

**HERRAMIENTA SOFTWARE PARA EL ESTUDIO DE FENOMENOS
AMBIENTALES, MEDIANTE EL MODELADO Y LA SIMULACION CON
DINAMICA DE SISTEMAS**

**OSCAR ENRIQUE SANTAMARIA PABON
ANDRES EDUARDO MENDEZ FORERO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS FISICOMECAICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA**

2004

**HERRAMIENTA SOFTWARE PARA EL ESTUDIO DE FENOMENOS
AMBIENTALES, MEDIANTE EL MODELADO Y LA SIMULACION CON
DINAMICA DE SISTEMAS**

**OSCAR ENRIQUE SANTAMARIA PABON
ANDRES EDUARDO MENDEZ FORERO**

**Proyecto de Grado para optar al título de
Ingeniero de Sistemas**

**Director
HUGO ANDRADE SOSA
Ingeniero de sistemas, M.Sc.**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2004

Dedicatorias

A Dios,

A nuestras familias,

A todas las personas que nos apoyaron y siempre confiaron en nuestras capacidades.

AGRADECIMIENTOS

Emiliano Lince

Ingeniero de Sistemas, Universidad Industrial de Santander. Grupo SIMON.

Cristina Durango Mendoza

Estudiante X Nivel Ingeniería de Sistemas Universidad Industrial de Santander

Oscar Iván Morales Duarte

Ingeniero de Sistemas, Universidad Industrial de Santander.

Jorge Alberto Ortiz Martínez

Estudiante V Nivel Ingeniería Electrónica Universidad Industrial de Santander.

Estudiantes de Sistemas Dinámicos 2 del primer y segundo semestre del año 2003

Estudiantes de la especialización en ingeniería ambiental del primer segundo semestre del año 2003

CONTENIDO

INTRODUCCION

1. PRESENTACION	6
2. PRELIMINARES	9
2.1 ESPECIFICACIONES DEL PROYECTO	9
2.1.1 Título.	9
2.1.2 Director de proyecto.	9
2.1.3 Codirector de proyecto.	9
2.1.4 Autores.	9
2.1.5 Entidades interesadas.	9
2.1.6 Objetivo General.	11
2.1.7 Objetivos específicos.	11
2.1.8 Justificación y Definición del Problema.	12
3. MARCO TEORICO Y ANTECEDENTES	16
3.1 EL ENFOQUE SISTEMICO Y EL MEDIO AMBIENTE	17
3.1.1 Aproximación al enfoque sistémico.	17
3.1.2 Desarrollo histórico de la teoría general de sistemas.	18
3.1.3 El medio ambiente como sistema.	23
3.1.4 Gestión Ambiental y Modelos Ambientales.	30
3.1.5 Pensando en el ambiente.	32

3.1.6	Algunos antecedentes de modelos ambientales utilizando la dinámica de sistemas.	33
3.2	MODELOS FORMALES Y REALIDAD VIRTUAL	38
3.2.1	Procesos de aprendizaje.	39
3.3	INFORMATICA Y LA CONSTRUCCION DE LA REALIDAD VIRTUAL	43
3.3.1	Modelamiento Matemático.	43
3.3.2	Modelamiento con Dinámica de Sistemas.	44
3.3.3	Software de Simulación con Dinámica de Sistemas.	46
3.4	DINAMICA DE SISTEMAS: FORMA DE PENSAMIENTO PARA LA REPRESENTACIÓN Y EL APRENDIZAJE	47
3.5	APRENDIZAJE DINAMICO-SITEMICO	49
3.5.1	Modelamiento.	49
3.5.2	Interacción simulada y mundos virtuales.	52
3.5.3	El doble ciclo de aprendizaje: paradigma y lenguaje.	53
3.6	EL LENGUAJE DINAMICO-SISTEMICO	53
3.6.1	Sistema de Lenguajes.	53
3.6.2	Prototipos de Complejidad creciente.	57
3.7	FORMAS METODOLOGICAS DE LA DINAMICA DE SISTEMAS	58
3.7.1	Metodología para la intervención.	59
3.7.2	Metodología para el aprendizaje.	59

3.8	PLATAFORMAS DE DINAMICA DE SISTEMAS PARA EL APRENDIZAJE	60
3.8.1	Micromundos.	60
3.8.2	Plataformas de Dinámica de sistemas.	62
3.9	INFORMATICA EDUCATIVA	63
3.9.1	Efectos de la Informática educativa.	63
3.9.2	Enseñanza Asistida por Ordenador (E.A.O.).	65
3.9.3	Herramientas de Autor.	66
3.9.4	Guías de Autoaprendizaje Apoyadas en Dinámica de Sistemas.	68
3.10	ALGUNOS CONCEPTOS DE DESARROLLO Y CONSTRUCCION DE SOFTWARE	68
3.10.1	El proceso unificado de desarrollo de software.	69
3.10.2	Construcción de software orientado a objetos.	71
3.10.3	Programación orientada a objetos.	71
3.10.4	Reutilización de software.	74
3.10.5	UML.	75
4.	DESARROLLO DEL PROYECTO	77
4.1	ACOPIO DE INFORMACION	79
4.2	TRABAJO CON ESTUDIANTES Y ENTIDADES INTERESADAS	82
4.3	MODELADO CON DINAMICA DE SISTEMAS	83
4.3.1	Planteamiento del problema.	85

4.3.2	Objetivo del modelo.	86
4.3.3	Modelado de la situación por prototipos.	87
4.3.4	Aplicación de la metodología a los demás modelos ambientales.	113
4.4	DESARROLLO DE LA HERRAMIENTA SOFTWARE	115
4.4.1	¿Qué es GAIA?	116
4.4.2	Tecnologías utilizadas para el desarrollo del software.	119
4.4.3	El proceso de desarrollo de GAIA.	119
5.	CONCLUSIONES	191
6.	RECOMENDACIONES	194
7.	BIBLIOGRAFIA	195
	ANEXOS	
	ANEXO A. MANUAL DEL USUARIO	197
	ANEXO A. MANUAL DEL USUARIO	198
	ANEXO B. GUIA DEL PROGRAMADOR	212

LISTA DE TABLAS

Tabla 1. Conocimientos previos.	7
Tabla 2. Simbología utilizada en Diagramas de Flujos-Niveles.	46
Tabla 3. Información encontrada dentro del software.	81
Tabla 4. Tipos de modelos según su origen.	114
Tabla 5. Glosario de términos fase de inicio.	132
Tabla 6. Artefactos y su evolución durante el desarrollo de GAIA. C – comenzar; R – refinar.	133
Tabla 7. Personal encargado de las pruebas exhaustivas.	183
Tabla 8. Pruebas a agregar articulo.	183
Tabla 9. Resultados de la prueba de editar artículos.	184
Tabla 10. Pruebas a mostrar artículo.	185
Tabla 11. Pruebas buscar artículo.	186
Tabla 12. Resultados de la prueba de buscar dentro de articulo.	186
Tabla 13. Resultados de la prueba al laboratorio de modelado.	187
Tabla 14. Pruebas a centros de navegación.	188
Tabla 15. Resultados de la prueba de guardar y cargar artículos en disco duro.	188
Tabla 16. Pruebas a glosario.	189
Tabla 17. Pruebas al caso de uso de manejo de usuarios.	189
Tabla 18. Pruebas al caso de uso temas relacionados.	190

LISTA DE FIGURAS

Figura 1. El Iceberg	13
Figura 2. Triangulo de Kanisza	32
Figura 3. Ciclos de aprendizaje "natural"	40
Figura 4. Ciclos de aprendizaje "artificial"	42
Figura 5. Aprendizaje y Modelamiento participativo	43
Figura 6. Modelado con Dinámica de sistemas, usando la perspectiva del pensamiento sistémico para observar la realidad	50
Figura 7. Los flujos de trabajo, tienen lugar en las cuatro fases expuestas en el proceso unificado	70
Figura 8. Esquema del proceso llevado a cabo para elaborar la herramienta software GAIA	77
Figura 9. Estructura general de un Sistema - Modelo	85
Figura 10. Diagrama causal del primer prototipo	88
Figura 11. Diagrama de Flujos y Niveles del primer prototipo	90
Figura 12. Comportamiento de referencia (simulación mental) del primer prototipo	91
Figura 13. Definición de la variable exógena <i>Tabla_DDT_aplica</i>	92
Figura 14. Resultados de la simulación del primer prototipo.	95
Figura 15. Representación en Flujos y Niveles del segundo prototipo	96
Figura 16. Resultados de las simulación del segundo prototipo.	98

Figura 17. Diagrama de Flujos y Niveles del tercer prototipo.	99
Figura 18. Comportamiento simulado del tercer prototipo	101
Figura 19. Comportamiento simulado del cuarto prototipo	102
Figura 20. Resultados de la simulación del cuarto prototipo.	104
Figura 21. Diagrama de flujo-Nivel del prototipo quinto	106
Figura 22. Definición de la variable exogena que representa la aplicación histórica de DDT desde el año 1943 a nuestros días. Los datos estan expresados en miles de toneladas	107
Figura 23. Simulación del quinto prototipo. DDT aplicado y cantidad de DDT en el suelo.	111
Figura 24. DDT en los peces.	112
Figura 25. Esquema de las tecnologías y conceptos que utilizan las Guías de autoaprendizaje apoyadas en dinámica de sistemas	117
Figura 26. Tecnologías y herramientas utilizadas para desarrollar GAIA.	119
Figura 27. Principales casos de uso (críticos).	121
Figura 28. Diagrama de actividades del caso de uso Iniciar sesión.	123
Figura 29. Diagrama de actividades del caso de uso Modelar.	124
Figura 30. Diagrama de actividades del caso de uso Experimentar.	125
Figura 31. Diagrama de actividades del caso de uso Buscar.	126
Figura 32. Diagrama de actividades del caso de uso usar tutoriales.	127
Figura 33. Diagrama de actividades del caso de uso usar ayuda.	128
Figura 34. Diagrama de actividades del caso de uso agregar articulo.	129

Figura 35. Diagrama de actividades del caso de uso navegar.	131
Figura 36. Modelo del Dominio Inicial.	132
Figura 37. Algunos prototipos iniciales de la interfaz de usuario.	133
Figura 38. Modelo del Dominio Refinado.	136
Figura 39. DSS para el caso de uso agregar articulo.	137
Figura 40. Modelo entidad – relación de la base de datos.	138
Figura 41. Algunas clases del dominio modificadas que intervienen en el caso de uso agregar artículo y que aparecen en el modelo de clases de software.	139
Figura 42. Relación entre clases del dominio, clases software y las clases base para definir las	140
Figura 43. Relación entre clases del dominio, clases software y las clases base para definir las	141
Figura 44. Diagrama de la clase TSolicitud, esta clase es una de las mas importantes dentro del software pues es la que permite la interacción y todas las operaciones con la base de datos	143
Figura 45. Detalle miembros diagrama de clases software	143
Figura 46. DSS para el caso de uso editar articulo.	144
Figura 47. Interfaz de usuario para editar articulo.	145
Figura 48. Incrementos sucesivos en la herramienta software para cubrir el caso de uso agregar articulo.	145
Figura 49. Jerarquía del árbol creado para poder guardar artículos.	146

Figura 50. Pantalla que sirve como interfaz de usuario para agregar un artículo.	147
Figura 51. DSS para el caso de uso mostrar articulo.	148
Figura 52. Clases software que intervienen en el caso de uso de mostrar articulo y su relación con la interfaz de usuario y la Base de Datos.	149
Figura 53. Especificación del diagrama de las clases que intervienen en el caso de uso mostrar.	149
Figura 54. Descripción de las clases del software.	150
Figura 55. Tabla Objeto Multimedia.	151
Figura 56. Relación entre los contenedores las clases del software y los tipos de archivo	152
Figura 57. Mini – Iteraciones dentro de la segunda iteración de la fase de elaboración	153
Figura 58. Interfaz de usuario para mostrar articulo.	153
Figura 59. DSS para el caso de uso modelar.	154
Figura 60. DSS para el caso de uso experimentar.	154
Figura 61. DSS para el caso de uso “Usar Laboratorio”.	155
Figura 62. Diagrama de clases del software y su relación con la interfaz de usuario.	156
Figura 63. Pequeñas iteraciones incluidas dentro de la tercera iteración de la fase de elaboración.	157
Figura 64. Pantalla de la interfaz del laboratorio de modelado.	157

Figura 65. Diagrama de Secuencia de estados para la búsqueda.	158
Figura 66. Clases que intervienen en la búsqueda y su relación con la interfaz de usuario.	158
Figura 67. Pantalla de la interfaz de usuario para búsqueda avanzada.	159
Figura 68. Panel izquierdo del entorno de navegación principal utilizado para ingresar las palabras.	160
Figura 69. Mini – Iteraciones en la cuarta Iteración de la fase de elaboración.	161
Figura 70. DSS para el caso de uso de agregar usuarios, observe que en realidad se amplio el caso de uso a manipulación de usuarios pues cobija también editar y eliminar usuarios.	162
Figura 71. Relación de las clases del software con la Base de Datos y la interfaz del usuario para iniciar sesión.	163
Figura 72. Relación de las clases del software con la Base de Datos y la interfaz del usuario para cambiar las opciones de usuario	163
Figura 73. Diagrama entidad relación con las tablas Usuario y ArtículoUsuario.	164
Figura 74. Diagrama de clases-software para el caso de uso de agregar articulo, observe que se agrega la clase TArticuloUsuario que permite relacionar el articulo con el usuario activo.	165

Figura 75. Pantalla de la interfaz principal de GAIA, en el panel izquierdo ahora aparecen los artículos que han sido creados por el usuario actual.	165
Figura 76. Pantalla de la interfaz de usuario utilizada para iniciar sesión.	166
Figura 77. Pantalla de opciones de usuario.	166
Figura 78. Miniproyectos dentro de la quinta iteración.	167
Figura 79. DSS del caso de uso de agregar concepto.	169
Figura 80. DSS del caso de uso relacionar concepto.	169
Figura 81. DSS del caso de uso eliminar concepto.	169
Figura 82. DSS del caso de uso Buscar concepto.	170
Figura 83. DSS del caso de uso Editar concepto.	170
Figura 84. Diagrama de clases y su relación con la interfaz de usuario y la base de datos.	171
Figura 85. Modelo de datos final, observe que se añaden nuevas tablas.	172
Figura 86. Entorno principal del Glosario.	173
Figura 87. Interfaz de usuario para relacionar conceptos con artículos.	173
Figura 88. Pantalla utilizada para filtrar artículos a relacionar.	173
Figura 89. Iteraciones intermedias dentro de la primera iteración de la fase de construcción.	174
Figura 90. DSS caso de uso guardar articulo a Disco Duro.	174

Figura 91. DSS caso de uso cargar articulo a la Base de Datos.	175
Figura 92. TDatosDD es la clase encargada de guardar datos a disco duro y de cargar los artículos desde disco al programa.	175
Figura 93. Especificación de TDatosDD.	176
Figura 94. Mini – proyectos dentro de la segunda iteración. De la fase de construcción	176
Figura 95. Diagrama de clases e interfaz de usuario para agregar temas relacionados.	177
Figura 96. Diagrama de clases e interfaz para mostrar temas relacionados.	178
Figura 97. Miniproyectos dentro de la tercera iteración de la fase de construcción.	179
Figura 98. Diagrama de Secuencia del Sistema para el caso de uso de buscar en artículo.	179
Figura 99. Pantalla utilizada para ingresar expresión a ser buscada en el artículo.	180
Figura 100. Pantalla del centro de contenidos.	181
Figura 101. Vista del centro multimedia.	181
Figura 102. Vista del centro Multimedia.	182

GLOSARIO

BUCLE DE REALIMENTACION: Cadena cerrada de causa y efecto.

COMPLEJIDAD: Existencia de muchas partes diferentes conectadas.

COMPLEJIDAD DINAMICA: Existencia de gran numero de conexiones posibles entre las partes.

DIAGRAMAS CAUSALES: representaciones de sistemas en las que se indican los bucles de realimentación que se dan entre los diversos elementos del sistema y la forma en que están relacionados.

FENOMENO: Hace referencia a la realidad al hecho como tal del cual se hacen abstracciones para desarrollar un modelo.

FLUJO: Medida del cambio a lo largo del tiempo; por ejemplo, el índice de natalidad, los gastos o el uso de los recursos naturales. Los flujos llevan a un cambio de nivel (se denominan también tasa).

HIPERMEDIA: Integración de gráficos, sonido y vídeo en un sistema que permite el almacenamiento y recuperación de la información de manera relacionada, por medio de referencias cruzadas.

HIPERTEXTO: es un documento digital que se puede leer de manera no secuencial y el sistema que lo implementa.

HERRAMIENTAS DE AUTOR: Las herramientas de autor tienen la finalidad de servir como elemento de escritura y de edición para sus

autores, proporcionando lenguajes visuales de comunicación, en los que la manipulación directa de objetos, las cajas de diálogo y los menús desplegables son las formas más usuales de interacción hombre-máquina.

LENGUAJE UNIFICADO DE MODELADO (UML): lenguaje estándar para modelado de software, es un lenguaje para visualizar, especificar, construir y documentar los componentes de un sistema con gran cantidad de software.

MODELO: Descripción simplificada y práctica del funcionamiento de algo.

MODELOS MENTALES: Ideas y creencias que nos sirven para guiar nuestros actos, explicar procesos de causa y efecto cuando los vemos y dotar de significado a nuestra experiencia.

NIVEL: Cantidad que se acumula a lo largo del tiempo; por ejemplo, la población, el número de miembros de una familia o el dinero de una cuenta de ahorros.

OBJETIVO: Estado al que se pretende llegar.

REALIMENTACION: Reacción de un sistema, que actúa después como estímulo para el mismo sistema, o información devuelta que influye en un paso ulterior.

SIMULACION: Reproducción del funcionamiento de un sistema mediante modelos. Suele referirse a los programas informáticos que se utilizan para modelar sistemas complejos.

SISTEMA: Entidad con una finalidad, que mantiene su funcionamiento y existencia como un todo mediante la interacción de sus partes.

SISTEMICO: Que utiliza ideas de los sistemas.

TEORIA GENERAL DE LOS SISTEMAS: Conjunto de ideas y practicas basado en que los sistemas complejos comparten, con independencia de su contenido, determinados principios organizativos que pueden desvelarse y reproducirse mediante modelos matemáticos.

USUARIO: persona que interactúa con el sistema, también es denominado aprendiz.

RESUMEN EN ESPAÑOL

TÍTULO: HERRAMIENTA SOFTWARE PARA EL ESTUDIO DE FENOMENOS AMBIENTALES, MEDIANTE EL MODELADO Y LA SIMULACION CON DINAMICA DE SISTEMAS

AUTORES: SANTAMARIA PABON, Oscar Enrique., MENDEZ FORERO, Andrés Eduardo **.

PALABRAS CLAVES: Dinámica de Sistemas, Modelo, Documentos Interactivos, experimentar, Modelado, Diagrama de Forrester, Guía de estudio, Herramienta de autor, Objeto OLE.

DESCRIPCIÓN: GAIA 1.0 esta dirigido principalmente a investigadores y estudiosos del tema ambiental que estén interesados en aprender dinámica de sistemas y aplicarlo a su rama de estudio. El software organiza un conjunto de artículos que tocan el tema ambiental estudiado con dinámica de sistemas, además de una recopilación de modelos ambientales hechos con dinámica de sistemas; para ello el software se vale de los denominados documentos interactivos que permiten mostrar información valiéndose de diferentes recursos (imágenes, videos, audio, modelos y otros objetos OLE), se le permite al usuario manipular los modelos que encuentra dentro de los artículos valiéndose de un laboratorio de modelado en el cual puede experimentar con estos modelos o crear desde cero nuevos modelos. También se cuenta con unos tutoriales que introducen al usuario al modelado con dinámica de sistemas y al uso de los diagramas de forrester.

El software también tiene las características de una herramienta de autor pues permite crear contenidos y guías de estudio (utilizando la funcionalidad de los documentos interactivos), estas guías de estudio pueden ser creadas, editadas o

** Facultad de Ciencias Físico – Mecánicas
Escuela de Ingeniería de Sistemas.
Director: Hugo Andrade Sosa.

eliminadas, además pueden ser compartidas por diferentes usuarios, fomentando de esta forma el trabajo en grupo.

Adicionalmente, GAIA 1.0 ofrece varias posibilidades de navegación y gran cantidad de opciones de búsqueda, lo cual es indispensable para que el usuario ubique fácilmente la información que solicita.

De esta forma GAIA 1.0 se presenta como una herramienta que sirve como punto de inicio para desarrollar estudios del medio ambiente utilizando dinámica de sistemas; además la plataforma software puede ser utilizada en otros campos del conocimiento, simplemente modificando sus contenidos y aprovechando las características de herramienta de autor que ésta posee.

RESUMEN EN INGLÉS

TITLE: SOFTWARE TOOL FOR THE STUDY OF ENVIRONMENTAL PHENOMENONS USING SYSTEMS DYNAMICS MODELING AND SIMULATION.*

AUTHORS: SANTAMARIA PABON, Oscar Enrique., MENDEZ FORERO, Andrés Eduardo **

KEY WORDS: Systems Dynamics, Model, Interactive Documents, experimentation, Modeling, Forrester Diagram, Study Guide, authoring tool, OLE Object.

DESCRIPTION: GAIA 1.0 is issued mainly to environmental investigators and students interested in learning systems dynamics and apply it to their study field. The software organizes a set of articles that talk about the study of the environment using systems dynamics; to do so the software uses interactive documents that allow the use of different resources (videos, sounds, images, models and other OLE objects); the user can manipulate every model that belongs to an article using the Modeling laboratory which can be used to create, edit and manipulate models. Also there is a set of tutorials that introduces the user to the systems dynamics modeling process and the use of forrester diagrams.

The software has the characteristics of an authoring tool because it allows the creation of contents and study guides (using the functionality of the

* Graduation Work

** Physics and Mechanics Sciences Faculty
Systems Engineer School
Director: Hugo Andrade Sosa

interactive documents), these study guides can be created, edited, deleted and can be shared between different users promoting the work group between the users of the software.

GAIA 1.0 also offers many navigation facilities and a lot of search options, that are very helpful in order to find exactly and quickly the information that the user needs.

In this way, GAIA 1.0 represents a tool that can be used as a starting point to develop environmental studies using systems dynamics: the software platform can also be used in other knowledge fields, simply by modifying its contents and using the authoring tool characteristics that the software has.

INTRODUCCION

El hombre siempre ha tenido un afán por conocer su hábitat, entenderlo y comprenderlo; mucho se ha hecho para tratar de conocer el por qué y el cómo de todos los fenómenos que día a día ocurren a nuestro alrededor, desde el hecho de la existencia del día y la noche, las lluvias, las inundaciones, incendios, huracanes, los ciclos fundamentales de la naturaleza y todos aquellos fenómenos donde jugamos un papel fundamental, pasando por todo lo relacionado con la contaminación de cada una de las esferas del planeta¹ y el estudio de la estrecha relación entre los seres humanos y su entorno.

El afán de comprender el entorno natural y la relación humana con éste, ha llevado a la comunidad científica a orientar esfuerzos en este aspecto; los trabajos con este objetivo, en el ámbito nacional e internacional, son abundantes al igual que el número de ramas del conocimiento que se han preocupado por el tema. Es así como ha surgido una interdisciplinariedad que ha alimentado benéficamente el conocimiento respecto a los diferentes fenómenos que ocurren en el planeta.

A pesar de esta interdisciplinariedad y por ser los fenómenos ambientales sistemas complejos, los acercamientos tradicionales para modelar y estudiar este tipo de fenómenos son muy limitados a la hora de explicar su complejidad y parecen mas modelos de tipo caja negra, lo cual poco ayuda a la comprensión de su complejidad

¹ Geosfera, Atmósfera, Hidrosfera, Magnetosfera, Biosfera

dinámica² y las relaciones entre las partes que lo conforman; es así como sin dejar a un lado ni menospreciar los aportes de otras formas de pensamiento y metodologías de modelado, resulta útil y benéfico abordar este proceso de modelamiento desde una perspectiva que observe o contemple a los fenómenos ambientales como sistemas abiertos³ que forman parte de un sistema⁴ mas grande llamado tierra; además de considerar el todo y las partes, así como las conexiones entre las partes, estudiando el todo para poder comprender mejor las partes. La observación de los fenómenos ambientales desde una perspectiva como ésta no es una idea nueva, de hecho, el pensamiento sistémico se caracteriza por tener este punto de vista holista del mundo y se ha utilizado en reiteradas ocasiones para explicar fenómenos diversos de la naturaleza y para abordar temas que tratan de la interacción del hombre con su entorno; es así como nacen las ideas de desarrollo sostenible, de ver a los seres humanos como parte del todo, formando parte de una interacción armoniosa entre los sistemas humanos y los demás sistemas de la naturaleza.

De esta forma se pretende superar el mecanicismo; la tantas veces utilizada forma de pensamiento analítico recomendada por Descartes, muy útil para explicar el funcionamiento de una maquina, ya que de hecho es posible desarmar cualquier maquina en sus piezas componentes, entender y comprender cada una de las piezas, estudiar el mecanismo por medio del cual interactúan entre si y luego ponerlo todo junto de nuevo y se entenderá la totalidad. Esta aproximación o forma de pensamiento es llamada también pensamiento reduccionista, debido a que trata de entender el todo reduciéndolo al estudio de sus partes. Pero definitivamente esto no es posible hacerlo con los sistemas vivientes y

² Existencia de gran numero de conexiones posibles entre las partes.

³ Sistemas que interactúan con su entorno, obteniendo de el los recursos que necesitan.

⁴ Entidad con una finalidad, que mantiene su funcionamiento y existencia como un todo mediante la interacción de sus partes.

naturales, pues si se divide una cosa o ser viviente⁵ en piezas simplemente muere y deja de existir como tal, lo cual lleva a inferir que la forma de pensamiento reduccionista/mecanicista no es apropiada para comprender los sistemas vivos.

Dicha forma de pensamiento, con énfasis en el todo, surgió con el nombre de pensamiento holista⁶ o pensamiento organicista que es el antecesor de lo que hoy se conoce como pensamiento sistémico; el mayor aporte que ofrece esta forma de pensamiento es su enfoque, no centrado en las partes sino en la totalidad, desde esta perspectiva se puede dividir un sistema en partes, pero estas partes no están aisladas y la naturaleza del todo es siempre diferente a la mera suma de sus partes. Esta forma de pensamiento esta basada en las conexiones y relaciones existentes entre los elementos componentes del sistema, lo cual resulta un punto de vista adecuado para el estudio de la naturaleza, que nunca muestra bloques ni elementos aislados sino mas bien una compleja red de relaciones constituyendo un todo unificado y dinámico.

Se consigue el lente adecuado; de la misma forma que no se intentaría observar organismos microscópicos con un telescopio, ni estudiar las estrellas y las galaxias con la ayuda de un microscopio, no se estudia a la naturaleza y sus fenómenos desde una lente que permita ver solo las piezas componentes sin ver la totalidad.

Se tiene la lente, la perspectiva y el enfoque, pero ahora se necesitan las herramientas de laboratorio para poder explicar y experimentar con lo que se observa a través de la lente; se necesita entonces una metodología de modelado que tenga un paradigma holista y que aproveche la visión del pensamiento sistémico; la dinámica de sistemas

⁵ Se hace referencia a cualquier sistema viviente y que interactúa con el medio ambiente.

⁶ Del Griego Holos (el todo).

aparece como una alternativa apropiada pues se presenta como un paradigma-lenguaje que comparte el afán holista del pensamiento sistémico, complementándose entre si, de tal forma que se adopta a la dinámica de sistemas como una metodología de modelado que comparte la manera de ver el mundo del pensamiento sistémico y cuyo objetivo es mostrar cómo comprender la estructura causal de los fenómenos de estudio, es decir, entender las relaciones entre los componentes del sistema y por tanto al sistema como totalidad; de esta forma se trabaja en una metodología de modelado que permite entender la estructura y funcionamiento del sistema.

A partir de las premisas señaladas, surge la idea de crear un ambiente computacional o herramienta software que permita apoyar la labor de aprender el mundo desde el enfoque sistémico y explicar los fenómenos con el apoyo de modelos de simulación. Además poner a disposición de los usuarios micromundos o plataformas de aprendizaje⁷ sobre diferentes fenómenos de la naturaleza, en las cuales la dinámica de sistemas sirve para recrear las explicaciones mediante modelos computacionales inteligibles que ayudan a comprender la estructura causal y a que el usuario construya los conocimientos que le permitan desarrollar sus propias plataformas.

El resultado es una herramienta software que pretende apoyar el aprendizaje a los estudiosos del tema ambiental o simplemente a cualquier persona con espíritu investigador, aplicando el pensamiento sistémico y el paradigma dinámico-sistémico para estudiar y desarrollar modelos de fenómenos ambientales, todo esto mediante modelos, ejemplos didácticos, aplicaciones y colocando a disposición del usuario un compendio de los logros teóricos y aplicativos de la comunidad

⁷ También denominados juegos de simulación, simuladores o plataformas de aprendizaje, los cuales pueden ser utilizados para el aprendizaje mediante la interacción y la realimentación (Simon, "realimentar para aprender", 1990).

internacional. Así, se ofrece al usuario, un pequeño laboratorio para el estudio de los fenómenos ambientales y el aprendizaje de la Dinámica de Sistemas, el cual no pretende ser una herramienta definitiva, sino servir como guía y abonar el camino para futuras aplicaciones de la Dinámica de Sistemas en el tema ambiental. Además, éste software posee las características de una herramienta de autor que permite ampliar su uso para la creación de guías de estudio y autoaprendizaje multimediales e interactivas en cualquier rama del conocimiento.

1. PRESENTACION

El presente documento contiene toda la información correspondiente al desarrollo del proyecto "HERRAMIENTA SOFTWARE PARA EL ESTUDIO DE FENOMENOS AMBIENTALES, MEDIANTE EL MODELADO Y LA SIMULACION CON DINAMICA DE SISTEMAS" el cual generó un software llamado GAIA, este documento explica los fundamentos teóricos que sirven como base para su desarrollo, la metodología seguida, los procedimientos seguidos, el conocimiento generado a partir del desarrollo del proyecto, así como todos los inconvenientes surgidos a lo largo del proceso investigativo y de elaboración del software.

El documento consta de las siguientes partes:

1.- Un capítulo llamado *preliminares*, que contiene un resumen de los datos más importantes del proyecto, en este resumen se encuentran los objetivos y alcances del proyecto, antecedentes, justificación y planteamiento del problema.

2.- Un *marco teórico y antecedentes* que ofrece una guía de los temas que son tratados a lo largo del documento y donde se encuentra todo el marco conceptual que rodea la postura de modelado utilizada y también a la Dinámica de Sistemas como forma de aprendizaje (aprender – haciendo y entendiendo), se plantean aquí elementos que sustentan el por qué abordar al medio ambiente desde la perspectiva dinámico – sistémica y los beneficios de utilizar la dinámica de sistemas para comprender mejor su complejidad, esta sección es relativamente extensa debido a el carácter exploratorio del proyecto y a la ausencia de trabajos similares a nivel nacional, de forma tal que este marco teórico sirva para

abonar el camino a futuras investigaciones donde se aplique la dinámica de sistemas y el paradigma dinámico – sistémico en el estudio del medio ambiente; se menciona aquí también las características de las herramientas computacionales que asisten los procesos de aprendizaje; además se mencionan algunos temas referentes al desarrollo del software. Algunos temas no son tenidos en cuenta porque se consideran de conocimiento general de los lectores; a continuación se mencionan los temas en los cuales el lector debe tener un grado o nivel de conocimiento mínimo, para poder entender toda la información presentada en este informe:

Tabla 1. Conocimientos previos.

Temas	Nivel de conocimiento
Teoría General de Sistemas	Medio
Pensamiento Sistémico	Medio
Dinámica de sistemas	Medio
Ing. Ambiental, Conocimientos de fenómenos ambientales	Bajo
Diseño orientado a objetos	Medio
Programación orientada a objetos	Medio
Diseño basado en componentes	Medio
Diagramas UML	Medio
Entorno de programación DELPHI	Bajo

3.- Un *Desarrollo del proyecto* en el cual se describe todo lo ocurrido durante la elaboración, se menciona en este aparte la recopilación de trabajos nacionales e internacionales sobre el tema, a partir de lo cual se obtuvo información del estado del arte, también se explica el trabajo realizado con los alumnos de Sistemas Dinámicos 2 del primer y segundo

periodo académico del año 2003 y con los estudiantes de la especialización en ingeniería ambiental de la UIS de las sedes de Bucaramanga y Bogota, se menciona también la metodología usada para la elaboración de los modelos con Dinámica de Sistemas y además detalles sobre cada una de las etapas de los ciclos del desarrollo y la descripción de los modelos en UML del software; en general se muestra todo el conocimiento que ha sido generado a partir de la elaboración de este proyecto y los procedimientos y metodologías utilizadas para llevarlo a conclusión.

4.- *Conclusiones y Recomendaciones* para las versiones o proyectos futuros. Muy importante al tratarse este de un proyecto "exploratorio" tanto dentro del grupo SIMON, como en la comunidad del Pensamiento Sistémico, la Dinámica de Sistemas y el tema ambiental a nivel nacional. Además con las recomendaciones dadas, es posible extender aun más la funcionalidad de la herramienta y facilitar el trabajo de las personas que estén interesadas en continuar el trabajo que ha sido empezado en este proyecto.

5.- Finalmente se anexan:

Manual del Usuario. Para facilitar el manejo de la herramienta para los usuarios, se explica aquí todas las funciones que contiene el software y la manera como el usuario puede accederlas.

Convenciones, estilo de programación y código de las clases del software. Sirven como punto de inicio para cualquier persona interesada en continuar el desarrollo de la plataforma software, se incluyen convenciones utilizadas en la programación, declaraciones de las clases del software, unidades incluidas dentro del proyecto (unidades de Delphi) y recomendaciones de implementación de código para futuras versiones.

2. PRELIMINARES

2.1 ESPECIFICACIONES DEL PROYECTO

2.1.1 Título.

HERRAMIENTA SOFTWARE PARA EL ESTUDIO DE FENOMENOS AMBIENTALES, MEDIANTE EL MODELADO Y LA SIMULACION CON DINAMICA DE SISTEMAS

2.1.2 Director de proyecto.

Profesor Hugo Hernando Andrade Sosa Facultad de Ciencias Físico - Mecánicas. Escuela de Ingeniería de Sistemas e informática.

2.1.3 Codirector de proyecto.

Profesor Edgar Castillo. Facultad de Ciencias Físico - Químicas. Director Centro de Estudios e Investigaciones Ambientales.

2.1.4 Autores.

OSCAR ENRIQUE SANTAMARÍA PABON CÓDIGO 1981105; ANDRÉS EDUARDO MÉNDEZ FORERO CÓDIGO 1981140.

2.1.5 Entidades interesadas.

Este proyecto es de interés en instituciones educativas y en los centros de investigación ambiental de Colombia, pero especialmente en:

- **UNIVERSIDAD INDUSTRIAL DE SANTANDER.** La Universidad Industrial de Santander se ha caracterizado por ser una de las mayores productoras de trabajos de investigación a nivel nacional, el presente proyecto puede sentar las bases para iniciar una labor investigativa en el campo de la Ingeniería ambiental y la Dinámica de Sistemas.

La Dinámica de Sistemas puede ser de gran utilidad para el estudio de diferentes áreas del conocimiento, no solo en el caso particular de los fenómenos ambientales que ocupa el presente proyecto, de tal forma que su aplicación podría ejecutarse en diferentes ramas del conocimiento impartidas dentro de la universidad. Además la plataforma⁸ desarrollada puede ser utilizada en otras áreas del conocimiento, no solo en el estudio del medio ambiente.

• **GRUPO SIMON DE INVESTIGACIONES.** Por la aplicación del Pensamiento sistémico y la Dinámica de Sistemas, que junto con el constructivismo estructuran la propuesta educativa del Grupo SIMON. El grupo SIMON siempre esta en la búsqueda de aplicar su propuesta educativa en diferentes áreas del conocimiento; en este caso se aplicara dicha propuesta al campo de la ingeniería ambiental y del estudio del medio ambiente.

• **ESPECIALIZACION EN INGENIERIA AMBIENTAL DE LA UIS.** Actualmente existe una materia donde se imparten conceptos fundamentales de Dinámica de Sistemas a los estudiantes de dicha especialización; la herramienta software que se obtenga como resultado de este proyecto puede ser de gran utilidad para los estudiantes de dicho curso y puede servir de apoyo en su proceso de aprendizaje.

• **CENTRO DE ESTUDIOS E INVESTIGACIONES AMBIENTALES.** El centro de estudios e investigaciones ambientales esta en constante exploración y búsqueda de nuevo conocimiento en el tema ambiental, el Pensamiento Sistémico y la Dinámica de Sistemas aplicado a su área de conocimiento, mostrado a través de la herramienta software producto de

⁸ NOTA DE LOS AUTORES: se hace referencia a plataforma como al "casarón" del software sin los contenidos, es decir al software con todas sus utilidades y con la base de datos en blanco.

este proyecto, podrá ser de gran ayuda para abordar el estudio y comprensión de algunos de los fenómenos ambientales que se estudien en este centro.

2.1.6 Objetivo General.

Desarrollo de una herramienta software para el estudio de fenómenos ambientales, mediante el modelado y la simulación con dinámica de sistemas; organizando información teórica al respecto, reproduciendo y elaborando ejemplos didácticos y aplicaciones; mostrando así la utilidad de la dinámica de sistemas como herramienta para la explicación de fenómenos ambientales.

2.1.7 Objetivos específicos.

- Acopiar y clasificar información competente acerca de trabajos realizados a nivel nacional e internacional sobre fenómenos ambientales, vistos desde un enfoque dinámico – sistémico con el fin de colocar a disposición de usuarios novatos dichas aplicaciones.

- Reproducir y elaborar modelos ilustrativos de fenómenos ambientales desarrollados con el enfoque del pensamiento Dinámico-Sistémico para garantizar y afirmar la utilidad de este enfoque y en particular de la Dinámica de Sistemas al modelar este tipo de fenómenos.

- Desarrollar una herramienta software para el estudio de fenómenos ambientales, que permita básicamente a los usuarios:
 - Operar con toda la información surgida de los objetivos específicos anteriores.
 - Diseñar y ejecutar experimentos con los modelos de ejemplo o los incluidos por el mismo usuario.
 - Elaborar sus propios modelos.

- Un continuo mantenimiento (correcciones y complementos) de todos los contenidos del software mediante:
 - ✓ Facilidades para modificar e incluir nueva información teórica.
 - ✓ Facilidades para incluir nuevos experimentos y modelos y en general para cambiar cualquier contenido del software.

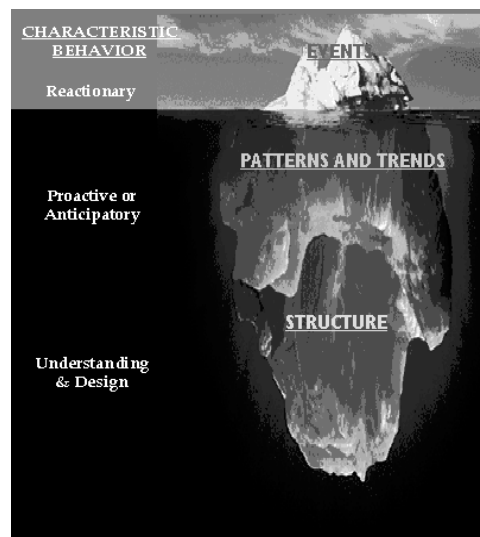
2.1.8 Justificación y Definición del Problema.

Se llama ambiente a todo aquello que rodea al ser humano, no solo el ambiente físico, sino también aspectos de orden social; incluso se tratan cuestiones tan delicadas como la religión o tan básicas como la alimentación. Los tremendos problemas que la humanidad enfrenta en este inicio de siglo, algunos bien conocidos por sus efectos directos sobre la salud humana, como la contaminación del aire, agua y suelo, y otros más sutiles y difíciles de detectar como la reducción de la biodiversidad o el desbalance en el ciclo del carbono; son realidades que deben entenderse. Finalmente resulta obvio que el planeta tiene límites para la asimilación de desechos y soportar abusos, es necesario saber que si se quiere seguir disfrutando la vida, se debe tener cuidado en la manera como se aprovechan los recursos que lo sostienen.

Como futuro ingeniero o científico, al estudiante no le basta entender las causas y los efectos de los problemas ambientales en términos exclusivamente cuantitativos. También debe ser capaz de entender el problema que se percibe y su solución potencial de manera cualitativa, es decir, comprender su estructura causal. A menudo los fenómenos son muy complejos y mas de las veces la explicación suele ser meramente superficial sin comprender realmente la complejidad del fenómeno; es allí donde entra el paradigma dinámico-sistémico. Alguna vez, se ha oído hablar acerca de la expresión "la punta del iceberg" que es usada

comúnmente para indicar que aquello que se ve es tan solo el anuncio parcial e incompleto de algo mayor que se oculta pues yace “bajo la superficie marina”. Según la atención prestada a la porción oculta del iceberg se pueden distinguir tres miradas: Una mirada superficial que aprecia tan solo la punta del iceberg. Una mirada de corta profundidad que se aventura tímidamente para ver una parte de la porción sumergida. Y, finalmente, una mirada profunda que logra admirar el iceberg en su totalidad, es decir, que mira el iceberg como sistema, es esta última mirada la que corresponde al paradigma de pensamiento de la dinámica de sistemas.

Figura 1. El Iceberg⁹



Esta forma o paradigma de pensamiento se expresa a través de un sistema de convenciones, es decir de un lenguaje particular, de modo que la dinámica de sistemas es concebida también como una unidad

⁹ Esta figura está inspirada en el “Diagrama del Iceberg” que aparece en el sitio web del curso sobre pensamiento sistémico que desarrolla el Laboratorio Langley de la NASA (<http://www.dcb.larc.nasa.gov/larcst/Iceberg.html>).

paradigma-lenguaje que permite utilizarla como una metodología de modelado

Es este paradigma-lenguaje el que será utilizado a lo largo del desarrollo del proyecto para estudiar diferentes fenómenos que ocurren en el medio ambiente. El estudio de la implícita y dinámica complejidad de estos fenómenos no es fácil; gracias al apoyo de la dinámica de sistemas este proceso podrá ser abordado, mostrando a su vez la viabilidad de la dinámica de sistemas como herramienta en esta labor.

De esta manera, los estudiosos de los fenómenos ambientales, y cualquier persona interesada en el tema ambiental podrá observar como por medio de la dinámica de sistemas pueden abordar los temas de estudio que les ocupan.

Para mostrar la factibilidad de uso de la dinámica de sistemas en el estudio de los fenómenos ambientales, y para apoyar el proceso de aprendizaje de estos fenómenos y de la Dinámica de sistemas misma, se desarrollo una herramienta software que permite al usuario experimentar con modelos y crear sus propios modelos de fenómenos ambientales, de tal forma que se puede ver la herramienta como un laboratorio de aprendizaje de dinámica de sistemas para expertos en el área de ambiental y para cualquier persona interesada en el tema. La herramienta también posee características de software de autor, o herramienta de autor que permite la creación de documentos interactivos que sirven para organizar guías de autoaprendizaje, que en este caso es de la dinámica de sistemas aplicada al modelado de fenómenos del medio ambiente.

Con el fin de conseguir los objetivos generales y específicos propuestos, es necesario entonces poner en practica diferentes tópicos de la ingeniería de sistemas como son la dinámica de sistemas y el pensamiento

sistémico, el análisis, diseño y construcción de software orientado a objetos, además del análisis y diseño de bases de datos; de tal forma se considera que este proyecto es apropiado como trabajo de grado¹⁰, requisito éste, para obtener el título de Ingenieros de Sistemas e Informática.

¹⁰ NOTA DE LOS AUTORES: No solo por la aplicación de los conceptos aprendidos a lo largo de la carrera de Ingeniería de Sistemas, además por el impacto que pueda tener este proyecto "exploratorio" en el estudio del medio ambiente.

3. MARCO TEORICO Y ANTECEDENTES

El carácter Práctico e Investigativo que envuelve el desarrollo de la Herramienta Software, se encuentra inmerso en un marco teórico bastante extenso; es necesario entonces hacer un aparte donde se traten los conceptos básicos que enmarcan el desarrollo del proyecto, es aquí donde de alguna manera se delimita lo correspondiente a los temas que moldean la postura de modelado ambiental que es adoptada en el proyecto¹¹, pues muchas veces no es claro el tratamiento que se le da a los conceptos de enfoque de sistemas, al pensamiento sistémico ni al modelado con Dinámica de Sistemas. Es importante resaltar aquí que no se pretende mostrar las ventajas o desventajas de la Dinámica de sistemas frente a otras metodologías de modelado, pues esta tiene un afán más hacia el aprendizaje sobre el fenómeno que un carácter predictivo o de pronóstico como las metodologías usadas comúnmente para representar fenómenos ocurridos en el medio ambiente. Además en la sección no se pretende solo plasmar conceptos sino generar y desarrollar un marco teórico que sirva para posteriores investigaciones donde los fenómenos del medio ambiente sean abordados con el Enfoque Sistémico y se utilice el Modelado con Dinámica de Sistemas. En este aparte también se encuentra un esbozo histórico y referencias a algunos trabajos anteriores que sirven como base o conocimiento inicial y que de alguna forma moldean el compendio teórico que se presenta.

También encontrará en esta sección la información correspondiente a conceptos básicos que enmarcan el desarrollo de la plataforma que sirve como base a la herramienta software.

¹¹ NOTA DE LOS AUTORES. Es importante destacar que todo el marco conceptual que rodea el tratamiento de los fenómenos ambientales usando el paradigma dinámico-sistémico tiene material suficiente como para escribir una monografía sobre el tema, de tal forma que se intenta aquí mostrar los elementos más importantes de esta postura.

Es importante mencionar que todo el marco conceptual referente al enfoque sistémico del medio ambiente tiene como base los trabajos realizados por Leonel Vega Mora en su trabajo Gestión Ambiental Sistémica; y en lo que respecta al modelamiento y a la Dinámica de Sistemas se toman como base los trabajos de Andrew Ford (Modelling the Environment), John D. Sterman (Business Dynamics) y Andrade, Dyner, Espinosa, Garay, Sotaquira (Pensamiento Sistémico: Diversidad en búsqueda de Unidad).

3.1 EL ENFOQUE SISTEMICO Y EL MEDIO AMBIENTE

3.1.1 Aproximación al enfoque sistémico.

Actualmente es axiomática la existencia de una interrelación entre todos y cada uno de los elementos constituyentes del mundo. La sociedad moderna se ha vuelto tan compleja que los cambios y medios tradicionales no son ya suficientes y se imponen actitudes de naturaleza holística o de sistemas, y generalista o interdisciplinaria.

Los complejos sistemas en tecnología, urbanización, gestión y protección del medio ambiente y en otros tantos campos de la vida contemporánea, piden cada vez más un acercamiento sintético en términos de "sistema", llegando incluso a tornarse indispensable. Se requiere por lo tanto, que los factores esenciales en problemas tan apremiantes como los de tipo ambiental, sean siempre considerados y evaluados como componentes interdependientes de un sistema total y en consecuencia de uno u otro modo se encuentran complejidades, "totalidades" o "sistemas", en todos los campos del conocimiento.

La palabra "sistema" tiene muchos significados, denotando en todo caso, el conjunto de principios, reglas, y/o cosas que ordenadamente

relacionadas entre si contribuyen a determinado objetivo [Bertalanfy, 1994].

El concepto de "sistema" ha invadido todos los campos de la ciencia y penetrado en el pensamiento y habla populares y en los medios de comunicación de masas, hasta el punto que el razonamiento en términos de sistemas desempeña un papel dominante en muy variados ámbitos, desde las empresas industriales y la logística militar hasta los temas reservados al medio ambiente y su gestión. Así pues, es muy común que se pida la aplicación de un enfoque sistémico a problemas apremiantes, tales como la contaminación ambiental, la congestión del tráfico, el planeamiento de ciudades, etc.

3.1.2 Desarrollo histórico de la teoría general de sistemas.

El termino sistema ha sido utilizado desde tiempos inmemoriales. Sin embargo, el concepto y enfoque de sistemas no fue evidente sino hasta principios del siglo XX, debido fundamentalmente a los importantes aportes del eminente biólogo Austriaco Ludwing Bon Bertalanfy, a raíz de la controversia presentada en la biología entre *mecanicismo* y *vitalismo u organicismo*¹², lo cual llevo a los biólogos al planteamiento de un punto de vista organismico, es decir, la consolidación de que los organismos vivos son cosas organizadas, lo que oriento la realización de innumerables estudios sobre el metabolismo, el crecimiento y la biofísica del organismo. Un paso en la dirección del punto de vista organismico fue la llamada "teoría de los sistemas abiertos y los estados uniformes", que

¹² El mecanicismo nacido de la física clásica del siglo XIX, suponía que el juego sin concierto de los átomos esta regido por leyes inexorables de la causalidad y generaba todos los fenómenos del mundo, inanimado, viviente y mental. No quedaba lugar para ninguna direccionalidad ni orden y por lo tanto, el mundo de los organismos aparecía como producto del azar, amasado por el juego sin sentido de mutaciones azarosas y el mundo mental como un fenómeno curioso y bastante inconsecuente de los acontecimientos materiales. El vitalismo u organicismo por su parte, era la doctrina de que los fenómenos vitales no eran explicables en términos de ciencia natural. Se tenían explicables sólo merced a la acción de factores animoides o enteecticos, -duendecillos dan ganas de decir-, que acechaban en la célula o el organismo; lo cual evidentemente era, ni mas ni menos, una declaración en quiebra de la ciencia. [Bertalanfy, 1950].

rápidamente condujo a una generalización mayor aún, que supuso la introducción de *nuevos modelos conceptuales*, aplicables a sistemas generalizados o a subclases, sin importar su particular género, ni la naturaleza de sus elementos componentes, ni las relaciones o fuerzas que los gobiernen [Bertalanfy 1950a].

Los modelos implantados con tal propósito tienen naturaleza interdisciplinaria y por lo tanto trascienden los compartimientos ordinarios de la ciencia y son aplicables a fenómenos en diferentes campos. Por este motivo pareció legítimo pedir una teoría no ya de sistemas de clase más o menos especial, sino de principios universales aplicables a los sistemas en general.

De ahí nace la Teoría General de los Sistemas –TGS-, cuyo tema es la formulación y derivación de aquellos principios que son validos para los sistemas en general y con la cual se alcanza un nivel en el que ya no se habla de entidades físicas y químicas sino que se discuten totalidades de naturaleza completamente general [Bertalanfy, 1950b, 1968, 1994].

Resumiendo se puede decir, que los motivos que condujeron a la postulación de la TGS se originaron con la inclusión de las ciencias biológicas, del comportamiento y sociales en la tecnología moderna. Eso exigió la generalización de conceptos científicos básicos e implicó nuevas categorías del pensamiento científico, en comparación con las de la física tradicional. Por todo esto, la TGS se constituye en una ciencia general de la "totalidad", es decir, problemas de organización, fenómenos no descomponibles en acontecimientos locales, interacciones dinámicas manifiestas en la diferencia de conducta de partes aisladas o en una configuración superior, etc., en una palabra, "sistemas", de varios órdenes, no comprensibles por investigación de sus respectivas partes aisladas.

• **Sistemas cerrados, sistemas abiertos y entropía.** Los sistemas pueden ser abstractos y concretos: Un sistema *abstracto* es una disposición ordenada de ideas independientes, mientras que un sistema *concreto* o *físico*, es un conjunto de elementos que actúan unidos para lograr un objetivo.

De acuerdo con la frontera que separa lo que pertenece o no al sistema, los sistemas pueden ser cerrados o abiertos:

Los sistemas cerrados, de los cuales se ha ocupado la física clásica. Son definidos como "aquellos que se consideran aislados del medio circundante o entorno y por lo tanto no exhiben entrada ni salida de materia o energía".

La termodinámica declara expresamente que sus leyes sólo se aplican a sistemas cerrados, y en particular, el segundo principio de la Termodinámica afirma que, en un sistema cerrado, cierta magnitud, la Entropía¹³, debe aumentar hasta el máximo, y el proceso acabará por detenerse en un estado de equilibrio.

Los sistemas abiertos, incluidos posteriormente en la expansión de la física, son definidos como "aquellos que intercambian materia y energía con el medio circundante y que exhiben importación y exportación, constitución y degradación de sus componentes materiales".

El mejor ejemplo quizá, de sistema abierto son los organismos, los cuales importan complejas moléculas ricas en energía libre y así, manteniéndose en estado uniforme, logran evitar el aumento de entropía, pudiendo desarrollarse hacia estados de orden y organización crecientes.

¹³ La entropía puede ser definida en varios sentidos figurados, así: En física, es una función termodinámica que mide la parte de la energía no utilizable contenida en un sistema. En mecánica, es una medida del desorden de un sistema.

En el caso planteado, el medio ambiente y todos los fenómenos que ocurren en él, corresponde a sistemas abiertos pues todos los elementos o subsistemas están relacionados entre si por una espesa red de conexiones y no se puede ver a un fenómeno específico como un sistema aislado sino mas bien como un subsistema de un gran todo llamado medio ambiente donde el intercambio de materia y energía ocurre continuamente.

• **Características Básicas de los sistemas.** Existen varios conceptos básicos en la TGS que es necesario tener en cuenta, a continuación se citan los mas importantes:

○ **Isomorfismo.** Como consecuencia de la existencia de propiedades generales de los sistemas, aparecen similitudes estructurales o isomorfismos en diferentes campos. En el caso del medio ambiente los ejemplos son numerosos, se puede aplicar una ley exponencial al crecimiento de ciertas células bacterianas, a poblaciones de animales de cualquier especie, a fenómenos de deforestación y contaminación, etc. Las entidades en cuestión, bacterias, animales, árboles, agentes contaminantes, etc., son completamente diferentes, sin embargo, la ley matemática es la misma.

En otras palabras, hay correspondencias entre los principios que rigen el comportamiento de entidades que son intrínsecamente muy distintas, y se debe, a que las entidades consideradas pueden verse, en ciertos aspectos, como sistemas o sea, complejos de elementos en interacción.

○ **Información y Realimentación.** La información constituye una importante característica de los sistemas, la cual no es expresable en general, en términos de energía, sino en términos de decisiones y como la entropía es una medida del desorden, la información será una medida del

orden o de la organización, por lo que se denomina como entropía negativa o neguentropía. Se puede decir que la información es el elemento básico de un sistema.

Una parte importante de dicha información entra a formar parte del proceso de realimentación, a merced del cual se mantiene constante la situación material y energética de los sistemas. Los dispositivos de realimentación se emplean mucho en la tecnología moderna para estabilizar determinada acción, como en los termostatos o los receptores de radio, o la dirección de acciones hacia determinada meta: las desviaciones se realimentan con información, hasta que se alcanza la meta o el blanco, como es el caso de los servomecanismos.

De manera similar, hay un gran número de fenómenos en los sistemas abiertos, que corresponden al modelo de realimentación, especialmente en los organismos vivos, los cuales consiguen tender "activamente" hacia estado de mayor organización, es decir, pasar de un estado de orden inferior a otro de orden superior, merced a "aprendizaje" mediante la información administrada al sistema mediante su mecanismo de realimentación. Es así como se observa que el planeta siempre tiende a mantener un equilibrio, por ejemplo en el caso de las cantidades de CO₂, las emisiones son captadas por los bosques; no obstante el hombre ha entrado como un elemento desbalanceador del sistema, lo cual ha provocado las fluctuaciones que hoy se conocen de los niveles de CO₂ en la atmósfera; es por esto que los seres humanos deben verse como parte integral del sistema tierra y no como elementos aislados; los ejemplos son muchos; dentro de la herramienta software hay gran cantidad de ejemplos donde el hombre de alguna forma rompe el equilibrio que existe naturalmente en los sistemas ambientales.

3.1.3 El medio ambiente como sistema.

La moderna concepción global e integral del medio ambiente como sistema está íntimamente ligada al reconocimiento de la interrelación directa entre el sistema natural y el hombre. Todo esto tendría que ver con la integración de dos nuevos axiomas en el mundo moderno: por un lado, las interdependencias del mundo viviente y, por otro lado, la congruencia de los conceptos Tierra-Mundo. El desarrollo conceptual de estos dos axiomas permitirá comprender con más claridad el medio ambiente como sistema.

• **Interdependencias del mundo viviente.** La interconexión del mundo viviente con la Tierra ha sido reconocida de forma intuitiva durante mucho tiempo, pero solo hasta el siglo XX no llegó a ser de uso general una terminología estándar para designar esas interconexiones específicas y sistemáticas del mundo natural.

El conocimiento de la humanidad sobre dicha interrelación fue creciendo gracias a las sucesivas aportaciones de conocimientos científicos. Un poco de revisión histórica permitirá comprender con más claridad dichas interdependencias.

La mayor preocupación de la ciencia biológica del siglo XVIII y gran parte del XIX estuvo orientada al descubrimiento de las interdependencias del mundo viviente mediante la taxonomía y luego, a partir del siglo XX mediante la sistemática, con las cuales los organismos comprendidos han sido clasificados, localizados y descritos, aunque esta tarea dista mucho de estar completa.

En el siglo XIX se descubrió que la distribución espacial de las plantas y de los animales no era fortuita ni estática, sino que estaba relacionada

con parámetros biológicos como la competencia, la simbiosis y la preferencia alimentaria y territorial.

Así mismo, gracias a los importantes aportes realizados en 1859 por el biólogo británico Charles Darwin en su obra *Origen de las especies mediante selección natural*, se fue entendiendo el mecanismo teórico de la evolución de las especies, al manifestar que, ... *en cualquier momento dado, la red de interdependencias del mundo viviente es un estado de equilibrio aproximado, aunque dinámico y por lo tanto, sujeto a cambios mediante fuerzas que actúan no sólo en el entorno físico externo de los organismos, sino mediante cambios genéticos en los propios organismos...* [Darwin. 1985] (citado por Leonel Vega Mora, Gestión Ambiental Sistémica).

En 1867, Ernst Haeckel propuso la palabra *ecología* para designar "el estudio de los sistemas vivientes con relación a su entorno". Con el transcurso del tiempo, el término ecología ha adquirido significados populares y filosóficos que van más allá de las precisas definiciones científicas. No obstante, la interconexión de las cosas en el tiempo era comprendida con menor facilidad que su interconexión en el espacio. Tal es el caso de la teoría de la evolución de Darwin, la cual se enfrentaba con intervalos de tiempo mucho mayores que las experiencias de cualquier individuo humano y más allá de la comprensión de la mayoría. Lo anterior obligó a un cambio en la concepción y el significado del tiempo en temas humanos, labor propiciada en parte por el trabajo del científico Albert Einstein, explorando la relatividad del tiempo y el espacio [Bernstein, 1993].

En 1929, Vernadsky publica su libro *Biosphere*, en donde define la biosfera como el área o el campo de la vida; una región donde las condiciones imperantes son tales que la entrada de la radiación solar

puede producir los cambios geoquímicos necesarios para que se origine la vida y que comprende la troposfera atmosférica, la hidrosfera u océanos y las capas superiores de la litosfera [Vernadsky, 1929].

Dado que el estudio de la biosfera en su conjunto, y el de sus interacciones es prácticamente imposible debido a su complejidad y extensión, en 1935 Tansley define el término ecosistema como una unidad funcional de menor amplitud de la biosfera para significar "un sistema definible o limitado de interrelaciones físicas y biológicas dinámicas y complejas que varían enormemente de tamaño y de complejidad, de lo diminuto o simple a lo grande y complejo" [Tansley, 1935] (citado por Leonel Vega Mora, Gestión Ambiental Sistémica).

El ecosistema está pues compuesto por dos componentes: la biocenosis o comunidad, es decir, la parte viva que lo habita, constituida por las poblaciones vegetales y animales y, el biotipo, o parte inanimada, el cual actúa como soporte o substrato, donde actúan los factores abióticos que determinan las características físico-químicas del medio.

Posteriormente en 1945, Vernadsky desarrolló el concepto de la noosfera, declarando que, "la noosfera, o área del pensamiento, es un fenómeno geológico nuevo en nuestro planeta, donde el hombre se convierte en una fuerza geológica de cambio a gran escala" [Vernadsky, 1945]. Los conceptos de biosfera y noosfera constituyen lo que se conoce como la "síntesis biosfera-noosfera", o congruencia Tierra - Mundo, es decir, el problema de la interdependencia del mundo viviente visto en términos políticos, donde se reconoce la enorme influencia y por lo tanto, responsabilidad del hombre en el mantenimiento del equilibrio natural, como se verá a continuación.

• **Congruencia Tierra - Mundo.** Con la publicación en 1979 por Lovelock de su libro *Gaia, Una nueva visión de la vida sobre la tierra*, se dio paso a una nueva concepción de la biosfera dentro de una nueva síntesis unificadora y dinámica del planeta. En dicho libro el autor desarrolló la propuesta de que: "la biosfera es una entidad que se autorregula con capacidad de mantener nuestro planeta sano, controlando el medio ambiente químico y físico" [Lovelock, 1979].

Dicha propuesta, conocida como la Hipótesis Gaia, constituye una progresión lógica de la síntesis biosfera - noosfera de Vernadsky y proporciona una interpretación unificadora de las relaciones entre los aspectos inanimados y animados de la Tierra.

La Tierra es un recurso finito y los recursos naturales que sustenta pueden variar con el tiempo y según las condiciones de su ordenación y los usos que se les den. Por todos es conocido que las crecientes necesidades humanas y el aumento de las actividades económicas ejercen una presión cada vez mayor sobre los recursos naturales, suscitando la competencia y los conflictos y llevando por supuesto, a un uso impropio de los mismos.

El Mundo ha sido considerado como esa parte de la Tierra que algunas sociedades concretas conocían, pero en definitiva era sólo esa parte. Así, el Mundo era el dominio de los humanos y cuando se utilizó el término para incluir toda la tierra como en los términos "el mundo de la naturaleza" o "el mundo natural", el dominio humano estaba implícito.

De esta manera, aunque Tierra y Mundo pueden ser conceptos separables, con los tiempos modernos, ha ocurrido un cambio significativo en la perspectiva del modo en que la gente ve y evalúa su relación con su

entorno¹⁴, hasta el punto de ver la especie humana como un componente en el interior de los sistemas vivos de la Tierra.

Ahora que la relación del Mundo con la Tierra está cambiando tan profundamente, es necesario abrir los ojos a este cambio y conocer sus implicaciones, ya que aunque Tierra y Mundo se han vuelto físicamente congruentes, aún no son universales y conceptualmente idénticos. En otras palabras, Tierra y Mundo deben ser considerados congruentes con los planes y las esperanzas humanas.

• **Concepto amplio del medio ambiente.** El carácter interactuante de los recursos naturales determina que la ecología, en cuanto a disciplina de síntesis, cuyo objetivo de estudio es el ecosistema, deba estar presente en todo lo que se refiere al conocimiento y manejo de dichos recursos. El alcance del término ecosistema es más conceptual que espacial y se refiere, como se ha descrito, a la organización de la vida y a las interacciones entre los componentes bióticos y abióticos, por lo que su reclusión dentro de unas fronteras determinadas, puede ser arbitraria.

Por estos motivos surge la necesidad del término medio ambiente, el cual de acuerdo con el *Diccionario de la Lengua Española* está definido como: "Elemento en que vive o se mueve una persona, animal o cosa"; y también como: "Conjunto de circunstancias físicas, culturales, económicas y sociales que rodean a la personas y a los seres vivos".

Estas dos definiciones implican que el medio ambiente es el entorno vital, o sea, el conjunto de factores abióticos (físico – naturales, estéticos culturales, sociales y económicos) y de factores bióticos o tróficos

¹⁴ Las imágenes simbólicas han sido importantes en este cambio, y una de ellas ha sido el efecto psicológico de la visión de la Tierra desde el espacio exterior, que constituye quizás el más significativo impacto del Programa Apolo de los Estados Unidos, al cambiar la imagen mental que el hombre tenía de la Tierra [Gore, 1993].

(parasitismo, prelación, competencia, etc.) que interaccionan entre sí, con el individuo y con la comunidad en que vive, determinando su forma, carácter, comportamiento y supervivencia.

La idea del medio ambiente abarca más amplitud que la del ecosistema, ya que además de los factores físico-naturales del biotipo, incluye factores perceptuales y socioeconómicos, inherentes a la presencia del ser humano. Si se considera al hombre independientemente del medio ambiente, la preocupación por el deterioro ambiental sería infundada, pues como lo demuestra la historia, la naturaleza ha tenido y tiene infinitas posibilidades y capacidades de sobreponerse a cualquier tipo de catástrofe antropogénica y más aun a las de tipo natural, que de hecho han sido las mas considerables.

En definitiva, lo que realmente el término significa son relaciones, puesto que indica una relación entre un objeto concreto y todo lo que le rodea y que directa o indirectamente le afecta y por lo tanto, la ciencia sobre el medio ambiente tiende a tener un carácter multidisciplinario y a ser una empresa colaboradora mas que singular.

Las relaciones ambientales se pueden, por lo tanto, investigar en una gran variedad de niveles, desde el subatómico hasta el cósmico, pero el enfoque principal de la ciencia sobre el medio ambiente en este trabajo consiste en las relaciones entre los humanos, principalmente sus organizaciones y los sistemas biogeológicos en los que viven.

Resumiendo, el medio ambiente no debería ser considerado como un sector más, en el cual se incorpora la formación de políticas, planes y proyectos a través de un conjunto de variables a las que pueda calificarse de ambientales, sino que en la ideología ambiental debe subyacer un enfoque sistémico que se caracterice por:

○ Visión de conjunto y por ello, concepción del medio ambiente como un conjunto de elementos en interacción dinámica entre los efectos de las intervenciones y las decisiones que se adopten.

○ Tratamiento multi-pluridisciplinario, como corresponde a esa visión de conjunto.

○ El uso de criterios racionales de sostenibilidad que permitan garantizar en el tiempo y el espacio, el aprovechamiento continuo de los recursos naturales como la protección del medio ambiente.

• **Gestión, administración y manejo ambiental.** Como consecuencia de la aparición del deterioro ambiental y ante la consideración del medio ambiente como sistema, surge la necesidad de "gestionar, administrar y manejar" el medio ambiente a fin de minimizar los problemas existentes y asegurar un equilibrio de fuerzas en la biosfera, mediante la creación de una alianza decidida entre el hombre y la naturaleza de la que el hace parte.

Así pues, para los efectos del presente marco conceptual, se define a la gestión ambiental como aquella que se ocupa de los temas relacionados con el medio ambiente contribuyendo a su conservación y comprende las responsabilidades, funciones, prácticas y recursos para determinar y llevar a cabo una política ambiental específica.

Así mismo, la administración y manejo ambiental será entendida como aquella parte de la administración que se ocupa de los temas relacionados con los recursos naturales y ambientales, siendo sinónimo de prever posibles impactos, organizar y aplicar métodos y tareas conducentes a

minimizar esos impactos, coordinar y controlar las actividades del hombre, en aras a la anulación, mitigación o corrección de los mismos.

3.1.4 Gestión Ambiental y Modelos Ambientales.

Los modelos son sustitutos de los sistemas reales y son usados cuando resulta difícil o engorroso trabajar directamente con el sistema real. Los planos de un arquitecto, el túnel de viento de un ingeniero, los gráficos de los economistas; todos representan algún aspecto de un sistema real (un edificio, un avión, o la economía nacional) y resultan útiles cuando ayudan a aprender mas acerca del sistema que representan.

Muchas personas han construido y usado modelos. Sus primeras experiencias muy seguramente envolvían modelos físicos como un avión o un barco de papel. Estos modelos resultaban fáciles de elaborar y de usar y hacían divertido el hecho de realizar experimentos. El proceso era sencillo, los aviones o barcos de papel eran puestos a prueba, se observaban los resultados, se hacían algunas pequeñas modificaciones y nuevamente se ponían a prueba; en el camino probando y volviendo a hacer los modelos de aviones y barcos de papel se aprendió de una manera rudimentaria los fundamentos del vuelo y la navegación. Muy seguramente, todas las personas aprendieron que no se puede hacer que un avión de papel vuele mas lejos simplemente ejerciendo mas fuerza a la hora de echarlo a volar. También se aprendió que cada modelo de avión parecía tener una trayectoria similar cada vez que se lanzaba y que esa trayectoria natural que mostraba podía modificarse dependiendo de la manera en que se moldearan las alas del avión.

En el caso particular de estudio, se realiza un enfoque en modelos matemáticos de sistemas ambientales, éstos usan ecuaciones diferenciales para representar las interconexiones en un sistema. Se hace énfasis en una categoría de modelos matemáticos, los cuales son

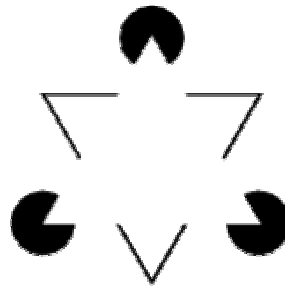
simulados en el computador. Son llamados Modelos de Simulación por Computador y mas específicamente dentro de esta categoría, se hace referencia a Modelos de Dinámica de Sistemas, de la cual se hablará con algo de detalle más adelante en el capítulo. De tal forma que con esta metodología de modelamiento no hay que preocuparse tanto por las ecuaciones que representan al modelo sino mas bien por pensar acerca de la mejor manera de construir un modelo que represente el sistema ambiental sobre el que se este trabajando. Si este trabajo es bien hecho, se puede entonces usar este modelo resultante, para realizar experimentos y probar políticas y estrategias en el sistema y recibir una realimentación por parte del modelo, de forma tal que se puedan ajustar nuevamente las estrategias para seguir determinada política; como se puede observar se aprende a partir de la información que devuelve el modelo y posteriormente se actúa nuevamente, formándose de esta manera un bucle de aprendizaje. Es de esta forma que las políticas que se piensan implementar sobre un sistema ambiental real pueden ser puestas a prueba en estos modelos matemáticos, lo cual sirve como apoyo en la tarea de la Gestión Ambiental; pues es mucho mas sencillo y menos riesgoso poner a prueba en el computador algunas acciones y ver el comportamiento del modelo bajo estas políticas que hacer ensayos directamente sobre el fenómeno o realidad. Es importante mencionar que al igual que una maqueta de un avión en el túnel de viento no se comporta exactamente igual que el avión real en un vuelo real, estos modelos no se comportarán exactamente igual que sus contrapartes del mundo real; pero igualmente dan unas pautas de comportamiento que pueden ser muy útiles para el estudio y la ejecución de políticas en el sistema real.

• **Modelos No Formales.** Los modelos, son usados todo el tiempo. Las imágenes que hay en la mente de las personas, son representaciones simplificadas de un sistema complejo y son comúnmente llamadas

“modelos mentales”. Peter Senge describe a los modelos mentales como “supuestos hondamente arraigados, generalizaciones e imágenes que influyen sobre nuestro modo de comprender el mundo y actuar”. Los modelos mentales, se usan constantemente para interpretar el mundo y realmente nunca o muy pocas veces la persona se da cuenta que los está usando.

Para entender qué tan rápido e inconscientemente son usados los modelos mentales, observe la figura 2 y trate de explicar lo observado.

Figura 2. Triangulo de Kanisza



Si su reacción es como la de los demás, seguramente vio un triángulo blanco superpuesto en la figura. Muy seguramente concluyó que el triángulo simplemente está allí; es la manera más sencilla de explicar el gráfico y esto se debe a los modelos mentales concebidos con anterioridad.

3.1.5 Pensando en el ambiente.

Aunque la mayoría de personas son muy diestras pensando en figuras geométricas y sus conceptos son muy claros y precisos al respecto, usualmente sienten frustración cuando piensan acerca de problemas serios en el ambiente; trátense de problemas localizados como la contaminación urbana o algún problema global como la acumulación de gases de efecto invernadero, la mayoría de las veces sus modelos

mentales resultan inadecuados para explicar el fenómeno a fondo, de tal forma que si se les diera la oportunidad de aplicar políticas ambientales, muy difícilmente harían lo adecuado, seguramente las primeras reacciones cuando actúan ante sistemas de tan alta complejidad no sean las adecuadas y terminen tratando de aplicar alguna política que ha funcionado en el pasado y pensar en aplicar esa política con "mas fuerza", lo que posiblemente no mejore la situación; es de recordar el caso del avión de papel donde aplicar mas fuerza puede hacer que se estrelle con mayor fuerza, es decir es posible que la situación empeore. De tal forma que es necesario aprender sobre el fenómeno para formarse un modelo mental que permita tomar mejores decisiones y aplicar políticas mas adecuadas; esta es la idea principal del aprendizaje por medio de la experimentación por computador; primero construir un modelo matemático que capture las interrelaciones claves en el sistema, luego se hacen experimentos con el modelo para aprender mas sobre el fenómeno y tener mayores bases para la toma de decisiones.

Pero estos modelos requieren ser realizados de una manera disciplinada y bajo cierta metodología, requieren que las suposiciones hechas sean claras y explicitas y que estén bien documentados de forma que otras personas puedan entender las suposiciones hechas y trabajar para mejorarlo haciendo que el modelo se aproxime aún más a la realidad.

3.1.6 Algunos antecedentes de modelos ambientales utilizando la dinámica de sistemas.

Son numerosos los estudios de tipo ambiental que han sido realizados apoyados en la dinámica de sistemas y muestran la factibilidad de uso de la dinámica de sistemas para abordarlos; a continuación se mencionan los que se consideran más importantes.

En 1970, el Club de Roma, una asociación privada compuesta por empresarios, científicos y políticos, encargó a un grupo de investigadores del Massachusetts Institute of Technology bajo la dirección del profesor Dennis L. Meadows, la realización de un estudio sobre las tendencias y los problemas económicos que amenazan a la sociedad global. Los resultados fueron publicados en marzo de 1972 bajo el título "Los Límites del Crecimiento"¹⁵. Este trabajo además de mostrar el uso de la dinámica de sistemas en el tema ambiental, es un hito en la historia de la Dinámica de sistemas y ayudó a su divulgación a nivel internacional, además logró crear gran controversia en el mundo científico, pues muchos lo consideran alarmista y pesimista.

En el estudio se utilizaron las técnicas de análisis de dinámica de sistemas más avanzadas del momento. En primer lugar se recopilaron datos sobre la evolución que habían tenido en los primeros setenta años del siglo XX un conjunto de variables: la población, la producción industrial y agrícola, la contaminación, las reservas conocidas de algunos minerales. Diseñaron fórmulas que relacionaban esas variables entre sí —la producción industrial con las existencias de recursos naturales, la contaminación con la producción industrial, la producción agrícola con la contaminación, la población con la producción agrícola, etc.— y comprobaron que esas ecuaciones sirvieran para describir con fidelidad las relaciones entre los datos conocidos que habían recopilado. Finalmente introdujeron el sistema completo en un ordenador y le pidieron que calculase los valores futuros de esas variables.

Las perspectivas resultaron muy negativas. Como consecuencia de la disminución de los recursos naturales, hacia el año 2000 se produciría una grave crisis en las producciones industrial y agrícola que invertirían el

¹⁵ Para mas información puede dirigirse a la pagina <http://www.ur.mx/tendencias/discurso/d-07.htm>. Donde encontrara un recuento de los resultados del estudio.

sentido de su evolución. Con algún retardo la población alcanzaría un máximo histórico a partir del cual disminuiría rápidamente. Hacia el año 2100 se estaría alcanzando un estado estacionario con producciones industrial y agrícola *per cápita* muy inferiores a las existentes al principio del siglo XX, y con la población humana en decadencia.

El club de Roma realizó algunos comentarios respecto al modelo que tiene un afán por mostrar la insostenibilidad el ritmo de crecimiento que lleva el planeta:

✓El mensaje de este libro es muy claro, no se puede crecer indefinidamente, el progreso tiene límites establecidos no por cuestiones tecnológicas sino por cuestiones ecológicas globales, existe la insostenibilidad ecológica.

✓Además, el crecimiento y el progreso nunca han sido uniformes para todos los habitantes del planeta, y esto es otra variable adicional que limita el crecimiento, la insostenibilidad social, que es un concepto bastante más reciente que el de la insostenibilidad ecológica que señala este libro.

✓El fundamento de las advertencias realizadas en este libro es sumamente válido para quienes entienden y aprecian los resultados provenientes de la simulación de modelos matemáticos, sin embargo hay muchas personas que simplemente no creen que el problema de anticipar lo que le puede pasar al sistema mundo dentro de 100 o 200 años sea posible matemáticamente, puesto que necesariamente se tendrían que hacer muchas simplificaciones que no sabemos qué consecuencia tengan en el resultado.

✓Muchos otros piensan que no vale la pena preocuparnos antes de tiempo por algo que seguramente resolverá la humanidad "a su debido tiempo".

- ✓Otros piensan que en cualquier momento podrán aparecer "de la nada" situaciones que modifiquen todas las anticipaciones realizadas.
- ✓Para cualquier persona debe ser cierto que si logramos hacer tender al sistema mundo a un estado de equilibrio global, todas esas cosas inesperadas, todas esas variables escondidas, podrán generar fluctuaciones respecto a ese estado global, pero muy difícilmente podrán generar un colapso global.
- ✓¿Qué cosa es un colapso global?
- ✓¿Han ocurrido colapsos globales anteriormente?
- ✓Ciertamente han existido guerras, plagas, epidemias y otras calamidades naturales de orden planetario o astronómico que han amenazado la vida sobre el planeta Tierra, y aparentemente de todas ellas la humanidad se ha recuperado.
- ✓Por supuesto que seguimos teniendo todas esas amenazas, la epidemia de Sida en África, hasta este momento, se estima que causará más de 20 millones de muertes y además ha reaparecido el virus de la Tuberculosis con una intensidad que aún desconocemos.
- ✓Lo que podemos señalar es que aunada a todas esas amenazas debemos considerar ahora, y nunca antes de ahora, el hecho de que la población del planeta ha crecido tanto y hemos sido capaces de desarrollar tanta tecnología, que podemos acabarnos el planeta.
- ✓No se puede negar el efecto positivo que este libro ha tenido en la mente y en la acción de quienes se han comprometido, por ejemplo a dejar de producir los compuestos químicos que estaban dañando la capa del ozono, sin embargo es necesario que tanto la conciencia como la acción a nivel individual participen para lograr el desarrollo sustentable.
- ✓Por definición desarrollo sustentable es tratar de hacer lo mejor con los recursos naturales que recibimos, siempre y cuando no dejemos menos recursos para el beneficio de las generaciones venideras. Esto por supuesto tiene el germen de la búsqueda de un equilibrio global; si no

podemos aumentar los recursos naturales que actualmente existen en la Tierra, al menos tratemos de conservarlos constantes.

Como se puede apreciar por los comentarios realizados por el club de Roma, el trabajo de Meadows es un gran pilar en el estudio del medio ambiente con la dinámica de sistemas y logro crear cierta conciencia a nivel mundial respecto a los posibles efectos que del comportamiento irracional de los seres humanos con su entorno.

Posteriormente a la publicación de esta investigación, los trabajos han sido varios, algunos de ellos son los siguientes:

✓An attempt to operationalize the recommendations of the "Limits to growth" study to sustain the future of mankind, elaborado por Surya RAJ, este trabajo pretende mostrar la manera en la que las recomendaciones hechas en el trabajo de Meadows sean aplicadas y llevadas a la realidad.

✓A simple and useful model of global scale pollution, por Robert Mackey. Explora el fenómeno de contaminación a escala global.

✓Simulation of policy alternatives to prevent deforestation and soil erosion in Turkey, por Yesim Tosan. Plantea el problema de deforestación y erosión del suelo en Turquía y propone políticas alternativas.

✓Modelo de deforestación de caparo, autores varios. Presenta el problema de deforestación en una zona boscosa de Venezuela.

Existen muchos otros ejemplos donde se estudia algún fenómeno del medio ambiente utilizando la dinámica de sistemas; dentro de la herramienta encontrara un gran compendio de artículos que sirven como apoyo en el proceso de aprendizaje y muestran varias aplicaciones de la dinámica de sistemas en el estudio del medio ambiente.

3.2 MODELOS FORMALES Y REALIDAD VIRTUAL

“La sola necesidad de vivir en interacción con nuestro hábitat natural, exige modelar la realidad, al menos con modelos mentales, y cuando el hombre no tan sólo vive sino que busca explicaciones a los múltiples problemas que la vida misma le plantea y lucha por lograr adaptar su medio para bien de las futuras generaciones (aunque no siempre lo ha hecho para bien), debe pasar a la definición formal de sus modelos y, si fuese posible, a su formulación matemática.”¹⁶

Del anterior planteamiento se desprende la idea de modelo formal como la explicación e interpretación de los fenómenos mediante postulados estructurados en un todo definido claramente y compartible por una comunidad. Cuando dichos modelos están formulados con la rigurosidad y verificabilidad que la ciencia actual reconoce, se denominan conocimiento científico. Igualmente es de destacar que los modelos pueden tener diferentes niveles de formalización, siendo el matemático uno de los más rigurosos y aceptables por la comunidad científica actual.

Cabe resaltar la relación entre modelos mentales y modelos formales, más los segundos no existen sin los primeros y a todo modelo formal le antecede un modelo mental y contiene los modelos mentales de sus creadores.

Cuando mediante el uso de los modelos formales, es posible representar la realidad descrita o, más explícitamente, describir el comportamiento del fenómeno modelado, bajo diferentes condiciones; se dice que el modelo puede simular el fenómeno y a esta representación, con la ayuda del computador, se le denomina realidad virtual.

¹⁶ ANDRADE, Hugo ; SOTAQUIRA, Ricardo. Pensamiento de Sistemas y Dinámica de Sistemas para el modelamiento de fenómenos de diversa naturaleza. Grupo SIMON de Investigación 1997.

La idea de realidad virtual que aquí se considera, corresponde más al comportamiento simulado que a imágenes dibujadas y con comportamiento parecido a la realidad. El comportamiento se deriva del modelo matemático, mediante la solución del mismo para cada instante de tiempo, de tal forma que se pueda describir las trayectorias temporales de las variables contempladas en el modelo. La idea de realidad virtual puede asociarse a la de los dibujos, cuando la dinámica de comportamiento de dichos dibujos está determinada de manera directa por los modelos matemáticos que la describen y permite simular. Es decir se asume que a toda realidad virtual corresponden modelos matemáticos (entendido matemático no solo como ecuaciones, pueden ser reglas de movimiento, interacción, desarrollo, etc. y objetos que actúan en función de dichas reglas).

3.2.1 Procesos de aprendizaje.

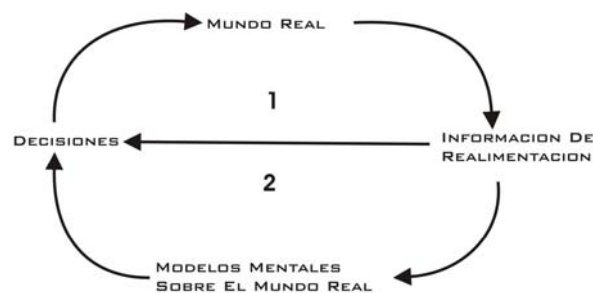
En este apartado se propone describir el proceso de aprendizaje mediante un sistema dinámico que integra el aprendizaje natural, el que se da actuando sobre la realidad misma, y el aprendizaje "artificial", el que se desarrolla sobre una realidad virtual. Finalmente se señala el rol del modelamiento, construcción de realidades virtuales, en la integración de los dos tipos de aprendizajes y garantía de un aprendizaje profundo.

- **Aprendizaje Natural.** Se asume aprendizaje como el proceso de transformación de los modelos mentales del aprendiz, modelos mentales que a su vez le orientan la comprensión y uso apropiado de los modelos formales. Es decir se asume que un conocimiento es propio del individuo cuando hace parte de sus modelos mentales y por ende guían su intervención en el mundo, con dichos modelos mentales y/o, además, con el apoyo de los modelos formales.

La figura 3 corresponde al sistema dinámico de aprendizaje natural, sistema, porque el proceso de aprendizaje se describe mediante cuatro componentes relacionados, y dinámico, porque las relaciones entre los elementos conforman una estructura de realimentación que genera dicha dinámica. Esta noción de aprendizaje, como ciclo de realimentación, la desarrolla Stermán en detalle con base en los planteamientos de importantes teóricos del aprendizaje organizacional como Argyris y Schon¹⁷.

En la estructura descrita por la figura 3 son visibles dos ciclos de realimentación, el ciclo 1 (mundo real -> información de realimentación -> decisiones -> mundo real) explica la dinámica de aprendizaje natural, superficial e inconsciente, fruto de una dinámica de prueba y error. En este tipo de aprendizaje la persona no ha definido explícitamente el modelo mental que refleja la idea del mundo real sobre el cual actúa y menos las ideas mediante las cuales percibe la información de realimentación para procesarla y tomar las decisiones pertinentes para afectar el mundo real u orientarlo a un comportamiento deseado, es decir actúa pero sin develar el modelo mental que posee sobre el mundo real.

Figura 3. Ciclos de aprendizaje "natural"



¹⁷ Stermán, John. 1994. Learning in and about complex systems. System Dynamics Review 10(2-3): 291-330.

El ciclo 2 de la misma figura, (mundo real -> información de realimentación -> modelos mentales sobre el mundo real -> decisiones -> mundo real...), describe la dinámica de aprendizaje natural, profunda o consciente. En esta dinámica los modelos mentales son explícitos y se van construyendo y reconstruyendo en el proceso de aprendizaje continuo del ser humano.

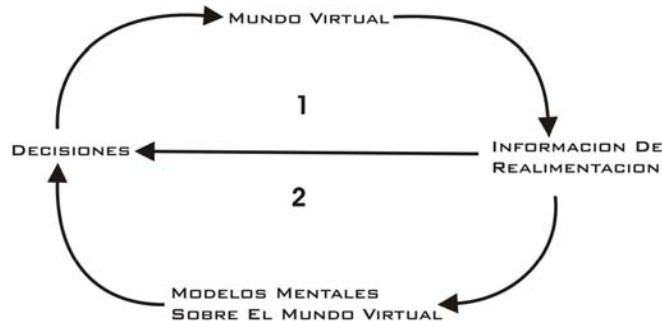
Cuando este ciclo se presenta, la persona posee, explícitamente definido, el modelo mental de la realidad que pretende explicar e intervenir. Este ciclo de la dinámica de aprendizaje es el que se esperaría promover en un proceso formal de aprendizaje profundo, y es el aprendizaje que posibilita los cambios más radicales, rápidos, reales y duraderos. Cuando actúa solo el ciclo 1 se corre el riesgo que la persona se acomode a una situación particular y actúe por dicho acomodo y no porque asuma para sí la nueva idea como la más apropiada.

Es de anotar que el sólo modelo mental ya explícitamente definido corresponde a cierto nivel de formalización, aunque sigue siendo modelo mental en la medida que corresponde al punto de vista del sujeto.

• **Aprendizaje Artificial.** El aprendizaje artificial sigue siendo tan real como el natural, es aprendizaje. El apellido de artificial no se le da por su carácter sino por el hecho de que se logra interactuando no sobre la realidad natural, sino sobre una realidad virtual (artificial).

La figura 4, explica el carácter artificial de este aprendizaje al mostrar cómo el proceso de experimentación se da sobre una realidad virtual y no sobre la realidad misma. Aquí se presentan los mismos dos ciclos descritos al plantear el aprendizaje natural.

Figura 4. Ciclos de aprendizaje "artificial"



El sólo hecho de experimentar sobre una realidad virtual no garantiza que se presente un aprendizaje profundo o de transformación de los modelos mentales del aprendiz, puede ser que usando la realidad virtual se oriente el proceso solo sobre el ciclo 1, en procura de un aprendizaje por prueba y error y uno que genere una visión profunda de la realidad en estudio, reflejada ésta en los modelos mentales explícitamente definidos.

• **Aprendizaje Artificial y modelamiento participativo.** Para garantizar que el proceso de aprendizaje "artificial" haga explícitos los modelos mentales que subyacen a las decisiones, o que describen y explican la dinámica de la realidad en estudio y orientan el proceso de adquisición y tratamiento de la información de realimentación para la toma de decisiones; se propone el recurso del modelamiento participativo; modelamiento, porque es la construcción de la realidad virtual mediante modelos formales y participativos, porque se prefiere en grupo y porque se desea resaltar la participación directa del aprendiz en dicha labor. La Figura 5 señala este modelamiento e integra el sistema de aprendizaje natural con el "artificial".

Figura 5. Aprendizaje y Modelamiento participativo



3.3 INFORMATICA Y LA CONSTRUCCION DE LA REALIDAD VIRTUAL

La ingeniería de los sistemas integra una serie de aportes filosóficos, de lo que se ha llamado teoría general de sistemas o pensamiento sistémico, y los aplica al tratamiento de complejidades con la ayuda de las matemáticas y de lenguajes y metodologías que reflejan su mayor potencialidad cuando se utilizan en entornos computacionales.

3.3.1 Modelamiento Matemático.

Ya antes en este documento se definieron los conceptos de modelo mental y modelo formal junto a la importancia de estos últimos en la construcción de las realidades virtuales, útiles informáticos de gran valor en la dinámica de aprendizaje "artificial". La informática desarrolla y aplica variados enfoques, lenguajes y metodologías de modelamiento.

El modelamiento matemático es orientado por dos enfoques fundamentales. El enfoque Conductista que intenta describir y explicar el comportamiento en función de él mismo, y el enfoque Estructuralista, el

cual asume una postura sistémica para describir y explicar el comportamiento de la realidad en función de la estructura de relaciones causales que conforman el sistema.

Existen diversos lenguajes y metodologías de enfoque Estructuralista, la que aquí se asume y que es coherente con los planteamientos hechos en todo este documento, es la dinámica de sistemas, lenguaje sistémico para asumir la realidad y metodología de modelamiento matemático que orienta la construcción de los modelos formales de simulación que fundamentan las realidades virtuales.

3.3.2 Modelamiento con Dinámica de Sistemas.

La dinámica de sistemas orienta el proceso de construcción de un modelo matemático estructural de un fenómeno, y posibilita simular su comportamiento dinámico en el transcurrir del tiempo, o de otra variable independiente. En general, el proceso de simulación proporciona el conjunto de valores de las variables en cada instante; esto es posible porque el modelo matemático y estructural involucra los elementos (variable y parámetros) fundamentales del fenómeno y las interacciones entre ellos. A su vez, las interacciones y las leyes que las rigen permiten determinar la variabilidad de cada variable en función de las demás; de los parámetros, del instante y de las condiciones iniciales; y, así, observar los efectos de la realimentación, base del comportamiento dinámico del fenómeno.

La dinámica de sistemas es una metodología que, inspirada en la teoría general de sistemas y en la teoría de los procesos de realimentación, la cibernética, guía mediante un conjunto de pasos bien definidos el proceso de construcción formal de modelos matemáticos. Este conjunto de pasos los resume el profesor Javier Aracil en su libro "Introducción a la Dinámica de Sistemas", así: "En primer lugar se observan los modos de

comportamiento del sistema real para tratar de identificar los elementos fundamentales del mismo; por ejemplo los síntomas de una perturbación. En segundo lugar, se buscan las estructuras de realimentación que puedan producir el comportamiento observado. En tercer lugar, a partir de la estructura identificada, se construye el modelo matemático de comportamiento del sistema en forma idónea para ser tratado sobre un computador. En cuarto lugar, el modelo se emplea para simular, como en un laboratorio, el comportamiento dinámico implícito en la estructura identificada. En quinto lugar, la estructura se modifica hasta que sus componentes y el comportamiento resultante coincidan con el comportamiento observado en el sistema real. Por último, en sexto lugar, se modifican las decisiones que puedan ser introducidas en el modelo de simulación hasta encontrar decisiones aceptables y utilizables que den lugar a un comportamiento real mejorado”.











Dos útiles son componentes fundamentales de la dinámica de sistemas, los diagramas causales, de los cuales ya se ha dado una idea al plantear la dinámica de aprendizaje, y los diagramas de Forrester o diagramas de flujos y niveles, los que corresponden a un modelo matemático basado en ecuaciones diferenciales lineales o no lineales.

Al hacer explícitas todas las ecuaciones que rigen las relaciones entre los diferentes elementos del diagrama de flujos-niveles, se obtiene un modelo matemático de simulación que, con la ayuda del computador, puede describir las trayectorias temporales de cada una de las variables.

Es justamente esta metodología la que se utiliza para modelar y simular los fenómenos ambientales que se encuentran como ejemplos dentro de la herramienta software; mas adelante en el apartado Metodológico se explicará con mayor profundidad todo el proceso para la elaboración de

los modelos. Los elementos fundamentales de los diagramas de flujos-niveles se describen en la tabla 2:

Tabla 2. Simbología utilizada en Diagramas de Flujos-Niveles.

	Nube: Representa una fuente o pozo. Se interpreta como un nivel inagotable y que no tiene interés
	Nivel: Es la variable de estado; representa una acumulación de flujos
	Flujo: Es la variación de un nivel; representa un cambio en el estado del sistema
	Canal de Material: Es la transmisión de una magnitud física que se conserva
	Canal de Información: Es la transmisión de información que no se necesita conservar
	Variable Auxiliar: Cantidad con cierto significado para el modelador (que no siempre tiene un significado físico en el mundo real) y con un tiempo de respuesta instantáneo
	Parámetro: Es un elemento del modelo independiente del sistema o una constante propia del sistema que no varía durante una corrida de simulación
	Variable Exógena: Variable cuya evolución es independiente de las del resto del sistema. Representa una acción del medio sobre el sistema
	Retardo: Es un elemento que simula retrasos en la transmisión de información o de material entre los elementos del sistema
	No-Linealidad : Representa una relación de no linealidad entre dos variables ¹⁸

3.3.3 Software de Simulación con Dinámica de Sistemas.

Los recursos computacionales duros, máquinas de cómputo, y los recursos blandos, software, hacen viable el proceso de modelamiento y simulación con dinámica de sistemas, destacando que en esta metodología el modelamiento utiliza la simulación como un soporte de modelamiento mismo, en la medida que orienta y permite validar los prototipos del modelo que se va construyendo.

¹⁸ El uso del símbolo empleado para representar las No-Linealidades, se sugiere en Martínez y Requena 1986, y no corresponde a la notación original de Forrester.

Entre mayores sean las capacidades del recurso computacional y más los servicios que brinde el software, mejor el modelador se podrá dedicar al componente creativo que exige el modelamiento y que fundamentalmente se encuentra en la tarea de hacer explícitos o construir y reconstruir los modelos mentales (fase de conceptualización); y a la tarea de formalización matemática .

La única herramienta software en español que, junto otras cuatro en inglés, soportan el proceso de modelamiento con dinámica de sistemas, se construyó en la Universidad Industrial de Santander, se denomina Evolución y actualmente circula la versión 3.5 beta 20¹⁹. Este software apoya fundamentalmente las fases de formalización y de simulación.

Para el caso en particular, la herramienta software para el estudio de fenómenos ambientales mediante el modelado y la simulación con dinámica de sistemas, reutiliza algunos componentes de software provenientes de la herramienta Evolución 3.5, de forma que dentro del software se tengan las facilidades básicas de modelado y simulación que ésta ofrece.

3.4 DINAMICA DE SISTEMAS: FORMA DE PENSAMIENTO PARA LA REPRESENTACIÓN Y EL APRENDIZAJE

El modo habitual como se ha entendido y se ha expuesto la Dinámica de Sistemas, es fundamentalmente, una metodología para construir modelos matemáticos de fenómenos de diversa índole; y aunque no se pretende dejar de un lado a la Dinámica de Sistemas como metodología, se quiere plantear también un modo de entender la Dinámica de sistemas como algo más que una metodología; se trata de verla como una forma o un

¹⁹ ANDRADE, Hugo; CUELLAR Mario; LINCE Emiliano. Herramienta informática desarrollada por el grupo SIMON de investigaciones, para el modelamiento y simulación con Dinámica de Sistemas. UIS, Bucaramanga 2004.

paradigma de pensamiento que se expresa a través de cierto sistema de convenciones, esto es, a través de un lenguaje particular. Esta perspectiva que concibe a la Dinámica de sistemas como una unidad paradigma-lenguaje, comprende en su interior la idea más limitada, aunque difundida, de la Dinámica de sistemas como metodología de modelado.

Hablar de un paradigma dinámico-sistémico es ambicioso, pues implica sugerir que se trata de una gran variedad de fenómenos complejos que pueden ser pensados según una "plantilla" muy particular: la "plantilla" de la Dinámica de Sistemas. Por supuesto, con esto no se está afirmando la primacía ni mucho menos la exclusividad, del paradigma dinámico – sistémico frente a otras formas de pensamiento, sistémicas o no. Se quiere mostrar que la pretensión es bastante amplia; podría decirse que es la pretensión de ver el mundo con los "ojos" de la Dinámica de Sistemas.

En la actualidad, esta manera de entender la Dinámica de Sistemas orientada hacia el aprendizaje, ha dado paso a nuevas herramientas computacionales basadas en modelos y conocidas como "micromundos". Se dejan entonces atrás los modelos de "caja negra" construidos por "especialistas en modelado", para prestarle atención al proceso de modelado como oportunidad para la construcción de conocimiento alrededor de un fenómeno, es decir, como oportunidad para el aprendizaje. Los modelos que aparecen en este caso – los micromundos – , son "cajas transparentes" y dentro de éstas se pueden apreciar los distintos conocimientos que las personas participantes tienen acerca del modelo, aunque no sean "especialistas en modelado".

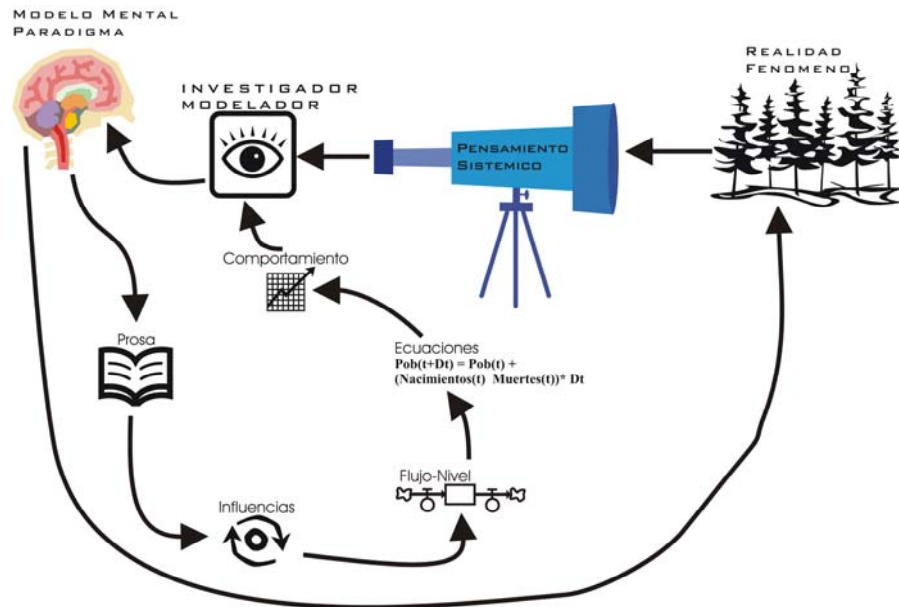
3.5 APRENDIZAJE DINAMICO-SITEMICO

3.5.1 Modelamiento.

Mediante la Dinámica de Sistemas se pretende ganar comprensión sobre la estructura causal compleja que subyace y determina la dinámica de comportamiento de los fenómenos. ¿Qué significa para el paradigma dinámico-sistémico el ganar comprensión acerca del fenómeno?

El estado de entendimiento de un fenómeno está representado por la imagen mental que de él se tiene; por el modelo mental que se hace de él. El modelo mental sobre un fenómeno particular está cambiando continuamente. Bien sea porque el contacto con el fenómeno proporciona nuevas percepciones y experiencias, o porque se reinterpretan experiencias y conceptualizaciones anteriores bajo una nueva luz. El modelo mental no solamente representa al fenómeno desde cierta perspectiva sino que también actúa como filtro en la relación con el fenómeno. Es decir, el modelo mental condiciona tanto las percepciones como las acciones sobre el fenómeno. De modo que la interacción entre el fenómeno y el modelo mental ocurre dentro de un ciclo de realimentación, como puede apreciarse en la figura 6.

Figura 6. Modelado con Dinámica de sistemas, usando la perspectiva del pensamiento sistémico para observar la realidad



En estos términos, ganar comprensión dinámico-sistémica sobre el fenómeno se entiende como un proceso en el cual un modelo mental se va modificando para incorporar gradualmente una representación de la estructura causal compleja que pueda explicar el comportamiento del fenómeno.

En consecuencia, con la Dinámica de Sistemas se pretende realizar un proceso dirigido de reformulación del modelo mental, de comprensión del fenómeno. Es un proceso dirigido por contraste con el proceso natural, descrito anteriormente, mediante el cual un modelo mental va cambiando. Pero la reformulación dirigida, a diferencia de la transformación natural, se enfrenta con una dificultad. El modelo mental sobre un fenómeno nunca se presenta frente a cada cual, es decir, permanece siempre oculto. El modelo mental interviene, por su papel de filtro, en cualquier interacción con el fenómeno: cuando se experimenta,

se reflexiona y/o se decide una acción sobre el fenómeno. Pero en su intervención permanece detrás de la interacción. En condiciones normales, no es posible ver el filtro a través del cual se está viendo el fenómeno.

Esta invisibilidad del modelo mental no significa problema en el proceso natural, pero si lo es para el proceso dirigido mediante la Dinámica de Sistemas. Para modificar intencionalmente un modelo mental es necesario tenerlo en alguna expresión visible. Es necesario tener una representación, un modelo del modelo mental. La Dinámica de Sistemas ofrece herramientas metodológicas para expresar el modelo mental mediante un modelo visible, con lo cual es posible desarrollar el proceso dirigido de reformulación del modelo mental hacia una mejor comprensión dinámico-sistémica del fenómeno. En consecuencia, el paradigma dinámico-sistémico es un paradigma de modelamiento, entendido el modelamiento como la actividad de modelar los modelos mentales de los fenómenos, como el re-representar del fenómeno.

De esta manera, el modelo expresado con los útiles dinámico-sistémicos, que como se verá más adelante, son lenguajes para representar la estructura causal, se convierte en un objeto mediador entre la interacción y la comprensión. El modelo dinámico-sistémico liga el fenómeno con el modelo mental sobre el fenómeno. Y la forma de esta ligazón, su estructura causal, es la de un doble ciclo de realimentación, como se muestra en la figura 6. El ciclo externo, entre el fenómeno y el modelo mental, ya había sido descrito. Mientras que el ciclo interno muestra el proceso mediante el cual se puede establecer una interacción con el modelo mental, visibilizado a través del modelo, o los modelos, dinámico-sistémico. Este último sirve de pantalla sobre la cual se proyecta un modelo mental, con lo cual se logra tener tanto al fenómeno como a esa versión aproximada del modelo mental, lo que hace posible su

comparación y, por lo tanto, hace posible mejorar la comprensión del fenómeno.

3.5.2 Interacción simulada y mundos virtuales.

El propósito del estudio dinámico-sistémico de fenómenos consiste en hacerse con un modelo mental desde el cual se pueda entender la dinámica de comportamiento mediante su explicación causal. Es a la luz de este propósito que tiene sentido la construcción de modelos dinámico-sistémicos que ligan el fenómeno con el modelo mental. Para satisfacer esta intención es necesario que los modelos dinámico-sistémicos representen esa relación esencial del fenómeno entre su estructura causal y su dinámica. Los modelos dinámico-sistémicos tienen entonces la forma de hipótesis causales que explican un comportamiento. Es decir, es posible preguntarle a un modelo dinámico-sistémico ¿porqué sucede tal o cual comportamiento? o ¿cuál podría ser el comportamiento si se dan estas condiciones?. En ambos casos la respuesta explicativa del modelo vendrá dada en términos de causalidades circulares.

Esta forma particular que toman los modelos dinámico-sistémicos permite que se pueda hacer con ellos una interacción simulada. De modo análogo como se sucede una interacción con el fenómeno, es posible interactuar con el modelo dinámico-sistémico. Es decir, es posible determinar acciones sobre el modelo y observar sus consecuencias dinámicas bajo ciertas condiciones supuestas -bajo determinado escenario-, debido a la capacidad de respuesta explicativa del modelo.

Una nueva mirada de la figura 6 revela, ahora, que el estudio de fenómenos en el paradigma dinámico-sistémico sucede en una interacción, mediada por el modelo mental, con un mundo real (el fenómeno), y además, con un mundo virtual (el modelo dinámico-sistémico). Podría entenderse el papel de la Dinámica de Sistemas como

el de un medio para la construcción de mundos virtuales con los cuales se pueda establecer una interacción simulada que ayude a comprender mejor el mundo real.

3.5.3 El doble ciclo de aprendizaje: paradigma y lenguaje.

Dicho de otro modo, el doble ciclo representa un proceso inagotable de aprendizaje dinámico-sistémico sobre el fenómeno en cuestión. Pero no solamente se está sucediendo un aprendizaje como modificación del modelo mental sobre el fenómeno. La figura 6 ilustra que este modelo mental está situado sobre un paradigma de pensamiento. Así en la medida en que un modelo mental va adquiriendo mayor riqueza respecto de la comprensión dinámico-sistémica del fenómeno, lo que en un plano más profundo va sucediendo es que se está experimentando un aprendizaje del paradigma dinámico-sistémico. Dicho de otra manera, el uso que se hace del sistema de lenguajes dinámico-sistémicos en la actividad de modelamiento va produciendo una apropiación progresiva del paradigma dinámico-sistémico. De este modo, lenguaje y paradigma en la Dinámica de Sistemas (DS) constituyen las dos caras de una moneda que se va reforzando gradualmente en la medida en que se mira y estudia el mundo dinámico-sistémicamente²⁰.

3.6 EL LENGUAJE DINAMICO-SISTEMICO

3.6.1 Sistema de Lenguajes.

Mediante la Dinámica de Sistemas los modelos mentales son visibilizados en forma de hipótesis estructurales causales del comportamiento del objeto de estudio. ¿Cuáles son los útiles metodológicos que ofrece para este modelamiento? La Dinámica de Sistemas proporciona un sistema de

²⁰ Cabe resaltar entonces que el simple uso de útiles de la Dinámica de Sistemas, como por ejemplo de los diagramas causales, no indica que se este haciendo un estudio dinámico-sistémico del fenómeno en cuestión.

lenguajes con los cuales es posible expresar la causalidad circular que puede subyacer en el fenómeno.

Los modelos dinámico-sistémicos son escritos en esos lenguajes. La figura 6 muestra los diferentes tipos de lenguajes: el lenguaje de prosa, el lenguaje de los diagramas causales, el lenguaje de los diagramas de niveles y flujos, el lenguaje matemático y el lenguaje visual computacional. Cada uno de ellos satisface de manera particular los requerimientos de un modelo dinámico-sistémico, esto es, la posibilidad de expresar hipótesis causales de la dinámica y la posibilidad de realizar con ellas una interacción simulada.

Mediante el lenguaje matemático puede representarse la estructura causal como un sistema de ecuaciones diferenciales ordinarias, generalmente no-lineales. Este sistema puede resolverse mediante su simulación por computador, transformándolo en un sistema de ecuaciones en diferencias. Con la representación matemática del sistema y haciendo uso de herramientas software especializadas se obtiene entonces, un modelo simulable en computador con amplias facilidades para la interacción en un lenguaje visual de definición de escenarios de simulación y presentación de resultados numéricos y gráficos.

Los lenguajes de diagramas causales y diagramas de niveles y flujos, como lo indican sus denominaciones, tienen un carácter gráfico. En cada uno de ellos se puede dibujar y apreciar visualmente la estructura causal con sus ciclos de realimentación. Para cada uno de estos diagramas existe una lógica que permite inferir comportamientos posibles del sistema a partir de las figuras que allí aparecen, de modo que se pueden realizar simulaciones "mentales".

Finalmente, en el lenguaje de prosa, que se utiliza cotidianamente, es posible hablar en los términos de las causalidades y cómo ellas determinan el comportamiento. El lenguaje de prosa permite comunicar fácilmente los supuestos de los modelos dinámico-sistémicos expresados en los otros lenguajes. Su papel no es el de facilitar una interacción simulada para la cual son mejores los otros lenguajes. Esto último es precisamente una característica general de los lenguajes de la Dinámica de Sistemas: cada uno de ellos satisface ciertas necesidades del modelador.

Mientras el lenguaje de prosa cumple un papel central en la posibilidad de comunicación y divulgación de los estudios con Dinámica de Sistemas; los lenguajes gráficos son instrumentos valiosos para lograr la comprensión de la complejidad estructural del fenómeno; y el lenguaje matemático conduce a la formulación de modelos sobre los cuales es posible elaborar construcciones en software para la interacción simulada con la cual se prueban alternativas de políticas de acción sobre el sistema. La variedad de lenguajes permite que la tarea de comprensión dinámico-sistémica del fenómeno se enriquezca de diferente modo con cada representación.

Ahora se verá con mayor detalle en qué consisten las diferencias de expresión y representación de cada uno de estos lenguajes. El lenguaje de prosa no es exclusivo de la Dinámica de Sistemas y ofrece una riqueza amplísima para expresar los supuestos causales de los modelos mentales. Sin embargo, y debido nuevamente a que es un lenguaje de propósito general, no ofrece un orden fijo, una gramática exclusiva, para representar las estructuras causales. Esto lo hace inapropiado para escribir redes causales muy complejas, y también para inferir posibles comportamientos generados por esas redes.

Los diagramas causales y los diagramas de niveles y flujos son, en cambio, lenguajes específicos de la Dinámica de Sistemas. Como ya se dijo, es posible con ellos dibujar la estructura causal. En el diagrama causal, o de influencias, claramente se expresan las relaciones de causa - efecto y los ciclos de realimentación. También es posible indicar el sentido de cambio que un elemento produce sobre otro y, en consecuencia, es posible señalar cuando un ciclo de realimentación es positivo, amplificador, o negativo, atenuador. La visión que ofrece un modelo expresado como diagrama causal tiene cualidades didácticas por la que es útil, además, para comunicar y discutir, aun con personas ajenas a la Dinámica de Sistemas, las hipótesis causales.

El diagrama de niveles o flujos, o de Forrester²¹, ofrece una mirada distinta de la estructura causal. Su lógica puede entenderse mediante una metáfora hidrodinámica. Un sistema hidrodinámico muy simple puede estar compuesto por un tanque que almacena líquido y un par de llaves o grifos que permite su llenado o su vaciado. En este sistema, el cambio es producido por la apertura o el cierre de las llaves; y el cambio genera un aumento o disminución en el nivel de líquido en el tanque, es decir, el estado de este sistema es representado por el valor actual de ese nivel. Metafóricamente, un fenómeno particular puede ser entendido en términos de su estado, sus niveles, y de aquello que produce su cambio, sus variables de flujo. El diagrama de niveles y flujos sirve de este modo para ilustrar el modo como los niveles del sistema cambian en virtud de los flujos que los afectan.

El diagrama de Forrester ofrece una mirada distinta de la estructura causal en la que prima la distinción entre estado y cambio, que no era evidente en el diagrama causal. Por consiguiente, este lenguaje ofrece

²¹ Esta denominación hace referencia al creador de la Dinámica de Sistemas, el profesor Jay W. Forrester.

una representación más estricta de las hipótesis causales y una fidelidad mayor en términos de las posibilidades de simulación del modelo. De hecho, el diagrama de Forrester puede ser entendido como el esqueleto del modelo matemático.

El lenguaje matemático, que tampoco es exclusivo de la Dinámica de Sistemas, ofrece una posibilidad de expresión ordenada del modelo. No es muy útil para "ver" la estructura causal, pero abre las puertas para expresar finalmente el modelo en un lenguaje computacional de modo que se pueda interactuar con él a través de simulaciones.

En conclusión, los diferentes lenguajes ofrecen posibilidades que son complementarias. Por esta razón conviene entenderlos como un sistema, que en su conjunto hace posible estudiar dinámico-sistémicamente un fenómeno mediante modelos. Su carácter de sistema también radica en que los lenguajes están imbricados en diferentes niveles de abstracción, desde el más bajo, en el lenguaje de prosa, pasando por el diagrama causal y luego el diagrama de niveles y flujos, hasta el más abstracto, el lenguaje matemático. De modo que entre un lenguaje y el siguiente, en este orden del sistema de lenguajes, hay un cambio de plano de abstracción y esta es otra razón de su riqueza expresiva para el estudio de la causalidad que genera el comportamiento de un fenómeno.

3.6.2 Prototipos de Complejidad creciente.

Volviendo al concepto de causalidad unidireccional se puede entender, ahora, que el estudio dinámico-sistémico de un fenómeno consiste en un iterar repetidamente en ese proceso de doble ciclo de realimentación. Esto implica que la comprensión de la complejidad se hace de manera gradual, progresiva. Esto es posible gracias a que el sistema de lenguajes permite no solamente navegar entre distintos niveles de abstracción y de expresión de la causalidad del sistema; sino que cada vez que se revisita

un lenguaje, es posible extender el modelo dinámico-sistémico que se ha construido con él. Es decir, se puede ver el proceso iterativo como la construcción de prototipos, de versiones de modelos que se van a mejorar en una siguiente iteración. Cada pasada por los lenguajes, con cada prototipo, va arrojando cada vez mejor la complejidad del fenómeno permitiendo su comprensión gradual.

3.7 FORMAS METODOLOGICAS DE LA DINAMICA DE SISTEMAS

El sistema de lenguajes de la DS ofrece al modelador un medio versátil para estudiar un fenómeno. Este atributo de versatilidad abre la posibilidad a diversas formas metodológicas de realizar un estudio dinámico-sistémico. En la literatura de la comunidad de DS se describen numerosas aplicaciones que emplean de distinta manera el sistema de lenguajes. Con una intención pedagógica, se va a caracterizar solamente dos formas metodológicas porque, en buena medida, son las más ampliamente utilizadas. Para distinguir las dos formas metodológicas conviene retornar al concepto de causalidad unidireccional, que ubica a la Dinámica de Sistemas como animadora de un proceso dirigido de interacción con el mundo, con un fenómeno particular. Una interacción que comprende las actividades de conocimiento y acción sobre el fenómeno, medidas por los modelos mentales. La dinámica de interacción, definida por el doble ciclo de realimentación, genera cambios en el fenómeno, en los modelos mentales y, por supuesto, en los modelos dinámico-sistémicos. Es posible imaginar un tipo de estudio que preste interés, casi exclusivo, al logro de un cambio en el fenómeno. Por contraste, se puede pensar en otro tipo de estudio animado principalmente por la modificación de modelos mentales. Así, se tendrían dos posibles formas metodológicas, que emplean de modo distinto el sistema de lenguajes para estimular la dinámica de transformación bien sea del fenómeno o bien sea de los modelos mentales. A la primera se le

denominará *metodología para la intervención*, y a la segunda *metodología para el aprendizaje*²².

3.7.1 Metodología para la intervención.

Esta primera forma metodológica está orientada a la transformación del comportamiento actual del fenómeno. En consecuencia, conviene pensar que este propósito se puede conseguir en dos fases. Una de comprensión dinámico-sistémica del fenómeno y otra de diseño de la intervención necesaria para modificarlo.

La *metodología para la intervención* ordena de un modo particular el sistema de lenguajes dinámico-sistémicos para lograr un cambio del fenómeno hacia cierta forma deseada de comportamiento. Los modelos dinámico-sistémicos tratan de ser representaciones fieles de la estructura causal que rige al fenómeno, para que mediante la simulación por computador sea posible hacer el análisis prospectivo de alternativas de intervención. En esta variante metodológica los modelos con DS se entienden más como representaciones del fenómeno que como expresiones de modelos mentales.

3.7.2 Metodología para el aprendizaje.

A través de todo el capítulo se ha tratado de ver al paradigma dinámico-sistémico como adecuado y necesario para comprender la complejidad dinámica de los fenómenos y más específicamente de aquellos pertenecientes al medio ambiente. En la medida en que se interiorice este paradigma, la percepción del mundo va a cambiar, porque esta percepción siempre está mediada por los modelos mentales, y de este modo toda interacción con el mundo se transforma. Bajo estas premisas

²² Las formas metodológicas expuestas están basados en dos estilos de praxis dinámico-sistémica reseñados en Sotaquirá, Gélvez, Cabrera "Hacia un mirar latinoamericano de la aplicación de la Dinámica de Sistemas en estudios organizacionales".

la *metodología para el aprendizaje* concluye que el cambio más profundo es aquel que sucede inicialmente en la manera de pensar acerca del mundo, es el cambio movido por el aprendizaje de un nuevo paradigma de pensamiento, el aprendizaje del paradigma dinámico-sistémico.

Pero como se explicó antes, el aprendizaje del paradigma de la DS es un resultado emergente de una dinámica de modelamiento en la cual se va usando el sistema de lenguajes para elaborar representaciones, y de este modo poner explícitos, los modelos mentales que se tienen. Es decir, el afianzamiento del paradigma sólo puede lograrse mediante la realización continua de ejercicios de modelamiento con DS, mediante la práctica permanente de destape y modificación de los modelos mentales con DS. De este modo la *metodología para el aprendizaje* hace énfasis en el cambio de los modelos mentales lo cual traerá como efecto el cambio en el fenómeno. Los modelos dinámico-sistémicos se entienden como expresiones de los modelos mentales.

Esta segunda variante metodológica se ha desarrollado principalmente en los dominios de la educación y de las organizaciones humanas. En este último, la DS ha sido integrada en el marco del aprendizaje organizacional (Senge; Morecroft y otros).

3.8 PLATAFORMAS DE DINAMICA DE SISTEMAS PARA EL APRENDIZAJE

3.8.1 Micromundos.

También denominados juegos de simulación, simuladores o plataformas para el aprendizaje, los cuales pueden ser utilizados como instrumentos de apoyo al aprendizaje de cualquier fenómeno que se quiera.

Los micromundos, son modelos de simulación que representan, en términos generales, aspectos de un sistema. Son 'juegos' que apoyan el aprendizaje organizacional, simulando las consecuencias de las decisiones que pueden tomarse dentro de las mismas. Estos son desarrollados para adquirir conocimiento a través de vivencias, en la forma como lo señala en antiguo proverbio chino:

"Cuando escucho, olvido; cuando veo, recuerdo; pero cuando hago, entiendo".

En estas circunstancias, no se pretende que el proceso de aprendizaje ocurra a través de la construcción de plataformas computacionales, sino a través de la operación de modelos explícitamente diseñados para reproducir cierto comportamiento que se observa en la vida real.

Los Micromundos se caracterizan por ser:

- **Laboratorios para el aprendizaje.** Procesos integrados de reflexión, conceptualización, simulación y juego, enfocados hacia el aprendizaje.
- **Modelos para el aprendizaje, no para convencer.** En situaciones en donde se revelan serias inconsistencias y contradicciones, con respecto a políticas operativas establecidas para el manejo de un sistema, organización o problema específico, la única estrategia posible a juicio de Senge, es la creación de modelos de simulación que apoyen el proceso de aprendizaje.
- **Entornos para la reflexión.** Facilitan procesos de: a) Conceptualización (creación de mapas causales), b) Reflexión (la sola simulación de la toma de decisiones no garantiza el aprendizaje), y c)

Cálculo (debe introducirse el computador cuando se ha entendido el propósito real del aprendizaje).

• **Entornos para el entendimiento de comportamientos dinámicos.**

Los modelos de la dinámica de sistemas contribuyen a la comprensión de la complejidad dinámica del fenómeno en estudio.

Con estos elementos se procede a establecer algunas de las características que deben poseer los micromundos para el adecuado apoyo a los proceso de aprendizaje. Estas son: a) que la historia a la cual se refiera sea fácil de entender, b) que el problema planteado tenga elementos de realidad y c) que la plataforma computacional sea amigable con el usuario.

3.8.2 Plataformas de Dinámica de sistemas.

Las plataformas de dinámica de sistemas corresponden a las plataformas computacionales que nos permiten el aprendizaje de la metodología como tal (de la dinámica de sistemas) y además el aprendizaje de algún tema o campo de estudio específico utilizando la Dinámica de Sistemas, permitiendo elaborar modelos con dinámica de sistemas y además manipular y jugar con esos modelos elaborados, permiten ejecutar todo el proceso de modelado utilizando específicamente el lenguaje de Flujos y Niveles y además poseen características adicionales que permiten el aprendizaje del proceso de modelado con Dinámica de Sistemas (Textos, tutoriales, elementos interactivos, etc.)

En resumen se puede decir que las plataformas de dinámica de sistemas poseen las características del software de simulación con dinámica de sistemas y además tienen elementos de los llamados micromundos, con el componente adicional educativo, el cual ayuda al aprendizaje tanto de la Dinámica de Sistemas como al proceso de aprendizaje del tema

especifico que se esté tratando en la plataforma p. ej.(plataforma de dinámica de sistemas para el estudio de las matemáticas, la biología, el medio ambiente, etc.).

3.9 INFORMATICA EDUCATIVA

Brevemente se puede conceptualizar la informática educativa como: Disciplina orientada a racionalizar y mejorar los procesos educativos mediante la sistematización de esfuerzos y aplicación de principios de ciencias de la educación. Significa la aplicación de utilitarios como herramientas de trabajo y la resolución de problemas reales para la creación de nuevas estructuras cognitivas del aprendiz.

3.9.1 Efectos de la Informática educativa.

El papel principal de la informática educativa es servir de estímulo a los sentidos del sujeto o alumno posibilitando su aprendizaje. El docente debe verla como una herramienta para mejorar el proceso de enseñanza por algunas razones:

- La creación del descubrimiento de principios y reglas básicas de inferencia y deducción. Esto permite aprender conceptos básicos que pueden ser transferidos a situaciones reales.
- Permite flexibilidad y adaptabilidad a los ritmos, intereses y posibilidades de los estudiantes.
- La destreza para la planificación de estrategias de resolución de problemas por ambas partes: docentes y alumnos.
- Se estimulan nuevas habilidades del pensamiento y la capacidad de descubrir por si mismo los conocimientos, incentivando la cooperación y la colaboración.
- Permite elaborar materiales de clase y tareas académicas. La posibilidad de auto elaboración de materiales es más cómoda y posible

gracias a la estructura abierta de los programas, a la facilidad para tratar datos de cualquier naturaleza o formato, y a la disponibilidad de recursos en la red mundial de información.

- La informática educativa fomenta la producción creativa en el alumno, motivándolo a adquirir nuevas estructuras cognitivas como producto de la resolución de necesidades reales. La disposición de materiales con animaciones, con formatos diversos torna ameno el aprendizaje.
- Puede usarse la programación para generar herramientas de trabajo (que presupone un alto grado de dominio de lenguajes de programación) o bien emplear los llamados lenguajes de autor. Estos últimos están pensados para el desarrollo rápido y simple de aplicaciones multimedia o simplemente aplicaciones de datos, sonido o video por separado.
- Dicha simplificación de procedimientos permite orientar el trabajo a la publicación electrónica. Por ejemplo Anet Help Tool de Anetsoft es una aplicación diseñada para que el docente y el alumno acopie y produzcan contenidos, desarrollen publicaciones electrónicas capaces de interactuar con el lector en ordenadores individuales, en redes locales o en Internet.
- Otro ejemplo del uso de lenguajes de autor es la producción de libros electrónicos mediante Acrobat de Adobe System. Los libros electrónicos pueden acopiarse en bancos de datos adecuadamente indexados, accesibles en forma individual o en red.
- En general es posible pensar que la informática educativa permite emplear utilitarios como herramientas de trabajo y la resolución de problemas reales para la creación de nuevas estructuras cognitivas de los educandos.
- Con las herramientas de autor, docente y estudiantes pueden producir libros electrónicos, libros multimedia, revistas interactivas, presentaciones, bancos de datos, programas didácticos, curriculum electrónicos, etc. Además los estudiantes se ven en la necesidad de aprender aspectos varios de programas utilitarios (como Word, Ami Pro,

Lotus, Excel, etc.) e informática en general (incluidos software y hardware). Esa necesidad de aprender surge de si mismo, y su aprendizaje se torna más ameno.

➤ La forma de trabajo es en equipo, interactuando con el medio, los recursos informáticos y el profesor. Las clases son prácticas al usar la informática educativa, y los aporte teóricos se realizan de manera colectiva, procurando que todos los participantes aporten sus preguntas y soluciones.

3.9.2 Enseñanza Asistida por Ordenador (E.A.O.).

La utilización inicial de la idea de E.A.O. como utilización de rutinas o de programas de tipo tutorial se ha quedado estrecha. Dentro de la educación se han ido haciendo y documentando muchas experiencias con otro tipo de materiales a lo largo de los últimos años (herramientas, aplicaciones, etc.) y el panorama se ha enriquecido notablemente: simulaciones, proceso de textos, gestores de bases de datos, gráficos, programas de diseño gráfico, hojas de cálculo, juegos educativos... Se ha alcanzado un nuevo significado del concepto de Enseñanza Asistida por Ordenador.

Una definición válida de E.A.O., por tanto, sería la siguiente: *"modalidad de comunicación indirecta entre alumno y profesor, que no se realiza por la línea más corta de la presencia física, sino describiendo un ángulo con un vértice en el ordenador."*

Otra definición muy interesante es la siguiente: **Enseñanza asistida por ordenador o computadora (EAO)**, es un tipo de programa educativo diseñado para servir como herramienta de aprendizaje (en inglés, *Computer-Aided Instruction* o *Computer-Assisted Instruction*, CAI). Los programas EAO utilizan ejercicios y sesiones de preguntas y respuestas para presentar un tema y verificar su comprensión por parte del

estudiante, permitiéndole también estudiar a su propio ritmo. Los temas y la complejidad van desde aritmética para principiantes hasta matemáticas avanzadas, ciencia, historia, estudios de informática y materias especializadas. EAO es sólo uno de la multitud de términos, la mayoría con significados equivalentes, relacionados con uso de las computadoras en la enseñanza. Otras expresiones son aprendizaje asistido por computadora, aprendizaje impulsado por computadora, aprendizaje basado en computadora, formación basada en ordenador o computadora e instrucción administrada por computadora.

3.9.3 Herramientas de Autor.

Hay dos tipos de usuarios en un sistema hipermedial: el autor y el lector. El autor es el escritor del hiperdocumento, es decir el que lo crea, establece sus relaciones y determina cómo va a ser la interacción con el lector. La información debe estar dividida en fragmentos que se encuentren relacionados entre sí y que sean significativos para el lector.

Las herramientas de autor tienen la finalidad de servir como elemento de escritura y de edición para sus autores, proporcionando lenguajes visuales de comunicación, en los que la manipulación directa de objetos, las cajas de diálogo y los menús desplegables son las formas más usuales de interacción hombre-máquina.

Un hipertexto puede ser el resultado de dos procesos diferentes: la creación a partir de diversos materiales (caso ideal) o la conversión de texto lineal en hipertexto.

Haciendo énfasis en el primer caso hay que decir que el papel del autor de un hipertexto es diferente al de un escritor tradicional, ya que aquél pierde parcialmente su autoridad para determinar cómo debe leerse su obra, qué secuencia debe seguirse para alcanzar un determinado tema. El

principal problema de la creación de un hipertexto reside en definir estructuras de texto completamente nuevas.

Las principales acciones que un autor debe llevar a cabo para crear un documento son:

- ✓ Preparación del material multimedia que formará parte del hiperdocumento.
- ✓ Organización de la información.
- ✓ Desarrollo de la aplicación.
- ✓ Definición de estructuras auxiliares, como por ejemplo, navegadores gráficos o índices que faciliten el uso y consulta del hiperdocumento.

Aunque no existen estándares aceptados, se pueden proponer algunas recomendaciones generales para los autores de hipertextos:

- ✓ Cada nodo debe ser autocontenido, esto es, dedicado a un único tema.
- ✓ La estructura de relaciones debe ser limpia (deben evitarse, p. ej. conexiones entre términos remotamente relacionados).
- ✓ Los nombres de nodos y enlaces deben ser significativos, a fin de que la red de información se haga explícita a sus lectores.

• **Requisitos de una herramienta de autor.**

- ✓ Debe proporcionar mecanismos para preparar y manipular el material multimedia.
- ✓ Debe posibilitar la creación y edición de nodos, así como su interconexión.
- ✓ Debe permitir la inclusión de objetos de información en los nodos que puedan actuar como anclas de los enlaces y que sean capaces de reaccionar

ante eventos (exteriores como p.ej. el movimiento del ratón, o interiores, p. ej. un tiempo desde el inicio de la sesión).

✓ Deben disponer de funciones con las que poder construir de forma sencilla, navegadores o mecanismos de ayuda a la navegación, así como comandos para la búsqueda de información.

3.9.4 Guías de Autoaprendizaje Apoyadas en Dinámica de Sistemas.

Las guías de autoaprendizaje, surgen como un híbrido entre los conceptos de A.A.O., las herramientas de autor, el concepto de hipermedia y las plataformas de dinámica de sistemas; pues poseen características de los programas que orientan el aprendizaje con ayuda del ordenador y además, poseen las características de una herramienta de autor pues permiten la creación y organización de documentos interactivos o multimediales para su posterior estudio, además de facilitar el enlace entre estos es decir la vinculación de un artículo con los demás; de tal forma que a diferencia de la enseñanza asistida por ordenador, no es necesario que los contenidos sean creados por un profesor; mas bien su objetivo es que el mismo aprendiz sea el que genere sus propias guías de estudio interactivas; además tiene una orientación hacia la inclusión de la dinámica de sistemas en el proceso de aprendizaje, pues permiten la utilización de entornos para el modelado con dinámica de sistemas dentro de los documentos interactivos.

3.10 ALGUNOS CONCEPTOS DE DESARROLLO Y CONSTRUCCION DE SOFTWARE

Ya se han observado los conceptos que enmarcan el tipo de software al cual pertenece la herramienta desarrollada (Posee las características de una Guía de autoaprendizaje apoyada en dinámica de sistemas); ahora se mencionan algunos conceptos básicos que rodean el proceso de su diseño y construcción.

3.10.1 El proceso unificado de desarrollo de software.

Es un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. El proceso está regido por tres características: está dirigido por casos de uso, centrado en la arquitectura, y es iterativo e incremental.

Dirigido por casos de uso: Un caso de uso representa un requisito funcional, que contesta a la siguiente pregunta: ¿Qué debe hacer el sistema para cada usuario? Un caso de uso es una descripción de un conjunto de secuencias de acciones que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado, donde un actor puede ser un usuario, un sistema o un rol.

Centrado en la arquitectura: Es un conjunto de decisiones acerca de la organización de un sistema software, de la selección de elementos y de las interfaces y las colaboraciones entre ellos. Se ve influida por varios factores como la plataforma en la que tiene que funcionar el software (hardware, sistema operativo, gestión de base de datos, protocolos de red), entre otros. Además, se relaciona con los casos de uso, para entender esto, se puede hacer una comparación entre función y forma, donde la función corresponde a los casos de uso y la forma a la arquitectura.

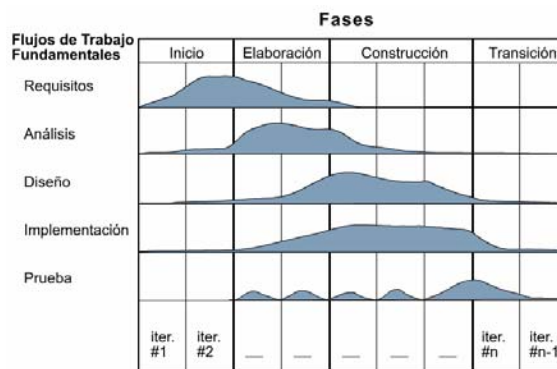
Es iterativo e incremental: Es iterativo porque el proyecto se puede dividir en mini proyectos, que es una iteración que resulta en un incremento, haciendo referencia a pasos en el flujo de trabajo y es incremental porque hay un crecimiento del producto. En cada iteración, se identifican y se especifican los casos de uso relevantes, se crea un diseño utilizando la arquitectura seleccionada como guía, se implementa el diseño mediante componentes y se verifica que los componentes satisfacen los casos de uso. Si se cumple con los objetivos se continua con la siguiente iteración,

en caso contrario, se revisan las decisiones previas y se prueba un nuevo enfoque. Los mini proyectos comienzan con los casos de uso, continua a través de un pequeño ciclo de vida en cascada -análisis, diseño, implementación y prueba- y los casos de uso que se desarrollaban en cada iteración terminan convirtiéndose en código ejecutable.

Un desarrollo iterativo, guiado por los casos de uso y centrado en la arquitectura, construye un software mediante pequeños incrementos, y añade cada incremento a la acumulación previa de incrementos de tal forma que siempre se tenga una construcción ejecutable. La arquitectura proporciona la estructura sobre la cual girar las iteraciones mientras que los casos de uso definen los objetivos y dirigen el trabajo en cada iteración.

De esta manera el proceso reduce el riesgo de grandes retrasos en la entrega de un producto, se fijan metas más inmediatas por lo cual se puede controlar mejor el avance del proyecto.

Figura 7. Los flujos de trabajo, tienen lugar en las cuatro fases expuestas en el proceso unificado



El proceso unificado se repite a lo largo de una serie de ciclos, cada ciclo concluye con una versión del producto y a su vez cada ciclo consta de

cuatro fases: Inicio, Elaboración, Construcción y transición. Cada una de estas fases se dividen a su vez en iteraciones.

En cada iteración, se identifica y especifican los casos de uso relevantes, se crea un diseño utilizando la arquitectura seleccionada como guía, se implementa el diseño mediante componentes, y verifica que los componentes satisfacen los casos de uso.

3.10.2 Construcción de software orientado a objetos.

Es el método de desarrollo de software que basa la arquitectura de cualquier sistema software en módulos deducidos de los tipos de objetos que manipula (en lugar de basarse en la función. o funciones a las que el sistema esta destinado a asegurar).

3.10.3 Programación orientada a objetos.

Grady Booch, autor del método de diseño orientado a objetos, define a la POO como:

“Un método de implementación en el que los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representan una instancia de alguna clase, y cuyas clases son todas miembros de una jerarquía de clases unidas mediante relaciones de herencia”.

Existen tres importantes partes en la definición: la programación orientada a objetos 1) Utiliza objetos no algorítmicos, como bloques de construcción (jerarquía de objetos); 2) cada objeto es una instancia de una clase, y 3) las clases se relacionan unas con otras por medio de relaciones de herencia.

Una programación puede parecer orientada a objetos, pero si cualquiera de estos elementos no existe, no es un programa orientado a objetos. Específicamente, la programación sin herencia es distinta de la programación orientada a objetos; se denomina programación con tipos abstractos de datos.

El concepto de objeto, al igual que los tipos abstractos de datos o tipos definidos por el usuario, es una colección de elementos de datos, junto con las funciones asociadas utilizadas para operar sobre estos datos. Sin embargo, la potencia real de los objetos reside en el modo como los objetos pueden definir otros objetos. Este proceso se denomina herencia y es el mecanismo que ayuda a construir programas que se modifican fácilmente y se adaptan a aplicaciones diferentes.

- **El Objeto.** La idea fundamental en los lenguajes orientados a objetos es combinar en una sola unidad datos y funciones que operen sobre estos datos. Tal unidad se denomina objeto.

- **Clases.** Una clase es la descripción de un conjunto de objetos; consta de métodos y datos que resumen características comunes de un conjunto de objetos. Se pueden definir muchos objetos de la misma clase. Dicho de otro modo, una clase es la declaración de un tipo de objeto.

- **Propiedades de la programación orientada a objetos.** Los cuatro elementos más importantes de este modelo de programación son:

- ✓ Abstracción
- ✓ Encapsulación
- ✓ Modularidad
- ✓ Jerarquía

Como sugiere Booch, si alguno de estos elementos no existe, se dice que no es programación orientada a objetos.

- **Abstracción.** La abstracción es uno de los medios más importantes, mediante el cual se enfrenta la complejidad inherente al software. La abstracción es la propiedad que permite representar las características esenciales de un objeto, sin preocuparse de las restantes características (no esenciales). Definir una abstracción significa describir una entidad del mundo real, no importa lo compleja que pueda ser, y a continuación utilizar esta descripción en un programa.

- **Encapsulación.** La encapsulación o encapsulamiento es la propiedad que permite asegurar que el contenido de la información de un objeto esta oculta del mundo exterior. El objeto A no conoce lo que hace el objeto B y viceversa.

- **Modularidad.** La modularidad es la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en si y de las restantes partes.

- **Jerarquía.** La jerarquía es una propiedad que permite una ordenación de las abstracciones. Las dos jerarquías más importantes de un sistema complejo son: estructura de clases (jerarquía "es-un" (is-a): generalización/especialización) y estructura de objetos (jerarquía "parte-de" (part-of): agregación).

Las jerarquías de generalización/especialización se conocen como herencia. Básicamente, la herencia define una relación, en donde una clase comparte la estructura o comportamiento definido en una o mas clases (herencia simple y herencia múltiple respectivamente).

La agregación es el concepto que permite el agrupamiento físico de estructuras relacionadas lógicamente. Así, un camión se compone de ruedas, motor, sistema de transmisión y chasis; en consecuencia un camión es una agregación, y ruedas, motor, transmisión y chasis son agregados de camión.

○ **Polimorfismo.** No suele ser considerada como una característica fundamental en los diferentes modelos de objetos, pero, dada su importancia, no tiene sentido considerar un objeto modelo que no soporte esta propiedad.

Es la propiedad que indica, literalmente, la posibilidad de que una entidad tome muchas formas. En términos prácticos el polimorfismo permite referirse a objetos de clases diferentes mediante el mismo elemento de programa y realizar la misma operación de diferentes formas, según sea el objeto que se referencie en ese momento.

3.10.4 Reutilización de software.

Cuando se construye un automóvil, un edificio o un dispositivo electrónico, se ensamblan una serie de piezas independientes, de modo que estos componentes se reutilicen, en vez de fabricarlos cada vez que se necesita construir un automóvil o un edificio. En la construcción de software, esta pregunta es continua. ¿Por qué no se utilizan programas ya contruidos para formar programas más grandes? Es decir, si en electrónica los computadores y sus periféricos se forman esencialmente con el ensamblado de circuitos integrados, ¿existe algún método que permita realizar grandes programas a partir de la utilización de otros programas ya realizados? ¿Es posible reutilizar estos componentes de software?

Las técnicas orientadas a objetos proporcionan un mecanismo para construir *componentes de software* reutilizables que posteriormente puedan ser interconectados entre si y formar grandes proyectos de software.

Las técnicas orientadas a objetos ofrecen una alternativa de escribir el mismo programa una y otra vez. El programador orientado a objetos modifica una funcionalidad del programa sustituyendo elementos antiguos u objetos por nuevos objetos, o bien conectando simplemente nuevos objetos en la aplicación.

La reutilización de código en reprogramación tradicional se puede realizar copiando y editando, mientras que en programación orientada a objetos se puede reutilizar el código, creando automáticamente una subclase y anulando alguno de sus métodos.

Muchos lenguajes orientados a objetos fomentan la reutilización mediante el uso de bibliotecas robustas de clases preconstruidas, así como otras herramientas, como *hojeadores* ("*browser*"), para localizar clases de interés y depuradores interactivos para ayudar al programador.

3.10.5 UML.

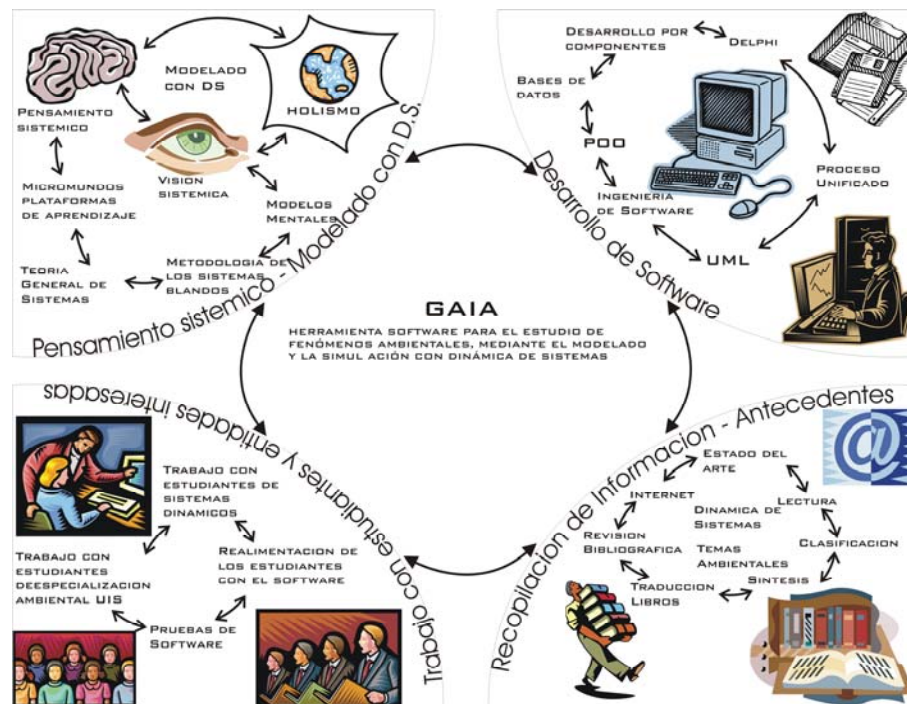
El lenguaje unificado de modelado (UML) es un lenguaje estándar de modelado para software – un lenguaje para la visualización, especificación, construcción y documentación de los artefactos de sistemas en los que el software juega un papel importante. Básicamente UML permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados. Por ejemplo, los símbolos o iconos característicos utilizados para capturar los requisitos son una elipse para representar un caso de uso y un monigote para representar el usuario que utiliza el caso de uso. De forma similar, el icono principal

utilizado en el diseño es un rectángulo para representar una clase. Estos iconos no son mas que una notación grafica, es decir, una sintaxis.

4. DESARROLLO DEL PROYECTO

Hasta el momento, se ha mencionado el marco conceptual que rodea la propuesta de la aplicación de la dinámica de sistemas para el estudio del medio ambiente; además de las características de las herramientas software que apoyan y promueven el aprendizaje a través de los medios computacionales y algunos conceptos básicos de desarrollo de software; es el momento de explicar como se utilizaron estos preceptos teóricos y todo este marco conceptual para moldear la herramienta software GAIA, la cual es creada con el fin de apoyar el aprendizaje de fenómenos del medio ambiente a través de la dinámica de sistemas y utilizando elementos multimediales y características de hipertexto; además de poseer funciones propias de una herramienta de autor.

Figura 8. Esquema del proceso llevado a cabo para elaborar la herramienta software GAIA



En la figura 8, se puede observar a grosso modo todo el proceso o conjunto de procesos que fueron necesarios para obtener la herramienta software. Como se puede observar, existen cuatro procesos bien definidos, los cuales no tiene un orden preestablecido, sino mas bien conforman un proceso mayor iterativo y cíclico, lo cual quiere decir que cada uno de estos procesos fue realizado a la par con los demás. Estos procesos son:

➤ **Acopio de Información.** Por medio de este acopio de información se obtiene el marco conceptual que sirve como base para el proceso investigativo, tanto para observar la factibilidad de uso de la Dinámica de Sistemas para el modelado y aprendizaje de fenómenos medio-ambientales como para observar el estado del arte de las herramientas computacionales que sirven como apoyo al aprendizaje. El producto mas importante de este trabajo de acopio de información es el conjunto de artículos y referencias sobre la dinámica de sistemas y el medio ambiente, reunidas dentro de la herramienta con el fin de servir como punto de inicio para investigación de diferentes temas por parte del aprendiz.

➤ **Trabajo con estudiantes y entidades Interesadas.** La labor realizada con los estudiantes de la materia Sistemas Dinámicos II del pregrado de Ing. de Sistemas y los estudiantes de la materia Modelado con Dinámica de Sistemas de la especialización en Ing. Ambiental, tuvo las características de un trabajo de grado en la modalidad docente, pues se guió el trabajo práctico con los estudiantes y se asistió en el proceso de aprendizaje. Con la ayuda de ellos se obtuvieron la mayoría de los modelos que son presentados en la herramienta, los cuales corresponden a los trabajos finales realizados en clase.

➤ **Modelado con Dinámica de Sistemas.** Corresponde al proceso por medio del cual se realizan los modelos de dinámica de sistemas del medio ambiente, este proceso viene guiado por la metodología de modelado de la Dinámica de Sistemas.

➤ **Desarrollo de Software.** Corresponde al proceso por medio del cual se elaboró la herramienta software, con base en los requisitos funcionales. Este proceso fue guiado a través de la metodología del proceso unificado de desarrollo de software.

Todos estos procesos juntos conforman un proceso mas grande y sirven como elementos moldeadores del producto final obteniendo como resultado conocimiento teórico – practico acerca del uso del paradigma dinámico – sistémico en el medio ambiente y además una herramienta que facilita el aprendizaje tanto de la dinámica de sistemas como de los fenómenos ambientales a través de la dinámica de sistemas. Es importante destacar, que la herramienta software puede también ser utilizada en otros campos del conocimiento humano, es decir que se puede separar la herramienta y orientarla al estudio de otra rama del conocimiento, por sus características multimediales y de autocreación de contenidos (Herramienta de autor), lo cual la convierte en una herramienta de uso generalizado donde se quiera utilizar el paradigma dinámico - sistémico para comprender alguna materia de estudio especifica o en casos donde se quiera utilizar las características y bondades del aprendizaje asistido por ordenador.

4.1 ACOPIO DE INFORMACION

El acopio de información es una fase o proceso fundamental en cualquier proyecto, mas aun si tiene algún carácter investigativo como en el caso de estudio. No es un proceso difícil sino más bien arduo y continuo, se realizó recopilación de información a lo largo de todo el proceso de

desarrollo del proyecto y gracias a este acopio continuo se obtuvo la información que sirve como base para los otros tres procesos llevados a cabo. Además a través de este proceso surge el marco teórico presentado en el capítulo anterior y gran parte de los artículos y documentos en formato pdf que se encuentran dentro de la herramienta software.

El acopio de información se llevó a cabo principalmente de tres fuentes diferentes:

- Documentos e información en Internet.
- Libros tanto propios como de bibliotecas (Biblioteca UIS, Biblioteca Universidad Nacional, SIMON).
- Revistas, artículos y ponencias propiedad del Grupo SIMON.

Es importante mencionar que la cantidad de libros especializados en el tema encontrados en las bibliotecas mencionadas es bastante bajo; por lo cual se hizo necesario la adquisición de libros, de los cuales cabe destacar el libro escrito por Andrew Ford Modelling the environment de Island Press; el cual sirvió como libro guía a los cursos de Sistemas Dinámicos II del primer y segundo semestre del año 2003, trabajo que dio como resultado la traducción completa del libro y la reproducción de los modelos que se encuentran en el, elementos que están contenidos dentro de la herramienta, es decir que allí encontrará una versión interactiva del libro. Por otro lado el libro Gestión Ambiental Sistémica de Leonel Vega Mora, sirvió de apoyo en la elaboración del marco conceptual referente al estudio del medio ambiente a través del enfoque dinámico – sistémico, además ayuda a entender una visión dinámica – sistémica del medio ambiente, este libro es muy recomendable para cualquier persona interesada en trabajar a fondo la aplicabilidad del enfoque sistémico al medio ambiente.

A continuación se presenta un resumen de los contenidos teóricos que encontrará dentro de la herramienta.

Tabla 3. Información encontrada dentro del software.

Tipo de Información	Descripción
Guía de estudio de Dinámica de sistemas aplicada al medio ambiente.	Cuestionario guía con las preguntas mas frecuentes sobre la dinámica de sistemas y su aplicación al estudio del medio ambiente. Estas preguntas se encuentran dentro de documentos interactivos de la herramienta software.
Artículos introductorios al estudio de la dinámica de sistemas.	Artículos en formato pdf, que introducen al aprendiz en el tema.
Libro Modeling the environment, An introduction to system Dynamics Modeling of Environmental System. Escrito por Andrew Ford.	Traducción completa del libro de Andrew Ford, aquí encuentra información suficiente para desarrollar un curso de dinámica de sistemas y medio ambiente para uno o dos semestres, varios ejemplos y aplicaciones de la dinámica de sistemas al estudio del medio ambiente. Los capítulos del libro se encuentran en documentos interactivos dentro de la herramienta.
Artículos y papers que muestran la aplicación de la dinámica de sistemas en el estudio de fenómenos ambientales.	Encontrara información de diferentes trabajos a nivel internacional donde se ha aplicado la dinámica de sistemas para el estudio del medio ambiente. La mayoría de los artículos están en formato pdf y .doc
Papers que muestran el concepto de los llamados	Se muestra la relación que tiene el pensamiento sistémico y la dinámica de

Environmental Management Systems (Sistemas de Gestión Ambiental)	sistemas con los sistemas de gestión ambiental. Estos archivos están en formato pdf.
Papers donde se explica el alcance de la norma ISO 14000	En estos articulo se muestra la relación existente entre la norma iso 14000, los EMS, el pensamiento sistémico y la dinámica de sistemas.

En total encontrará dentro de la herramienta un conjunto de 150 archivos en formato pdf con los contenidos antes mencionados. En la bibliografía podrá encontrar más información sobre todo el material utilizado para el desarrollo del proyecto.

En general el proceso de acopio de información fue continuo y apoyó el desarrollo de todo el proyecto.

4.2 TRABAJO CON ESTUDIANTES Y ENTIDADES INTERESADAS

Como se mencionó en el capítulo de preliminares, el proyecto es de interés para varias entidades entre las cuales se encuentra el Centro de estudios en ingeniería ambiental CEIAM y la especialización de ingeniería ambiental de la UIS, entidades estas que de alguna forma intervinieron en el proceso de desarrollo del proyecto. El CEIAM, prestó colaboración facilitando información y asesoría a los proyectos semestrales correspondientes a los alumnos de Sistemas Dinámicos II del primer y segundo semestre del año 2003; el centro facilitó libros y además dictó charlas sobre los tópicos correspondientes a los trabajos semestrales de los alumnos (diferentes temas del medio ambiente).

Es importante resaltar el trabajo realizado con los estudiantes de Sistemas Dinámicos II, pues en los cursos del primer y segundo semestre

del año 2003 se hizo hincapié en aplicar la Dinámica de Sistemas en el estudio de fenómenos del medio ambiente y la mayoría de los proyectos semestrales corresponden a modelos medio - ambientales elaborados a través del modelamiento Dinámico Sistémico; allí se fomentó el trabajo en grupo y la investigación en los alumnos, un trabajo multidisciplinario pues además de aprender sobre la Dinámica de Sistemas aprendieron sobre los fenómenos estudiados; a ellos se les prestó asesoría para la elaboración de sus trabajos finales que dieron como resultado gran parte de los modelos que se encuentran dentro de la herramienta software.

Un trabajo similar pero de menor intensidad en cuanto a tiempo fue realizado con los estudiantes de la especialización en Ingeniería ambiental de la UIS sedes Bucaramanga y Bogota; aunque el trabajo con ellos estuvo mas centrado en observar las dificultades a la hora de aprender el paradigma dinámico – sistémico, dio como resultado un conjunto de preguntas, el cual fue ampliado y corresponde a las preguntas guías que orientan el estudio dentro de la herramienta software.

Como se puede observar el trabajo con los alumnos tiene matices de tipo docente, por lo que se considera que dentro de este proceso se llevó a cabo un trabajo similar al que se realiza en los proyectos de grado en la modalidad de docencia.

Este trabajo esta estrechamente relacionado con el proceso de desarrollo de modelos con dinámica de sistemas, pues es esta metodología la que se utilizó en el trabajo conjunto con los alumnos para desarrollar los modelos.

4.3 MODELADO CON DINAMICA DE SISTEMAS

En el marco teórico se mencionó la metodología de modelado con dinámica de sistemas; este proceso es el que fue llevado a cabo para la

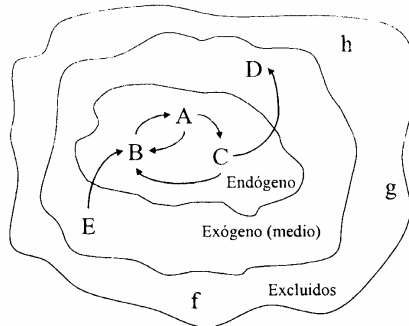
elaboración de cada uno de los modelos presentes en la herramienta software.

A continuación se presenta a manera de ejemplo el proceso que se llevó a cabo para la elaboración de uno de los modelos; como es lógico el proceso que se siguió es el mismo para todos los otros modelos desarrollados; de tal forma que queda entendido que la metodología aquí explicada fue aplicada para el desarrollo de todos los modelos y fue aplicada en conjunto con los alumnos de sistemas dinámicos II.

El modelo que a continuación se presenta corresponde al modelo del flujo del DDT en el ambiente; se toma este modelo por ser muy ilustrativo, además de que representa un grave problema ambiental e ilustra los errores que se cometen a la hora de pensar en las consecuencias a futuro de las acciones que cada cual realiza; es decir, ilustra la poca visión a largo plazo que se tiene a la hora de actuar sobre el medio ambiente.

Como ya se observó con anterioridad, lo primero es obtener un "modelo en prosa" que describa en forma de cuento o historieta la situación problema; además se debe delimitar el objeto que se va a modelar ver figura 9; es necesario recordar que el proceso de modelado con dinámica de sistemas es iterativo e incremental y que de tal forma los límites del modelo van cambiando, pues se trata de prototipos de complejidad creciente que cada vez abarcan más la realidad; aquí hay una discusión interesante. ¿Hasta donde se debe iterar el modelo?, la respuesta más aceptada es iterar hasta donde se obtenga un producto (modelo) que sirva para satisfacer los objetivos y fines con el cual fue desarrollado.

Figura 9. Estructura general de un Sistema - Modelo²³



4.3.1 Planteamiento del problema.

El DDT es un compuesto orgánico; es tóxico para los insectos pero no para las plantaciones y es extremadamente estable. Una simple aplicación permanece en el terreno y sigue siendo tóxica para los insectos por un largo tiempo. Se usó por primera vez en 1940 con gran éxito, y se pensó que sería un insecticida ideal. Botkin y Keller (1995, 208) comentan que los científicos de la época pensaron que habían conseguido la “bala mágica” – un químico que rápidamente localizaba los individuos de especies particulares y los mataba, sin efectos negativos en ninguna otra forma de vida”. Se ha estimado que el DDT y otros insecticidas han prevenido la muerte prematura de al menos unos 7 millones de personas en manos de enfermedades transmitidas por insectos tales como la malaria (Miller 1997, 382) En 1948, Paul Miller recibió el premio Nobel por su trabajo en los DDT; catorce años después aparecen estudios que previenen acerca del uso continuado de DDT, se argumenta que los insecticidas y otros contaminantes causan efectos adversos en otros seres vivos, y no solamente en los insectos.

El DDT presenta serios problemas debido a su gran solubilidad en las grasas; se almacena en la grasa animal y a través de esta se mueve a lo

²³ Esta figura es tomada del libro Pensamiento Sistémico: Diversidad en búsqueda de unidad, pag. 191.

largo de la cadena alimenticia. Miller (1997, 385) cita un ejemplo de un estuario cerca de Long Island (Nueva York, Estados Unidos), donde las medidas de DDT en el agua eran cercanas a las 2.5 partes por trillón. En el plancton y en los pequeños peces, las concentraciones eran próximas a 0.5 partes por millón. Los peces más grandes se comen a los chicos y las águilas consumen a los peces más grandes. La concentración de DDT en las águilas esta considerada en 25 partes por millón, aproximadamente diez millones de veces mas que la concentración del DDT en el agua.

El DDT es especialmente peligroso en los carnívoros que se encuentran en el tope de la cadena alimenticia, tales como las águilas, los halcones, los pelícanos, etc. Afecta la hormona que regula los niveles de calcio, causando que los cascarones de sus huevos sean sumamente frágiles y por consiguiente afectando su reproducción. Por ejemplo, estudios indican que el halcón peregrino está en grave peligro de extinción debido a problemas de tipo reproductivo asociados directamente con el DDT.

Todo esto ha conducido a la prohibición del uso de DDT en varios países. En Estados Unidos fue prohibido en el año de 1971, nueve años después de la publicación de los primeros estudios que advertían los problemas por el uso del DDT. A partir de esa prohibición ha habido una dramática recuperación de las poblaciones de aves tales como el pelicano marrón en la costa de California (Estado Unidos). El DDT aún sigue siendo usado en varias partes del mundo a pesar de el efecto tan nocivo que tienen sobre las poblaciones de aves y del decaimiento que ha tenido su efecto sobre las poblaciones de insectos pues estos se han hecho mas resistentes al DDT. Las Naciones Unidas y la organización mundial de la salud han reportado un uso anual de más de treinta mil toneladas métricas por año.

4.3.2 Objetivo del modelo.

El objetivo fundamental del modelo es apreciar y comprender el flujo del DDT en el ambiente y el tiempo que este demora en ser degradado en sustancias no dañinas a los seres vivos. Además se pretende observar los efectos de su uso en los seres vivos y como estos captan el DDT que es vertido en el suelo.

4.3.3 Modelado de la situación por prototipos.

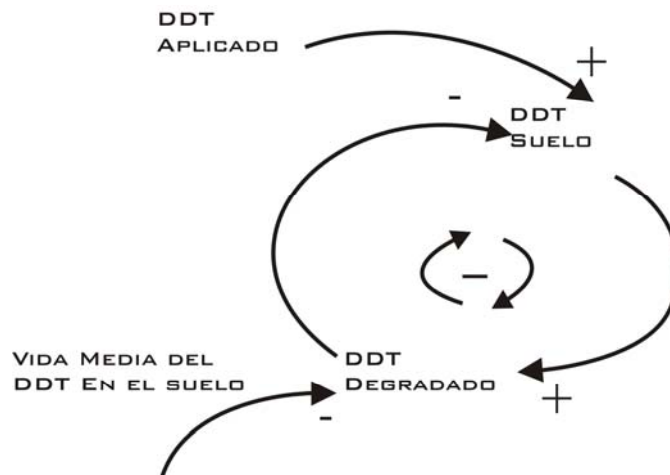
El proceso de modelado implica un desarrollo a partir de un núcleo central que se va constituyendo en el modelo final a medida que gana en detalle y en cobertura, es decir, como producto de la inclusión o desagregación de elementos del sistema y la clarificación de las relaciones entre ellos. El proceso de incremento de la complejidad se utiliza aquí como recurso que posibilita la presentación progresiva de la Dinámica de Sistemas en términos de todos sus útiles (prosa, diagramas causales, forrester), este incremento progresivo de la complejidad se va realizando en cada iteración teniendo siempre en mente el objetivo con el que se diseña el modelo.

• **Primer prototipo.** La idea es empezar aquí con el núcleo que poco a poco se ira extendiendo ganando complejidad y extensión.

La idea es simular la acumulación del DDT en el suelo, simplemente existe una entrada de DDT al suelo, el cual paulatinamente va acumulando esa cantidad de DDT que le entra del medio; además de la acumulación, existe el proceso de degradación por medio del cual el suelo transforma el DDT en sustancias no dañinas para los seres vivos; lo cual ocurre a través de complejos procesos químicos de descomposición, fotodescomposición y metabolismo biológico. Para efectos prácticos se asume que se vierten al suelo 100 toneladas de DDT en el año 1950, lo cual permitirá observar el comportamiento de esa cantidad de DDT en el suelo con el paso del tiempo.

Ahora se continúa con el diagrama causal que corresponde al observado en la figura 10.

Figura 10. Diagrama causal del primer prototipo



Resulta claro que mientras mayor sea el DDT aplicado al suelo, la cantidad de DDT en el suelo deberá aumentar; de igual forma mientras mas DDT acumulado exista, la cantidad degradada debe ser mayor, pero por el contrario, al aumentar la degradación del DDT, la cantidad de este acumulada en el suelo disminuye y por último es claro que mientras mayor sea la vida media del DDT en el suelo, el DDT degradado será menor.

La lectura del bucle se basó en las relaciones de influencia ya descritas entre parejas de elementos, pero no en términos de los signos que acompañan los extremos de las flechas, las cuales muestran el sentido de la influencia. Los signos usados son el menos (-) y el mas (+); el signo mas refleja que la relación entre los dos elementos se da en el mismo sentido. Por ejemplo mientras mas DDT se aplique al suelo, la cantidad de DDT en el suelo aumentara corresponde a una influencia con signo + pues ambos elementos van en el mismo sentido; por otro lado se tiene que cuando mayor sea el DDT degradado, menor será el DDT acumulado

en el suelo lo cual indica que se trata de una influencia de compensación y es identificada con un signo -.

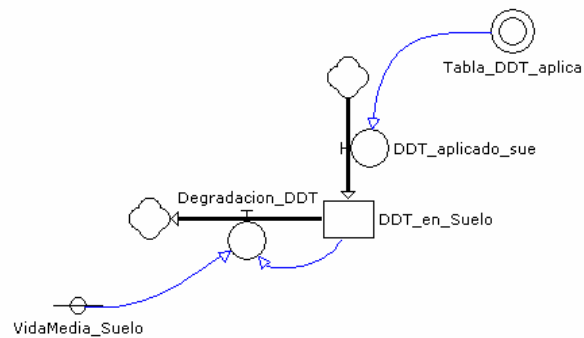
En general el comportamiento de los bucles puede ser de dos tipos, de compensación o de refuerzo; los bucles de refuerzo (identificados con un signo +) son aquellos que amplifican el cambio original. Dicho de otro modo, el cambio recorre todo el sistema produciendo mas cambios en la misma dirección. Por otro lado los bucles de compensación (identificados con signo -) ocurren cuando los cambios registrados en todo el sistema se oponen al cambio original para amortiguar el efecto.

En el caso del bucle anterior (el diagrama causal de la Figura 10), se trata de un bucle de compensación. Pues el recorrido por este hace que el efecto producido por uno de los elementos sea amortiguado por el otro durante las sucesivas iteraciones.

Hay una regla muy sencilla para saber si un bucle completo es de refuerzo o de compensación, por muchos o diversos que sean los vínculos que contenga. Si la secuencia entera tiene un número par de vínculos negativos (de compensación), el bucle será de refuerzo (se incluyen aquí los bucles que no contengan ningún vinculo de compensación). Si tiene un número impar de vínculos negativos (de compensación), el bucle será de compensación.

Expresando esto en términos de flujos y niveles se obtiene el siguiente diagrama de Forrester.

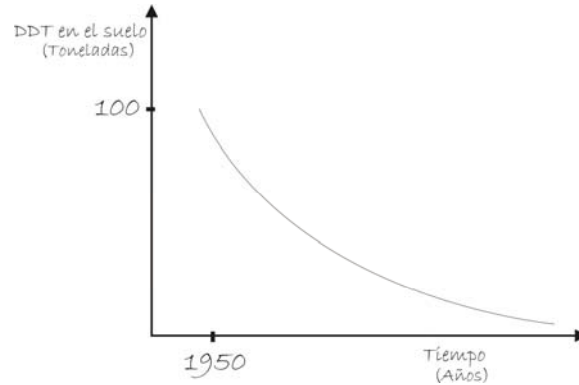
Figura 11. Diagrama de Flujos y Niveles del primer prototipo



La elaboración correcta del diagrama de Forrester es un punto muy importante; muchas veces durante el proceso de modelado no se utilizan los diagramas causales sino mas bien se va directamente al diagrama de Flujos y Niveles; sea cual sea el caso, es necesario hacer una buena diferenciación entre lo que es un nivel y lo que corresponde a un flujo. De modo general se puede decir que un Nivel corresponde al material que se acumula en el tiempo y el flujo es lo que hace que ese nivel cambie en el tiempo. En base a esas premisas se hace el paso del diagrama de influencias al de flujos y niveles. Además de los flujos y niveles, se observa la inclusión de un parámetro en el diagrama, estos parámetros corresponden a valores que permanecerán fijos a lo largo de la simulación.

o **Modo de referencia.** El análisis del diagrama de influencias que incluyó el bucle de compensación junto con las consideraciones iniciales planteadas para formular el modelo (prosa), permite discutir la tendencia de comportamiento. Se trata de imaginar la trayectoria o trayectorias temporales, es decir, hacer una cierta "simulación mental". Este comportamiento se denomina **modo o comportamiento de referencia** y aporta un elemento de comparación y/o validación de las trayectorias que reporta el modelo matemático.

Figura 12. Comportamiento de referencia (simulación mental) del primer prototipo



La figura 12 muestra el posible comportamiento del nivel de DDT en el suelo al principio empezaría en 100 debido a que se vierten 100 toneladas de DDT en el año 1950 y luego esa cantidad de DDT decrece debido a la degradación en el suelo.

o **Ecuaciones del primer prototipo.** El diagrama de flujo – nivel se corresponde con un modelo en ecuaciones, no todas explícitamente definidas. A continuación se enuncian las respectivas ecuaciones:

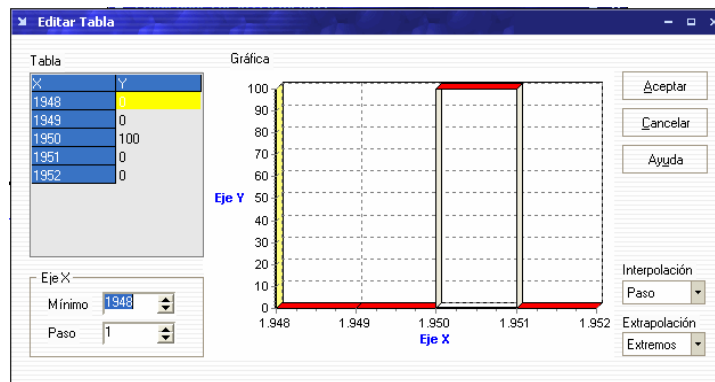
$$\frac{D(\text{DDT_en_suelo})}{dt} = \text{DDT_aplicado_sue}(t) - \text{Degradacion_DDT}(t)$$

Ecuación diferencial de primer orden asociada con toda relación flujo – nivel. En términos generales, el cambio de un nivel en el tiempo, es igual al flujo neto que actúa sobre el.

Ahora se pasa a definir cada uno de los flujos; en este caso teniendo en cuenta que el objetivo del modelo es observar el comportamiento del flujo de DDT en el ambiente; y particularmente para este primer prototipo solo en el suelo; se considerara que se hace una aplicación de 100 Toneladas de DDT en el año 1950; para definir este flujo, se utiliza una variable exógena (Tabla_DDT_aplica), la cual define la entrada de 100 toneladas

de DDT en el año 1950; en la Figura 13 se observa como definir esta variable exógena usando el módulo de laboratorio de la herramienta software GAIA (para mas información sobre el uso del modulo de laboratorio y de la elaboración de diagramas de Forrester ver el manual del usuario anexo).

Figura 13. Definición de la variable exógena *Tabla_DDT_aplica*



Como se puede observar en la gráfica en el año 1950 la gráfica asciende de cero a cien, esto representa el vertimiento de las 100 toneladas de DDT en el suelo; la interpolación utilizada es paso y la extrapolación de la gráfica es extremos lo cual significa que desde 1951 en adelante tomará el valor de cero e igualmente antes del año 1950 también tomara el valor de cero. Esta gráfica queda expresada como una interpolación y el software la representa mediante la siguiente función.

$INTPASO(2,1948,1,0,0,100,0,0)$

El flujo de entrada *DDT_aplicado_sue* queda determinado por los valores de la tabla, de la siguiente manera, la expresión que determina este flujo estaría dada por:

$$DDT_aplicado_sue(t) = Tabla_DDT_aplica$$

Ahora se definirá el flujo de salida; normalmente, las ecuaciones que implican el uso de vida media usan un factor de 1.44 por la vida media, esto debido a que el tiempo promedio es aproximadamente 1.44 veces mas largo que la vida media:

$$\text{Degradacion_DDT}(t) = \text{DDT_en_suelo}/1.44 * \text{VidaMedia_Suelo}$$

No obstante, se utilizará un factor de conversión de 1.5 para multiplicar la vida media del DDT en el suelo, esto debido a que un investigador de apellido Rander que elaboró varios modelos sobre el flujo de DDT recomienda esta aproximación, debido a observaciones hechas por el mismo; aquí cabe mencionar la importancia de la investigación en el tema que se va a modelar, como se puede ver, a partir de la propuesta hecha por Rander, se realiza un cambio (aunque ligero) a la ecuación que corresponde a la degradación de DDT. Asumiendo esta recomendación, la ecuación quedaría expresada de la siguiente manera:

$$\text{Degradacion_DDT}(t) = \text{DDT_en_suelo}/1.5 * \text{VidaMedia_Suelo}$$

Para la ecuación diferencial asociada con la relación flujo – nivel, es posible obtener su solución en el proceso de simulación, mediante un método numérico. En la forma de Euler esta ecuación diferencial se puede expresar como:

$$\text{DDT_en_suelo}(t+\Delta t) = \text{DDT_en_suelo}(t) + (\text{DDT_aplicado_sue}(t) - \text{Degradacion_DDT}(t)) * \Delta t$$

En donde Δt corresponde al paso de integración y al intervalo de tiempo con el cual se van registrando los eventos del fenómeno simulado. Este delta de tiempo debe ser seleccionado de tal forma, que se garantice la exactitud numérica de los cálculos (dependiendo del algoritmo de solución numérica que se asuma) y la apreciación de los eventos del fenómeno. En términos generales, el paso de integración debe dar cobertura o permitir el registro, del evento significativo de mayor velocidad considerado en el modelo. Además la ecuación de nivel en la forma de Euler, deja ver la

aproximación que se hace al asumir los flujos como constantes durante el Δt , lo cual indica la necesidad de definir un Δt que permita esa aproximación.

Las simplificaciones asumidas para representar el fenómeno, han conducido a un modelo matemático lineal, definido por una ecuación diferencial asociada al nivel y por dos ecuaciones de flujo. En la medida en que el modelo gane complejidad y cobertura, generalmente se convertirá en un modelo matemático no lineal, que se apreciará en las ecuaciones de flujo.

○ **Simulación.** Con el modelo explícitamente definido y con un escenario que determina las condiciones de simulación, es posible obtener una solución numérica que ofrece trayectorias temporales para cada una de las variables consideradas en el modelo. Además de definir el escenario, se requiere la información para la solución numérica: Δt , Algoritmo de integración, tiempo inicial y tiempo final, por lo menos.

Simulación de ejemplo:

Escenario:

DDT_en_suelo(0) = 0 Toneladas Cantidad Inicial de DDT en el suelo.

VidaMedia_Suelo = 10 años Vida media del DDT nocivo en el suelo.

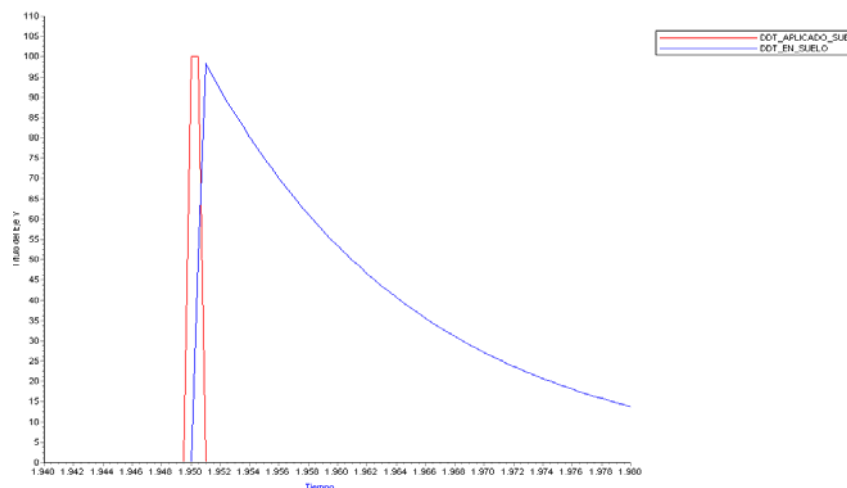
Otras condiciones de simulación:

$\Delta t = 0.5$ años²⁴; tiempo inicial = 1940; tiempo final = 1980; Algoritmo de Integración: Euler.

²⁴ el Δt nos representa el intervalo de tiempo en el que cada iteración es realizada, en la mayoría de los casos se eligen de manera tal que permitan tomar dentro de la simulación el fenómeno que mas

La figura 14 se obtuvo al realizar la solución numérica de la simulación correspondiente al primer prototipo. Para ello, se utilizó el modulo de laboratorio que se encuentra dentro de la herramienta software, el cual reutiliza componentes del software EVOLUCION 3.5 beta 20 desarrollado por el grupo SIMON de Investigaciones en Modelado y Simulación de la Universidad Industrial de Santander.

Figura 14. Resultados de la simulación del primer prototipo.



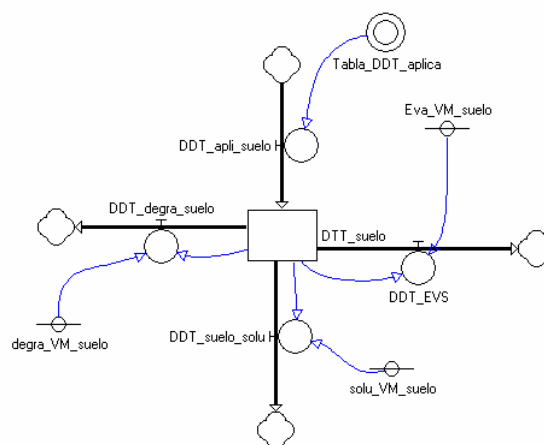
Como se puede observar en la simulación, la línea roja que representa el DDT aplicado al suelo asciende a 100 toneladas en el año 1950 y luego vuelve a cero; la línea azul representa el DDT que está acumulado en el suelo, el cual salta inmediatamente a 100 toneladas y declina paulatinamente; luego de 10 años, aproximadamente 50 toneladas permanecen en el suelo y después de 20 años, aproximadamente unas 25 toneladas siguen allí.

• **Segundo Prototipo.** Ahora se procede a expandir el modelo inicial, tomando en cuenta el DDT que sale del suelo gracias a la evaporación,

rápidamente ocurre dentro del modelo, no obstante su elección depende mucho del criterio del modelador.

este DDT que sale del suelo esta influenciado por la vida media del DDT en el proceso de evaporación, la cual según estudios, está estimada en dos años; por otro lado se tiene el DDT que sale del suelo a los ríos y a los mares, es importante mencionar que el DDT es muy poco soluble en el agua, pero se adhiere muy fuertemente a pequeñas partículas sólidas del suelo; la cantidad de DDT que sale del suelo a por medio de este proceso de escorrentía es muy poco y se estima una vida media de unos 500 años. El diagrama de Flujos y Niveles quedaría representado de la siguiente manera:

Figura 15. Representación en Flujos y Niveles del segundo prototipo



o **Ecuaciones del segundo prototipo.** Se añadieron dos flujos al modelo, cada uno de los cuales tiene sus respectivas ecuaciones, ambos flujos ayudan a acercarse más al modelo a la manera real como el DDT es degradado y desaparece del suelo.

Los dos nuevos flujos son: *DDT_EVES* el cual representa la evaporación del DDT; y el otro flujo es *DDT_suelo_solu* que representa el paso del DDT a diversas fuentes de agua por escorrentía, ambas ecuaciones vienen determinadas por la vida media del DDT en cada fenómeno, tanto en el de

escorrentía como en el de evaporación; las ecuaciones que las definen son:

$$\frac{D(\text{DTT}_{\text{suelo}})}{dt} = \text{DDT}_{\text{aplicado_sue}}(t) - \text{DDT}_{\text{degra_suelo}}(t) - \text{DDT}_{\text{suelo_solu}}(t) - \text{DDT}_{\text{EVS}}(t)$$

El flujo DDT_{EVS} esta dado por:

$$\text{DDT}_{\text{EVS}}(t) = \text{DTT}_{\text{suelo}} / (1.5 * \text{Eva}_{\text{VM_suelo}})$$

Donde $\text{Eva}_{\text{VM_suelo}}$ representa la vida media del DDT en su proceso de evaporación.

Por otro lado el flujo $\text{DDT}_{\text{suelo_solu}}$ viene dado por la expresión:

$$\text{DDT}_{\text{suelo_solu}}(t) = \text{DTT}_{\text{suelo}} / (\text{solu}_{\text{VM_suelo}} * 1.5)$$

$\text{Solu}_{\text{VM_suelo}}$ corresponde a la vida media del DDT en el proceso de escorrentía.

Los otros dos flujos son los mismos que estaban antes en el primer prototipo y por tanto no se vuelven a describir pues no se les hizo ningún cambio; solo se cambio el nombre del flujo que corresponde a la degradación del DDT, ahora se llama $\text{DDT}_{\text{degra_suelo}}$ para hacer mas explicito que corresponde al DDT que es degradado en el suelo.

o **Simulación del segundo prototipo.** A continuación se muestra la simulación del modelo con las modificaciones que se le realizaron.

Escenario:

DDT_en_suelo(0) = 0 Toneladas

Cantidad Inicial de DDT en el suelo.

VidaMedia_Suelo = 10 años

Vida media del DDT nocivo en el suelo.

Eva_VM_suelo = 2 años

Vida media del DDT en el proceso de evaporación.

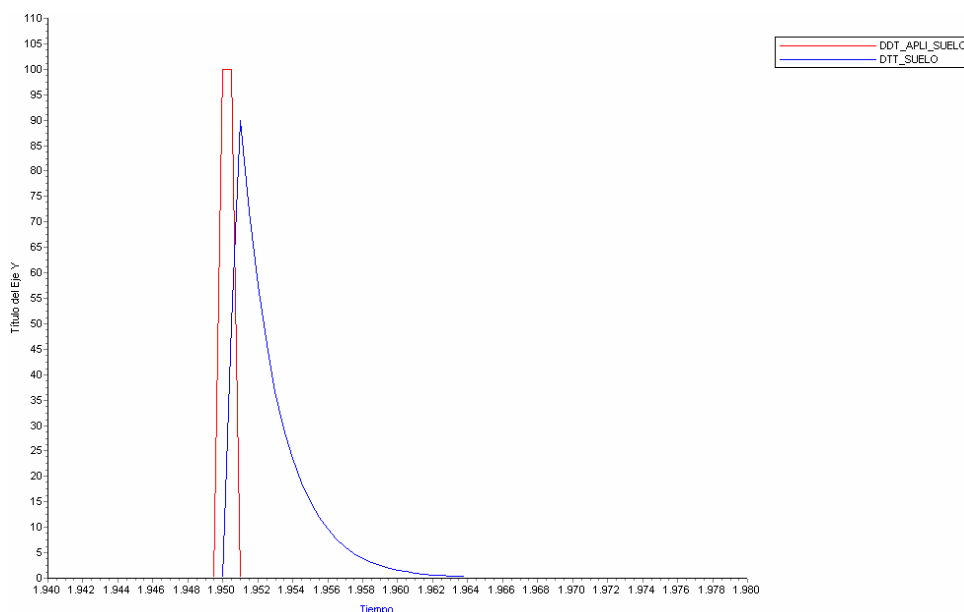
Solu_VM_suelo = 500 años

Vida media del DDT en el proceso de escorrentía.

Otras condiciones de simulación:

$\Delta t = 0.5$ años; tiempo inicial = 1940; tiempo final = 1980; Algoritmo de Integración: Euler.

Figura 16. Resultados de las simulación del segundo prototipo.

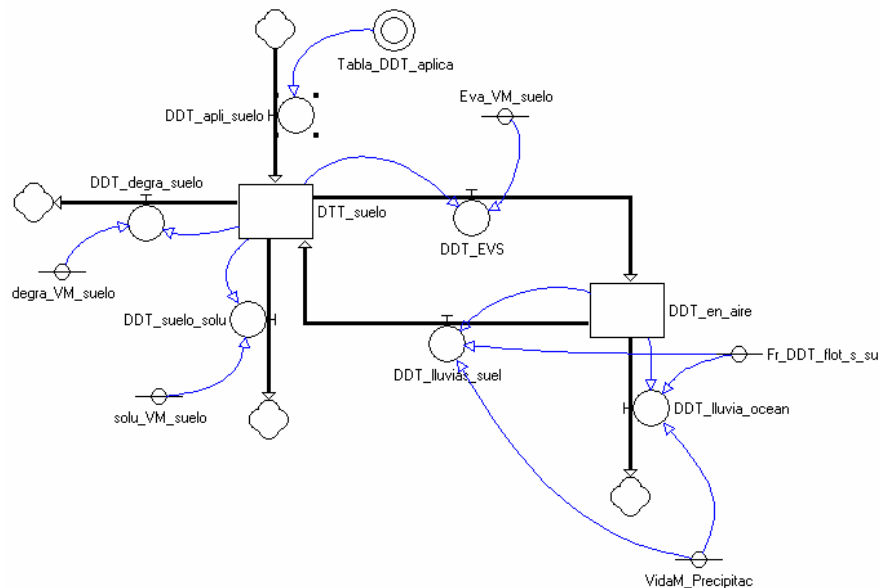


Igual que en la pasada simulación aquí el DDT aplicado pasa de 0 a 100 en el año 1950; pero ahora el comportamiento del DDT acumulado es un poco diferente, el DDT acumulado salta inmediatamente pero esta vez no llega hasta 100 debido a que la evaporación actúa rápidamente y no deja

que todas las 100 toneladas se acumulen. Este Modelo muestra que muy poca cantidad quedará en el suelo luego de 10 años.

• **Tercer Prototipo.** En este punto la pregunta obvia es ¿Cuál es el destino del DDT que deja el suelo debido a la evaporación? Y ¿Qué sucede con el DDT que se va por escorrentía a los sumideros de agua?. Este tercer prototipo resuelve la primera de las dos preguntas. Una vez las moléculas de DDT están en el aire, estas son muy móviles en este medio y pueden recorrer grandes distancias antes de caer nuevamente al suelo o a las fuentes de agua, principalmente por medio de precipitaciones. Si llueve cada 2 o 3 semanas, se asume que el DDT muy pronto dejará la atmósfera y caerá nuevamente a la superficie.

Figura 17. Diagrama de Flujos y Niveles del tercer prototipo.



○ **Ecuaciones del tercer prototipo.** Como se observa en el diagrama de Flujos y Niveles, ahora existen dos niveles, uno de ellos es el que estaba inicialmente desde el primer prototipo representando la cantidad de DDT acumulado en el suelo, el segundo corresponde al DDT acumulado en el

aire; se dice entonces que se tiene un modelo de segundo orden por la existencia de dos ecuaciones de nivel. Las correspondientes ecuaciones de Nivel quedan entonces representadas por:

$$\frac{D(\text{DDT}_{\text{suelo}})}{dt} = \text{DDT}_{\text{apli}_{\text{suelo}}(t)} + \text{DDT}_{\text{lluvias}_{\text{suel}}(t)} - \text{DDT}_{\text{degra}_{\text{suelo}}(t)} - \text{DDT}_{\text{suelo}_{\text{solu}}(t)} - \text{DDT}_{\text{EVS}}(t)$$

$$\frac{D(\text{DDT}_{\text{en}_{\text{aire}}})}{dt} = \text{DDT}_{\text{EVS}}(t) - \text{DDT}_{\text{lluvias}_{\text{suel}}(t)} - \text{DDT}_{\text{lluvia}_{\text{ocean}}(t)}$$

Las ecuaciones de los nuevos flujos vienen dadas por:

$$\text{DDT}_{\text{lluvia}_{\text{ocean}}(t)} = \text{DDT}_{\text{en}_{\text{aire}}} * (1 - \text{Fr}_{\text{DDT}_{\text{flot}_{\text{s}_{\text{su}}}}}) / (1.5 * \text{VidaM}_{\text{Precipitac}})$$

Donde $\text{Fr}_{\text{DDT}_{\text{flot}_{\text{s}_{\text{su}}}}$ corresponde a la fracción de DDT que cae por efectos de las lluvias sobre el suelo, es por esto que el Nivel $\text{DDT}_{\text{en}_{\text{aire}}}$ es multiplicado por $(1 - \text{Fr}_{\text{DDT}_{\text{flot}_{\text{s}_{\text{su}}}}})$ lo cual corresponde a la fracción de DDT que cae por efectos de la lluvia sobre el océano o los sumideros de agua, el parámetro $\text{VidaM}_{\text{Precipitac}}$ corresponde a la vida media del DDT en el proceso de precipitación que es estimado en unas 2 o 3 semanas. La otra ecuación de flujo corresponde al DDT que va nuevamente del aire al suelo por medio de las precipitaciones, y esta expresada por:

$$\text{DDT}_{\text{lluvias}_{\text{suel}}(t)} = \text{DDT}_{\text{en}_{\text{aire}}} * \text{Fr}_{\text{DDT}_{\text{flot}_{\text{s}_{\text{su}}}}} / (1.5 * \text{VidaM}_{\text{Precipitac}})$$

o **Simulación del tercer prototipo.** A continuación se muestran las condiciones utilizadas en la simulación y los resultados obtenidos

Escenario:

DDT_en_suelo(0) = 0 Toneladas

Cantidad Inicial de DDT en el suelo.

DDT_en_aire(0) = 0 Toneladas

Cantidad Inicial de DDT en el aire.

VidaMedia_Suelo = 10 años

Vida media del DDT nocivo en el suelo.

Eva_VM_suelo = 2 años

Vida media del DDT en el proceso de evaporación.

Solu_VM_suelo = 500 años

Vida media del DDT en el proceso de escorrentía.

Fr_DDT_flot_s_su = 0.3

Fracción de lluvias que transportan DDT que caen sobre el suelo (30% aprox.).

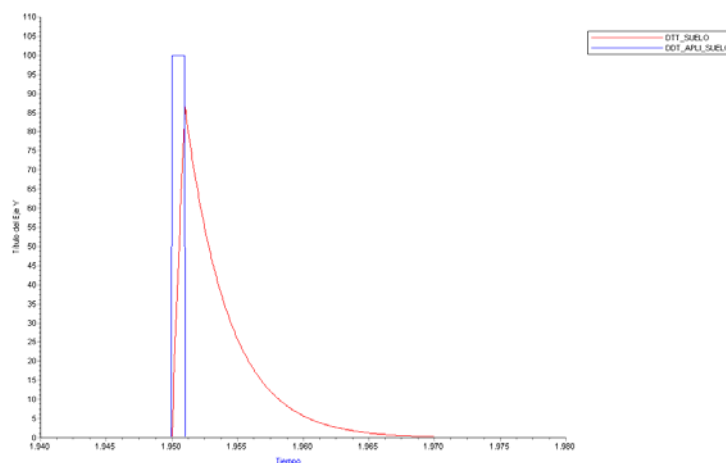
VidaM_Precipitac = 0.05 años

Vida media del DDT en el proceso de precipitación. Se estima que caen lluvias cada 2 a 3 semanas

Otras condiciones de simulación:

$\Delta t = 0.05$ años; tiempo inicial = 1940; tiempo final = 1980; Algoritmo de Integración: Euler.

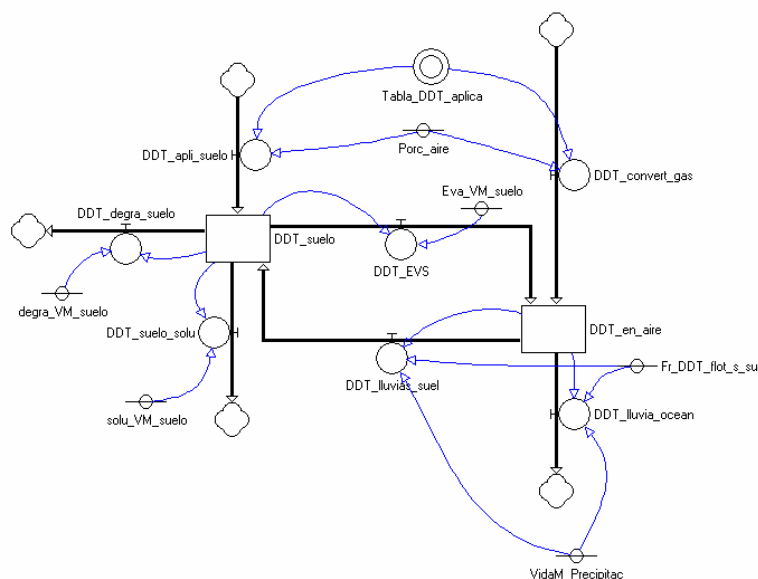
Figura 18. Comportamiento simulado del tercer prototipo



Como se puede apreciar en el gráfico del comportamiento de tercer prototipo, el comportamiento es similar al de las anteriores simulaciones, lo único diferente que se puede apreciar es que el nivel de DDT en el suelo cae más lentamente debido al regreso de DDT desde el aire gracias a las precipitaciones.

• **Cuarto Prototipo.** En este prototipo se estima la posibilidad de que parte del DDT se convierta en partículas que van directamente al aire, durante su aplicación, de tal forma que un porcentaje del DDT aplicado va al suelo y otro al aire; la representación de Flujos y Niveles es la siguiente.

Figura 19. Comportamiento simulado del cuarto prototipo



o **Ecuaciones de cuarto prototipo.** Las ecuaciones de Nivel del cuarto prototipo son:

$$\frac{D(DDT_{suelo})}{dt} = DDT_{apli_suelo}(t) + DDT_{lluvias_suel}(t) - DDT_{degra_suelo}(t) - DDT_{suelo_solu}(t) - DDT_{EV3}(t)$$

Pero la ecuación de nivel del DDT_en_aire cambia como se muestra a continuación.

$$\frac{D(\text{DDT_en_aire})}{dt} = \text{DDT_EVS}(t) + \text{DDT_convert_gas}(t) - \text{DDT_lluvias_suel}(t) - \text{DDT_lluvia_ocean}(t)$$

El Flujo *DDT_apli_suelo* cambia su definición, ya que como se ha dicho, una parte del DDT va al suelo y la otra directamente al aire.

$$\text{DDT_apli_sue}(t) = \text{Tabla_DDT_aplica} * (1 - \text{Porc_aire})$$

Donde *(1-Porc_aire)* representa la porción de DDT que va al suelo.

El Flujo *DDT_convert_gas* igualmente se ve influenciado por el parámetro *Porc_aire* de la siguiente manera.

$$\text{DDT_convert_gas}(t) = \text{Tabla_DDT_aplica} * \text{Porc_aire}$$

o **Simulación del cuarto prototipo.** Como es lógico, el parámetro *Porc_aire* influye en los flujos de *DDT_apli_suelo* y *DDT_convert_gas*.

Escenario:

DDT_en_suelo(0) = 0 Toneladas	Cantidad Inicial de DDT en el suelo.
DDT_en_aire(0) = 0 Toneladas	Cantidad Inicial de DDT en el aire.
VidaMedia_Suelo = 10 años	Vida media del DDT nocivo en el suelo.
Eva_VM_suelo = 2 años	Vida media del DDT en el proceso de evaporación.

Solu_VM_suelo = 500 años

Vida media del DDT en el proceso de escorrentía.

Fr_DDT_flot_s_su = 0.3

Fracción de lluvias que transportan DDT que caen sobre el suelo (30% aprox.).

VidaM_Precipitac = 0.05 años

Vida media del DDT en el proceso de precipitación. Se estima que caen lluvias cada 2 a 3 semanas.

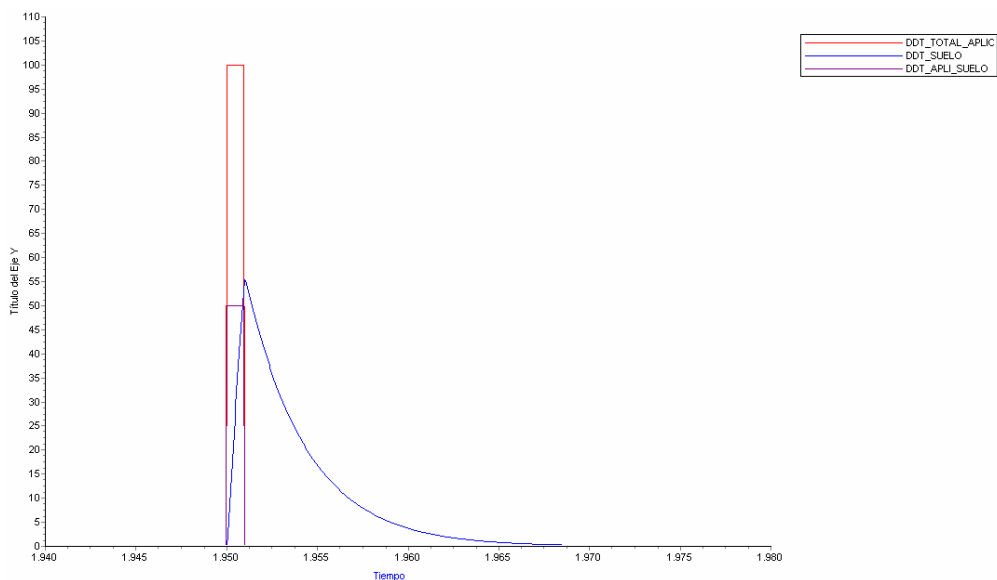
Porc_aire = 0.5

Fracción del DDT que va directamente al aire.

Otras condiciones de simulación:

$\Delta t = 0.05$ años; tiempo inicial = 1940; tiempo final = 1980; Algoritmo de Integración: Euler.

Figura 20. Resultados de la simulación del cuarto prototipo.



Se observa que el DDT aplicado al suelo que corresponde a la trayectoria de color violeta llega solo hasta la mitad por el efecto de la fracción que va directamente al aire.

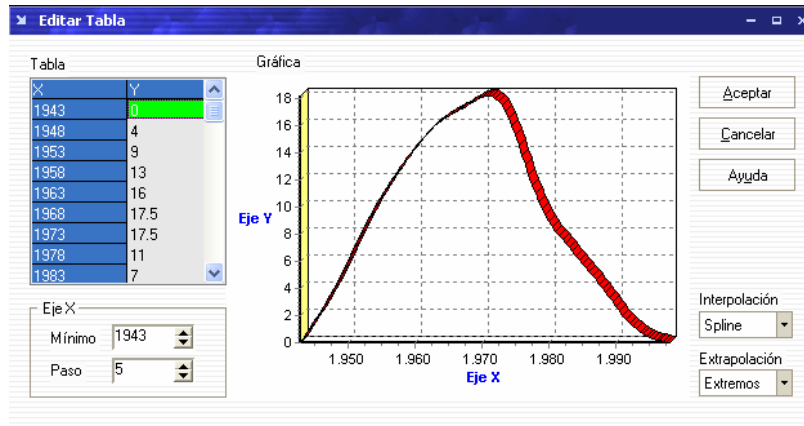
• **Quinto prototipo.** Hasta el momento se ha ido modelando la manera como el DDT fluye en el medio ambiente considerando principalmente el suelo y el aire, ahora el modelo abarcará el paso del DDT por el océano y la concentración del DDT en los peces.

La representación del flujo de DDT en el ambiente será ampliada considerando el paso del DDT a los océanos, el cual puede darse de dos formas, primero a partir de la escorrentía de agua con DDT que llega a los ríos donde se acumula y luego es transportada hasta el océano; por otro lado se tiene las precipitaciones directas sobre mares y océanos que colaboran con la acumulación de DDT en estas grandes masas de agua; a partir de esto es posible mostrar la captación que ocurre de DDT en los océanos por parte de los peces.

Es importante mencionar que se hace una mejora en la apreciación que se tiene sobre la aplicación de DDT, en los anteriores prototipos se había considerado una simple aplicación de 100 toneladas en el año 1950 y se observaba luego como esta cantidad se degradaba paulatinamente; ahora la idea es expresar el uso a nivel mundial de DDT desde 1943 hasta el momento cuando su uso es casi nulo en comparación con años anteriores.

Estas consideraciones implican que el modelo además de mostrar el comportamiento del flujo de DDT en el ambiente, sirve para probar qué tanto tiempo puede tomar que la reducción de la aplicación de DDT se vea reflejada en la concentración de DDT en los peces.

Figura 22. Definición de la variable exógena que representa la aplicación histórica de DDT desde el año 1943 a nuestros días. Los datos están expresados en miles de toneladas



Ahora los flujos DDT_apli_suelo y DDT_conver_gas están influenciados por la acción de esta variable exógena definida por la tabla de la Figura 22.

Las ecuaciones de nivel son las siguientes:

$$\frac{D(\text{DDT}_{\text{suelo}})}{dt} = \text{DDT}_{\text{apli}_{\text{suelo}}(t) + \text{DDT}_{\text{lluvias}_{\text{suelo}}(t) - \text{DDT}_{\text{degra}_{\text{suelo}}(t) - \text{DDT}_{\text{suelo}_{\text{solu}}(t) - \text{DDT}_{\text{Evaporacion}}(t)}$$

$$\frac{D(\text{DDT}_{\text{en}_{\text{aire}}})}{dt} = \text{DDT}_{\text{Evaporacion}}(t) + \text{DDT}_{\text{convert}_{\text{gas}}(t) - \text{DDT}_{\text{lluvias}_{\text{suelo}}(t) - \text{DDT}_{\text{lluvia}_{\text{ocean}}(t)}$$

$$\frac{D(\text{DDT}_{\text{en}_{\text{rios}}})}{dt} = \text{DDT}_{\text{suelo}_{\text{solu}}(t) - \text{Flujo}_{\text{al}_{\text{oceano}}(t)}$$

$$\frac{D(\text{DDT}_{\text{en}_{\text{oceano}}})}{dt} = \text{DDT}_{\text{lluvia}_{\text{ocean}}(t) + \text{Flujo}_{\text{al}_{\text{oceano}}(t) + \text{Peces}_{\text{muer}_{\text{oceano}}(t) + \text{Des}_{\text{toxicos}}(t) - \text{Capt}_{\text{DDT}_{\text{peces}}(t) - \text{Degradac}_{\text{oceano}}(t)}$$

$$\frac{D(\text{DDT}_{\text{en}_{\text{peces}}})}{dt} = \text{Capt}_{\text{DDT}_{\text{peces}}(t) - \text{exc}_{\text{no}_{\text{danindas}}(t) - \text{Peces}_{\text{consumidos}}(t) - \text{Peces}_{\text{muer}_{\text{oceano}}(t) - \text{Des}_{\text{toxicos}}(t)}$$

Los flujos que se añadieron son:

$$\text{Flujo}_{\text{al}_{\text{oceano}}(t) = \text{DDT}_{\text{en}_{\text{rios}}} / (1.5 * \text{VidaM}_{\text{Flujo}_{\text{ocea}}})$$

Agua que llega a los océanos desde los ríos. VidaM_Flujo_ocea representa la vida media del DDT en su proceso de transporte desde los ríos a los cecéanos.

$$\text{Degradac_oceano}(t) = \text{DDT_en_oceano}/(1.5 \cdot \text{VidaM_oceano})$$

Degradación de DDT en el océano. El parámetro VidaM_oceano indica la vida media de la degradación de DDT en el océano.

$$\text{Capt_DDT_peces}(t) = \text{Pes_cuer_coa} \cdot \text{Peso_peces} \cdot \text{conc_plancton}$$

Captación de DDT por parte de los peces. Donde Pes_cuer_coa es la Carga alimenticia anual de los peces; Peso_peces es el peso de los peces y Concentración en el plancton corresponde a la concentración de DDT en el plancton y esta definida como:

$$\text{conc_plancton} = \text{fact_on_plankoce} \cdot \text{conc_oceano}$$

Donde Fact_on_plankoce es un factor de concentración de plancton en el océano y conc_oceano que corresponde a la concentración de plancton en el océano esta dada por:

$$\text{conc_oceano} = \text{DDT_en_oceano}/\text{Masa_capa_mixta}$$

Es decir que la concentración de DDT en el océano se expresa como el cociente del DDT_en_oceano entre la masa mixta de agua con DDT y otros organismos.

$$\text{Exc_no_daninas}(t) = \text{Fr_Degradada} \cdot \text{Tasa_exc_DDT_pez}$$

Este flujo corresponde al DDT degradado por los peces y convertido en excreciones no dañinas. Fr_Degradada indica la fracción degradada de la cantidad de excreción de los peces y la tasa de excreción de DDT por parte de los peces expresada por Tasa_exc_DDT_pez.

$$\text{Tasa_exc_DDT_pez} = \text{DDT_en_peces}/(1.5 \cdot \text{VidaM_des_peces})$$

Es decir el cociente de la cantidad acumulada de DDT en los peces entre el promedio de excreciones de los peces, o lo que es igual 1.5 por la vida media de las excreciones en el pez.

$$\text{Peces_Consumidos}(t) = \text{Fr_consumida} \cdot \text{Muerte_peces}$$

El flujo Peces_consumidos(t) representa los peces consumidos por pesca. Fr_consumida representa la fracción de peces muertos que son consumidos y muerte_peces esta expresada de la siguiente manera.

$$\text{Muerte_peces} = \text{DDT_en_peces} / (1.5 * \text{VidaM_peces})$$

Muerte peces expresa la cantidad en toneladas de DDT contenidas en peces que mueren, esta determinada por el cociente entre DDT_en_peces y el promedio de vida de los peces que equivale a 1.5 por la VidaM_peces.

$$\text{Pec_muer_oceanico}(t) = \text{Muerte_peces} * (1 - \text{Fr_consumida})$$

Representa el DDT que vuelve al océano por muerte de peces que tienen concentración de DDT en su organismo. Se obtiene de multiplicar las toneladas de DDT que están en los peces muertos por (1-Fr_consumida) es decir por la fracción de peces que no son consumidos.

$$\text{Des_toxicos}(t) = \text{Tasa_exc_DDT_pez} * (1 - \text{Fr_Degradada})$$

Excreciones toxicas de los peces que vuelven al océano, se multiplica la tasa de excreción de DDT de los peces por (1-Fr_Degradada), o sea la fracción no degradada de DDT.

Además, se utiliza una variable auxiliar que sirve para calcular la concentración del DDT en los peces; la ecuación utilizada para este cálculo es:

$$\text{Conc_en_pez} = \text{DDT_en_peces} / \text{Peso_peces}$$

o **Simulación del quinto prototipo.** Como es lógico, el parámetro Porc_aire influye en los flujos de *DDT_apli_suelo* y *DDT_convert_gas*.

Escenario:

DDT_en_suelo(0) = 0 Toneladas Cantidad Inicial de DDT en el suelo.

DDT_en_aire(0) = 0 Toneladas Cantidad Inicial de DDT en el aire.

DDT_en_rios(0) = 0 Toneladas	Cantidad Inicial de DDT en los rios.
DDT_en_oceano(0) = 0 Toneladas	Cantidad Inicial de DDT en océanos.
DDT_en_peces(0) = 0 Toneladas	Cantidad Inicial de DDT en los peces.
degra_VM_Suelo = 10 años	Vida media del DDT nocivo en el suelo.
Eva_VM_suelo = 2 años	Vida media del DDT en el proceso de evaporación.
Solu_VM_suelo = 500 años	Vida media del DDT en el proceso de escorrentía.
Fr_DDT_flot_s_su = 0.3	Fracción de lluvias que transportan DDT que caen sobre el suelo (30% aprox.).
VidaM_Precipitac = 0.05 años	Vida media del DDT en el proceso de precipitación. Se estima que caen lluvias cada 2 a 3 semanas.
Porc_aire = 0.5	Fracción del DDT que va directamente al aire.
VidaM_Flujo_ocea = 0.1 años	Vida Media del Flujo de DDT de los ríos a los océanos.
VidaM_oceano = 15 años	Vida Media del DDT en el océano.
VidaM_des_peces = 0.3 años	Vida Medio de los desechos de los peces.
Fr_Degradada = 0.1	Fracción Degradada de los desechos de los peces.
Fr_consumida = 0.5	Fracción Consumida de peces.
VidaM_peces = 3 años	Vida Media de los peces.
Peso_peces = 6×10^8	Masa de los peces en el océano.
Pes_cuer_coa = 10 (Tons/año)	Carga alimenticia anual.

fact_on_plankoce = 2000

factor de concentración de
plancton en el océano.

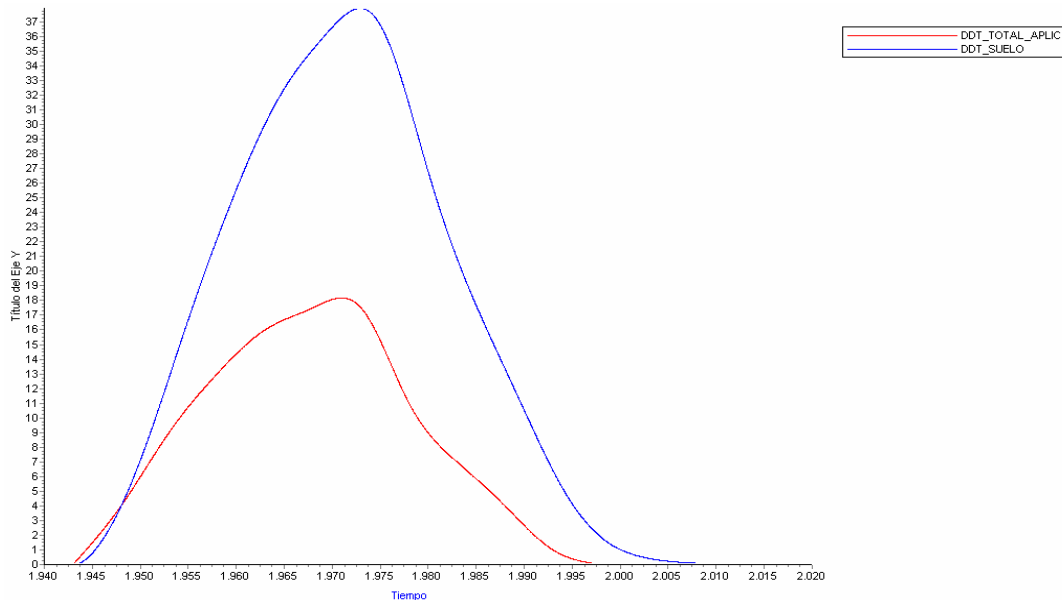
Masa_capa_mixta = 3×10^{16}

masa de capa mixta agua con
organismos y DDT.

Otras condiciones de simulación:

$\Delta t = 0.05$ años; tiempo inicial = 1940; tiempo final = 2020; Algoritmo de Integración: Euler.

Figura 23. Simulación del quinto prototipo. DDT aplicado y cantidad de DDT en el suelo.



La aplicación del DDT corresponde a la línea roja y la línea azul representa la cantidad acumulada del DDT en el suelo, se puede observar como a pesar de haber disminuido totalmente la aplicación del DDT en el año 1995 aproximadamente, todavía para el año 2010 quedan residuos de DDT. Las medidas están en miles de toneladas.

Figura 24. DDT en los peces.



Uno de los objetivos del modelo es el de mostrar los efectos del DDT en los seres vivos, y observar los efectos de aplicar DDT en el suelo; como se puede observar por medio de esta gráfica, a pesar de que alrededor del año 1995 el DDT deja de ser aplicado, para el año 2010 aun quedan grandes cantidades de DDT en los peces. Al igual que en la anterior, esta gráfica muestra el DDT en peces expresado en miles de toneladas.

De las simulaciones anteriores se observa cómo las acciones que se tomen pueden tener efectos colaterales adversos en el futuro, es un claro ejemplo del comportamiento contraintuitivo de los sistemas complejos; es muy común ver ejemplos como estos en el manejo del medio ambiente, por lo tanto se hace necesario un estudio mas profundo de los posibles efectos que pueda tener la aplicación de determinada política ambiental a la hora de llevarla a cabo.

○ **Consideraciones para el n-esimo prototipo.** La comprensión y representación de un fenómeno desde una perspectiva dinámico – sistémica, constituye un proceso siempre inacabado; constantemente

habrá preguntas por responder que demanden un nuevo prototipo que amplíe la cobertura del fenómeno en estudio, por ejemplo es posible preguntarse qué pasa con el resto de seres vivos en la cadena alimenticia cómo los afecta el DDT; los efectos sobre los seres humanos, etc.

Un supuesto básico del modelamiento sistémico, es que asume la búsqueda de un modelo, sin perder la visión holista, es decir, sabiendo que concluido éste hay que derribar sus fronteras para integrar otros elementos, con lo cual no habrá nunca un modelo acabado.

Y a pesar de lo siempre inacabado del modelo, el proceso de comprenderlo o llevarlo a cierto nivel de desarrollo, aporta al cambio y a la construcción de los modelos mentales del modelador. Además, según la evaluación del mismo, puede constituir en un útil e importante apoyo para diseñar y experimentar políticas que, aunque no siempre reflejen posibles y confiables resultados desde una perspectiva cuantitativa, sí pueden serlo desde una perspectiva cualitativa.

El proceso de comprensión al cual se ha hecho mención, deja abierta la posibilidad de que la utilidad del modelo trascienda el ámbito de la toma de decisiones exactas (en lo cuantitativo) al del aprendizaje. En este caso, lo útil es la concientización sobre los efectos del DDT y el tiempo que este continua siendo dañino aun después de parar su aplicación.

4.3.4 Aplicación de la metodología a los demás modelos ambientales.

Toda la metodología antes descrita, es la que se utilizó para desarrollar todos los modelos que se encuentran dentro de la herramienta, no se considera necesario colocar paso a paso la elaboración de cada uno de esos modelos en este documento escrito, ya que es exactamente igual que con el modelo del DDT antes descrito.

Se identifican tres grandes grupos de modelos dentro de la herramienta de acuerdo a su origen:

Tabla 4. Tipos de modelos según su origen.

Origen del modelo (tipo de trabajo)	Descripción
Reproducción modelos Libro Modeling the Environment de Andrew Ford.	Reproducción de todos los modelos y aplicaciones de la Dinámica de Sistemas al medio ambiente que se encuentran dentro del libro utilizando.
Reproducción de tesis y trabajos investigativos que utilicen la Dinámica de Sistemas para estudiar fenómenos de tipo ambiental.	Reproducción del modelo de "Efectos de Contenidos Húmedos en Rellenos Sanitarios". Es una Tesis de Grado presentada en la facultad de Ingeniería y Administración del Instituto de la Fuerza Armada de Tecnología (En Estados Unidos) para obtener el título de Master de Ciencias en Ingeniería y Administración Ambiental. [Craig, P. Eck.]
Elaboración de modelos piloto o prototipos de diferentes fenómenos ambientales con los estudiantes de pregrado y la especialización en ingeniería ambiental.	Estos modelos a los cuales se hace referencia abarcan entre otros los siguientes temas: Cadena Alimenticia. Cambio Global del Medio ambiente. Modelos Depredador-Presa. Ciclos Básicos de la Naturaleza. Contaminación de aguas. Contaminación aire. Deforestación.

	<p>Efecto Invernadero. Manejo de Basuras. Se denominan prototipos o modelos piloto pues no son elaborados por expertos en los temas mencionados y no tienen la rigurosidad de un trabajo investigativo como en caso del modelo de "Efectos de Contenidos Húmedos en Rellenos Sanitarios".</p>
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Todos estos modelos se encuentran dentro de la herramienta y pueden ser utilizados por los estudiantes para aprender mas sobre el tema y sobre la Dinámica de sistemas, en ellos podrá probar políticas cambiándole valores a los parámetros de los modelos y observando su comportamiento.

A continuación se muestra el desarrollo de la herramienta que sirve como plataforma para crear y mostrar modelos de dinámica de sistemas de manera interactiva y como parte de lo que se denomina Documentos Interactivos.

4.4 DESARROLLO DE LA HERRAMIENTA SOFTWARE

La herramienta software GAIA fue creada pensando desde un comienzo en el objetivo general que guía el desarrollo del proyecto, es decir crear un software que permita al usuario aprender Dinámica de Sistemas y aplicarla al estudio del medio ambiente. Para obtener una herramienta de calidad, fue necesario entonces plantear unos requisitos que representen las diferentes facilidades que ofrece la herramienta, teniendo en cuenta que estos requisitos deben tener el fin de cobijar los objetivos general y específicos planteados desde el inicio del proyecto.

Es así como surge la idea de combinar cualidades de diferentes tipos de herramientas orientadas hacia el aprendizaje asistido por computador. Al terminar la sección anterior se menciona la herramienta software que sirve como plataforma para mostrar y crear modelos de Dinámica de Sistemas, los cuales forman parte integral de unos documentos que se denominan Documentos Interactivos.

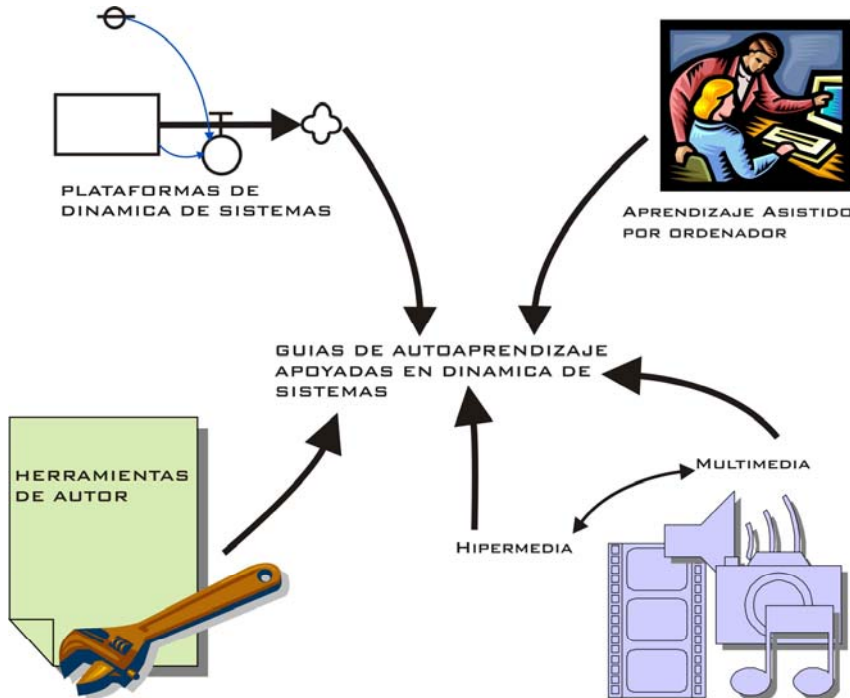
Los documentos interactivos pretenden conjugar y reunir en un mismo concepto la idea de aprendizaje asistido por computador, elementos multimediales, plataformas de dinámica de sistemas y herramientas de autor. Este concepto de Documento Interactivo, surgió como una solución a los diversos requisitos y objetivos planteados durante la fase de Identificación y definición de requisitos y la fase de análisis.

4.4.1 ¿Qué es GAIA?

Es una Herramienta software para el estudio de fenómenos ambientales, mediante el modelado y la simulación con dinámica de sistemas; que contendrá información teórica de Dinámica de Sistemas, Fenómenos Ambientales y modelamiento de este tipo de fenómenos con DS; además de ejemplos didácticos, aplicaciones (modelos ejemplo) y elementos multimediales.

La herramienta GAIA tiene las características propias de una *Guía de autoaprendizaje apoyada en Dinámica de Sistemas*, este concepto surge por las necesidades del presente proyecto y como ya se dijo en el marco teórico, combina los conceptos de Aprendizaje Asistido por Computador, herramientas de autor, y plataformas de Dinámica de Sistemas. Para hacerse una mejor idea de las características del software y las tecnologías que implementa, observe la Figura 25.

Figura 25. Esquema de las tecnologías y conceptos que utilizan las Guías de autoaprendizaje apoyadas en dinámica de sistemas



Con base en esto se obtienen las siguientes características (las cuales corresponden a los requisitos del software):

- Operar con toda la información recopilada (modelos, textos, audio, video, y en general cualquier tipo de información en cualquier formato).
- Diseñar y ejecutar experimentos en un módulo de Laboratorio de Dinámica de Sistemas con los ejemplos ya existentes y con los modelos creados por el usuario.
- Elaborar sus propios modelos (Modelar).
- Mantener continuamente (correcciones y complementos) de todos los contenidos del software mediante:
 - Facilidades para modificar e incluir nueva información.

- Facilidades para incluir nuevos experimentos y modelos y en general para cambiar cualquier contenido del software.
- Crear sus propios contenidos (documentos interactivos), es decir brindar la capacidad de generar sus propias guías de estudio y aprendizaje.
- Capacidad de transportar los documentos interactivos creados y compartirlos con otros usuarios de la herramienta.
- Facilidades de Navegación por los contenidos que se encuentren dentro de la herramienta (múltiples entornos al mismo tiempo).
- Amplias opciones de búsqueda avanzada para localizar de manera eficiente la información que requiera el usuario.
- Como soporte didáctico se incluyen dos tutoriales:
 - Un tutorial por medio del cual el usuario aprende los conceptos básicos del proceso de modelado con Dinámica de Sistemas.
 - Tutorial que enseña los conceptos de Flujos y Niveles y la utilización de los elementos básicos del diagrama de Forrester.
- Elementos de ayuda que permiten el aprendizaje de las diferentes opciones y comandos que tiene la herramienta.
- Creación de múltiples usuarios con independencia de contenidos, es decir cada usuario puede tener los contenidos que desee independientes de los demás usuarios, es como tener varias guías de autoaprendizaje en una sola herramienta.
- Opciones de administración, existe un súper usuario que es el que permite crear y borrar nuevos usuarios y además cada uno de los usuarios puede modificar su nombre de usuario y contraseña.
- Un glosario con los conceptos más importantes enunciados en los contenidos de la herramienta; y facilidades para que el usuario añada nuevos conceptos.

4.4.2 Tecnologías utilizadas para el desarrollo del software.

Para lograr que la herramienta cobijara todos las características antes mencionadas, se necesito de la puesta en practica de varias tecnologías informáticas y plataformas de desarrollo. En la Figura 26, se puede observar fácilmente cuales tecnologías ayudaron en la creación de GAIA.

Figura 26. Tecnologías y herramientas utilizadas para desarrollar GAIA.



A lo largo de la siguiente sección se describe el proceso llevado a cabo para poder llevar a conclusión el proyecto, allí encontrará la manera como fueron aplicadas estas tecnologías durante el desarrollo del proyecto.

4.4.3 El proceso de desarrollo de GAIA.

Como se mencionó con anterioridad, se utilizó el proceso unificado de Desarrollo de Software para la elaboración de la herramienta software GAIA.

La metodología del proceso unificado de desarrollo de software es considerada adecuada para la elaboración de la herramienta pues esta dirigida por casos de uso, es decir teniendo siempre en cuenta los requisitos funcionales que este debe tener para cumplir sus objetivos. Este proceso se llevo a cabo a lo largo de un conjunto de fases e iteraciones cada una de las cuales esta enmarcada dentro de unos flujos de trabajo bien definidos (Requisitos, Análisis, Diseño, Implementación,

Prueba); al final de cada una de las fases se obtiene un conjunto de artefactos; es decir, ciertos modelos o documentos que han sido desarrollados hasta alcanzar un estado predefinido; además al final de cada fase, se obtiene un producto software que cada vez se acerca mas al cubrimiento de los objetivos con los que fue creado, es decir a satisfacer los casos de uso que debe satisfacer. Es necesario mencionar que lo importante en el proceso unificado de desarrollo no es recopilar un conjunto de documentos, diagramas y otro tipo de artefactos y seguir un proceso rígido, sino mas bien seguir un proceso iterativo que muestre en cada paso una aproximación incremental al producto deseado y mostrar un comportamiento flexible, lo cual quiere decir que los casos de uso especificados desde un comienzo no son rígidos sino que mas bien se van moldeando con el paso de cada iteración.

La idea no es mostrar todos los artefactos y las versiones de todas las iteraciones realizadas a lo largo del proceso de desarrollo, lo cual implicaría una extensión bastante considerable del documento, mas bien a continuación se muestran las iteraciones relevantes de cada fase y los hitos (versiones) mas importantes elaboradas hasta la conclusión del software.

• **Fase de Inicio (Conceptualización).** La fase de inicio se caracteriza por la definición de los casos de uso críticos que guiarán el desarrollo del software a lo largo de su proceso de vida. Para ello primero se enuncian una lista de requisitos funcionales y a partir de esa lista se elabora un Modelo de Casos de Uso. Cabe mencionar que no todos los casos de uso son escritos en la fase de inicio, mas bien la labor es posible describirla como un taller de requisitos que permite bosquejar a grandes rasgos las características de la herramienta.

La lista de requisitos corresponde a la lista de características enunciada anteriormente en el numeral 4.4.1.

A partir de estos requisitos, se elaboran los casos de uso críticos más importantes.

Figura 27. Principales casos de uso (críticos).



o **Descripción del actor Usuario.** Como se puede observar en el diagrama de casos de uso críticos, solo existe un actor, el Usuario quien es la persona que interactúa con la herramienta, en este caso el usuario hace las veces de aprendiz y es el que crea y utiliza todos los contenidos que se encuentran dentro del software.

○ **Especificación del caso de Uso Iniciar Sesión.** El Usuario utiliza el caso de uso Iniciar Sesión para acceder a la herramienta con su nombre de usuario y contraseña, de tal forma, el usuario solo podrá interactuar con los artículos que el ha creado; es así como la herramienta permite que cada usuario mantenga dentro del software documentos sobre el tema que desee sin interferir en los demás artículos, lo cual permite que en una misma herramienta existan guías de autoaprendizaje de diferentes temas y diferentes usuarios.

Los contenidos iniciales de Gaia estarán disponibles solo para el Usuario tipo Administrador de la Herramienta. Nombre de usuario: usuario1 y contraseña: contraseña1.

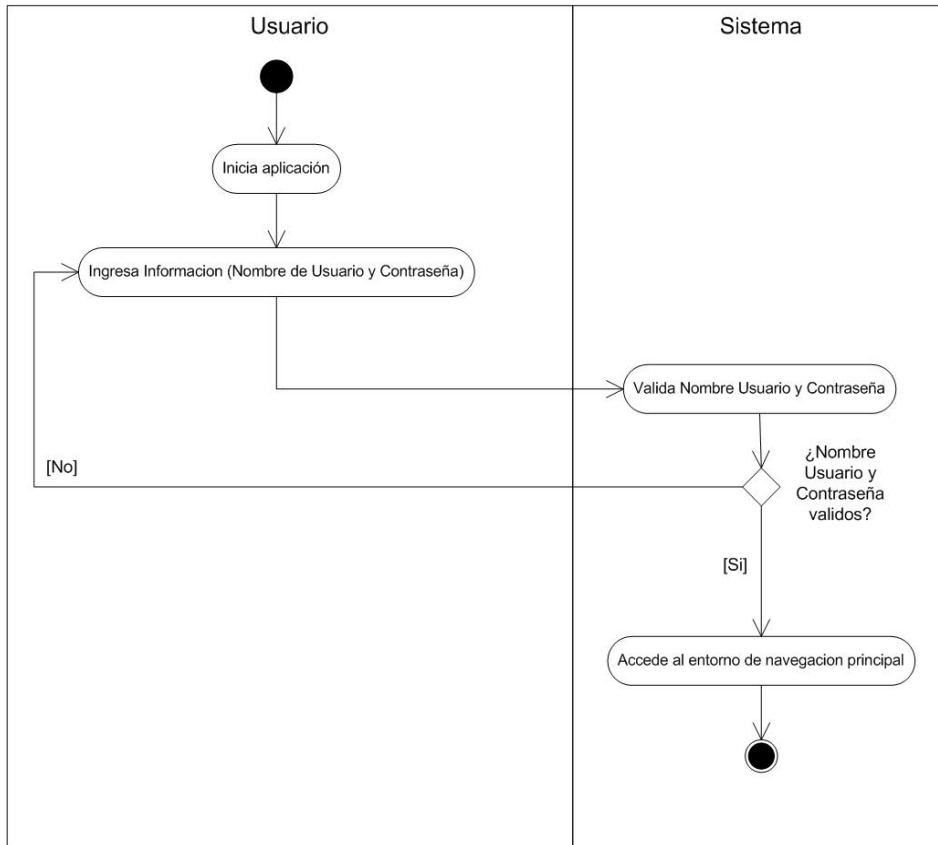
➤ **Descripción paso a paso inicial.** El usuario inicia la herramienta y aparece la pantalla de inicio de sesión, a partir de ahí los pasos que debe seguir son:

1. Ingresar datos que le pide el sistema²⁵, nombre de usuario y contraseña.
2. Pide al sistema que valide las entradas.
3. El sistema valida y busca en la base de datos para ver si la información ingresada es correcta, a partir de esto filtra los registros que corresponden a los artículos del usuario.

➤ **Diagrama de actividades del caso de uso Iniciar Sesión.**

²⁵ NOTA DE LOS AUTORES: para efectos de referirnos a la herramienta software durante el proceso de desarrollo, utilizaremos indistintamente las palabras sistema y herramienta.

Figura 28. Diagrama de actividades del caso de uso Iniciar sesión.



o **Especificación del caso de Uso Modelar.** El caso de uso modelar permite al usuario elaborar sus propios modelos de Dinámica de Sistemas. Para ello es necesario que el usuario este registrado y halla ingresado correctamente por la pantalla de Inicio de Sesión. Una vez allí dentro del entorno principal de navegación podrá acceder al laboratorio de modelado que le permite elaborar sus propios modelos.

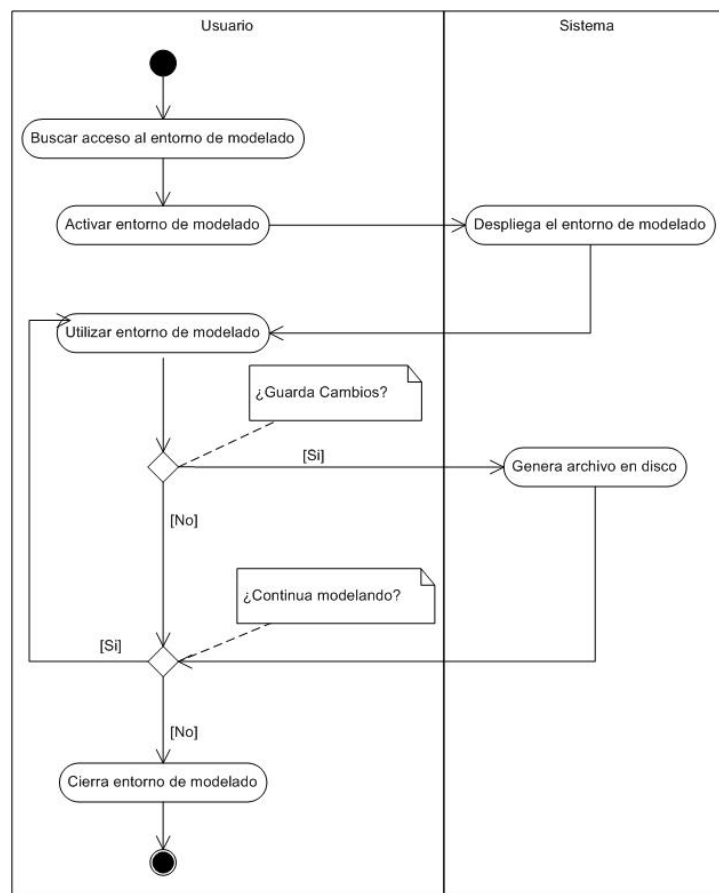
➤ **Descripción paso a paso inicial.** Luego de haber ingresado al sistema el usuario puede realizar varias acciones dentro del software, una de ellas es abrir el entorno de modelado con Dinámica de Sistemas, para ello debe:

1. Buscar acceso al entorno de modelado.

2. Activar el entorno de modelado.
3. Utilizar el entorno de Modelado.
4. Guardar o no el archivo.
5. Salir del entorno.

➤ **Diagrama de actividades del caso de uso Modelar.**

Figura 29. Diagrama de actividades del caso de uso Modelar.



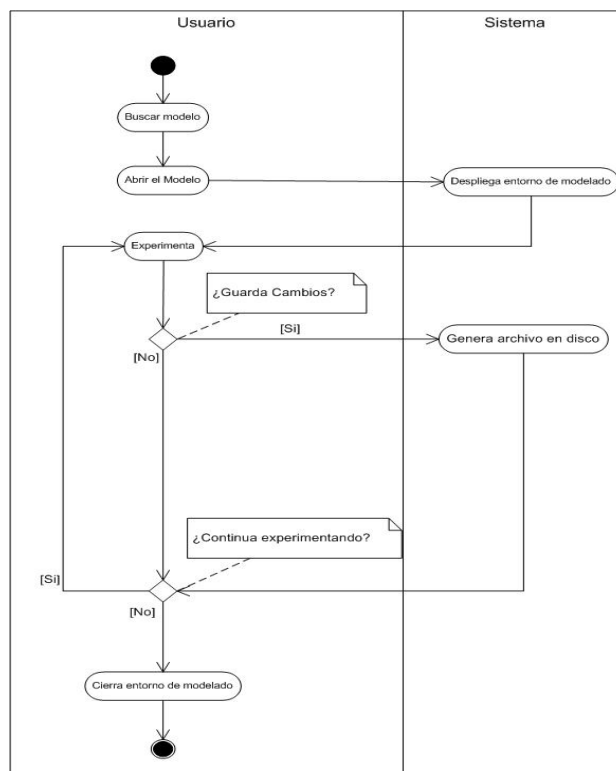
- **Especificación del caso de uso Experimental.** Para que el usuario pueda experimentar deben existir modelos dentro de la base de datos o debe haber creado alguno modelo, Obviamente el usuario debe estar en el entorno principal de navegación y haber iniciado exitosamente sesión.

➤ **Descripción paso a paso inicial.** Una vez dentro del entorno principal de navegación el usuario debe:

1. Buscar el modelo que deseé y que se encuentre dentro de la Base de Datos de la herramienta o en el Disco Duro.
2. Abrir el modelo, lo cual ocasionará que este modelo se abra dentro del entorno de modelado que le permite realizar experimentos con el modelo.
3. Al terminar con el proceso de experimentación, el usuario puede cerrar el entorno de Modelado y seguir navegando dentro de la herramienta.

➤ **Diagrama de actividades del caso de uso Experimentar.**

Figura 30. Diagrama de actividades del caso de uso Experimentar.



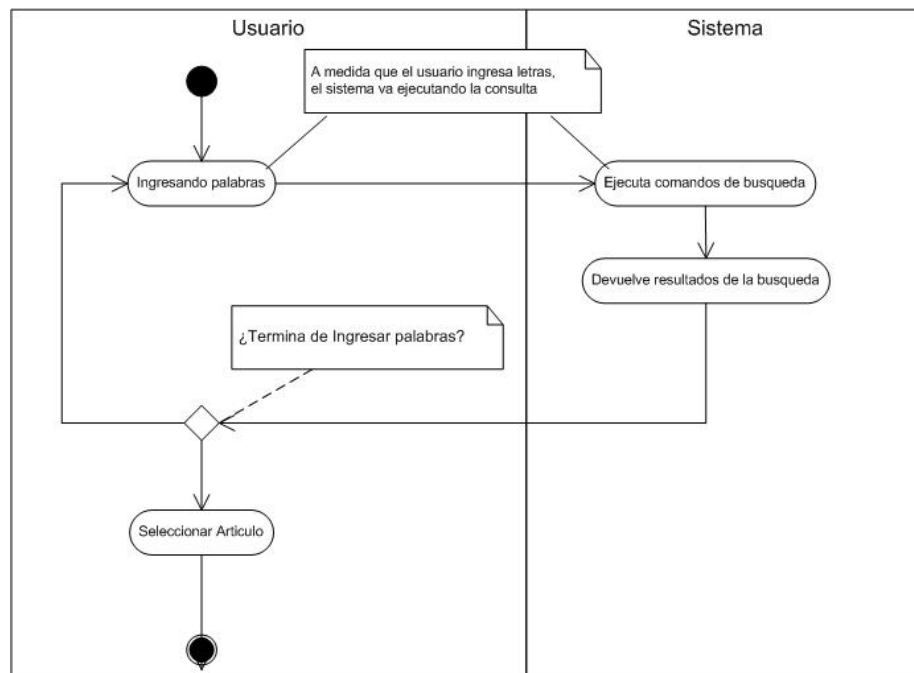
○ **Especificación del caso de uso Buscar.** Este caso de uso describe el proceso de búsqueda de artículos dentro del sistema.

➤ **Descripción paso a paso inicial.** Para realizar una búsqueda el usuario debe encontrarse en el entorno principal de navegación y seguir los siguientes pasos:

1. Ingresar palabras que pertenecen al Título Principal.
2. Ejecutar la búsqueda.
3. Seleccionar artículo.

➤ **Diagrama de actividades del caso de uso Buscar.**

Figura 31. Diagrama de actividades del caso de uso Buscar.



○ **Especificación del caso de uso Usar Tutoriales.** Este caso de uso corresponde a la selección de alguno de los tutoriales que se encuentran

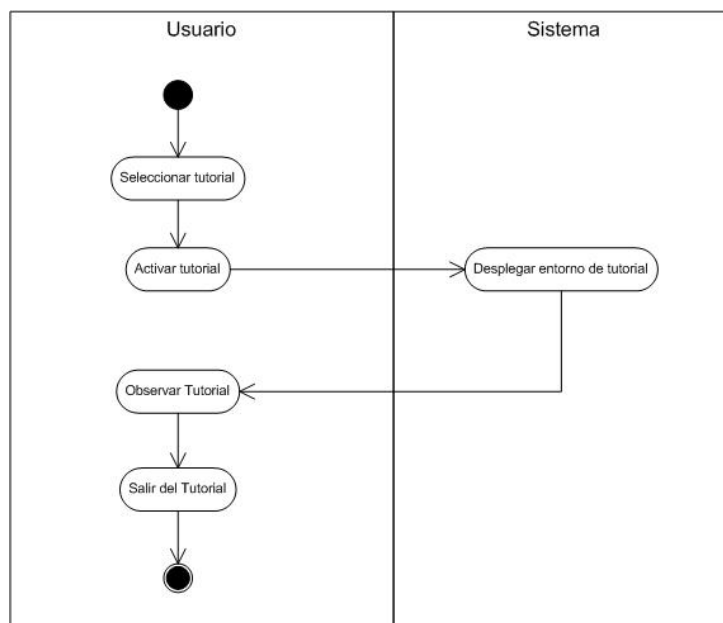
dentro de la herramienta y su activación para que de esta manera el tutorial enseñe su contenido al usuario.

➤ **Descripción paso a paso inicial.** Como en los casos de uso anteriores y los que se verán más adelante, es necesario que el usuario se encuentre dentro de la pantalla principal de navegación, el flujo de eventos esta dado mas o menos por los siguientes pasos.

1. Seleccionar algún tutorial.
2. Activar el tutorial.
3. Observar el tutorial.

➤ **Diagrama de actividades del caso de uso Usar Tutoriales.**

Figura 32. Diagrama de actividades del caso de uso usar tutoriales.



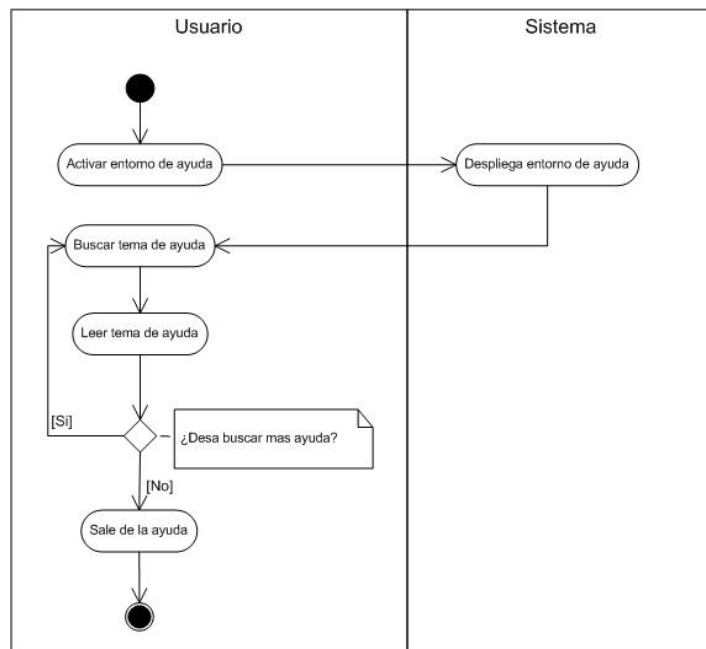
○ **Especificación del caso de uso Usar Ayuda.** El usuario utiliza este caso de uso para obtener información acerca del uso del sistema y el uso de las diferentes opciones que este ofrece.

➤ **Descripción paso a paso inicial.** La descripción general del caso de uso Usar ayuda puede expresarse mediante los siguientes pasos:

1. Activar el entorno de ayuda.
2. Interactuar con el entorno de ayuda.
3. Cerrar Ayuda.

➤ **Diagrama de actividades del caso de uso Usar Ayuda.**

Figura 33. Diagrama de actividades del caso de uso usar ayuda.



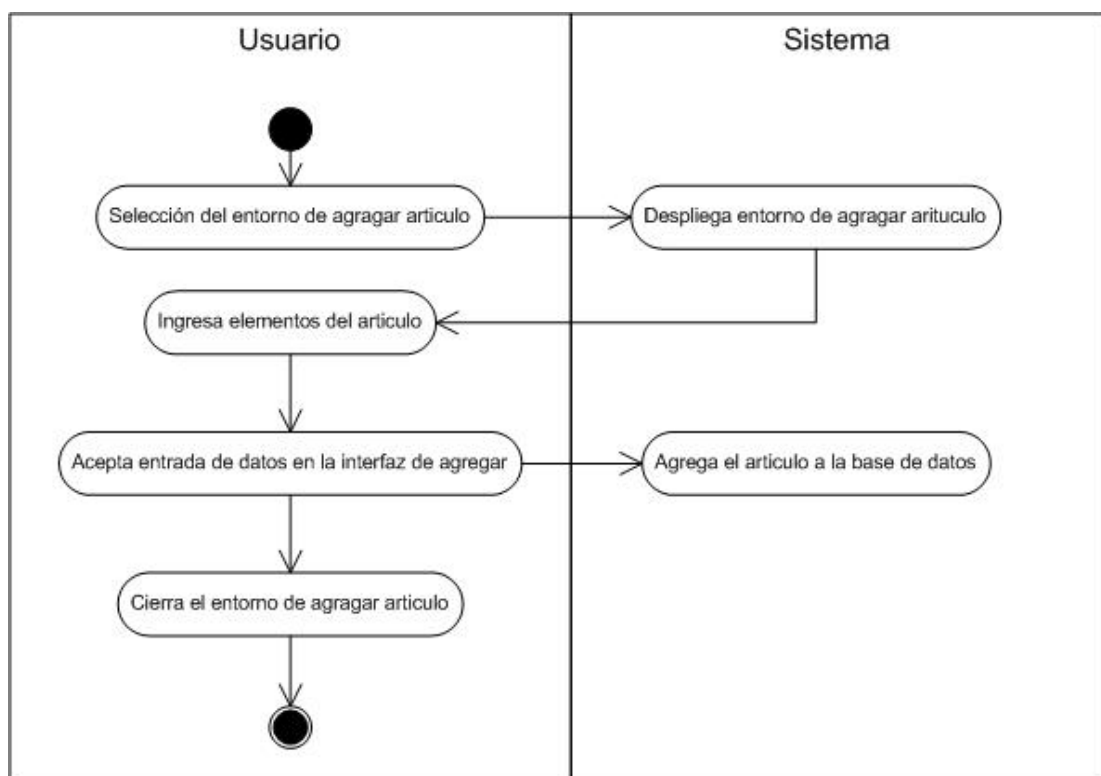
○ **Especificación del caso de uso Agragar Articulo.** Este caso de uso corresponde a la inclusión de un articulo dentro de la herramienta.

➤ **Descripción paso a paso inicial.** La descripción general del caso de uso agregar artículo puede expresarse mediante los siguientes pasos:

1. Selección de la información que el usuario va a agregar.
2. Abrir un entorno de agregar artículo.
3. Ingresar la información del artículo.
4. Aceptar la información digitada.
5. salir del entorno.

➤ **Diagrama de actividades del caso de uso Agregar Artículo.**

Figura 34. Diagrama de actividades del caso de uso agregar artículo.



○ **Especificación del caso de uso Navegar (Múltiples entornos de trabajo).** Este caso de uso de alguna forma puede considerarse como

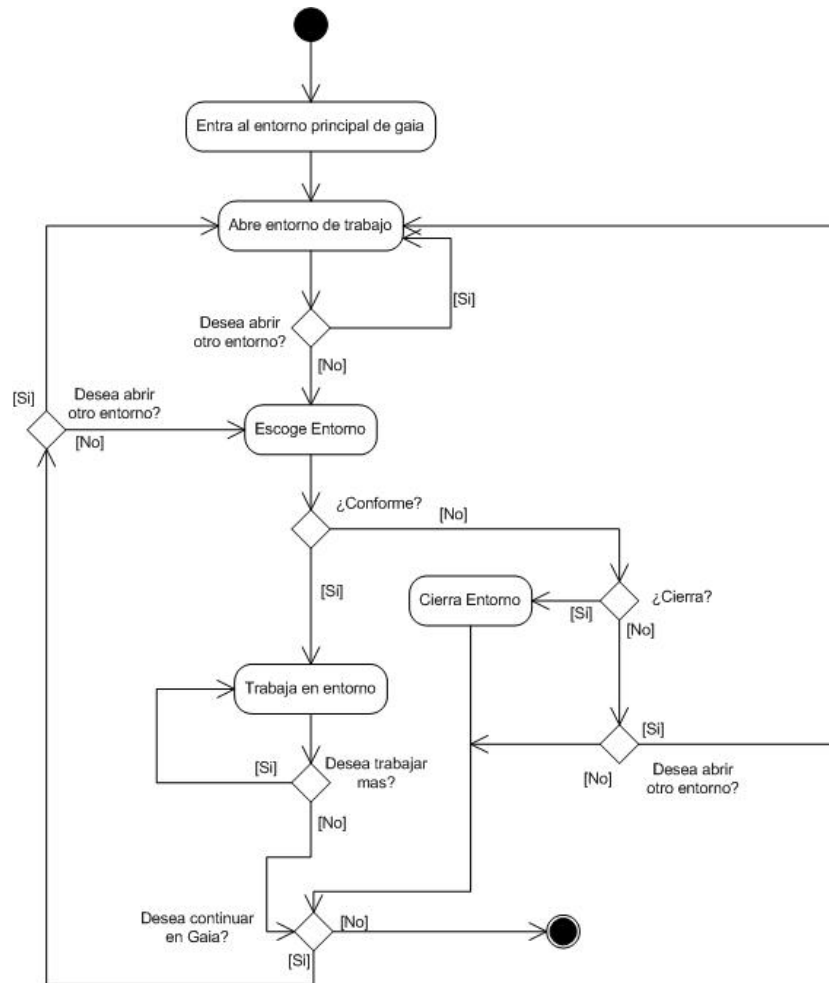
abstracto, pues normalmente se considera la navegación del usuario por el sistema como algo que surge simplemente cuando se está ya en estados altos o avanzados en el proceso del desarrollo del software; por el contrario se quiere hacer hincapié en este proceso de navegación, de forma tal que guíe el trabajo realizado para elaborar la herramienta; la idea es que el usuario puede mantener N cantidad de entornos de trabajo al mismo tiempo, donde desarrolle el resto de casos de uso, todo esto sin que ninguno de los casos de uso interfieran entre si.

➤ **Descripción paso a paso inicial.** Como se ha explicado antes, el usuario siempre estará navegando en el sistema, pero la navegación dentro de gaia debe permitir multiples entornos, los siguientes pasos explican mejor la idea:

1. Iniciar Gaia.
2. Entrar al entorno principal de navegación.
3. Abrir algún entorno(s) de trabajo (que corresponde a algun otro caso de uso).
4. Cerrar entorno(s) que desee.
5. Repetir cuantas veces quiera pasos 3 y 4.
6. Salir de Gaia

➤ **Diagrama de actividades del caso de uso Navegar.**

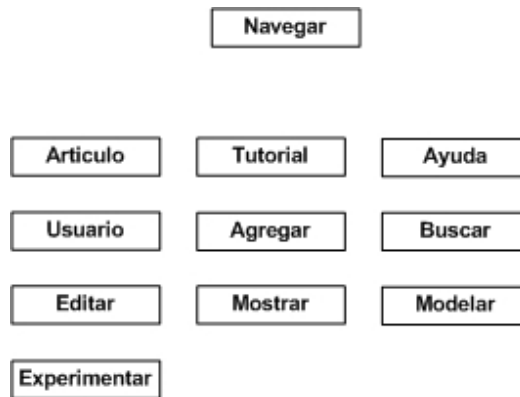
Figura 35. Diagrama de actividades del caso de uso navegar.



Como se dijo antes, este caso de uso es tenido siempre en cuenta a la hora de desarrollar el software y cada elaboración de los casos de uso tiene en cuenta la navegación múltiple, es decir permitir múltiples entornos al mismo tiempo.

• **Modelo del Dominio Inicial.** El modelo del dominio proporciona un diccionario visual del vocabulario y conceptos del dominio a partir de los cuales se pueden nombrar algunas cosas del diseño de software. A continuación se muestra un conjunto de clases inicial generados para el Dominio de la aplicación.

Figura 36. Modelo del Dominio Inicial.



• **Glosario de Términos.**

Tabla 5. Glosario de términos fase de inicio.

Término	Definición e Información
Artículo	Corresponde a un documento interactivo que posee texto, fotos, video, audio, modelos de D.S., y cualquier otro tipo de archivo, además de tener una estructura bien definida.
Modelar	Hace referencia al desarrollo de modelos de Dinámica de sistemas.
Estructura	Se refiere a la organización y orden que tiene el artículo y que es generada por el usuario, le permite colocar texto con indentación de cualquier nivel mediante la tabla estructura, teniendo un párrafo padre y uno hijo.
Experimentar	Manipular y hacer pruebas (aplicar políticas) con los modelos de dinámica de sistemas

• **Muestra de los artefactos del Proceso Unificado y evolución temporal.**

En la siguiente tabla se muestran los artefactos que serán utilizados a lo largo de las fases y sus respectivas iteraciones.

Tabla 6. Artefactos y su evolución durante el desarrollo de GAIA. C – comenzar; R – refinar.

Disciplina	Artefacto	Inicio	Elab.	Const.	Trans.
Modelado del negocio	Modelo del Dominio	C	R		
Requisitos	Modelo de Casos de Uso	C	R		
	Diagramas de actividades	C	R		
	Glosario	C	R		
	Prototipos Interfaz usuario	C	R		
Diseño	Modelo de Diseño		C	R	
	Modelo de Datos		C	R	
Implementación y pruebas	Modelo de Implementación		C	R	R
	Modelo de Pruebas		C	R	

- **Prototipos Iniciales de la Interfaz de usuario.** En base en los casos de uso, se desarrollan los siguientes prototipos de interfaz de usuario:

Figura 37. Algunos prototipos iniciales de la interfaz de usuario.





• **Fase de Elaboración.** En la fase de inicio solo se realizó una iteración que dio como resultado un bosquejo general de los casos de uso y una primera aproximación al modelo del dominio. Estos artefactos son retomados en esta fase, la cual se puede definir brevemente en la siguiente frase:

Construir el núcleo central de la arquitectura, definir la mayoría de requisitos, y estimar la planificación y los recursos globales, además de implementar las clases de software que se tengan definidas.

Puede parecer que en la primera iteración en la fase de inicio se abarcó demasiado y se intentó definir la mayoría de los requisitos, no obstante la idea es solo dar una visión general de los casos de uso críticos; en el proceso de elaboración a partir de una serie de iteraciones, se irán perfeccionando estos casos de uso a la vez que se van convirtiendo en código ejecutable.

En este punto es válido mencionar que durante el desarrollo del proyecto, la delimitación entre la fase de elaboración y la de construcción no está realmente definida, pues en las iteraciones que conforman ambas fases se obtiene productos software que cubren diversos casos de uso, la diferencia entre una fase y la otra es que en la segunda (la construcción), ya no se definen mas casos de uso, sino que mas bien se refinan las implementaciones de código para que el software cubra aun mejor los casos de uso especificados y de esta forma los requisitos iniciales queden completamente satisfechos.

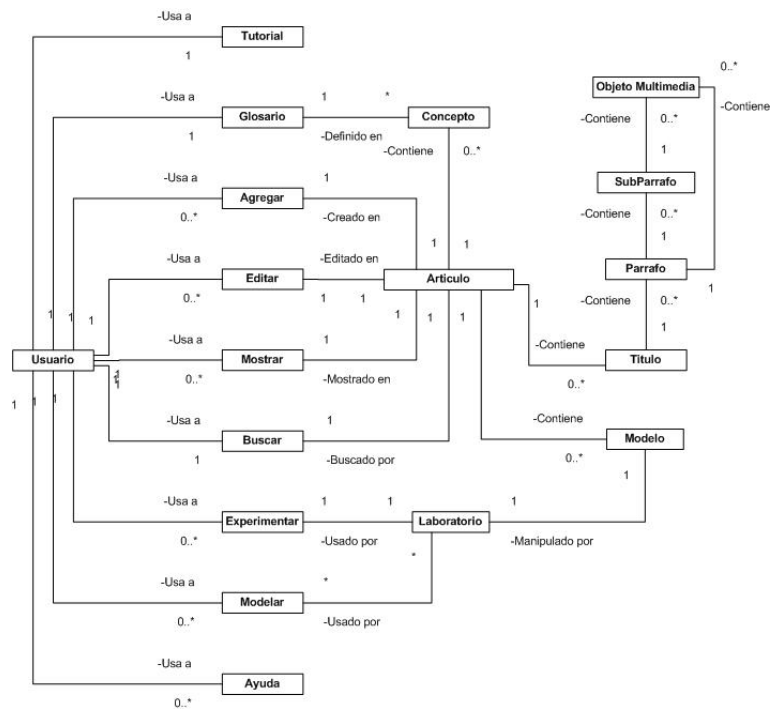
Es importante mencionar que en cada iteración se realizaron pruebas con el fin de comprobar la funcionalidad con la que fue diseñada e implementada la porción de software que cubre parcial o totalmente determinado caso de uso. En la fase de construcción y de transición el proceso de pruebas es más riguroso y se hacen pruebas de integración del sistema, de estabilidad, robustez entre otras.

○ **Primera Iteración.** En esta primera iteración dentro de la fase de elaboración, se refina el Modelo del Dominio Inicial y se especifica con más detalle el caso de uso agregar artículo y el caso de uso de edición de articulo, el cual puede ser considerado como una extensión del caso de uso agregar; además se realizan las primeras tareas de desarrollo de software ejecutable.

Para refinar la especificación del caso de uso agregar artículo se utiliza Un Diagrama de Secuencia del Sistema.

➤ **Refinamiento del Modelo del Dominio.** En la primera aproximación hecha al modelo del dominio solo se colocaron unas pocas clases conceptuales. En esta iteración, se añaden nuevos elementos al modelo y se especifican las asociaciones entre los elementos. El resultado es un modelo de Dominio bien completo que muestra las relaciones entre las llamadas clases conceptuales y que muestra una visión un poco más amplia y clara del sistema. Este modelo de dominio ayuda a responder (en parte) la pregunta ¿Cómo se utilizará el sistema?

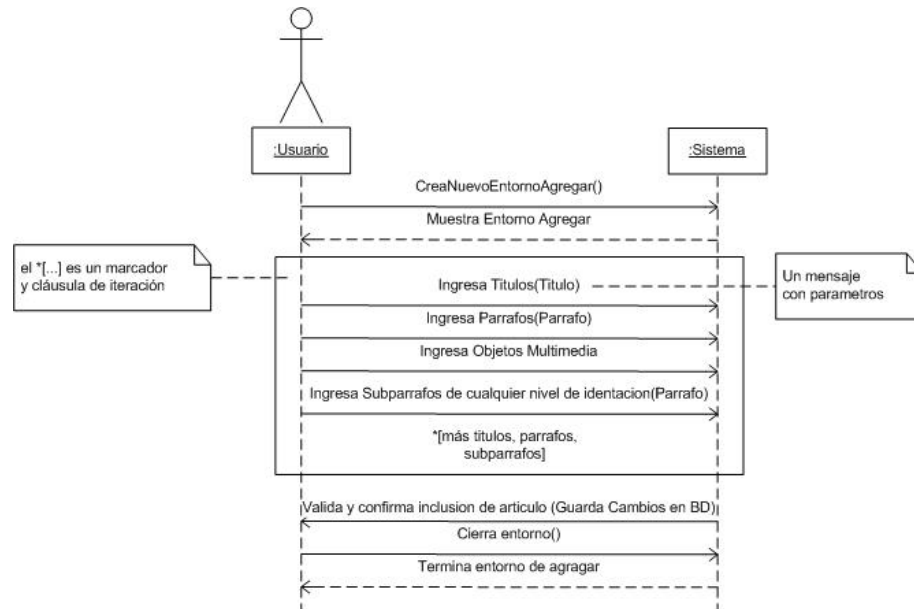
Figura 38. Modelo del Dominio Refinado.



➤ **Diagrama de Secuencia del sistema para el caso de uso agregar artículo (DSS).** Se muestra aquí el escenario principal de éxito del caso de uso, la utilidad de los diagramas de secuencia es mostrar el proceso de

interacción y los eventos que el usuario emite sobre el sistema y su correspondiente respuesta.

Figura 39. DSS para el caso de uso agregar artículo.



➤ **Primer acercamiento al modelo de Datos.** Con base en los casos de uso y en el modelo del dominio, se tiene información suficiente como para hacer un primer diseño de la base de datos.

Primero, se define un objetivo general de la base de datos; este objetivo sería: Mantener los datos correspondientes a los artículos de los diferentes usuarios que forman parte de la herramienta software.

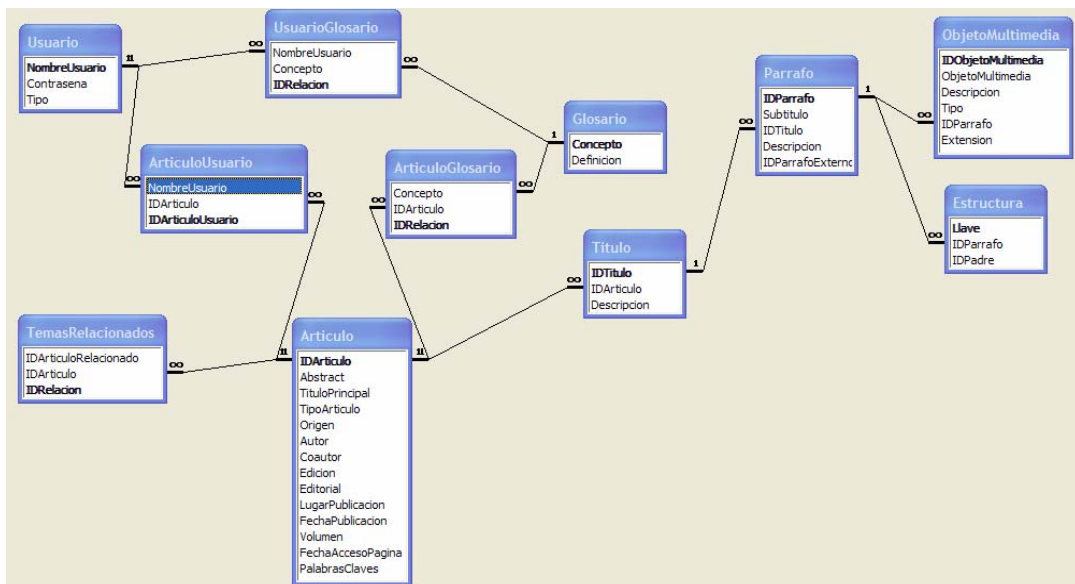
Además, para ayudar al cumplimiento del objetivo general, se definen unos objetivos específicos que guían el desarrollo de la base de datos.

- ✓ Es necesario tener la información básica del artículo (fuente, fecha de consulta, tipo de artículo, etc).

- ✓ Se debe llevar registro de los títulos, párrafos y subpárrafos de cada artículo.
- ✓ Se debe llevar un registro de la estructura del artículo.
- ✓ Se debe llevar un registro de los objetos multimedia.
- ✓ Debe permitir que cada usuario tenga sus propios artículos independientes del resto de usuarios.

Teniendo esto en cuenta y algunas de las clases conceptuales del modelo del dominio, se obtiene el siguiente diseño de la Base de Datos:

Figura 40. Modelo entidad – relación de la base de datos.

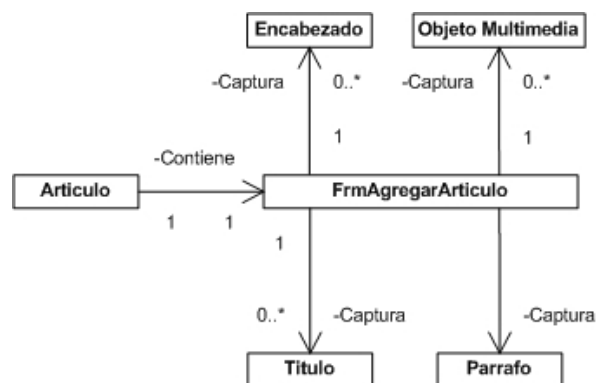


Como puede observarse, se encuentran algunas tablas que tienen los mismos nombres que las clases conceptuales del modelo del dominio; a partir de esta base de datos, ya es posible empezar a ponerle “un poco de músculo al sistema” y desarrollar código ejecutable que presente solución al caso de uso agregar artículo; para lo cual primero se realiza un diseño con el que se obtiene un diagrama de clases de diseño.

➤ **Diseño de objetos.** Después de la identificación de los requisitos y la creación de un modelo del dominio, se tienen las bases para hacer un modelo de clases del software, añadirle métodos y definir el paso de mensajes entre los objetos para satisfacer los requisitos.

Se definen en este diagrama las clases del software, sus responsabilidades y las relaciones entre ellas. Ésta iteración se centra en las clases software que cubren el caso de uso de agregar artículo.

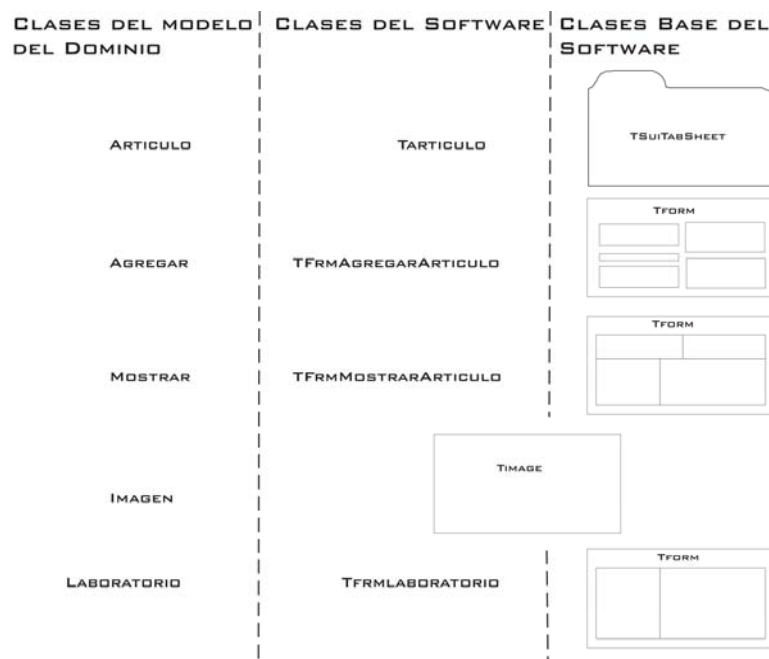
Figura 41. Algunas clases del dominio modificadas que intervienen en el caso de uso agregar artículo y que aparecen en el modelo de clases de software.



Observe que ahora aparece una clase llamada FrmAgregarArticulo, que contiene a las otras clases mencionadas en la grafica anterior; esta clase como es de esperar hereda de TForm (formulario delphi) lo cual permite instanciar un objeto de esta clase tantas veces como se quiera, por lo tanto, contribuye a realizar una interfaz de usuario como la que se describe en el caso de uso navegar. De forma tal que las clases reales implementadas en el lenguaje de programación Delphi mezclen a algunas de las clases del modelo del dominio con algunas clases propias de delphi para el desarrollo de interfaces de usuario.

A continuación se explica aquí de manera general cómo estarán constituidas las clases del software y su relación con la interfaz del usuario:

Figura 42. Relación entre clases del dominio, clases software y las clases base para definir las.

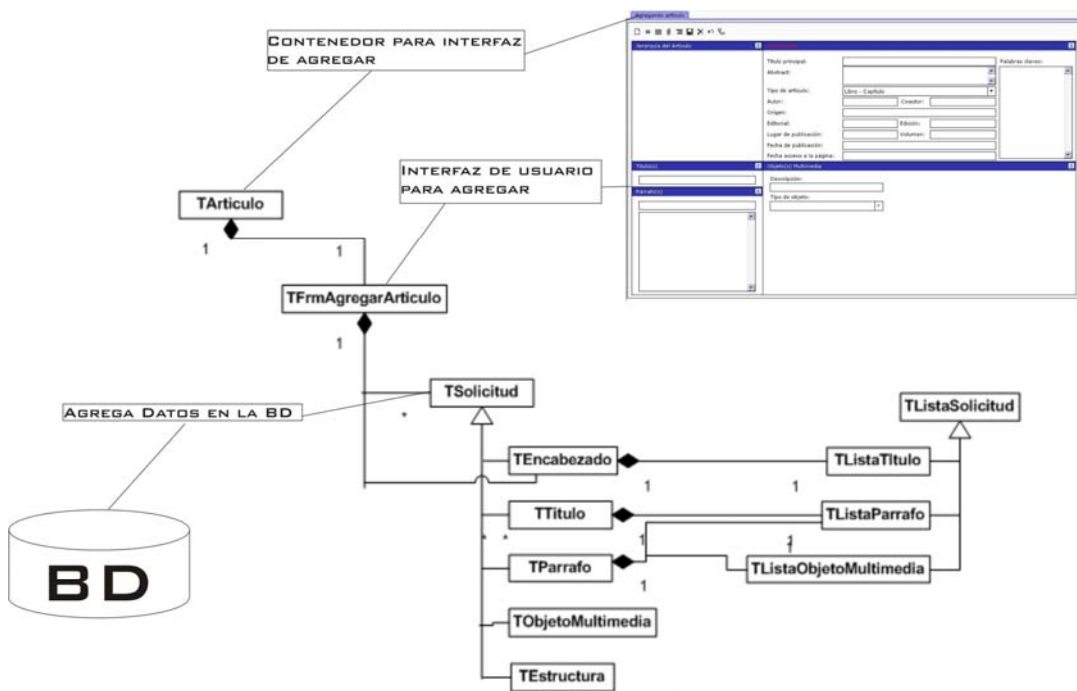


En la figura anterior se muestra la relación entre las clases del dominio y las clases software; y los componentes básicos para lograr el objetivo de tener múltiples entornos de trabajo. Cada entorno de trabajo está contenido dentro de un TArticulo que corresponde a la unidad fundamental de navegación dentro de Gaia; la clase TArticulo hereda de TSuiTabSheet y TArticulo es posible que contenga a TFrmAgregarArticulo, TFrmMostrarArticulo, TImage o a TFrmLaboratorio, o dicho más claramente, dentro de cada pestaña de la interfaz, podrá ver, agregar o

editar un artículo, además podrá abrir un entorno de laboratorio o alguna imagen.

En la figura 43 se muestra el diagrama de clases de software, su relación con la interface y la base de datos; como se puede observar algunos nombres cambian y surgen nuevas clases (TListaSolicitud, TListaTitulo, TListaParrafo, TListaObjetoMultimedia, TEstructura), además se reúnen las clases con base en una organización jerárquica y de composición, es decir con relaciones de herencia y de agregación o composición.

Figura 43. Relación entre clases del dominio, clases software y las clases base para definir las.



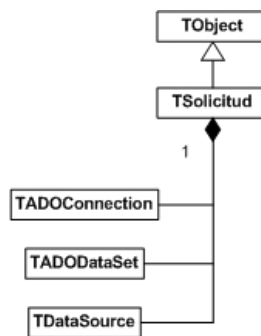
La convención de nombres en Delphi es colocar una T antes de cualquier clase. Es así entonces como surgen las siguientes clases del software:

- ✓ TArticulo que hereda de la Clase TSuiTabSheet que es una clase creada por *sunisoft* y que tiene facilidades para el cambio del tema

del entorno; TArticulo se convierte en la unidad fundamental del entorno de gaia, que sirve como contenedor de los entornos de agregar, editar, mostrar imágenes, mostrar articulo y del laboratorio de modelado.

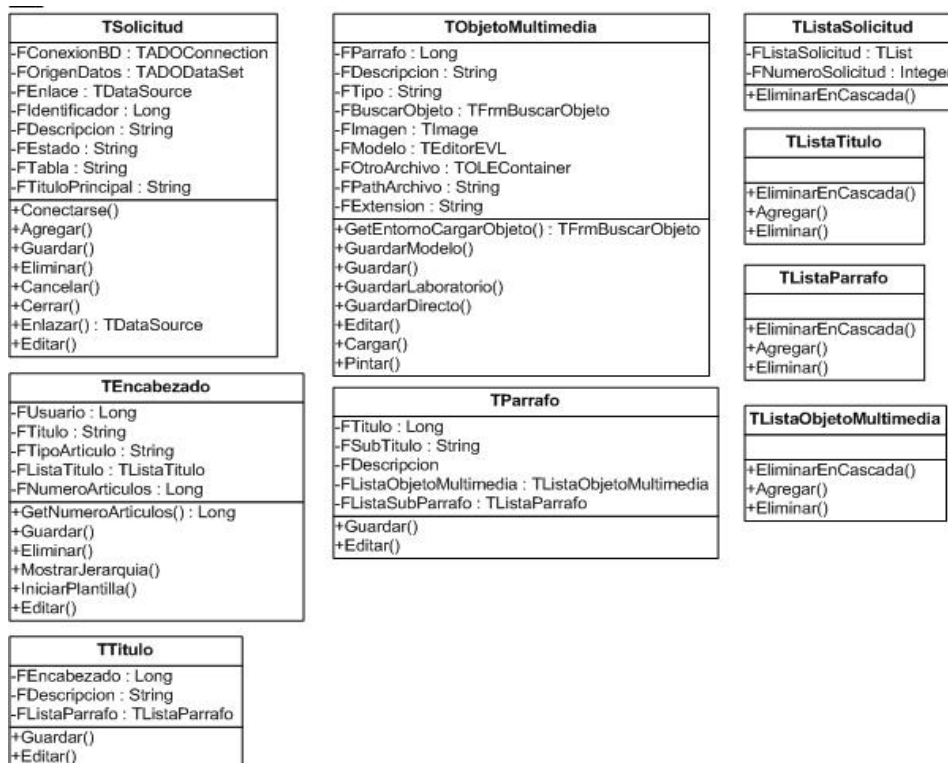
- ✓ TFrmAgregarArticulo hereda de TForm y corresponde al medio por el cual el usuario ingresa los datos para crear el articulo.
- ✓ TSolicitud permite la comunicación con la base de datos, agregar, consultar, eliminar, etc. Es una clase "madre" pues se especializa en TEncabezado, TTitulo, TParrafo, TObjetoMultimedia, TEstructura los cuales sobrescriben los metodos de agregar, editar, eliminar y guardar de la clase TSolicitud, esta ultima clase hija mencionada (TEstructura), permite la creación de la estructura del usuario, sobrescribe el método guardar de TSolicitud y construye la estructura de párrafos y subpárrafos en la tabla "Estructura" de la Base de Datos; cabe mencionar que la clase TSolicitud hereda de TObject, la cual esta en el tope del diagrama de clases de Delphi. En la figura 43 se pueden observar las clases que componen a TSolicitud, las cuales permiten las tareas antes mencionadas.
- ✓ TListaSolicitud, permite tener un listado de de alguno de los tipos de solicitud, es así como se observa que un encabezado tiene una lista de títulos, un titulo una lista de párrafos, un párrafo una lista de párrafos que representan los llamados subpárrafos y que pueden tener el nivel de indentacion que se desee, además los párrafos también tiene una lista de objetos multimedia; la descripción anterior corresponde a la estructura general utilizada dentro de los artículos de Gaia.

Figura 44. Diagrama de la clase TSolicitud, esta clase es una de las mas importantes dentro del software pues es la que permite la interacción y todas las operaciones con la base de datos.



A continuación se muestra el detalle de los miembros del diagrama de clases software de la figura 43.

Figura 45. Detalle miembros diagrama de clases software.

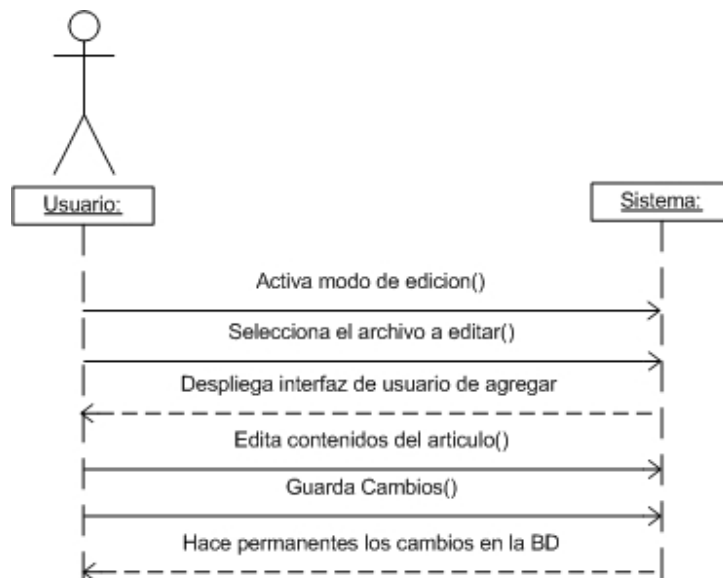


En la figura 45 se muestran las especificaciones de las clases software utilizadas, nótese que también se muestran otros métodos diferentes a agregar, es decir métodos que intervienen en el desarrollo de otros casos de uso.

➤ **Diagrama de Secuencia del Sistema para el caso de uso de editar artículo.**

El caso de uso de edición de artículos puede considerarse como una extensión del caso de uso de agregar artículos, de hecho se utiliza la misma interfaz de usuario, la diferencia es que en el modo edición se reconstruye el artículo dentro de la interfaz de usuario de agregar y se permite modificar la información del artículo.

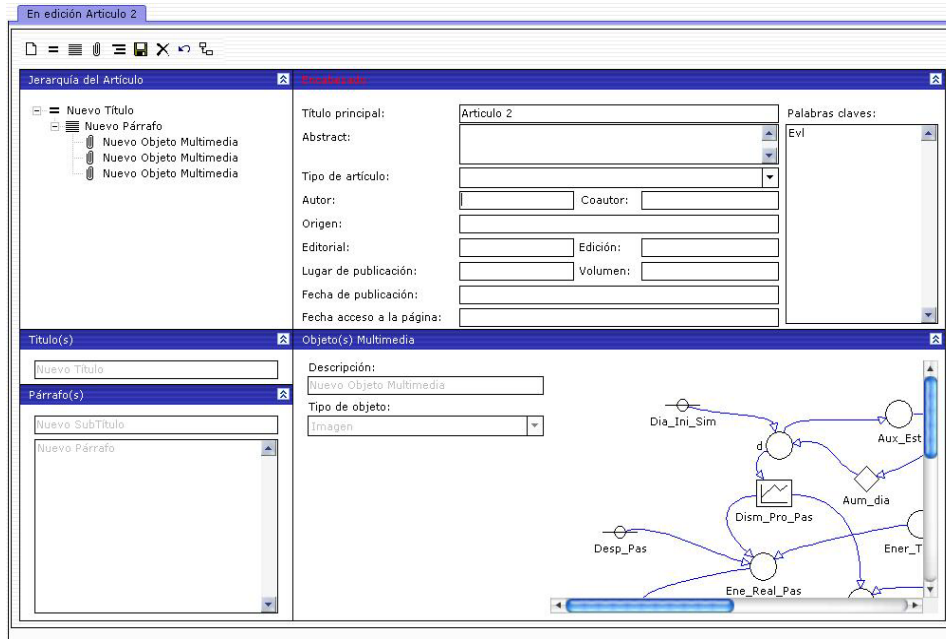
Figura 46. DSS para el caso de uso editar artículo.



➤ **Interfaz de usuario para editar artículo.** Como se menciona con anterioridad la interfaz de usuario para editar es la misma que para

agregar, la única diferencia es que abre un artículo que ya ha sido guardado, reconstruye su estructura y permite modificar sus contenidos.

Figura 47. Interfaz de usuario para editar artículo.



➤ **Miniproyectos dentro de la primera iteración de la fase de elaboración.**

Figura 48. Incrementos sucesivos en la herramienta software para cubrir el caso de uso agregar artículo.



➤ **Dificultades encontradas en la primera iteración de la fase de elaboración.** El principal problema encontrado a la hora de desarrollar esta primera iteración corresponde a la inclusión de los objetos

multimedia dentro de la base de datos; lo cual fue solucionado utilizando un comando de delphi el cual lee de un archivo de disco duro y lo carga a memoria y al campo de la base de datos; este comando corresponde a una función que encapsula el comportamiento general de objetos de gran tamaño que son guardados dentro de un campo de una tabla en una base de datos; a continuación se muestra la sintaxis de ejemplo:

```
TBlobField(FOrigenDatos.FieldByName('ObjetoMultimedia')).LoadFromFile('C:\Objeto'+FExtension);
```

Esta línea de código permite la inclusión de un archivo en disco duro dentro de la base de datos.

Otro inconveniente se presentó agregando y haciendo solicitudes sobre la base de datos, estas al estar abiertas copaban el número máximo de solicitudes disponibles por parte del manejador de BD de Access; lo cual obligó a cambiar la manera de guardar los datos, de forma tal que todas las solicitudes son hechas solo una vez por cada una; dicho mas claramente, antes todas las solicitudes estaban conectadas ahora cada solicitud se conecta, guarda y se desconecta, solo una a la vez, recorriendo un árbol que contiene a todas las solicitudes; este árbol se recorre fácilmente hasta el nivel de párrafos, para recorrer los párrafos que están identados, se utilizó un algoritmo recursivo que recorre esta relación entre párrafos padres y párrafos hijos, la estructura general de este árbol es la siguiente:

Figura 49. Jerarquía del árbol creado para poder guardar artículos.



Algo importante que se debe mencionar es que para la creación de cada uno de los elementos del artículo, el que lo agrega (el que agrega un elemento) es el padre, a excepción del encabezado que llama el mismo al método agregar sin necesidad que otro deba pasarle un mensaje para que este método se active; es así que para agregar un título, este se agrega a través de la lista de títulos que hace un llamado al método de agregar de Título; igualmente sucede con los párrafos y los objetos multimedia. Otra clase que se implementó para llevar un control del orden y la jerarquía del artículo es T Estructura que permite llevar dentro de la tabla Estructura este orden y decir qué párrafo es padre de cual párrafo y así permitir el nivel de indentación que el usuario desee dentro del artículo.

➤ **Interfaz de usuario de agregar artículo.**

Figura 50. Pantalla que sirve como interfaz de usuario para agregar un artículo.

The screenshot shows a web application window titled "Agregando artículo". The interface is organized into several panels:

- Jerarquía del Artículo:** An empty panel on the left side.
- Encabezado:** A central panel containing several input fields:
 - Título principal: [Text input]
 - Abstract: [Text input]
 - Tipo de artículo: [Dropdown menu, currently showing "Libro - Capitulo"]
 - Autor: [Text input] Coautor: [Text input]
 - Origen: [Text input]
 - Editorial: [Text input] Edición: [Text input]
 - Lugar de publicación: [Text input] Volumen: [Text input]
 - Fecha de publicación: [Text input]
 - Fecha acceso a la página: [Text input]
- Palabras claves:** A panel on the right with an empty list box.
- Titulo(s):** A panel below the header with a single text input field.
- Párrafo(s):** A panel below the titles with a text input field and a vertical scrollbar.
- Objeto(s) Multimedia:** A panel on the right with:
 - Descripción: [Text input]
 - Tipo de objeto: [Dropdown menu]

o **Segunda Iteración.** En esta segunda iteración se refina el caso de uso mostrar artículo, se realiza el diseño de sus clases software y se implementan convirtiéndose en código ejecutable que satisface los requisitos del caso de uso mostrar artículo.

➤ **Diagrama de Secuencia del sistema para el caso de uso mostrar artículo (DSS).**

Figura 51. DSS para el caso de uso mostrar artículo.



➤ **Diseño de objetos.** A partir de algunas clases del modelo del dominio y teniendo en cuenta la especificación del caso de uso mostrar artículo, se crean las clases software que permiten mostrar el artículo; al igual que en el caso de uso de agregar, aquí se utiliza a TArticulo el cual contiene un TFrmMostrarArticulo, permitiendo instancias tantas veces como se quiera a TFrmMostrarArticulo, lo cual permite tener múltiples entornos de mostrar al mismo tiempo.

Figura 52. Clases software que intervienen en el caso de uso de mostrar articulo y su relación con la interfaz de usuario y la Base de Datos.

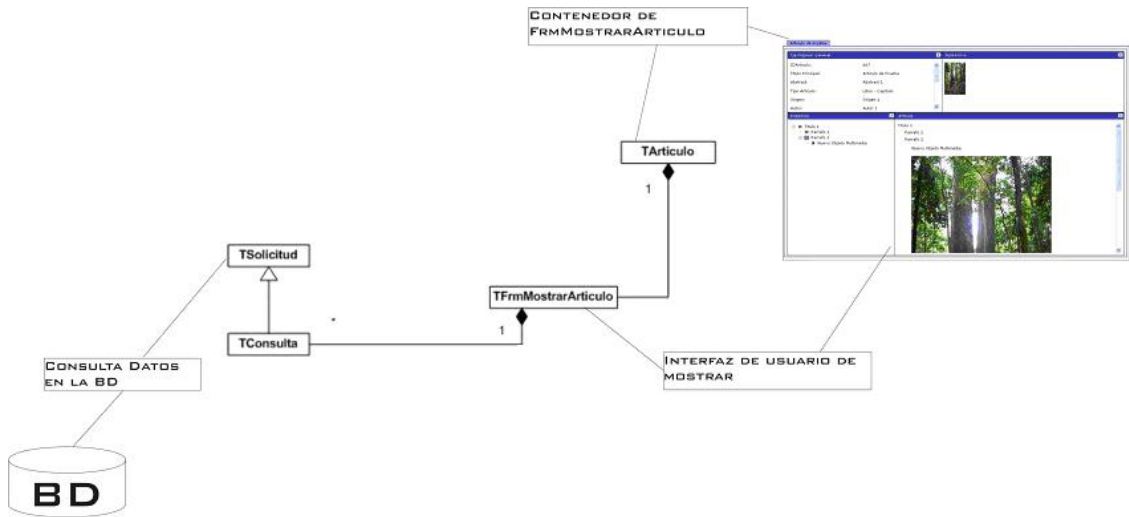
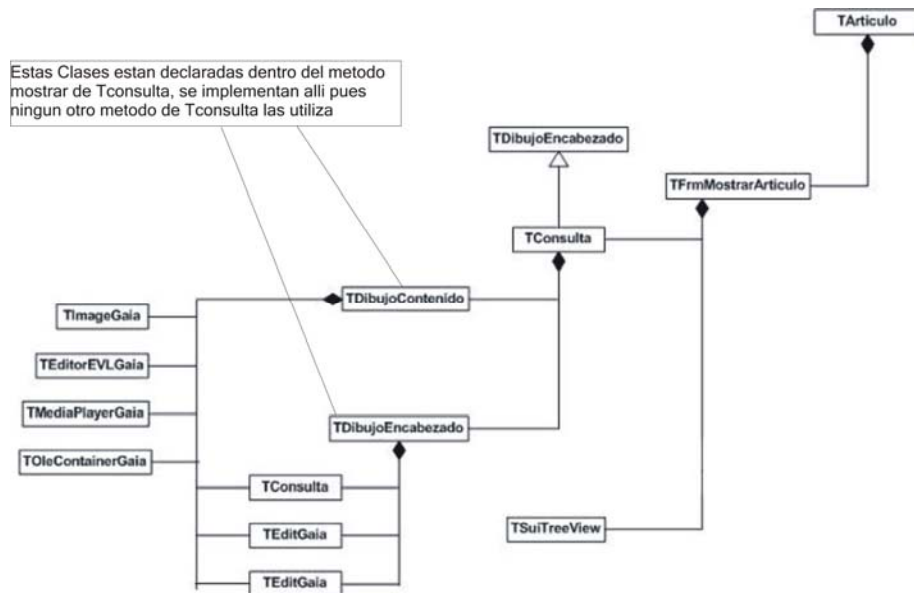
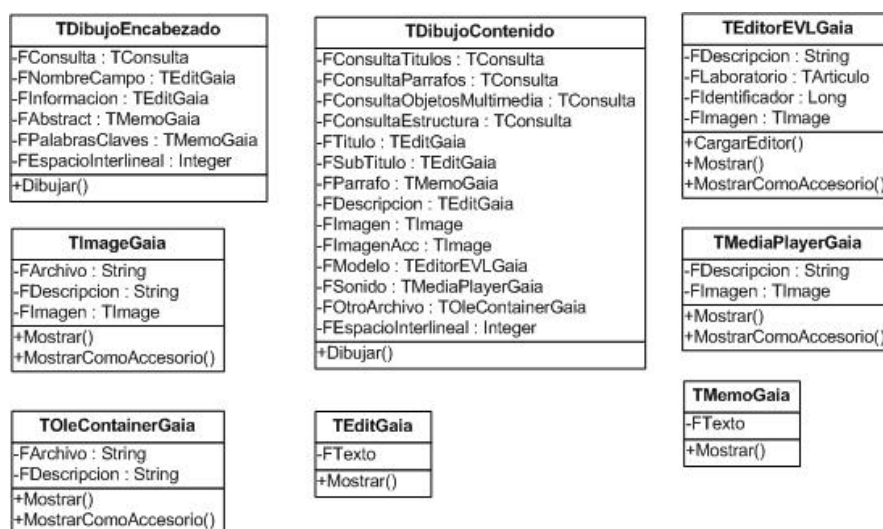


Figura 53. Especificación del diagrama de las clases que intervienen en el caso de uso mostrar.



Como se observa en el diagrama anterior, TConsulta contiene a TDibujoEncabezado y a TDibujoContenido las cuales son las encargadas de dibujar y mostrar el encabezado y el cuerpo del artículo, por otra parte TSuiTreeview es el mismo TTreeView de Delphi que permite colocarle un entorno visual; cuando este árbol jerárquico con la estructura del artículo es generado, a cada uno de sus elementos se le asocia un puntero que "apunta" al control que contiene la descripción del título, párrafo, subpárrafo u objeto multimedia, permitiendo de esta forma la navegación del artículo por parte del usuario.

Figura 54. Descripción de las clases del software.



➤ **Dificultades encontradas en la segunda iteración de la fase de elaboración.** La idea principal del software es permitir insertar dentro de un artículo cualquier tipo de información en cualquier formato y por supuesto poder mostrarla, en el caso de uso de agregar la parte de insertarlo fue resuelta, ahora el problema era el de mostrar estos contenidos con la jerarquía con la que fueron creados; en parte la solución estaba en extraer la información que ya está dentro de la BD con la jerarquía con la que la creó el usuario; pero además está el problema

de mostrar cada objeto multimedia dentro de un contenedor que pueda leer el tipo de archivo al cual pertenece, para ello en la tabla Objeto Multimedia se colocan los siguientes campos que sirven para que el software identifique el tipo de archivo y el contenedor que mas le conviene a ese tipo de archivo.

Figura 55. Tabla Objeto Multimedia.

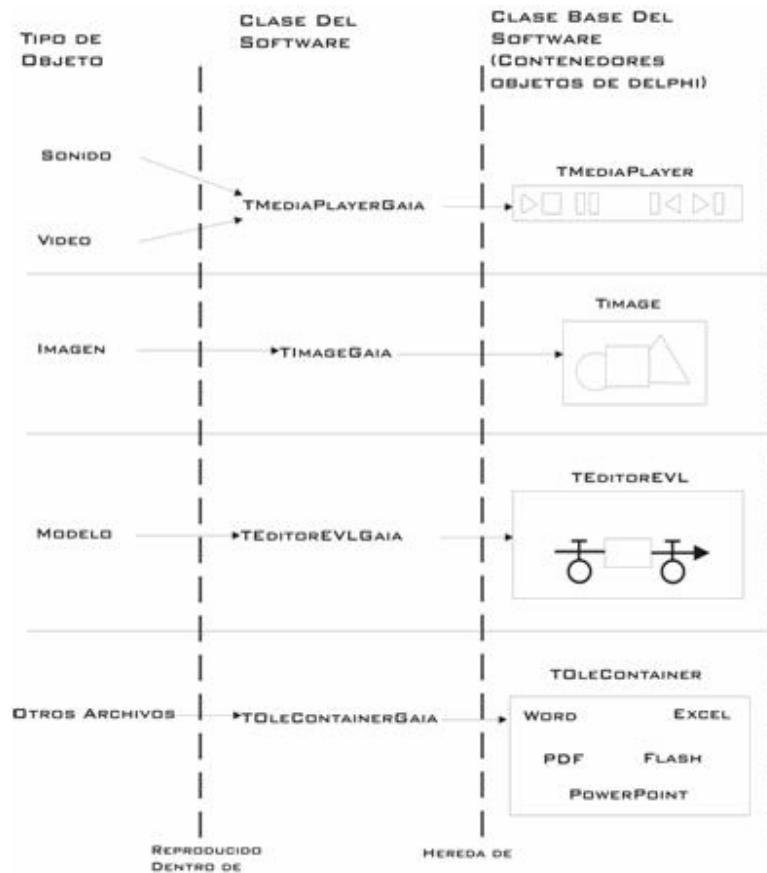
IDObjetoMultim	ObjetoMultimedia	Descripcion	Tipo	IDParrafo	Extension
541	Long binary data	Nuevo Objeto Multimedia	Sonido	1076	.mp3
542	Long binary data	Nuevo Objeto Multimedia	Sonido	1077	.mp3
544	Long binary data	Nuevo Objeto Multimedia	Imagen	1079	.jpg
*	(AutoNumber)				

Los posibles tipos de archivos que se definieron son:

- ✓ Imagen.
- ✓ Video.
- ✓ Sonido.
- ✓ Modelo.
- ✓ Otros Archivos.

Es así como el método mostrar de la clase TDibujoContenidos selecciona el contenedor apropiado para el objeto dependiendo de su tipo y lo coloca dentro del contenedor de todos los elementos del articulo.

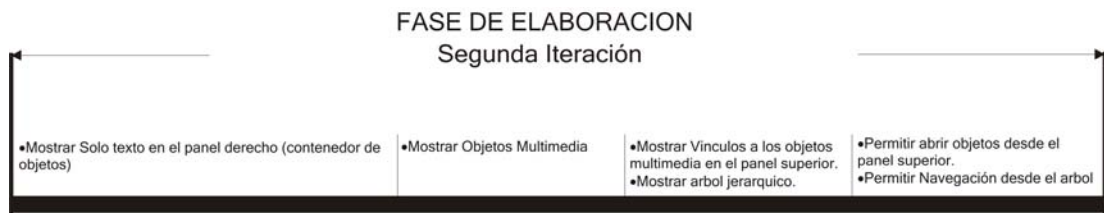
Figura 56. Relación entre los contenedores las clases del software y los tipos de archivo.



Igualmente existió el problema de generar la estructura real del artículo en el árbol jerárquico (TSuiTreeView), ya que existen párrafos que son subpárrafos de otros párrafos, es así como hay nodos padres y nodos hijos, de tal forma que para mostrar el artículo junto con su jerarquía correcta, se van imprimiendo los objetos en el panel contenedor de objetos (panel derecho) y a medida que se crea el objeto, se crea el elemento relacionado en el árbol de la izquierda, como es lógico, en el árbol quedan todos los párrafos en el mismo nivel de indentación, para recrear la jerarquía real se recurre a la tabla estructura que permite saber que párrafo es papa de cual párrafo y de esta forma con el método

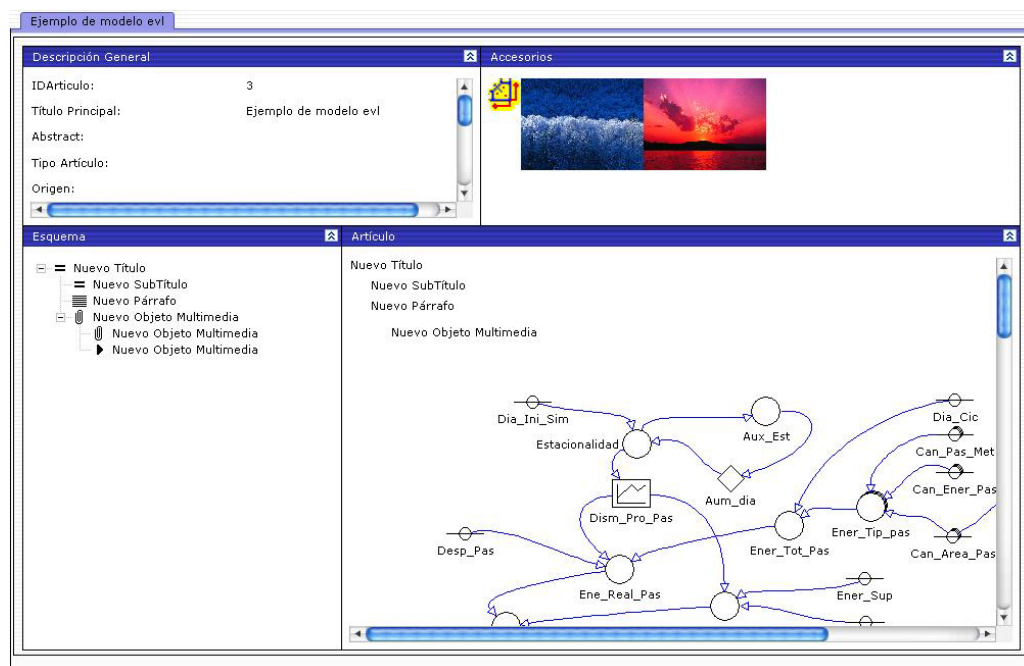
MoveTo(NodoDestino,Parentesco) de Delphi, se reordenan los nodos del árbol de acuerdo a la estructura real del artículo.

Figura 57. Mini – Iteraciones dentro de la segunda iteración de la fase de elaboración.



➤ **Interfaz de usuario para mostrar artículos.**

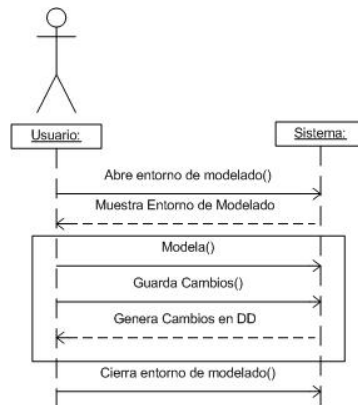
Figura 58. Interfaz de usuario para mostrar articulo.



o **Tercera Iteración.** En esta tercera iteración se formaliza el caso de uso de modelar y experimentar y se desarrollan las clases software que permiten al usuario llevar a cabo satisfactoriamente estos casos de uso.

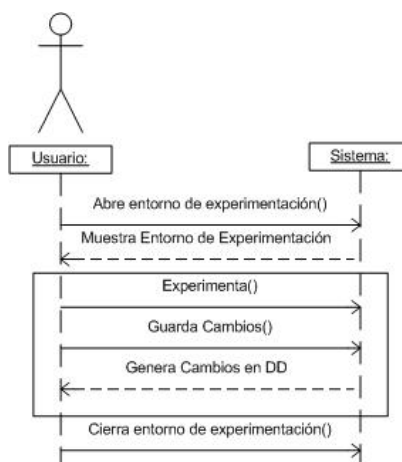
➤ **Diagrama de Secuencia del sistema para el caso de uso Modelar.**

Figura 59. DSS para el caso de uso modelar.



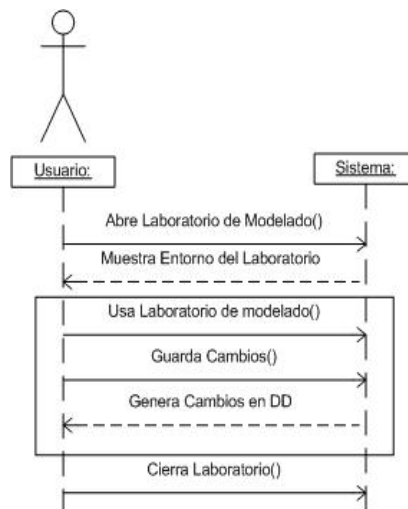
Mediante el diagrama de secuencia de estados anterior se formaliza el caso de uso correspondiente a modelar, en este caso modelamiento con dinámica de sistemas; a continuación se muestra el caso de uso correspondiente a experimentar.

Figura 60. DSS para el caso de uso experimentar.



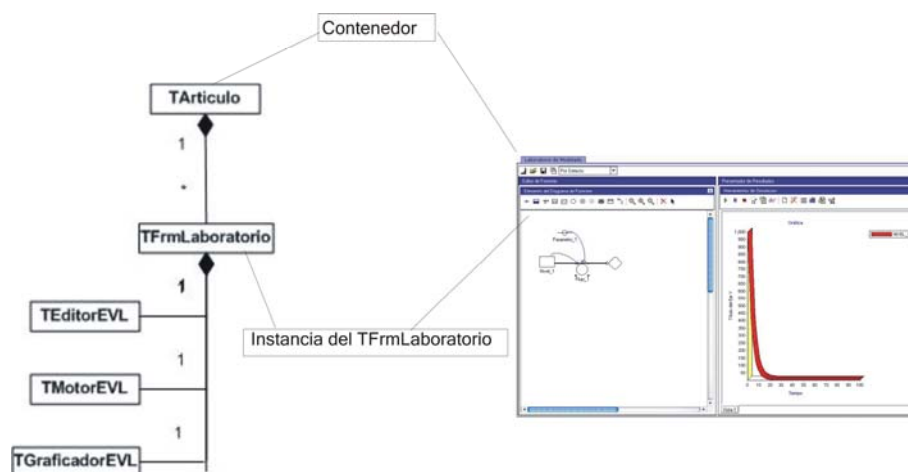
Tal como se puede observar en los DSS anteriores, ambos casos de uso son muy similares por un lado se habla de modelar y por otro de experimentar, pero en realidad son actividades afines tal como se ha visto con anterioridad a la hora de explicar el modelamiento con dinámica de sistemas como vía para la comprensión del fenómeno mediante la construcción de modelos, la experimentación con los mismos y la posterior modificación del modelo gracias a los resultados del proceso de experimentación. En los dos diagramas anteriores se observa que ambos inician abriendo el entorno de modelado o de experimentación, el cual realmente es el mismo y corresponde a lo que se ha denominado laboratorio de modelado el cual tiene las facilidades para modelar y para experimentar con el modelo, por otro lado se observa que el usuario experimenta o modela, lo cual se puede condensar en manipular diagramas de forrester y sus elementos. De tal forma es posible condensar esos dos casos de uso en uno solo denominado Uso del laboratorio de modelado.

Figura 61. DSS para el caso de uso "Usar Laboratorio".



➤ **Diseño de objetos.** Mas que diseño de objetos, en este caso se debe hablar de reutilización de software puesto que se utilizan componentes y clases desarrolladas para el software evolución 3.5, estas clases son el pilar fundamental del laboratorio y permiten implementar gran parte de las facilidades ofrecidas por el software Evolución 3.5, también se crea una clase necesaria para poder colocar la plantilla del laboratorio de modelado dentro de un contenedor de hojas (TArticulo, el cual como se ha mostrado con anterioridad, hereda de TSuiTabSheet); el diagrama de clases queda como sigue:

Figura 62. Diagrama de clases del software y su relación con la interfaz de usuario.



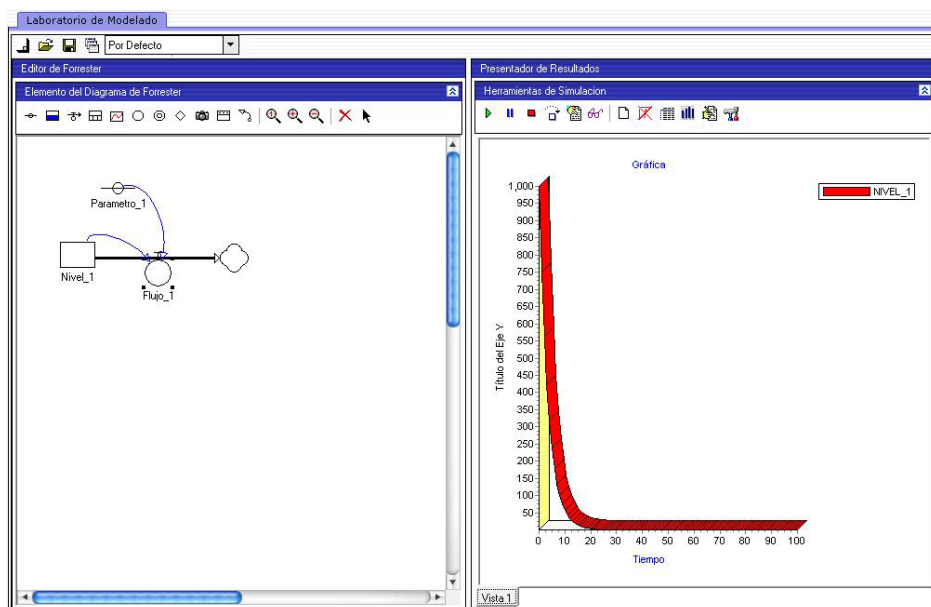
➤ **Dificultades encontradas en la tercera iteración de la fase de elaboración.** En esta iteración, la dificultad radico precisamente en la correcta utilización de software desarrollado por otros programadores, para lo cual se recurrió al manual del programador de Evolución 3.5 donde explican todos y cada uno de los métodos y propiedades de las clases que permiten la utilización de la mayoría de las facilidades del software Evolución.

Figura 63. Pequeñas iteraciones incluidas dentro de la tercera iteración de la fase de elaboración.



➤ **Interfaz de usuario del laboratorio de modelado.**

Figura 64. Pantalla de la interfaz del laboratorio de modelado.



○ **Cuarta Iteración.** Mediante las iteraciones anteriores se obtiene un producto software que permite agregar artículos, mostrarlos y operar con un laboratorio de dinámica de sistemas; es necesario que estos contenidos creados se puedan localizar fácilmente, en esta iteración se realizan los pasos necesarios para poder buscar artículos. Antes de la implementación de la búsqueda, en el panel izquierdo se cargaban todos

los artículos y no se podía realizar búsquedas sobre esa lista. En esta cuarta iteración también se implementa la inclusión de palabras claves dentro del artículo, lo cual le añade mas formas de búsqueda al articulo.

Figura 65. Diagrama de Secuencia de estados para la búsqueda.

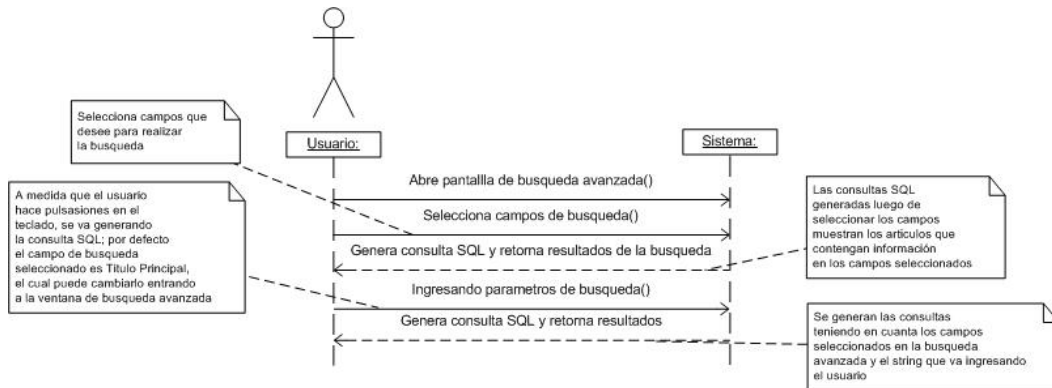
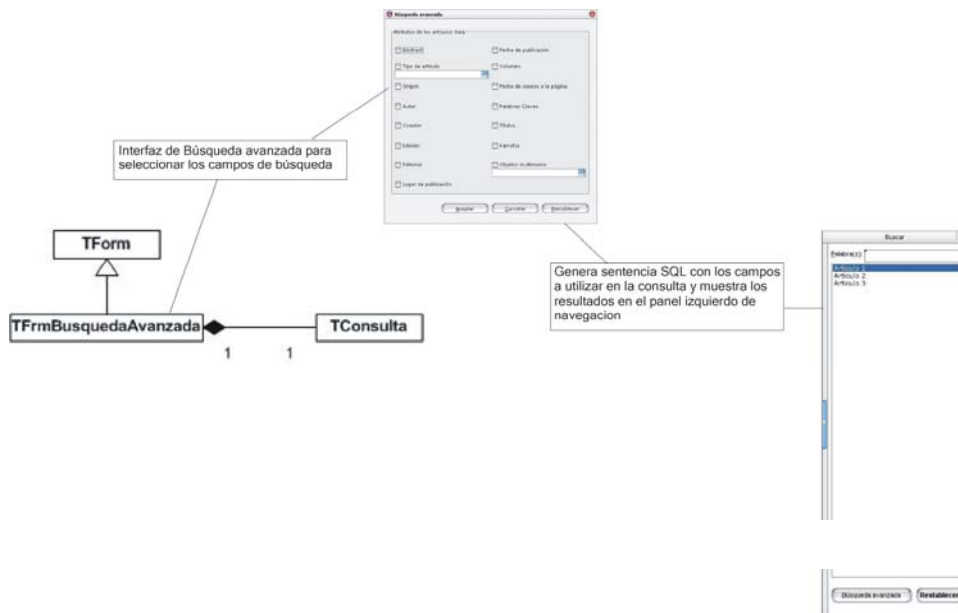


Figura 66. Clases que intervienen en la búsqueda y su relación con la interfaz de usuario.



➤ **Interfaz de usuario para la búsqueda.**

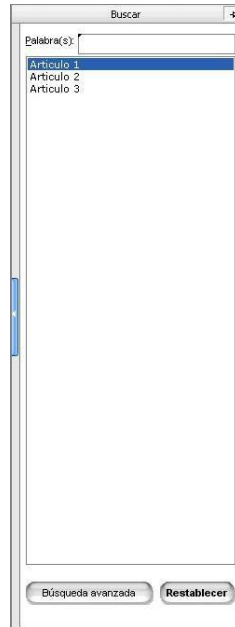
La búsqueda avanzada permite seleccionar el campo(s) por medio de los cuales se va a realizar la búsqueda, en la figura 67 se muestra la pantalla de selección de campos para la búsqueda.

Figura 67. Pantalla de la interfaz de usuario para búsqueda avanzada.



Por defecto la búsqueda en el panel izquierdo de navegación es hecha por el campo Titulo principal, lo cual puede ser cambiado en la pantalla de búsqueda avanzada; en la siguiente figura se muestra el panel izquierdo del entorno principal de GAIA que sirve para realizar las búsquedas dentro de la Base de Datos.

Figura 68. Panel izquierdo del entorno de navegación principal utilizado para ingresar las palabras.



➤ **Inclusión de palabras claves en el artículo.** Como se menciono con anterioridad, en esta iteración se agrega la funcionalidad de las palabras claves, por lo que el caso de uso de agregar articulo se le añade la funcionalidad de ingresar las palabras claves; la interfaz de usuario es la misma pues ya se había creado el espacio para el ingreso de las palabras claves, lo que se hizo en esta mini – iteración fue enlazar esta caja de texto donde se ingresan las palabras claves con la base de datos para así poder guardarlas, esta inclusión de palabras claves permite seleccionar en búsqueda avanzada el campo palabras claves para de esta forma buscar artículos utilizando este recurso.

➤ **Pequeñas Iteraciones dentro de la cuarta Iteración.** Al igual que las iteraciones anteriores, esta se subdivide en unas más pequeñas o mini-iteraciones.

Figura 69. Mini – Iteraciones en la cuarta Iteración de la fase de elaboración.



o **Quinta Iteración.** En las iteraciones anteriores, se formalizaron los casos de uso mas importantes y además se realizaron los diagramas de clases de software que sirvieron para implementarlos en código; además se hicieron varias pruebas cuyo propósito es el de comprobar el resultado de la implementación²⁶. En esta iteración que es considerada transitiva pues a partir de aquí termina la fase de elaboración para empezar con la de construcción²⁷, se aborda en esta iteración el caso de uso agregar usuario, lo cual va a moldear a los casos de uso de agregar y mostrar, pues al existir usuarios los contenidos deben quedar filtrados por usuarios, permitiendo independencia de contenidos entre los usuarios del software.

Es importante aclarar que en general las iteraciones en la fase siguiente (de construcción) son muy similares a las que se vieron en la fase de elaboración con la diferencia que tienen un poco menos de especificación de requisitos (descripción de casos de uso) y algo mas de pruebas de implementación.

➤ **Especificación del caso de uso agregar usuario.** Este caso de uso permite la existencia de múltiples usuarios dentro del software lo cual

²⁶ Estas pruebas que en su totalidad fueron del tipo "caja negra" sirvieron como apoyo a la depuración de errores en cada uno de los casos de uso implementados.

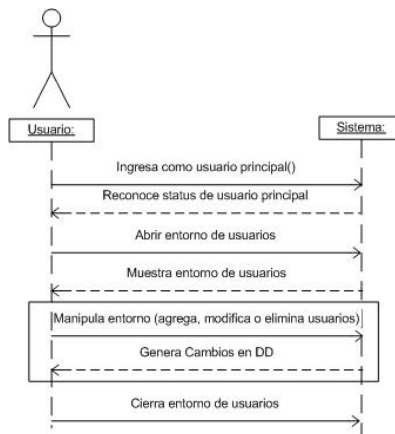
²⁷ Se considera como iteración transitiva pues ya los casos de uso mas importantes han sido formalizados e implementados y en esta iteración se realizan mas pruebas al producto software que va construido y además se termina de formalizar la mayoría de casos de uso y se implementa algo de software que es pilar para el resto del desarrollo de la herramienta (inclusión de usuarios).

deja la posibilidad de tener múltiples guías de estudio, dependiendo del usuario que este utilizando en un momento dado la herramienta. La descripción del caso de uso agregar es sencilla y se puede expresar mediante los siguientes pasos:

1. Ingresar al sistema con el usuario principal (Hace las veces de administrador).
2. Entrar al entorno de agregar usuarios.
3. Ingresar datos del nuevo usuario.
4. Aceptar los datos suministrados.

➤ **Diagrama de secuencia del sistema para el caso de uso agregar usuario.**

Figura 70. DSS para el caso de uso de agregar usuarios, observe que en realidad se amplió el caso de uso a manipulación de usuarios pues cobija también editar y eliminar usuarios.



➤ **Diseño de objetos.** Como se observa en la especificación del caso de uso, este se puede subdividir en dos partes, una correspondiente al inicio de sesión que permite el ingreso del usuario y le da a conocer al sistema el usuario que esta trabajando en el software. De tal forma se tienen los

siguientes diagramas de clases tanto para el inicio de sesión como para el entorno de manejo de usuarios:

Figura 71. Relación de las clases del software con la Base de Datos y la interfaz del usuario para iniciar sesión.

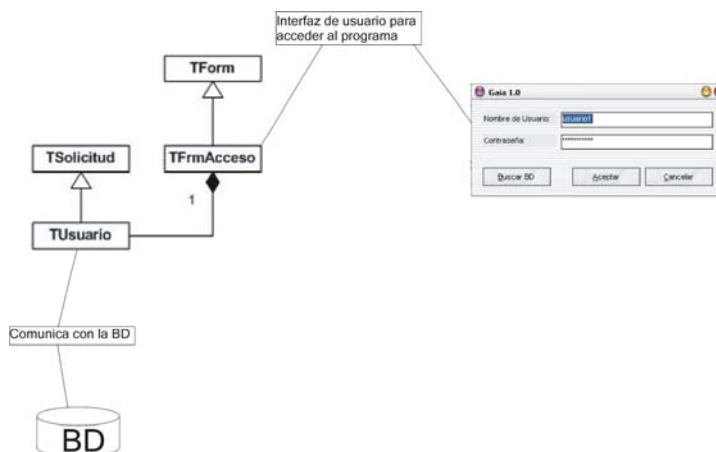
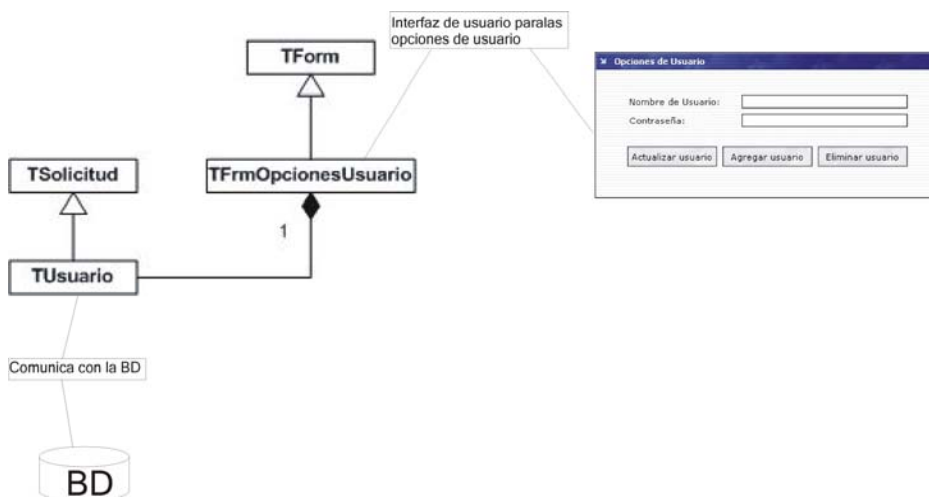
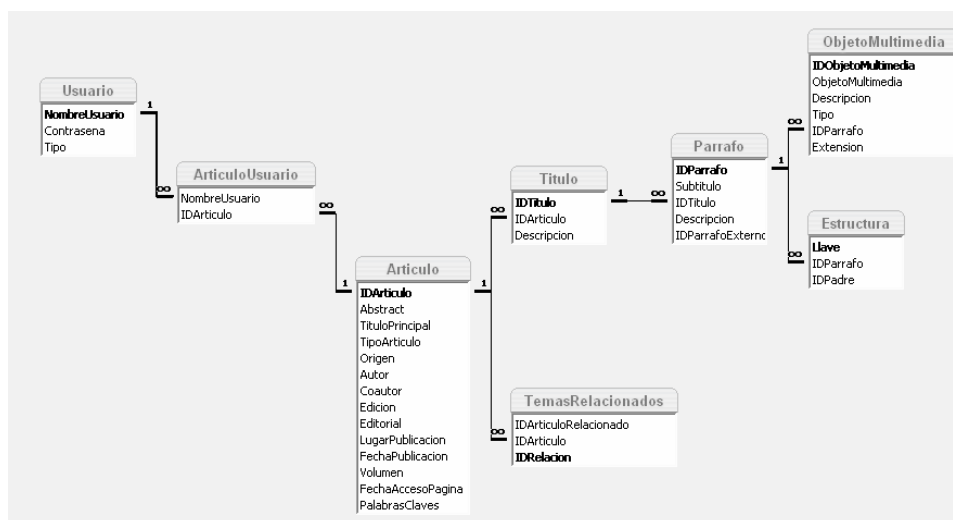


Figura 72. Relación de las clases del software con la Base de Datos y la interfaz del usuario para cambiar las opciones de usuario.



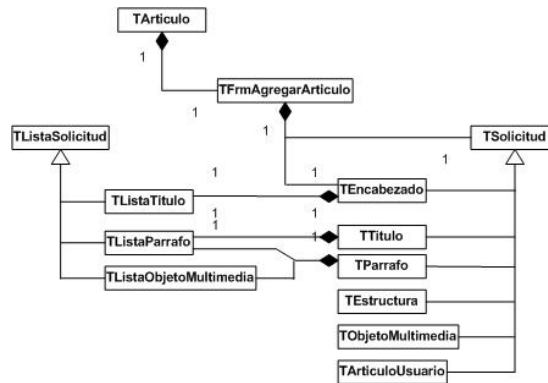
➤ **Aumento de funcionalidad a los casos de uso principales (Agregar, Mostrar, Laboratorio).** Gracias a la inclusión de usuarios en el software, es posible ahora poder relacionar a cada artículo con el usuario que lo creo, permitiendo la antes mencionada independencia de contenidos para cada uno de los usuarios de la herramienta. Para esto, se utilizan las tablas Usuario y ArtículoUsuario.

Figura 73. Diagrama entidad relación con las tablas Usuario y ArtículoUsuario.



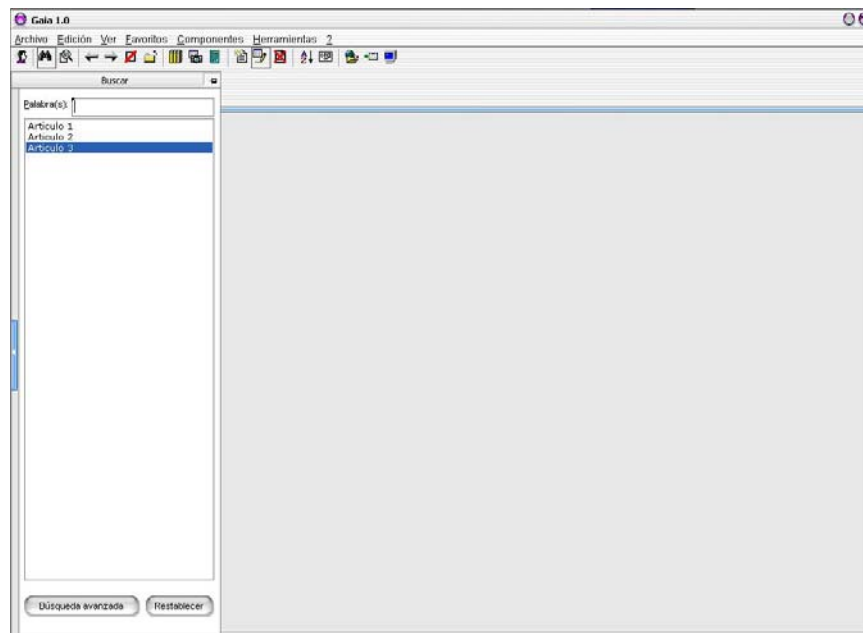
➤ **Mejoras a la funcionalidad del caso de uso agregar artículo.** Con la inclusión de usuarios, ahora el sistema al agregar un artículo, relaciona automáticamente el artículo con el usuario que este utilizando el software, esta dupla *Usuario, Artículo* queda registrada dentro de la tabla ArtículoUsuario.

Figura 74. Diagrama de clases-software para el caso de uso de agregar artículo, observe que se agrega la clase TArticuloUsuario que permite relacionar el artículo con el usuario activo.



➤ **Filtrado de los artículos en el panel izquierdo por usuario.**

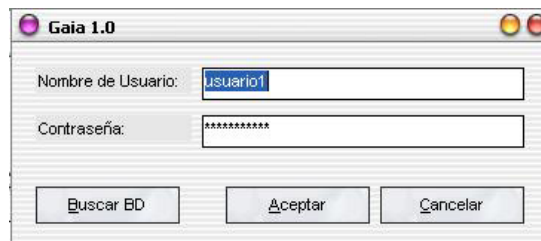
Figura 75. Pantalla de la interfaz principal de GAIA, en el panel izquierdo ahora aparecen los artículos que han sido creados por el usuario actual.



Ahora con la existencia de usuarios, los artículos son filtrados dependiendo del usuario que este activo de modo tal que solo aparezcan los que han sido creados por el usuario actual.

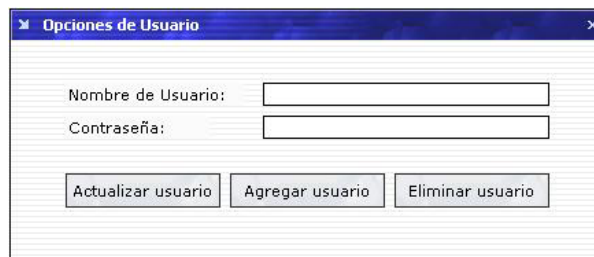
➤ **Interfaz de usuario utilizada para la manipulación de los usuarios.**

Figura 76. Pantalla de la interfaz de usuario utilizada para iniciar sesión.



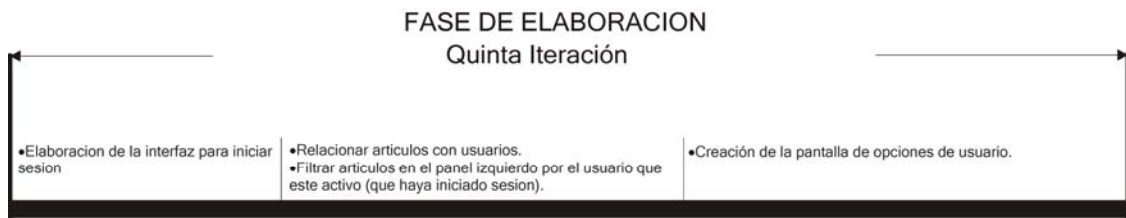
El usuario por defecto permite crear mas cuentas, tiene los privilegios de administrador. Una vez dentro del software, es posible acceder a las opciones de usuario que le permiten actualizar, agregar o eliminar un usuario

Figura 77. Pantalla de opciones de usuario.



➤ **Pequeñas Iteraciones dentro de la quinta iteración de la fase de elaboración.**

Figura 78. Miniproyectos dentro de la quinta iteración.



• **Fase de Construcción.** En la fase de elaboración se formalizaron los casos de uso más importantes y se desarrollaron, obteniendo de esta forma un producto software que cubre la funcionalidad básica de la mayoría de estos casos de uso. Para el momento de la fase de Construcción se tenían los siguientes artefactos o productos:

1. Modelo del dominio completo.
2. Formalización y definición más a fondo de los casos de uso de agregar, mostrar y buscar artículos, experimentar, modelar, agregar usuarios.
3. Modelo de datos que satisface las necesidades de los casos de uso antes mencionados.
4. Desarrollar los casos de uso anteriores con clase de software y su implementación en código.

En la fase de construcción se implementan los casos de uso de usar glosario, guardar artículo y se extienden y refinan las implementaciones de los casos de uso desarrollados durante la fase de elaboración, al igual que en la fase anterior, se realizan pruebas en cada una de las iteraciones

para comprobar el cubrimiento de la implementación con el caso de uso correspondiente, además se realizan al final de la fase un conjunto de pruebas que sirven para corregir posibles errores, estas pruebas son realizadas con un conjunto de personas que utilizan toda la funcionalidad del software y desarrollan "casos de prueba" para comprobar el correcto cubrimiento de todos los casos de uso.

○ **Primera Iteración.** En esta primera iteración se define el caso de uso de usar glosario; este caso de uso no es considerado como un caso de uso primario²⁸ tal como lo son los anteriores casos de uso mencionados en la fase de elaboración, a continuación se muestra un diagrama de secuencia del sistema que "formaliza" el caso de uso de usar glosario.

➤ **Diagrama de Secuencia del Sistema para el caso de uso "usar glosario".**

El caso de uso "usar glosario" esta dividido en cinco casos de uso que lo especifican mas explícitamente, estos casos de uso son:

1. Agregar Concepto.
2. Relacionar Concepto.
3. Editar Concepto.
4. Eliminar Concepto.
5. Buscar Concepto.

A continuación se presentan los Diagramas de Secuencia del Sistema para cada uno de estos casos de uso particulares.

²⁸ NOTA DE LOS AUTORES: Se hace referencia a caso de uso primario cuando se trata de casos de uso que ayudan a cubrir las necesidades o requisitos fundamentales por los cuales es creado el software.

Figura 79. DSS del caso de uso de agregar concepto.

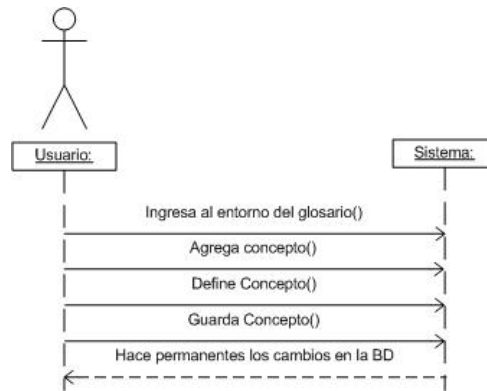


Figura 80. DSS del caso de uso relacionar concepto.

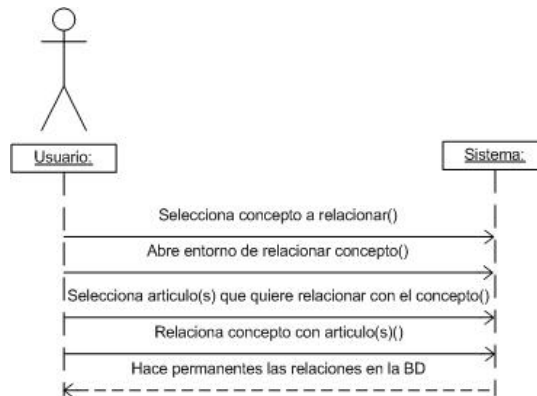


Figura 81. DSS del caso de uso eliminar concepto.

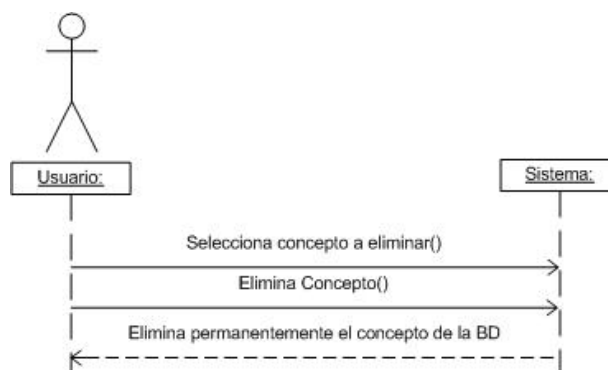


Figura 82. DSS del caso de uso Buscar concepto.

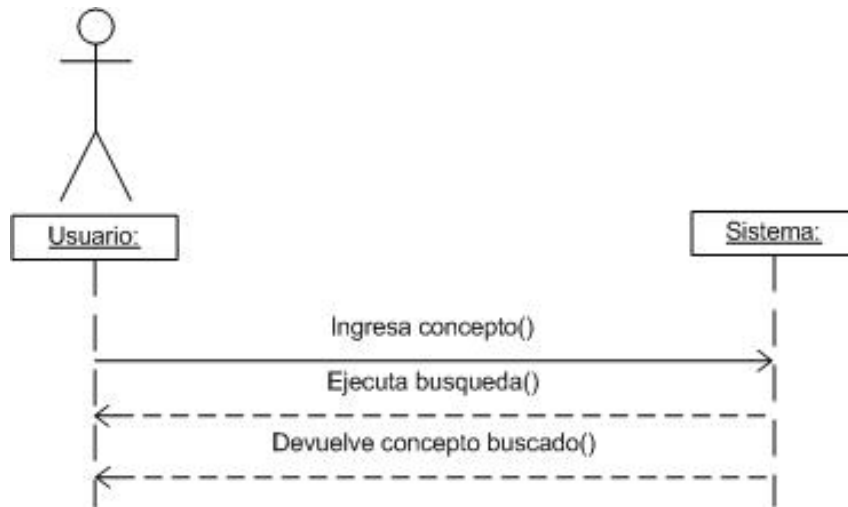
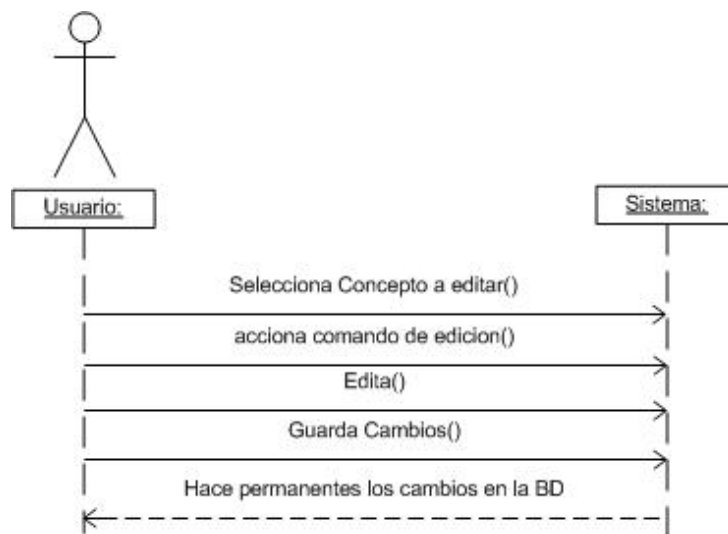
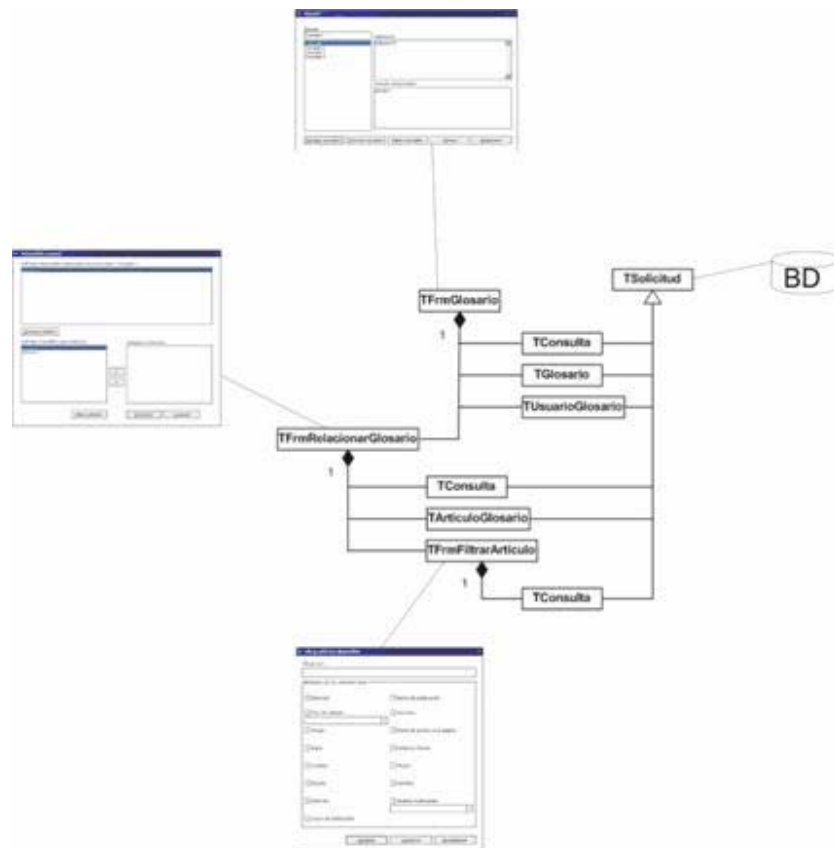


Figura 83. DSS del caso de uso Editar concepto.



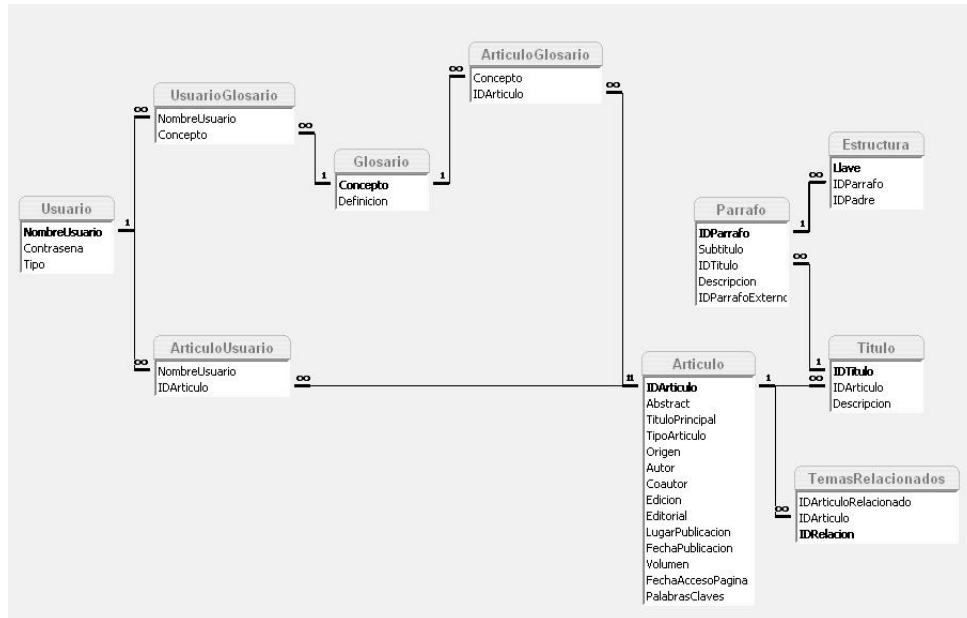
➤ **Diseño de las clases para el caso de uso "usar glosario"**. Como se menciono con anterioridad, este caso de uso esta compuesto por casos de uso mas pequeños, en la figura 74 puede observar el diagrama de clases de software resultante para cubrir todos estos casos de uso.

Figura 84. Diagrama de clases y su relación con la interfaz de usuario y la base de datos.



➤ **Modificaciones al modelo de datos.** Para poder implementar las facilidades que ofrece el glosario, es necesario añadir nuevas tablas y relaciones a la base de datos.

Figura 85. Modelo de datos final, observe que se añaden nuevas tablas.



➤ **Interfaz de usuario para la manipulación del Glosario.** Hay tres pantallas que permiten la manipulación del glosario y el cubrimiento de los casos de uso especificados en esta sección, la Figura 86 muestra la pantalla principal del glosario, donde se puede buscar, agregar, editar, borrar o abrir el entorno para relacionar un concepto con algún artículo que se encuentre dentro de la base de datos (ver Figura 87); en la figura 88 se muestra la pantalla utilizada para filtrar artículos, es la misma pantalla utilizada en el caso de uso de búsqueda avanzada, esta permite buscar de diferentes maneras el artículo(s) que se desea(n) relacionar con algún concepto específico.

Figura 86. Entorno principal del Glosario.

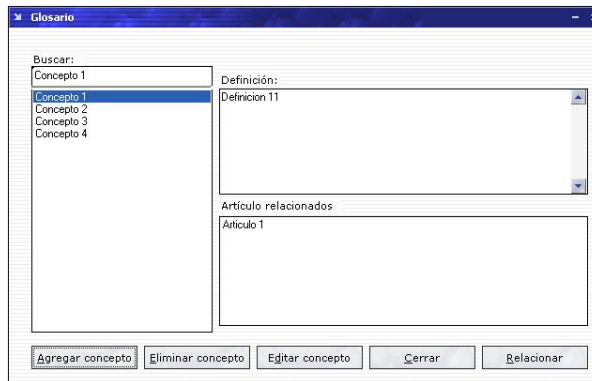


Figura 87. Interfaz de usuario para relacionar conceptos con artículos.

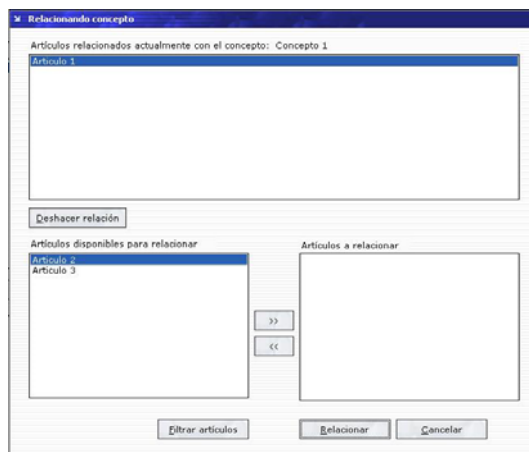
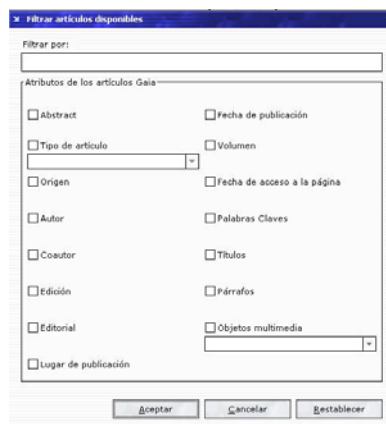


Figura 88. Pantalla utilizada para filtrar artículos a relacionar.



➤ **Mini – proyectos dentro de la primera iteración de la fase de construcción.**

Figura 89. Iteraciones intermedias dentro de la primera iteración de la fase de construcción.



○ **Segunda Iteración.** Para hacer una herramienta portable y que permita el intercambio de información entre diferentes usuarios que esten distantes físicamente, es necesario implementar un caso de uso que permita guardar los artículos en disco duro y cargarlos desde el disco a la BD de la herramienta. En esta segunda iteración se describe este caso de uso y se realiza su implementación.

➤ **Diagrama de Secuencia del Sistema para el caso de uso guardar a Disco Duro.**

Figura 90. DSS caso de uso guardar artículo a Disco Duro.

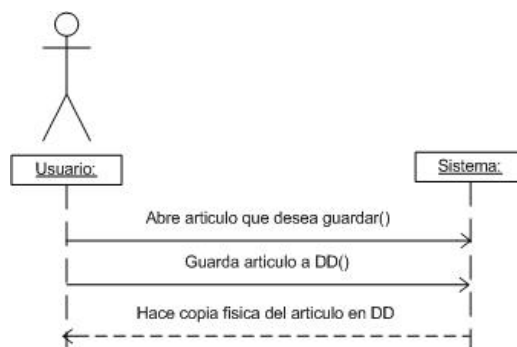
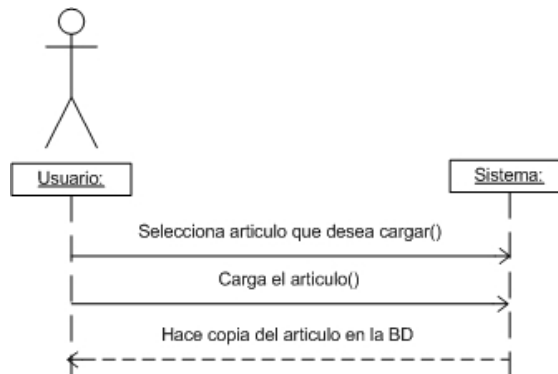
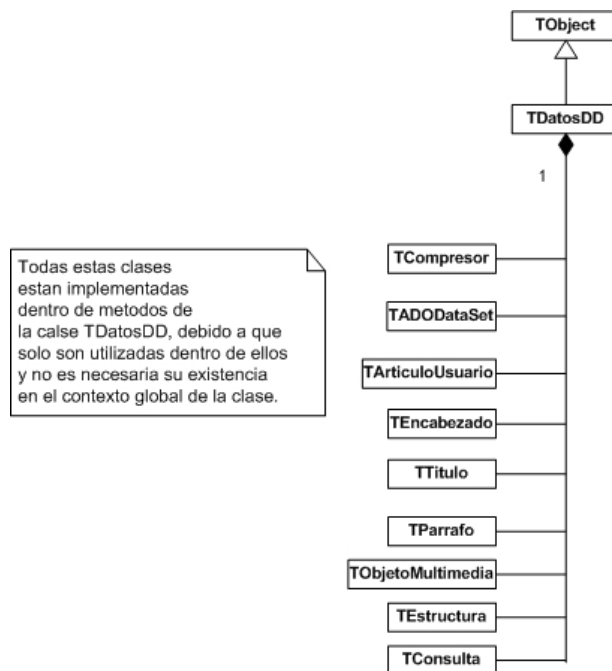


Figura 91. DSS caso de uso cargar artículo a la Base de Datos.



➤ **Diseño de las clases para el caso de uso guardar a Disco Duro y cargar a base de datos.**

Figura 92. TDatosDD es la clase encargada de guardar datos a disco duro y de cargar los artículos desde disco al programa.



En la siguiente figura se especifica con más detalle la clase TDatosDD, con sus métodos y propiedades.

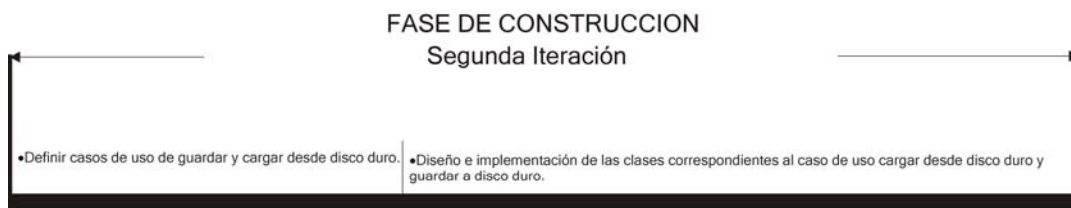
Figura 93. Especificación de TDatosDD.

TDatosDD
-FIdentificador : Long
-FNombreArchivo : String
-FNombreArchivoCargar : String
-FDialogSalvar : TSaveDialog
-FDialogCargar : TOpenDialog
-CrearArchivosXML()
-GenerarArchivoGaia()
-ExtraerArchivosXML()
-SubirArchivosXMLBaseDatos()
+Guardar()
+Cargar()

Los métodos de la clase TDatosDD son los que contienen las clases del diagrama de la figura 93; observe que para implementar copia de archivos desde y hacia disco duro se utilizan archivos XML los cuales son generados por el metodo CrearArchivosXML(), y cargados a la Base de Datos por medio de los métodos ExtraerArchivosXML() y SubirArchivosXMLBaseDatos().

➤ **Iteraciones intermedias.**

Figura 94. Mini – proyectos dentro de la segunda iteración. De la fase de construcción.



○ **Tercera Iteración.** Al momento de iniciar esta tercera iteración, todos los casos de uso críticos y su funcionalidad básica estaban completamente

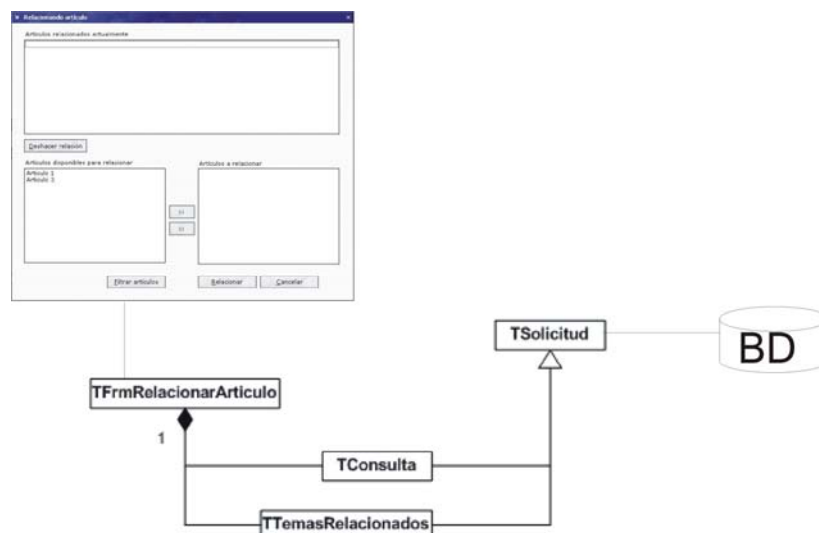
definidos e implementados; en esta iteración se refinan los casos de uso de agregar y mostrar artículo; se implementan los temas relacionados.

➤ **Agregar Temas Relacionados.** Los temas relacionados permiten al usuario crear vínculos entre diferentes artículos que tengan temas afines. Para agregar un tema relacionado el usuario debe:

1. Estar en el entorno de agregar artículo (debe estar agregando el artículo que quiere relacionar con otro(s) artículo(s)).
2. Abrir el entorno de agregar artículo.
3. Seleccionar el artículo(s) a relacionar.
4. Establecer relación.

➤ **Diagrama de clases e interfaz de usuario para crear temas relacionados.**

Figura 95. Diagrama de clases e interfaz de usuario para agregar temas relacionados.

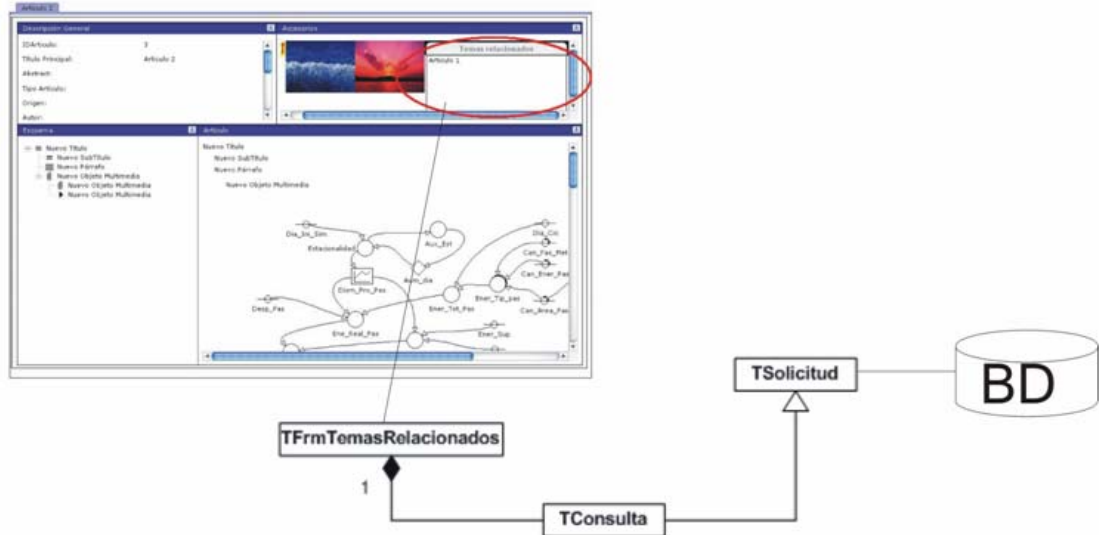


➤ **Mostrar temas relacionados.** Una vez hechas los vínculos entre temas relacionados el usuario debe poder ver los temas relacionados de cada artículo para ello se muestran los vínculos a los artículos relacionados dentro de el contenedor de objetos multimedia en el entorno de mostrar artículo. Los pasos para acceder a los temas relacionados de un artículo son:

1. Debe existir un artículo abierto que contenga temas relacionados.
2. En el panel superior derecho (contenedor de objetos multimedia) selecciona el artículo relacionado que quiera acceder.

➤ **Diagrama de clases relación con la interfaz de usuario para mostrar temas relacionados y la base de datos.**

Figura 96. Diagrama de clases e interfaz para mostrar temas relacionados.



El registro de todos los temas relacionados se lleva en la tabla TemasRelacionados de la base de datos.

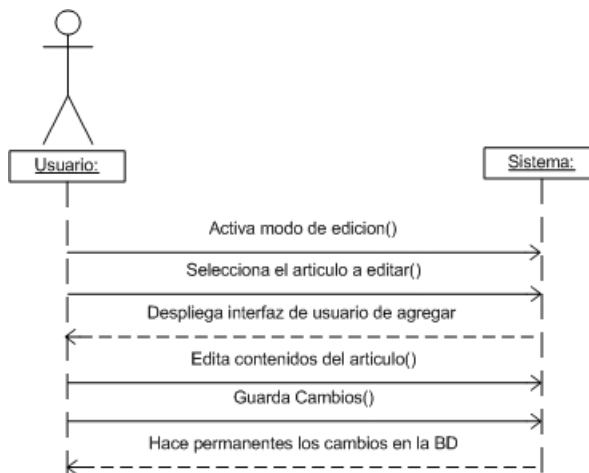
➤ **Iteraciones intermedias.**

Figura 97. Miniproyectos dentro de la tercera iteración de la fase de construcción.



○ **Cuarta Iteración.** En esta iteración se refina el caso de uso buscar, se extiende ahora la búsqueda y se realizan búsquedas de palabras dentro del artículo que tenga seleccionado el usuario.

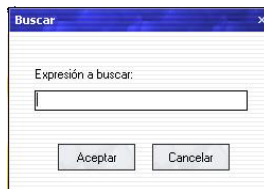
Figura 98. Diagrama de Secuencia del Sistema para el caso de uso de buscar en artículo.



Para la implementación de esta búsqueda se utiliza el componente de delphi TFindDialog, el cual busca dentro de los RichTextBox el texto que esta siendo buscado.

➤ **Interfaz de búsqueda en artículo.**

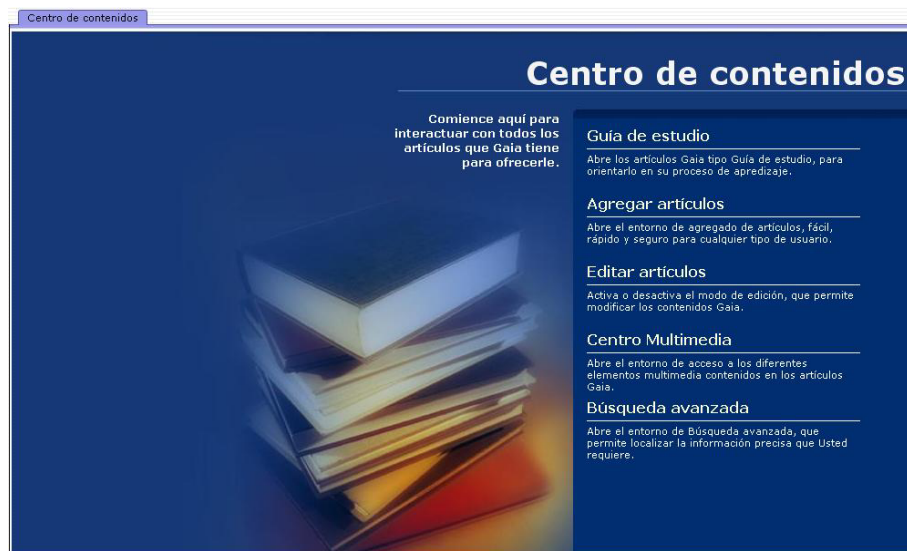
Figura 99. Pantalla utilizada para ingresar expresión a ser buscada en el artículo.



○ **Quinta Iteración.** En esta quinta iteración se implementan mas facilidades de navegación para el usuario; se crean los llamados centros de navegación que sirven como punto de inicio para la exploración y navegación de los contenidos de la herramienta. Cada centro sirve para agrupar las utilidades del software por tipos, es asi como se tienen tres grandes centro de navegación, estos son:

➤ **Centro de Contenidos.**

Figura 100. Pantalla del centro de contenidos.



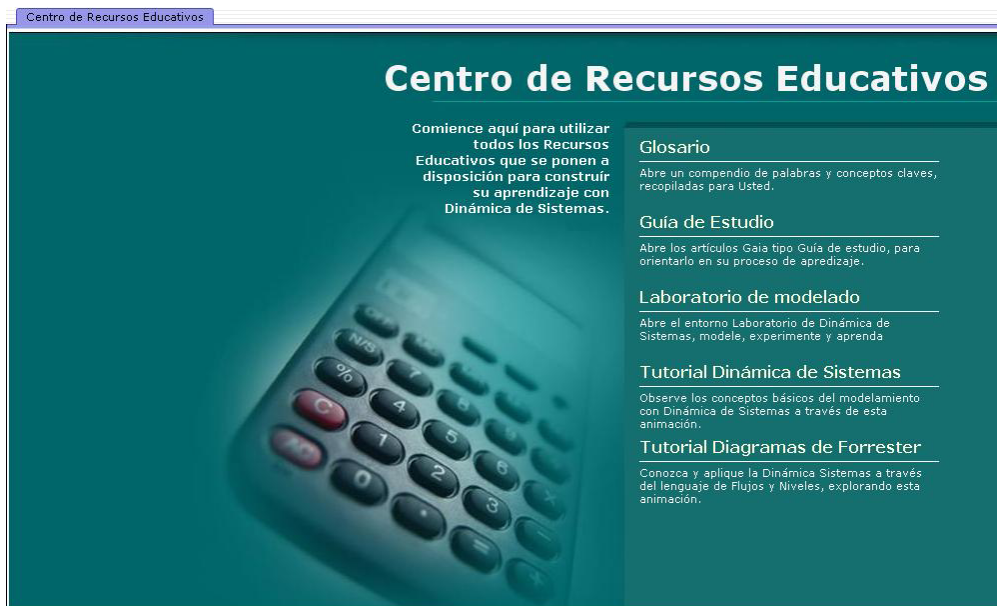
➤ **Centro Multimedia.**

Figura 101. Vista del centro multimedia.



➤ **Centro de recursos educativos.**

Figura 102. Vista del centro Multimedia.



• **Fase de Transición.** A lo largo del desarrollo del proyecto no fue necesaria la creación de casos de prueba formales, debido a que los casos de uso son muy puntuales y estrictos, es decir que no afectaban muchas cosas en el sistema al mismo tiempo, por lo cual los resultados de las acciones del usuario sobre el sistema son bastante predecibles y no implican un alto riesgo para la estabilidad total del sistema, es así como cada vez que se obtenía un un producto software, la funcionalidad de este era probada para seguir así realizando incrementos sucesivos hasta llegar al producto final. En esta fase de transición se realizan pruebas un poco más exhaustivas con un grupo de usuarios, primero con la ayuda de tres ingenieros de sistemas que realizaron pruebas exhaustivas a todos los casos de uso, es decir cada uno le realizo pruebas totales al sistema; los resultados de las pruebas son los siguientes.

➤ **Personal encargado de realizar pruebas exhaustivas al sistema:**

Tabla 7. Personal encargado de las pruebas exhaustivas.

Nombre:	Probar:
Oscar Iván Morales Duarte (Ing. De Sistemas UIS).	Agregar Articulo Editar Articulo
Andrés Ernesto Jiménez Posada (Ing. De Sistemas UIS).	Mostrar Articulo Búsqueda de Artículos Búsqueda dentro del articulo
Cristina Durango Mendoza (Estudiante de Ingeniería de Sistemas 10° Nivel).	Laboratorio de modelado Centros de Navegación Guardar y Cargar desde Disco Duro
Jorge Alberto Ortiz Martínez (Estudiante 5° Nivel de Ingeniería Electrónica).	Glosario Manejo de Usuarios Temas Relacionados

➤ **Resultados de las prueba de agregar articulo:**

Tabla 8. Pruebas a agregar artículo.

Criterio	Observación
Interfaz de usuario	Correcto.
Conexión a la Base de Datos	Correcto.
Procedimiento para Ingresar Títulos, Párrafos, Sub Párrafos y Objetos Multimedia	Correcto para todos los casos menos en los objetos multimedia, al agregar un objeto multimedia y luego cambiarlo por otro, el archivo temporal anterior aun queda en Disco Duro.

Procedimiento para eliminar Títulos, Párrafos, Sub Párrafos y Objetos multimedia,	Falla cuando no hay ningún elemento seleccionado y se da el comando de eliminar.
Agregar Párrafos de gran extensión	Correcto.
Agregar Objetos multimedia de gran tamaño.	Correcto, pero es un poco demorado, se recomienda mejorar procedimiento de carga.
Funcionamiento de múltiples entornos de agregar.	Correcto.
Prueba de aplicación de los temas del entorno.	Correcto.

Se solucionó el error de eliminar colocando el puntero auxiliar de selección (pAuxSolicitud) en Null (nulo) cada vez que el TTreeView (arbol jerarquico) no tiene foco. Para solucionar el problema del archivo temporal que queda en el disco duro, se agrega una variable al formulario TFrmBuscarObjeto que guarda el string correspondiente al archivo anterior que había sido cargado, luego cada vez que se carga un archivo multimedia, se pregunta si el archivo anterior existe y se procede a borrarlo en caso afirmativo.

➤ **Resultados de las prueba de editar articulo:**

Tabla 9. Resultados de la prueba de editar artículos.

Criterio	Observación
Creación correcta de la estructura del archivo dentro de la interfaz de editar (es la misma que la de agregar)	Correcto.
Conexión a la Base de Datos	Correcto.

Interfaz Grafica	Correcto.
Procedimiento para eliminar Títulos, Párrafos, Sub Párrafos y Objetos multimedia,	Correcto.
Agregar Párrafos de gran extensión	Correcto.
Agregar Objetos multimedia de gran tamaño.	Correcto.
Funcionamiento de múltiples entornos de edición.	Correcto.
Prueba de aplicación de los temas del entorno.	Correcto.

➤ **Resultados de las prueba de Mostrar artículo:**

Tabla 10. Pruebas a mostrar artículo.

Criterio	Observación
Interfaz de usuario.	Correcto.
Conexión a la Base de Datos.	Correcto.
Creación correcta del árbol jerárquico con la estructura real del artículo.	Correcto.
Creación Correcta del cuerpo del articulo	Falla mostrando texto, los cuadros de texto utilizados a veces no alcanzan a abarcar todo el texto y se ve cortado, también los espacios entre textos varían mucho
Funcionan Vínculos para abrir objetos multimedia desde la	Correcto.

interfaz de mostrar articulo.	
Funcionamiento de múltiples entornos de mostrar articulo.	Correcto.
Prueba de aplicación de los temas del entorno.	Correcto.

Se soluciona el problema del texto agregándole un TLabel a TMemoGaia al cual se le pasa el texto que viene de TFrmAgregarArticulo, se aprovecha la propiedad autosize del TLabel y se la pasa el valor Height del TLabel al RichTextBox; no se utiliza el TLabel directamente pues el control TRichTextBox ofrece mejores prestaciones para el manejo de texto.

➤ **Resultados de las prueba de Buscar articulo:**

Tabla 11. Pruebas buscar artículo.

Criterio	Observación
Interfaz de usuario.	Correcto.
Conexión a la Base de Datos.	Correcto.
Verificación de búsqueda por campos.	Correcto.

➤ **Resultados de las prueba de Buscar dentro de articulo:**

Tabla 12. Resultados de la prueba de buscar dentro de articulo.

Criterio	Observación
-----------------	--------------------

Obtención de resultados correctos en la búsqueda	Correcto.
Conexión a la Base de Datos	Correcto.
Interfaz Grafica	Correcto.

➤ **Resultados de las prueba del Laboratorio de Modelado:**

Tabla 13. Resultados de la prueba al laboratorio de modelado.

Criterio	Observación
Resultados correctos de las simulaciones	Correcto.
Conexión a la Base de Datos (para obtener archivos desde la base de datos)	Correcto.
Interfaz Grafica	Correcto.
Aplicación de las propiedades de simulación .	Correcto.
Cambio de las propiedades de la grafica.	Correcto.
Funcionamiento del editor de Forrester	Correcto.
Funcionamiento de múltiples instancias del laboratorio de modelado.	Correcto.
Funcionamiento del entorno dividido de trabajo (editor forrester y graficador).	Correcto.
Prueba de aplicación de los temas	Correcto.

del entorno.	
--------------	--

➤ **Resultados de las prueba de Centros de Navegación:**

Tabla 14. Pruebas a centros de navegación.

Criterio	Observación
Interfaz de usuario.	Correcto.
Conexión a la Base de Datos.	Correcto.
Casos de uso agrupados adecuadamente.	Correcto.
Vínculos correctos	Correcto.
Prueba de aplicación de los temas del entorno.	Correcto.

➤ **Resultados de las prueba de guardar y cargar artículos de disco duro:**

Tabla 15. Resultados de la prueba de guardar y cargar artículos en disco duro.

Criterio	Observación
Guarda correctamente archivo.	Correcto.
Carga correctamente archivo.	Solo carga desde el directorio raíz (C:\)
Verificación de integridad de la información del articulo guardado.	Correcto.
Guardar en un computador con el software y cargarlo en otro.	Correcto.

➤ **Resultados de las prueba del Glosario:**

Tabla 16. Pruebas a glosario.

Criterio	Observación
Interfaz de usuario.	Correcto.
Conexión a la Base de Datos.	Correcto.
Editar Concepto.	Correcto.
Filtrar Artículos a relacionar.	Correcto.
Relacionar concepto con artículo(s).	Correcto.
Agregar concepto.	Correcto, se recomienda limpiar la lista de artículos relacionados y la caja de texto del concepto.

➤ **Resultados de las prueba de Manejo de Usuarios:**

Tabla 17. Pruebas al caso de uso de manejo de usuarios.

Criterio	Observación
Iniciar/cambiar sesión y filtrar archivos por usuario.	Correcto.
Conexión con la Base de Datos.	Correcto.
Agregar Usuarios.	Correcto.
Editar Usuarios.	Correcto.
Eliminar Usuarios.	Correcto.

➤ **Resultados de las prueba de Temas Relacionados:**

Tabla 18. Pruebas al caso de uso temas relacionados.

Criterio	Observación
Relacionar artículos.	Correcto.
Conexión con la Base de Datos.	Correcto.
Filtrar Artículos a relacionar.	Correcto.
Deshacer Relación.	Correcto.

El software también fue puesto a prueba con los estudiantes de la especialización en ingeniería ambiental de la UIS de las sedes de Bogota y Bucaramanga, ellos dieron algunas recomendaciones respecto a la interfaz de usuario y su interacción con el software ayudo a depurar el código fuente.

Como todo software, es posible que se presenten errores inesperados durante la utilización del mismo, la intención es minimizar los errores al máximo, en el momento de escribir este documento, el software se presentaba estable, pero con el afán de una mayor depuración y de obtener un producto libre de errores, se iniciaron mas pruebas con la ayuda de un grupo de estudiantes del Centro de Estudios e Investigaciones Ambientales (CEIAM); por otra parte el software será utilizado en los cursos de Dinámica de Sistemas de la Especialización en Ingeniería Ambiental de la Universidad Industrial de Santander, lo que permite depurar aún mas la herramienta de forma tal que el producto sea lo suficientemente confiable y pueda ofrecerse a otros centros de estudios en el país y divulgar su uso, no solo en el campo ambiental sino también en cualquier área del conocimiento donde se pretenda o donde se quiera utilizar las características y prestaciones de las guías de autoaprendizaje apoyadas en dinámica de sistemas.

5. CONCLUSIONES

Se llevó a cabo un proceso investigativo que dió como resultado un compendio de información que muestra los estudios realizados a nivel internacional en el tema del medio ambiente utilizando la Dinámica de sistemas; a nivel nacional los trabajos al respecto son escasos y no se encontró ninguna investigación importante. Así, los resultados de esta recopilación presentados en la herramienta pueden servir como punto de inicio para investigaciones de fenómenos ambientales utilizando la dinámica de sistemas.

Se realizó un trabajo de reproducción y elaboración de modelos; la reproducción corresponde al desarrollo de los modelos del libro Modeling the environment de Andrew Ford usando el software evolución 3.5 y el modelo de la tesis "Efectos de Contenidos Húmedos en Rellenos Sanitarios" la cual es de gran importancia para el Centro de Estudios e Investigaciones Ambientales (CEIAM). La elaboración de modelos se llevo a cabo en un trabajo conjunto con los estudiantes de Sistemas Dinámicos 2 del primer y segundo semestre del año 2003 de la materia Sistemas Dinámicos 2; éste trabajo enriquece la formación como ingenieros pues permite al futuro profesional en esta rama del conocimiento desarrollar habilidades y destrezas para realizar labores de docencia.

Se desarrolló una herramienta software que organiza toda la información resultante del proceso investigativo y de la reproducción y desarrollo de modelos; ésta organización se hace por medio de los documentos interactivos de la herramienta, los cuales permiten incluir texto, imágenes, modelos, videos, audio y otros tipos de objetos OLE. Los documentos interactivos son la base para organizar la información pero

igualmente la herramienta brinda opciones para organizar dichos documentos y amplias facilidades de búsqueda exhaustiva para así encontrar fácilmente la información requerida por el aprendiz; también la herramienta cuenta con un laboratorio de modelado con dinámica de sistemas que permite manipular todos los modelos que se encuentran dentro de los documentos interactivos. El software cuenta también con características propias de una herramienta de autor pues permite al aprendiz crear sus propios documentos, editarlos y eliminarlos, es decir, realizar un mantenimiento de los contenidos; además posee facilidades de portabilidad de información ya que permite guardar a disco duro los documentos interactivos y de esta forma es posible que usuarios de la herramienta intercambien documentos interactivos entre ellos fomentando así el trabajo en equipos entre la futura comunidad de usuarios de la herramienta.

Gracias al proceso de desarrollo del software se fortalecieron los conocimientos adquiridos a lo largo de la carrera, y se profundizaron algunos como Dinámica de Sistemas, construcción de software orientado a objetos, Proceso Unificado De Desarrollo de Software y UML. Además se aprendió a utilizar la herramienta Flash para el desarrollo de las animaciones correspondientes a los tutoriales que se encuentran dentro de la herramienta.

Finalmente el trabajo realizado en equipo para el desarrollo de la herramienta y para el desarrollo de los modelos con los alumnos enriquece al futuro ingeniero y lo prepara para su desempeño profesional dentro de grupos de trabajo.

La herramienta GAIA 1.0 es el resultado de este trabajo conjunto de investigación, elaboración y reproducción de modelos y desarrollo de software; y tiene el afán de difundir el uso de la dinámica de sistemas a

las personas que estudian los fenómenos del medio ambiente y servir como punto de inicio a investigadores que quieran entender y aplicar la dinámica de sistemas en el tema ambiental.

6. RECOMENDACIONES

Es importante dejar plasmadas un conjunto de recomendaciones que sirvan como guía para personas que deseen desarrollar una nueva versión del software.

- En una futura versión del software sería bastante útil desarrollar un modulo de red que permita intercambiar fácilmente los documentos interactivos de cada usuario y que además tenga las facilidades propias de los foros de discusión para de esta forma apoyar el trabajo en grupos de usuarios.
- Se podrían desarrollar juegos educativos que hagan aun más ameno el proceso de aprendizaje de la Dinámica de Sistemas.
- Es posible que se desarrollen animadores en tres dimensiones para que los resultados de la simulación sean más vistosos gráficamente. P. Ej. Representar el crecimiento de un bosque mediante un animador 3D que permita observar el crecimiento de arbolitos que se van dibujando dependiendo de los valores del Nivel y las variables que intervengan en el diagrama de forrester
- Implementar la impresión de los documentos interactivos y la funcionalidad de hacer y deshacer.
- En general las nuevas adiciones al software deben contribuir con el propósito de hacer agradable el aprendizaje de la dinámica de sistemas y agregarle más prestaciones a los llamados documentos interactivos que son creados dentro de la herramienta.

7. BIBLIOGRAFIA

ANDRADE SOSA, Hugo Hernando; DYNER, Isaac; ESPINOSA, Angela; LOPEZ GARAY, Hernan y SOTAQUIRA, Ricardo. Pensamiento Sistémico: Diversidad en Busca de Unidad. Primera Edición. Bucaramanga: Ediciones Universidad Industrial de Santander, 2001. 423p.

ARACIL SANTONJA, Javier. Introducción a la Dinámica de Sistemas. Primera Edición. Madrid: Alianza Editorial, 1992. 209 p.

BERNSTEIN, J. Einstein. El hombre y su obra. Traducción de José Luis Sebastián Franco de la obra original Einstein. Editorial Mc Graw Hill. Interamericana de España S.A. Madrid, 1993.

BERTALANFY, Ludwing Von. Introducción a la Teoría General de los sistemas. Primera Edición. México D.F.: Fondo de cultura económica, 1986. 297p.

CRAIG P. Eck, B.S, Effects of moisture contenet in solid waste landfills, thesis.

ENKERLIN; CANO; CORREA Y ROBLES. Vida, ambiente y desarrollo en el siglo XXI: lecciones y acciones. Primera Edición. Mexico D.F.: Grupo editorial Iberoamerica, 2000. 258p.

FORD, Andrew. Modeling the environment: An Introduction to system dynamics modeling of the environment. Primera Edición. Washington: Island Press, 1999. 401p.

J. GLYNN; GARY HEINKE. Ingeniería Ambiental. Segunda Edición. México D.F.: Prentice Hall, 1999. 425p.

JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James. El Proceso Unificado de desarrollo de software. Madrid. Adison Wesley, 2000. 438p.

LARMAN, Craig. UML y Patrones: Una Introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda Edición. Madrid: Prentice Hall, 2003. 590p.

LOVELOCK, J. E. Gaia, A new Look at Life on Herat. Oxford University Press. Oxford, 1979.

LUDEVID ANGLADA, Manuel. El cambio global en el medio ambiente, introducción a sus causas humanas. Primera Edición. Mexico D.F.: AlfaOmega Editores, 1998. 332p.

MARTINEZ, Silvio; REQUENA, Alberto. Dinámica de sistemas I. Simulación por Ordenador. Alianza Editorial, 1996. 185p.

MARTINEZ, Silvio; REQUENA, Alberto. Dinámica de sistemas I. Simulación Modelos. Alianza Editorial, 1996. 295p.

MIHELIC, James. Fundamentos de Ingeniería ambiental. Primera Edición. México D.F.: Limusa Wiley, 1998. 332p.

MILLER, G. Tyler Jr. 1998 Living in the environment, Belmont, CA; Wadsworth Publishing.

O'CONNOR, Joseph. Introducción al pensamiento sistémico. Primera Edición. Barcelona: URANO, 1997. 302p.

SENGE, Peter M. La Quinta Disciplina en la práctica. Primera Edición. México D.F.: Garnica, 1999. 490p.

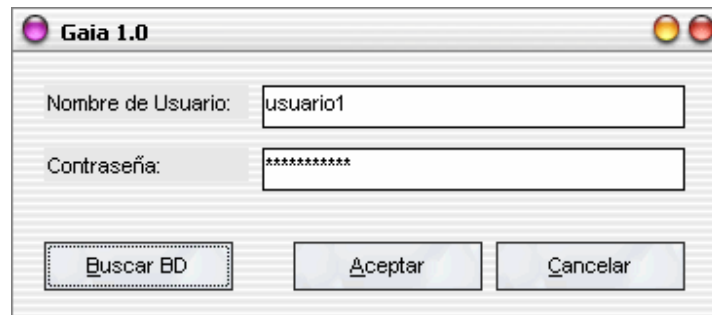
VERNADSKY, V.I. La Biosphere and the Noösphere. American Scientist Journal 33, enero 1945. pp. 1-12. USA, 1945.

VERNADSKY, V.I. La Biosphere. Libraire Felix Alcan. Paris, 1929.

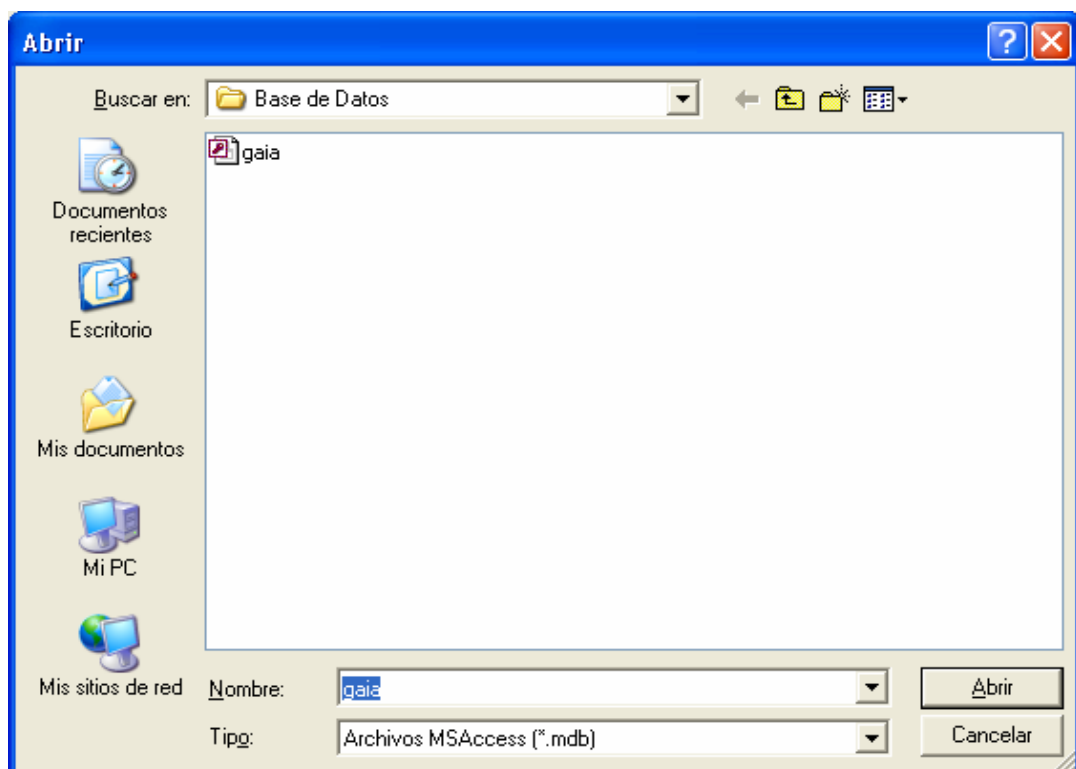
VEGA MORA, Leonel. Gestión ambiental sistémica Un nuevo enfoque funcional y organizacional para el fortalecimiento de la gestión ambiental pública empresarial y ciudadana en el ámbito estatal. Primera Edición. Bogota: SIGMA, 2001. 280p.

ANEXOS

ANEXO A. MANUAL DEL USUARIO

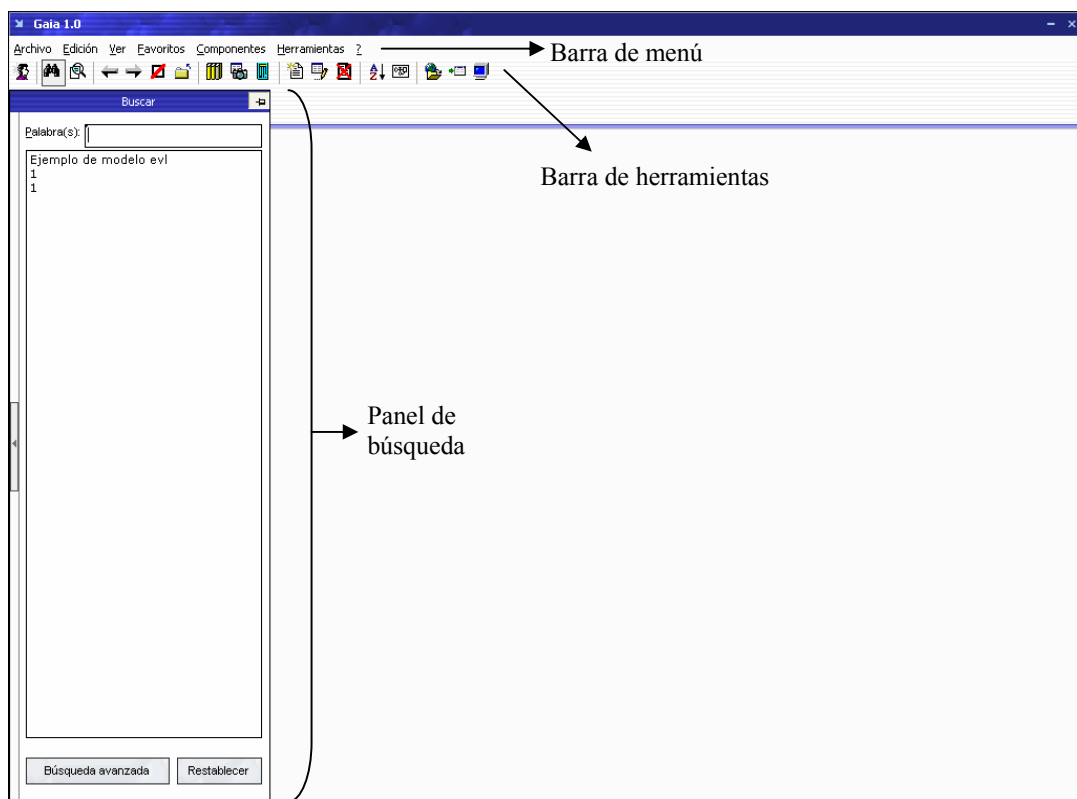


Al iniciar GAIA se muestra la ventana de registro de usuario en la cual se digitan el nombre de usuario y la contraseña para acceder a la aplicación, esta ventana también ofrece la opción de cargar una base de datos desde donde el usuario lo requiera presionando el botón Buscar BD.



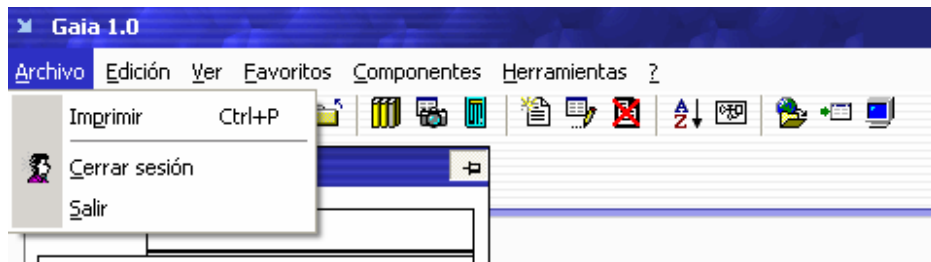
Al presionar este botón se abre la ventana de abrir archivo donde se busca el origen de datos.

Habiendo cargado el origen de datos y después de identificarnos ingresamos a la aplicación donde encontramos los siguientes elementos en el entorno principal de la aplicación:

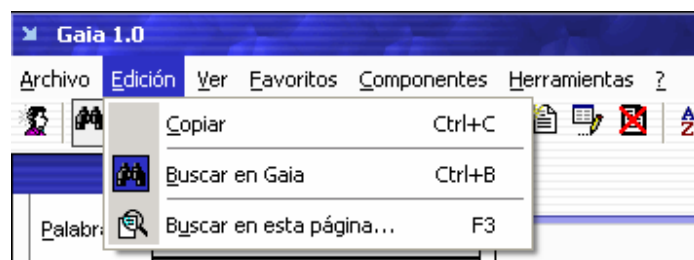


Una barra de herramientas, una barra de menú y el panel de búsqueda.

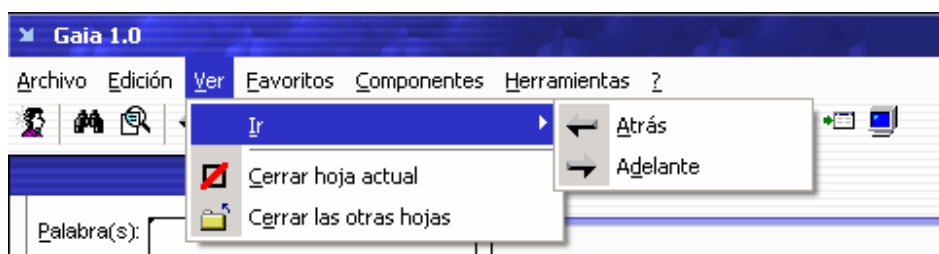
En la barra de menú encontramos Archivo, Edición, Ver, Favoritos, Componentes, Herramientas y Ayuda.



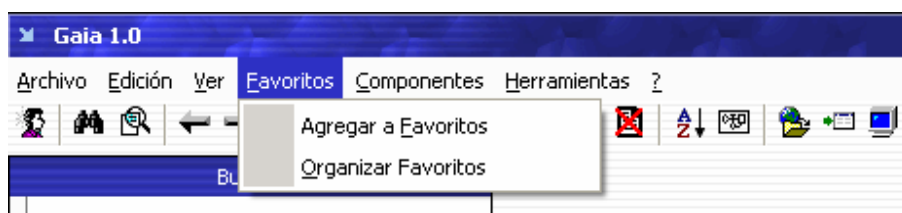
Dentro del menú de archivo encontramos las opciones de Imprimir, Cerrar sesión y Salir. La opción Imprimir no ha sido implementada en esta versión del software, la opción Cerrar sesión permite volver a la ventana de registro de usuario para ingresar como otro usuario o cambiar el origen de datos, y la opción Salir cierra la aplicación.



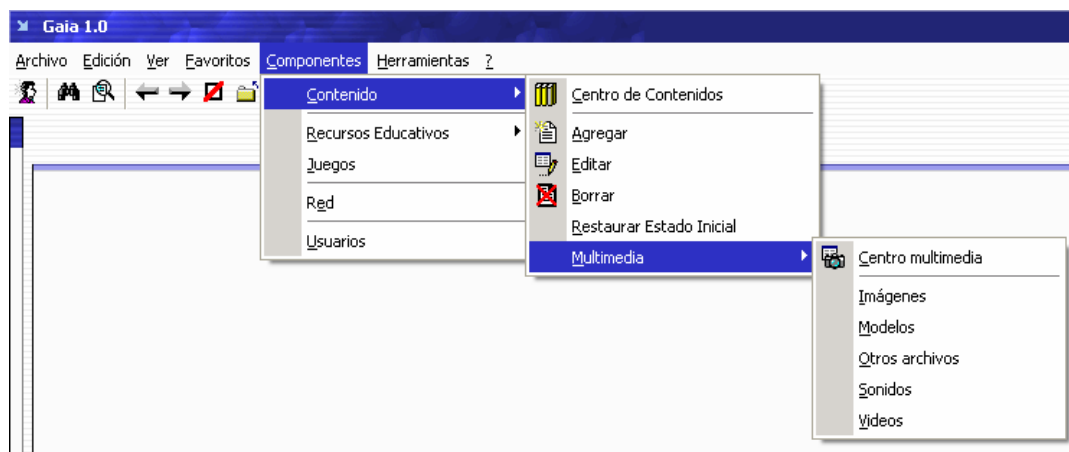
Dentro del menú de Edición encontramos las opciones de Copiar, Buscar en Gaia y Buscar en esta página. La opción Copiar no esta implementada en esta versión del software, la opción Buscar en Gaia permite localizar un artículo contenido en el origen de datos previamente cargado utilizando el panel de búsqueda, y la opción Buscar en esta página permite ubicar una expresión dentro del articulo que se esta trabajando.



Dentro del menú Ver encontramos las opciones Ir, Cerrar hoja actual, Cerrar las otras hojas. La opción Ir despliega un submenú con las opciones Atrás y Adelante para navegar por las hojas que estén abiertas, la opción Cerrar hoja actual permite cerrar la hoja en la que se este trabajando en ese momento, y la opción Cerrar las otras hojas permite cerrar todas las hojas menos en la que se este trabajando en ese momento.



El menú Favoritos tiene las opciones Agregar a Favoritos y Organizar Favoritos pero estas funciones no se implementaron en esta versión del software.

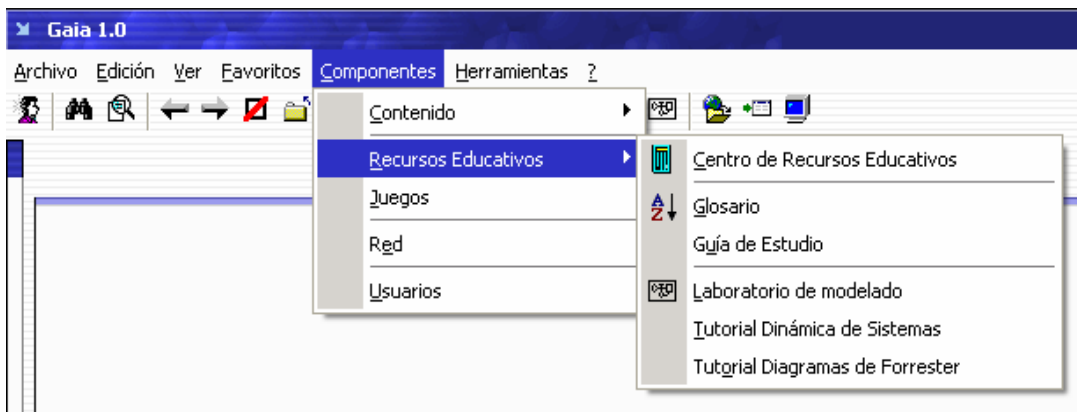


En el menú Componentes se encuentran las opciones de Contenido, Recursos educativos, Juegos, Red y Usuarios. La opción Contenido despliega un submenú con las opciones Centro de contenidos, que abre un entorno de navegación que facilita la utilización de los contenidos de Gaia; Agregar, que permite acceder al entorno de agregado de artículos para crear un nuevo artículo; Editar, que activa o desactiva el modo de

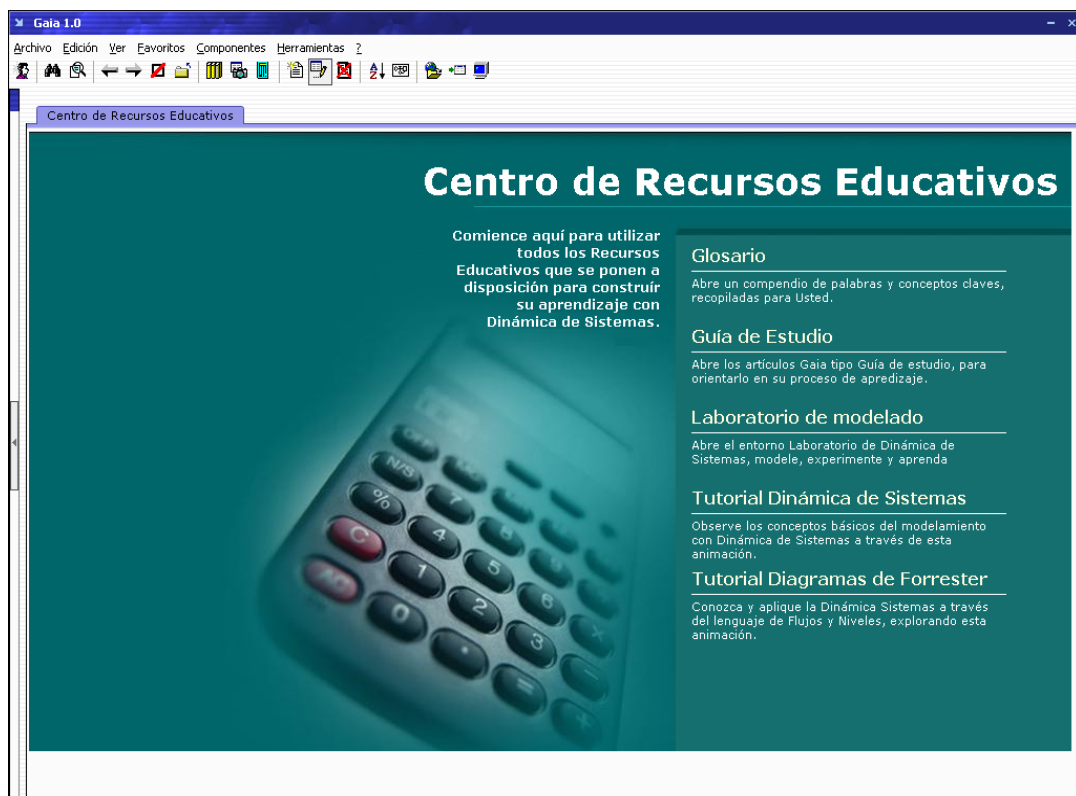
edición para modificar el contenido de un artículo, para que sea modificado el artículo debe abrirse después de que se active el modo de edición; Borrar, que elimina el artículo en el que se este trabajando de la base de datos; Restaurar Estado Inicial, que devuelve la base de datos a su estado inicial borrando todos los artículos que no estaban en la base de datos original y cancelando las modificaciones hechas a los artículos que estaban en la base de datos original; y Multimedia, que despliega un submenú con las opciones de Centro de multimedia, que también es un entorno de navegación para facilitar la utilización de las opciones de multimedia, y las opciones de Imágenes, Modelos, Otros archivos, Sonidos y Videos, que sirven para filtrar los artículos que se muestran en el panel de búsqueda según tengan algunos de estos elementos contenidos en ellos.

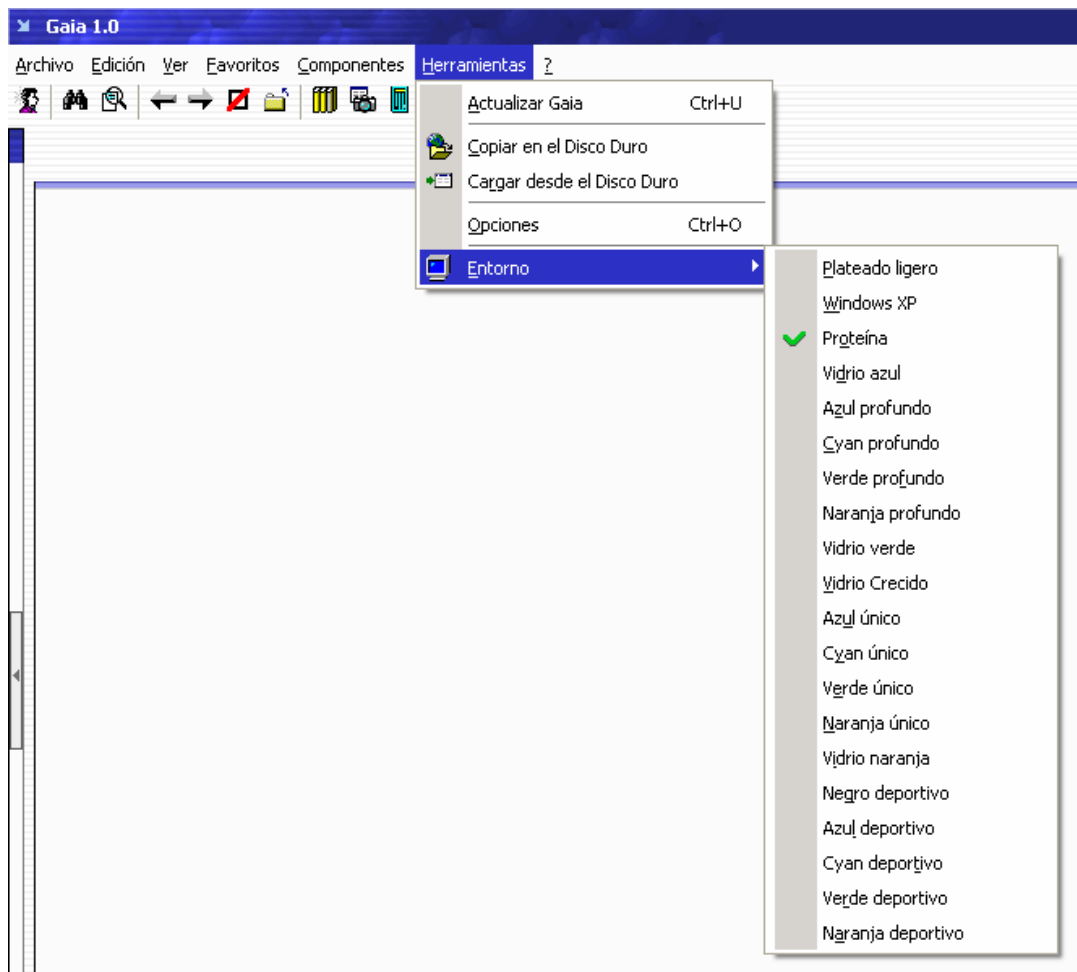
Los entornos de navegación Centro de contenidos y Centro de multimedia ofrecen opciones de navegación contenidas en los menús de multimedia y contenidos y claramente explicadas.



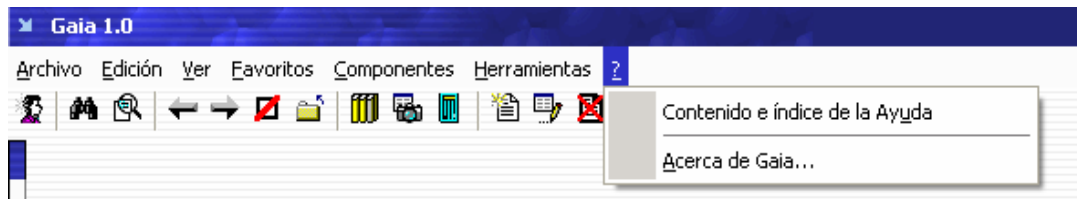


La opción de Recursos Educativos despliega un submenú con las opciones Centro de recursos educativos, que es también un entorno de navegación; Glosario, que permite buscar un concepto y visualizar su definición y los temas relacionados a este concepto, dentro del glosario también se pueden agregar nuevos conceptos y modificar los ya existentes; Guía de estudio, que pone en el panel de búsqueda todos los artículos que sean del tipo Guía de estudio; Laboratorio de modelado, que abre el entorno de modelación donde se construyen los modelos que se incluirán en los artículos; y los tutoriales de dinámica de sistemas, que le explica al usuario como elaborar modelos teniendo en cuenta la metodología de la dinámica de sistemas; y diagramas de Forrester, que le ofrece al usuario otra alternativa para la elaboración de modelos con los diagramas de Forrester. Las opciones de Juegos y Red tampoco están implementadas en esta versión del software; la opción usuarios permite que se modifiquen los datos de un usuario ya existente y que se creen nuevos usuarios o se eliminen viejos.





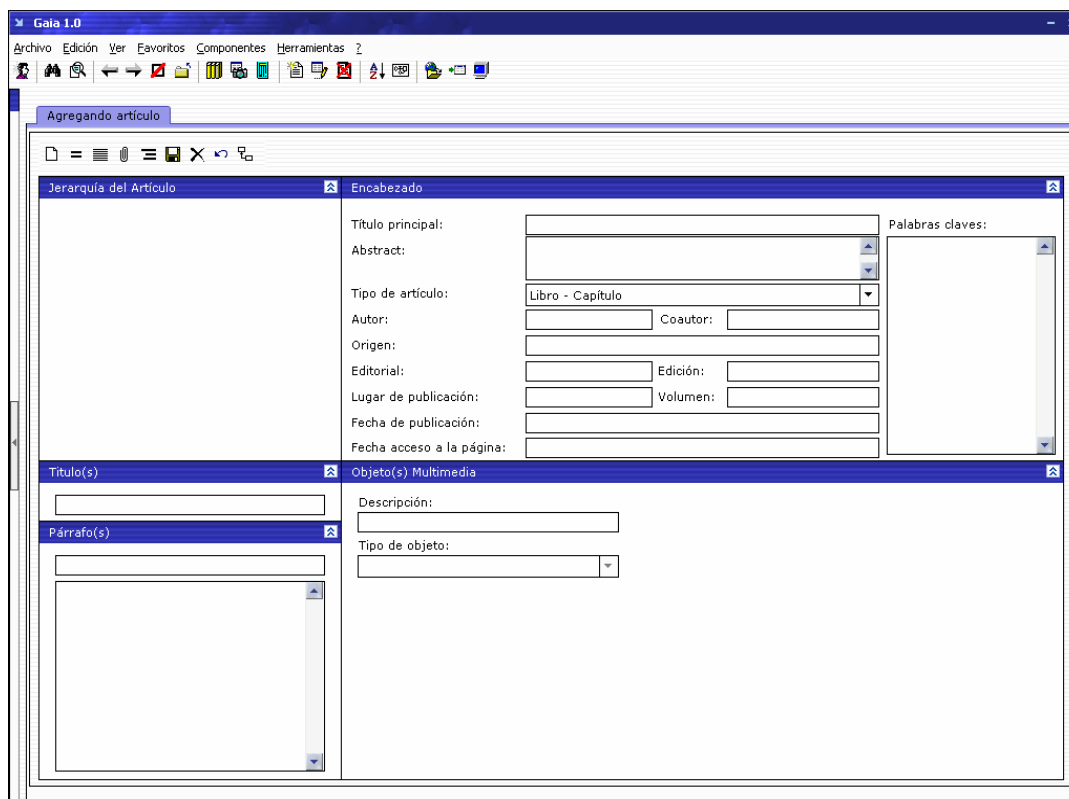
En el menú herramientas se encuentran las opciones Actualizar Gaia, Copiar en el disco duro, Cargar desde el disco duro, Opciones y Entorno. Las opciones Actualizar Gaia y Opciones no se implementaron en esta versión del software; las opciones Copiar en el disco duro y Cargar desde el disco duro permiten respectivamente crear un archivo .GAI donde esta se guarda un artículo con todo su contenido y cargar este archivo para visualizar su contenido; y la opción Entorno que permite escoger un skin para modificar el ambiente de trabajo de la aplicación.



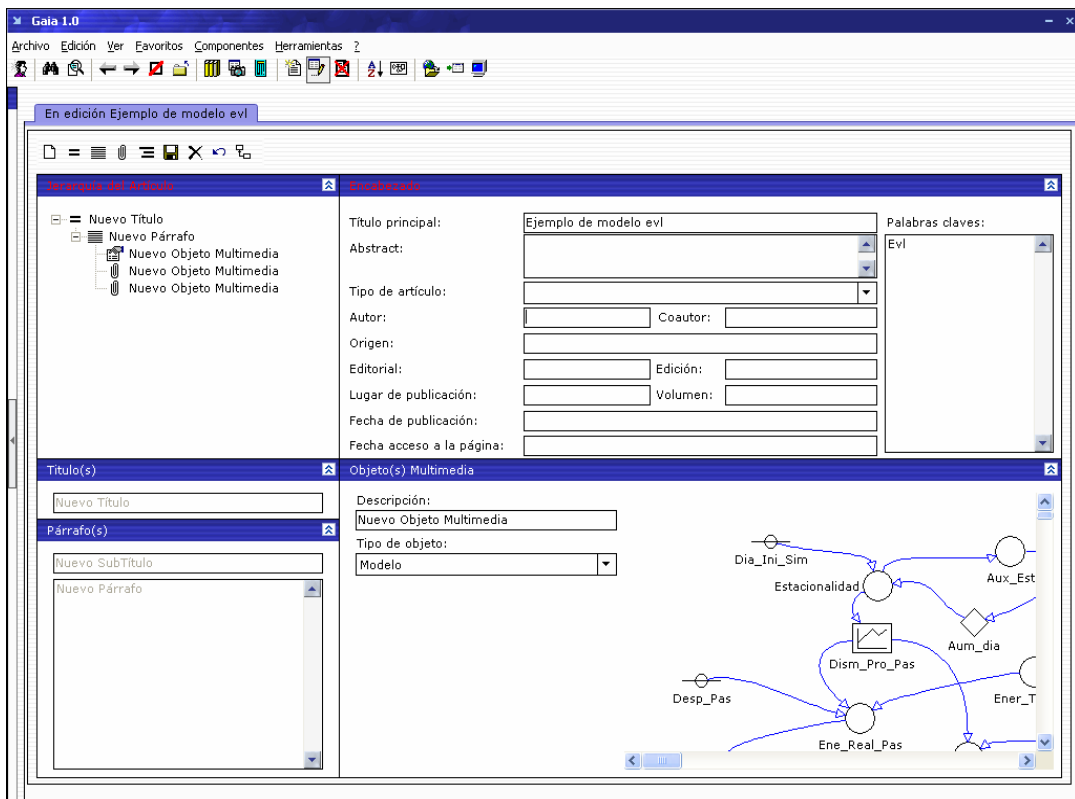
Por último el menú de ayuda ofrece las opciones de Contenido e índice de la ayuda y acerca de Gaia. La opción de Contenido e índice de la ayuda nos permite acceder a las ayudas y buscar por temas lo que necesitemos acerca del funcionamiento de la aplicación; y la opción de Acerca de Gaia nos muestra información acerca de la elaboración de la aplicación.



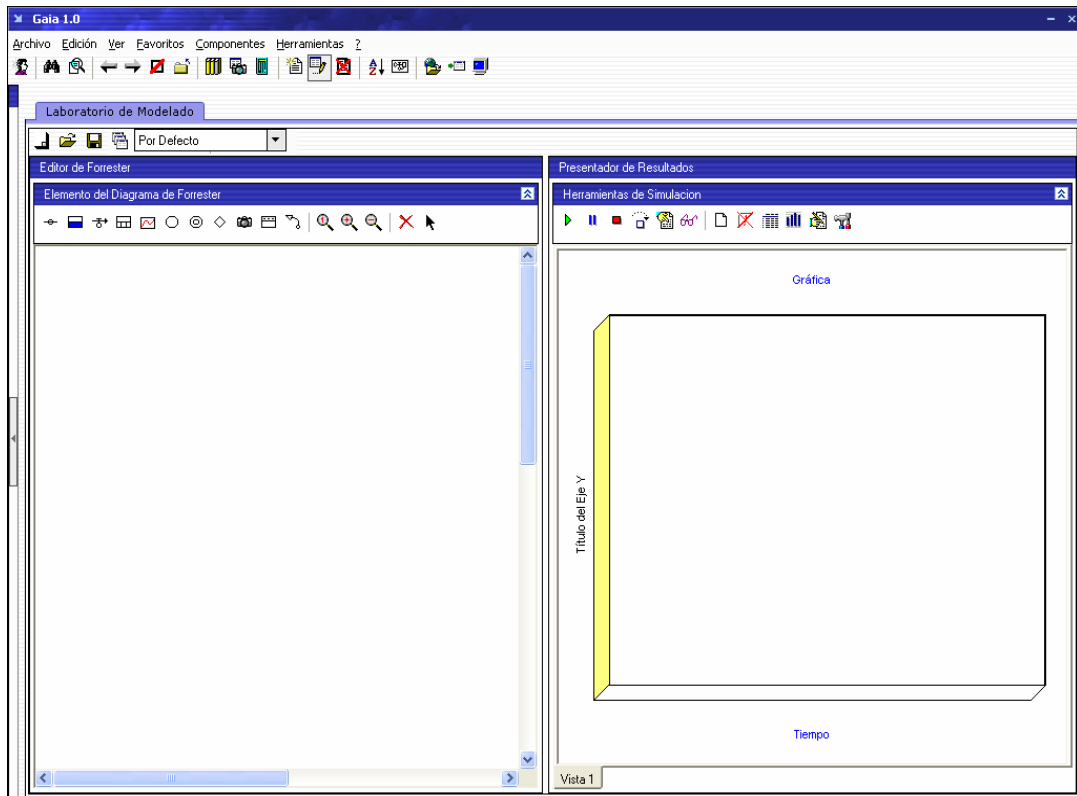
Todos los iconos en la barra de herramientas están contenidos en los menús y su funcionamiento ya ha sido explicado.



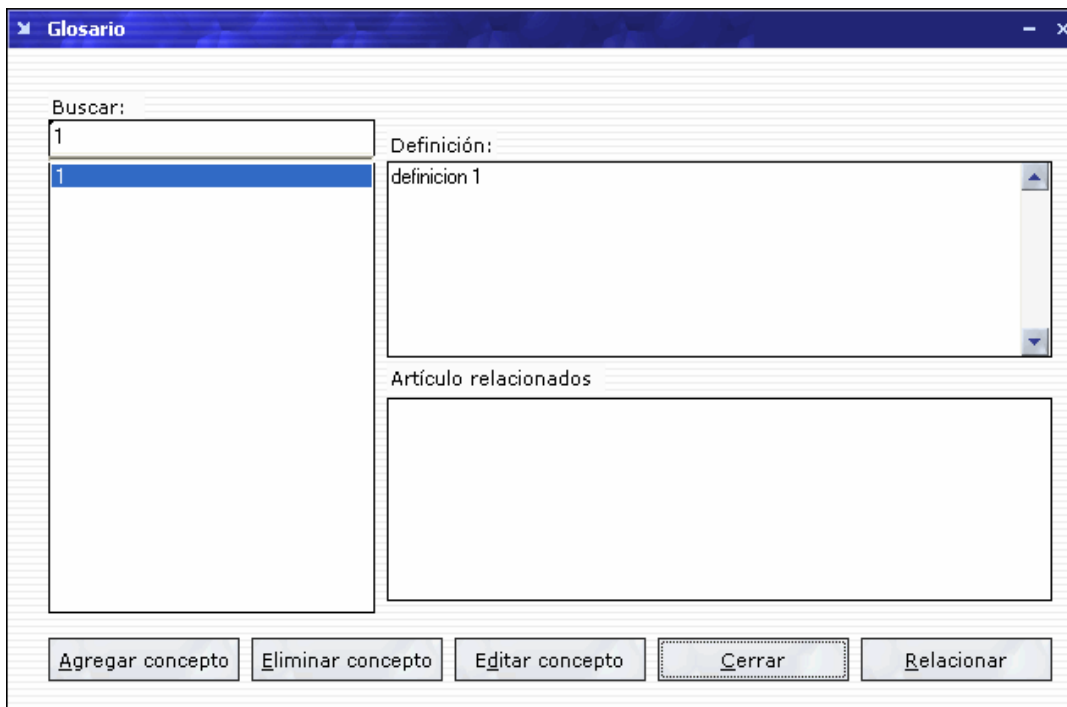
Para agregar un nuevo artículo se tiene que abrir la hoja agregando artículo en la cual encontramos una plantilla de trabajo en la que se suministra información acerca del contenido del artículo que vamos a agregar y nos permite, mediante una pequeña barra de herramientas en la parte superior izquierda de la hoja, inicializar la plantilla de trabajo, es decir volver a poner en blanco la plantilla, insertar títulos, párrafos subpárrafos y elementos multimedia al contenido del artículo, grabar el artículo, eliminar el elemento del artículo que sea seleccionado, deshacer un cambio y establecer una relación con uno o varios de los artículos existentes.



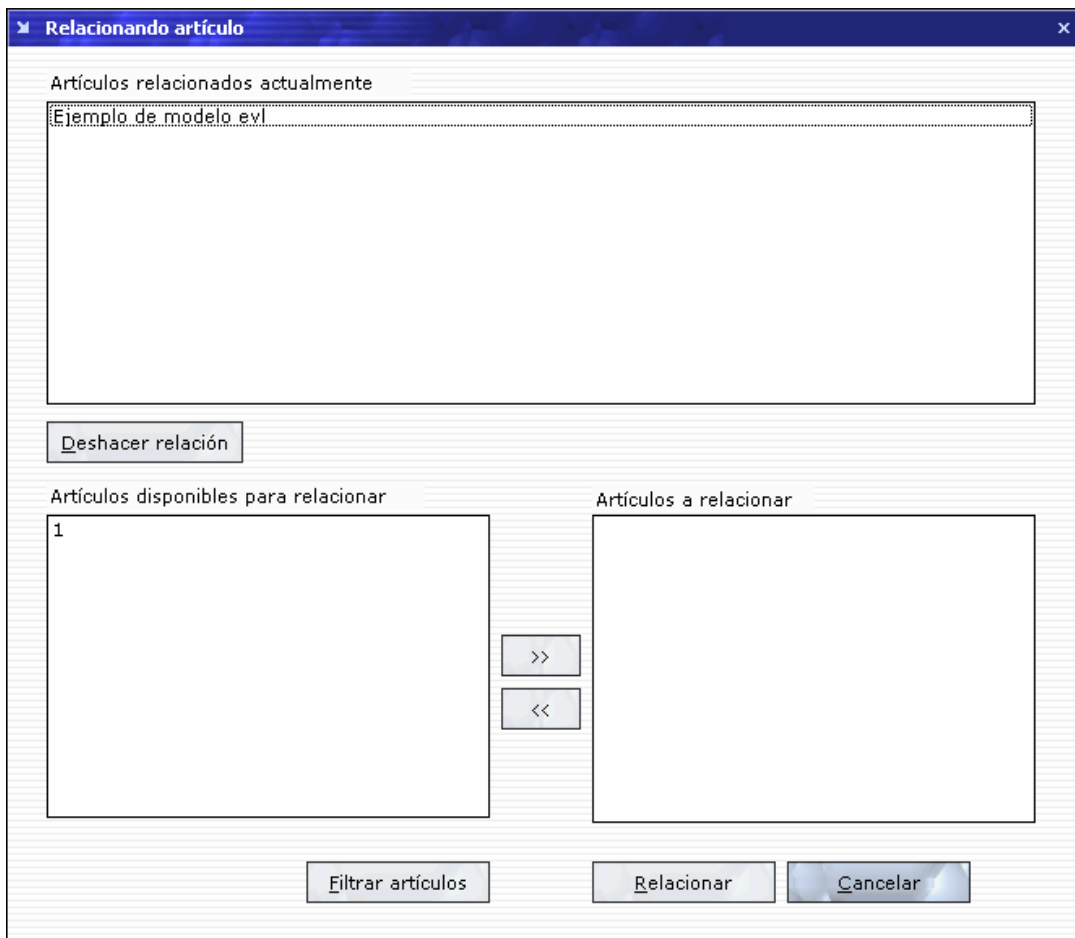
Para editar un artículo primero hay que activar el modo de edición y después abrir el artículo, para activar el modo de edición se puede hacer clic sobre el icono en la barra de herramientas o en el menú Componentes, Contenido, Editar, o a través del entorno de navegación Centro de Contenidos, cuando el modo de edición este activo, el icono editar tanto en la barra de herramientas como en el menú Contenido aparecerán activos y al abrir un articulo, el título de la hoja aparecerá precedido de la frase "En edición", entonces se podrán hacer cambios en su contenido.



En el laboratorio de modelado se crean los modelos que se pueden añadir a los artículos, en el panel de la izquierda se construye el modelo y en el de la derecha se corre la simulación, con ayuda de la barra de herramientas en la parte superior izquierda de la hoja se puede guardar o abrir un modelo, también ofrece la opción de inicializar el laboratorio de modelado y cambiar el escenario.



En el glosario se ofrecen las opciones de Agregar, Eliminar, Editar, Relacionar y buscar un concepto o definición de un término que nos resulte desconocido.



Para relacionar artículos entre sí, estos ya deben de estar creados, esta opción permite establecer relaciones entre artículos del mismo tema.

ANEXO B. GUIA DEL PROGRAMADOR

La siguiente lista de unidades .pas conforma Gaia 1.0, con su respectiva declaración de clases y la funcionalidad en general de cada unidad.

Primero que todo, se siguió con los siguientes parámetros de programación:

- El nombre físico de las Unidades .pas, vienen precedidas por la vocal U (*UAcceso.pas*).
- El nombre de las variables que son parte de cada una de las clases, están precedidas por la letra F (*FBuscarUsuario*).
- El nombre de las clases diseñadas e implementadas en Gaia, se le antepone la letra T (*TUsuario*).
- Para determinar los bloques de código fácilmente, es decir, un bloque for, un bloque if – else, etc.; se debe tabular según el nivel de profundidad que sea necesario, por ejemplo:

```
for i:=0 to Consulta.Enlazar.DataSet.RecordCount-1 do
begin
  PauxParrafo:=ParrafoPadre;
  cmdSubParrafoClick(nil);
  (PauxSubParrafo^ as TParrafo).Conectarse(frmAcceso.Usuario.DirBaseDatos,'Parrafo');
  (PauxSubParrafo^ as TParrafo).Editar(Consulta.Enlazar.DataSet['IDParrafo']);
  (PauxSubParrafo^ as TParrafo).SubTitulo:=Consulta.Enlazar.DataSet['Subtitulo'];
  txtSubtitulo.Text:=(PauxSubParrafo^ as TParrafo).SubTitulo;
  (PauxSubParrafo^ as TParrafo).Descripcion.Text:=Consulta.Enlazar.DataSet['Descripcion'];
  txtParrafo.Text:=(PauxSubParrafo^ as TParrafo).Descripcion.Text;
  (PauxSubParrafo^ as TParrafo).Cerrar;
  PauxParrafo:=PauxSubParrafo;
  EditarObjetosMultimedia(Consulta.Enlazar.DataSet['IDParrafo']);
  EditarSubParrafos(Consulta.Enlazar.DataSet['IDParrafo'],PauxSubParrafo);
  Consulta.Enlazar.DataSet.Next;
End;
```

- Es muy importante que durante la codificación del software, se hagan comentarios al respecto de determinado segmento de programación.
- Se aconseja que el nombre de las funciones y procedimientos sean muy dicentes, preferible usar nombres largos.
- El nombre de las variables deben ser muy explicitos y se les debe agregar un prefijo que indique de que tipo es (frmAgregarArticulo, PTSolicitud, txtNombreUsuario).

UAcceso.pas

Funcionalidad: La presente unidad es el formulario de inicio de sesión, como el cambio de sesión.

Instancia de clases:

```

type
TfrmAcceso = class(TForm)
  suiForm1: TsuiForm;
  txtUsuario: TsuiEdit;
  txtContrasena: TsuiEdit;
  cmdAceptar: TsuiButton;
  cmdCancelar: TsuiButton;
  suiEdit1: TsuiEdit;
  suiEdit2: TsuiEdit;
  BuscarBD: TOpenDialog;
  cmdBuscarBD: TsuiButton;
  procedure suitepcmdCancelarClick(Sender: TObject);
  procedure suitepcmdAceptarClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
  procedure cmdBuscarBDClick(Sender: TObject);
private
  { Private declarations }
  FBuscarUsuario:TUsuario;
public
  { Public declarations }
published
  property Usuario:TUsuario read FBuscarUsuario;
end;

```

UAcercadeGaia.pas

Funcionalidad: Es el formulario que presenta el acerca de Gaia.

Instancia de clases:

```
type
TfrmAcercaGaia = class(TForm)
  suiAcercadeGaia: TsuiForm;
  ricContenido: TRichEdit;
  Shape1: TShape;
  lblTitulo: TLabel;
  cmdContactenos: TsuiButton;
  cmdCreditos: TsuiButton;
  cmdContribuyentes: TsuiButton;
  cmdConsultantes: TsuiButton;
  procedure cmdContactenosClick(Sender: TObject);
  procedure cmdCreditosClick(Sender: TObject);
  procedure cmdContribuyentesClick(Sender: TObject);
  procedure cmdConsultantesClick(Sender: TObject);
  procedure FormShow(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

UTemasRelacionados.pas

Funcionalidad: Formulario donde se muestran los respectivos artículos relacionados de cada uno de los artículos de Gaia, es mostrado en el Panel de Accesorios de los artículos.

Instancia de clases:

```
type
TfrmTemasRelacionados = class(TForm)
  lstArticulosRelacionados: TsuiDBLookupListBox;
  suiScrollBar1: TsuiScrollBar;
  procedure FormCreate(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure lstArticulosRelacionadosDbClick(Sender: TObject);
private
  { Private declarations }
end;
```

```

FArticulosRelacionados:TConsulta;
FIdentificador:Longint;
FMostrar, FComparar, FOperadores, FParametros, FEnlace:TStringList;
procedure OnPosChange(var Msg: TWmWindowPosChanging);
    message WM_WINDOWPOSCHANGING;
procedure ConsultarArticulosRelacionados;

public
    { Public declarations }
published
    property IdArticulo:Longint write FIdentificador;
end;

```

UAgregarArticulo.pas

Funcionalidad: Formulario de interfaz, donde el usuario construye y guarda sus respectivos artículos, es una de las unidades más importantes en donde se muestra un control exhaustivo con el manejo de punteros e información en memoria.

Instancia de clases:

```

type
    PTSolicitud = ^TSolicitud;

TfrmAgregarArticulo = class(TForm)
    SUIAgregarArticulo: TsuiForm;
    PanEncabezado: TsuiPanel;
    txtAutor: TsuiDBEdit;
    txtCoautor: TsuiDBEdit;
    txtOrigen: TsuiDBEdit;
    txtEditorial: TsuiDBEdit;
    txtEdicion: TsuiDBEdit;
    txtVolumen: TsuiDBEdit;
    txtLugarPublicacion: TsuiDBEdit;
    txtFechaPublicacion: TsuiDBEdit;
    txtFechaAccesoPagina: TsuiDBEdit;
    PanTitulo: TsuiPanel;
    PanParrafo: TsuiPanel;
    PanObjetoMultimedia: TsuiPanel;
    PanJerarquia: TsuiPanel;
    treJerarquia: TsuiTreeView;
    txtTituloPrincipal: TsuiDBEdit;

```

```
txtAbstract: TsuiDBMemo;
suiToolBar1: TsuiToolBar;
cmdBorrarPlantilla: TToolButton;
cmdTituloNuevo: TToolButton;
cmdParrafoNuevo: TToolButton;
cmdEliminar: TToolButton;
cmdDeshacer: TToolButton;
ImageList1: TImageList;
cmdGuardar: TToolButton;
cmdNuevoObjetoMultimedia: TToolButton;
suiScrollBar2: TsuiScrollBar;
cmdSubParrafo: TToolButton;
suiEdit1: TsuiEdit;
suiEdit2: TsuiEdit;
suiEdit3: TsuiEdit;
suiEdit4: TsuiEdit;
suiEdit5: TsuiEdit;
suiEdit6: TsuiEdit;
suiEdit7: TsuiEdit;
suiEdit8: TsuiEdit;
suiEdit9: TsuiEdit;
suiEdit10: TsuiEdit;
suiEdit11: TsuiEdit;
suiEdit12: TsuiEdit;
suiEdit13: TsuiEdit;
suiEdit14: TsuiEdit;
cmbTipoArticulo: TsuiDBComboBox;
txtTitulo: TsuiEdit;
txtSubtitulo: TsuiEdit;
txtParrafo: TsuiMemo;
txtObjetoMultimedia: TsuiEdit;
cmbTipoObjeto: TsuiComboBox;
EspacioObjeto: TScrollBar;
suiEdit15: TsuiEdit;
txtPalabrasClaves: TsuiDBMemo;
suiScrollBar1: TsuiScrollBar;
suiScrollBar3: TsuiScrollBar;
cmdRelacionar: TToolButton;
procedure PanEncabezadoEnter(Sender: TObject);
procedure PanEncabezadoExit(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure cmdBorrarPlantillaClick(Sender: TObject);
```

```

procedure cmdTituloNuevoClick(Sender: TObject);
procedure cmdParrafoNuevoClick(Sender: TObject);
procedure cmdNuevoObjetoMultimediaClick(Sender: TObject);
procedure cmdGuardarClick(Sender: TObject);
procedure cmdEliminarClick(Sender: TObject);
procedure cmdSubParrafoClick(Sender: TObject);
procedure treJerarquiaClick(Sender: TObject);
procedure PanTituloEnter(Sender: TObject);
procedure PanTituloExit(Sender: TObject);
procedure PanParrafoExit(Sender: TObject);
procedure PanParrafoEnter(Sender: TObject);
procedure PanObjetoMultimediaEnter(Sender: TObject);
procedure PanObjetoMultimediaExit(Sender: TObject);
procedure PanJerarquiaEnter(Sender: TObject);
procedure PanJerarquiaExit(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure FormPaint(Sender: TObject);
procedure txtTituloKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure txtSubtituloKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure txtParrafoKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure txtObjetoMultimediaKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure cmbTipoObjetoClick(Sender: TObject);
procedure txtTituloChange(Sender: TObject);
procedure txtSubtituloChange(Sender: TObject);
procedure txtParrafoChange(Sender: TObject);
procedure txtObjetoMultimediaChange(Sender: TObject);
procedure cmdRelacionarClick(Sender: TObject);
private
  Encabezado: TEncabezado;
  ArticuloUsuario: TArticuloUsuario;

pSolicitud, pSolTituloAnt, pSolTituloAct, pSolParrafoAnt, pSolParrafoAct, pSolObjetoMultimediaAnt, p
SolObjetoMultimediaAct, pSolSubParrafoAnt, pSolSubParrafoAct, pAuxSolicitud, PauxTitulo, PauxPar
rafo, PauxSubParrafo, PauxObjetoMultimedia, Paux: PTSolicitud;
  NodoSeleccionado: TTreeNode;
  FArticuloGuardado: boolean;
  RelacionarArticulo: TfrmRelacionarArticulo;
  procedure EditarTitulos(Identificador: Longint);

```

```

procedure EditarParrafos(Identificador:Longint);
procedure EditarObjetosMultimedia(Identificador:Longint);
procedure EditarSubParrafos(Identificador:Longint;ParrajoPadre:PTSolicitud);
procedure LimpiarObjetos(Espacio:TScrollBox);
public
  procedure Editar(Identificador:Longint);
end;

```

UArticulo.pas

Funcionalidad: Unidad estructural donde se encapsulan las diferentes clases que forman parte de un artículo Gaia: plantilla de agregado de datos, plantilla de muestreo de datos insertados, plantilla de modelado y la plantilla de imagen.

Instancia de clases:

```

type TArticulo = class(TsuiTabSheet)
  FPlantillaInsercionDatos:TfrmAgregarArticulo;
  FPlantillaMostrarDatos:TfrmMostrarArticulo;
  FPlantillaModelado:TfrmLaboratorio;
  FPlantillaImágenes:TImage;
private
  FIdentificador:Longint;
protected
public
  procedure Agregar(Vineta:TsuiTabSheet);
  procedure Mostrar(Identificador:Longint;Vineta:TsuiTabSheet);
  procedure Eliminar(Identificador:Longint);
  procedure Editar(Identificador:Longint;Vineta:TsuiTabSheet);
  procedure BuscarEnArticulo;
  procedure Modelar(Vineta:TsuiTabSheet);
  procedure MostrarImagen(Imagen:TPicture;Vineta:TsuiTabSheet);
published
  property Identificador:Longint read FIdentificador write FIdentificador;
  property PlantillaModelado:TfrmLaboratorio read FPlantillaModelado;
end;

```

UBuscarObjeto.pas

Funcionalidad: Formulario interfaz de usuario, encargado de capturar el objeto multimedia candidato a ser guardado en la base de datos por parte del usuario.

Instancia de clases:

```
type
TfrmBuscarObjeto = class(TForm)
    suiBuscarObjeto: TsuiForm;
    AbrirObjeto: TOpenDialog;
    suiPanel1: TsuiPanel;
    cmdCargarObjeto: TsuiButton;
    cmdCancelar: TsuiButton;
    Espacio: TScrollBox;
    ObjetoImagen: TsuiImagePanel;
    cmdAjustar: TsuiButton;
    cmdEliminar: TsuiButton;
    ObjetoModelo: TEditorEVL;
    ObjetoSonido: TMediaPlayer;
    ObjetoOtroArchivo: TOLEContainer;
    procedure cmdCargarObjetoClick(Sender: TObject);
    procedure cmdAjustarClick(Sender: TObject);
    procedure cmdEliminarClick(Sender: TObject);
    procedure cmdCancelarClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
private
    FTipoObjeto,FArchivoAnterior,FArchivoActual:string;
    procedure CopiarArchivo(ArchivoOrigen,ArchivoDestino:string);
    { Private declarations }
public
    function ExtensionArchivo:string;
    procedure CargarEditor(Archivo:string);
    { Public declarations }
published
    property AbrirTipoObjeto:string write FTipoObjeto;
    property ArchivoAnterior:string write FArchivoAnterior;
    property ArchivoActual:string read FArchivoActual;
end;
```

UBusquedaAvanzada.pas

Funcionalidad: Formulario interfaz de usuario, encargado de capturar los parámetros de búsqueda, seleccionados por parte del usuario.

Instancia de clases:

```
type
TfrmBusquedaAvanzada = class(TForm)
    suifrmBusquedaAvanzada: TsuiForm;
```

```

cmdAceptar: TsuiButton;
cmdCancelar: TsuiButton;
suiChkAtributosBusqueda: TsuiCheckGroup;
suiCmbTipoArticulo: TsuiComboBox;
suiCmbObjetosMultimedia: TsuiComboBox;
suiChkAbstract: TsuiCheckBox;
suiChkTipoArticulo: TsuiCheckBox;
suiChkOrigen: TsuiCheckBox;
suiChkAutor: TsuiCheckBox;
suiChkCoautor: TsuiCheckBox;
suiChkEdicion: TsuiCheckBox;
suiChkEditorial: TsuiCheckBox;
suiChkLugarPublicacion: TsuiCheckBox;
suiChkFechaPublicacion: TsuiCheckBox;
suiChkVolumen: TsuiCheckBox;
suiChkFechaAccesoPagina: TsuiCheckBox;
suiChkTitulos: TsuiCheckBox;
suiChkParrafos: TsuiCheckBox;
suiChkObjetosMultimedia: TsuiCheckBox;
suiChkPalabrasClaves: TsuiCheckBox;
cmdRestablecer: TsuiButton;
procedure suiChkTipoArticuloClick(Sender: TObject);
procedure suiChkObjetosMultimediaClick(Sender: TObject);
procedure cmdCancelarClick(Sender: TObject);
procedure cmdAceptarClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure cmdRestablecerClick(Sender: TObject);
private
    FAMostrar, FAComparar, FCompararCon, FParametros, FOperadores: TStringList;
    FListaArticulos: TConsulta;

FBuscarGuiasEstudio, FBuscarImágenes, FBuscarModelos, FBuscarOtrosArchivos, FBuscarSonidos, F
BuscarVideos: TNotifyEvent;
    { Private declarations }
public
    procedure Restablecer;
    procedure ProcedimientoBuscarGuiasEstudio(Sender: TObject);
    procedure ProcedimientoBuscarImágenes(Sender: TObject);
    procedure ProcedimientoBuscarModelos(Sender: TObject);
    procedure ProcedimientoBuscarOtrosArchivos(Sender: TObject);
    procedure ProcedimientoBuscarSonidos(Sender: TObject);

```

```

    procedure ProcedimientoBuscarVideos(Sender:TObject);
published
    property ListaArticulos:TConsulta read FListaArticulos;
    property BuscarGuiasEstudio:TNotifyEvent read FBuscarGuiasEstudio;
    property BuscarImagenes:TNotifyEvent read FBuscarImagenes;
    property BuscarModelos:TNotifyEvent read FBuscarModelos;
    property BuscarOtrosArchivos:TNotifyEvent read FBuscarOtrosArchivos;
    property BuscarSonidos:TNotifyEvent read FBuscarSonidos;
    property BuscarVideos:TNotifyEvent read FBuscarVideos;
    { Public declarations }
end;

```

UControles.pas

Funcionalidad: Unidad donde se implementan los controles de usa Gaia para mostrar la información de sus artículos.

Instancia de clases:

```

{Clase TEditGaia}
type
    TEditGaia= class(TRichEdit)
    private
        FTexto:TLabel;
    protected
    public
        constructor Create(AOwner:TComponent); override;
        procedure Mostrar(FPantalla:TScrollBox;Contenido:string;PosX,PosY:integer);
    published
end;

```

```

{Clase TMemogaia}
TMemogaia= class(TRichEdit)
    private
        FTexto:TLabel;
    protected
    public
        constructor Create(AOwner:TComponent); override;
        procedure Mostrar(FPantalla:TScrollBox;Contenido:string;PosX,PosY:integer);
    published
end;

```

```

{Clase TImageGaia}
type

```

```

TImageGaia = class(TImage)
private
    FDescripcion:string;
    FImagen:TArticulo;
    FArchivoImagen:string;
    procedure ClickImagenGaia(Sender:TObject); overload;
protected
public
    constructor Create(AOwner:TComponent); override;
    destructor Destroy; override;
    procedure
Mostrar(FPantalla:TScrollBox;ArchivoImagen,Descripcion:string;PosX,PosY:integer);
    procedure
MostrarComoAccesorio(FPantalla:TScrollBox;ArchivoImagen,Descripcion:string;PosX,PosY:integer);
    published
end;

{Clase TEditorEVLGaia}
type
TEditorEVLGaia = class(TEditorEVL)
private
    FDescripcion:string;
    FLaboratorio:TArticulo;
    FIdentificador:Longint;
    FImagen:TImage;
    procedure CargarEditor(Archivo:string);
    procedure ClickImagenEditorEVLGaia(Sender:TObject); overload;
protected
public
    constructor Create(AOwner:TComponent); override;
    destructor Destroy; override;
    procedure Mostrar(FPantalla:TScrollBox;ArchivoModelo:string;PosX,PosY:integer);
    procedure
MostrarComoAccesorio(FPantalla:TScrollBox;ArchivoImagen,Descripcion:string;PosX,PosY:integer;Identificador:Longint);
    published
    property Accesorio:TImage read FImagen;
end;

{Clase TMediaPlayerGaia}
type

```

```

TMediaPlayerGaia = class(TMediaPlayer)
private
    FDescripcion:string;
    FImagen:TImage;
    procedure ClickImagenMediaPlayerGaia(Sender:TObject); overload;
protected
public
    constructor Create(AOwner:TComponent); override;
    destructor Destroy; override;
    procedure Mostrar(FPantalla:TScrollBox;Archivo,Descripcion:string;PosX,PosY:integer);
    procedure
MostrarComoAccesorio(FPantalla:TScrollBox;Archivo,Descripcion:string;PosX,PosY:integer);
published
    property Accesorio:TImage read FImagen;
end;

{Clase TOleContainerGaia}
type
TOleContainerGaia = class(TOleContainer)
private
    FArchivo,FDescripcion:string;
    FImagen:TImage;
    procedure ClickImagenOleContainerGaia(Sender:TObject); overload;
protected
public
    constructor Create(AOwner:TComponent); override;
    destructor Destroy; override;
    procedure Mostrar(FPantalla:TScrollBox;Archivo,Descripcion:string;PosX,PosY:integer);
    procedure
MostrarComoAccesorio(FPantalla:TScrollBox;Archivo,Descripcion:string;PosX,PosY:integer);
published
    property Accesorio:TImage read FImagen;
end;

{Clase TFormTemasRelacionadosGaia}
type
TFormTemasRelacionadosGaia = class(TfrmTemasRelacionados)
private
    FDescripcion:string;
protected
public
    constructor Create(AOwner:TComponent); override;

```

```

destructor Destroy; override;
procedure
MostrarComoAccesorio(FPantalla:TScrollBox;ListaArticulos:TConsulta;Descripcion:string;PosX,PosY:integer;Identificador:Longint);
published
end;

```

UDatosDD.pas

Funcionalidad: Unidad encargada exclusivamente en la importación y exportación de los artículos de Gaia.

Instancia de clases:

```

type
TDatosDD = class(TObject)
private
FIdentificador:Longint;
FNombreArchivo,FNombreArchivoCargar:string;
FDialogoSalvar:TSaveDialog;
FDialogoCargar:TOpenDialog;
procedure CrearArchivosXML;
procedure GenerarArchivoGaia;
procedure ExtraerArchivosXML;
procedure SubirArchivosXMLBaseDatos;
protected
public
constructor Create;
destructor Destroy; override;
procedure Guardar(Identificador:Longint);
procedure Cargar;
published
end;

```

UDibujar.pas

Funcionalidad: Unidad que desempeña la función de crear y colocar los controles de Gaia ordenadamente, los cuales mostrarán la información de los artículos.

Instancia de clases:

```

{Clase TDibujoEncabezado}
type
TDibujoEncabezado = class(TObject)

```

```

private
  FConsulta:TConsulta;
  FNombreCampo,FInformacion:TEditGaia;
  FAbstract, FPalabrasClaves:TMemoGaia;
  FEspacioInterlineal:integer;
protected
public
  constructor Create;
  destructor Destroy;
  procedure Dibujar(IdEncabezado:Longint;Destino:TScrollBox);
published
  property EspacioInterlineal:integer read FEspacioInterlineal write FEspacioInterlineal;
end;

{Clase TDibujoContenido}
type
  TDibujoContenido = class(TObject)
  private
    FConsultaTitulos, FConsultaParrafos, FConsultaObjetosMultimedia, FConsultaEstructura,
    FConsultaTemasRelacionados:TConsulta;
    FTitulo, FSubTitulo:TEditGaia;
    FParrafo:TMemoGaia;
    FDescripcion:TEditGaia;
    FImagen, FImagenAcc:TImageGaia;
    FModelo:TEditorEVLGaia;
    FSonido:TMediaPlayerGaia;
    FOtroArchivo:TOLEContainerGaia;
    FTemasRelacionados:TFormTemasRelacionadosGaia;
    FEspacioInterlineal:integer;
    procedure ObtenerNodoPorTexto(Arbol:TsuiTreeView;Texto:String;var
    NodoSubtitulo,NodoParrafo:TTreeNode);
  protected
  public
    constructor Create;
    destructor Destroy;
    procedure Dibujar(IdEncabezado:Longint;Destino,Accesorio:TScrollBox;Arbol:TsuiTreeView);
  published
    property EspacioInterlineal:integer read FEspacioInterlineal write FEspacioInterlineal;
  end;

```

UFiltrarArticulosDisponibles.pas

Funcionalidad: Formulario interfaz de usuario encargada de hacer el respectivo filtro sobre los artículos que se encuentran disponibles a relacionar con un artículo candidato.

Instancia de clases:

type

```
TfrmFiltrarArticulosDisponibles = class(TForm)
  suiFiltrarArticulosDisponibles: TsuiForm;
  suiCmdAceptar: TsuiButton;
  suiCmdCancelar: TsuiButton;
  suiChkAtributosBusqueda: TsuiCheckGroup;
  suiCmbTipoArticulo: TsuiComboBox;
  suiCmbObjetosMultimedia: TsuiComboBox;
  suiChkAbstract: TsuiCheckBox;
  suiChkTipoArticulo: TsuiCheckBox;
  suiChkOrigen: TsuiCheckBox;
  suiChkAutor: TsuiCheckBox;
  suiChkCoautor: TsuiCheckBox;
  suiChkEdicion: TsuiCheckBox;
  suiChkEditorial: TsuiCheckBox;
  suiChkLugarPublicacion: TsuiCheckBox;
  suiChkFechaPublicacion: TsuiCheckBox;
  suiChkVolumen: TsuiCheckBox;
  suiChkFechaAccesoPagina: TsuiCheckBox;
  suiChkTitulos: TsuiCheckBox;
  suiChkParrafos: TsuiCheckBox;
  suiChkObjetosMultimedia: TsuiCheckBox;
  suiChkPalabrasClaves: TsuiCheckBox;
  cmdRestablecer: TsuiButton;
  txtExpresion: TsuiEdit;
  suiEdit1: TsuiEdit;
  procedure suiChkTipoArticuloClick(Sender: TObject);
  procedure suiChkObjetosMultimediaClick(Sender: TObject);
  procedure suiCmdCancelarClick(Sender: TObject);
  procedure suiCmdAceptarClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
  procedure cmdRestablecerClick(Sender: TObject);
  procedure suiChkAbstractClick(Sender: TObject);
  procedure suiChkOrigenClick(Sender: TObject);
  procedure suiChkAutorClick(Sender: TObject);
  procedure suiChkCoautorClick(Sender: TObject);
```

```

procedure suiChkEdicionClick(Sender: TObject);
procedure suiChkEditorialClick(Sender: TObject);
procedure suiChkLugarPublicacionClick(Sender: TObject);
procedure suiChkFechaPublicacionClick(Sender: TObject);
procedure suiChkVolumenClick(Sender: TObject);
procedure suiChkFechaAccesoPaginaClick(Sender: TObject);
procedure suiChkPalabrasClavesClick(Sender: TObject);
procedure suiChkTitulosClick(Sender: TObject);
procedure suiChkParrafosClick(Sender: TObject);
procedure FormPaint(Sender: TObject);
private
  FAMostrar, FAComparar, FCompararCon, FParametros, FOperadores:TStringList;
  FListaArticulos:TConsulta;
  FIdentificador:Longint;
  { Private declarations }
public
  procedure Restablecer;
published
  property FiltroArticulosDisponibles:TConsulta read FListaArticulos;
  property Identificador:Longint write FIdentificador;
  { Public declarations }
end;

```

UGlosario.pas

Funcionalidad: Formulario interfaz de usuario que desempeña la función de comunicar el glosario con su respectivo usuario, para agregar, editar, borrar y relacionar conceptos.

Instancia de clases:

```

type
  TfrmGlosario = class(TForm)
    suiGlosario: TsuiForm;
    suiEdit1: TsuiEdit;
    cmbParametro: TsuiComboBox;
    IstConceptos: TsuiDBLookupListBox;
    suiScrollBar1: TsuiScrollBar;
    suiEdit2: TsuiEdit;
    suiEdit3: TsuiEdit;
    suiScrollBar3: TsuiScrollBar;
    cmdAgregarSalvar: TsuiButton;
    cmdEliminar: TsuiButton;
    cmdRelacionar: TsuiButton;

```

```

suiScrollBar2: TsuiScrollBar;
lstArticulosRelacionados: TsuiDBLookupListBox;
memDefinicion: TsuiDBMemo;
cmdCerrarCancelar: TsuiButton;
cmdEditar: TsuiButton;
procedure FormPaint(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure lstConceptosClick(Sender: TObject);
procedure cmdAgregarSalvarClick(Sender: TObject);
procedure cmdEliminarClick(Sender: TObject);
procedure cmdCerrarCancelarClick(Sender: TObject);
procedure cmdRelacionarClick(Sender: TObject);
procedure cmbParametroKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure lstArticulosRelacionadosDbClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure cmdEditarClick(Sender: TObject);
private
  { Private declarations }
  FMostrar,FComparar,FOperadores,FParametros,FEnlace:TStringList;
  FConceptos,FArticulosRelacionados:TConsulta;
  FGlosario:TGlosario;
  FUsuarioGlosario:TUsuarioGlosario;
  FRelacionarGlosario:TfrmRelacionarGlosario;
  procedure ConsultarConceptos;
  procedure ConsultarArticulosRelacionados;
public
  { Public declarations }
end;

```

UListaSolicitud.pas

Funcionalidad: Unidad donde se implementan las respectivas listas de títulos, párrafos y objetos multimedia.

Instancia de clases:

```

type
  {Clase TListaSolicitud}
  TListaSolicitud = class(TObject)
  private
  protected

```

```

FListaSolicitud:TList;
FNumeroSolicitud:Integer;
procedure EliminarEnCascada(Posicion:integer); virtual; abstract;
public
constructor Create;
procedure Agregar; virtual; abstract;
procedure Eliminar(Posicion:Integer); virtual; abstract;
function Seleccionar(Posicion:Integer):Pointer;
function CantidadSolicitudes:Integer;
procedure MostrarLista(ContenedorLista:TsuilistBox);
destructor Destroy; override;
end;

```

```

type
{Clase TListaTitulo}
TListaTitulo = class(TListaSolicitud)
private
protected
public
procedure Agregar; override;
procedure Eliminar(Posicion:Integer); override;
procedure EliminarEnCascada(Posicion:integer); override;
destructor Destroy; override;
end;

```

```

type
{Clase TListaParrafo}
TListaParrafo = class(TListaSolicitud)
private
protected
procedure EliminarEnCascada(Posicion:integer); override;
public
procedure Agregar; override;
procedure Eliminar(Posicion:Integer); override;
destructor Destroy; override;
published
end;

```

```

type
{Clase TListaObjetoMultimedia}
TListaObjetoMultimedia = class(TListaSolicitud)

```

```

private
protected
    procedure EliminarEnCascada(Posicion:integer); override;
public
    procedure Agregar; override;
    procedure Eliminar(Posicion:Integer); override;
    destructor Destroy; override;
published
end;

```

UManejadorError.pas

Funcionalidad: Unidad donde se describen los errores o posibles excepciones generadas por parte del usuario.

Instancia de clases:

```

type
    TManejadorError = class(TObject)
    public
        procedure MensajeError(mensaje:string);
    end;

```

UMenuEnPantalla.pas

Funcionalidad: Unidad donde se implementa la clase que representan los Centros de Navegación de Gaia.

Instancia de clases:

```

type
    TMenu = class(TsuiTabSheet)
        FPantalla:TfrmMenu;
    private
    protected
    public
        procedure AgregarItem(Item,Leyenda:string;Comando:TNotifyEvent);
        procedure Mostrar(Vineta:TsuiTabSheet);
    published
    end;

```

UMostrarArticulo.pas

Funcionalidad: Unidad interfaz de usuario, en otras palabras, cascarón contenedora de los controles Gaia que serán los directos responsables de mostrar la información de los respectivos artículos.

Instancia de clases:

```
type
TfrmMostrarArticulo = class(TForm)
  suiMostrarArticulo: TsuiForm;
  panEsquema: TsuiPanel;
  panArticulo: TsuiPanel;
  panAccesorios: TsuiPanel;
  scrEspacio: TScrollBox;
  panDescripcion: TsuiPanel;
  scrDescripcion: TScrollBox;
  treJerarquia: TsuiTreeView;
  Buscar: TFindDialog;
  BuscarExpresion: TsuiInputDialog;
  scrAccesorios: TScrollBox;
  ImageList1: TImageList;
  procedure treJerarquiaClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
  procedure FormPaint(Sender: TObject);
  procedure BuscarFind(Sender: TObject);
private
  p: ^TEdit;
  Informacion:TConsulta;
  FIdentificador:Longint;
  FTituloPrincipal:string;
  procedure DestruirControles(ScrollBox:TScrollBox);
  { Private declarations }
public
  procedure MostrarContenido(Identificador:Longint);
published
  property TituloPrincipal:string read FTituloPrincipal;
  { Public declarations }
end;
```

UOpcionesUsuario.pas

Funcionalidad: Unidad interfaz de usuario, que permite al usuario Administrador eliminar a otros usuarios y sus respectivos contenidos y a los demás usuarios cambiar su nombre usuario y/o contraseña.

Instancia de clases:

```
type
TfrmOpcionesUsuario = class(TForm)
  suiOpcionesUsuario: TsuiForm;
  suiEdit1: TsuiEdit;
  suiEdit2: TsuiEdit;
  txtNombreUsuario: TsuiEdit;
  txtContraseña: TsuiEdit;
  cmdCambiar: TsuiButton;
  cmdAgregar: TsuiButton;
  cmdEliminar: TsuiButton;
  procedure FormCreate(Sender: TObject);
  procedure cmdCambiarClick(Sender: TObject);
  procedure cmdAgregarClick(Sender: TObject);
  procedure cmdEliminarClick(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure FormPaint(Sender: TObject);
private
  { Private declarations }
  FUsuario:TUsuario;
public
  { Public declarations }
end;
```

UPantallaMenu.pas

Funcionalidad: Unidad plantilla y parte de la interfaz de usuario, que forma parte visual de los centros de navegación de Gaia. Esta plantilla contendrá los ítems y fondo de un Centro de Navegación.

Instancia de clases:

```
type
TMenuItem = class(TLabel)
private
  FBarra,FSeparador:TShape;
  FLeyenda:TLabel;
  procedure MostrarBarra(Sender:TObject);
```

```

    procedure OcultarBarra(Sender:TObject);
protected
public
    constructor Create(AOwner:TComponent); override;
    destructor Destroy; override;
    procedure Mostrar;
published
end;

type
TfrmMenu = class(TForm)
    suiMenu: TsuiForm;
    ImagenFondo: TImage;
    lblTitulo: TLabel;
    lblLeyenda: TLabel;
    scrMenu: TScrollBar;
private
    FItemMenu:TMenuItem;
    { Private declarations }
public
    { Public declarations }
    procedure
AgregarItem(Item,Leyenda:string;Comando:TNotifyEvent;PantallaMenu:TfrmMenu);
end;

```

UPrincipal.pas

Funcionalidad: Formulario principal de Gaia.

Instancia de clases:

```

type
TfrmPrincipal = class(TForm)
    mnuArchivo: TMenuItem;
    mnuImprimir: TMenuItem;
    N3: TMenuItem;
    mnuSalir: TMenuItem;
    mnuEdicion: TMenuItem;
    mnuCopiar: TMenuItem;
    N4: TMenuItem;
    mnuBuscarGaia: TMenuItem;
    N5: TMenuItem;
    mnuBuscarPagina: TMenuItem;
    mnuVer: TMenuItem;

```

mnuIr: TMenuItem;
mnuAtras: TMenuItem;
mnuAdelante: TMenuItem;
N6: TMenuItem;
mnuFavoritos: TMenuItem;
mnuAgregarFavoritos: TMenuItem;
mnuOrganizarFavoritos: TMenuItem;
mnuComponentes: TMenuItem;
mnuContenido: TMenuItem;
mnuCentroContenidos: TMenuItem;
N14: TMenuItem;
mnuLaboratorio: TMenuItem;
mnuAgregar: TMenuItem;
mnuBorrarArticuloActual: TMenuItem;
mnuRestaurarEstadoInicial: TMenuItem;
mnuTutorialDinamicaSistemas: TMenuItem;
N20: TMenuItem;
mnuRecursosEducativos: TMenuItem;
mnuCentroRecursosEducativos: TMenuItem;
N16: TMenuItem;
mnuGlosario: TMenuItem;
mnuGuiaEstudio: TMenuItem;
mnuMultimedia: TMenuItem;
mnuImágenes: TMenuItem;
mnuSonidos: TMenuItem;
mnuVideos: TMenuItem;
mnuAnimaciones: TMenuItem;
mnuJuegos: TMenuItem;
N21: TMenuItem;
mnuRed: TMenuItem;
mnuHerramientas: TMenuItem;
mnuActualizar: TMenuItem;
N8: TMenuItem;
mnuCopiarDiscoDuro: TMenuItem;
N10: TMenuItem;
mnuOpciones: TMenuItem;
N2: TMenuItem;
mnuAyudaContenido: TMenuItem;
N12: TMenuItem;
mnuAyudaAcercade: TMenuItem;
SUIMenuPrincipal: TsuiMainMenu;
SUIPrincipal: TsuiForm;

N11: TMenuItem;
mnuEntorno: TMenuItem;
mnuPlateadoligero: TMenuItem;
mnuWindowsXP: TMenuItem;
mnuAzulprofundo: TMenuItem;
mnuProteina: TMenuItem;
mnuVidrioazul: TMenuItem;
mnuEditar: TMenuItem;
mnuCyanprofundo: TMenuItem;
mnuVerdeprofundo: TMenuItem;
mnuNaranjaprofundo: TMenuItem;
mnuVidrioverde: TMenuItem;
mnuVidriocrecido: TMenuItem;
mnuAzulunico: TMenuItem;
mnuCyanunico: TMenuItem;
mnuVerdeunico: TMenuItem;
mnuNaranjaunico: TMenuItem;
mnuVidrionaranja: TMenuItem;
mnuNegrodeportivo: TMenuItem;
mnuAzuldeportivo: TMenuItem;
mnuCyandeportivo: TMenuItem;
mnuVerdedeportivo: TMenuItem;
mnuNaranjadeportivo: TMenuItem;
SUITemaArchivo: TsuiFileTheme;
mnuCerrarHojaActual: TMenuItem;
SUIMensaje: TsuiMessageDialog;
ImageList1: TImageList;
suiToolBar1: TsuiToolBar;
cmdAgregarArticulo: TToolButton;
cmdBorrarArticulo: TToolButton;
cmdCerrarSesion: TToolButton;
cmdCerrarOtrasSesiones: TToolButton;
mnuCerrarOtrasHojas: TMenuItem;
cmdEditarArticulo: TToolButton;
ToolButton1: TToolButton;
ToolButton2: TToolButton;
suiSideChannel1: TsuiSideChannel;
Label1: TLabel;
cmdBusquedaAvanzada: TsuiButton;
IstArticulos: TsuiDBLookupListBox;
suiCmbParametro: TsuiComboBox;
Entorno: TsuiPageControl;

```
cmdRestablecer: TsuiButton;  
mnuCargarDiscoDuro: TMenuItem;  
ToolButton3: TToolButton;  
cmdCopiarDD: TToolButton;  
cmdCargarDD: TToolButton;  
N7: TMenuItem;  
mnuUSuarios: TMenuItem;  
mnuCerrarSesion: TMenuItem;  
ToolButton4: TToolButton;  
cmdCerrarSecion: TToolButton;  
cmdBuscarGaia: TToolButton;  
cmdBuscarPagina: TToolButton;  
ToolButton5: TToolButton;  
ToolButton6: TToolButton;  
ToolButton7: TToolButton;  
mnuModelos: TMenuItem;  
mnuTutorialDiagramasForrester: TMenuItem;  
N9: TMenuItem;  
N13: TMenuItem;  
mnuCentromultimedia: TMenuItem;  
ToolButton8: TToolButton;  
ToolButton9: TToolButton;  
cmdGlosario: TToolButton;  
ToolButton10: TToolButton;  
ToolButton11: TToolButton;  
ToolButton12: TToolButton;  
ToolButton13: TToolButton;  
procedure mnuAgregaClick(Sender: TObject);  
procedure mnuPlateadoligeroClick(Sender: TObject);  
procedure mnuWindowsXPClick(Sender: TObject);  
procedure mnuProteinaClick(Sender: TObject);  
procedure mnuVidrioazulClick(Sender: TObject);  
procedure mnuAzulprofundoClick(Sender: TObject);  
procedure mnuCyanprofundoClick(Sender: TObject);  
procedure mnuVerdeprofundoClick(Sender: TObject);  
procedure mnuNaranjaProfundoClick(Sender: TObject);  
procedure mnuVidrioverdeClick(Sender: TObject);  
procedure mnuVidriorecidoClick(Sender: TObject);  
procedure mnuAzulunicoClick(Sender: TObject);  
procedure mnuCyanunicoClick(Sender: TObject);  
procedure mnuVerdeunicoClick(Sender: TObject);  
procedure mnuNaranjaunicoClick(Sender: TObject);
```

```

procedure mnuVidrionaranjaClick(Sender: TObject);
procedure mnuNegrodeportivoClick(Sender: TObject);
procedure mnuAzuldeportivoClick(Sender: TObject);
procedure mnuCyandeportivoClick(Sender: TObject);
procedure mnuVerdedeportivoClick(Sender: TObject);
procedure mnuNaranjadeportivoClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure mnuCerrarHojaActualClick(Sender: TObject);
procedure lstArticulosDbClick(Sender: TObject);
procedure mnuBorrarArticuloActualClick(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure mnuEditarClick(Sender: TObject);
procedure cmdBusquedaAvanzadaClick(Sender: TObject);
procedure suiCmbParametroKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure mnuCerrarOtrasHojasClick(Sender: TObject);
procedure mnuSalirClick(Sender: TObject);
procedure mnuBuscarPaginaClick(Sender: TObject);
procedure mnuLaboratorioClick(Sender: TObject);
procedure cmdRestablecerClick(Sender: TObject);
procedure mnuCopiarDiscoDuroClick(Sender: TObject);
procedure mnuCargarDiscoDuroClick(Sender: TObject);
procedure mnuUSuariosClick(Sender: TObject);
procedure mnuCerrarSesionClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
procedure mnuBuscarGaiaClick(Sender: TObject);
procedure mnuCentroContenidosClick(Sender: TObject);
procedure mnuCentromultimediaClick(Sender: TObject);
procedure mnuCentroRecursosEducativosClick(Sender: TObject);
procedure cmdEditarArticuloClick(Sender: TObject);
procedure mnuAtrasClick(Sender: TObject);
procedure mnuAdelanteClick(Sender: TObject);
procedure mnuGlosarioClick(Sender: TObject);
procedure cmdBuscarGaiaClick(Sender: TObject);
procedure mnuAyudaAcercadeClick(Sender: TObject);
private
  { Private declarations }
  FArticulo:TArticulo;
  FBusquedaAvanzada:TfrmBusquedaAvanzada;
  ManError: TManejadorError;
  FEnEdicion:boolean;

```

```

FDatosDD:TDatosDD;
FUsuarios:TfrmOpcionesUsuario;
FMenu:TMenu;
FGlosario:TfrmGlosario;
procedure PasarTemaArticulos(Tema:TsuiUIStyle;TemaArchivo:TsuiFileTheme);
function CargarCentros(TituloCentro:string):boolean;
public
  procedure CargarArticuloDesdeLista(Lista:TsuiDBLookupListBox);
published
  { Public declarations }
  property EnEdicion:boolean read FEnEdicion;
end;

```

URelacionArticulo.pas

Funcionalidad: Formulario interfaz de usuario, que ofrece la posibilidad de relacionar y deshacer relaciones entre artículos Gaia.

Instancia de clases:

```

type
TfrmRelacionarArticulo = class(TForm)
  suiRelacionarArticulo: TsuiForm;
  suiEdit1: TsuiEdit;
  suiEdit2: TsuiEdit;
  IstArticulosDisponibles: TsuiDBLookupListBox;
  IstArticulosARelacionar: TsuiListView;
  suiEdit3: TsuiEdit;
  cmdAgregar: TsuiButton;
  cmdBorrar: TsuiButton;
  cmdRelacionar: TsuiButton;
  cmdCancelar: TsuiButton;
  cmdFiltrar: TsuiButton;
  cmdDeshacer: TsuiButton;
  IstArticulosRelacionados: TsuiDBLookupListBox;
  procedure FormPaint(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure cmdAgregarClick(Sender: TObject);
  procedure cmdRelacionarClick(Sender: TObject);
  procedure cmdCancelarClick(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure FormDestroy(Sender: TObject);
  procedure cmdDeshacerClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);

```

```

procedure cmdBorrarClick(Sender: TObject);
procedure cmdFiltrarClick(Sender: TObject);
procedure IstArticulosRelacionadosDbClick(Sender: TObject);
private
  { Private declarations }
  FMostrar,FComparar,FOperadores,FParametros,FEnlace:TStringList;
  FArticulosRelacionados,FArticulosDisponibles,FBorrarRelacion:TConsulta;
  FRelacionar:TTemasRelacionados;
  FIdentificador:Longint;
  FFiltrar:TfrmFiltrarArticulosDisponibles;
  procedure ConsultarArticulosRelacionados;
  procedure ConsultarArticulosDisponibles;
public
  { Public declarations }
published
  property Identificador:Longint write FIdentificador;
  property ArticulosDisponibles:TConsulta read FArticulosDisponibles;
end;

```

URelacionGlosario.pas

Funcionalidad: Formulario interfaz de usuario, que ofrece la posibilidad de relacionar y deshacer relaciones entre artículos Gaia y conceptos Gaia.

Instancia de clases:

```

type
  TfrmRelacionarGlosario = class(TForm)
    suiRelacionarGlosario: TsuiForm;
    suiEdit1: TsuiEdit;
    suiEdit2: TsuiEdit;
    IstArticulosDisponibles: TsuiDBLookupListBox;
    IstArticulosARelacionar: TsuiListView;
    suiEdit3: TsuiEdit;
    cmdAgregar: TsuiButton;
    cmdBorrar: TsuiButton;
    cmdRelacionar: TsuiButton;
    cmdCancelar: TsuiButton;
    cmdFiltrar: TsuiButton;
    cmdDeshacer: TsuiButton;
    IstArticulosRelacionados: TsuiDBLookupListBox;
    txtConcepto: TsuiEdit;
    procedure FormPaint(Sender: TObject);
  end;

```

```

procedure FormShow(Sender: TObject);
procedure cmdAgregarClick(Sender: TObject);
procedure cmdRelacionarClick(Sender: TObject);
procedure cmdCancelarClick(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormDestroy(Sender: TObject);
procedure cmdDeshacerClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure cmdBorrarClick(Sender: TObject);
procedure cmdFiltrarClick(Sender: TObject);
procedure lstArticulosRelacionadosDbClick(Sender: TObject);
private
  { Private declarations }
  FMostrar,FComparar,FOperadores,FParametros,FEnlace:TStringList;
  FArticulosRelacionados,FArticulosDisponibles,FBorrarRelacion:TConsulta;
  FRelacionar:TArticuloGlosario;
  FIdentificador:Longint;
  FFiltrar:TfrmFiltrarArticulosDisponibles;
  procedure ConsultarArticulosRelacionados;
  procedure ConsultarArticulosDisponibles;
public
  { Public declarations }
published
  property Identificador:Longint write FIdentificador;
  property ArticulosDisponibles:TConsulta read FArticulosDisponibles;
end;

```

USolicitud.pas

Funcionalidad: Unidad muy importante de Gaia, donde se implementa las diferentes Solicitudes a la Base de Datos por parte del usuario, por ejemplo: agregar títulos, agregar párrafos, eliminar títulos, agregar artículo, etc. Además se instancia la función recursiva Recorrer en carga de recorrer todos los párrafos de párrafos que hacen parte de un artículos.

Instancia de clases:

```

type
  {Clase TSolicitud}
  TSolicitud = class(TObject)
  private

```

```

protected
    FConexionBD:TADOConnection;
    FOrigenDatos:TADODataset;
    FEnlace: TDataSource;
    FIdentificador:Longint;
    FDescripcion:string;
    FEstado:string;
    FTabla:string;
    FTituloPrincipal:string;
public
    procedure Conectarse(BaseDatos,Tabla:string);
    procedure Agregar;
    procedure Guardar; virtual; abstract;
    procedure Eliminar; virtual;
    procedure Cancelar;
    procedure Cerrar;
    function Enlazar:TDataSource;
    procedure Editar(Identificador:Longint); virtual; abstract;
published
    property Identificador: Longint read FIdentificador;
    property Descripcion: string read FDescripcion write FDescripcion;
    property Estado:string read FEstado write FEstado;
    property PasarDatos:TADODataset read FOrigenDatos write FOrigenDatos;
end;

type
    {Clase TEncabezado}
    TEncabezado = class(TSolicitud)
    private
    protected
        FUsuario:Longint;
        FTitulo,FTipoArticulo:string;
        FListaTitulo:TListaTitulo;
        FNumeroArticulos:Longint;
        function GetNumeroArticulos:Longint;
    public
        constructor Create;
        destructor Destroy; override;
        procedure Guardar; override;
        procedure Eliminar; override;
        procedure MostrarJerarquia(Arbol:TsuiTreeView);
        procedure IniciarPlantilla;

```

```

    procedure Editar(Identificador:Longint); override;
published
    property Usuario:Longint write FUsuario;
    property Titulo:string read FTitulo;
    property TipoArticulo:string read FTipoArticulo;
    property ListaTitulo:TListaTitulo read FListaTitulo;
    property NumeroArticulos:Longint read GetNumeroArticulos;
end;

type
    {Clase TTitulo}
    TTitulo = class(TSolicitud)
    private
    protected
        FEncabezado:Longint;
        FDescripcion:string;
        FListaParrafo:TListaParrafo;
    public
        constructor Create;
        destructor Destroy; override;
        procedure Guardar; override;
        procedure Editar(Identificador:Longint); override;
    published
        property Encabezado:Longint read FEncabezado write FEncabezado;
        property Descripcion:string read FDescripcion write FDescripcion;
        property ListaParrafo:TListaParrafo read FListaParrafo;
    end;

type
    {Clase TParrafo}
    TParrafo = class(TSolicitud)
    private
    protected
        FTitulo:Longint;
        FSubTitulo:string;
        FDescripcion:TMemo;
        FListaObjetoMultimedia:TListaObjetoMultimedia;
        FListaSubParrafo:TListaParrafo;
    public
        constructor Create;
        destructor Destroy; override;
        procedure Guardar; override;

```

```

    procedure Editar(Identificador:Longint); override;
published
    property Titulo:Longint write FTitulo;
    property SubTitulo:string read FSubTitulo write FSubTitulo;
    property Descripcion:TMemo read FDescripcion write FDescripcion;
    property ListaObjetoMultimedia:TListaObjetoMultimedia read FListaObjetoMultimedia;
    property ListaSubParrafo:TListaParrafo read FListaSubParrafo;
end;

type
    {Clase TObjetoMultimedia}
    TObjetoMultimedia = class(TSolicitud)
    private
        FParrafo:Longint;
        FDescripcion:string;
        FTipo:string;
        FBuscarObjeto:TfrmBuscarObjeto;
        FImagen:TImage;
        FModelo:TEditorEVL;
        FSonido:TMediaPlayer;
        FOtroArchivo:TOleContainer;
        FPathArchivo:string;
        FExtension:string;
        function GetEntornoCargarObjeto:TfrmBuscarObjeto;
        procedure GuardarModelo(Archivo:string);
    protected
    public
        constructor Create;
        destructor Destroy; override;
        procedure Guardar; override;
        procedure GuardarLaboratorio;
        procedure GuardarDirecto;
        procedure Editar(Identificador:Longint); override;
        procedure Cargar;
        procedure Pintar(Destino:TScrollBox);
    published
        property Parrafo:Longint write FParrafo;
        property Descripcion:string read FDescripcion write FDescripcion;
        property Tipo:string read FTipo write FTipo;
        property EntornoCargarObjeto:TfrmBuscarObjeto read GetEntornoCargarObjeto;
        property ObjetoImagen:TImage read FImagen;
        property ObjetoModelo:TEditorEVL read FModelo;

```

```

    property ExtensionArchivo:string read FExtension write FExtension;
end;

type
  {Clase TConsulta}
  TConsulta = class(TSolicitud)
  private
    FCadenaBusqueda: string;
    FPantallaEncabezado,FPantallaContenido:TScrollBox;
    function  ObtenerCadena(CamposAMostrar, CamposAComparar, OperadoresComparacion
,Parametros, OperadoresLogicos: TStringList): string;
  protected
  public
    constructor Create;
    destructor Destroy; override;
    procedure  Consultar(CamposAMostrar, CamposAComparar, OperadoresComparacion,
Parametros, OperadoresLogicos: TStringList);
    procedure
Mostrar(Identificador:Longint;PantallaEncabezado,PantallaContenido,PantallaAccesorios:TScrollB
ox;Arbol:TsuTreeView);
    procedure Eliminar; override;
    procedure Guardar; override;
    procedure GuardarADisco(NombreArchivo:string);
    procedure RecuperarDeDisco(NombreArchivo:string);
  published
end;

type
  {Clase TEstructura}
  TEstructura = class(TSolicitud)
  private
  protected
  public
    constructor Create;
    destructor Destroy; override;
    procedure Guardar(IDParrafoHijo, IDParrafoPadre:Longint); overload;
  published
end;

type
  {Clase TTemasRelacionados}
  TTemasRelacionados = class(TSolicitud)

```

```

private
protected
public
    constructor Create;
    destructor Destroy; override;
    procedure Guardar(IDArticuloRelacionado, IDArticulo:Longint); overload;
published
end;

```

```

type
{Clase TUsuario}
TUsuario = class(TSolicitud)
private
    FNombreUsuario,FContrasena,FDirBaseDatos,FTipo:string;
    FEncontrado:boolean;
    FConsulta:TConsulta;
    Seleccionar,Comparar,Operadores,Parametros,Enlace:TStringList;
    procedure BorrarSusArticulos(NombreUsuario:string);
protected
public
    constructor Create;
    destructor Destroy; override;
    function Buscar(NombreUsuario,Contrasena:string):boolean;
    procedure Guardar(NombreUsuario,Contrasena:string); overload;
    procedure CambiarDatos(NombreUsuario,NuevoNombreUsuario,NuevaContrasena:string);
    function Eliminar(NombreUsuario:string):boolean;
published
    property DirBaseDatos:string read FDirBaseDatos write FDirBaseDatos;
    property NombreUsuario:string read FNombreUsuario write FNombreUsuario;
    property TipoUsuario:string read FTipo;
end;

```

```

type
{Clase TArticuloUsuario}
TArticuloUsuario = class(TSolicitud)
private
protected
public
    constructor Create;
    destructor Destroy; override;
    procedure Guardar(NombreUsuario:string; IDArticulo:Longint); overload;
published

```

```
end;
```

```
type
```

```
{Clase TGlosario}
```

```
TGlosario = class(TSolicitud)
```

```
private
```

```
protected
```

```
public
```

```
constructor Create;
```

```
destructor Destroy; override;
```

```
procedure Guardar(Concepto,Definicion:string); overload;
```

```
procedure Editar(Concepto:string);
```

```
published
```

```
end;
```

```
type
```

```
{Clase TUsuarioGlosario}
```

```
TUsuarioGlosario = class(TSolicitud)
```

```
private
```

```
protected
```

```
public
```

```
constructor Create;
```

```
destructor Destroy; override;
```

```
procedure Guardar(NombreUsuario,Concepto:string); overload;
```

```
published
```

```
end;
```

```
type
```

```
{Clase TArticuloGlosario}
```

```
TArticuloGlosario = class(TSolicitud)
```

```
private
```

```
protected
```

```
public
```

```
constructor Create;
```

```
destructor Destroy; override;
```

```
procedure Guardar(IDArticulo:Longint;Concepto:string); overload;
```

```
published
```

```
end;
```

```
procedure
```

```
Recorrer(NumeroSubParrafos:integer;Parrafo:TParrafo;Arbol:TsuiTreeView;NodoPadre:TTreeNode);
```

ULaboratorio.pas

Funcionalidad: Unidad interfaz de usuario, donde se permite al Investigador hacer sus respectivos experimentos, modelado y simulación.

Instancia de clases:

type

```
TfrmLaboratorio = class(TForm)
  suiFormLaboratorio: TsuiForm;
  suiPanelForrester: TsuiPanel;
  Separador: TSplitter;
  suiPanelSimulacion: TsuiPanel;
  suiPanelControlForrester: TsuiPanel;
  suiPanelControlSimulacion: TsuiPanel;
  EditorEVL1: TEditorEVL;
  GraficadorEVL1: TGraficadorEVL;
  MotorEvl1: TMotorEvl;
  ImgForrester: TImageList;
  suiToolBarForrester: TsuiToolBar;
  tbParametro: TToolButton;
  tbNivel: TToolButton;
  tbFlujo: TToolButton;
  tbRetardo: TToolButton;
  tbTabla: TToolButton;
  tbVariable: TToolButton;
  tbExogena: TToolButton;
  tbInicioCiclo: TToolButton;
  tbClon: TToolButton;
  tbSector: TToolButton;
  tbInformacion: TToolButton;
  ToolButton12: TToolButton;
  tbZoom1: TToolButton;
  tbZoomMas: TToolButton;
  tbZoomMenos: TToolButton;
  ToolButton16: TToolButton;
  tbEliminar: TToolButton;
  ToolButton1: TToolButton;
  ImgSimulacion: TImageList;
  suiToolBarSimulacion: TsuiToolBar;
  tbIniciar: TToolButton;
  tbPausar: TToolButton;
```

```

tbParar: TToolButton;
tbUnPaso: TToolButton;
tbCondiciones: TToolButton;
tbCambiarValores: TToolButton;
ToolButton23: TToolButton;
tbNuevaVista: TToolButton;
tbEliminarVista: TToolButton;
tbVistaTabla: TToolButton;
tbVistaGrafica: TToolButton;
tbTrayectorias: TToolButton;
tbPropiedadesGrafica: TToolButton;
suiToolBar1: TsuiToolBar;
tbAbrir: TToolButton;
tbGuardar: TToolButton;
suiToolBarEscenarios: TsuiToolBar;
tbEscenarios: TToolButton;
OpenDialog1: TOpenDialog;
SaveDialog1: TSaveDialog;
ToolButton2: TToolButton;
suiComboBoxEscenarios: TsuiComboBox;
procedure FormCreate(Sender: TObject);
procedure MotorEvl1Preparar(Sender: TObject; var HuboError: Boolean);
procedure MotorEvl1PasoSimulacion(Sender: TObject);
procedure FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure suiToolBarForresterClick(Sender: TObject);
procedure tbZoom1Click(Sender: TObject);
procedure tbZoomMasClick(Sender: TObject);
procedure tbZoomMenosClick(Sender: TObject);
procedure tbEliminarClick(Sender: TObject);
procedure tbEscenariosClick(Sender: TObject);
procedure suiComboBoxEscenariosChange(Sender: TObject);
procedure ToolButton1Click(Sender: TObject);
procedure tbIniciarClick(Sender: TObject);
procedure tbPausarClick(Sender: TObject);
procedure tbPararClick(Sender: TObject);
procedure tbUnPasoClick(Sender: TObject);
procedure tbCondicionesClick(Sender: TObject);
procedure tbCambiarValoresClick(Sender: TObject);
procedure tbNuevaVistaClick(Sender: TObject);
procedure tbEliminarVistaClick(Sender: TObject);
procedure tbVistaTablaClick(Sender: TObject);

```

```
procedure tbVistaGraficaClick(Sender: TObject);
procedure tbTrayectoriasClick(Sender: TObject);
procedure tbPropiedadesGraficaClick(Sender: TObject);
procedure tbGuardarClick(Sender: TObject);
procedure tbAbrirClick(Sender: TObject);
procedure ToolButton2Click(Sender: TObject);
procedure FormDestroy(Sender: TObject);
procedure SeparadorMoved(Sender: TObject);
private
  FrmCondiciones: TFCondiciones;
  FrmModificarVariables: TFrmCondicionesSimulacion;
  CondicionesSimulacion: TCondiciones;
  FGuardarBaseDatos: Boolean;
  FIdentificador: Longint;
  procedure Cargar_archivo(Archivo:string);
  procedure LlenarEscenarios;
  procedure IniciarDiagrama;
  procedure GuardarEnBD(Identificador:Longint);
public
  CambioMotor: Boolean;
  procedure AbrirDesdeBD(Identificador:Longint);
end;
```