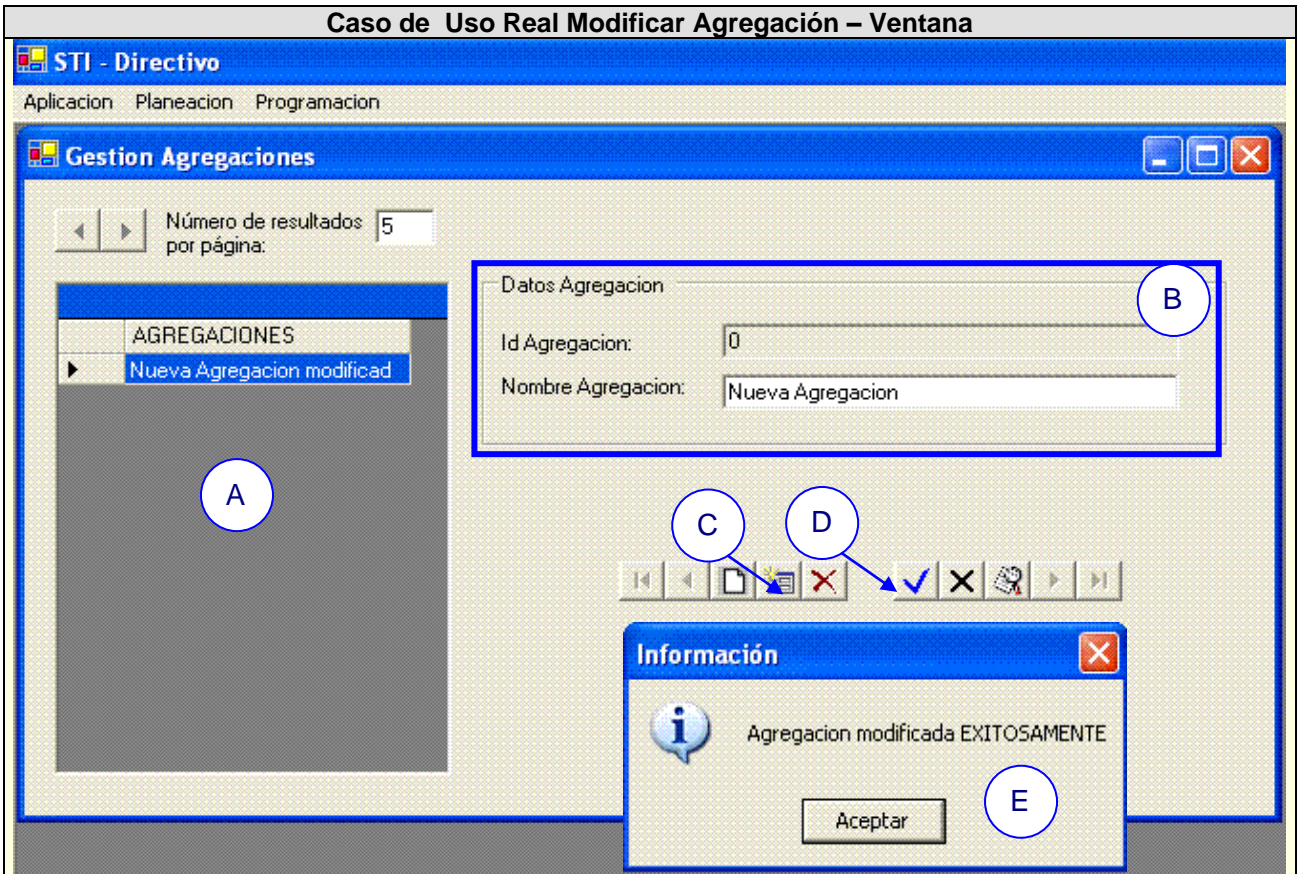
☑ **Caso de Uso Real Consultar Agregación**



CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario pulsa A que es la opción consultar agregación	2. se carga el formulario B para que el usuario digite el criterio de búsqueda
3. El usuario digita en B el nombre de la agregación que desea consultar	4. Se muestran en C todas las agregaciones que coincidan con el criterio de búsqueda y se muestran los demás datos del primer registro en D .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. Se le informa al usuario que la agregación que se desea consultar no existe en la base de datos.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Caso de Uso Consultar Agregación – Expandido] 	

Tabla 66. Caso de Uso Real Consultar Agregación

☑ **Caso de Uso Real Modificar Agregación**



CURSO NORMAL DE LOS EVENTOS

Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A la agregación a modificar	2. Se carga la información de la agregación en B
3. El usuario da clic en C para modificar los campos de la agregación	4. Se habilitan los campos modificables de la agregación en B para que el usuario digite la nueva información
5. El usuario digita la nueva información en B de la agregación, que en este caso es solo el nombre y pulsa D para enviar las modificaciones a la base de datos.	6. Guarda en la base de datos las modificaciones de la agregación y retorna un mensaje de éxito como E .

CURSO ALTERNO 1

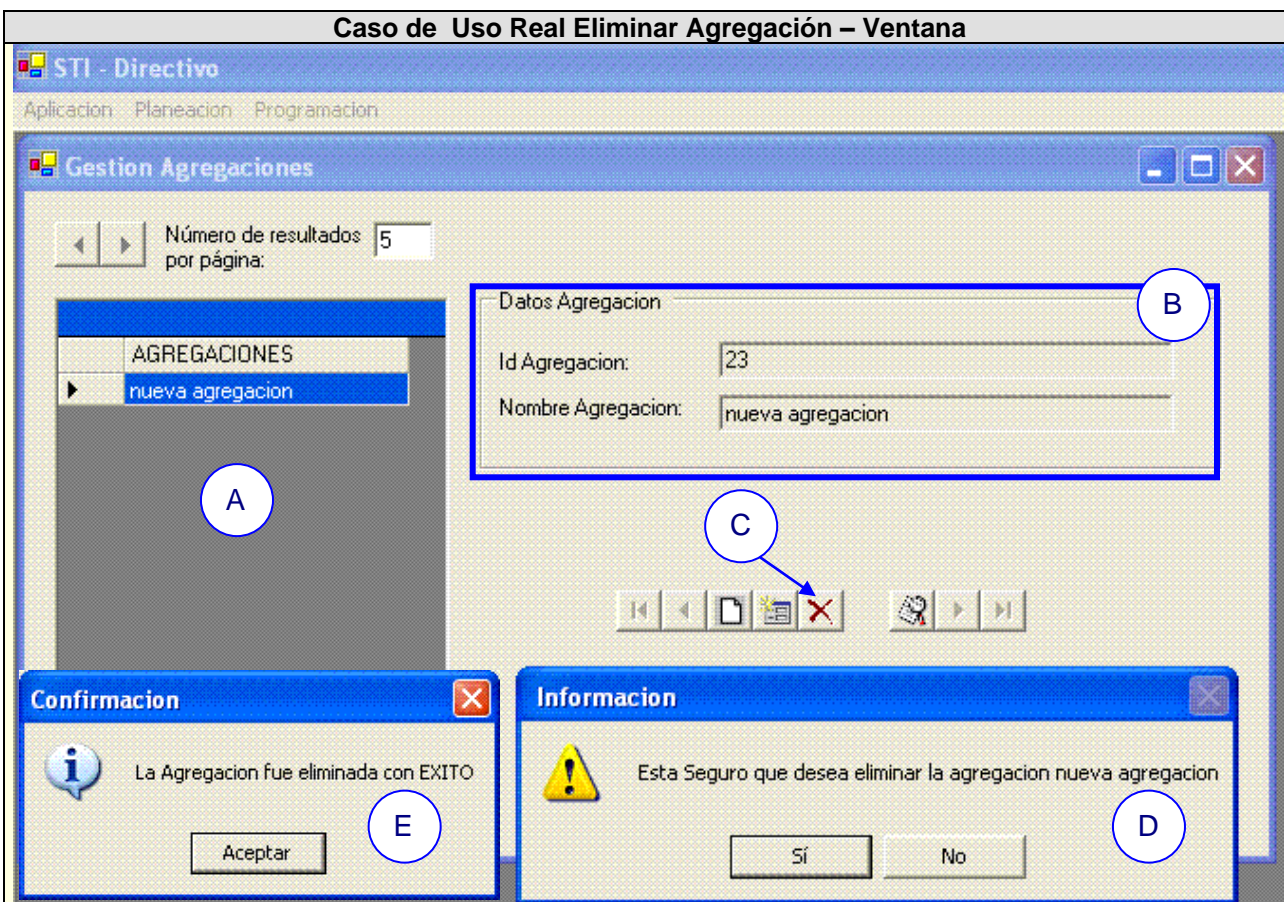
Acción del Actor	Respuesta del Sistema
4.	4. No se pudieron guardar las modificaciones, se retorna un msj de error

TRAZABILIDAD

Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Caso de Uso Modificar Agregación – Expandido] 	

Tabla 67. Caso de Uso Real Modificar Agregación

☑ **Caso de Uso Real Eliminar Agregación**

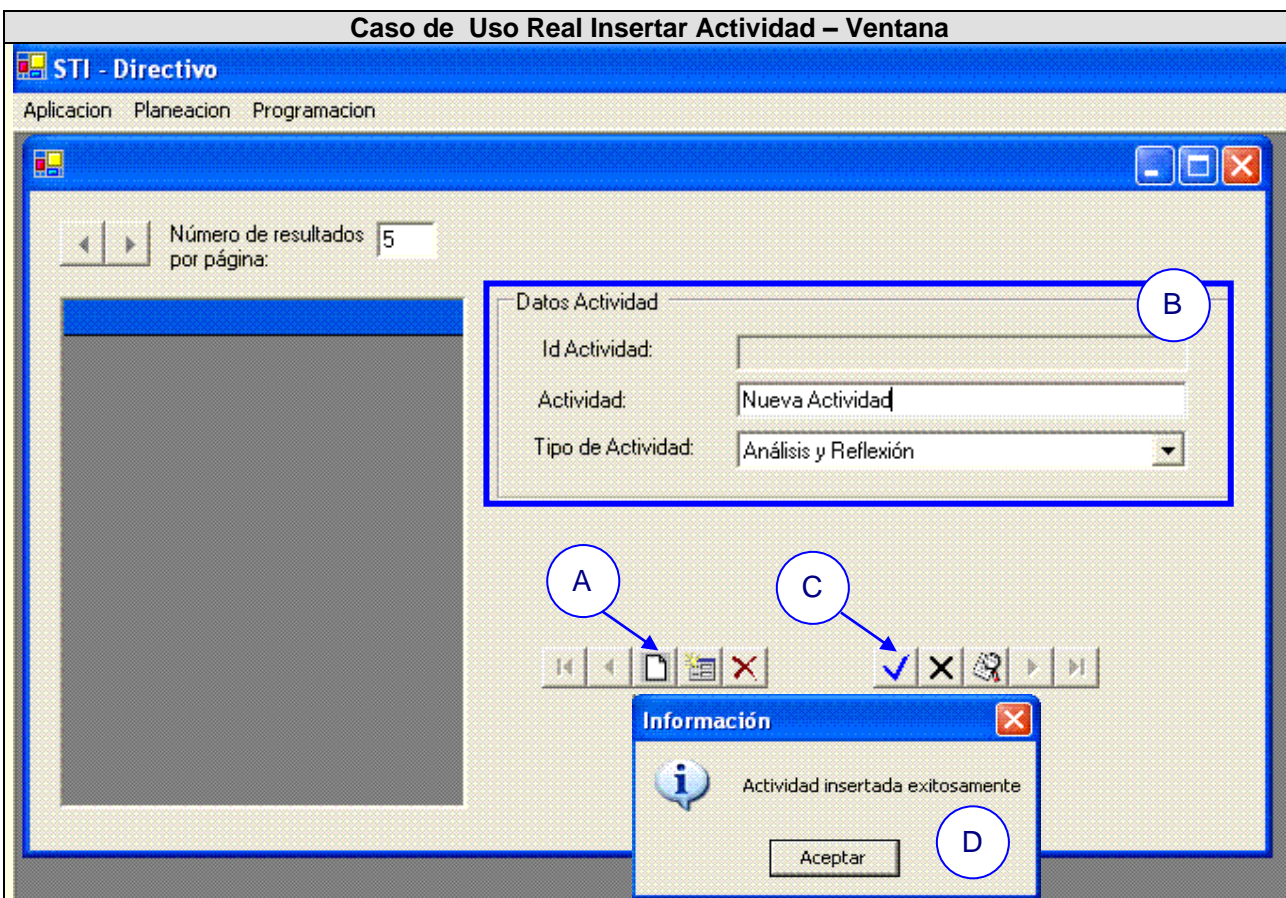


CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A la agregación a eliminar	2. se muestran los datos de la agregación en B
3. El usuario pulsa C para eliminar la agregación escogida	4. Se carga el cuadro D de confirmación para la eliminación
5. El usuario confirma el deseo de eliminar la agregación escogida pulsando aceptar en D	6. Se elimina la agregación de la base de datos y retorna un mensaje de éxito como E .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo eliminar la agregación, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Caso de Uso Eliminar Agregación – Expandido] 	

Tabla 68. Caso de Uso Real Eliminar Agregación

ANEXO 25. Casos de Uso Reales para la Gestión de Actividades

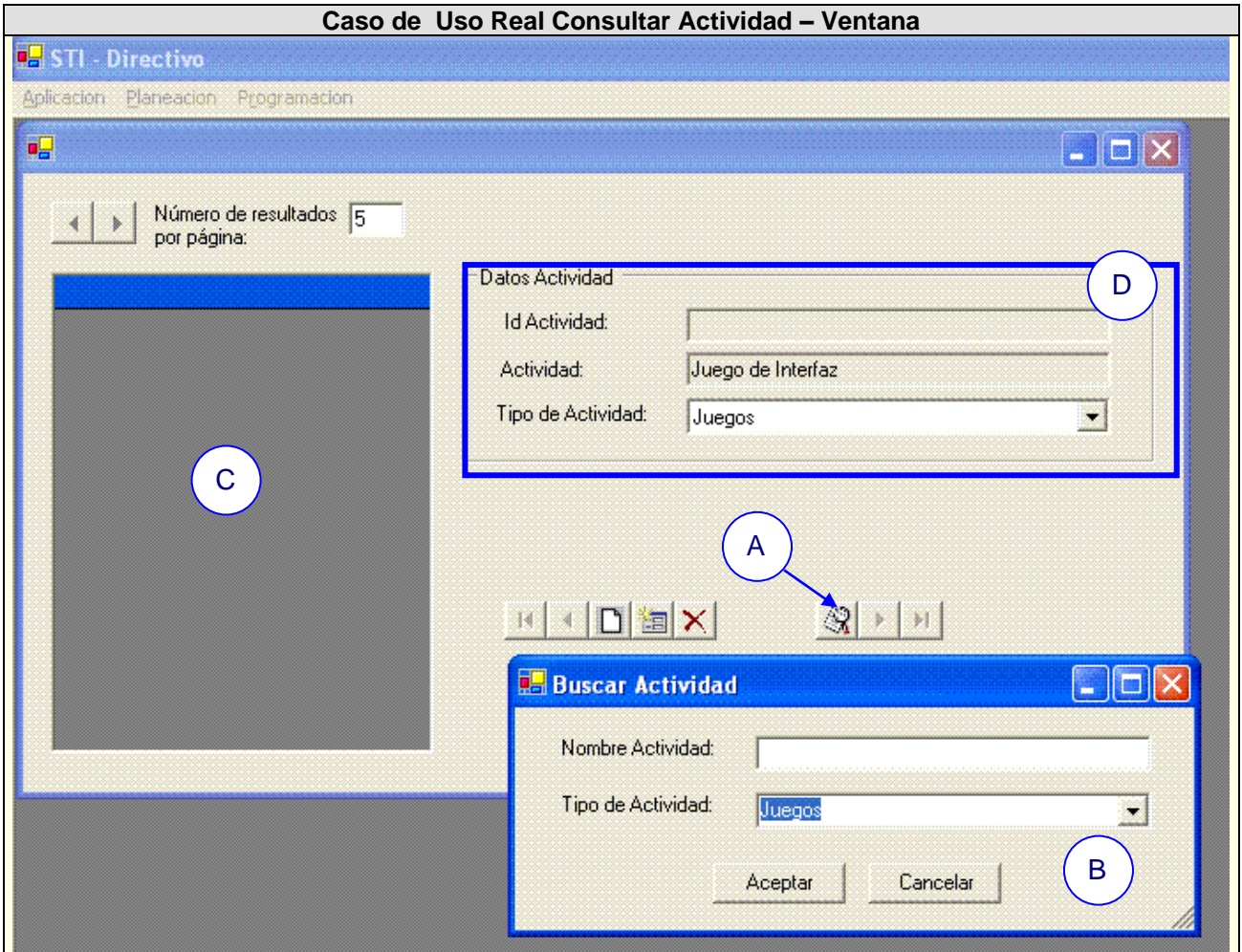
Caso de Uso Real Insertar Actividad



CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario pulsa A que es la opción insertar una nueva actividad	2. se habilitan los campos de B para que el usuario pueda digitar la información
3. El usuario digita el nombre de la nueva actividad y escoge el tipo de actividad en los campos de B y envía a guardar a la base de datos la nueva actividad pulsando C	4. Guarda en la base de datos la nueva actividad y retorna un mensaje de éxito como D y luego se deshabilita C nuevamente.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudo guardar la nueva actividad, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Caso de uso Insertar Actividad - Expandido] 	

Tabla 69. Caso de uso Real Insertar Actividad

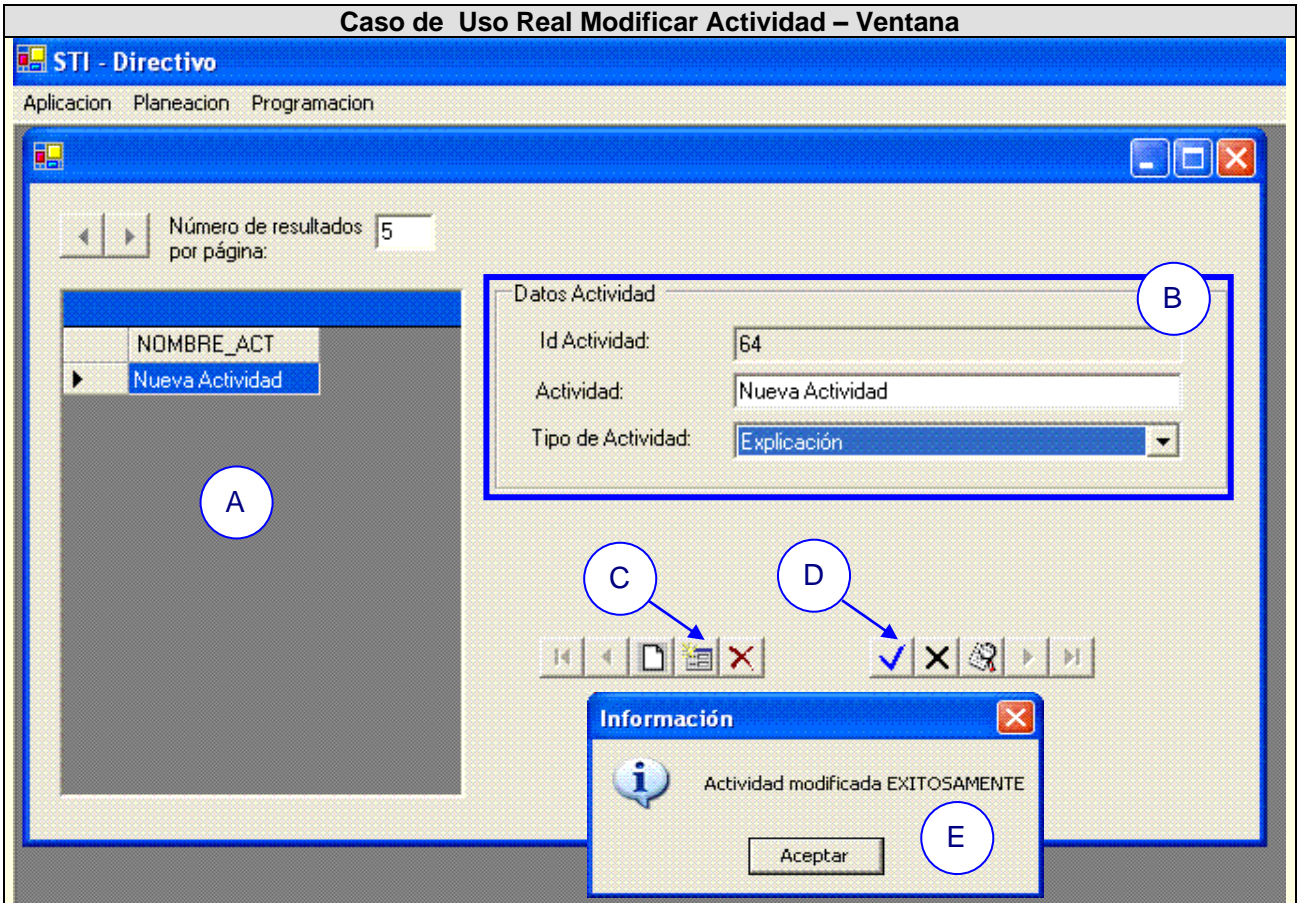
☑ **Caso de Uso Real Consultar Actividad**



CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario pulsa A que es la opción consultar actividad	2. se carga el formulario B para que el usuario digite el criterio de búsqueda o escoja el tipo de actividad que desea buscar
3. El usuario digita el nombre de la actividad o el tipo de actividad que desea consultar en B	4. Se muestran todas las actividades que coincida con el criterio de consulta en C y los demás datos del primer registro en D .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. Se le informa al usuario que la actividad que se desea consultar no existe en la base de datos.
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Caso de Uso Consultar Actividad – Expandido] 	

Tabla 70. Caso de Uso Real Consultar Actividad

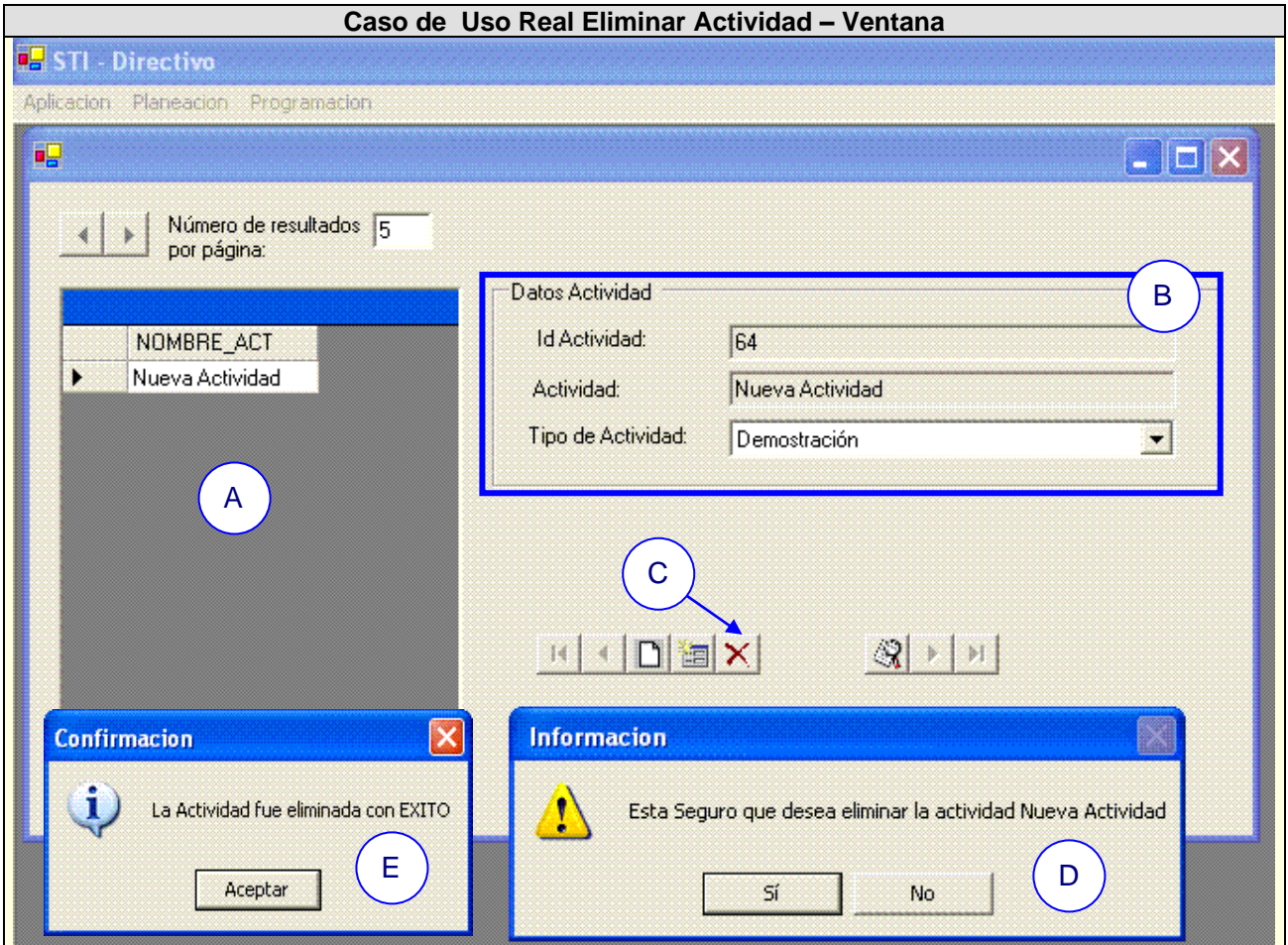
☑ **Caso de Uso Real Modificar Actividad**



CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A la actividad a modificar	2. se cargan los datos de la actividad en B
3. El usuario pulsa C para modificar los campos de la actividad	4. Se habilitan los campos modificables de la actividad en B para que el usuario digite la información
5. El usuario digita el nuevo nombre de la actividad a modificar o puede cambiar el tipo de actividad de la misma en B y pulsa D	6. Se guarda en la base de datos las modificaciones de la actividad y retorna un mensaje de éxito como E , y se deshabilita D nuevamente.
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	4. No se pudieron guardar las modificaciones, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> Artefactos del Análisis: <ul style="list-style-type: none"> [Caso de Uso Modificar Actividad - Expandido] 	

Tabla 71. Caso de Uso Real Modificar Actividad

☑ **Caso de Uso Real Eliminar Actividad**



CURSO NORMAL DE LOS EVENTOS	
Acción del Actor	Respuesta del Sistema
1. Este caso de uso comienza cuando el usuario escoge de A la actividad a eliminar	2. se muestran los datos de la actividad en B y un cuadro de dialogo de confirmación
3. El usuario manda a eliminar la actividad pulsando C	4. Se muestra el cuadro de confirmación D
5. El usuario confirma la eliminación de la actividad pulsando aceptar en D	6. Se elimina la actividad de la base de datos y retorna un mensaje de éxito como E .
CURSO ALTERNO 1	
Acción del Actor	Respuesta del Sistema
4.	6. No se pudo eliminar la actividad, se retorna un mensaje de error
TRAZABILIDAD	
Artefactos Anteriores	Artefactos Posteriores
<ul style="list-style-type: none"> • Artefactos del Análisis: <ul style="list-style-type: none"> ○ [Caso de Uso Eliminar Actividad – Expandido] 	

Tabla 72. Caso de Uso Real Eliminar Actividad

ANEXO 26. Diagramas de Secuencia Gestión Ítems

☑ Diagrama de Secuencia Insertar Ítems

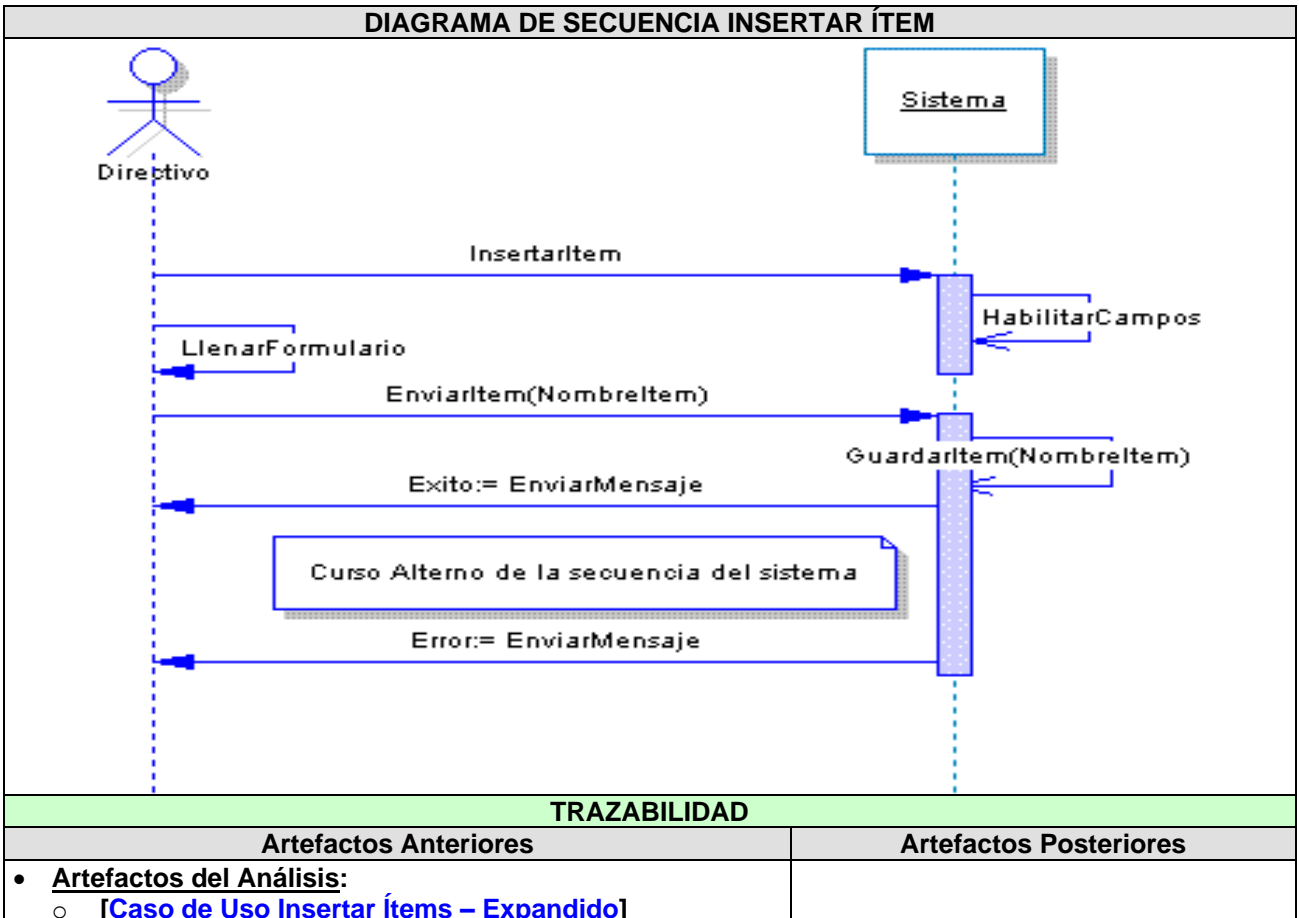


Tabla 73. Diagrama de Secuencia Insertar Ítem

☑ Diagrama de Secuencia Consultar Ítems

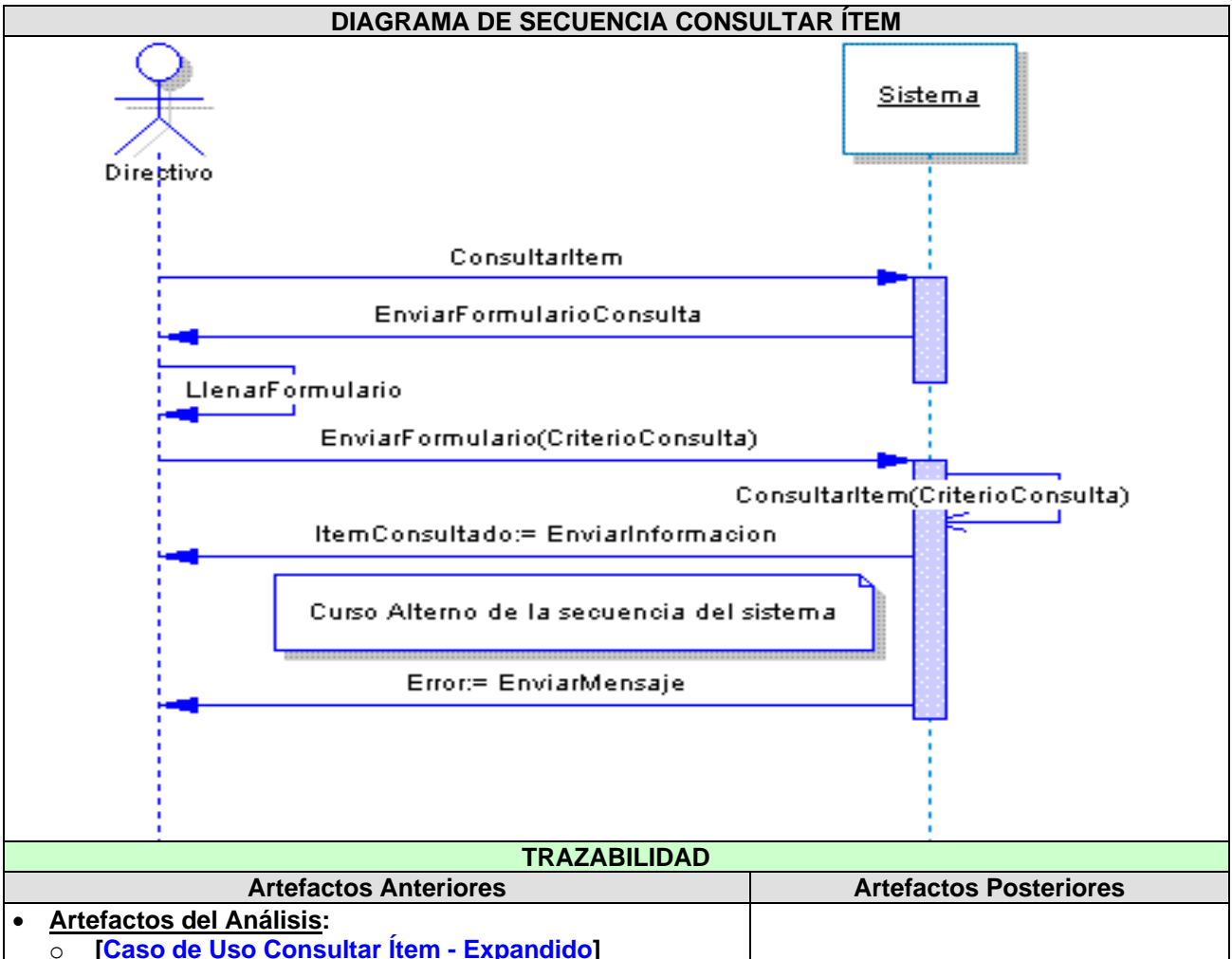


Tabla 74. Diagrama de Secuencia Consultar Ítem

☑ Diagrama de Secuencia Modificar Ítems

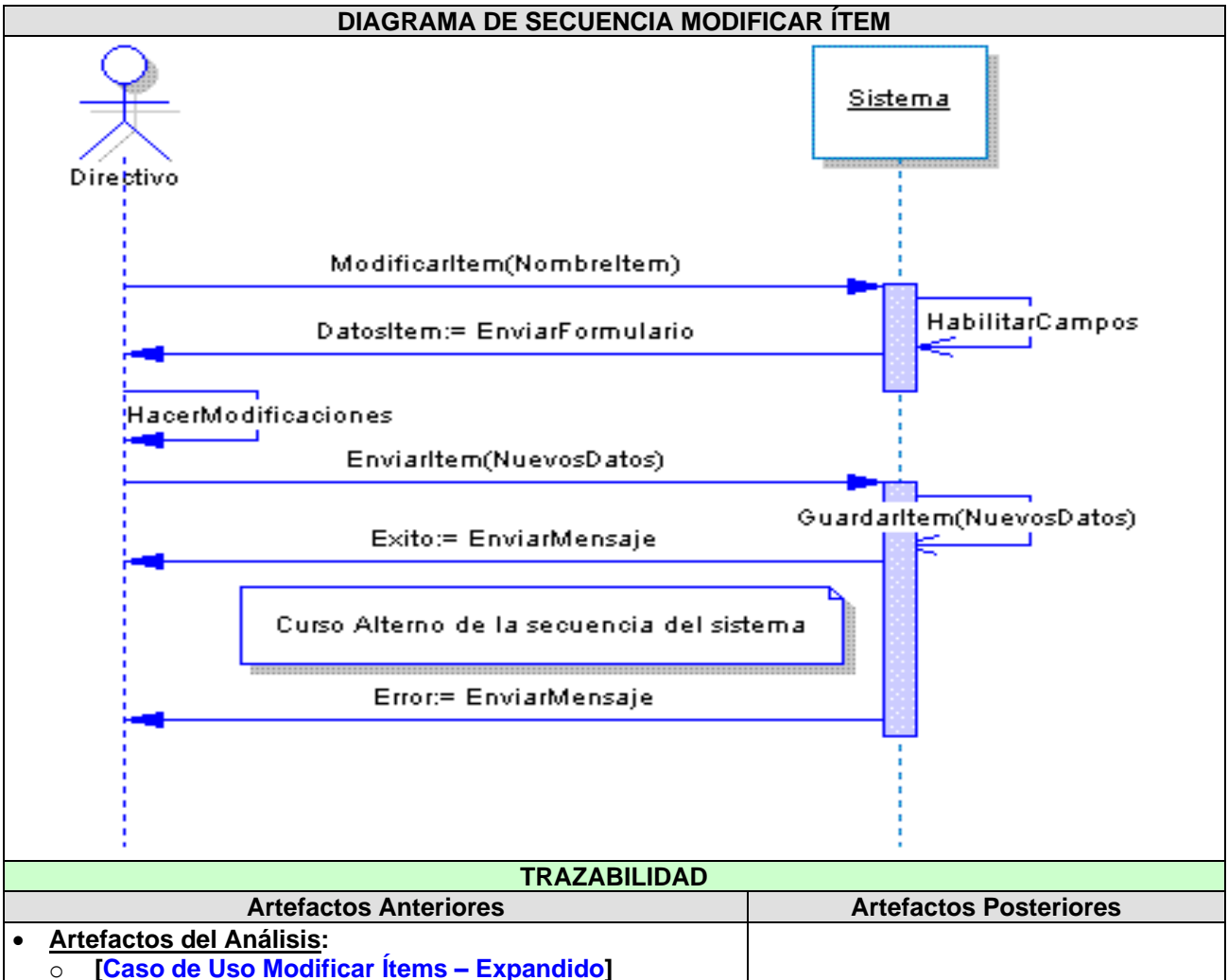


Tabla 75. Diagrama de Secuencia Modificar Ítem

☑ Diagrama de Secuencia Eliminar Ítems

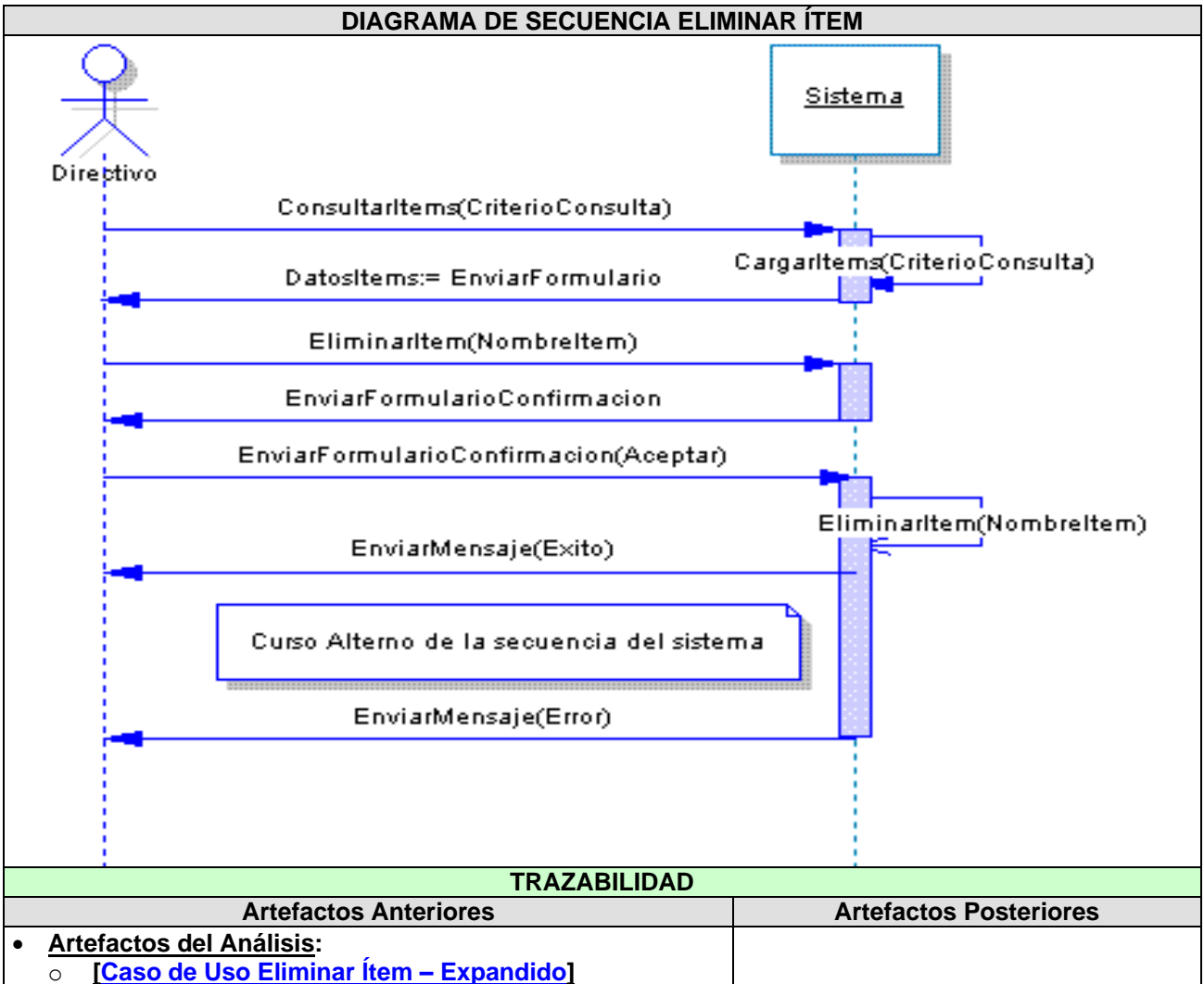


Tabla 76. Diagrama de Secuencia Eliminar Ítem

ANEXO 27. Diagramas de Secuencia Gestión Contenidos

☑ Diagrama de Secuencia Definir Contenidos Programáticos

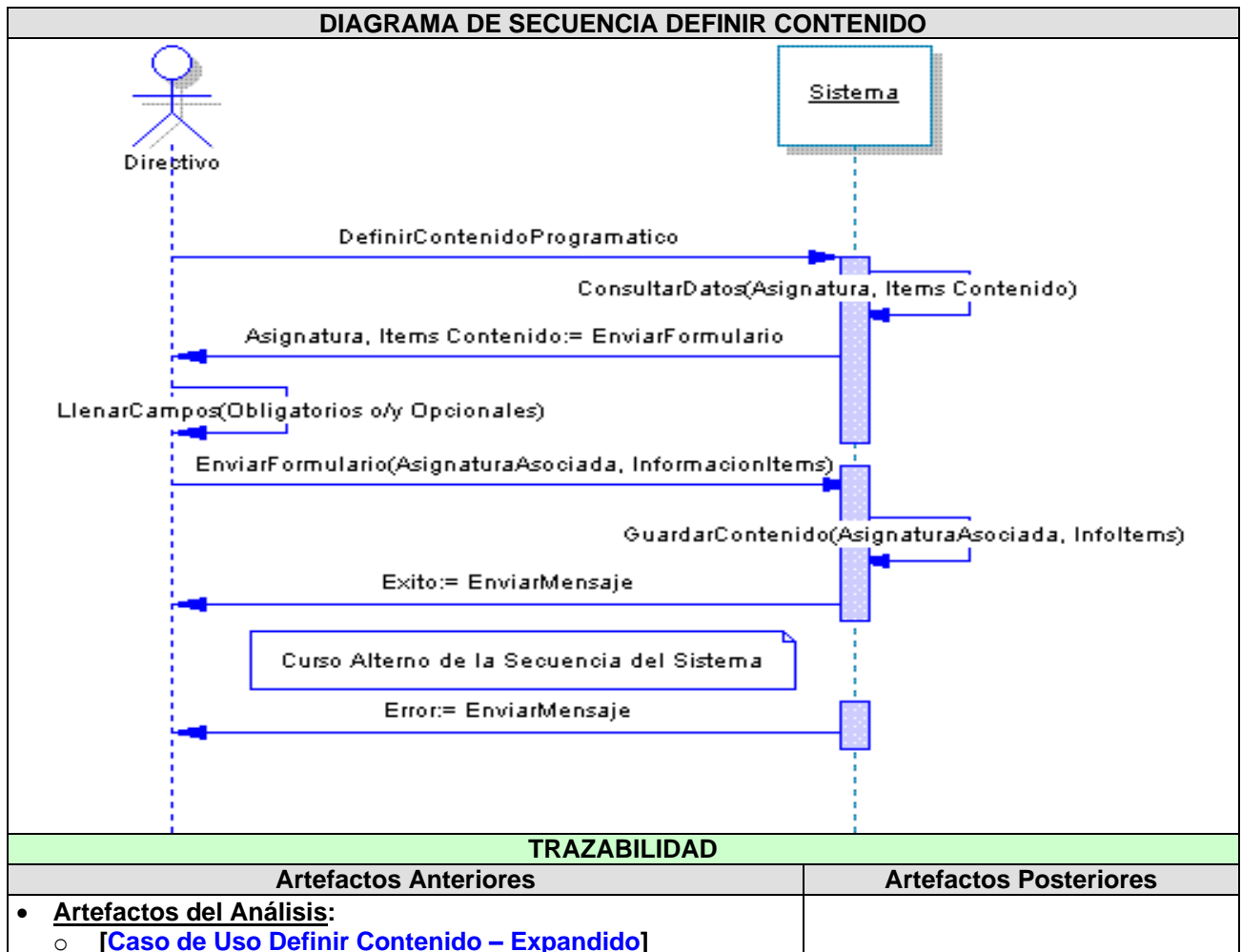


Tabla 77. Diagrama de Secuencia Definir Contenido

☑ Diagrama de Secuencia Consultar Contenidos Programáticos

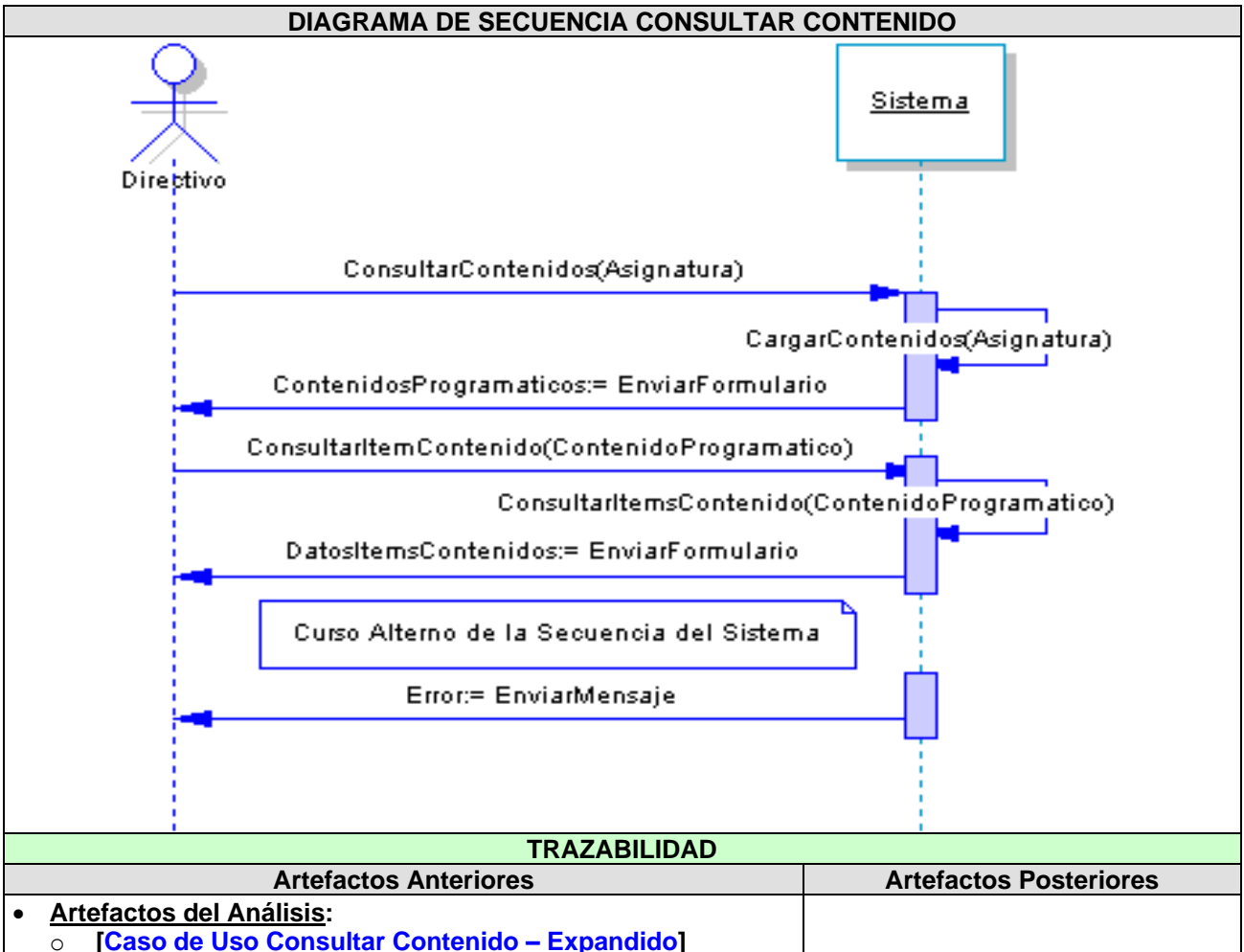


Tabla 78. Diagrama de Secuencia Consultar Contenido

☑ Diagrama de Secuencia Activar Contenidos Programáticos

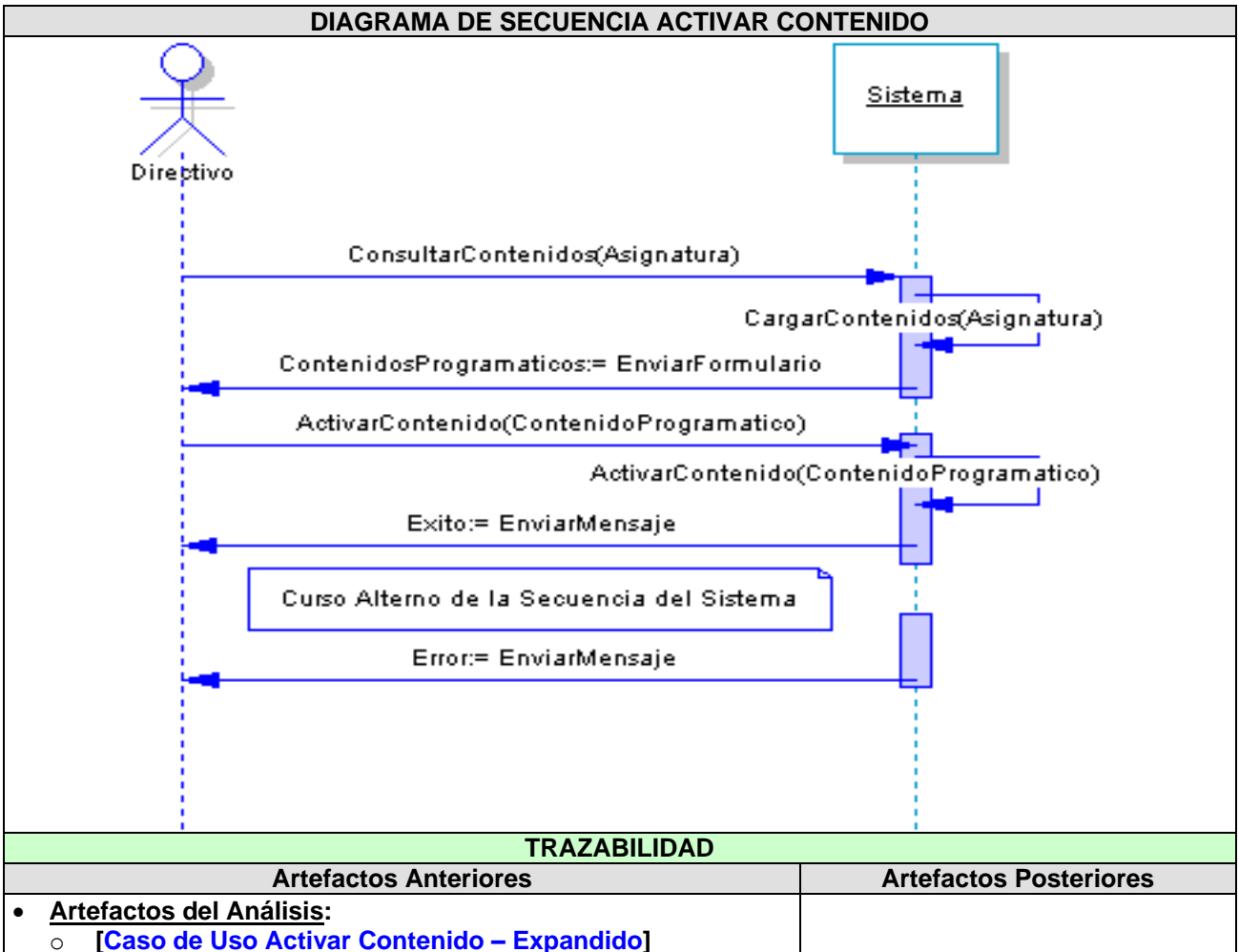


Tabla 79. Diagrama de Secuencia Activar Contenido

☑ Diagrama de Secuencia Modificar Contenido

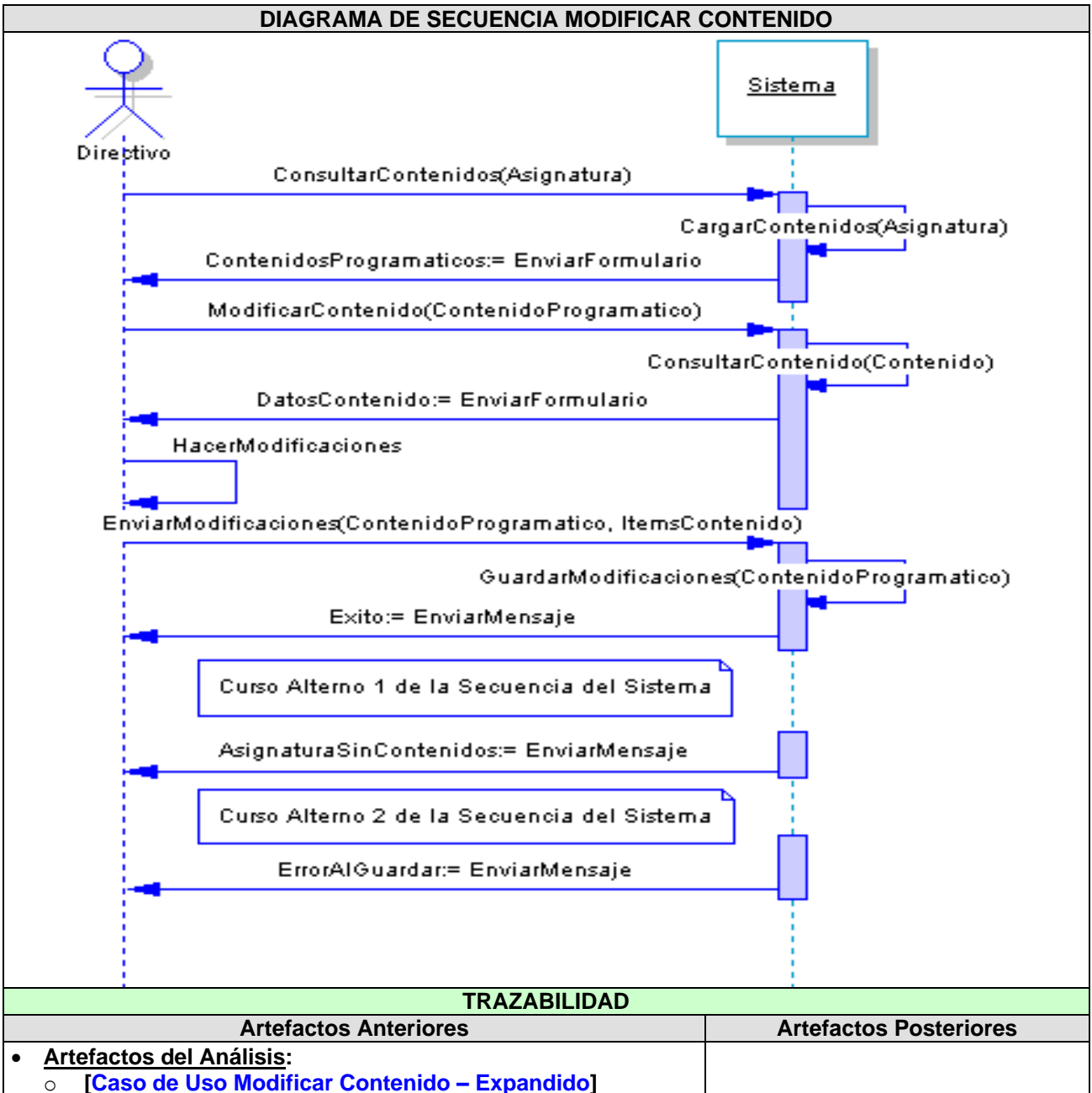


Tabla 80. Diagrama de Secuencia Modificar Contenido

ANEXO 28. Diagramas de Secuencia Gestión Agregaciones

☑ Diagrama de Secuencia Insertar Agregaciones

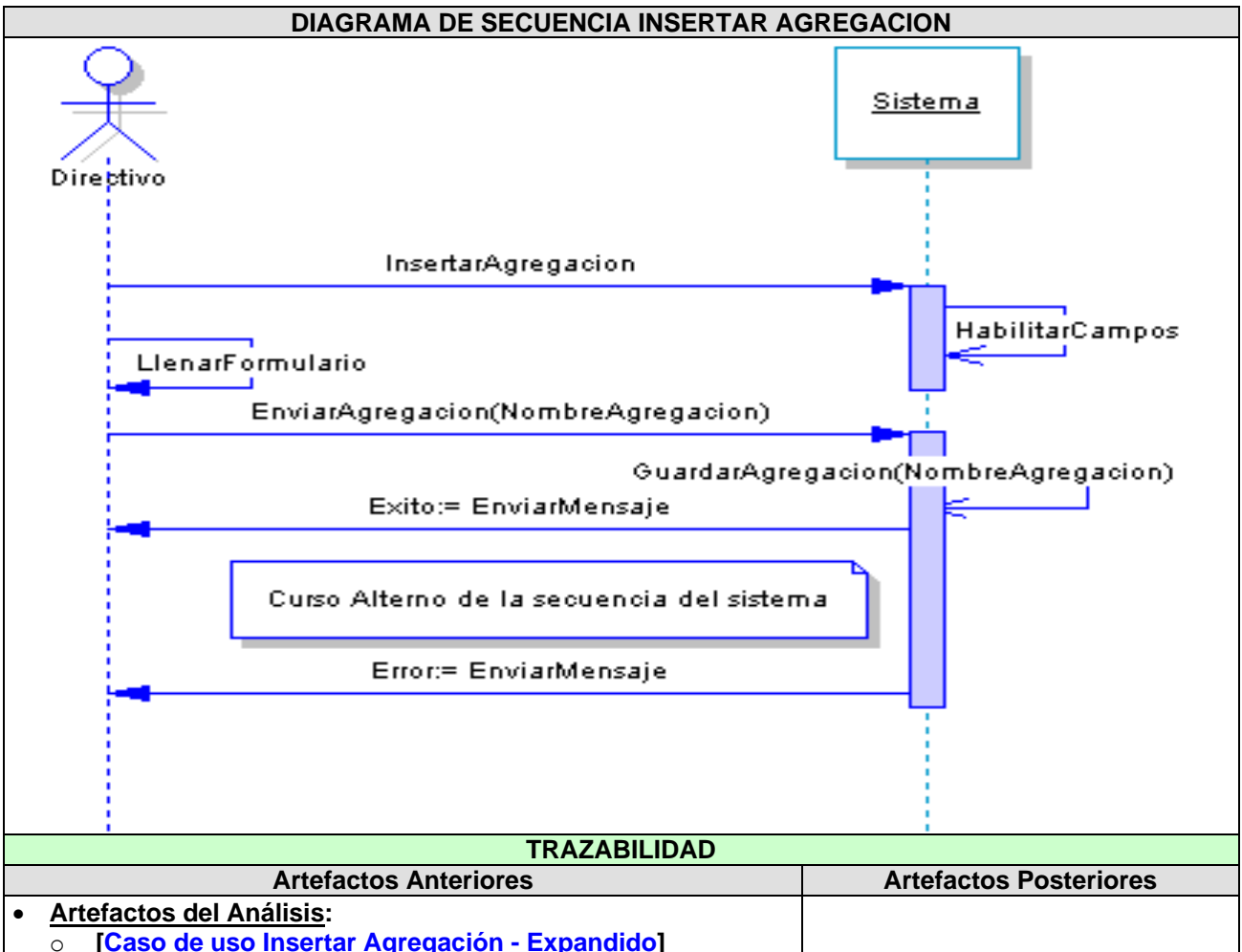


Tabla 81. Diagrama de Secuencia Insertar Agregación

☑ Diagrama de Secuencia Consultar Agregación

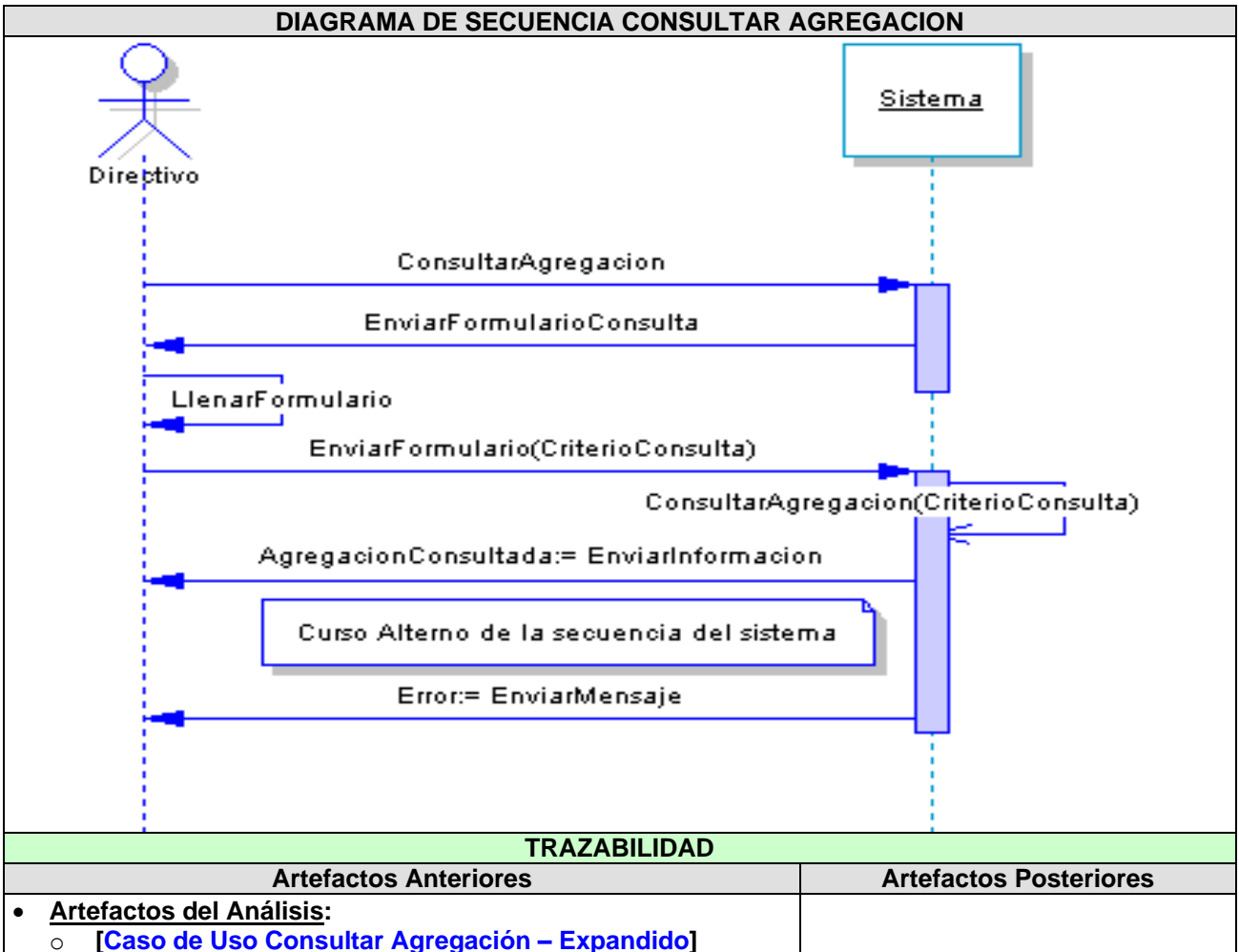


Tabla 82. Diagrama de Secuencia Consultar Agregación

☑ Diagrama de Secuencia Modificar Agregación

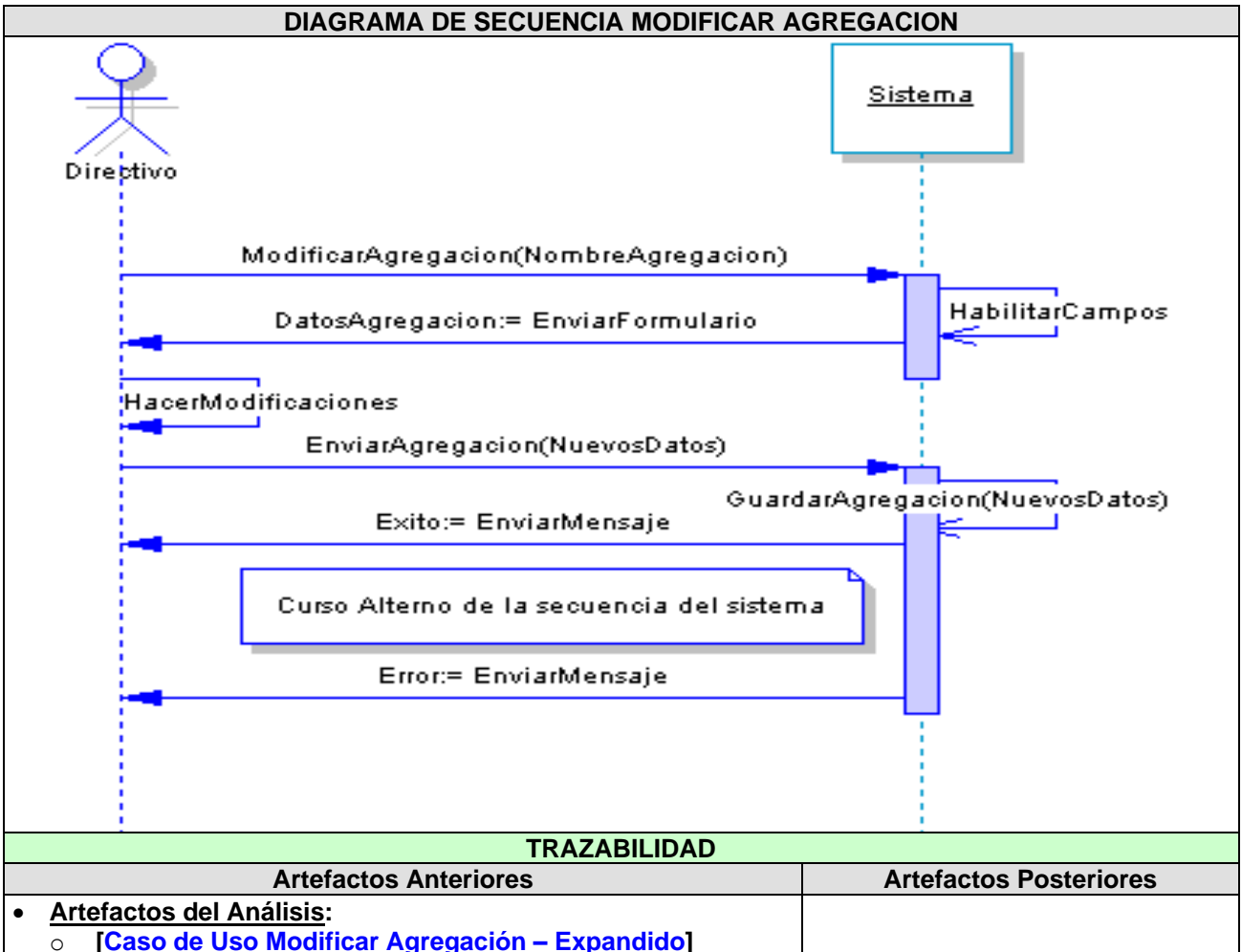


Tabla 83. Diagrama de Secuencia Modificar Agregación

☑ Diagrama de Secuencia Eliminar Agregación

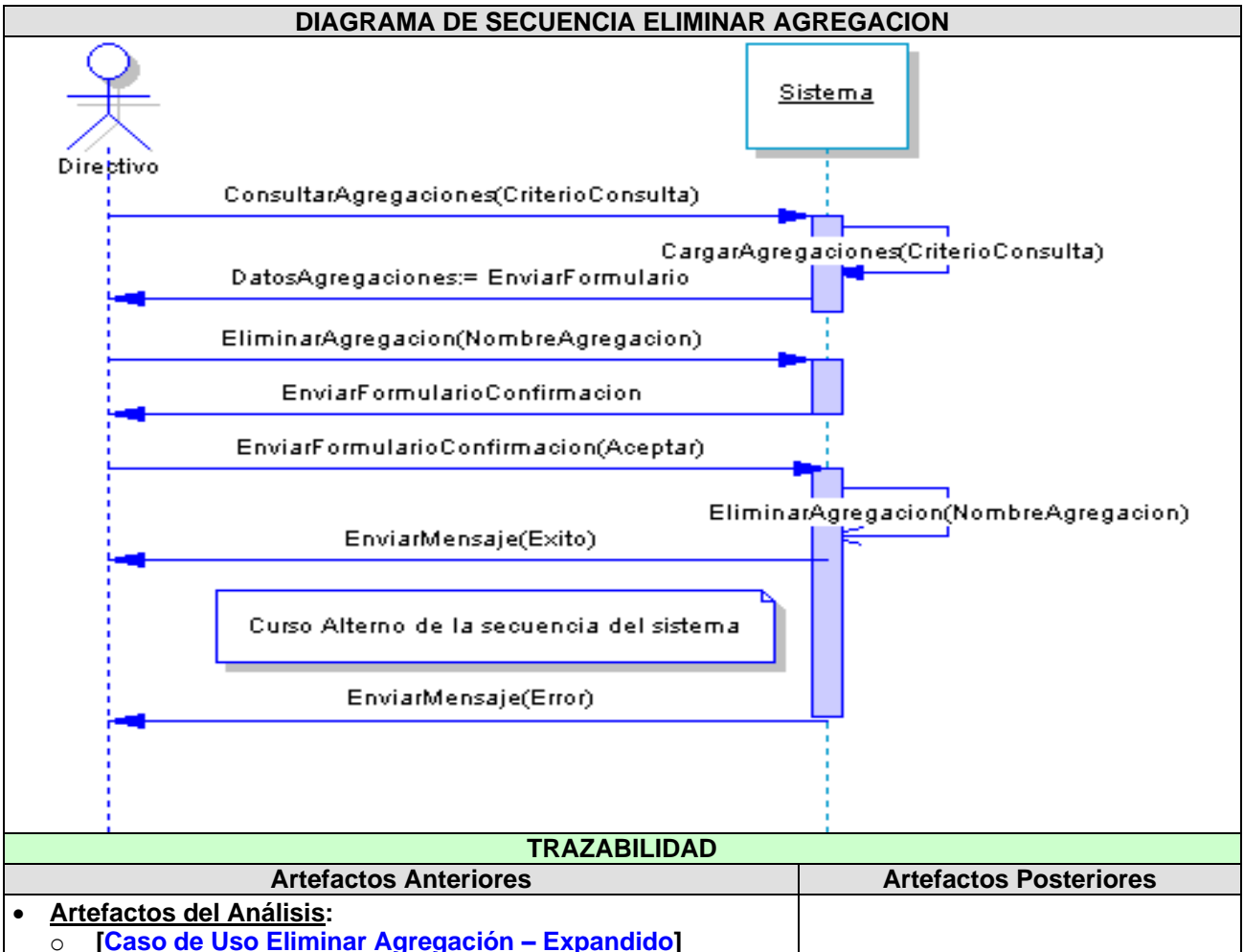


Tabla 84. Diagrama de Secuencia Eliminar Agregación

ANEXO 29. Diagramas de Secuencia Gestión Actividades

☑ Diagrama de Secuencia Insertar Actividad

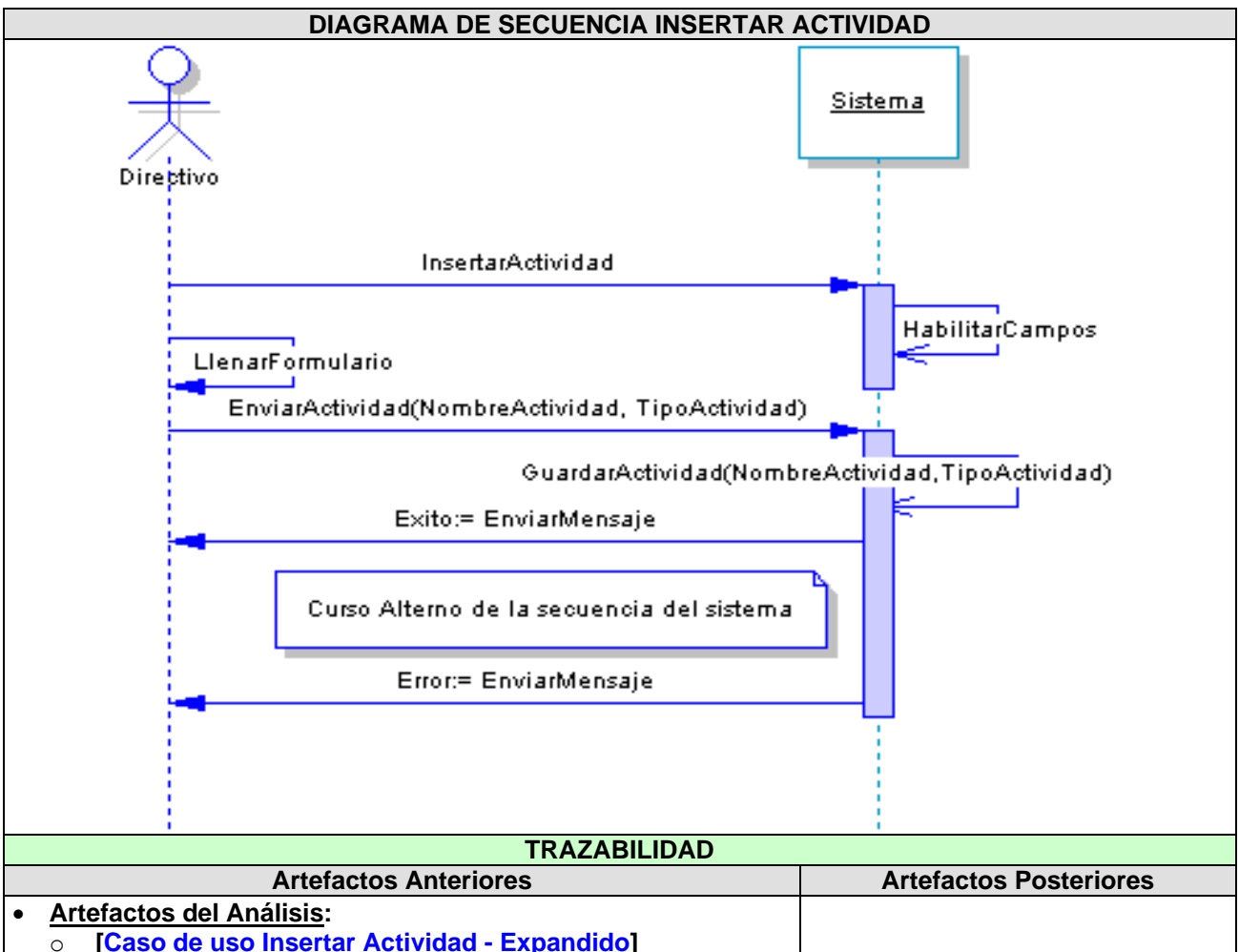


Tabla 85. Diagrama de Secuencia Insertar Actividad

☑ Diagrama de Secuencia Consultar Actividad

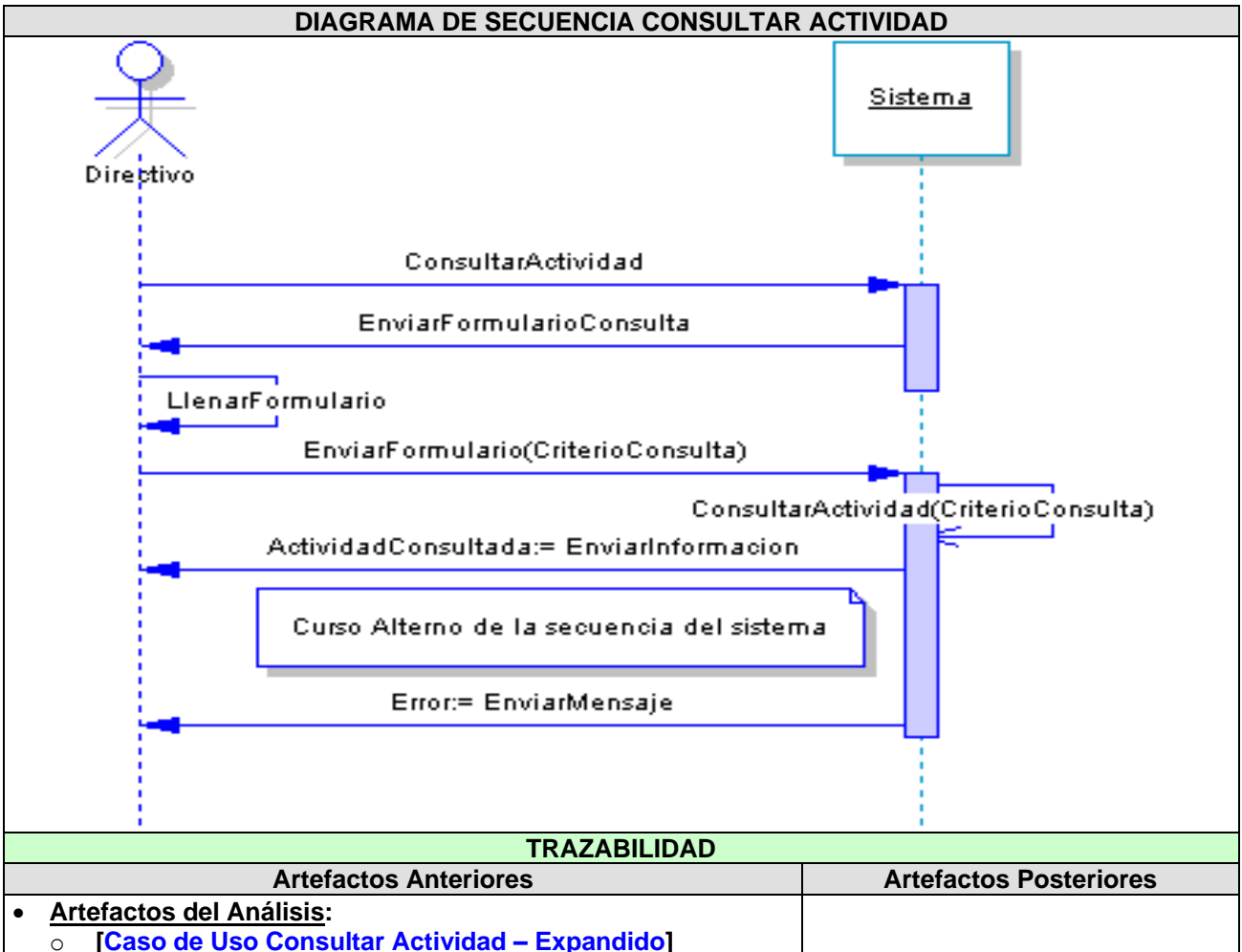


Tabla 86. Diagrama de Secuencia Consultar Actividad

☑ Diagrama de Secuencia Modificar Actividad

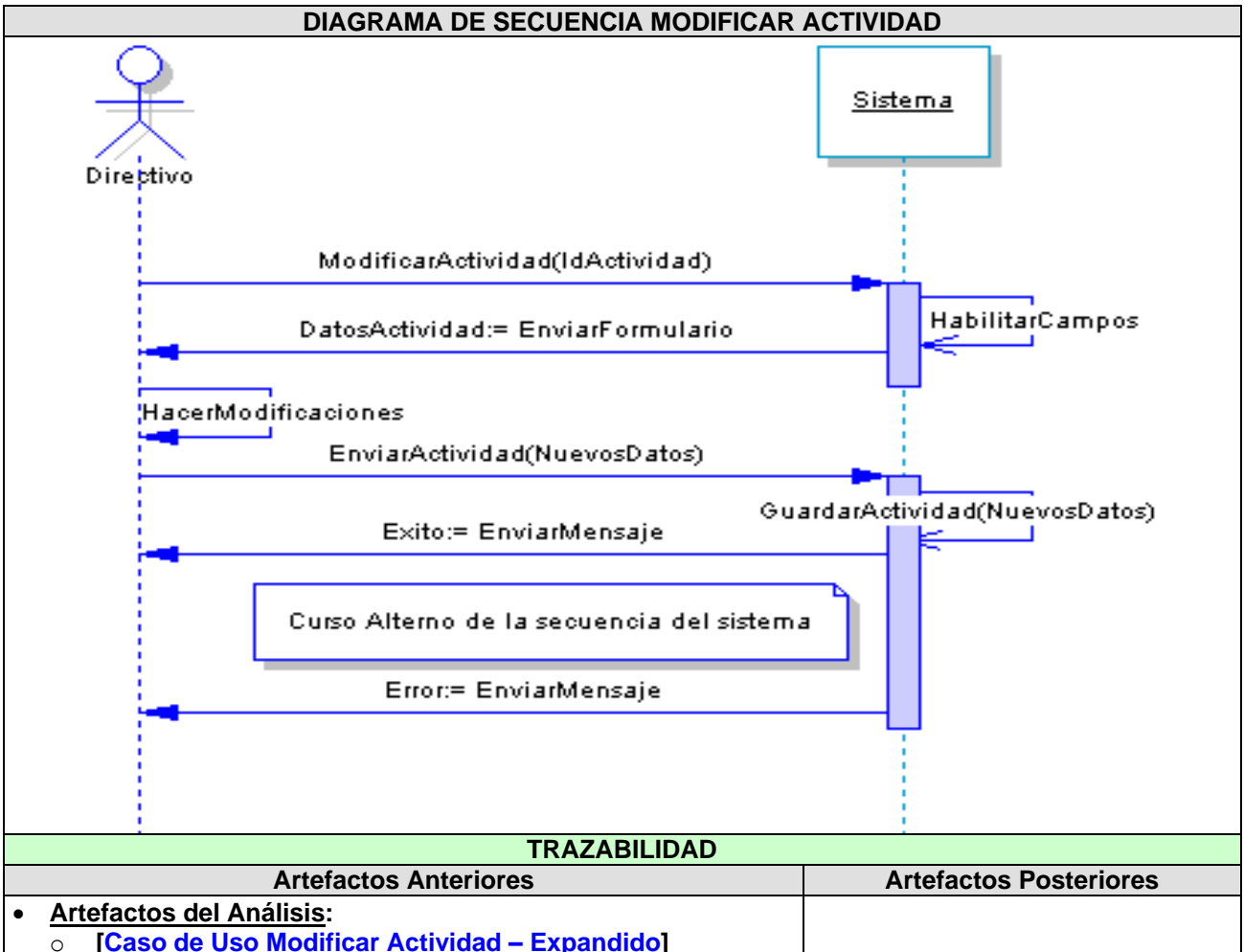


Tabla 87. Diagrama de Secuencia Modificar Actividad

☑ Diagrama de Secuencia Eliminar Actividad

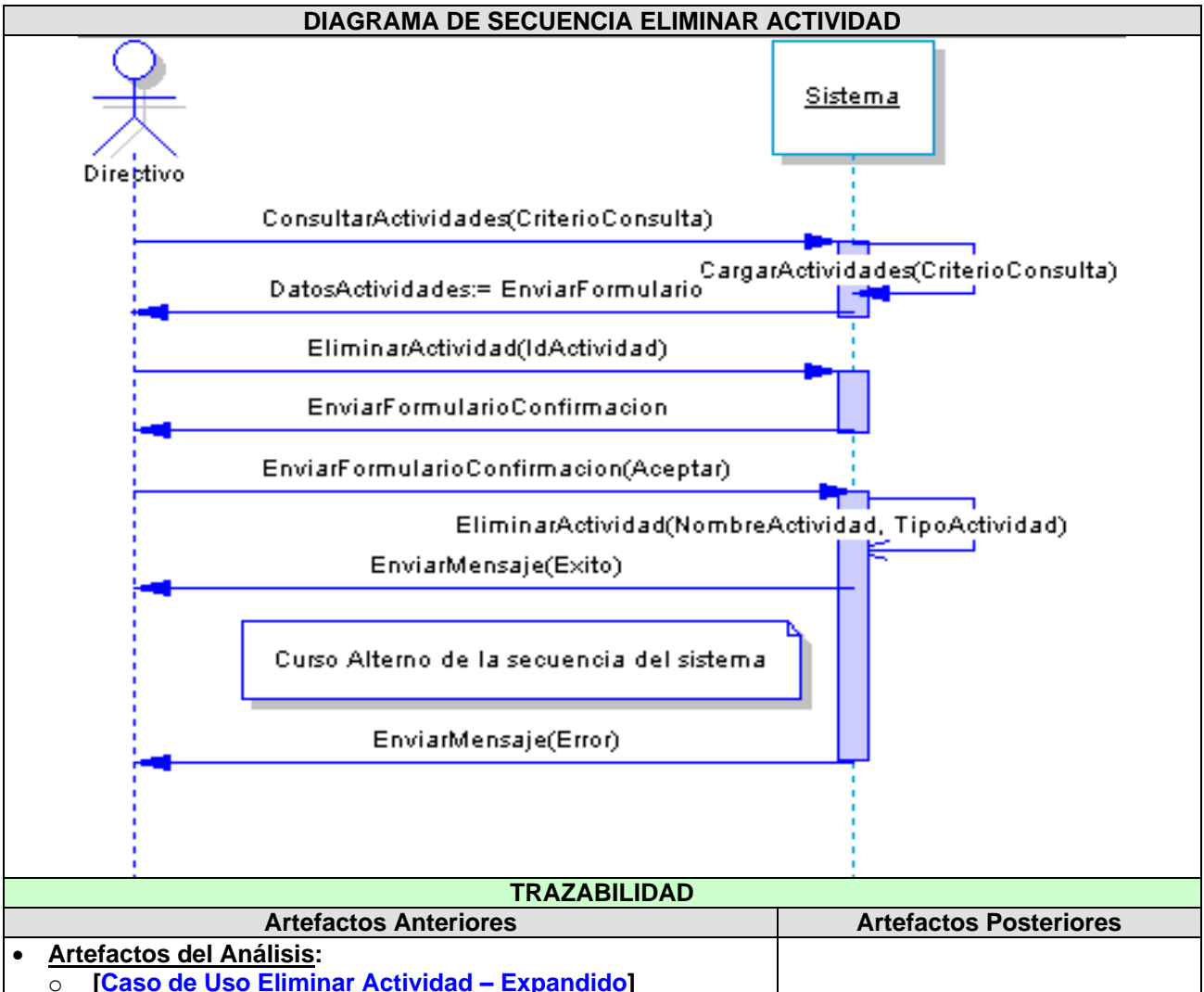
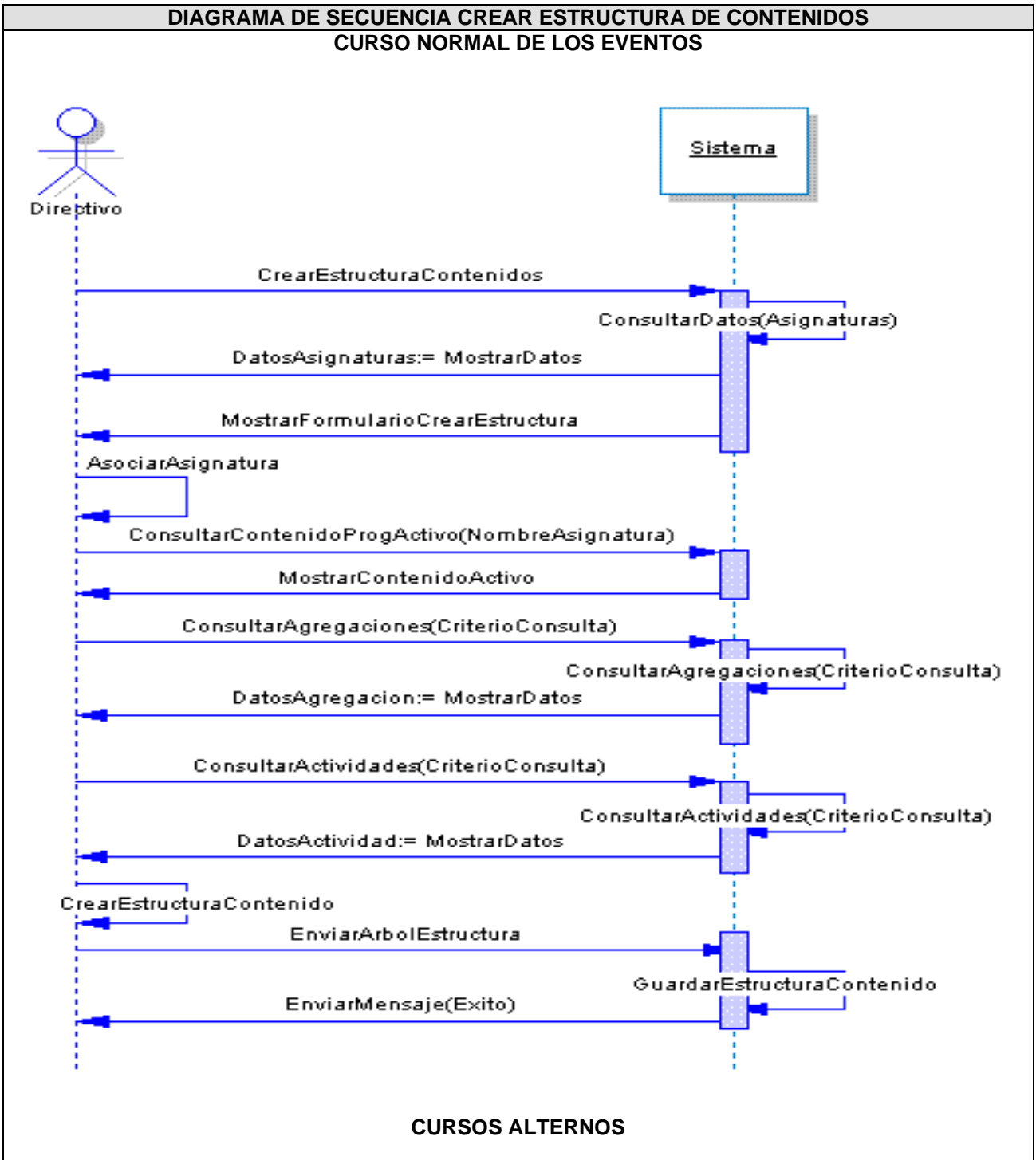


Tabla 88. Diagrama de Secuencia Eliminar Actividad

ANEXO 30. Diagrama de Secuencia Crear Estructura de Contenidos



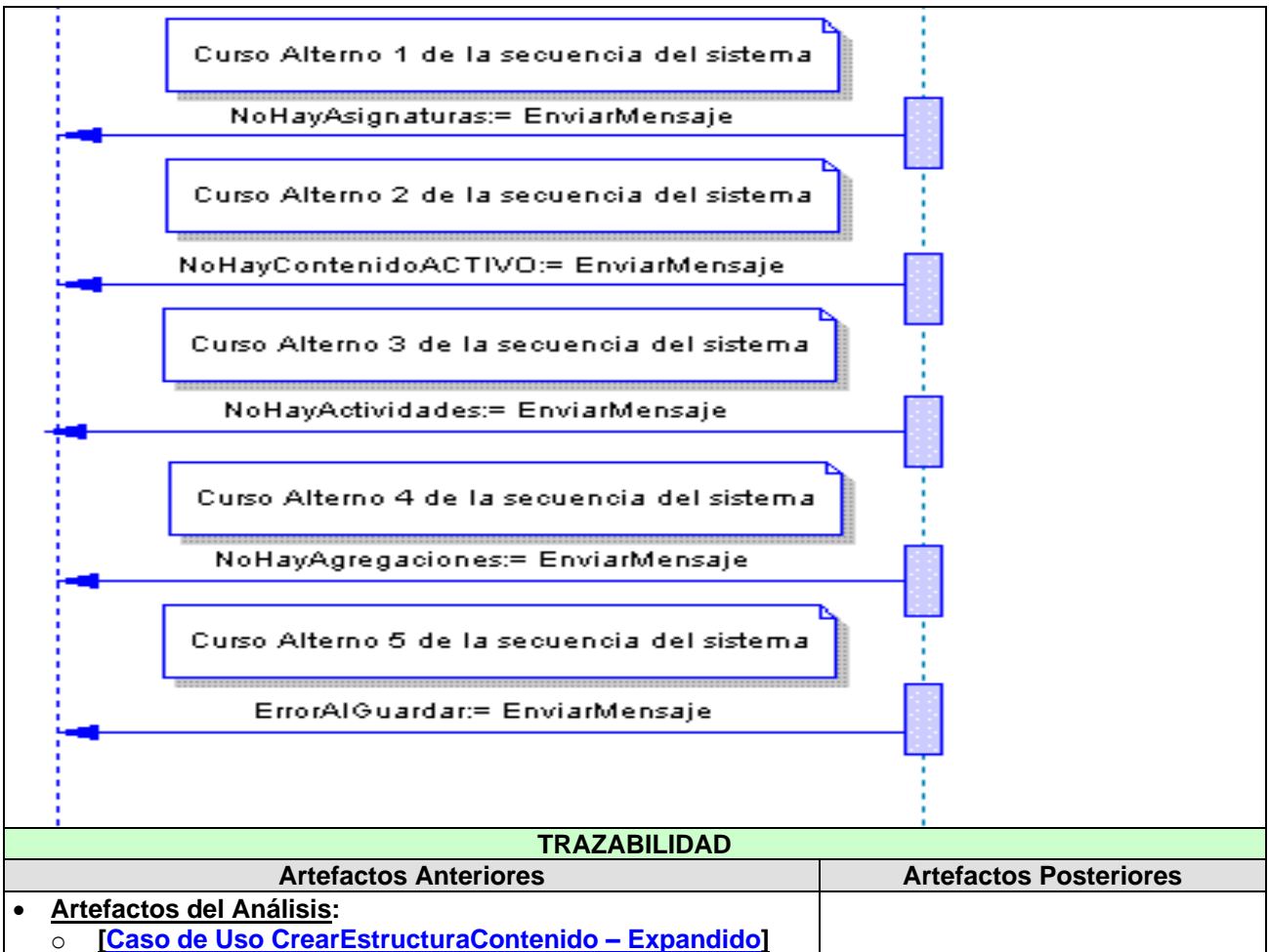


Tabla 89. Diagrama de Secuencia Crear Estructura

ANEXO 31. Estructura del Algoritmo ID3

La estructura del ID3 es iterativa. Se escoge un subconjunto de los casos de entrenamiento, el cual se denomina *ventana*, y se forma un árbol de decisión a partir de ese subconjunto. Este árbol clasifica correctamente los casos de la *ventana*. Luego se intenta clasificar con ese árbol los restantes casos del conjunto de entrenamiento. Si el árbol da una correcta clasificación para todos los casos, entonces termina el proceso. Si no, entonces se seleccionan unos de los objetos incorrectamente clasificados, se agregan a la ventana y se repite el proceso, de esta manera, es usual hallar árboles de decisión correctos luego de unas pocas iteraciones para conjuntos de entrenamiento de hasta 30000 objetos, descritos por medio de hasta 50 atributos.

El árbol de decisión se recorre desde la raíz, y tanto en ella como en cada uno de los demás nodos se decide cual rama tomar basándonos en el valor de algún atributo del ejemplar que se esté clasificando, hasta llegar a un nodo terminal

(hoja), que, como se mencionó antes, corresponde a la clase en que queda clasificado el ejemplar.

Suponiendo que el conjunto de casos de entrenamiento posee p ejemplares de clase P y n ejemplares de clase N, para el algoritmo ID3 asumimos lo siguiente:

1- Un árbol de decisión correcto clasificará los ejemplares, en cada clase, en las mismas proporciones en que están representados en el conjunto de entrenamiento. O sea que tomando un ejemplar al azar, este se clasificará como P con probabilidad $p/(p + n)$, y como clase N con probabilidad $n/(p + n)$.

2- El árbol de decisión se puede considerar como fuente de un mensaje: " P " o " N ", y la información esperada para generar este mensaje será entonces (con logaritmos base 2):

$$I(p,n) = -(p/(p + n)) * \text{LOG}(p/(p + n)) - (n/(p + n)) * \text{LOG}(n/(p + n))$$

Si usamos un atributo A, que tenga v valores posibles, como raíz del árbol, las v ramas salientes de la raíz conducen a su vez a v sub-árboles. Tomando un la i -ésima rama, que tiene p_i objetos clase P y n_i objetos clase N, la información requerida para ese i -ésimo sub-árbol será $I(p_i, n_i)$, y entonces la información esperada requerida para el árbol con A como raíz se obtiene como el promedio ponderado:

$$E(A) = ((p_1+n_1) * I(p_1, n_1) + (p_2+n_2) * I(p_2, n_2) + \dots + (p_v+n_v) * I(p_v, n_v)) / (p+n)$$

Donde la ponderación para la i -ésima rama es la proporción de los objetos que pertenecen a esa rama, por lo tanto, la información ganada la ramificarse gracias al atributo A es:

$$\text{Ganancia}(A) = I(p,n) - E(A)$$

un buen criterio parece ser escoger el atributo que gana la mayor información. ID3 examina todos los atributos y escoge el de máxima ganancia, forma la ramificación y usa el mismo proceso recursivamente para formar sub-árboles a partir de los v nodos generados

ANEXO 32. Pseudocódigo del Algoritmo ID3

Algoritmo ID3(Ejemplos, Atributo_salida, Atributos)

Ejemplos: Ejemplos de aprendizaje.

Atributo_salida: Atributo a predecir por el árbol.

Atributos: Lista de atributos a comprobar por el árbol.

- (1) Crear una *raíz* para el árbol.
- (2) **Si** todos los ejemplos son positivos **Return**(*raíz*,+)
- (3) **Si** todos los ejemplos son negativos **Return**(*raíz*,-)
- (4) **Si** *Atributos*= \emptyset **Return**(*raíz*,*l*)
(*l* es el máximo valor común de *Atributo_salida* en *Ejemplos*)

Si ninguna de las anteriores condiciones se cumple

Begin

- (1) *A* es el atributo de *Atributos* que mejor clasifica *Ejemplos*
- (2) El atributo de decisión para *raíz* es *A*
- (3) Para cada posible valor *vi* de *A*
 - (3.1) Añadir una rama a *raíz* con el test *A=vi*
 - (3.2) *Ejemplos_vi* es el subconjunto de *Ejemplos* con valor *vi* para *A*
 - (3.3) **Si** *Ejemplos_vi*= \emptyset
entonces añadir un nodo (*n*,*l*) a partir de la rama creada.
(*l* es el máximo valor común de *Atributo_salida* en *Ejemplos*).
sino añadir a la rama creada el subárbol
ID3(*Ejemplos_vi*, *Atributo_salida*, *Atributos*-{*A*})

End

Return(*raíz*)

ANEXO 33. Pseudocodigo Algoritmo Deliberar

- (1) Calculamos la cantidad de rutas positivas del árbol
- (2) Ubicamos las rutas positivas recorriendo el árbol y agrupando los valores de estas
- (3) Separamos los atributos Sistema y Estudiante de la experiencia del estudiante que se desea diagnosticar
- (4) Encontramos los valores de los atributos en la experiencia del estudiante

Para cada Ruta Positiva

Compara los atributos del estudiante y retorna la mejor opcion

Fin Para

ANEXO 34. Codigo Algoritmo ID3

```
public class Atributo
{
    #region Componentes
        private ArrayList mValues;
        private string mName;
        private object mLabel;
    #endregion

    #region Constructor
    /// <summary>
    /// Inicializa una nueva instancia de la clase Atributo
    /// </summary>
```

```

/// <param name="name">Nombre del atributo</param>
/// <param name="values">Conjunto de valores posibles para el atributo</param>
public Atributo(string name, string[] values)
{
    mName = name;
    mValues = new ArrayList(values);
    mValues.Sort();
}

/// <summary>
/// Inicializa una nueva instancia de la clase atributo
/// </summary>
/// <param name="Label">Un valor especifico en caso de ser hoja y no tener nombre ni
valores</param>
public Atributo(object Label)
{
    mLabel = Label;
    mName = string.Empty;
    mValues = null;
}

#endregion

#region Propiedades
/// <summary>
/// Retorna el nombre del atributo
/// </summary>
public string AttributeName
{
    get
    {
        return mName;
    }
}

/// <summary>
/// Retorna un array con los valores posibles del atributo
/// </summary>
public string[] values
{
    get
    {
        if (mValues != null)
            return (string[])mValues.ToArray(typeof(string));
        else
            return null;
    }
}

/// <summary>
/// Evalua si un valor es permitido para el atributo
/// </summary>
/// <param name="value">Nombre del atributo</param>
/// <returns>True si el valor es permitido, False en caso contrario</returns>
public bool isValidValue(string value)
{
    return indexValue(value) >= 0;
}

/// <summary>

```

```

/// Retorna el indice de un valor
/// </summary>
/// <param name="value">Valor a ser indexado</param>
/// <returns>Indice donde se encuentra el valor, -1 en caso contrario</returns>
public int indexValue(string value)
{
    if (mValues != null)
        return mValues.BinarySearch(value);
    else
        return -1;
}

/// <summary>
/// Convierte a string la clase
/// </summary>
/// <returns>El nombre del atributo si existe, si no la etiqueta asociada</returns>
public override string ToString()
{
    if (mName != string.Empty)
    {
        return " IF ( " + mName + " ) ES: ";
    }
    else
    {
        return mLabel.ToString();
    }
}

#endregion

```

```

public class DecisionTreeNode
{
    #region Componentes
        private ArrayList mChilds = null;
        private Atributo mAttribute;
    #endregion

    #region Constructor
        /// <summary>
        /// Inicializa una nueva instancia de DecisionTreeNode
        /// </summary>
        /// <param name="attribute">Atributo al cual el nodo esta ligado</param>
        public DecisionTreeNode(Atributo attribute)
        {
            if (attribute.values != null)
            {
                mChilds = new ArrayList(attribute.values.Length);
                for (int i = 0; i < attribute.values.Length; i++)
                    mChilds.Add(null);
            }
            else
            {
                mChilds = new ArrayList(1);
                mChilds.Add(null);
            }
            mAttribute = attribute;
        }
    }
}

```

```

#endregion

#region Propiedades
/// <summary>
/// Adiciona un DecisionTreeNode hijo al DecisionTreeNode indicado por el nombre
/// </summary>
/// <param name="DecisionTreeNode">DecisionTreeNode hijo que se adicionara</param>
/// <param name="ValueName">Nombre del DecisionTreeNode padre al que se le adicionara el
hijo</param>
public void AddDecisionTreeNode(D DecisionTreeNode treeNode, string ValueName)
{
    int index = mAttribute.indexValue(ValueName);
    mChilds[index] = treeNode;
}

/// <summary>
/// Retorna el numero de hijos
/// </summary>
public int totalChilds
{
    get
    {
        return mChilds.Count;
    }
}

/// <summary>
/// Retorna el hijo indicado por el indice
/// </summary>
/// <param name="index">Indice del hijo</param>
/// <returns>Un objeto DecisionTreeNode que representa el hijo</returns>
public DecisionTreeNode getChild(int index)
{
    return (DecisionTreeNode)mChilds[index];
}

/// <summary>
/// Retorna el Atributo al que esta ligado el padre
/// </summary>
public Atributo attribute
{
    get
    {
        return mAttribute;
    }
}

/// <summary>
/// Retorna un hijo identificado por el nombre del parametro
/// </summary>
/// <param name="branchName">Nombre del hijo</param>
/// <returns>Un objeto DecisionTreeNode que representa el hijo</returns>
public DecisionTreeNode getChildByBranchName(string branchName)
{
    int index = mAttribute.indexValue(branchName);
    return (DecisionTreeNode)mChilds[index];
}

#endregion

```

```

public class DecisionTreeID3
{
    #region Componentes
        private DataTable mSamples;
        private int mTotalPositives = 0;
        private int mTotal = 0;
        private string mTargetAttribute = "result";
        private double mEntropySet = 0.0;
    #endregion

    #region Constructor

    public DecisionTreeID3()
    {
    }

    #endregion

    #region Propiedades

    /// <summary>
    /// Retorna el total de los valores positivos en la tabla de ejemplos
    /// </summary>
    /// <param name="samples">DataTable con los ejemplos</param>
    /// <returns>Numero total de valores positivos en los ejemplos</returns>
    private int countTotalPositives(DataTable samples)
    {
        int result = 0;

        foreach (DataRow aRow in samples.Rows)
        {
            if ((bool)aRow[mTargetAttribute] == true)
                result++;
        }

        return result;
    }

    /// <summary>
    /// Calcula la entropia por medio de la siguiente formula
    ///  $-p+\log_2p+ - p-\log_2p-$ 
    ///
    /// Donde: p+ es el numero de valores positivos
    ///         p- es el numero de valores negativos
    /// </summary>
    /// <param name="positives">Cantidad de valores positivos</param>
    /// <param name="negatives">Cantidad de valores negativos</param>
    /// <returns>Retorna el valor de la Entropia</returns>
    private double calcEntropy(int positives, int negatives)
    {
        int total = positives + negatives;
        if (total == 0) return 0;
        double ratioPositive = (double)positives/total;
        double ratioNegative = (double)negatives/total;

        if (ratioPositive != 0)
            ratioPositive = -(ratioPositive) * System.Math.Log(ratioPositive, 2);
        if (ratioNegative != 0)
    }
}

```

```

        ratioNegative = - (ratioNegative) * System.Math.Log(ratioNegative, 2);

        double result = ratioPositive + ratioNegative;

        return result;
    }

    /// <summary>
    /// Recorre la tabla de ejemplos verificando si un atributo es negativo o positivo para contarlos y
    /// retornarlos por referencia
    /// </summary>
    /// <param name="samples">DataTable con los ejemplos</param>
    /// <param name="attribute">Atributo que va a revisarse</param>
    /// <param name="value">Valor permitido para el atributo</param>
    /// <param name="positives">Variable por referencia que cuenta todos los resultados
    /// positivos</param>
    /// <param name="negatives">Variable por referencia que cuenta todos los resultados
    /// negativos</param>
    private void getValuesToAttribute(DataTable samples, Atributo attribute, string value, out int positives,
    out int negatives)
    {
        positives = 0;
        negatives = 0;

        foreach (DataRow aRow in samples.Rows)
        {
            if ( ((string)aRow[attribute.AttributeName] == value) )
                if ( (bool)aRow[mTargetAttribute] == true)
                    positives++;
                else
                    negatives++;
        }
    }

    /// <summary>
    /// Calcula la ganancia de un atributo
    /// </summary>
    /// <param name="attribute">Atributo a ser calculado</param>
    /// <returns>Ganancia del atributo</returns>
    private double gain(DataTable samples, Atributo attribute)
    {
        string[] values = attribute.values;
        double sum = 0.0;

        for (int i = 0; i < values.Length; i++)
        {
            int positives, negatives;

            positives = negatives = 0;

            getValuesToAttribute(samples, attribute, values[i], out positives, out negatives);

            double entropy = calcEntropy(positives, negatives);
            sum += -(double)(positives + negatives)/mTotal * entropy;
        }
        return mEntropySet + sum;
    }

    /// <summary>
    /// Retorna el mejor atributo calculando la ganancia de cada uno de los posibles valores

```



```

/// </summary>
/// <param name="atributes">El vector con los valores del atributo</param>
/// <returns>Retorna el atributo que tiene la mayor ganacia</returns>
private Atributo getBestAttribute(DataTable samples, Atributo[] atributes)
{
    double maxGain = -1.0;
    Atributo result = null;

    foreach (Atributo attribute in atributes)
    {
        double aux = gain(samples, attribute);
        if (aux > maxGain)
        {
            maxGain = aux;
            result = attribute;
        }
    }
    return result;
}

/// <summary>
/// Retorna true si todos los ejemplos de la muestra son positivos
/// </summary>
/// <param name="samples">DataTable con los ejemplos</param>
/// <param name="targetAttribute">Atributo o columna que se desea calcular</param>
/// <returns>True si todos los ejemplos son positivos, si no False</returns>
private bool allSamplesPositives(DataTable samples, string targetAttribute)
{
    foreach (DataRow row in samples.Rows)
    {
        if ( (bool)row[targetAttribute] == false)
            return false;
    }

    return true;
}

/// <summary>
/// Retorna true si todos los ejemplos de la muestra son negativos
/// </summary>
/// <param name="samples">DataTable con los ejemplos</param>
/// <param name="targetAttribute">Atributo o columna que se desea calcular</param>
/// <returns>True si todos los ejemplos son negativos, si no False</returns>
private bool allSamplesNegatives(DataTable samples, string targetAttribute)
{
    foreach (DataRow row in samples.Rows)
    {
        if ( (bool)row[targetAttribute] == true)
            return false;
    }

    return true;
}

/// <summary>
/// Retorna un arreglo con todos los valores distintos de un atributo en la tabla de ejemplos
/// </summary>
/// <param name="samples">DataTable con los ejemplos</param>
/// <param name="targetAttribute">Atributo o columna que se desea calcular</param>
/// <returns>Un ArrayList con los valores distintos</returns>

```

```

private ArrayList getDistinctValues(DataTable samples, string targetAttribute)
{
    ArrayList distinctValues = new ArrayList(samples.Rows.Count);

    foreach(DataRow row in samples.Rows)
    {
        if (distinctValues.IndexOf(row[targetAttribute]) == -1)
            distinctValues.Add(row[targetAttribute]);
    }

    return distinctValues;
}

/// <summary>
/// Retorna el valor mas comun dentro de una tabla de ejemplos
/// </summary>
/// <param name="samples">DataTable con los ejemplos</param>
/// <param name="targetAttribute">Atributo o columna que se desea calcular</param>
/// <returns>Retorna un objeto atributo mas comun dentro de la tabla de ejemplos</returns>
private object getMostCommonValue(DataTable samples, string targetAttribute)
{
    ArrayList distinctValues = getDistinctValues(samples, targetAttribute);
    int[] count = new int[distinctValues.Count];

    foreach(DataRow row in samples.Rows)
    {
        int index = distinctValues.IndexOf(row[targetAttribute]);
        count[index]++;
    }

    int MaxIndex = 0;
    int MaxCount = 0;

    for (int i = 0; i < count.Length; i++)
    {
        if (count[i] > MaxCount)
        {
            MaxCount = count[i];
            MaxIndex = i;
        }
    }
    if (count.Length == 0) return "NHSI";
    return distinctValues[MaxIndex];
}

/// <summary>
/// Crea un Arbol de Decision basado en los ejemplos de la DataTable
/// </summary>
/// <param name="samples">Tabla de ejemplos</param>
/// <param name="targetAttribute">Atributo o columna que se desea calcular</param>
/// <returns>La raiz del arbol de decision para ser mostrada</returns></returns?>
private DecisionTreeNode internalMountTree(DataTable samples, string targetAttribute, Atributo[]
attributes)
{
    if (allSamplesPositives(samples, targetAttribute) == true)
        return new DecisionTreeNode(new Atributo(true));

    if (allSamplesNegatives(samples, targetAttribute) == true)
        return new DecisionTreeNode(new Atributo(false));
}

```

```

        if (attributes.Length == 0)
            return new DecisionTreeNode(new Atributo(getMostCommonValue(samples,
targetAttribute)));

        mTotal = samples.Rows.Count;
        mTargetAttribute = targetAttribute;
        mTotalPositives = countTotalPositives(samples);

        mEntropySet = calcEntropy(mTotalPositives, mTotal - mTotalPositives);

        Atributo bestAttribute = getBestAttribute(samples, attributes);

        DecisionTreeNode root = new DecisionTreeNode(bestAttribute);

        DataTable aSample = samples.Clone();

        foreach(string value in bestAttribute.values)
        {
            // Selecciona todas los elementos que tienen el valor escogido del atributo
            aSample.Rows.Clear();
            DataRow[] rows = samples.Select(bestAttribute.AttributeName + " = " + "\"" + value +
            "");

            foreach(DataRow row in rows)
            {
                aSample.Rows.Add(row.ItemArray);
            }

            // Crea una nueva lista de atributos quitando el atributo elegido que es el mejor
            atributo
            ArrayList aAttributes = new ArrayList(attributes.Length - 1);
            for(int i = 0; i < attributes.Length; i++)
            {
                if (attributes[i].AttributeName != bestAttribute.AttributeName)
                    aAttributes.Add(attributes[i]);
            }

            //Evalua si aun hay ejemplos para jerarquizar, si hay recursivamente crea los
            nuevos nodos hijos
            // si no devuelve el valor mas comun
            if (aSample.Rows.Count == 0)
            {
                return new DecisionTreeNode(new
                Atributo(getMostCommonValue(aSample, targetAttribute)));
            }
            else
            {
                DecisionTreeID3 dc3 = new DecisionTreeID3();
                DecisionTreeNode ChildNode = dc3.mountTree(aSample, targetAttribute,
                (Atributo[])aAttributes.ToArray(typeof(Atributo)));
                root.AddDecisionTreeNode(ChildNode, value);
            }
        }

        return root;
    }

    /// <summary>
    /// Crea el arbol de decision basado en los ejemplos
    /// </summary>

```

```

/// <param name="samples">DataTable con los ejemplos</param>
/// <param name="targetAttribute">Atributo o columna que se desea calcular</param>
/// <returns>La raiz del arbol creado</returns></returns?>
public DecisionTreeNode mountTree(DataTable samples, string targetAttribute, Atributo[] attributes)
{
    mSamples = samples;
    return internalMountTree(mSamples, targetAttribute, attributes);
}
#endregion

```

ANEXO 35. Código Algoritmo Deliberar

/// Calcula la cantidad de rutas positivas dentro del Arbol de decisión por medio de la recursión

```

public void NumeroRutasPositivas(DecisionTreeNode raiz, ref int Numero)
{
    if (raiz.attribute.ToString() == "True")
        Numero = Numero+1;
    if (raiz.attribute.values != null)
    {
        for (int i=0; i<raiz.attribute.values.Length; i++)
        {
            DecisionTreeNode childNode = raiz.getChildByBranchName(raiz.attribute.values[i]);
            NumeroRutasPositivas(childNode, ref Numero);
        }
    }
}

```

/// Encuentra los valores de los atributos que conllevan una ruta positiva

```

public void ValoresRutasPositivas(DecisionTreeNode raiz, ref ArrayList Valores, ref ArrayList RutasPositivas)
{
    if (raiz.attribute.values != null)
    {
        for (int i=0; i<raiz.attribute.values.Length; i++)
        {
            DecisionTreeNode childNode = raiz.getChildByBranchName(raiz.attribute.values[i]);
            Valores.Add(raiz.attribute.values[i]);
            ValoresRutasPositivas(childNode, ref Valores, ref RutasPositivas);
        }
        if (Valores.Count>=1)
            Valores.RemoveAt(Valores.Count-1);
    }
    else
    {
        if (raiz.attribute.ToString() == "True")
        {
            ArrayList Ruta = new ArrayList();
            Ruta = (ArrayList)Valores.Clone();
            RutasPositivas.Add(Ruta);
            Valores.RemoveAt(Valores.Count-1);
        }
        else
            Valores.RemoveAt(Valores.Count-1);
    }
}

```

/// Determina si un valor de atributo se encuentra en un conjunto de datos

```

public bool Esta(string nombre, DataSet dsConjuntoDatos)
{
    foreach (DataRow fila in dsConjuntoDatos.Tables[0].Rows)
    {
        if (nombre == fila[0].ToString())
            return true;
    }
    return false;
}

/// Separa los atributos Sistema de los atributos Estudiante
public void SeparacionAtributos(ArrayList Atributos, out ArrayList AtrSistema, out ArrayList AtrEstudiante)
{
    AtrSistema = new ArrayList();
    AtrEstudiante = new ArrayList();
    DataSet dsSistemaAtr = new DataSet();
    dsSistemaAtr = fachada.Inteligencia_AtributosSistema();
    foreach (int IdAtr in Atributos)
    {
        if (Esta(IdAtr.ToString(), dsSistemaAtr))
            AtrSistema.Add(IdAtr);
        else
            AtrEstudiante.Add(IdAtr);
    }
}

/// Encuentra la experiencia en la cual la mayor cantidad de atributos del estudiante coinciden
public ArrayList EncontrarMasParecido (ArrayList Estudiante, ArrayList ExpPositivas)
{
    int valorActual =0;
    int nuevoValor=0;
    int indice=0;
    ArrayList RespuestaActual = new ArrayList();
    ArrayList RespuestaTotal = new ArrayList();
    for (int i=0; i<ExpPositivas.Count; i++)
    {
        nuevoValor = BuscarValoracion(Estudiante, (ArrayList)ExpPositivas[i], out RespuestaActual);
        if (nuevoValor > valorActual)
        {
            valorActual = nuevoValor;
            indice =i;
            RespuestaTotal = (ArrayList)RespuestaActual.Clone();
        }
    }
    return RespuestaTotal;
}

/// Valora la experiencia del estudiante para elegir la mas aproximada
public int BuscarValoracion (ArrayList Estudiante, ArrayList RutaPositiva, out ArrayList SalidaTotal)
{
    int valoracion =0;
    SalidaTotal = (ArrayList)RutaPositiva.Clone();
    foreach (object atributo in Estudiante.ToArray())
    {
        foreach (object valor in RutaPositiva.ToArray())
        {
            if (atributo.ToString() == valor.ToString())
            {
                SalidaTotal.Remove(valor);
            }
        }
    }
}

```

```

        valoracion ++;
    }
}
return valoracion;
}

/// Busca un atributo en la lista de atributos
public bool Comparar (string IdAtr, ArrayList IdsAtr)
{
    foreach (object ID in IdsAtr.ToArray())
    {
        if (ID.ToString() == IdAtr)
            return true;
    }
    return false;
}

/// Algoritmo principal que maneja la deliberacion
public ArrayList Deliberar(string IDPerfil, DecisionTreeNode raiz)
{
    // Calculamos la cantidad de rutas positivas
    int RutasPositivas = 0;
    NumeroRutasPositivas(raiz, ref RutasPositivas);

    //Ubicamos las rutas positivas
    ArrayList alRuta = new ArrayList();
    ArrayList alTodasRutasPositivas = new ArrayList();
    ValoresRutasPositivas(raiz, ref alRuta, ref alTodasRutasPositivas);

    //Separamos los atributos Sistema y los atributos Estudiante en la experiencia del
    estudiante
    DataSet dsAtributosEstudiante = new DataSet();
    dsAtributosEstudiante = fachada.Inteligencia_ValoresAtributosUsuarioID(IDPerfil);
    ArrayList alAtrEst = new ArrayList();
    foreach (DataRow fila in dsAtributosEstudiante.Tables[0].Rows)
    {
        alAtrEst.Add(fila[3]);
    }
    ArrayList alSistema;
    ArrayList alEstudiante;
    SeparacionAtributos(alAtrEst,out alSistema,out alEstudiante);

    //Obtenemos los valores para compararlos
    ArrayList alValoresEstudiante = new ArrayList();
    ArrayList alValoresSistema = new ArrayList();
    foreach (DataRow fila in dsAtributosEstudiante.Tables[0].Rows)
    {
        if (Comparar(fila[3].ToString(),alEstudiante))
            alValoresEstudiante.Add(fila[1].ToString());
        else
            alValoresSistema.Add(fila[1].ToString());
    }
    return EncontrarMasParecido(alValoresEstudiante, alTodasRutasPositivas);
}

```

ANEXO 36. Pantallazos Modulo Inteligente

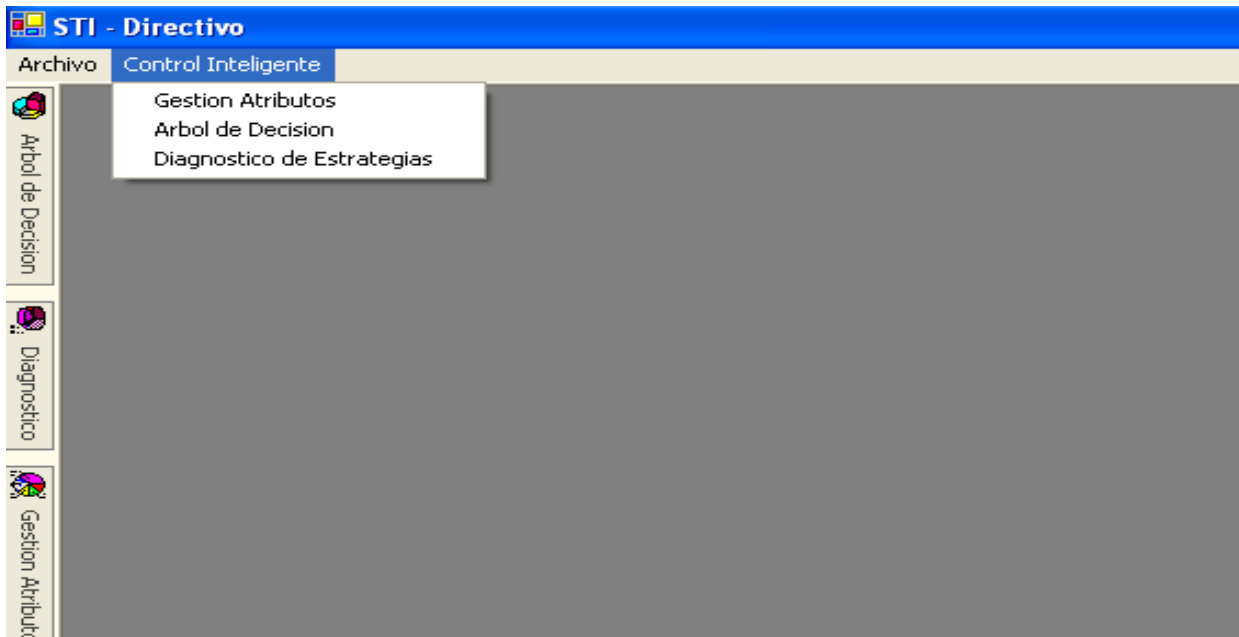


Figura 35. Pantalla Inicial con las Opciones

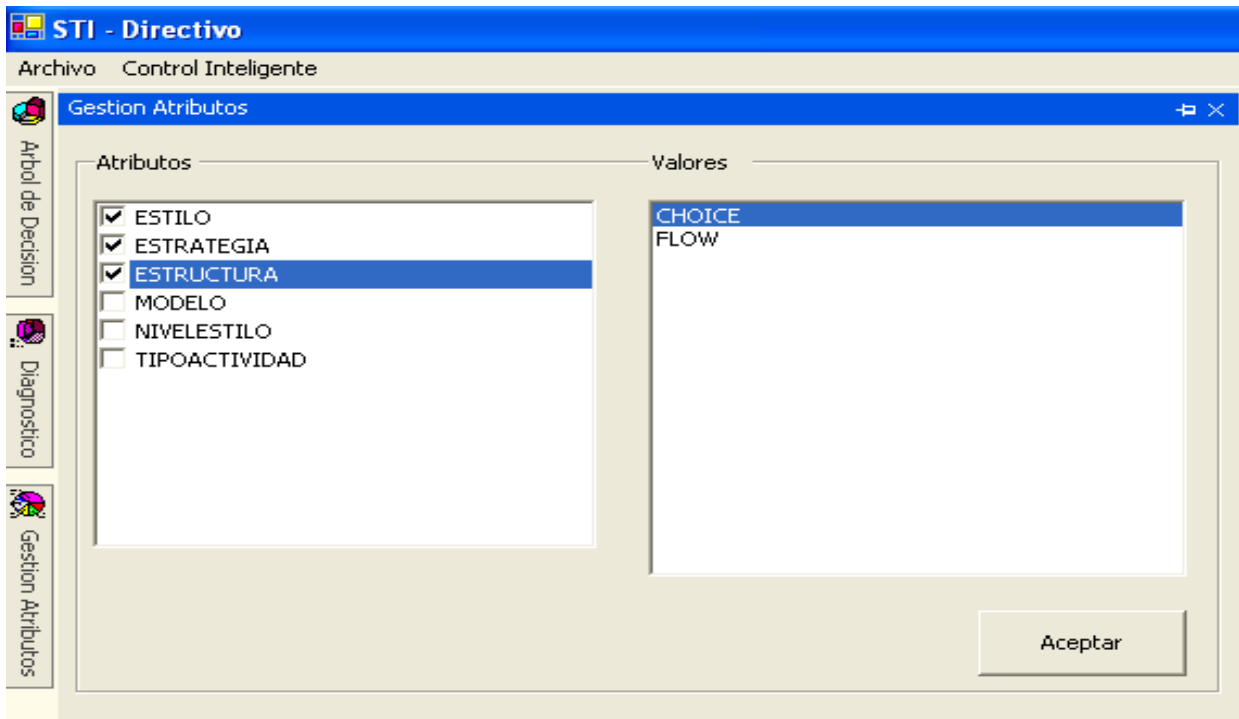


Figura 36. Gestión de Atributos

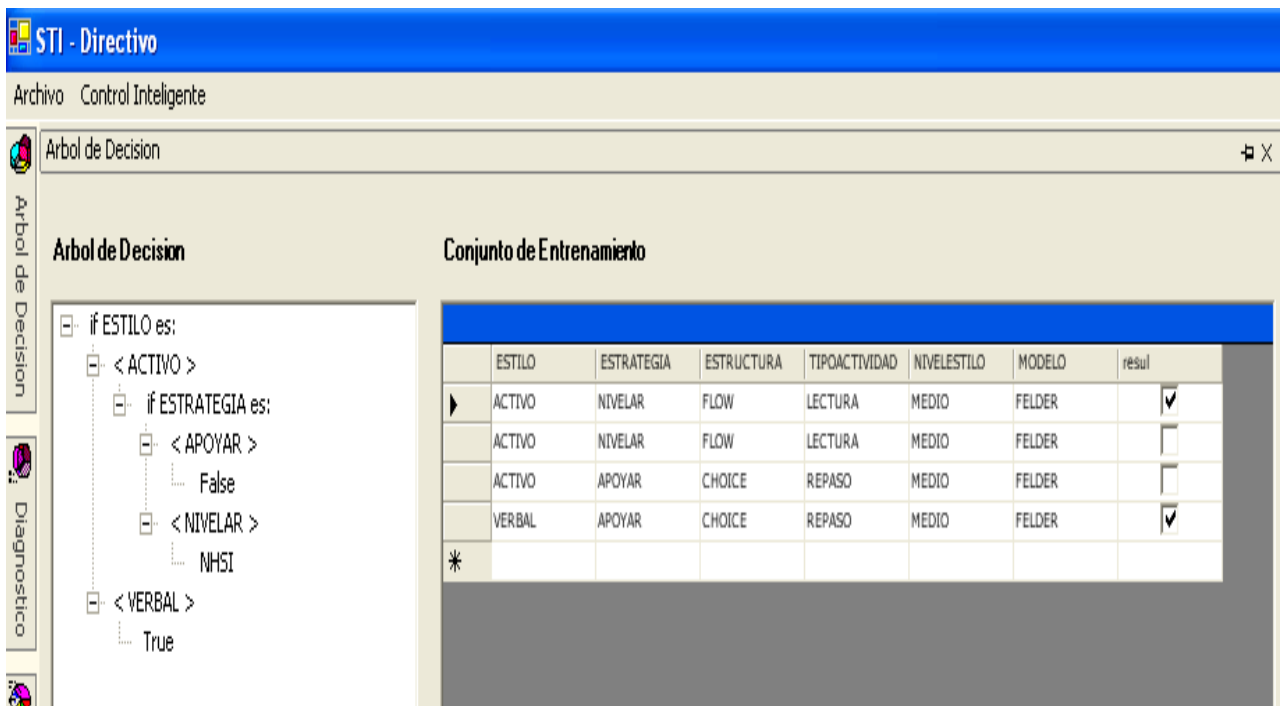


Figura 37. Árbol de decisión generado y su conjunto de experiencias – Caso 1

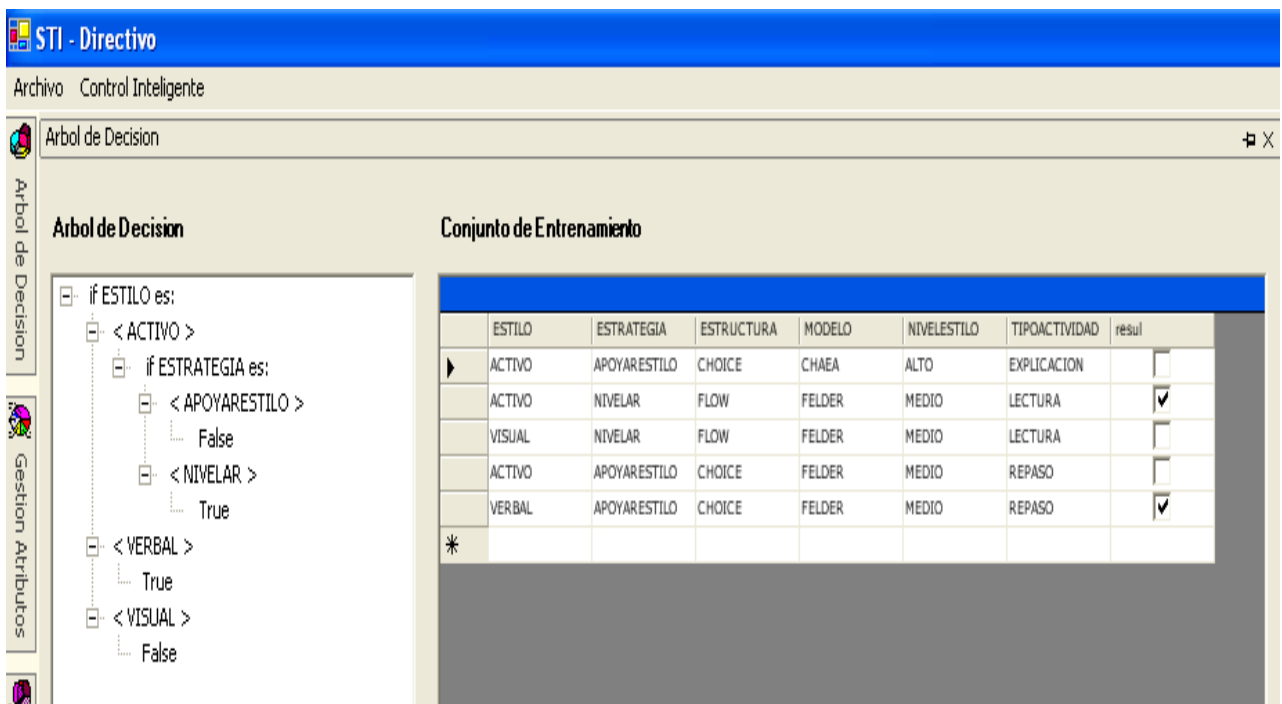


Figura 38. Árbol de decisión generado y su conjunto de experiencias – Caso 2

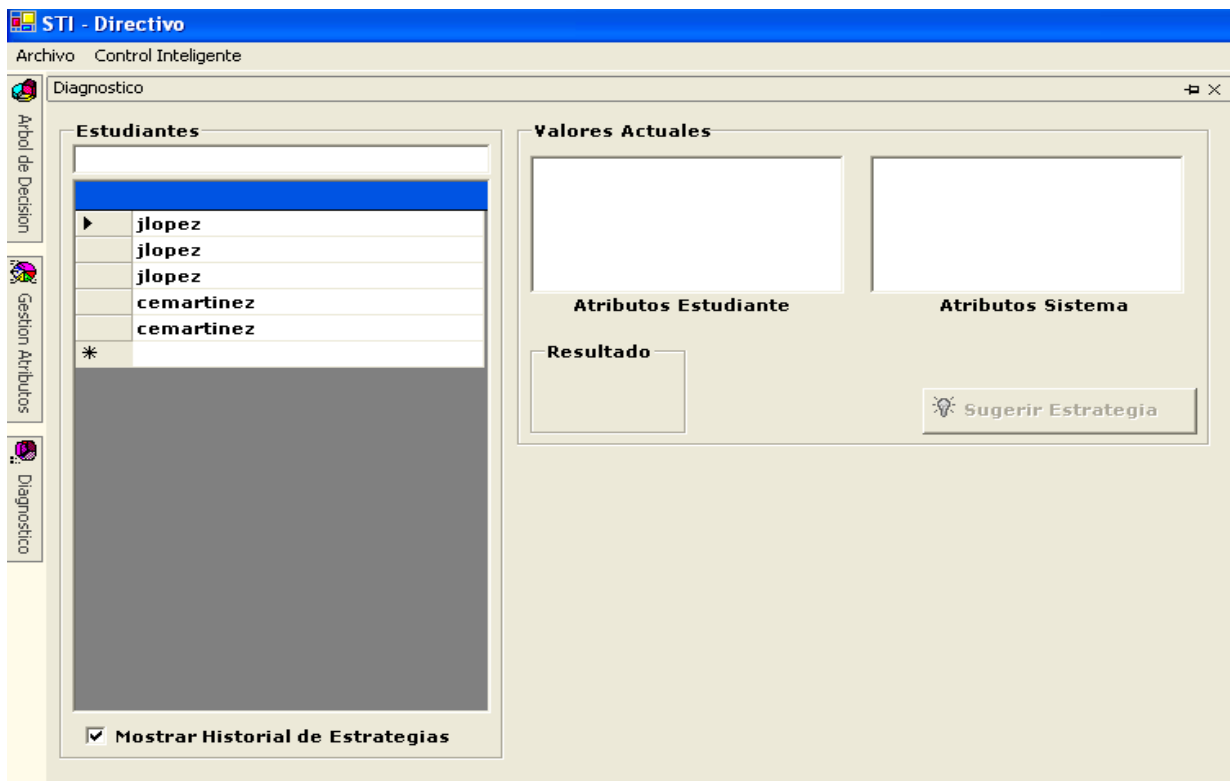


Figura 39. Pantalla de diagnostico

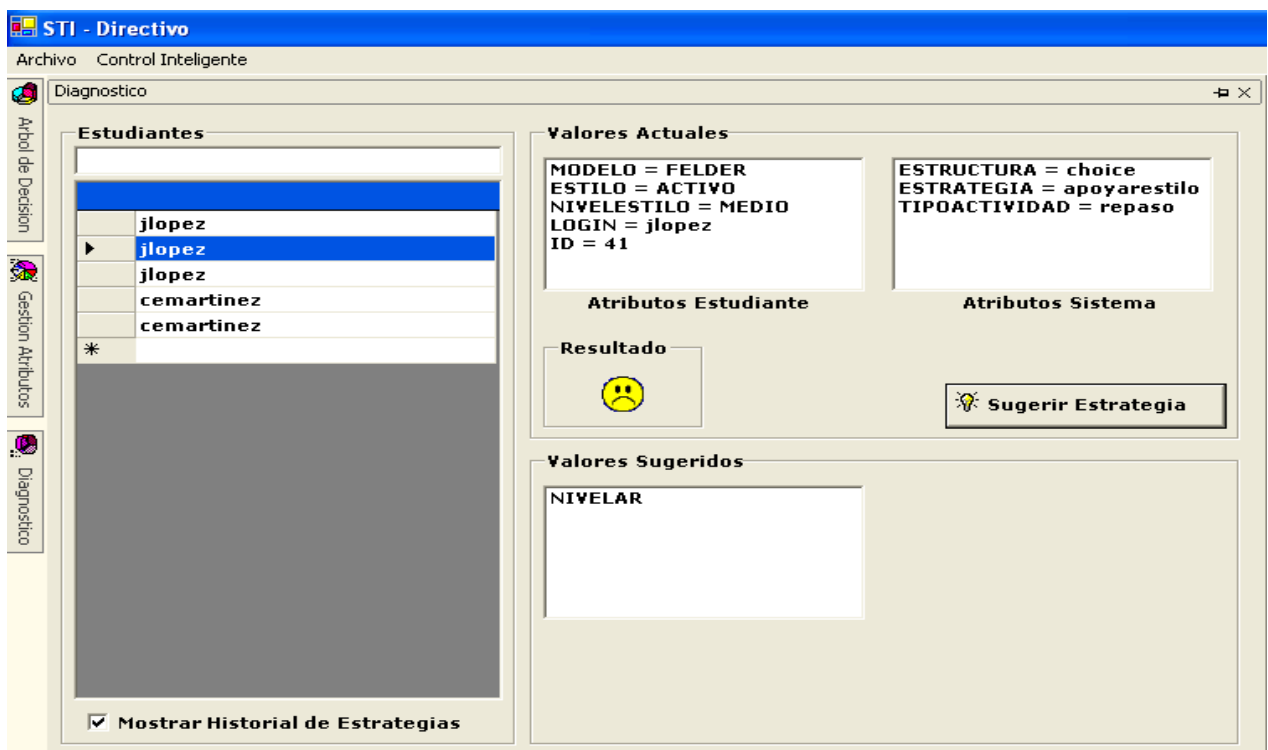


Figura 40. Pantalla de generación de resultados

ANEXO 37. Glosario

Pedagogía: Que pretende educar, enseñar o instruir en un campo determinado: visión pedagógica; material pedagógico; intención pedagógica.

Modelo Pedagógico: Finalidad o sentido que se da a la enseñanza o transmisión de conocimiento a través de un curso presencial o a distancia (en línea). En un modelo pedagógico se establecen los lineamientos sobre cuya base se derivan posteriormente los propósitos y los objetivos.

Currículo: Este concepto ha evolucionado desde el simple listado de asignaturas de un programa de estudio hasta considerarlo como la totalidad de experiencias formativas por las cuales pasa un individuo a lo largo de su vida. En algunos conceptos, el currículo se circunscribe a la institución educativa, mientras en otros más amplios, se entiende por currículo cualquier experiencia que contribuya a la formación de un individuo dentro y fuera de dicha institución.

Aprendizaje: Es el Proceso a través del cual se adquieren habilidades, destrezas y conocimientos en sus distintas dimensiones: conceptos, procedimientos, actitudes y valores. Como resultado de la experiencia, la instrucción o la observación.

Estilos de Aprendizaje: Son las diversas maneras que la persona tiene para aprender, que le son propios y hay que tener en cuenta para cualquier instrucción o enseñanza. Es de la manera en que los alumnos adquieren nuevos conocimientos, aprenda sin dificultades y de una manera mas rápida y confiable

TIC's: Tecnologías de La información y las Comunicaciones. Medios o recursos de las redes de comunicación electrónica, Como el correo electrónico, simulaciones en ambientes multiusuarios, grupos de discusión, videoconferencias, foros, chats, entre otros.

Abstracción: La abstracción es la capacidad mental superior que tiene todo ser humano para poder deducir la esencia de un concepto o situación determinada, Sin tener al objeto de manera tangible.

Pensamiento Sistémico: Pensamiento que permite percibir el conocimiento como toda una red de articulaciones y no como una serie de puntos acumulados aisladamente. A través de este pensamiento el estudiante aprende a no caer en la falacia de desarticular las cosas, sino de ver la realidad como un sistema de causas y consecuencias.

Experimentación: Se refiere al hábito y al método para intentar nuevas técnicas y nuevos materiales, para probar, para examinar alternativas distintas a las establecidas por la tradición. Tiene que ver con la exploración de territorios desconocidos y de probar diferentes rutas para la salida.

Colaboración y Cooperación: Se refiere a trabajar en equipo, a comunicar conceptos entre teorías, desarrollar proyectos a través del consenso del grupo. Articular, re-enunciar y clasificar conceptos de manera que contribuyan al progreso y desempeño de todo el grupo envuelto en la tarea.

Autoevaluación: Revisión detallada y periódica del propio responsable de las acciones emprendidas para mejorar el Funcionamiento de determinada área, unidad, órgano, sistema o procedimiento, a fin de medir el grado de eficiencia, eficacia y congruencia en su operación.

Coevaluación: Proceso que tiene como finalidad determinar el grado de eficacia y eficiencia, a través de las actividades grupales o que involucren a un tercer actor a parte del maestro y el estudiante protagonista del proceso de aprendizaje.

Heteroevaluación: Proceso para determinar que tan bien han sido empleados los recursos destinados a alcanzar los objetivos previstos, a través de diferentes métodos, evitando las desviaciones y adoptando medidas correctivas que garanticen el cumplimiento adecuado de las metas presupuestadas, este podría enfocarse de acuerdo al estilo de aprendizaje del estudiante.

Módulo: Elemento adoptado como unidad de medida para determinar las proporciones entre las diferentes partes de un Curso, son los temas que conforman los contenidos de dicho curso. Cantidad menor proporcionada a un todo.

Tutor: Persona encargada de cuidar a un menor de edad, y de administrar sus bienes. Defensor, protector. En cuanto a sistemas tutores inteligentes se refiere no es que se reemplace al profesor, sino que se implementa un modulo inteligente que ayude a dicho profesor e incluso que en su ausencia pueda tomar decisiones elementales frente al estudiante.

Conocimiento: El Conocimiento se considera como un artículo que puede ser transferido entre personas y sistemas en vez de una propiedad inherente como lo es la inteligencia. El conocimiento es la Información acerca del mundo la cual permite tomar decisiones. Es cuanto a sistemas tutores se refiere es el modulo encargado de dominio del problema, ,es decir, el que contiene la información de las estructuras y los contenidos del curso.

Interfaz: Zona de contacto, conexión entre dos componentes de "hardware", entre dos aplicaciones o entre un usuario y una aplicación. Para el caso de Sistemas

Tutores Inteligentes es el modulo compuesto por un sistema multiagente que monitorea esta interacción y la guarda en el modulo del estudiante.

Inteligencia Artificial (IA): Rama de la Informática que trata de construir programas que emulen métodos de razonamiento análogos a los de los humanos.

Modelo: Proyecto o reproducción a escala reducida o no de una pieza escultórica m. Objeto que se reproduce. Ejemplar que uno se propone en la realización de una obra

Conexionista: Dar enlace a unas cosas con otras. Establecer una conexión entre varios elementos ya sean por ejemplo de electricidad como encajar las dos piezas del enchufe; o del conocimiento como conectar los nuevos conceptos con los conocimientos ya adquiridos y aumentar el conocimiento.

Simbólico: Representación de un concepto a través de letras o símbolos, habilidad que permite a las personas captar de manera más rápida y eficiente los nuevos conceptos e incorporarlos a su estructura mental.

UML: (Lenguaje Unificado para la Construcción de Modelos) que se define como un “lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas software...”[48]. Es un sistema notacional (que entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos.

Estándares: Son acuerdos documentados que contienen especificaciones técnicas u otros criterios específicos para ser usados como referentes, guías o definiciones de características, para asegurar que materiales, productos, procesos y servicios son obtenidos o han sido realizados de acuerdo a sus propósitos.

Arquitectura: Una arquitectura es una categoría programática de la infraestructura física y lógica que puede tener un sistema de información ya sea orientado a objetos o definido con otro tipo de paradigma.

Modelo Cliente / Servidor : Es una arquitectura que divide y especializa programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realiza se efectúe con la mayor eficiencia, y permita simplificar las actualizaciones y mantenimiento del sistema. Sus ventajas son que el servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes, además se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre[49].

Estrategias de Aprendizaje: Proceso de toma de decisiones, consciente e intencional, en el que el estudiante elige y recupera los conocimientos que necesita para hacer su trabajo.[8]