

APLICACIÓN WEB PARA LA ENSEÑANZA DE FIGURAS RETÓRICAS

ERIKA TATIANA MÉNDEZ CIFUENTES
JANETH SUSANA DÍAZ CARRILLO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2012

APLICACIÓN WEB PARA LA ENSEÑANZA DE FIGURAS RETÓRICAS

ERIKA TATIANA MÉNDEZ CIFUENTES
JANETH SUSANA DÍAZ CARRILLO

Trabajo de Grado para optar por el título de Ingeniera de Sistemas

Director
SONIA CRISTINA GAMBOA SARMIENTO
Doctora en Educación

Codirector
JUAN CARLOS RINCÓN ACUÑA
Ingeniero de Sistemas

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO-MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2012

DEDICATORIA

A Dios, que me ha dado la fuerza para sobrellevar situaciones difíciles, la sabiduría para alcanzar este logro en mi vida y me ha rodeado de personas maravillosas.

A mis padres Miguel Méndez y Oliva Cifuentes, por todo el amor y apoyo brindado durante toda mi vida y por regalarme la mejor educación, que permitieron traerme a este punto de mi vida.

A mi hermana, Marcella Martínez, quién durante toda mi vida me ha apoyado en cada decisión importante y que con sus consejos me ha hecho mejor persona.

A todos mis amigos, que han hecho que mi vida como estudiante fuera una experiencia inolvidable.

A mis profesores de colegio y universidad, que con el conocimiento que me brindaron, hicieron que me desarrollara como profesional.

A todas las personas que con su apoyo han hecho que este sueño se vuelva realidad.

Erika Tatiana Méndez Cifuentes

DEDICATORIA

A Dios gracias por la salud, sabiduría por permitirme cumplir una más de mis metas y sueños, a mis padres Juan Díaz y Mariela Carrillo por todo el amor, apoyo, paciencia y confianza que siempre me han brindado, son mi fortaleza en este camino de la vida, a mis hermanos, amigos y a mis compañeros con quienes he compartido estos años de vida y durante la carrera y han sido un apoyo más en todo momento.

Gracias a todos

Janeth Susana Díaz Carrillo

AGRADECIMIENTOS

A nuestros directores de proyecto profesora Sonia Cristina Gamboa sarmiento y a el Ingeniero Jun Carlos Rincón Acuña por la confianza que depositaron en nosotras, por el apoyo y la guía en la elaboración y desarrollo de este proyecto.

A la Universidad Industrial de Santander y a la Escuela de Ingeniería Sistemas e Informática, y profesores por la formación profesional recibida.

A nuestros familiares y amigos, que nos apoyaron durante este proceso.

CONTENIDO

INTRODUCCIÓN	14
1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA	15
1.1 PLANTEAMIENTO DEL PROBLEMA	15
1.2 JUSTIFICACIÓN	15
2. OBJETIVOS DEL PROYECTO	16
2.1 OBJETIVO GENERAL	16
2.2 OBJETIVOS ESPECÍFICOS	16
3. MARCO TEÓRICO	17
3.1 LA INTELIGENCIA ARTIFICIAL	17
3.2 AGENTES INTELIGENTES	19
3.3 FIGURAS RETÓRICAS	20
3.4 SILOGISMO	21
3.4.1 Silogismo categórico	21
3.4.2 El Entimema	21
3.4.3 Silogismos hipotéticos, alternativos y disyuntivos	21
3.4.4 Reglas o axiomas de validez	22
3.4.5 Los teoremas generales del silogismo	23
3.5 ANALOGÍA	23
3.5.1 El fundamento de la analogía	25
3.5.2 Estructura de la analogía	26
3.6 FALACIA	26
3.6.1 Falacias formales	27
3.6.2 Falacias semilógicas o verbales	27
3.6.3 Falacias maternas	27
3.7 SÍMIL	28
3.7.1 Uso del símil	28
3.8 RAZONAMIENTO LÓGICO	29

3.8.1 Razonamiento deductivo	30
4. ESTADO DEL ARTE	32
5. ESTRUCTURA LÓGICA	34
5.1 LISTA DE CLASES	34
5.2 LISTA DE ARCHIVOS	42
5.2.1 Lista de archivos carpeta components.....	42
5.2.2 Lista de archivos carpeta includes	43
5.3 DEFINICIÓN DE FUNCIONES	44
5.3.1 Definición de funciones carpeta includes	44
5.3.1.1 analogias.php.....	44
5.3.1.2 falacias.php.....	45
5.3.1.3 razonamiento.php	49
5.3.1.4 silogismos.php	50
5.3.1.5 simil.php.....	58
5.3.2 Definición de funciones carpeta components.....	60
5.3.2.1 analogias.php.....	60
5.3.2.2 falacias.php.....	61
5.3.2.3 razonamiento.php	61
5.3.2.4 silogismos.php	61
5.3.2.5 simil.php.....	61
5.4 MODELO ENTIDAD-RELACIÓN	62
5.5 DIAGRAMA DE FLUJO FIGURAS RETÓRICAS.....	62
5.5.1 Diagrama de flujo silogismos	63
5.5.2 Diagrama de flujo falacias	67
5.5.3 Diagrama de flujo símil	72
5.5.4 Diagrama de flujo analogías	73
5.5.5 Diagrama de flujo razonamiento	74
6. DISEÑO COMPUTACIONAL	76
6.1 DEFINICIÓN ACTORES	76
6.2 CASOS DE USO.....	77

6.3 NAVEGACIÓN POR VENTANAS.....	81
6.4 PROTOTIPOS DE LA APLICACIÓN	83
6.5 EVALUACIÓN.....	104
CONCLUSIONES	110
RECOMENDACIONES.....	111
BIBLIOGRAFÍA.....	112

LISTA DE FIGURAS

Figura 1. Estructura de una Analogía	26
Figura 2. Modelo entidad-relación.....	62
Figura 3. Diagrama de flujo de Silogismos 1	63
Figura 4. Diagrama de flujo de Silogismos 2	64
Figura 5. Diagrama de flujo de Silogismos 3	65
Figura 6. Diagrama de flujo de Silogismos 4	66
Figura 7. Diagrama de flujo de Falacias 1	67
Figura 8. Diagrama de flujo de Falacias 2	68
Figura 9. Diagrama de flujo de Falacias 3	69
Figura 10. Diagrama de flujo de Falacias 4	70
Figura 11. Diagrama de flujo de Falacias 5	71
Figura 12. Diagrama de flujo de Símil.....	72
Figura 13. Diagrama de flujo de Analogías.....	73
Figura 14. Diagrama de flujo de Razonamiento 1.....	74
Figura 15. Diagrama de flujo de Razonamiento 2.....	75
Figura 16. Navegación por ventanas de la aplicación desarrollada para el estudiante.....	82
Figura 17. Navegación por ventanas de la aplicación desarrollada para el profesor	82
Figura 18. Página de Registro de Usuario	83
Figura 19. Página de Inicio de Sesión	84
Figura 20. Página Principal.....	85
Figura 21. Página 1 de Teoría de Silogismo.....	86
Figura 22. Página 2 de Teoría de Silogismo.....	87
Figura 23. Página 3 de Teoría de Silogismo.....	87
Figura 24. Página de Teoría de Razonamiento Lógico.....	88
Figura 25. Página 1 de Teoría de Analogía	89
Figura 26. Página 2 de Teoría de Analogía	89
Figura 27. Página 3 de Teoría de Analogía	90
Figura 28. Página 4 de Teoría de Analogía	90
Figura 29. Página de Teoría de Falacia.....	91
Figura 30. Página de Teoría de Símil	92
Figura 31. Página de Ejemplos de Silogismo	93
Figura 32. Página de Ejemplos de Razonamiento Lógico	94
Figura 33. Página de Ejemplos de Analogía.....	95
Figura 34. Página de Ejemplos de Símil	96
Figura 35. Página de Ejemplos de Falacia	97
Figura 36. Página de Ejercicios de Silogismo	98
Figura 37. Página de Ejercicios de Razonamiento Lógico.....	99
Figura 38. Página de Ejercicios de Analogía	100
Figura 39. Página de Ejercicios de Símil	101

Figura 40.Pagina 1 de Ejercicios de Falacia	102
Figura 41.Pagina 2 de Ejercicios de Falacia	102
Figura 42.Pagina de seguimiento a estudiantes	103
Figura 43.Mensaje de acceso denegado a estudiantes.....	104
Figura 44.Elección del producto (código fuente), las características (eficiencia, funcionalidad y usabilidad) y las subcaracterísticas para la evaluación del producto software.....	105
Figura 45.Asignación de métricas a cada subcaracterística (I).....	106
Figura 46.Asignación de métricas a cada subcaracterística (II).....	106
Figura 47.Asignación de métricas a cada subcaracterística (III).....	106
Figura 48.Asignación de métricas a cada subcaracterística (IV)	106
Figura 49.Asignación de métricas a cada subcaracterística (V)	106
Figura 50.Asignación de métricas a cada subcaracterística (VI)	106
Figura 51.Resultados obtenidos de la evaluación de la aplicación web desarrollada usando el software QUIS.....	106

LISTA DE TABLAS

Tabla 1. Definición de actores.....	76
Tabla 2. Narrativa del caso de uso Realizar ejercicio	77
Tabla 3. Narrativa del caso de uso Registrar usuario	78
Tabla 4. Narrativa del caso de uso Revisar ejemplo.....	78
Tabla 5. Narrativa del caso de uso Seleccionar tema.....	79
Tabla 6. Narrativa del caso de uso Revisar teoría	80
Tabla 7. Narrativa del caso de uso Revisar ejercicios realizados	81

RESUMEN

TÍTULO

APLICACIÓN WEB PARA ENSEÑANZA DE FIGURAS RETORICAS.ⁱ

AUTORES

JANETH SUSANA DÍAZ CARRILLO, ERIKA TATIANA MÉNDEZ CIFUENTES.ⁱⁱ

PALABRAS CLAVES

Aplicación web, tecnología de información, enseñanza, razonamiento lógico, figuras retóricas.

DESCRIPCIÓN

Este trabajo de grado propone el desarrollo de una aplicación web para la enseñanza de figuras retóricas partiendo del funcionamiento de una herramienta ya existente, cuyo lenguaje de programación con los años a resultado obsoleto. Y surge la necesidad de realizar una propuesta que evite que una situación como esta vuelva a ocurrir, implementándose en un lenguaje de programación estándar (PHP y HTML).

Para el desarrollo se implementa una metodología que consiste en el análisis de la herramienta existente que permite entender su funcionamiento, detectar las fallas y realizar nuevas propuestas para mejorar el funcionamiento que se refleja en la aplicación desarrollada en este proyecto. La cual pretende poner a prueba los conocimientos que el usuario posee sobre las temáticas manejadas mediante la realización de ejercicios. Adicionalmente se propone mejorar la interacción con el usuario por medio de la visualización de fundamentos teóricos apoyados con diversos ejemplos que sirven como soporte para la enseñanza de las figuras retóricas manejadas.

La importancia del desarrollo de aplicaciones web con fines educativos se centra en el apoyo que se le da al estudiante para complementar su formación académica interactuando con las tecnologías de información que simulan la relación existente en la vida real entre maestro y estudiante. Refiriéndose explícitamente a las figuras retóricas que se definen como representaciones del razonamiento lógico de una persona que busca mejorar los procesos comunicativos, toma de decisiones y comprensión del punto de vista de otro, siendo el tema central de este proyecto que tiene como objetivo general la enseñanza de las figuras retóricas: silogismo, analogía, falacia y símil.

ⁱ Trabajo de grado

ⁱⁱ Facultad de Ingeniería Físico-mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Sonia Cristina Gamboa Sarmiento. Codirector Juan Carlos Rincón Acuña

ABSTRACT

TITLE: WEB APPLICATION FOR THE EDUCATION OF RHETORIC FIGURES.ⁱⁱⁱ

AUTHORS: JANETH SUSANA DIAZ CARRILLO, ERIKA TATIANA MÉNDEZ CIFUENTES.^{iv}

KEYWORDS: Web application, information technology, education, logical reasoning, rhetoric figures.

DESCRIPTION

This graduate project proposes the development of a web application that teaches about rhetoric figures starting from the functionality of an existing tool, whose programming language has been obsolete throughout the years. And the need to make a proposition so that situations like this will not happen again rises, implementing itself in a standard programming language (PHP and HTML).

For the development a methodology it is implemented that consists of analyzing the existing tool that allows us to understand its functionality, detects failures and creates new proposals to better the function that is reflected in this project, which intends to test the knowledge that the user possesses in respect to the thematic used by performing exercises. In addition, user interaction is intended to be improved through the visualization of theoretical fundamentals with the aid of a diversity of examples that act as a support to the education of the rhetoric figures used.

The importance of the development of web applications with educational goals is centered in the help that is provided to the student to complement his/her academic formation interacting with the information technologies that simulate the existent relation in real life between professor and student. Referring explicitly to the rhetoric figures that are defined as representations of the logical reasoning that a person looks for to improve communication processes, decision making and the comprehension of others point of view, it being the central theme of this project with the main objective of teaching rhetoric figures: syllogism, analogy, fallacy and simile.

ⁱⁱⁱ Graduation project

^{iv} Physical and Mechanical Engineering Faculty. School of Systems and Informatics Engineer. Director Sonia Cristina Gamboa Sarmiento. Codirector Juan Carlos Rincón Acuña

INTRODUCCIÓN

En la actualidad existen sistemas computacionales que sirven de apoyo a los estudiantes en el proceso de aprendizaje y enseñanza, representando la interacción que hay entre un maestro y el estudiante. A través del tiempo estos sistemas computacionales han ido evolucionando y haciendo parte de la vida cotidiana del estudiante ya que permiten poner en práctica los conocimientos teóricos adquiridos mediante la realización de tareas dentro del marco del aprendizaje interactivo, desarrollando así un modelo de sus conocimientos, habilidades y experiencia apoyándose en las tecnologías de información y comunicación (TICs).

Con este proyecto se pretende integrar el uso de la tecnología con los conocimientos sobre figuras retóricas en una aplicación que permite al estudiante practicar sobre una temática propuesta para poner a prueba y consolidar sus conocimientos previos, así como adquirir nuevos conocimientos sobre la temática manejada. Basándose en el manejo de estudiantes, permite adaptar las estrategias de instrucción, de explicación, ejemplos, prácticas y problemas, según se crea necesario.

Este proyecto consta de seis capítulos. En el primer capítulo se realiza un planteamiento del problema y se justifica la solución planteada en este proyecto. En el segundo capítulo se plantean los objetivos propuestos.

En el tercer capítulo se presenta el marco teórico que es recopilación de información que se utiliza como base para el desarrollo de este proyecto. En el cuarto capítulo se encuentra el estado del arte que brinda información sobre diferentes aplicaciones que se han desarrollado.

En el quinto capítulo se encuentra la estructura lógica de la aplicación a la que se le realiza la reingeniería de la cual se selecciona las clases y funciones que se utilizan para el desarrollo del motor de inferencia.

En el sexto capítulo se encuentra el diseño computacional donde se encuentra el desarrollo del cual hace parte los actores, narrativa de los casos de uso, la navegación por ventanas y los prototipos de esta aplicación de la nueva aplicación

Para finalizar se presentan las conclusiones y recomendaciones sugeridas con la elaboración de este proyecto

1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

El Grupo de investigación Filosofía y Enseñanza de la Filosofía –de la Escuela de Ingeniería de Sistemas e Informática– se ocupa, como una de sus líneas de investigación, del diseño de aplicativos software que promuevan habilidades y conocimientos para el ejercicio, no sólo de la filosofía, sino de la ciudadanía; entre ellos el razonamiento lógico y la construcción de discursos con base en la estructura de las figuras retóricas. Para ello se apela a la construcción de servicios web que garanticen la práctica de ejercicios didácticos de forma individual en pro de mejorar procesos comunicativos, toma de decisiones, comprensión del punto de vista del otro, entre otros.

1.2 JUSTIFICACIÓN

Uno de los presupuestos para el aprendizaje de la filosofía para el Grupo es el de que en la medida que un aprendiz es consciente de cómo funcionan sus procesos mentales se facilita el aprendizaje de los conocimientos y habilidades relacionados con ellos. De allí que si se logra una representación de los mismos, y a partir de ella, una automatización que muestre, que disponga, al aprendiz una demostración de tal representación, el aprendiz tendrá la oportunidad de comprender el fenómeno representado, aun cuando en la realidad sus procesos mentales funcionen de manera diferente.

Por su parte, el razonamiento lógico es una de las habilidades que se requiere para el ejercicio de la ciudadanía, de los procesos comunicativos, de toma de decisiones, etc. Esta forma de razonamiento ha sido teorizada y formalizada, entre otras, mediante las llamadas figuras retóricas, las cuales permiten comprender tanto formas de razonar, como formas de discurso, mediante estructuras que cuentan con elementos invariantes. Entre ellas, las figuras retóricas que son de interés para este proyecto son el silogismo, el símil, la analogía y la falacia.

Este proyecto propone el desarrollo de un sitio web que, mediante un motor de inferencia, permite que un aprendiz realice ejercicios de diferentes niveles de complejidad para lograr un nivel de comprensión de tales figuras. Para ello se llevará a cabo un proceso de reingeniería de una herramienta ya existente que con los años ha resultado obsoleta.

2. OBJETIVOS DEL PROYECTO

2.1 OBJETIVO GENERAL

Diseñar y publicar como aplicación web un motor de inferencia para la enseñanza de figuras retóricas

2.2 OBJETIVOS ESPECÍFICOS

- Determinar la estructura lógica para la operación de las figuras retóricas.
- Diseñar una estrategia didáctica para garantizar el aprendizaje con elementos y actividades propios de las figuras retóricas.
- Diseñar la interfaz de usuario que sea eficiente, eficaz y efectiva tanto para el alumno como para el docente.
- Desarrollar un servicio web para validar la estrategia didáctica de aprendizaje de figuras retóricas

3. MARCO TEÓRICO

Diferentes sistemas computacionales tienen como finalidad apoyar el proceso de enseñanza y en particular de aprendizaje de los estudiantes, sin embargo, son pocas las aplicaciones que pueden demostrar que apoyan en el logro de los contenidos y/o habilidades propuestos.

En muchos países, la educación ha sido y está siendo fuertemente influenciada por la inserción de la tecnología. Esto se puede observar, entre otros, en la optimización de recursos, la mejora de los procesos de enseñanza y aprendizaje, una educación más equitativa, generar una formación continua y mejorar la “sintonía” entre la escuela y la sociedad.¹

El software educativo juega un papel importante como apoyo al docente y al estudiante en la formación de este último. Si bien existen importantes y numerosos desarrollos de sistemas, los resultados no han sido lo esperado. En su mayoría corresponden a sistemas de práctica y ejercitación cuya principal característica es entregar al alumno la posibilidad de ejercitarse en una determinada tarea una vez obtenidos los conocimientos necesarios para el dominio de la misma.²

A todo esto, unimos el evidente incremento en la utilización de tecnologías de la información y las comunicaciones en los procesos de enseñanza y aprendizaje, y en este contexto las más demandadas son precisamente las tecnologías basadas en inteligencia artificial, como los tutores inteligentes y los sistemas de gestión del aprendizaje.

3.1 LA INTELIGENCIA ARTIFICIAL

La Inteligencia Artificial (IA), puede considerarse como una de las disciplinas más nuevas, siendo curiosamente considerada a la vez como una gran desconocida y una de las que más interés profano despierta. Esto es debido a que poca gente tiene claro qué es la IA, pero sin embargo es considerada por una gran mayoría de científicos como la disciplina donde han pensado alguna vez en trabajar.

La IA trata de desarrollar sistemas que piensen y actúen racionalmente. Actualmente abarca una gran cantidad de áreas, desde algunas muy generales

¹Bruner, J. (2000). Educación: Escenarios de Futuro. Nuevas Tecnologías y Sociedad de la Información. PREAL: Promoción de la Reforma Educacional en América Latina y el Caribe.

²Gros, B. (1997). *Diseños y programas educativos Pautas pedagógicas para la elaboración de software*. Barcelona.

como razonamiento, búsqueda, etc. A otras más específicas como los sistemas expertos, sistemas de diagnóstico, etc.

Se podría indicar que la IA puede ser aplicada hoy en día a infinidad de disciplinas científicas, ya que, la IA es susceptible a aparecer donde se requiera el intelecto humano. En este aspecto cada nueva incursión de la IA en un campo diferente supone el empleo de una metodología de aplicación propia del investigador que lo intenta, pues a diferencia de otras disciplinas, no existen ciertos estándares o criterios que unifiquen el proceso de aproximación de la IA a otras áreas y en consecuencia, a problemas reales.³

La representación del conocimiento y el razonamiento es un área de la inteligencia artificial cuyo objetivo fundamental es representar el conocimiento de una manera que facilite la inferencia (sacar conclusiones) a partir de dicho conocimiento. Analiza cómo pensar formalmente - cómo usar un sistema de símbolos para representar un dominio del discurso (aquello de lo que se puede hablar), junto con funciones que permitan inferir (realizar un razonamiento formal) sobre los objetos. Generalmente, se usa algún tipo de lógica para proveer una semántica formal de como las funciones de razonamiento se aplican a los símbolos del dominio del discurso, además de proveer operadores como cuantificadores, operadores modales, etc. Esto, junto a una teoría de interpretación, da significado a las frases en la lógica.

Cuando diseñamos una representación del conocimiento (y un sistema de representación del conocimiento para interpretar frases en la lógica para poder derivar inferencias de ellas) tenemos que hacer elecciones a lo largo de un número de ámbitos de diseño. La decisión más importante que hay que tomar es la expresividad de la representación del conocimiento. Cuanto más expresiva es, decir algo es más fácil y más compacto. Sin embargo, cuanto más expresivo es un lenguaje, más difícil es derivar inferencias automáticamente de él. Un ejemplo de una representación del conocimiento poco expresiva es la lógica proposicional; que estudia la formación de proposiciones complejas a partir de proposiciones simples, y la inferencia de proposiciones a partir de proposiciones, pero sin tener en cuenta la estructura interna de las proposiciones más simples.

Hay diferentes formas de representación de conocimiento y el razonamiento que cumple con el objetivo fundamental que se mencionó anteriormente, entre éstas, podemos encontrar a las figuras retóricas.

Actualmente en la IA ha surgido un nuevo paradigma conocido como «paradigma de agentes». Dicho paradigma aborda el desarrollo de entidades que puedan

³V. Julián, V. B. (Mayo-Junio de 2000). *Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial*. Recuperado el 01 de Septiembre de 2012, de <http://www.ati.es/novatica/2000/145/vjulia-145.pdf>

actuar de forma autónoma y razonada. Y si se retoma la definición dada anteriormente donde se consideraba a la IA como un medio para el desarrollo de sistemas que piensen y actúen racionalmente, se puede pensar que la IA en su conjunto, trata realmente de construir dichas entidades autónomas e inteligentes.

De acuerdo con esta visión, se puede considerar a la IA como una disciplina orientada a la construcción de agentes inteligentes donde se integran las diferentes áreas que ésta comprende.⁴

3.2 AGENTES INTELIGENTES

Los agentes inteligentes se pueden definir como una entidad software que, basándose en su propio conocimiento, realiza un conjunto de operaciones destinadas a satisfacer las necesidades de un usuario o de otro programa, bien por iniciativa propia o porque alguno de éstos lo requiere.⁵

Hay agentes inteligentes diseñados específicamente para procesar consultas que poseen al menos uno de los siguientes elementos: capacidad de proceso, conocimiento del entorno donde se mueven e información de un dominio. Estos agentes son llamados agentes inteligentes de la información.

Un agente inteligente de la información, tiene capacidad de proceso puesto que puede descomponer una consulta en subconsultas y asociar los distintos términos resultantes a otros términos relacionados o afines. Su conocimiento del entorno le viene dado por su propio conocimiento y por el de otros agentes que se comunican con él (el conocimiento puede ser adquirido: del mismo usuario o de otros agentes con los que se encuentra mientras realiza una tarea determinada). En todo momento, el agente inteligente de la información deberá saber a qué información acceder o a qué otro agente dirigirse para obtenerla.

Tanto los agentes inteligentes, como los agentes inteligentes de la información sirven como base para muchas técnicas de aprendizaje; estos sistemas pueden ayudar a la labor del docente o del alumno, sin involucrarse directamente en la actividad instructiva. Un profesor auxiliar, puede proporcionarle al profesor, la carpeta de aprendizaje de los estudiantes, incluso la actuación de aprendizaje, las equivocaciones, nivel de esfuerzo y motivaciones, entre otros aspectos. Este

⁴V. Julián, V. B. (Mayo-Junio de 2000). *Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial*. Recuperado el 01 de Septiembre de 2012, de <http://www.ati.es/novatica/2000/145/vjulia-145.pdf>

⁵Vargas-Quesada, P. H. (Abril de 1999). *El profesional de la información. Agentes inteligentes: definición y tipología. Los agentes de información*. Recuperado el 01 de Septiembre de 2012, de http://www.elprofesionaldelainformacion.com/contenidos/1999/abril/agentes_inteligentes_definicion_y_tipologia_los_agentes_de_informacion.html

archivador puede ayudar al profesor a entender al estudiante y responder apropiadamente a cada uno. El estudiante auxiliar, por ejemplo, puede ayudar al estudiante a coleccionar información para realizar la actividad de aprendizaje.

Los agentes inteligentes, proporcionan a los estudiantes recursos que pueden clasificarse en tres categorías: contenidos (libros, bibliotecas, museos, bases de datos, etc.), soporte computacional (calculadora, software, etc.) y comunidad (compañeros de estudio, profesores, padres, voluntarios, etc.). De esta manera el estudiante dispone de múltiples medios y de un contexto, por lo que el agente es presentado en un medio ambiente de aprendizaje social⁶. Un agente educativo no sólo puede imponer directamente sus perspectivas en el estudiante, sino que también estimula el aprendizaje y colabora entre los estudiantes humanos.

3.3 FIGURAS RETÓRICAS

Se entiende por figura retórica toda modificación del uso normal y corriente del lenguaje; para que dicha modificación se considere realmente como una figura retórica debe obedecer al propósito de alcanzar una expresión innovadora y atrayente. Las figuras aparecen, sobre todo, en el ámbito del lenguaje literario, lo cual no significa que sólo se encuentren en él. Tanto en el uso cotidiano como en los distintos lenguajes especializados es posible hallar un inventario rico y variado de figuras; algunas de ellas se utilizan casi inconscientemente, otras son deliberadas, y responden a los más diversos propósitos: la persuasión (en el mundo de la publicidad), la necesidad de captar rápidamente la atención del lector y de saber mantenerla (en el periodismo), el afán de crear un efecto estético (en la literatura), etc.

Retórica es el arte de persuadir o convencer a un auditorio no sólo con los recursos de argumentación y figuras gramaticales, sino con la voluntad de mostrar la verdad, la belleza y el bien de una posición ante un problema o de una manera de ver un asunto en particular. En su aplicación contemporánea al terreno de la literatura, la retórica se ocupa de estudiar aquellos recursos expresivos que permiten al escritor conseguir los fines que en cada caso se proponga, y que pueden ser muy diversos: persuadir, enseñar, entretener, emocionar, crear belleza, etc.⁷

⁶C. Choua, T. C. (2002). *Refining the learning companion: the past, present, future of educational agents*, Computer & Educacion.

⁷Perelman, C. (1997). *El imperio retórico. Retorica y argumentacion*. Editorial Norma S.A.

A la hora de analizar una determinada figura retórica no basta con identificarla, sino que es preciso valorar y explicar la importancia y significación que dicha figura alcanza en un texto concreto. El objetivo que se debe perseguir es, la adquisición de la capacidad para determinar en cada caso el significado de una figura, es decir, para explicar qué función desempeña o qué efecto artístico produce, y por qué ha sido utilizada por el autor. Por último, téngase en cuenta que, aunque las definiciones estrictas y "académicas" de las figuras no importan tanto como su identificación y análisis, el alumno está obligado a conocer la terminología y a aplicarla con acierto.⁸

3.4 SILOGISMO

3.4.1 Silogismo categórico

El silogismo categórico se define como una forma de razonamiento consistente en tres proposiciones categóricas que contienen tres términos, y solo tres; dos de las proposiciones son las premisas, la tercera es la conclusión. Una inferencia silogística puede interpretarse, en realidad, como una comparación de las relaciones entre cada uno de dos términos y un tercero, para descubrir las que existen entre los dos términos. El término contenido en ambas premisas es el *término medio*, el predicado de la conclusión el *término mayor*, y su sujeto el *término menor*.

3.4.2 El Entimema

Aunque el razonamiento silogístico aparece con frecuencia en el habla cotidiana, no se suele advertir su presencia porque se lo formula de manera incompleta. Un silogismo enunciado incompletamente, en el que una de las premisas o la conclusión está tácita e inexpresada, recibe el nombre de *entimema*. Un silogismo en el que no se expresa la premisa mayor es un entimema de *primer orden*. Los entimemas que suprimen la premisa menor son de *segundo orden*.

Aunque los entimemas no representan una forma nueva de inferencia, su reconocimiento es de gran importancia práctica. Se cree a menudo que las

⁸ Proyecto Aula Literatura. Figuras literarias. Teoría. (s.f.). Recuperado el 01 de 09 de 2012, de http://lenguayliteratura.org/ltr/index.php?option=com_content&task=view&id=1654&Itemid=1

llamadas inferencias inductivas son un modo especial de razonamiento, cuando de hecho se trata simplemente de entimemas de primer orden.⁹

3.4.3 Silogismos hipotéticos, alternativos y disyuntivos

Los silogismos hipotéticos contienen tres proposiciones: la primera premisa, o premisa *mayor*, es una proposición hipotética; la segunda premisa, o premisa *menor*, y la conclusión son proposiciones categóricas. Dentro de los silogismos hipotéticos, se encuentran los *silogismos hipotéticos puros*, que contienen dos proposiciones hipotéticas como premisas y otra como conclusión. El valor de este tipo de razonamiento es muy evidente, pues a menudo nos es más fácil establecer la verdad de una proposición hipotética y la de su antecedente que la del consecuente, que puede, pues, ser establecido *indirectamente*, como conclusión de la inferencia.

Ahora, los silogismos alternativos mixtos, son inferencias válidas que pueden extraerse de una proposición alternativa como premisa mayor y una categórica como premisa menor. Un silogismo alternativo es válido, cuando la premisa menor niega una de las alternantes y la conclusión afirma la verdad de la otra. Esta clase de silogismo se emplea frecuentemente con el objeto de *eliminar* explicaciones o soluciones propuestas para diversos problemas.

Por último, al hablar de silogismos disyuntivos, se refiere a razonamientos donde la premisa mayor es una disyunción y la menor una proposición categórica; afirmar una disyunción significa que una, por lo menos, de las disyuntivas, es falsa. Un silogismo disyuntivo es válido si la premisa menor afirma una de las disyuntivas y la conclusión niega la otra. La forma esquemática del razonamiento es: no se da el caso de que A sea B y C sea D; A es B; por lo tanto, C no es D. Se dice que la inferencia está en el *modus ponendotollens*, pues al afirmar (en la premisa menor), negamos (en la conclusión).

3.4.4. Reglas o axiomas de validez

Estudiando el silogismo como una forma de inferencia por medio de la cual se fija la conexión entre dos términos sobre la base de sus relaciones con un tercer común a ambas, se podrá “ver” que estos axiomas expresan realmente las condiciones de validez. Los axiomas del silogismo categórico se dividen en dos

⁹Nagel, M. C. (s.f.). Introducción a la lógica y al método científico. II Lógica aplicada y método científico. Buenos Aires: Amorrortu editores.

clases: los que tratan de la cantidad o la distribución de los términos, y los que tratan de la calidad de las proposiciones.

Axiomas de cantidad

1. El término medio debe estar distribuido por lo menos una vez.
2. En la conclusión no puede figurar ningún término distribuido que no lo esté en las premisas.

Axiomas de calidad

1. De dos premisas negativas no se obtiene ninguna conclusión.
2. Si una premisa es negativa, la conclusión debe ser negativa.
3. Si ninguna de las premisas es negativa, la conclusión debe ser afirmativa.

3.4.5 Los teoremas generales del silogismo

Teorema I. El número de términos distribuidos en la conclusión debe ser *menor* por lo menos en una unidad al número total de términos distribuidos en las premisas.

Teorema II. De dos premisas particulares no se desprende ninguna conclusión. Dos premisas particulares pueden ser: a) ambas negativas, b) ambas afirmativas, c) una afirmativa y una negativa.

Teorema III. Si una premisa es particular, la conclusión debe ser particular.

Teorema IV. Si la premisa mayor es una proposición particular afirmativa y la menor una proposición universal negativa, no puede haber conclusión.¹⁰

3.5 ANALOGÍA

Las concepciones y papel de la analogía han variado en la historia de la filosofía. Mientras que para algunos pensadores, como Platón o Santo Tomás, la analogía presenta un tipo de razonamiento específico e indispensable, para otros como los empiristas, ella se limita a afirmar una semejanza bastante débil y sirve para la invención de hipótesis, pero debe ser eliminada en la formulación de los resultados de la investigación científica.

¹⁰ Nagel, M. C. (s.f.). Introducción a la lógica y al método científico. II Lógica aplicada y método científico. Buenos Aires: Amorrortu editores.

Un teórico de la argumentación constatará que el recurso a la analogía constituye una de las características de la comunicación y del razonamiento no formales, que en algunos casos la analogía podrá ser eliminada en fin de cuentas cuando la conclusión a la cual se llega se resume en una fórmula matemática, pero que muy a menudo sobre todo, ella se encuentra en el centro de una visión original, sea del universo, sea de las relaciones entre el hombre y la divinidad.

Pase lo que pase, para conservarla en su especificidad, será preciso interpretar la analogía en función de su sentido etimológico de proporción. Ella difiere de la proporción puramente matemática, en tanto que no plantea la igualdad de dos relaciones, sino que afirma una semejanza de relaciones. En la analogía se afirma que *a* es a *b* como *c* es a *d*, se trata de una relación cualquiera que se asimila a otra relación. Entre la pareja a-b, el tema de la analogía y la pareja c-d el foro de la analogía, no se afirma una igualdad simétrica por definición, sino una asimilación que tiene por fin aclarar, estructurar y evaluar el tema gracias a lo que se sabe del foro, lo que implica que el foro proviene de un dominio heterogéneo, puesto que es mejor conocido que el del tema.

La analogía de cuatro términos puede expresarse por medio de tres, uno de ellos puede ser repetido en el tema y en el foro; su esquema será: *b* es a *c* como *a* es a *b*.¹¹

Para hablar de analogía, primero se debe distinguir entre identidad, igualdad y semejanza. Hablando en sentido estricto, la identidad es la unidad en la sustancia; la igualdad, la unidad en la cantidad y la semejanza, la unidad en la cualidad. La analogía es una semejanza en sentido estricto, es decir, una semejanza imperfecta, que no llega a la igualdad, y por eso contiene también desemejanzas y diferencias.¹²

La analogía propiamente hablando es una forma de predicación. Y la predicación de un nombre común a los varios sujetos a que se atribuye puede realizarse de tres maneras:

- Tomando el nombre con la misma significación en todos los casos, es decir, con significaciones perfectamente semejantes o iguales (*predicación unívoca*)

¹¹Perelman, C. (1997). El imperio retórico. Retorica y argumentacion. Editorial Norma S.A.

¹²López, J. G. (s.f.). Obtenido de <http://dspace.unav.es/dspace/bitstream/10171/1882/1/05.%20Jes%20C3%BA%20GARC%20C3%8DA%20L%20C3%93PEZ,%20La%20analog%C3%ADa%20en%20general.pdf>

- Tomando el nombre con una significación completamente diversa en cada caso, con significaciones que son enteramente desemejantes o diferentes (*predicación equívoca*)
- Tomando el nombre con una significación en parte semejante y en parte desemejante en cada caso, o sea, con significaciones que son simplemente semejantes, pero no iguales, y por eso entrañan también desemejanzas o diferencias (*predicación análoga*).

La analogía es, pues, intermedia entre la univocidad y la equivocidad, y por eso participa en cierto modo de esos dos extremos entre los que está colocada. Así, conviene con la univocidad en la unidad del nombre y en la semejanza de las significaciones ligadas a ese nombre, pero difiere de ella en que no se trata de una semejanza perfecta (igualdad), sino de una semejanza imperfecta (con desigualdad). Por otro lado, conviene con la equivocidad en la unidad del nombre y en la desemejanza de las significaciones ligadas a él, pero difiere de la misma en que no se trata de una desemejanza completa, de una diversidad total, sino de una desemejanza parcial, de una desemejanza semejante.¹³

Por lo dicho hasta ahora se ve que la analogía se centra en la significación de los nombres, de ciertos nombres. Pero la significación de un nombre se puede tomar de dos maneras:

- **La analogía simple**, que se conoce con el nombre de *analogía de atribución*, se da cuando se compara un término con otro, una forma con otra, es decir se trata de semejanza de formas.
- **La analogía compuesta**, que recibe el nombre de *analogía de proporcionalidad*, se da cuando se compara una relación entre dos términos o formas con otra relación semejante. Se trata de la semejanza de dos o más relaciones o proporciones. Por consiguiente esta analogía exige al menos cuatro términos comparados dos a dos.

3.5.1 El fundamento de la analogía

En sentido real y en sentido lógico. En efecto, todo nombre apunta a un concepto en lo cual consiste su significación, pero el concepto es doble: formal y objetivo.

¹³ López, J. G. (s.f.). Obtenido de

<http://dspace.unav.es/dspace/bitstream/10171/1882/1/05.%20Jes%C3%BAs%20GARC%C3%8DA%20L%C3%93PEZ,%20La%20analog%C3%ADa%20en%20general.pdf>

Formal: La misma representación mental que formamos con nuestro acto de entender.

Objetivo: Es algo real. Una esencia o una forma que existe en la realidad y en tanto que existe en ella, a la cual, le acontece el ser concebida o ser representada en un concepto formal.

3.5.2 Estructura de la analogía

Se considera que las partes que constituyen la analogía son el análogo, la trama o relación analógica y el tópico. El análogo es núcleo central de la analogía que representa el mensaje, el conocimiento ya conocido. La trama o relación analógica es el conjunto de relaciones que se establecen para comparar características semejantes de determinadas partes del análogo y del tópico, y el tópico está formado por los contenidos conceptuales, procedimentales o actitudinales desconocidos que se pretenden enseñar, es decir, por el conocimiento o conjunto de conocimientos de la materia en estudio.¹⁴

Figura 1. Estructura de una Analogía



Fuente: Estructura de las analogías y su uso didáctico

La analogía consiste en establecer un conjunto de relaciones para comparar características semejantes de determinadas partes de la representación de las estructuras del análogo y el tópico, y permitir mediante dicha comparación la comprensión del tópico.¹⁵

3.6 FALACIA

Las falacias se definen como errores de razonamiento. En general, las falacias se clasifican en: Puramente lógicas o formales, Semilógicas o verbales y Materiales.

¹⁴González, T. M. (s.f.). *ESTRUCTURAS DE LAS ANALOGÍAS Y SU USO DIDÁCTICO*. Obtenido de <http://www.grupoblascabrera.org/didactica/pdf/Analogias%20y%20uso%20didactico.pdf>

¹⁵DIDÁCTICO, T. M. (s.f.). Obtenido de <http://www.grupoblascabrera.org/didactica/pdf/Analogias%20y%20uso%20didactico.pdf>

3.6.1 Falacias formales

Se trata de razonamientos que no se ajustan a los tipos de inferencia válida, en relación con los diversos test o reglas que diferencian las formas válidas de razonamiento de las que no lo son. Cuando se les da forma hipotética, todas las falacias resultan razonamientos que parecen de la afirmación del consecuente o de la negación del antecedente o bien que afirman una implicación o conexión lógica donde no la hay. Ejemplo de esto último es el silogismo en el que el término medio no está distribuido. Esto lo reduce a un razonamiento con cuatro términos cuyas premisas no brindan ninguna base o prueba para afirmar la conclusión.

3.6.2 Falacias semilógicas o verbales

Se trata de formas aparente válidas de inferencia, cuyo cuidadoso examen permite advertir que no lo son. Tal apariencia de validez obedece a una ambigüedad, esto es, al uso de la misma palabra o signo verbal para representar dos conceptos diferentes.

El razonamiento parece ser de la forma: A es B, y B es C, por lo tanto, A es C; pero en realidad, es de la forma siguiente: A es B, y D es C, por lo tanto A es C.

Conviene observar que no todo ejemplo de ambigüedad es una falacia. Si consultamos cualquier diccionario erudito de la lengua española, veremos que hay pocas palabras con un significado único. Hay una sola manera de escribir “si”, pero muchas maneras de decirlo, cada una de las cuales presenta matices diferentes de significado. La falacia solo aparece cuando se afirma que ciertas premisas exigen determinada conclusión y esta afirmación resulta falsa a causa de la ausencia de una conexión real entre ellas, ausencia encubierta por el uso de la misma palabra para referirse a dos cosas distintas.

3.6.3 Falacias materiales

Según el uso popular, se dice que un argumento es falaz o que “contiene una falacia en alguna parte” si conduce a conclusión falsa. Aceptando que la lógica no se identifica con todo conocimiento y no puede garantizar la verdad material de todas las conclusiones, afirmar que solo ella sea capaz de indicar qué conclusiones son, de hecho, falsas, resulta inadmisibles. Una conclusión puede ser de hecho falsa, y sin embargo ser absolutamente correcto el razonamiento por el cual se la dedujo (a partir de una premisa falsa). Pero cuando deducimos una

proposición de otra proposición falsa no conseguimos probar, su verdad material. Podemos designar como falacias materiales a las pruebas falsas o ilusorias.

Si el uso de A que sigue al suceso B es o no causado por éste, es una cuestión de hecho y no de lógica; pero la suposición de todo lo que sigue a un suceso es causado por éste, es falsa, y los razonamientos basados en ella carecen de valor. Es también falacia la pretensión de haber probado una proposición introducida de contrabando en las premisas, de una forma más disimulada (esto recibe el nombre de petición de principio).

Hay otras formas de falacias que simplemente mencionaremos como el *argumento circular*: consiste en introducir en las premisas una proposición que depende de aquella que se discute. *La falacia de la pregunta falsa, también llamada falacia de la pregunta múltiple*: en la medida en que toda pregunta solicita una información, en lugar de darla, no es una proposición y no puede ser verdadera ni falsa. *La falacia del argumentum ad hominem*: un recurso muy antiguo pero aun difundido, para negar la fuerza lógica de un argumento (y de este modo crear la apariencia de que se prueba lo opuesto), es injuriar a quien lo expone.¹⁶

3.7 SÍMIL

Símil es un término con origen en el vocablo latino *similis* que hace referencia a lo semejante¹⁷. El concepto, por lo tanto, se utiliza para establecer una comparación entre dos cosas. Símil es una figura retórica que consiste en la comparación expresa entre una cosa y otra, para dar una idea eficaz de una de ellas. Al establecer la comparación por semejanza, se trasladan las características simbólicas o físicas de uno a otro. Los símiles apelan a elementos de relación como “que”, “como” o “cual”.

3.7.1 Uso de los símiles

Es una figura muy común en la literatura, en la poesía o el teatro. Es directa y explícitamente expresa el vínculo simbólico entre las dos realidades comparadas, utilizando un término de comparación, algo que hace que se destaque de la analogía de otra figura con la que a menudo se confunden. Es un recurso preferido por el lenguaje poético, muy utilizada por burla o ironía, también se

¹⁶ Nagel, M. C. (s.f.). Introducción a la lógica y al método científico. I Lógica formal. Buenos Aires: Amorrortu editores.

¹⁷ Negroponte, N (1995) El Mundo digital. Barcelona: Ediciones B.

puede avanzar en la discusión y dar a ver la realidad difícil de definir que no sea a través de las imágenes.

Pertenece a la clase de "figuras de la semejanza". Es de suma importancia en el lenguaje, tanto en su frecuencia por su función, opera una reconciliación inesperada e innecesaria entre dos realidades, a priori desconocidos entre sí, pero con una relación de semejanza y contigüidad semántica. "Enfatizar las comparaciones y semejanzas entre las cosas, pero no cambian el significado de las palabras". Siendo así una unidad fuertemente incrustado en su sintaxis y el contexto del discurso.

En tanto, los símiles pueden ser:

- Reversibles (se disponen los dos términos sucesivamente en diferente orden).
- Graduados (inferioridad, igualdad o superioridad).¹⁸

3.8 RAZONAMIENTO LÓGICO

El razonamiento es un proceso mediante el cual se obtienen conclusiones a partir de hechos, creencias y normas. El razonamiento es una habilidad del pensamiento por lo que también se llama *raciocinio*.

Se expresa en la argumentación o conjunto de afirmaciones relacionadas de manera tal que uno de ellos, llamado conclusión, se infiere del o los otros, llamados premisas. El término razonamiento es el punto de separación entre el instinto y el pensamiento, el instinto es la reacción de cualquier ser vivo. Por otro lado el razonar nos hace analizar y desarrollar un criterio propio, el razonar es a su vez la separación entre un ser vivo y el hombre.

El razonamiento es una inferencia de una proposición o juicio a otra. Entendemos por inferencia un paso del pensamiento, un paso mental. Todo razonamiento puede revestir validez en la medida que se pueda suponer que se refiere a un mismo objeto real, por eso es que cada razonamiento sólo puede referirse a un solo objeto. Este objeto, en el curso del razonamiento, formalmente considerado, no está explícitamente mencionado, por lo que se le conoce como "objeto supuesto".¹⁹

El razonamiento lógico o causal es una operación lógica mediante la cual, partiendo de uno o más juicios, se deriva la validez, la posibilidad o la falsedad de otro juicio distinto. El estudio de los argumentos corresponde a la lógica, de modo que a ella también le corresponde indirectamente el estudio del razonamiento. Por lo general, los juicios en que se basa un razonamiento expresan conocimientos ya adquiridos o, por lo menos, postulados como hipótesis.

¹⁸ Florencia. (s.f.). *Definición de símil*. Obtenido de <http://www.definicionabc.com/comunicacion/simil.php>

¹⁹ León, C. (Junio de 2011). *Tipos de Razonamiento*.

Es posible distinguir varios tipos de razonamiento lógico. Por ejemplo el razonamiento deductivo (estrictamente lógico), el razonamiento inductivo (donde interviene la probabilidad y la formulación de conjeturas).

La conclusión puede no ser una consecuencia lógica de las premisas y aun así dar lugar a un razonamiento, ya que un mal razonamiento aún es un razonamiento (en sentido amplio, no en el sentido de la lógica). Los razonamientos pueden ser válidos (correctos) o no válidos (incorrectos). Los razonamientos no válidos que, sin embargo, parecen serlo, se denominan falacias.²⁰

3.8.1 Razonamiento deductivo

Un razonamiento es deductivo si a partir de premisas verdaderas su conclusión es necesariamente verdadera. Entonces se afirma que la conclusión es consecuencia lógica de las premisas o que las premisas implican la conclusión. Los razonamientos deductivos tienen la propiedad de transmisión o preservación de la verdad porque si las premisas son verdaderas se asegura que la conclusión también lo es.

El razonamiento deductivo se mueve de lo general a lo particular. Toma una premisa general y deduce conclusiones particulares. Un argumento deductivo válido es aquel en el que la conclusión necesariamente se deriva de la premisa. Puede ser que la premisa no sea verdadera pero, no obstante, la forma del argumento es válida.

Todo en la conclusión de un argumento deductivo válido debe también estar contenido en las premisas. La verdad (o veracidad) de la conclusión de un argumento deductivo depende de dos cosas: la condición de correcta (o validez) de la forma del argumento, y la verdad (o veracidad) de la premisa. La validez de la forma está determinada por la aplicación de las reglas establecidas. Así que, la única debilidad de un argumento deductivo es el verdadero valor de sus premisas. Sus conclusiones son únicamente tan buenas como sus premisas. O, para decirlo de otra manera, sus presuposiciones siempre determinarán sus conclusiones.

El pensamiento deductivo parte de categorías generales para hacer afirmaciones sobre casos particulares. Va de lo general a lo particular. Es una forma de razonamiento donde se infiere una conclusión a partir de una o varias premisas. El filósofo griego Aristóteles, con el fin de reflejar el pensamiento racional, fue el primero en establecer los principios formales del razonamiento deductivo.

²⁰ León, C. (Junio de 2011). *Tipos de Razonamiento* .

Existen dos formas básicas del razonamiento deductivo:

- a) El inmediato, que se da cuando la única operación lógica que se realiza es la modificación de un juicio
- b) El mediato, que se da cuando es necesario realizar una relación de mediación entre dos o más juicios para obtener una conclusión.²¹

²¹León, C. (Junio de 2011). *Tipos de Razonamiento* .

4. ESTADO DEL ARTE

Entre los diferentes software que se pueden encontrar para el aprendizaje de lógica de primer orden y figuras retóricas se encuentran:

- AProS. Tiene como funciones la posibilidad de deducciones naturales de proposiciones y predicados lógicos y se encuentra disponible en <http://www.phil.cmu.edu/projects/apros/>
- Bertie3. Este software es usado para la deducción proposicional y de predicados lógicos. Se encuentra disponible en <http://selfpace.uconn.edu/BertieTootie/software.htm>
- Expression Evaluator. Usando este software se puede realizar una interpretación semántica que evalúa fórmulas de predicados lógicos. Se puede encontrar en <http://www.cc.utah.edu/~nahaj/logic/evaluate/>
- Gateway tologic. Tiene la función de realizar deducciones naturales de predicados lógicos y se encuentra disponible en <http://logik.phl.univie.ac.at/~chris/formular-uk.html>
- Computerleerprogramma lógica (Logic tutor). Es utilizado como deductor natural de proposiciones y predicados lógicos. Y el archivo descargable se encuentra en Descargable de <ftp://ftp.th.vu.nl/pub/docs/Logica/LogFile.exe>
- InferenceEngine. Este es un motor de inferencia lógica que se puede encontrar vía web para su posterior descarga en <http://blue.butler.edu/~sglennan/InferenceEngine.html>
- Jape. La función de este software es la deducción natural de predicados lógicos y se puede encontrar en <http://www.jape.org.uk/>
- LogicAnimations. Su función es realizar computación semántica en proposiciones, lógica, dinámica y lógica modal. Se puede encontrar en <http://staff.science.uva.nl/~jaspars/animations/>
- LogicDaemon. Por medio del uso de este software se pretende realizar deducciones naturales de predicados lógicos, tablas de verdad y proposiciones lógicas. Se encuentra disponible en <http://logic.tamu.edu/>
- LogicforFun. Con esta plataforma web se busca enseñar lógica a través de rompecabezas y acertijos. Se puede encontrar en <http://logic4fun.rsise.anu.edu.au/>
- LogicCoach III. Por medio de este software se puede trabajar con Tablas de verdad y lógica de primer orden, se puede encontrar en <http://academic.csuohio.edu/polen/>

- LogicsWorkbench. En este software se realizan prueba de proposiciones lógicas y se puede encontrar en la dirección <http://www.lwb.unibe.ch/>
- Tableau III. Este software es usado para deducción natural de predicados lógicos y encuentra disponible en <http://logic.philosophy.ox.ac.uk/>
- Asistente para la Deducción Natural (ADN). Esta herramienta tiene como función asistir al alumno que se inicia en la técnica de deducción natural a elaborar sus propias deducciones y se encuentra disponible en <http://www.dccia.ua.es/logica/ADN/index.htm>

5. ESTRUCTURA LÓGICA

En el siguiente apartado se encuentran descritos las clases, las funciones y los archivos que corresponden al motor de inferencia de la aplicación a la cual se le realiza el proceso de reingeniería. Esta información se encuentra en dos carpetas (components e includes), en la carpeta components, se encuentran los archivos falacias.php, analogías.php, simil.php, razonamiento.php, silogismos.php, donde se administran los ejercicios de cada figura. En la carpeta includes se encuentran los archivos analogías.php, falacias.php, razonamiento.php, maestro.php, simil.php y silogismos.php, donde se especifica el funcionamiento del desarrollo de los ejercicios dispuestos para cada una de las figuras. También se muestra el modelo entidad-relación de la base de datos que se utiliza en el funcionamiento de la aplicación inicial a la cual se le incluye modificaciones para el funcionamiento de la aplicación propuesta. Por último, se encuentran los diagramas de flujo que explican el funcionamiento de cada figura retórica (analogías, razonamiento lógico, símiles, falacias y silogismos).

5.1 LISTA DE CLASES

En este apartado se encuentra la lista de las clases correspondientes a la estructura del motor de inferencia usado en la aplicación inicial. Estas clases se encuentran definidas dentro de los archivos que se están contenidos en la carpeta includes (analogías.php, falacias.php, razonamiento.php, simil.php y silogismo.php).

La sintaxis de la explicación de las clases es: nombre de la clase, archivo donde se encuentra la clase y la explicación correspondiente

- **classmosejemploextendsmosDBTable** (silogismos.php): Esta clase se extiende de mosDBTable, y en ella se encuentra toda la especificación para poder trabajar con los ejemplos de los silogismos; dentro de los atributos definidos se encuentran el id del ejemplo, que especifica el identificador del ejemplo, el nombre del ejemplo, el nivel en el cual se encuentra ubicado y el identificador de la figura, que relaciona a un ejemplo con una figura específica, en este caso, los silogismos.

Esta clase contiene las siguientes funciones:

- `mosejemplos`. Realiza la conexión con la base de datos utilizando la id del ejemplo.
 - `averiguarejemplo`. Permite realizar la búsqueda de un ejemplo basándose en su nombre.
 - `asignarejemplo`. Esta función permite obtener un ejemplo aleatorio basándose en el id del curso, mientras que no se pida un ejemplo en particular.
 - `loadfromvector`. Carga en un vector los atributos (id, nombre, texto, usuario, estado, nivel e idfigura) de un ejemplo.
 - `update`. Mediante el id del ejemplo, se procede a actualizar la información del ejemplo.
 - `insertexample`. Con esta función se realiza la inserción de ejemplos en la base de datos, comprobando mediante el id del ejemplo, que no se encuentre previamente registrado.
 - `insert`. Se realiza la inserción de una nueva fila de ejemplos en la base de datos.
 - `borrarejemplo`. Con esta función se borra un ejemplo de la base de datos, utilizando el id del ejemplo como identificador de cual ejemplo se quiere borrar.
 - `mostrarterminossegunejemplo`. Utilizando el id del ejemplo, se muestran los términos que el ejemplo contenga, mostrando el id, el nombre y el padre de los términos.
 - `objetossegunejemplo`. Mediante el id del ejemplo, se muestra una lista de los objetos que contiene dicho ejemplo, mostrando el id del objeto, la descripción del objeto y el ejemplo al que pertenece.
 - `asignarobjetoaunejemplo`. Utilizando esta función se le pueden asignar objetos a un ejemplo que es identificado por el id del ejemplo y los objetos son seleccionados de una lista que muestra los objetos disponibles.
 - `mostrarplantillaincluirterminos`. Con esta función, se muestra la plantilla que se usa para la inclusión de términos en ejemplos que es diferenciado con el id del ejemplo.
- **`classmosterminosextendsmosDBTable`** (`silogismos.php`): Esta clase se extiende de `mosDBTable` y contiene dentro de los atributos el id del término, que se usa para identificar cada término guardado en la base de datos, el id del padre, que identifica quien es el padre de cada término, el nombre del término, el género y el id del ejemplo, utilizando para asociar cada término a determinado ejemplo.

Esta clase contiene las siguientes funciones:

- mosterminos. Esta función se utiliza para realizar la conexión con la base de datos, utilizando el id del término.
 - buscarid. Se realiza la búsqueda del id de un término, utilizando como base de búsqueda el id del ejemplo.
 - loadfromvector. Carga en un vector los atributos (id, idpadre, nombre, género y el id del ejemplo) de un término.
 - insert. Se realiza la inserción de una nueva fila de términos en la base de datos.
 - borrardeejemplo. Con esta función se realiza la eliminación de un término usando como base el id del ejemplo al que pertenece el término que se quiere eliminar.
- **classmospremisasextendsmosDBTable** (silogismos.php): Esta clase se extiende de mosDBTable; dentro de sus atributos se encuentran, el id de la premisa, el id del ejemplo al que pertenece la premisa, el cuantificado de la premisa, el sujeto, la copula, el predicado de la premisa, la premisa, la clase, la cantidad y la cualidad.

Esta clase contiene las siguientes funciones:

- mospremisas. Esta función se utiliza para realizar la conexión con la base de datos, utilizando el id de la premisa.
- guardarpremise. Con esta función se guarda una premisa en la base de datos dependiendo del caso en el que la estructura de la premisa se ajuste, usando el id del ejemplo al que pertenece la premisa, el sujeto, el predicado y el género.
- verify. Se realiza la verificación que la premisa que se quiere guardar en la base de datos no se encuentre ya en ella, utilizando como base de búsqueda el id del ejemplo y el id de la premisa.
- buscarpadre. Se realiza la búsqueda del padre de un término, usando como parámetro de búsqueda el id del padre y se muestra el resultado en un lista.
- buscarhijo. Se usa esta función para realizar la búsqueda de los hijos de un término; usando una búsqueda parecida a la búsqueda del padre, usando como parámetro base el id del padre.
- buscarobrino. Mediante el id del padre de un término, se realiza la búsqueda recursiva de los sobrinos de ese término; buscando primero los hermanos del término y luego se buscan los tíos del término.

- `validarpremise`. Utilizando el id del ejemplo, se realiza la búsqueda de una premisa que pertenezca a ese ejemplo y se valida la premisa.
 - `verpremisas`. Con esta función se pueden ver las premisas que pertenezcan a un determinado ejemplo, realizando una búsqueda de las premisas, usando como base el id del ejemplo.
 - `estandarizarpremise`. De una cadena de premisa, quita los espacios que puedan haber entre palabras para poder estandarizar la premisa.
 - `validarsintaxispremise`. Realiza la validación de la sintaxis que trae una premisa, en cuanto a cuantificador, sujeto, copula y predicado.
 - `armarpremisedeunejemplo`. Utilizando el id de un ejemplo, se buscan los términos que pertenezcan a ese ejemplo y posteriormente, se arma una premisa utilizando estos términos.
 - `loadfromvector`. Carga en un vector los atributos (id, idejemplo, cuantificador, sujeto, copula, premisa y clase) de una premisa.
 - `insert`. Se realiza la inserción de una nueva fila de premisas en la base de datos.
 - `update`. Mediante del id de la premisa, se procede a actualizar la información de una premisa.
 - `borrardesdeejemplo`. Con el id del ejemplo, se realiza la búsqueda de una premisa y se elimina de la base de datos.
- **classmossilogismos** (silogismos.php): En esta clase se encuentran los atributos de los silogismos, dentro de los que se encuentran, mayor, que corresponde a la premisa mayor; menor, que corresponde a la premisa menor; conclusión, que corresponde a la conclusión del silogismo; figura, que especifica a que figura pertenece, en este caso, a los silogismos; modo, que contiene la combinación de los juicios de calidad; medio, que contiene el término medio; estado, que corresponde al estado del silogismo; estadoarmar y el mensaje.

Esta clase contiene las siguientes funciones:

- `mossilogismos`. Con los parámetros de la premisa mayor, la premisa menor y la conclusión se realiza la conexión a la base de datos.
- `armarsilogismodeunejemplo`. Utilizando el id del ejemplo, se realiza la búsqueda de las premisas que pertenezcan al ejemplo y luego, se halla, la premisa mayor, la premisa menor y la conclusión, para por último, armar el silogismo con su estructura base.

- `escribirsilogismo`. Con esta función se muestra la estructura para ingresar un silogismo a la página, con sus componentes (premisa mayor, premisa menor, conclusión, modo, figura, término medio y términos).
- `almacenaractividad`. Con la información ingresada en la función anterior, se procede a almacenar dicha información en la base de datos.
- `validarsilogismo`. Utilizando el identificador de un silogismo, se realiza la validación de los componentes del silogismo, utilizando la función `validarregla1`.
- `validarsilogismosinconclusion`. Usando esta función se obtiene la conclusión de un silogismo que no la posee; utilizando la función `validarregla1`, se usan los componentes del silogismo (términos, premisa mayor, premisa menor, sujeto, predicado, cuantificador y copula), para poder tanto hallar como completar la conclusión.
- `hallarmodo`. Se realiza la búsqueda del modo, usando como parámetros base la figura y los tres términos del silogismo (premisa mayor, premisa menor y conclusión).
- `validarregla1`. Con esta función se valida la primera regla para silogismos, donde se comprueba que el silogismo tenga tres términos (premisa mayor, menor y conclusión) y que los términos mayor y menor se relacionen con el término del medio.
- `validarregla2`. Se valida la regla 2 para los silogismos y comprobar que los términos no pueden tener mayor distribución en la conclusión que en las premisas.
- `validarregla3`. Se realiza la validación de la regla 3 para los silogismos, donde se comprueba que el término medio no aparezca en la conclusión.
- `validarregla4`. Se comprueba la regla 4 para silogismos, donde el término medio debe tomarse por lo menos una vez en toda su extensión.
- `validarregla5`. Mediante esta función se valida la regla 5 para silogismos, que dice, que de premisas negativas no se puede concluir nada.
- `validarregla6`. Con esta función, se valida la regla 6 para silogismos, centrándose en la conclusión que es la parte más débil, ya que, la conclusión puede ser negativa si alguna premisa es negativa, o puede ser particular, si una premisa es particular.
- `validarregla7`. Usando esta función se valida la regla 7 para los silogismos, comprobando que de premisas particulares no se puede llegar a ninguna conclusión.
- `validarregla8`. Se valida la regla 8 para silogismos, que afirma que, de premisas afirmativas no puede inferir una conclusión negativa.

- validarreglas. Con esta función, se realiza la validación de las 8 reglas para silogismos, dejando la regla 1 para validar al final, usando en todos los casos como base un silogismo en particular.
 - insert. Se realiza la inserción de una nueva fila de silogismos en la base de datos.
- **classmosanalogias** (analogias.php): Esta clase permite interactuar con las analogías que se encuentren en la base de datos, definiendo los atributos que tiene una analogía (analogía, que define la analogía completa para utilizar, el término a, el término b, el término c, el término d, idA, se refiere al identificador del término A, idB, se refiere al identificador del término B, idC, se refiere al identificador del término C, idD, se refiere al identificador del término D, pA, que se refiere al identificador del prefijo A, pB, que se refiere al identificador del prefijo B, pC, que se refiere al identificador del prefijo C, pD, que se refiere al identificador del prefijo D, idrelacion, donde se especifica el identificador de la relación, el estado y el mensaje.

Esta clase incluye las siguientes funciones:

- mostrarrelaciones. Con esta función se muestran las relaciones existentes, usando el id del ejemplo como parámetro de búsqueda, mostrando una lista de la relación de los objetos con los ejemplos.
 - mostrarparejas. Muestra las parejas que se están definidas entre objetos, relacionando también cada objeto relacionado con un término de la base de datos.
 - validarparejas. Esta función se utiliza para validar las parejas que tienen una estructura de dos prefijos y el identificador de la relación entre ellos.
 - armaranalogiadeunejemplo. Por medio de esta función se arma una analogía que corresponda a un ejemplo, utilizando el id del ejemplo como identificador base para encontrar primero los objetos y la relación entre ellos y para posteriormente formar la analogía.
- **classmosfalacias** (falacias.php): En esta clase se especifican las acciones que se pueden realizar para trabajar con las falacias. Esta clase tiene como atributos iniciales, la premisa mayor, la premisa menor, la conclusión, la especificación de la figura, el modo, el medio, el estado y el mensaje de retroalimentación.

Se encuentran definidas también las siguientes funciones:

- `mosfalacias`. Con esta función y utilizando la premisa mayor, la premisa menor y la conclusión, se realiza la conexión con la base de datos para poder utilizar toda la información que allí se encuentra almacenada.
- `armarfalaciadeunejemplo`. Se realiza la formación de una falacia usando como base el id del ejemplo y se procede a hallar las dos premisas, la figura, para determinar cómo se ubicará el término medio y hallar la conclusión para que por último solo sea armar la falacia que corresponda a dicho ejemplo base.
- `escribiralacia`. Se muestra la estructura que se utiliza para poder ingresar por medio del teclado los componentes de una falacia (premise mayor, premise menor, conclusión, modo, figura, término medio y los términos que se usaran en la falacia).
- `almacenaracidad`. Con la información ingresada en la función anterior (`escribiralacia`), se procede a almacenar dicha información en la base de datos.
- `validaralacia_BASICA`. Esta función se utiliza para validar una falacia de una forma básica, teniendo como base la falacia que se quiere validar y utilizando la función `validarregla1`, comprobando todos los componentes de la falacia y mostrando si su estructura es válida o inválida.
- `validaralacia`. Con esta función se realiza la validación de una falacia, utilizando como base la falacia a validar y comprobando que sus componentes estén correctamente relacionados utilizando como base la función `validarregla1` y por último se realiza la creación de la conclusión de la falacia.
- `hallarmodo`. Se realiza la búsqueda del modo, usando como base la figura y una falacia.
- `validarregla1`. Por medio de esta función, se realiza la validación de la primer regla para las falacias, que consiste en que, una falacia debe tener tres términos (mayor, menor y medio) y que los términos mayor y menor deben estar relacionados a través del término medio.
- `validarregla2`. Se válida la regla 2 para las falacias y comprobar que los términos no pueden tener mayor distribución en la conclusión que en las premisas.
- `validarregla3`. Se realiza la validación de la regla 3 para las falacias, donde se comprueba que el término medio no aparezca en la conclusión.
- `validarregla4`. Se comprueba la regla 4 para las falacias, donde el término medio debe tomarse por lo menos una vez en toda su extensión.

- validarregla5. Mediante esta función se valida la regla 5 para falacias, que dice, que de premisas negativas no se puede concluir nada.
 - validarregla6. Con esta función, se valida la regla 6 para falacias, centrándose en la conclusión que es la parte más débil, ya que, la conclusión puede ser negativa si alguna premisa es negativa, o puede ser particular, si una premisa es particular.
 - validarregla7. Usando esta función se valida la regla 7 para falacias, comprobando que de premisas particulares no se puede llegar a ninguna conclusión.
 - validarregla8. Se valida la regla 8 para falacias, que afirma que, de premisas afirmativas no puede inferir una conclusión negativa.
 - validarreglas. Con esta función, se realiza la validación de las 8 reglas para falacias, dejando la regla 1 para validar al final, usando en todos los casos como base una falacia en particular, que en este caso, es la falacia a la que se le desea validar todos sus componentes y su estructura.
 - insert. Se realiza la inserción de una nueva fila de falacias en la base de datos.
- **classmossimil** (simil.php): En esta clase se encuentran especificados todos los procesos que se pueden realizar con los símiles y que son usados para el uso de los ejercicios que están propuestos para ser desarrollados por los usuarios. Tiene como atributos base, el símil que contiene toda la información del mismo, el término a y el término b, los identificadores tanto del término a como del término b, los prefijos a y b, el identificador de la propiedad, la propiedad, el estado y un mensaje de retroalimentación.

En esta clase se encuentran definidas las siguientes funciones:

- mostrarpropiedades. Utilizando el id de un ejemplo, se realiza una búsqueda de los objetos de ese ejemplo que se encuentran relacionados, mostrando como resultado la propiedad que comparten.
- mostrarterminos. Con esta función, se muestran los términos, usando como base el término b y mostrando aquellos objetos que tengan propiedades.
- validarterminos. Se realiza la validación de los términos que componen una estructura de dos términos y una conclusión.
- armarsimildeunejemplo. Usando el id de un ejemplo como base, se realiza la formación de un símil para asignarlo a dicho ejemplo, escogiendo la propiedad que habrá entre objetos, así como los valores de cada uno de

los componentes del símil, para al final solo realizar la formación del símil para dicho ejemplo.

- **parejadelobjeto.** Por medio de esta función y usando como base el id de un objeto se realiza la búsqueda de otro objeto que comparta una propiedad.

5.2 LISTA DE ARCHIVOS

En este apartado se encuentran los archivos que se usan en la aplicación inicial y que están directamente relacionados con el funcionamiento de las figuras retóricas, estos archivos se encuentran en dos carpetas (components e includes), donde en la carpeta components se encuentra el código fuente de la administración de los ejercicios de cada figura retórica, y en la carpeta includes, se encuentra el código fuente para el desarrollo de los ejercicios de cada figura retórica

5.2.1 Lista de archivos de la carpeta components:

- **falacias.php:** Contiene las funciones y clases para el manejo de falacias, dentro de esta información se encuentra el id del ejercicio, el nivel y el id del ejemplo, que los parámetros utilizados en las funciones descritas más adelante.
- **analogias.php** Contiene diferentes campos y funciones para poder trabajar con las analogías, entre los campos definidos se encuentra, el id del ejemplo, el ejercicio, nivel, curso, entre otros y dentro de las funciones se encuentran, generar ejercicio y validar ejercicio definiendo las especificaciones para poder realizar las funciones mencionadas.
- **razonamiento.php:** Contiene parámetros y funciones que se pueden aplicar a cualquiera de las figuras retóricas trabajadas, y realiza la función de comparar la información ingresada por el usuario con la respuesta contenida en la base de datos de acuerdo al ejercicio y al nivel escogido y analiza si esta comparación es la respuesta correcta o no.
- **simil.php:** Dentro de este archivo se encuentran las funciones y clases para el manejo y aplicación de los símiles, con los parámetros que se deben utilizar para realizar funciones como generarejercicio o validarejercicio entre otras funciones, a su vez se especifica los diferentes tipos de casos que se pueden presentar y como es manejada la información que se ingresa por parte del usuario.

- **silogismos.php**: Su estructura especifica las funciones, clases y parámetros necesarios para trabajar ejercicios relacionados con silogismos; las principales funciones que se tiene en este archivo son generarejercicio donde se tienen en cuenta parámetros como nivel, id del ejemplo y el ejercicio, para que el usuario pueda ingresar la información relacionada con el mismo y la función validarejercicio devuelva una respuesta si es correcto o no.
- **ejercicios.php**: Dentro de este se encuentran los parámetros para poder especificar qué tipo de ejercicios están disponibles para el usuario, dependiendo de diferentes parámetros que el mismo usuario debe especificar, dentro de los casos disponibles para la generación de ejercicios se encuentra "curso", "practica" y "ejecutar", donde para cada uno se genera un tipo de figura retórica y un ejercicio de uso.

5.2.3 Lista de archivos de la carpeta includes:

- **razonamiento.php**: Los razonamientos que se utilizarán tendrán la siguiente estructura: razonamiento de cuatro términos A, B, C, D donde se expresan A es a B como C es a D, los términos arman el razonamiento mediante propiedad de los objetos. La propiedad de mundos para buscar razonamiento, puede ser implementada.
- **silogismos.php**: El software está estructurado de la siguiente manera: Armar premisas dado un conjunto de términos, generar todas las premisas validas posibles por una única vez y las almacenan en la base de datos para armar y validar Silogismos.
- **analogias.php**: El software está estructurado de la siguiente manera: Analogía de cuatro términos A, B, C, D donde se expresa A es a B como C es a D; los términos arman la analogía mediante relación de los objetos, incluso podemos utilizar la relación de mundos para buscar analogías.
- **falacias.php**: El software está estructurado de la siguiente manera: Armar premisas dado un conjunto de términos, genera todas las premisas validas posibles por una única vez y las almacena en la base de datos para armar y validar falacias
- **simil.php**: Este software está estructurado de la siguiente manera: Símil de cuatro términos A, B, C, D donde se expresa, A es a B como C es a D. Los términos arman la símil mediante propiedad de los objetos, incluso podemos utilizar la propiedad de mundos para buscar la símil.

5.3 DEFINICIÓN DE FUNCIONES

En este apartado se encuentra la definición y explicación de las funciones correspondientes a cada una de las clases que se encuentran en los archivos de las carpetas components e includes; la definición se realiza mostrando el nombre exacto de la función tal y como aparece en los archivos y la explicación describe que realiza cada función y cómo lo realiza.

Para esto se especifica a cual clase y a que archivo pertenece cada función.

5.3.1 Definición de funciones de la carpeta includes:

5.3.1.1 analogias.php

- class mesonil
 - **functionmostrarpropiedades(\$idejemplo)**: Para el uso de esta función, se utiliza como parámetro inicial el id del ejemplo, y a partir de este se realiza una búsqueda de las propiedades de los objetos que concuerden con el id del ejemplo inicial y se ordenan de forma aleatoria; dependiendo de las filas que resulten de esta búsqueda, si no hay ninguna, se muestra un mensaje diciendo que no hay suficientes propiedades definidas para generar el ejercicio o en caso contrario, si hay más de una fila como resultado, se muestra al usuario para que escoja la que mejor se le parece que se acomoda al ejercicio.
 - **functionmostrarparejas(\$idC, \$idD, \$idrelacion, \$nombrecontrol)**: Con esta función se pretende mostrar las parejas que existen; para esto se realiza una búsqueda de todas las relaciones entre objetos y dependiendo de si hay o no concordancias, se guarda tanto el id del objeto 1 como la del objeto 2 y de la relación existente entre ellos.
 - **functionvalidarparejas(\$P1, \$P2, \$R)**: Usando esta función se pretenden validar las parejas que se pueden formar usando las premisas y la relación como parámetros iniciales, y dependiendo de tanto las premisas como la relación, se hacen comparaciones de las premisas escogidas para saber si son iguales o si la relación entre ellas que se escogió es la verdadera o no.

- **functionarmaranalogiasdeunejemplo(\$idejemplo)**: Con base de esta función, se busca armar analogías aleatoriamente, usando como base el id del ejemplo, para eso, se realiza una búsqueda entre los ejemplos y los objetos con base en el id del ejemplo dado; de acuerdo a las coincidencias que se encuentren en esta búsqueda, y ordenándolas aleatoriamente; después de esta búsqueda, se realiza una nueva buscando de acuerdo a un id de relación específica así como del mismo id del ejemplo, si el resultado es de más de 2 filas, se realiza la analogía , con los dos objetos y el conector entre ellos y en el caso que no hayan más de dos, se mostrara un mensaje anunciando que no hay suficientes relaciones para generar el ejercicio.

5.3.1.2. falacias.php

- classmosfalacias
 - **functionmosfalacias(&\$db)**: Por medio de esta función se realiza la conexión a la base de datos para poder utilizar toda la información disponible sobre falacias.
 - **functionarmarfalaciasdeunejemplo(\$idejemplo)**: Para utilizar esta función, que tiene como parámetro inicial el id del ejemplo, se realiza una búsqueda de la premisa mayor, donde concuerde con el idejemplo y ordenando esta búsqueda por el id de la premisa; al obtener este resultado, si no hay columnas que concuerden, se muestra un mensaje o en el caso contrario, se halla tanto la figura como la premisa menos, dependiendo de los cuatro casos disponibles buscando el sujeto y el predicado y por último se haya la conclusión de la tabla de premisas donde la figura concuerde con el id de la figura correspondiente, y dependiendo de los 4 casos disponibles; por último, se unen los 3 componentes, devolviendo la falacia compuesta.
 - **funcionescribirfalacia(\$ver = false)**: Con esta función, permite ingresar la información de la falacia, como lo es, la premisa mayor, la premisa menor, la conclusión, el modo, la figura, el término medio y los términos utilizados; a partir de ahí se empieza a usar parámetros para comparar que la

información pueda ser correcta y pueda ser usada, por último se devuelve el valor del medio.

- **functionalmacenaractividad()**: Usando esta función, se puede almacenar una actividad, tomando la información de modo, figura, término medio y los términos usados, luego se hacen comparaciones para saber si la información se puede usar, generando el valor de estado y devolviendo en una variable este valor.
- **functionvalidarfalacia_BASICA(\$falacia)**: A través de esta función se puede validar una falacia, utilizando primero el parámetro \$falacia como base; esta se utiliza para hallar el valor de \$figura, aplicando la función validarregla1, dependiendo del valor que tome la figura y mientras sea mayor que cero, existen cuatro casos donde se realizan comparaciones para saber si el estado del término medio es válido o invalido; en el caso que sea no sea invalido, se asigna a la falacia, la figura, el término medio y el mensaje.
- **functionvalidarfalacia(\$falacia)**: Por medio de esta función se busca realizar una validación de la falacia generando la conclusión; para eso se utiliza el parámetro base que es la falacia y a su vez se utiliza la función validarregla1, para obtener el valor de la figura, dependiendo de este valor mientras sea mayor que cero, existen cuatro casos posibles dentro de los cuatro cada uno obtiene la conclusión que se genere con la información de la falacia; ahora se procede a hallar las clases, diferenciando los posibles cuantificadores con las copulas tanto para la premisa mayor como para la premisa menor, dejando siempre la opción de decir cuando es invalido validar una falacia dependiendo de la sintaxis; por último se halla la conclusión haciendo una búsqueda desde los datos de modos y dependiendo de los resultados, se une a la falacia correspondiente, devolviendo la falacia validada.
- **functionhallarmodo(\$figura, \$falacia)**: Esta función utiliza como parámetros iniciales la figura y la falacia correspondiente, con esta información, realiza una búsqueda de los modos que coincidan con los parámetros, y si se encuentra alguna fila, se devuelve el nombre, o en el

caso que no halla coincidencia, se devuelve un espacio vacío.

- **functionvalidarregla1(\$falacia)**: Utilizando esta función, con el parámetro inicial \$falacia, se permite validar la primera regla de tener tres términos, mayor, medio y menor y que los términos mayores y menor se relacionen con el medio; también se, diferencia dependiendo si el término mayor es su sujeto o un predicado y se muestra el caso en el que la regla 1 no se cumpla, devolviendo la figura validada.
- **functionvalidarregla2(\$falacia)**: Por medio de esta función, se realiza la validación de la regla 2, tomando como parámetro inicial el silogismo; esto se realiza haciendo una búsqueda de modos, mostrando toda la información de las concordancias, pero en el caso que no haya ningún resultado se devuelve un true, o en el caso que las filas sean mayor que cero, se devuelve un false.
- **functionvalidarregla3(\$falacia)**: Con esta función y usando como parámetro inicial una falacia se pretende validar la regla 3, donde dice que el término medio no puede aparecer en la conclusión, esto se realiza comparando la información tanto del término medio como de la conclusión y si esta comparación es cierta, devuelve un false o en el caso contrario, devuelve un true, validando la regla 3.
- **function validarregla4(\$falacia)**: A través de esta función, se quiere validar la regla 4 usando como parámetro inicial una falacia donde se encuentra toda la información; para esto se realiza una búsqueda de toda la información de modos donde concuerde tanto la figura como la premisa mayor y a esta búsqueda se le suma, un nuevo resultado de las concordancias con la premisa menor; de toda esta búsqueda se obtienen los resultados, y si la cantidad de filas es igual a cero, se devuelve un false, pero si es diferente de cero se devuelve un true.
- **functionvalidarregla5(\$falacia)**: Usando esta función se pretende validar la regla 5 donde no se pueden usar premisas negativas, ya que a partir de ellas no se puede concluir nada; para esto se utiliza el parámetro inicial

\$falacia, y se hace una comparación de tanto la premisa mayor como de la premisa menor, si estas dos comparaciones son ciertas se devuelve un false, pero si alguna de las dos o las dos no se cumplen, se devuelve un true, validando la regla 5.

- **function validarregla6(\$falacia):** Esta función permite validar la regla 6 donde se mira la conclusión de la falacia que es el parámetro inicial de la función; inicializando dos variables como true y realizando una comparación de la premisa mayor y la premisa menor y si estas comparaciones son ciertas, se realiza una última comparación, relacionada con la conclusión y mientras todas las comparaciones realizadas sean ciertas, la primera variable inicializada tomara el valor de false, pero si las 3 comparaciones realizadas son falsas, la primera variable tomara el valor de true; a continuación, se realiza de nuevo la misma comparación, pero ahora para que la segunda variable tome el valor de acuerdo al resultado de las 3 comparaciones; devolviendo por último, tanto el valor de la variable 1 (\$respuesta1) como de la variable 2 (\$respuesta2).
- **\$functionvalidarregla7(\$falacia):** A partir de esta función, se quiere validar la regla 7 donde, de premisas particulares no se puede concluir nada; para esto usando las premisas mayor y menor de la falacia, se hace una comparación doble, donde las dos se deben cumplir para que la comparación se cumpla, si se cumple se devuelve un false, y si no se cumplen las dos, se devuelve un true, validando la regla nombrada.
- **functionvalidarregla8(\$falacia):** Utilizando esta función, se quiere validar la regla 8 que dice que de premisas afirmativas no se puede inferir una conclusión negativa; para eso se utilizan las premisas mayor y menor de la falacia y se realiza una comparación donde las dos se deben cumplir, en el caso que esto suceda, ocurre una tercer comparación usando la conclusión y de estas 3 comparación, si las 3 se cumplen se devolverá un false o en el caso contrario, devuelve un true.

- **functionvalidarreglas(\$falacia)**: Por medio de esta función, se validan las 8 reglas al tiempo, dejando la primera regla para validar al final, llamando cada función donde se valida cada regla por separado y describiendo en que consiste cada regla y dado el caso en el que no se cumpla, que parte es la que está mal planteada.
- **functioninsert()**: Con esta función se inserta un nuevo objeto en la base de datos; si resulta algún problema, se devuelve un mensaje y se devuelve false, o en el caso contrario, se devuelve un true.

5.3.1.3 razonamiento.php

- classmosrazonamiento
 - **functionmostrarpropiedades(\$idejemplo)**: Se hace una lista con relaciones de términos falsas, y se busca las parejas que puedan tener relación.
 - **functionmostrarterminos(\$idB, \$B, \$idpropiedad, \$nombrecontrol)**: De hace una lista con pares de términos falsos, y se busca las parejas que puedan tener relación.
 - **functionvalidarterminos(\$P1, \$P2)**: Se validan pares de acuerdo con la id de su estructura.
 - **functionimprimir(\$lst)**: En esta función se formatea para obtener una salida más amigable.
 - **functionarmarrazonamiento(\$idobjeto, \$idejemplo)**: Se genera de manera aleatoria un razonamiento valido para un ejemplo específico, lo que primero se debe hacer es escoger y verificar que existan las propiedades, las relaciones de del objeto1 y del objeto2 que se desea seleccionar para así poder generar un ejercicio.
 - **functionparejadelobjeto(\$idobj)**: Se crea un segundo objeto, se escogen las propiedades de los objetos y se verifica que existan términos que compartan dichas propiedades para el ejemplo de razonamientos determinado así terminar el procedimiento.

5.3.1.4 silogismos.php

- classmosejemploextendsmosDBTable
 - **functionmosejemplos(&\$db)**: Por medio de esta función se realiza la conexión a la base de datos y queda guardada cierta información como los números de los ejemplos y el nombre de la base de datos en "\$this".
 - **functionaveriguaridejemplo(\$nom)**: Con esta función se realiza una búsqueda de un ejemplo de acuerdo al parámetro \$nom y si la búsqueda es nula, se devuelve un null.
 - **functionasignarejemplo (\$idcurso, \$idejemplo)**: Mediante esta función y usando el id del ejemplo y el id del curso, se obtienen ejemplos, pero solo los que correspondan a cierto curso y al obtenerlos se ordenan de forma aleatoria para mostrar no siempre el mismo ejemplo.
 - **functionloadfromvector(\$doc)**: En esta función, se carga un vector con 7 parámetros como el id, el nombre, el texto, el usuario, el estado, el nivel, y el id de la figura y la información se guarda de acuerdo al parámetro \$doc.
 - **functionupdate(\$updateNulls=false)**: Por medio de esta función, se realiza la actualización a una fila, indicando cual fila de acuerdo al "id" y de acuerdo a la existencia o no de algún error, se retorna un false que indica que si hubo un error o un true indicando que no.
 - **functioninsertexample(\$idejemplo, \$idp, \$ter, \$g, \$usuario, \$estado, \$nivel)**: En esta función se realiza la inserción de nuevos ejemplos, guardando la id del ejemplo y asignándole espacio en la base de datos de términos, usando campos como id, id del padre, genero, entre otros.
 - **functioninsert()**: Con esta función se inserta un nuevo objeto en la base de datos; si resulta algún problema, se devuelve un mensaje y se devuelve false, o en el caso contrario, se devuelve un true.
 - **functionborrarejemplo(\$id)**: A través de esta función se borra un ejemplo usando el id como indicador, utilizando otras funciones como borrar desde ejemplo y cargando el ejemplo en una variable se procede a borrarla.

- **functionmostrarterminossegunejemplo(\$id)**: Con esta función, se muestran los términos de acuerdo al id de un ejemplo, realizando primero una búsqueda de acuerdo al id y guardándola en una variable; a partir de ahí se hace la creación de la interfaz en la que se mostraran los resultados.
 - **functionobjetossegunejemplo(\$id)**: Mediante esta función y utilizando el id, se obtiene una lista de los objetos con algunas descripciones y ordenados por el nombre, este resultado es guardado en una variable y se muestra el código de la interfaz para la presentación del resultado.
 - **functionasignarobjetoaunejemplo(\$id)**: Por medio de esta función, se realiza la selección de la lista de objetos que se encuentre en la base de datos y se muestra la interfaz para que con el id del ejemplo se le asigne un objeto.
 - **functionmostrarplantillaincluirterminos(\$id)**: En esta función se puede observar el código de la interfaz para incluir términos, en la interfaz aparecen parámetros como idejemplo, para saber a qué ejemplo se refiere, los id's tanto de las premisas como de los términos, que se ha estipulado que sean máximo 15 términos y por último se describe el resto de interfaz.
- **classmosterminosextdensmosDBTable**
 - **functionmosterminos(&\$db)**: Esta función realiza la conexión con la base de datos que se relacionan con términos por medio del id.
 - **functionbuscarid()**: Con esta función, se obtiene el id de un término, buscándolo con ayuda del id del ejemplo y el nombre de la base de datos disponible; al obtener el resultado, se cuentan las filas y si el número es igual a cero, se devuelve un null, o en el caso contrario devuelve la fila correspondiente.
 - **functionloadfromvector(\$doc)**: A través de esta función, se carga en un vector 5 campos de información, guiados por el parámetro "\$doc"; el vector contiene el id que en este

caso sería el mismo \$doc, también tiene el id del padre, el nombre, el género y el id del ejemplo al que corresponde.

- **functioninsert()**: Con esta función se inserta un nuevo objeto en la base de datos; si resulta algún problema, se devuelve un mensaje y se devuelve false, o en el caso contrario, se devuelve un true.
 - **functionborrardesdeejemplo(\$idejemplo)**: Esta función tiene como objetivo, que utilizando el id del ejemplo se realice una búsqueda de términos en la base de datos, contando las filas que coincidan, a partir de esto, se borra cada fila mostrada.
- **classmospremisasextendsmosDBTable**
 - **functionmospremisas(&\$db)**: Por medio de esta función se realiza la conexión a la base de datos, conectando con la tabla premisas y poniendo como parámetro, el nombre de la base de datos relacionada con figuras retóricas.
 - **functionguardarpremisas(\$idejemplo, \$sujeto, \$predicado, \$genero, \$opción)**: Con esta función se busca armar premisas utilizando los parámetros idejemplo, sujeto, predicado, género y opción; estas se arman de acuerdo a diferentes opciones, que en este caso son 4 tipos, para sacar los respectivos cuantificadores y por último se verifica que la premisa haya quedado bien generada con la función verify().
 - **functionverify()**: A través de esta función, se hace una búsqueda de todas las premisas existentes en la base de datos que concuerden con el id del ejemplo y de la premisa; al realizar la búsqueda, se sabe si ya hay una premisa con estos datos en la base de datos para no insertar una repetida, dado el caso que no este, se realiza su inserción normal, pero dado el caso que si este, se muestra un mensaje diciendo que ya se encuentra en la base de datos.
 - **functionbuscarpadre(\$idpadre, \$termino, \$genter)**: En esta función se realiza la búsqueda del padre de una premisa usando parámetros, como idpadre y término, y mientras el id del padre no sea null y el término no este vacío, se obtiene toda la información de los términos y

mientras en esta búsqueda hayan filas se arma las premisas usando la función guardarpremisas y se guarda todo el resultado en "\$mipremisa".

- **functionbuscarhijo(\$idpadre, \$termino, \$genter):** Con esta función, se realiza la búsqueda de hijos utilizando parámetros iniciales como idpadre y término, con estos términos se realiza una búsqueda de todos los términos que concuerden con los parámetros iniciales y con estos resultados, se arman las premisas guardándolas en "\$mipremisa".
- **functionbuscarsobrino(\$id, \$idpadre, \$termino, \$genter):** A través de esta función, se quiere buscar a los sobrinos de premisas, primero haciendo una búsqueda de todos los términos que concuerden con el idpadre y el id de términos, con el resultado de esta búsqueda, se arman las premisas guardándolas en buscarhijo, ahora se realiza una nueva búsqueda con la información que concuerde con el mismo padre buscando por el id de los términos, con este resultado de búsqueda se arman los sobrinos usando tanto las funciones buscarhijo como buscarsobrino y guardando todo en \$mipremisa.
- **functionvalidarpremisa(\$idejemplo, \$cad):** Por medio de esta función, se validan premisas, usando para ellos los parámetros idejemplo y cad; usando una búsqueda de las premisas que concuerde con los dos parámetros iniciales, con este resultado, si devuelve una sola fila, es decir la premisa, la complementa, pero sino la encuentra, devuelve el resultado vacío.
- **functionverpremisas(\$idejemplo):** Esta función, permite ver las premisas que concuerden con el parámetro inicial idejemplo; para esto, realiza una búsqueda de la premisa que concuerde con el ejemplo y a partir del resultado, mientras no sea vacío, hace una nueva búsqueda de toda la información del ejemplo que concuerde con el mismo parámetro inicial, con la información obtenida, la muestra en una lista, y para imprimir los términos de la premisa de ese ejemplo, realiza una búsqueda similar con toda la información de los términos que concuerden con el parámetro idejemplo y con el resultado va imprimiendo los

términos correspondientes; por último imprime las premisas completas de ese ejemplo.

- **functionestandarizarpremisa(\$cad)**: Con esta función, se busca quitar los espacios entre palabras que pueda surgir, y como el nombre de la función dice, busca estandarizarlo y dejar las premisas en un solo formato para ser más fáciles de manipular.
- **functionvalidarsintaxispremisa(\$idejemplo,\$cad)**: A través de esta función, se busca valida la sintaxis de las premisas que tanto se ingresen a la base de datos como se usen para realizar ejercicios en cuanto a cuantificadores, sujeto, copula y predicado, primero definiendo tanto los cuantificadores como las copulas permitidas y en adelante se empieza a particionar la premisa y validar que la sintaxis usada sea la permitida, devolviendo mensajes como "falta el cuantificador o está mal ubicado".
- **functionarmarpremisasdeunejemplo(\$idejemplo)**: Usando esta función, se busca armar una premisa a partir de un ejemplo; para esto, primero se usa la función `borrardesdeejemplo`, usando para parámetro `idejemplo`, siguiendo con una búsqueda de términos que concuerden también con `idejemplo`; a partir de este resultado, y contando las filas resultantes, se usan las funciones `buscarnpadre` y `buscarsobrino` para armar las premisas.
- **functionloadfromvector(\$doc)**: Con esta función, se carga un vector de 8 filas, que información que concuerde con el parámetro inicial que es `$doc`; en el vector se guarda información como `idejemplo`, cuantificador, sujeto, entre otros.
- **functioninsert()**: Con esta función se inserta un nuevo objeto en la base de datos; si resulta algún problema, se devuelve un mensaje y se devuelve `false`, o en el caso contrario, se devuelve un `true`.
- **functionupdate(\$updateNulls=false)**: Por medio de esta función, se realiza la actualización a una fila, indicando cual fila de acuerdo al "id" y de acuerdo a la existencia o no de algún error, se retorna un `false` que indica que si hubo un error o un `true` indicando que no.

- **functionborrardesdeejemplo(\$idejemplo)**: Esta función tiene como objetivo, que utilizando el id del ejemplo se realice una búsqueda de términos en la base de datos, contando las filas que coincidan, a partir de esto, se borra cada fila mostrada.
- **classmossilogismos**
 - **functionmossilogismos(\$db)**: Utilizando esta función se realiza la conexión a la base de datos para poder utilizar todos los campos e información que se conecte con silogismos y sus componentes.
 - **functionarmarsilogismodeunejemplo(\$idejemplo)**: A través de esta función se busca armar silogismos, utilizando como base un id de ejemplo, realizando una busca de premisas donde concuerden con el id del ejemplo, si de esta búsqueda no resultan filas se muestra un mensaje, pero en el otro caso, es decir, que hayan más de una fila, primero se busca la figura aleatoriamente, luego se busca la premisa mayor, donde se especifican los 4 casos que pueden surgir, seguidamente se busca la premisa menor y por último la conclusión, especificando los 4 casos posibles y por último, se une toda la información anterior.
 - **functionescribirisilogismos(\$ver = false)**: En esta función se busca que por medio de la interfaz que es descrita, se pueda escribir, la premisa mayor, la premisa menor y la conclusión, especificando a su vez el modo, la figura, el término medio y los términos usados; con esta información, se procede a asignarle variables para poder almacenarla en la base de datos.
 - **functionalmacenaractividad()**: Con esta función, se usa toda la información que en la función escribirisilogismos(), se trae de nuevo con sus respectivas variables, y se almacena tanto el medio como el estado, retornando estos dos valores en la variable \$alm.
 - **functionvalidarsilogismos(\$silogismo)**: Usando esta función, se puede validar un silogismo, primero trayendo el silogismo que se quiere validar y primero se toma la figura; y va tomando cada caso disponible que se pueda

presentar, sacando así, el sujeto y el predicado, para saber si el invalido o no; en caso que sea invalido, se usa la función hallarmodo, y en el caso que no sea invalido se obtienen el estado, la figura, el término medio y el silogismo para devolver el silogismo validado.

- **functionvalidarsilogismosinconclusion(\$silogismo)**: Por medio de esta función y utilizando un silogismo se puede realizar su validación pero sin contener la conclusión como componente, para esto, primero se debe obtener la conclusión de la información con la que se cuenta; seguidamente se hallan las clases que correspondan, dependiendo del cuantificador y de la copula y se encuentra la conclusión dependiendo del caso que se presente, devolviendo por último el silogismo validado.
- **functionhallarmodo(\$figura, \$silogismo)**: Esta función utiliza como parámetros iniciales la figura y el silogismo correspondiente, con esta información, realiza una búsqueda de los modos que coincidan con los parámetros, y si se encuentra alguna fila, se devuelve el nombre, o en el caso que no halla coincidencia, se devuelve un espacio vacío.
- **functionvalidarregla1(\$silogismo)**: Utilizando esta función, con el parámetro inicial \$silogismos, se permite validar la primera regla de tener tres términos, mayor, medio y menor y que los términos mayores y menor se relacionen con el medio; también se, diferencia dependiendo si el término mayor es su sujeto o un predicado y se muestra el caso en el que la regla 1 no se cumpla, devolviendo la figura validada.
- **functionvalidarregla2(\$silogismo)**: Por medio de esta función, se realiza la validación de la regla 2, tomando como parámetro inicial el silogismo; esto se realiza haciendo una búsqueda de modos, mostrando toda la información de las concordancias, pero en el caso que no haya ningún resultado se devuelve un true, o en el caso que las filas sean mayor que cero, se devuelve un false.
- **functionvalidarregla3(\$silogismo)**: Con esta función y usando como parámetro inicial un silogismo se pretende validar la regla 3, donde dice que el término medio no

puede aparecer en la conclusión, esto se realiza comparando la información tanto del término medio como de la conclusión y si esta comparación es cierta, devuelve un false o en el caso contrario, devuelve un true, validando la regla 3.

- **function validarregla4(\$silogismo):** A través de esta función, se quiere validar la regla 4 usando como parámetro inicial el silogismo donde se encuentra toda la información; para esto se realiza una búsqueda de toda la información de modos donde concuerde tanto la figura como la premisa mayor y a esta búsqueda se le suma, un nuevo resultado de las concordancias con la premisa menor; de toda esta búsqueda se obtienen los resultados, y si la cantidad de filas es igual a cero, se devuelve un false, pero si es diferente de cero se devuelve un true.
- **function validarregla5(\$silogismo):** Usando esta función se pretende validar la regla 5 donde no se pueden usar premisas negativas, ya que a partir de ellas no se puede concluir nada; para esto se utiliza el parámetro inicial \$silogismo, y se hace una comparación de tanto la premisa mayor como de la premisa menor, si estas dos comparaciones son ciertas se devuelve un false, pero si alguna de las dos o las dos no se cumplen, se devuelve un true, validando la regla 5.
- **function validarregla6(\$silogismo):** Esta función permite validar la regla 6 donde se mira la conclusión del silogismo que es el parámetro inicial de la función; inicializando dos variables como true y realizando una comparación de la premisa mayor y la premisa menor y si estas comparaciones son ciertas, se realiza una última comparación, relacionada con la conclusión y mientras todas las comparaciones realizadas sean ciertas, la primera variable inicializada tomara el valor de false, pero si las 3 comparaciones realizadas son falsas, la primera variable tomara el valor de true; a continuación, se realiza de nuevo la misma comparación, pero ahora para que la segunda variable tome el valor de acuerdo al resultado de las 3 comparaciones; devolviendo por último, tanto el valor de la

variable 1 (\$respuesta1) como de la variable 2 (\$respuesta2).

- **\$functionvalidarregla7(\$silogismo)**: A partir de esta función, se quiere validar la regla 7 donde, de premisas particulares no se puede concluir nada; para esto usando las premisas mayor y menor del silogismo, se hace una comparación doble, donde las dos se deben cumplir para que la comparación se cumpla, pero si se cumple se devuelve un false, y si no se cumplen las dos, se devuelve un true, validando la regla nombrada.
- **functionvalidarregla8(\$silogismo)**: Utilizando esta función, se quiere validar la regla 8 que dice que de premisas afirmativas no se puede inferir una conclusión negativa; para eso se utilizan las premisas mayor y menor del silogismo y se realiza una comparación donde las dos se deben cumplir, en el caso que esto suceda, ocurre una tercer comparación usando la conclusión y de estas 3 comparación, si las 3 se cumplen se devolverá un false o en el caso contrario, devuelve un true.
- **functionvalidarreglas(\$silogismo)**: Por medio de esta función, se validan las 8 reglas al tiempo, dejando la primera regla para validar al final, llamando cada función donde se valida cada regla por separado y describiendo en que consiste cada regla y dado el caso en el que no se cumpla, que parte es la que está mal planteada.
- **functioninsert()**: Con esta función se inserta un nuevo objeto en la base de datos; si resulta algún problema, se devuelve un mensaje y se devuelve false, o en el caso contrario, se devuelve un true.

5.3.1.5simil.php

- class mesonil
 - **functionmostrarpropiedades(\$idejemplo)**: Para el uso de esta función, se utiliza como parámetro inicial el id del ejemplo, y a partir de este se realiza una búsqueda de las propiedades de los objetos que concuerde con el id del ejemplo inicial y se ordenar de forma aleatoria;

dependiendo de las filas que resulten de esta búsqueda, si no hay ninguna, se muestra un mensaje diciendo que no hay suficientes propiedades definidas para generar el ejercicio o en caso contrario, si hay más de una fila como resultado, se muestra al usuario para que escoja la que mejor se le parece que se acomoda al ejercicio.

- **functionmostrarterminos(\$idB, \$B, \$idpropiedad, \$nombrecontrol)**: A través de esta función, se busca mostrar los términos relacionados con los símiles, utilizando 4 parámetros iniciales y empezando con realizar una búsqueda de las propiedades de los objetos, ordenándolas aleatoriamente; a partir de este resultado si no surgió ningún resultado, se muestra un mensaje diciendo que no existen los términos para generar el ejercicio, pero si hay por lo menos una fila como resultado, se le muestra al usuario para que pueda escoger la más indicada para el ejercicio propuesto.
- **functionvalidarterminos(\$P1, \$P2)**: Por medio de esta función, se realiza la validación de términos, usando como parámetros iniciales \$P1 y \$P2, y para realizar esta función, surge una comparación para saber si los términos son iguales, dado el caso, se muestra un mensaje diciéndole esto al usuario, o en el caso contrario, se comprueban si las propiedades son iguales y si este es el caso, se le muestra al usuario un mensaje con este aviso o en el último caso, se le dice que no comparten cierta propiedad, devolviendo la respuesta dependiendo del caso que surja.
- **functionarmarsimildeunejemplo(\$idejemplo)**: Con esta función, se desea armar un símil utilizando como parámetro inicial idejemplo, primero haciendo una búsqueda de las propiedades de los objetos que estén relacionadas con el ejemplo en particular, de esta búsqueda, se realiza una nueva búsqueda, para cada objeto resultante que tanto coincidan con el ejemplo en cuestión y con la propiedad resultando de la búsqueda anterior, de esta nueva búsqueda, se sacan los resultantes y si no hay ninguna fila que concuerde se muestra un mensaje diciendo que no hay suficientes términos para el ejemplo que compartan propiedades o en el caso contrario, donde si surgieron filas

coincidentes, se inicia la carga de símil, con el objeto, el nombre, prefijo, conector y la propiedad y devolviendo un true para el estado indicando que si se pudo armar el símil correspondiente.

- **functionparejadelobjeto(\$idobj):** En esta función que utiliza como parámetro inicial el id del objeto, se inicial realizando una búsqueda de las propiedades de los objetos ordenándolos aleatoriamente, si de esta búsqueda no sale ninguna fila, se muestra un mensaje anunciando esta situación o en el caso contrario se realiza una nueva búsqueda, de las propiedades de los objetos que se relacionen con un ejemplo en particular y donde también coincida con una propiedad en particular, si de esta nueva búsqueda sale máximo una fila, se le muestra al usuario un mensaje que dice que no hay suficientes términos que compartan la propiedad para este ejemplo o en el caso contrario se empieza a cargar un símil con los objetos, prefijo, conector, las propiedades.

5.3.2 DEFINICIÓN DE FUNCIONES DE LA CARPETA COMPONENTS:

5.3.2.1 analogias.php

- **functiongenerarejercicios(\$nivel, \$idejemplo, \$ejercicio):** Esta función es para organizar la información a mostrar en la pantalla para el ejercicio NIVEL UNO de acuerdo con las opciones seleccionadas, toma como parámetros el nivel, el idejemplo y el idejercicio.
- **functionvalidarejercicios(\$nivel, \$id, \$idejercicio):** Esta función se encarga de validar un símil con la información registrada por el usuario, el procedimiento consiste en verificar el id de la propiedad.

5.3.2.2 falacias.php

- **functiongenerarejercicios(\$nivel, \$idejemplo, \$ejercicio)**: Esta función es para organizar la información a mostrar en la pantalla para el ejercicio NIVEL UNO de acuerdo con las opciones seleccionadas, toma como parámetros el nivel, el idejemplo y el idejercicio.

5.3.2.3 razonamiento.php

- **functiongenerarejercicios(\$nivel, \$idejemplo, \$ejercicio)**: Esta función es para organizar la información a mostrar en la pantalla para el ejercicio NIVEL UNO de acuerdo con las opciones seleccionadas, toma como parámetros el nivel, el idejemplo y el idejercicio.
- **functionvalidarejercicios(\$nivel, \$id, \$idejercicio)**: Esta función se encarga de validar un razonamiento con la información registrada por el usuario; el procedimiento consiste en verificar el id de la propiedad.

5.3.2.4 silogismos.php

- **functiongenerarejercicios(\$nivel, \$idejemplo, \$ejercicio)**: Esta función es para organizar la información a mostrar en la pantalla para el ejercicio NIVEL UNO de acuerdo con las opciones seleccionadas, toma como parámetros el nivel, el idejemplo y el idejercicio.

5.3.2.5 simil.php

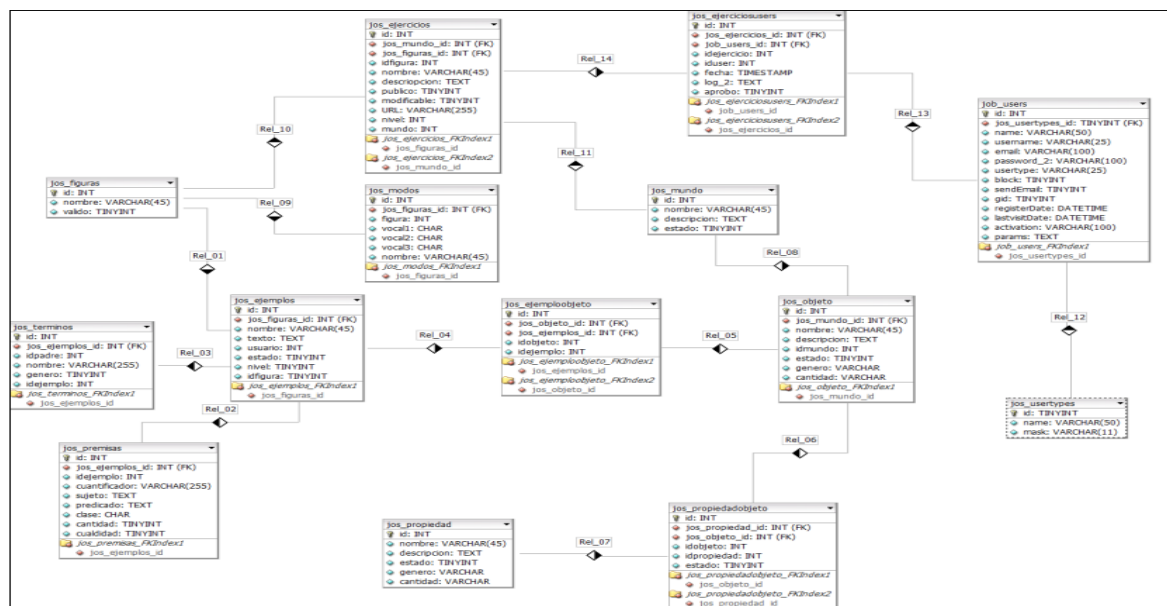
- **functiongenerarejercicios(\$nivel, \$idejemplo, \$ejercicio)**: Esta función es para organizar la información a mostrar en la pantalla para el ejercicio NIVEL UNO de acuerdo con las opciones seleccionadas, toma como parámetros el nivel, el idejemplo y el idejercicio.

- **funcionvalidarejercicios(\$nivel, \$id, \$idejercicio):** Esta función se encarga de validar un razonamiento con la información registrada por el usuario; el procedimiento consiste en verificar el id de la propiedad.

5.4 MODELO ENTIDAD-RELACIÓN

En este apartado se encuentra el modelo entidad-relación de la base de datos que se utiliza en la aplicación propuesta. Este modelo cuenta con 14 tablas que son el resultado de la independización del modelo entidad-relación de la aplicación inicial, el cual cuenta con 52 tablas, de las cuales 36 corresponden a los contenidos que Joomla tiene predeterminados.

Figura 2. Modelo entidad-relación



5.4 DIAGRAMAS DE FLUJO DE FIGURAS RETÓRICAS

En este apartado se encuentran los diagramas de flujo correspondientes al funcionamiento lógico de cada figura retórica. Estos diagramas de flujo se

realizaron con el fin de establecer la estructura lógica de cada figura retórica y reflejar la forma de operación real de las figuras.

5.5.1 Diagrama de flujo para silogismos

Figura 3. Diagrama de flujo de Silogismos 1

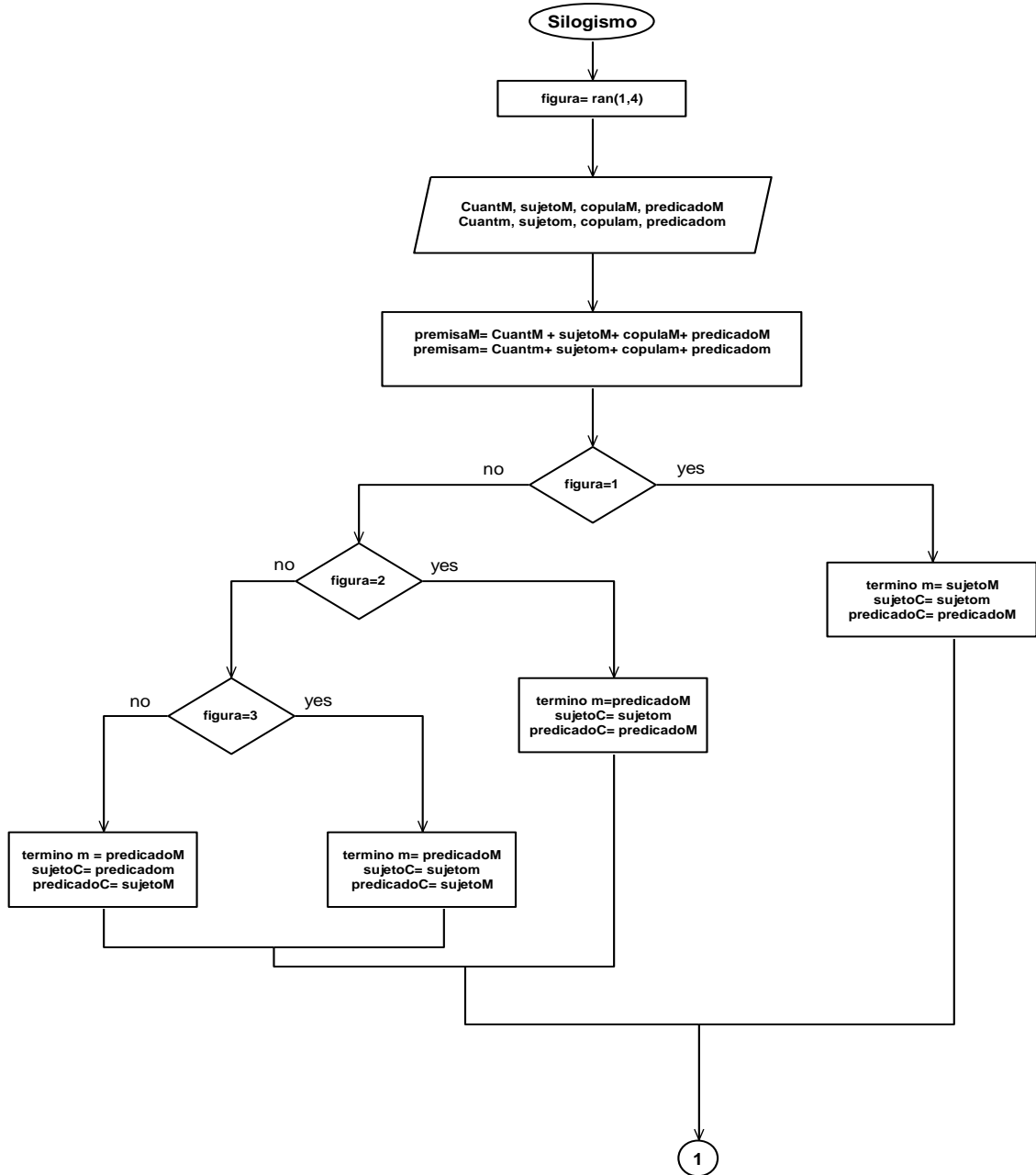


Figura 4. Diagrama de flujo de Silogismos 2

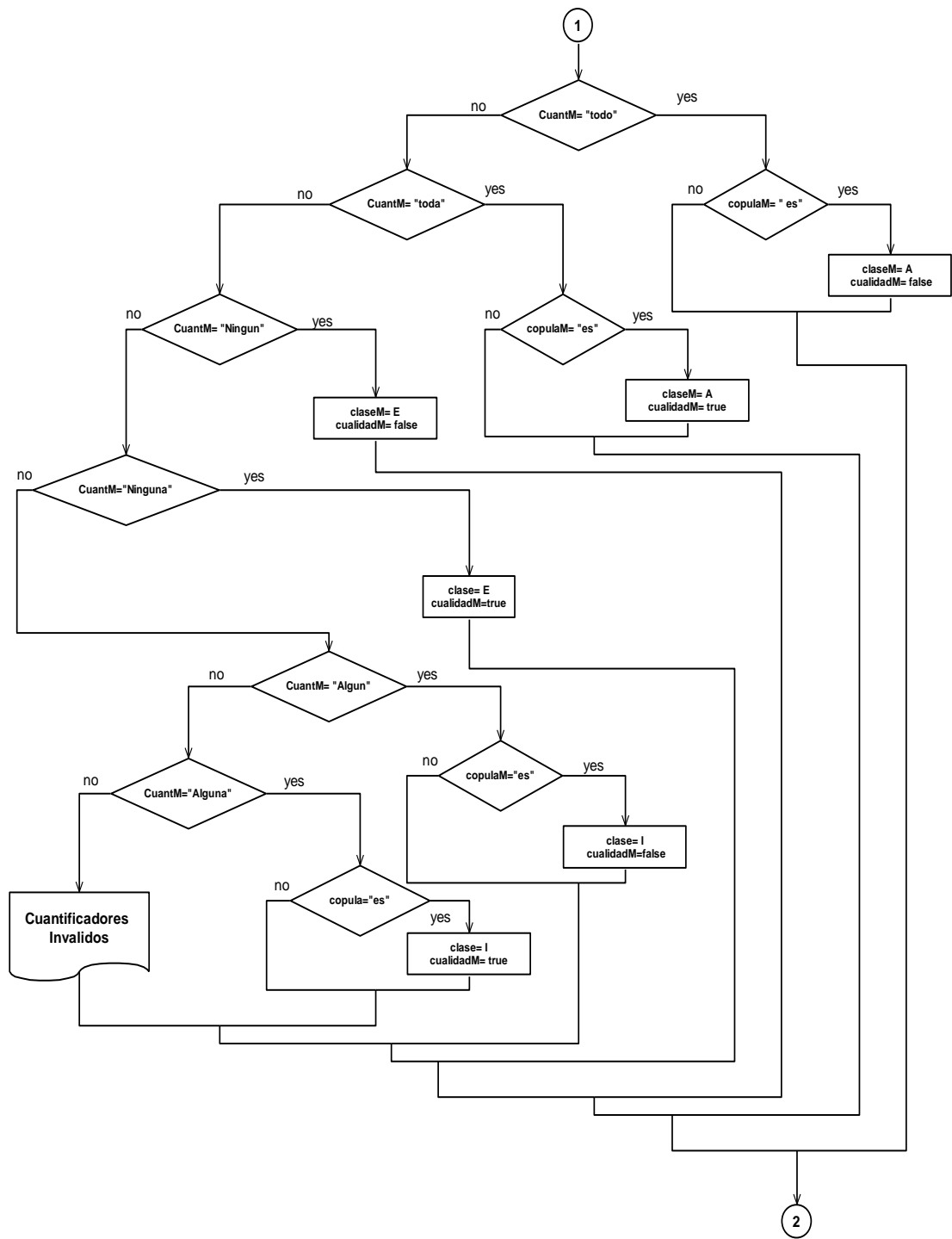


Figura 5. Diagrama de flujo de Silogismos 3

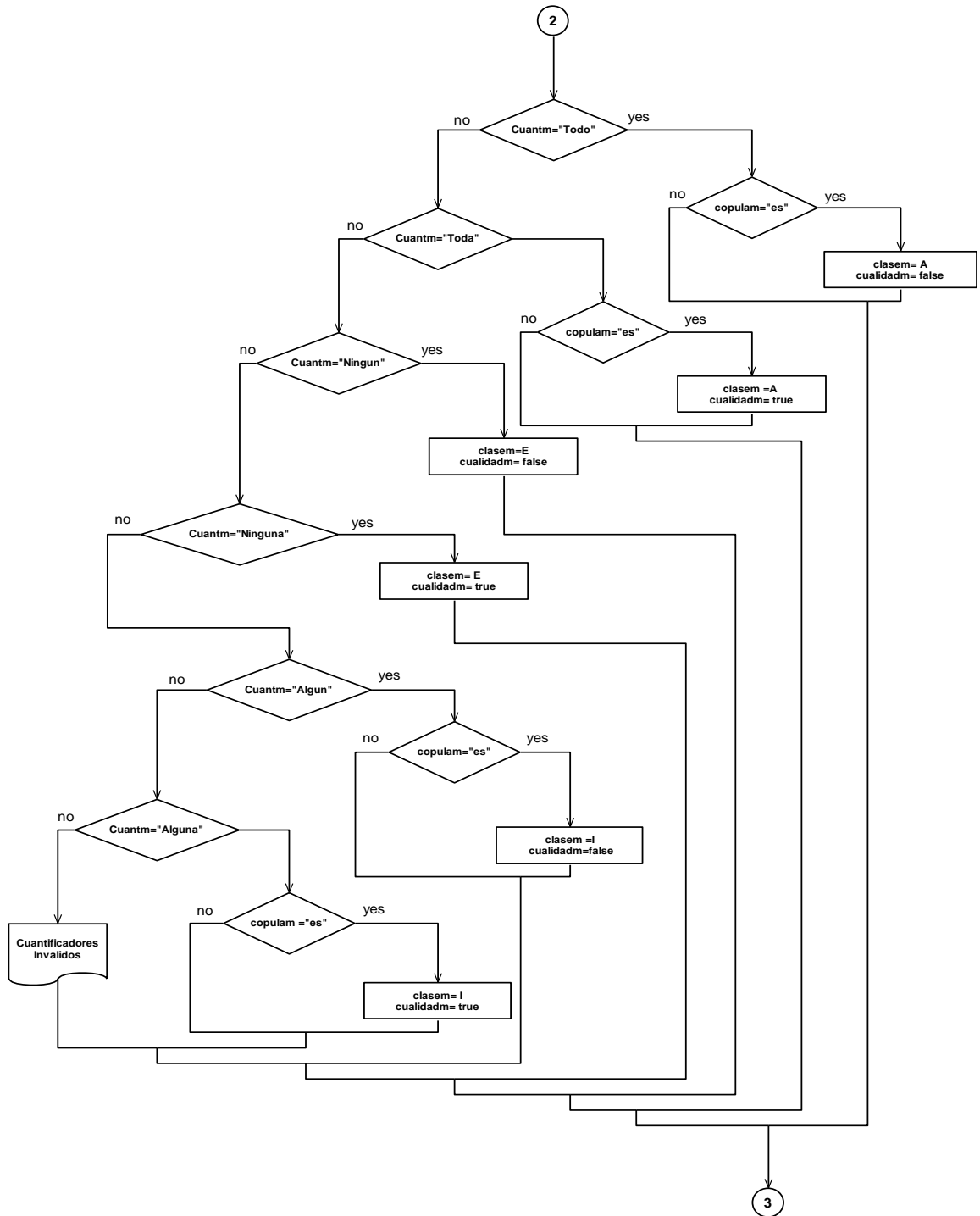
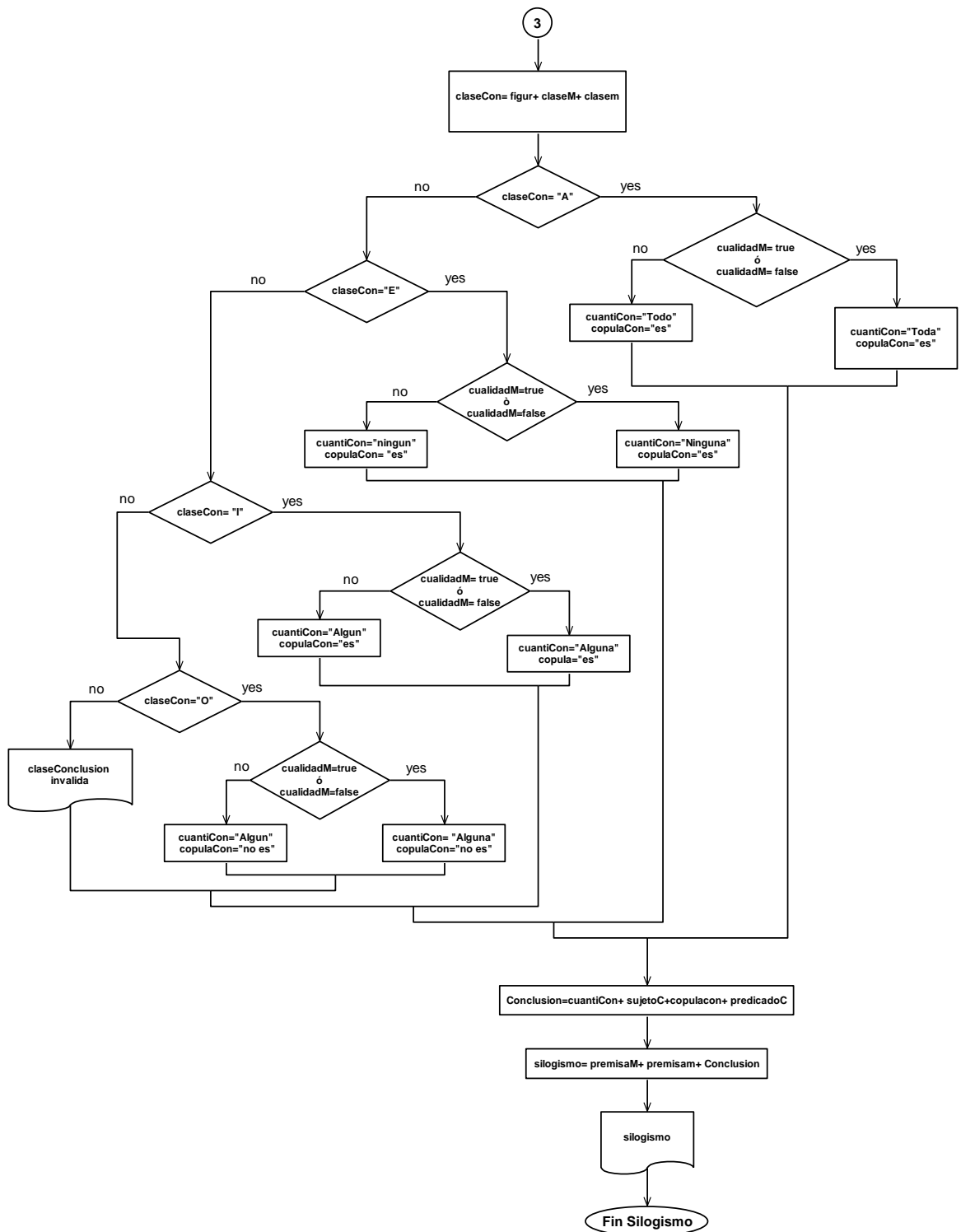


Figura 6. Diagrama de flujo de Silogismos 4



5.5.2 Diagrama de flujo para las falacias

Figura 7. Diagrama de flujo de Falacias 1

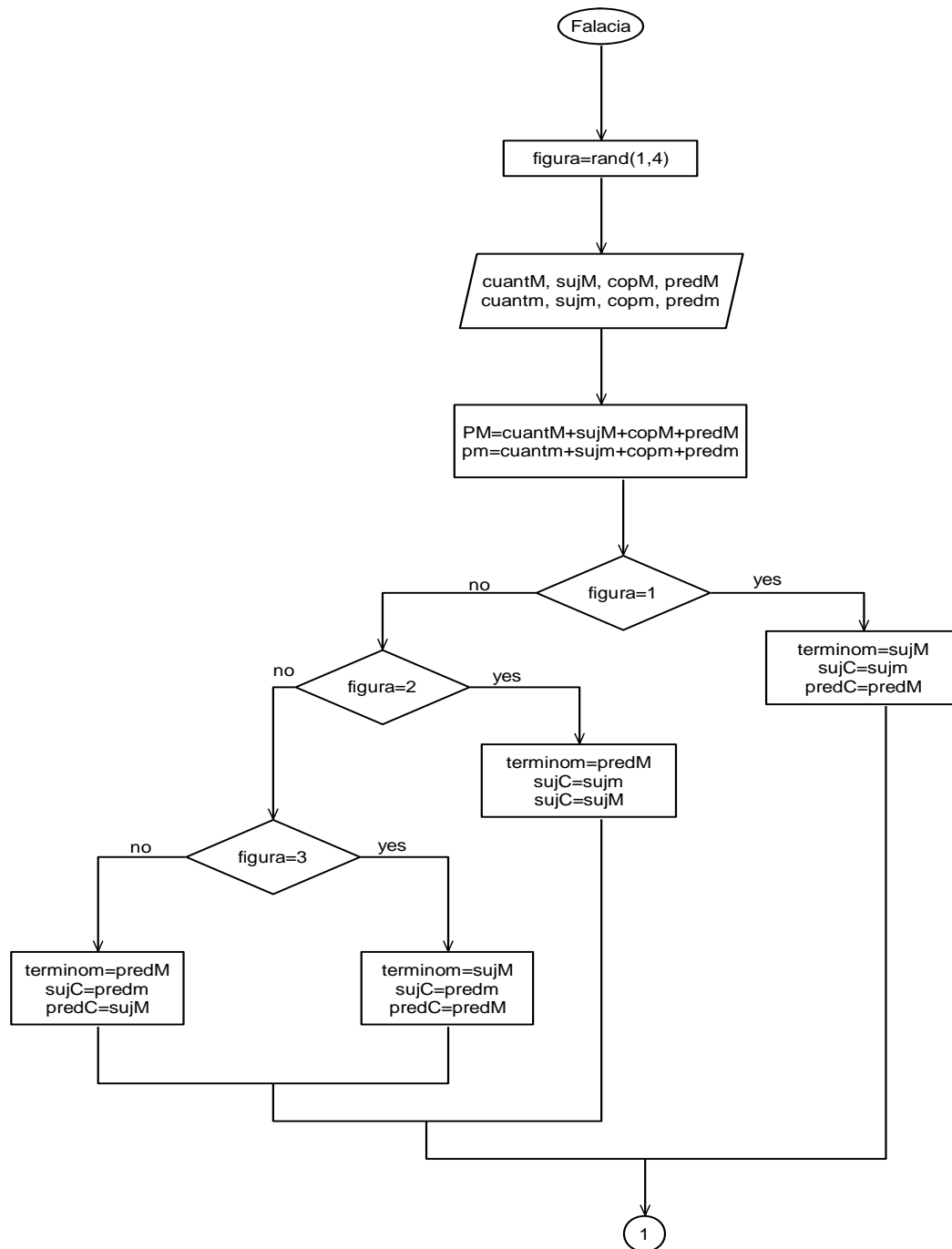


Figura 8. Diagrama de flujo de Falacias 2

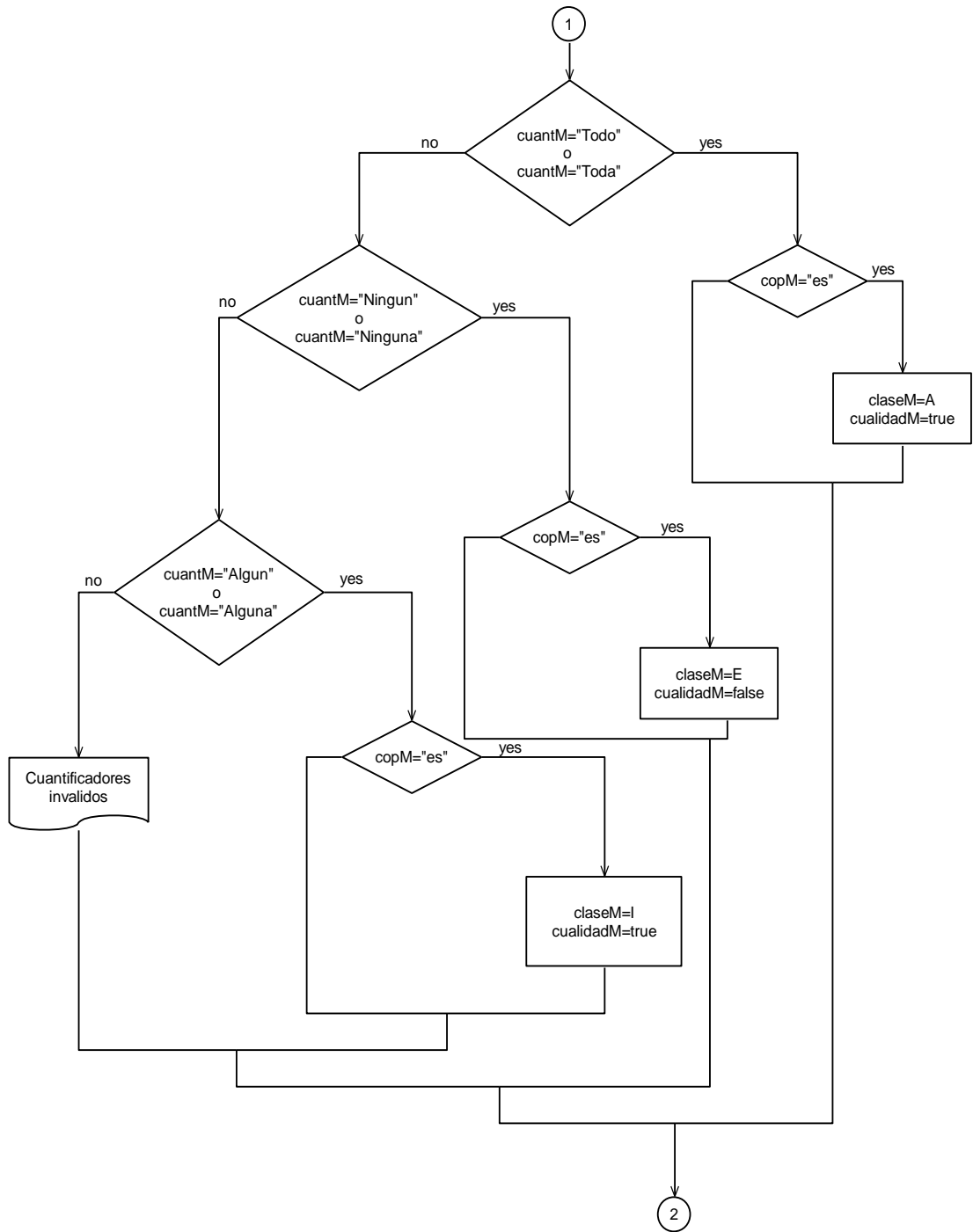


Figura 9. Diagrama de flujo de Falacias 3

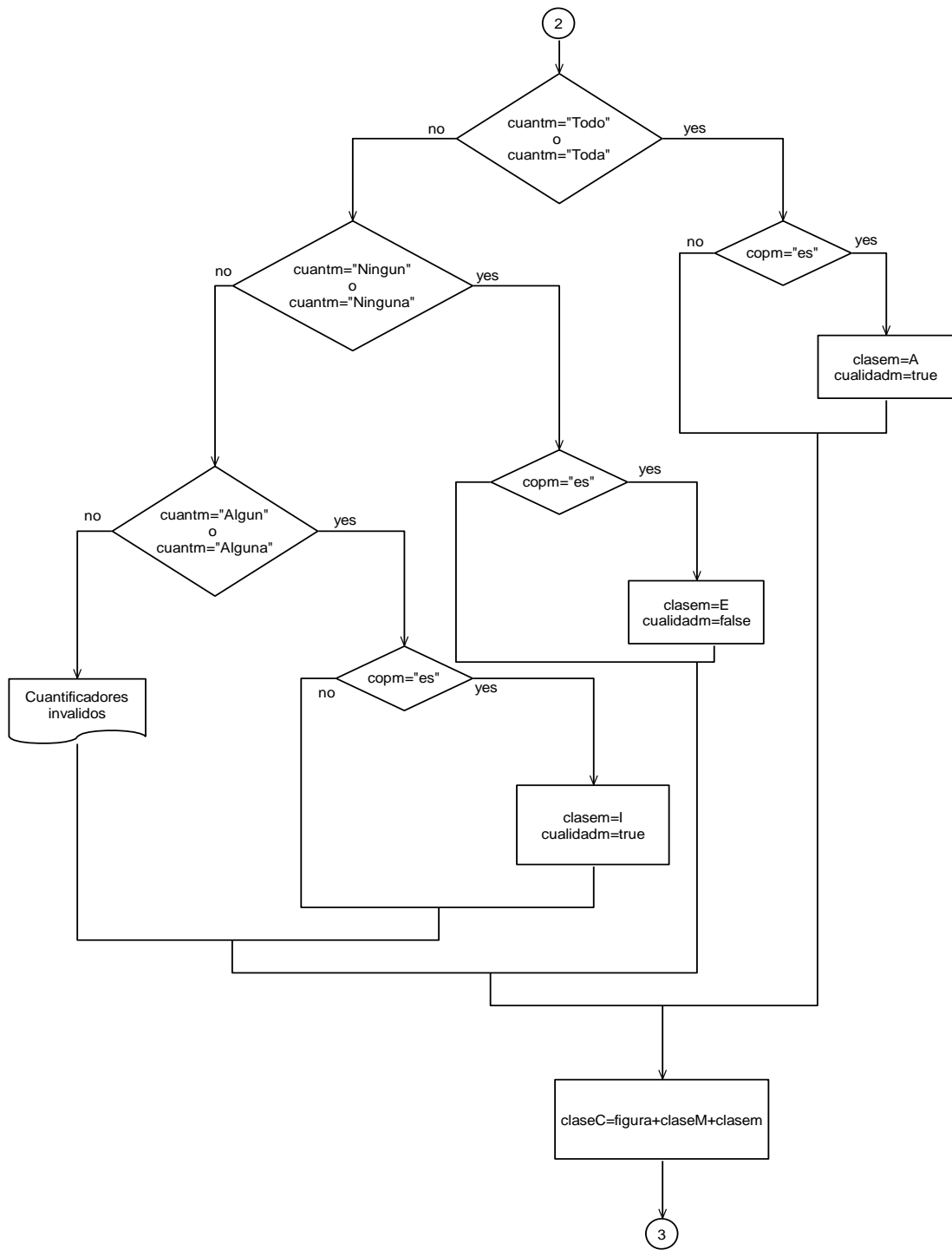


Figura 10. Diagrama de flujo de Falacias 4

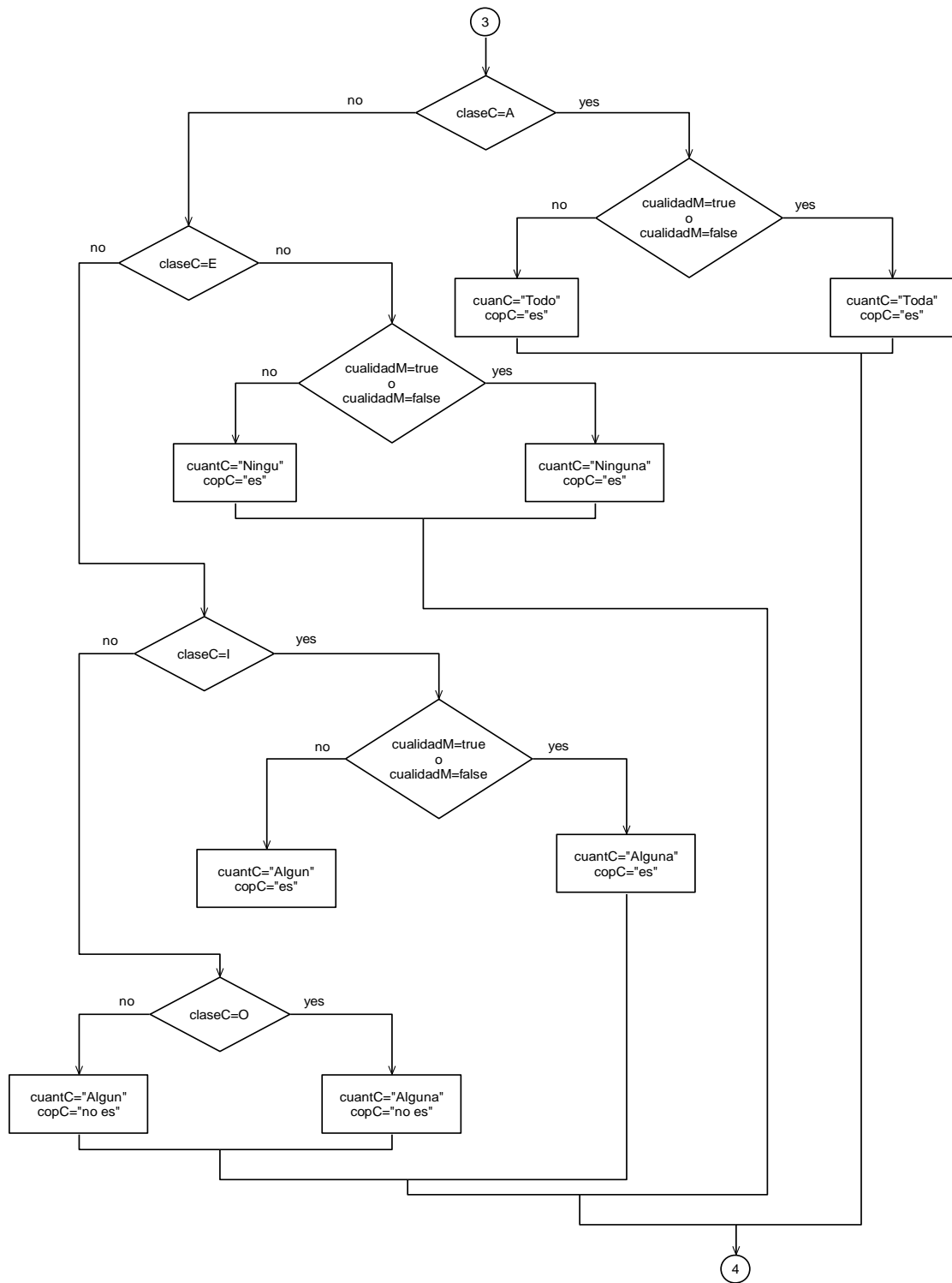
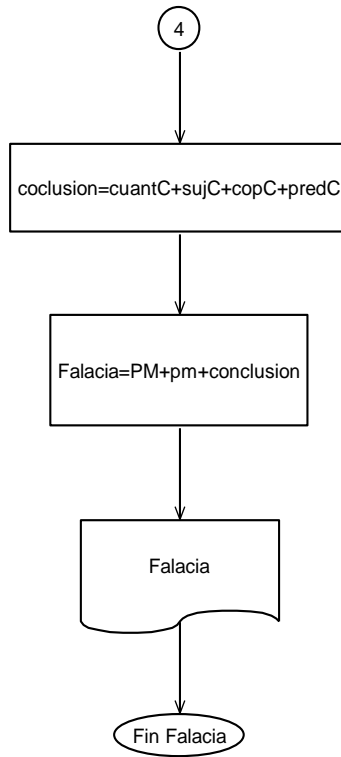
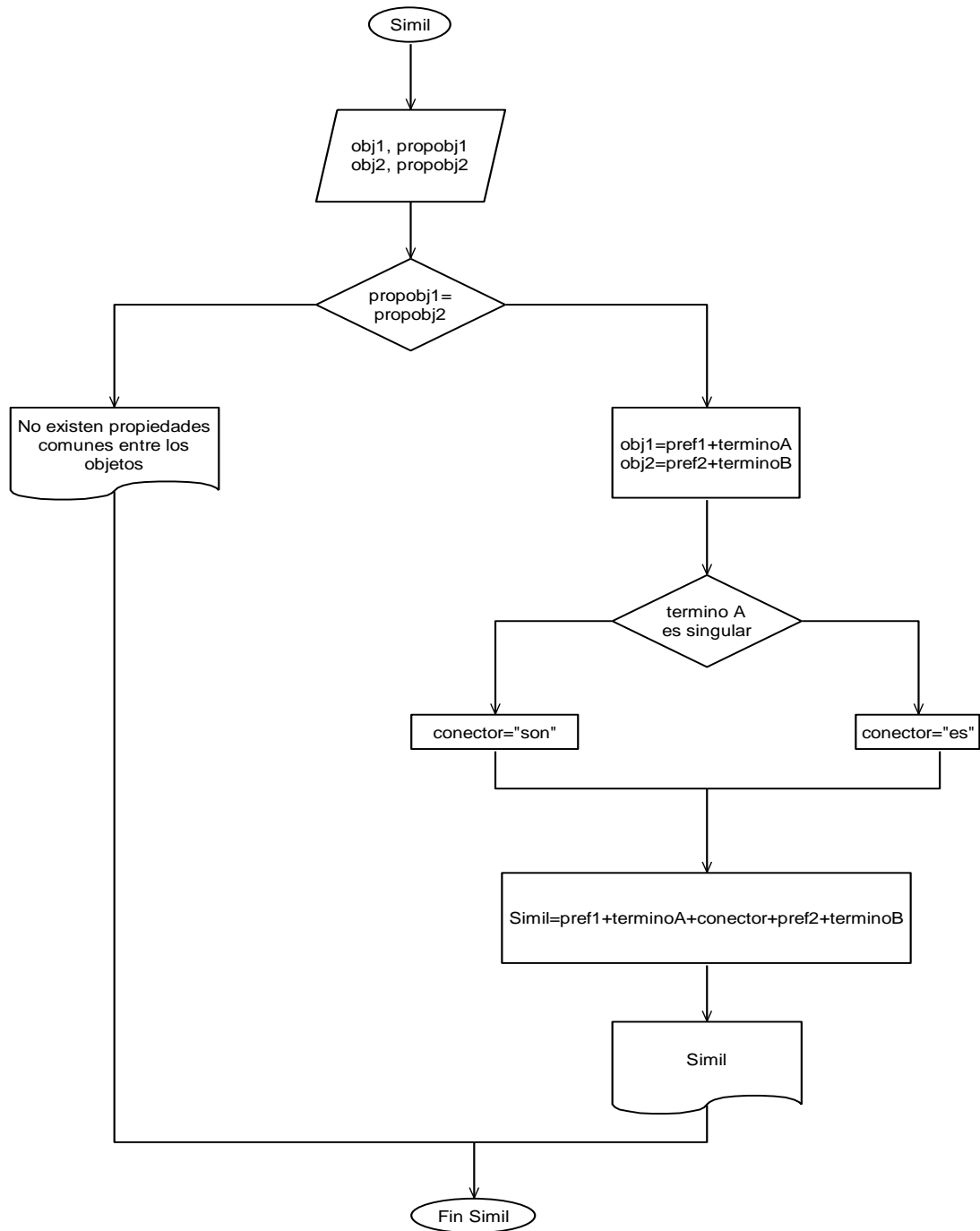


Figura 11. Diagrama de flujo de Falacias 5



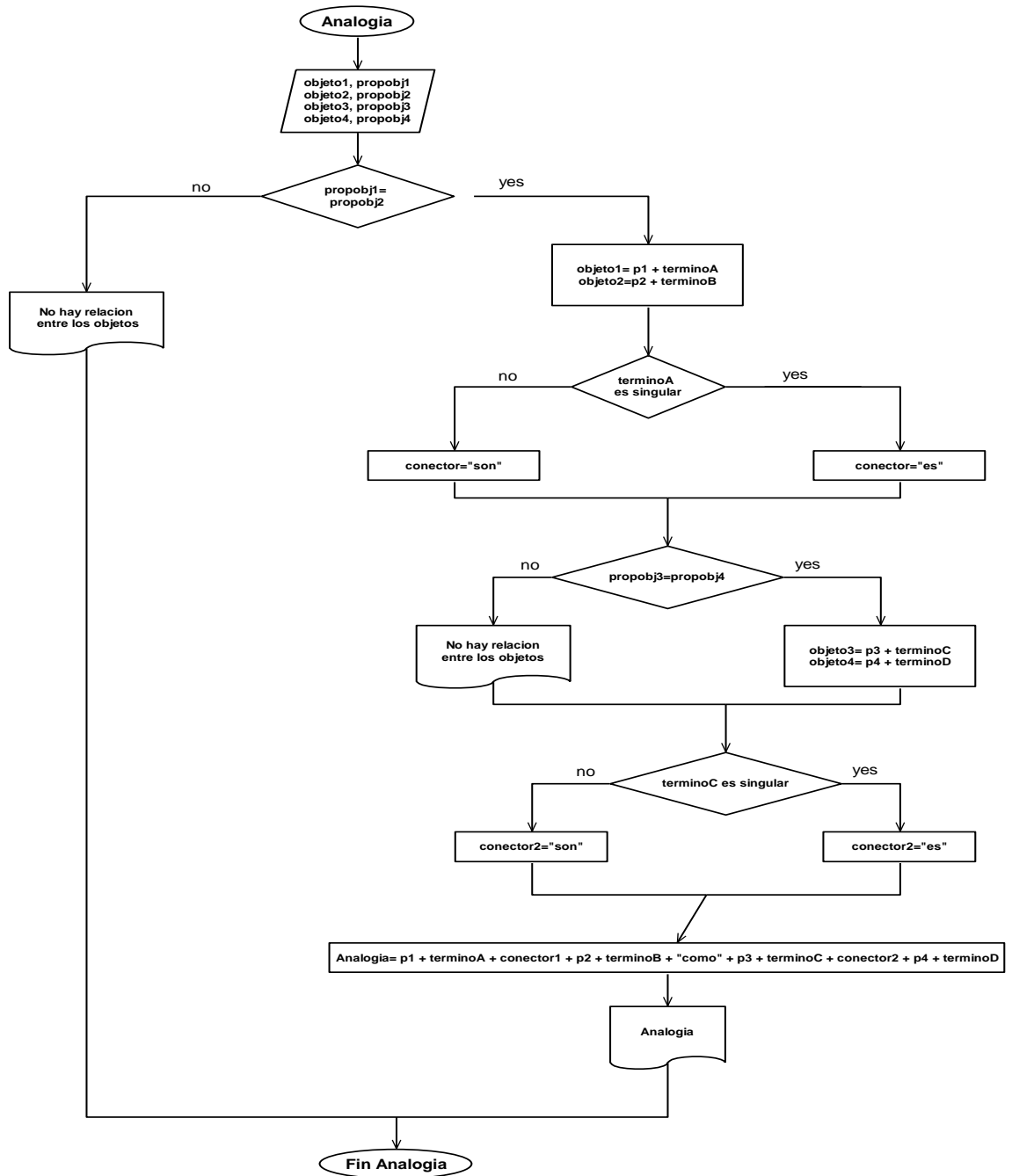
5.5.3 Diagrama de flujo para símiles

Figura 12. Diagrama de flujo de Símil



5.5.4 Diagrama de flujo para analogías

Figura 13. Diagrama de flujo de Analogías



5.5.5 Diagrama de flujo para razonamiento

Figura 14. Diagrama de flujo de Razonamientos 1

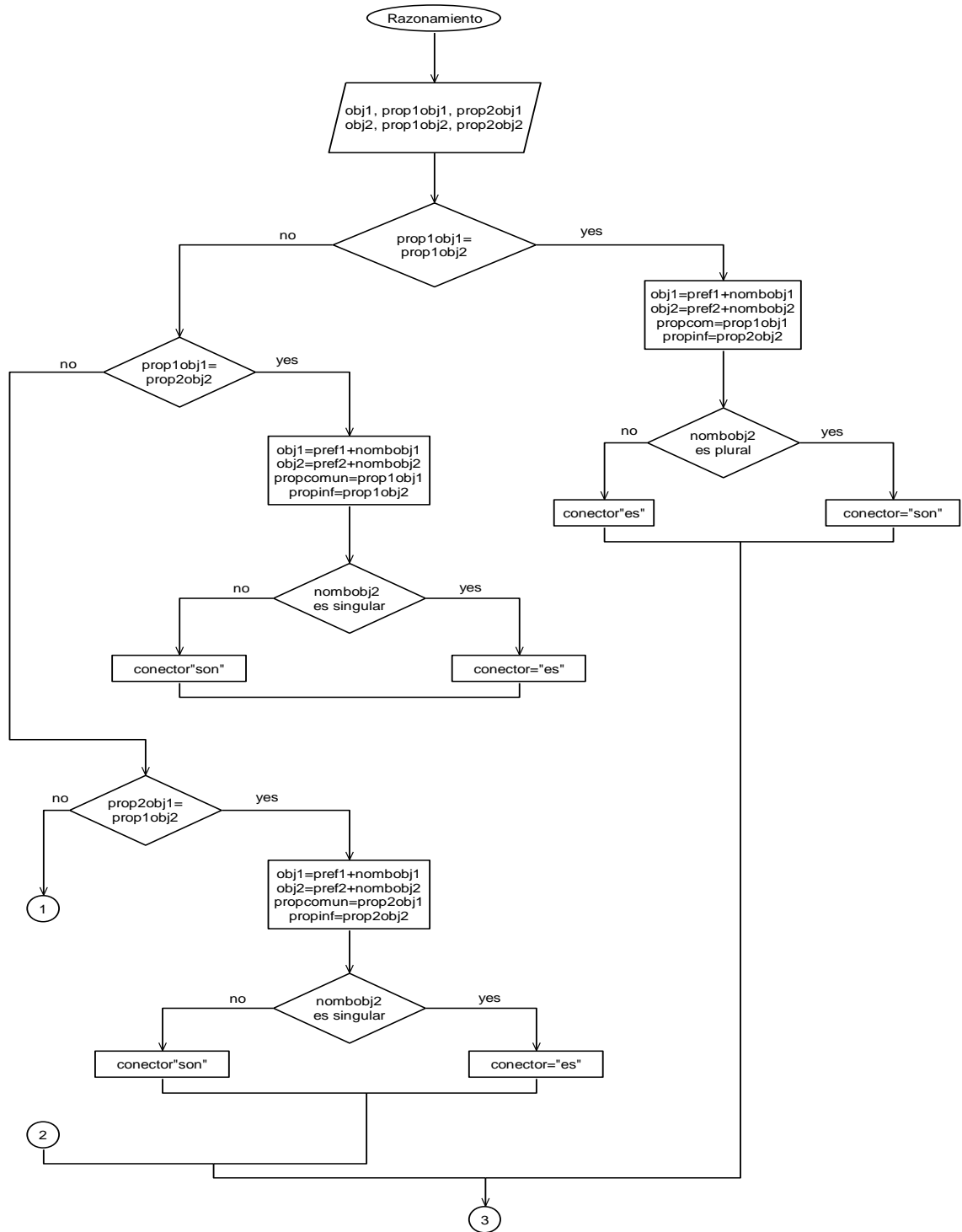
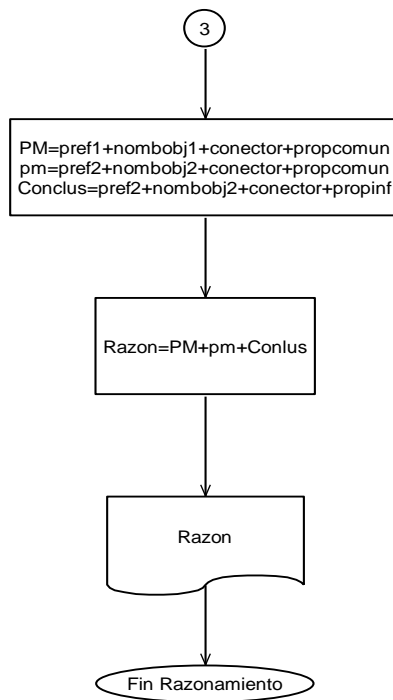
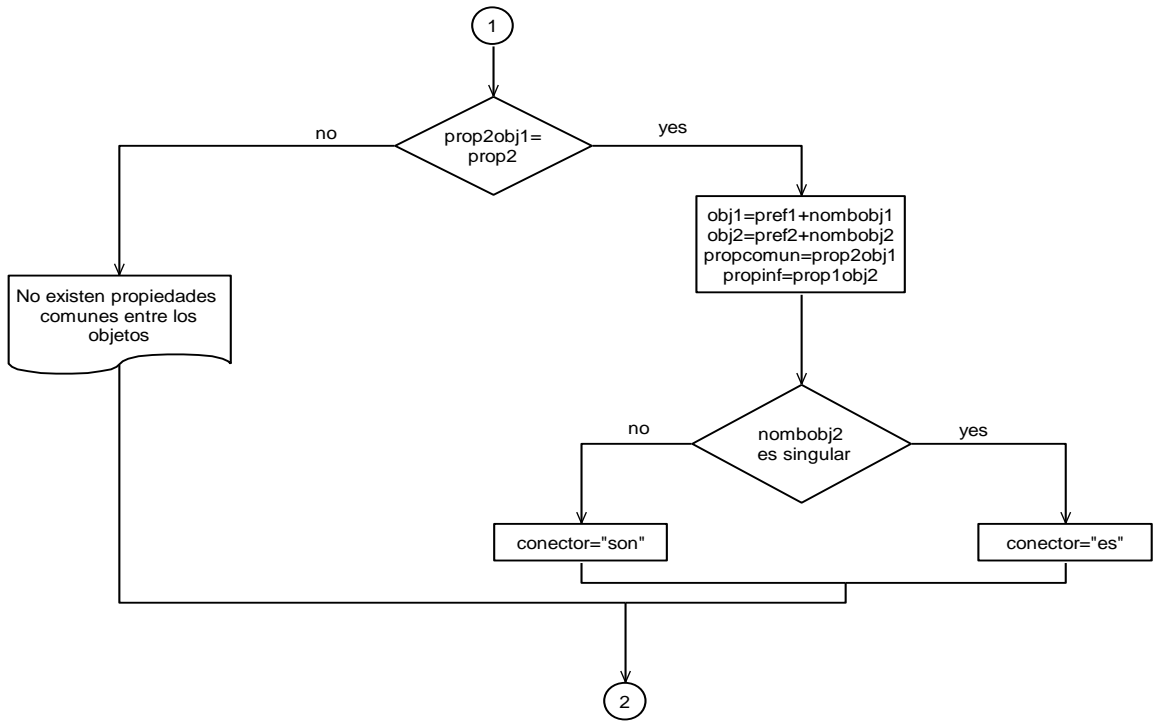


Figura 15. Diagrama de flujo de Razonamientos 2



6. DISEÑO COMPUTACIONAL

En este apartado se encuentra la propuesta para la nueva aplicación web, que contiene los actores que están involucrados con el uso de la aplicación, los casos de uso que los actores realizan en la aplicación, la navegación por ventanas que ilustra cómo es la navegación que el usuario puede hacer en la aplicación y los prototipos de la interfaz de usuario.

6.1 DEFINICIÓN DE ACTORES

En este apartado se encuentra la definición de los actores involucrados en la aplicación junto con su descripción, el tipo de actor y los casos de uso en los que interviene.

Tabla 1. Definición de actores

ACTOR(ES)	
Actor	Estudiante
Casos de Uso	Registrar usuario, Realizar ejercicio, Revisar ejemplo, Revisar teoría, Escoger tema
Tipo	Primario
Descripción	Se encarga de registrarse en la aplicación web para poder acceder a los temas sobre figuras retóricas, así como revisar la teoría sobre cada tema, los ejemplos que existen y poder realizar ejercicios para poder comprobar si el tema ha sido entendido o comprobar que tiene un manejo sobre él.
Actor	Profesor
Casos de Uso	Registrar usuario, Realizar ejercicio, Revisar ejemplo, Revisar teoría, Escoger tema, Revisar ejercicios realizados
Tipo	Primario
Descripción	Se encarga de registrarse en la aplicación web, para poder tener acceso a la teoría, ejemplos y ejercicios sobre figuras retóricas, así como revisar que ejercicios han realizado los

	estudiantes.

6.2 CASOS DE USO

En este apartado se encuentra la descripción de los casos de uso definidos para la aplicación, donde se especifica la narrativa correspondiente a cada caso.

Tabla 2. Narrativa del caso de uso Realizar ejercicio

<i>Nombre Caso de Uso</i>	Realizar ejercicio
<i>Actor(es) Participantes</i>	Iniciado por el usuario
<i>Flujo de eventos</i>	<ol style="list-style-type: none"> 1. Este caso de uso se inicia cuando el usuario debe realizar un ejercicio 2. El usuario selecciona la figura retórica. 3. El usuario completa la información que hace falta sobre el ejercicio propuesto 4. Para completar la realización del ejercicio, se le muestra un mensaje al usuario donde se confirma si la respuesta es correcta o incorrecta.
<i>Precondición</i>	<ul style="list-style-type: none"> • El usuario debe estar registrado en la aplicación web
<i>Sub-flujos</i>	<ul style="list-style-type: none"> • Si no existe ningún ejercicio registrado se le mostrará un mensaje al usuario informándole que no hay ejercicios propuestos.
<i>Poscondición</i>	<ul style="list-style-type: none"> • En el sistema se guardará el ejercicio realizado por el usuario y el usuario tiene la opción de realizar otro ejercicio o regresar a la página principal.

Tabla 3. Narrativa del caso de uso Registrar usuario

<i>Nombre Caso de Uso</i>	Registrar usuario
<i>Actor(es) Participantes</i>	Iniciado por el estudiante o profesor
<i>Flujo de eventos</i>	<ol style="list-style-type: none"> 1. Este caso de uso se inicia cuando un estudiante o profesor deseare registrarse en la aplicación web 2. El usuario ingresa la información solicitada en el formulario de registro 3. Para finalizar el registro del usuario, este confirma los datos ingresados 4. Se le muestra al usuario un mensaje de confirmación de registro, donde se indica que su información se ha registrado exitosamente
<i>Precondición</i>	Ninguna
<i>Sub-flujos</i>	<ul style="list-style-type: none"> • Si en los campos obligatorios del formulario no se ingresa la información, se le mostrara un mensaje que el campo no puede estar vacío • Si el usuario ya se encuentra registrado se le mostrara un mensaje informándole que su información ya se encuentra en la base de datos
<i>Poscondición</i>	<ul style="list-style-type: none"> • El usuario ha sido registrado en la base de datos exitosamente.

Tabla 4. Narrativa del caso de uso Revisar ejemplo

<i>Nombre Caso de Uso</i>	Revisar ejemplo
<i>Actor(es) Participantes</i>	Iniciado por el usuario

<i>Flujo de eventos</i>	<ol style="list-style-type: none"> 1. Este caso de uso se inicia cuando el usuario quiere revisar ejemplos acerca de un tema 2. El usuario selecciona la figura retórica de la cual quiere revisar los ejemplos que hayan 3. Seguido, selecciona la opción de revisar ejemplos 4. Como resultado, se le mostrará los ejemplos sobre el tema que el usuario haya seleccionado 5. Para completar la revisión de ejemplos, el usuario regresa a la página principal o selecciona si ahora desea realizar ejercicios sobre el tema o revisar la teoría disponible sobre el tema.
<i>Precondición</i>	<ul style="list-style-type: none"> • El usuario debe estar registrado en la aplicación web
<i>Sub-flujos</i>	<ul style="list-style-type: none"> • Si no existe ningún ejemplo registrado sobre el tema, se le mostrará un mensaje al usuario informándole que no hay ningún ejemplo para mostrar
<i>Poscondición</i>	<ul style="list-style-type: none"> • El usuario debe seleccionar entre revisar la teoría disponible sobre el tema escogido, realizar ejercicios o salir de la aplicación.

Tabla 5. Narrativa del caso de uso Seleccionar tema

<i>Nombre Caso de Uso</i>	Seleccionar tema
<i>Actor(es) Participantes</i>	Iniciado por el usuario
<i>Flujo de eventos</i>	<ol style="list-style-type: none"> 1. Este caso de uso se inicia cuando el usuario debe seleccionar un tema disponible 2. Al usuario se le muestra un listado con los temas disponibles 3. El usuario selecciona el tema que desea estudiar 4. Para finalizar, al usuario se le muestran las opciones disponibles para el tema seleccionado.
<i>Precondición</i>	<ul style="list-style-type: none"> • El usuario debe estar registrado en la aplicación web

<i>Sub-flujos</i>	<ul style="list-style-type: none"> • Si no existe ningún tema disponible se le mostrará un mensaje avisándole que no existen temas disponibles en el momento.
<i>Poscondición</i>	<ul style="list-style-type: none"> • El usuario debe seleccionar entre revisar la teoría disponible sobre el tema seleccionado, revisar ejemplos sobre el tema seleccionado o realizar ejercicios relacionados con el tema seleccionado.

Tabla 6. Narrativa del caso de uso Revisar teoría

<i>Nombre Caso de Uso</i>	Revisar teoría
<i>Actor(es) Participantes</i>	Iniciado por el usuario
<i>Flujo de eventos</i>	<ol style="list-style-type: none"> 1. Este caso de uso se inicia cuando el usuario desea revisar la teoría de una figura retórica 2. El usuario selecciona la opción de teoría del menú la figura retórica de la cual quiere revisar la teoría 3. Para finalizar, al usuario se le muestra la teoría correspondiente a la figura retórica que selecciono.
<i>Precondición</i>	<ul style="list-style-type: none"> • El usuario debe estar registrado en la aplicación web
<i>Sub-flujos</i>	<ul style="list-style-type: none"> • Si no existe teoría disponible, se le mostrará un mensaje al usuario informándole que en el momento no hay teoría disponible para mostrar.
<i>Poscondición</i>	<ul style="list-style-type: none"> • El usuario debe seleccionar entre revisar los ejemplos disponibles sobre el tema escogido, realizar ejercicios o salir de la aplicación.

Tabla 7. Narrativa del caso de uso Revisar ejercicios realizados

<i>Nombre Caso de Uso</i>	Revisar ejercicios realizados
<i>Actor(es) Participantes</i>	Iniciado por el profesor
<i>Flujo de eventos</i>	<ol style="list-style-type: none"> 1. Este caso de uso se inicia cuando el profesor desea revisar los ejercicios realizados por algún estudiante 2. El profesor selecciona la opción de revisar ejercicios en el menú superior 3. El profesor escoge el estudiante del cual desea revisar los ejercicios que ha realizado 4. Al profesor se le muestra un listado correspondiente a los ejercicios que el estudiante ha realizado, junto con la respuesta dada por el estudiante 5. Para finalizar, el profesor tiene la opción de revisar los ejercicios realizados por otro estudiante o cambiar de opción, usando el menú superior.
<i>Precondición</i>	<ul style="list-style-type: none"> • El profesor debe estar registrado en la aplicación web
<i>Sub-flujos</i>	<ul style="list-style-type: none"> • Si no existen ejercicios realizados por el estudiante seleccionado, se le muestra un mensaje informándole que no hay ejercicios realizados por el estudiante para mostrar.
<i>Poscondición</i>	Ninguna

6.3 NAVEGACIÓN POR VENTANAS

En este apartado se encuentra el diagrama de la navegación por ventanas, que ilustra la navegación que el usuario tiene disponible en la aplicación.

Figura 16. Navegación por ventanas de la aplicación desarrollada para el estudiante

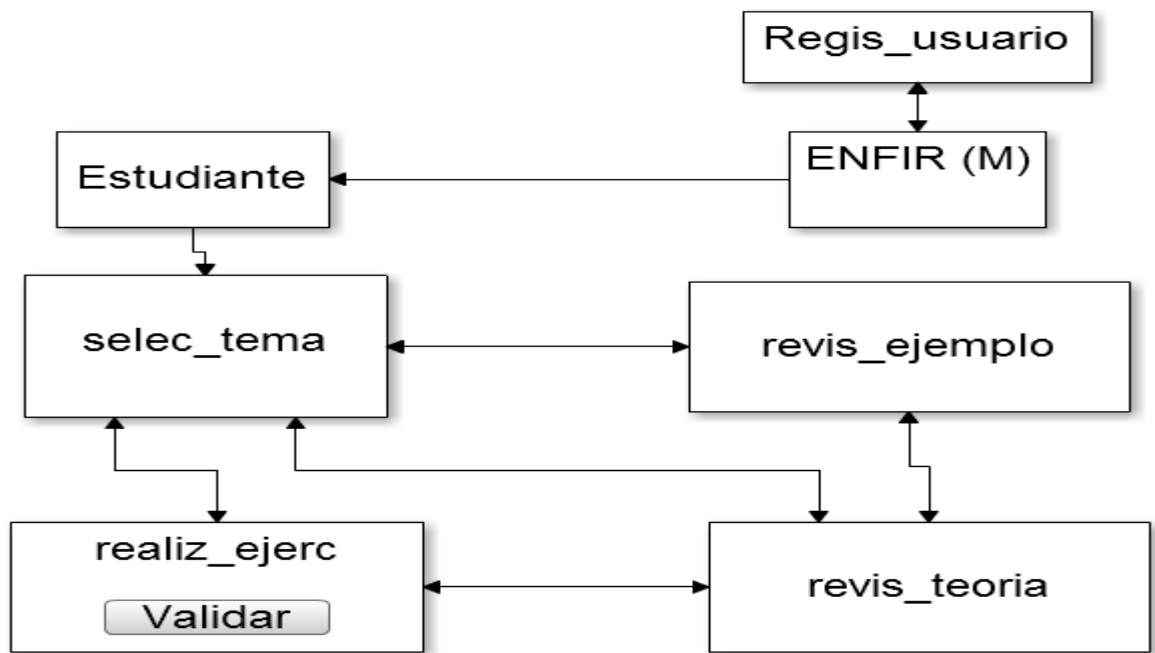
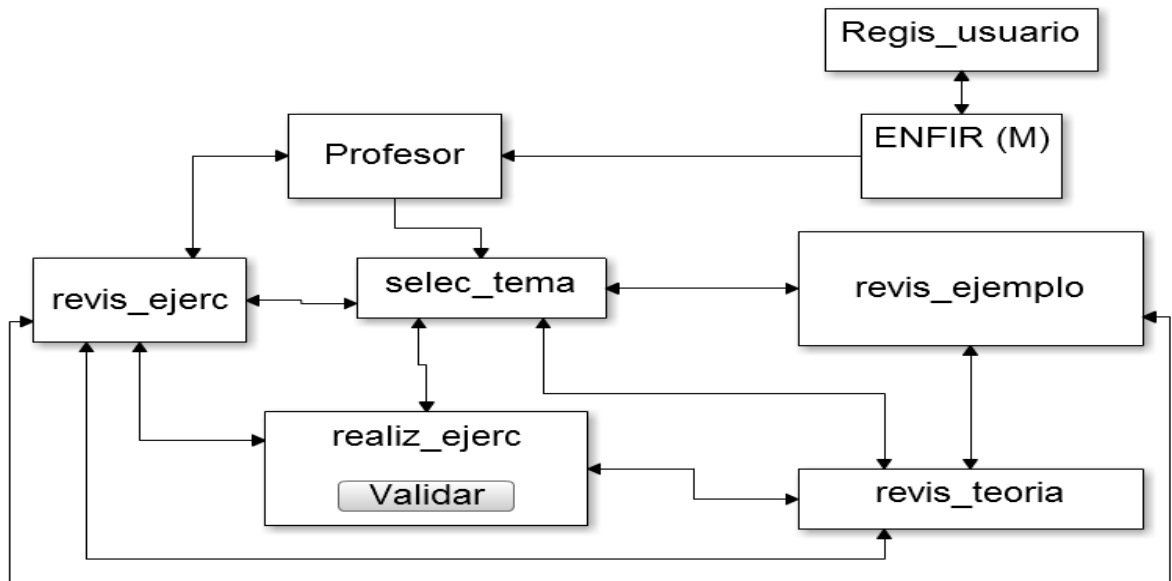


Figura 17. Navegación por ventanas de la aplicación desarrollada para el profesor



6.4 PROTOTIPOS

En este apartado se encuentran los prototipos correspondientes a la propuesta desarrollada para la nueva aplicación web, que consta de la página de registro de usuario, página de inicio de sesión, página principal, página de ejercicios, página de ejemplos, página de ejercicios y página de revisión de ejercicios realizados.

Figura 18. Página de Registro de Usuario

En esta página se encuentra el formulario de registro para nuevos usuarios, donde este tiene que completar ciertos campos que son obligatorios (nombre, usuario, contraseña, confirmar contraseña y debe especificar el rol de registro, ya sea como estudiante o como profesor) y para poder completar su registro en la página debe confirmar que todos los datos ingresados son correctos o en el caso contrario, cancelar el proceso de registro.

Grupo
Filosofía
y enseñanza de la
filosofía...

Inicio Fig. retóricas Silogismo Razonamiento Analogía Falacia Simil

REGISTRO DE USUARIO

* campos obligatorio

Nombre:*

Usuario:*

Contraseña:*

Confirmar contraseña:*

Tipo de Usuario:*

Confirmar Cancelar

Figura 19. Página de Inicio de Sesión

Esta es la página inicial con la que se encuentran los usuarios registrados o no registrados. Los usuarios registrados deben ingresar su usuario y contraseña así como especificar el rol en el cual se encuentran registrados para poder acceder a la página donde aparecen la lista de figuras retóricas; y los usuarios no registrados deben acceder a la opción de “registrarse” que los redirige a la página de registro de usuario para ingresar sus datos y completar su registro en la base de datos.

Grupo
Filosofía
y enseñanza de la
filosofía...

Inicio Fig. retóricas Silogismo Razonamiento Analogía Falacia Simil

Usuario:

Contraseña:

Tipo de Usuario:

[Registrarse](#)

Figura 20. Página Principal

La página principal cuenta con una lista de las figuras retóricas disponibles (silogismo, razonamiento, analogía, falacia y símil). Cada figura cuenta con una lista de opciones, donde el usuario elige entre teoría, que contiene la definición formal de cada figura, ejemplos, donde se encuentran diferentes ejemplos de cada figura, y ejercicios, donde el usuario desarrolla diferentes ejercicios presentados sobre cada figura retórica.

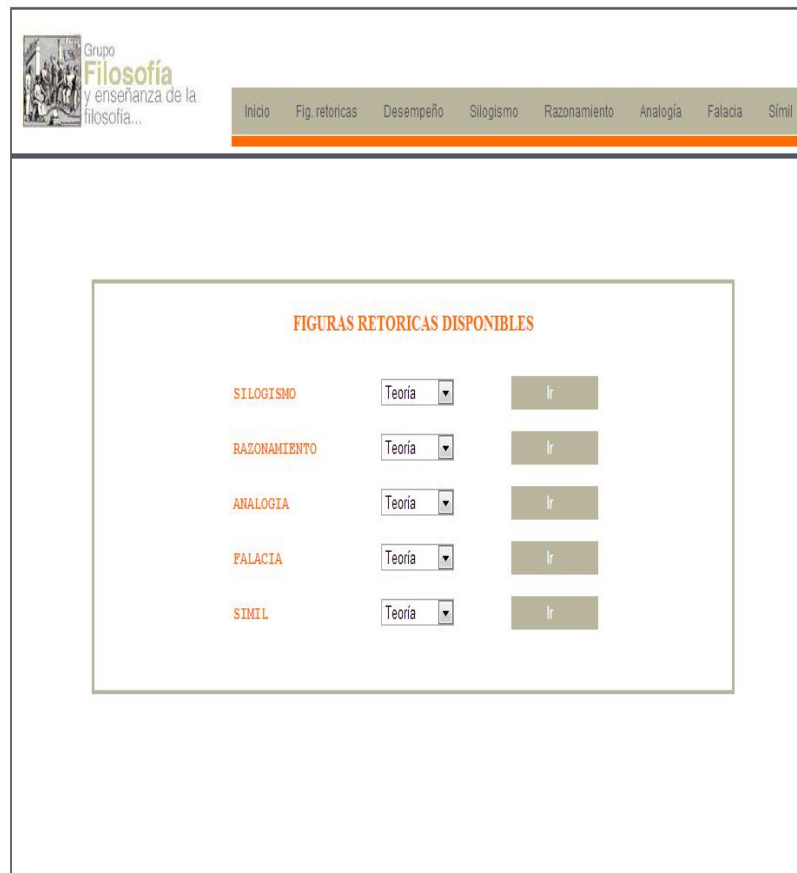


Figura 21. Página 1 de Teoría de Silogismo

En esta página, se encuentra la teoría correspondiente al silogismo, donde se encuentra la definición, los tipos de silogismos y las reglas de los silogismos. A través de esta página, el usuario tiene la opción de revisar los ejemplos disponibles sobre silogismos, realizar ejercicios sobre el tema y utilizando el menú superior, el usuario puede cambiar de actividad, ya sea cambiar de figura, ir a la pagina principal o si lo desea, cerrar la sesión.

Grupo **Filosofía** y enseñanza de la filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Símil

SILOGISMO

Silogismo categórico

El silogismo categórico se define como una forma de razonamiento consistente en tres proposiciones categóricas que contienen tres términos, y solo tres; dos de las proposiciones son las premisas, la tercera es la conclusión. Una inferencia silogística puede interpretarse, en realidad, como una comparación de las relaciones entre cada uno de dos términos y un tercero, para descubrir las que existen entre los dos términos. El término contenido en ambas premisas es el término medio, el predicado de la conclusión el término mayor, y su sujeto el término menor.

El Entimema

Aunque el razonamiento silogístico aparece con frecuencia en el habla cotidiana, no se suele advertir su presencia porque se lo formula de manera incompleta. Un silogismo enunciado incompletamente, en el que una de las premisas o la conclusión está tácita e inexpressada, recibe el nombre de entimema. Un silogismo en el que no se expresa la premisa mayor es un entimema de primer orden. Los entimemas que suprimen la premisa menor son de segundo orden.

Aunque los entimemas no representan una forma nueva de inferencia, su reconocimiento es de gran importancia práctica. Se cree a menudo que las llamadas inferencias inductivas son un modo especial de razonamiento, cuando de hecho se trata simplemente de entimemas de primer orden.

Silogismos hipotéticos, alternativos y disyuntivos

Los silogismos hipotéticos contienen tres proposiciones: la primera premisa, o premisa mayor, es una proposición hipotética; la segunda premisa, o premisa menor, y la conclusión son proposiciones categóricas. Dentro de los silogismos hipotéticos, se encuentran los silogismos hipotéticos puros,

Ejercicios Ejemplos

Figura 22. Página 2 de Teoría de Silogismo

Grupo **Filosofía** y enseñanza de la filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Simil

que contienen dos proposiciones hipotéticas como premisas y otra como conclusión. El valor de este tipo de razonamiento es muy evidente, pues a menudo nos es más fácil establecer la verdad de una proposición hipotética y la de su antecedente que la del consecuente, que puede, pues, ser establecido indirectamente, como conclusión de la inferencia.

Ahora, los silogismos alternativos mixtos, son inferencias válidas que pueden extraerse de una proposición alternativa como premisa mayor y una categórica como premisa menor. Un silogismo alternativo es válido, cuando la premisa menor niega una de las alternantes y la conclusión afirma la verdad de la otra. Esta clase de silogismo se emplea frecuentemente con el objeto de eliminar explicaciones o soluciones propuestas para diversos problemas.

Por último, al hablar de silogismos disyuntivos, se refiere a razonamientos donde la premisa mayor es una disyunción y la menor una proposición categórica; afirmar una disyunción significa que una, por lo menos, de las disyuntivas, es falsa. Un silogismo disyuntivo es válido si la premisa menor afirma una de las disyuntivas y la conclusión niega la otra. La forma esquemática del razonamiento es: no se da el caso de que A sea B y C sea D, A es B; por lo tanto, C no es D. Se dice que la inferencia está en el modus ponendo tollens, pues al afirmar (en la premisa menor), negamos (en la conclusión).

Reglas o axiomas de validez

Estudiando el silogismo como una forma de inferencia por medio de la cual se fija la conexión entre dos términos sobre la base de sus relaciones con un tercer común a ambas, se podrá "ver" que estos axiomas expresan realmente las condiciones de validez. Los axiomas del silogismo categórico se dividen en dos clases: los que tratan de la cantidad o la distribución de los términos, y los que tratan de la calidad de las proposiciones.

Axiomas de cantidad

Ejercicios Ejemplos

Figura 23. Pagina 3 de Teoría de Silogismo

Grupo **Filosofía** y enseñanza de la filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Simil

1. El término medio debe estar distribuido por lo menos una vez.
2. En la conclusión no puede figurar ningún término distribuido que no lo esté en las premisas.

Axiomas de calidad

1. De dos premisas negativas no se obtiene ninguna conclusión.
2. Si una premisa es negativa, la conclusión debe ser negativa.
3. Si ninguna de las premisas es negativa, la conclusión debe ser afirmativa.

Los teoremas generales del silogismo

Teorema I. El número de términos distribuidos en la conclusión debe ser menor por lo menos en una unidad al número total de términos distribuidos en las premisas.

Teorema II. De dos premisas particulares no se desprende ninguna conclusión. Dos premisas particulares pueden ser: a) ambas negativas, b) ambas afirmativas, c) una afirmativa y una negativa.

Teorema III. Si una premisa es particular, la conclusión debe ser particular.

Teorema IV. Si una premisa es particular, la conclusión debe ser particular.

Teorema V. Si la premisa mayor es una proposición particular afirmativa y la menor una proposición universal negativa, no puede haber conclusión.

Ejercicios Ejemplos

Figura 24. Página de Teoría de Razonamiento Lógico

En esta página se encuentra la teoría del razonamiento lógico, donde se encuentra la definición de razonamiento lógico, así como los tipos de razonamiento que existen. A través de esta página, el usuario tiene la opción de revisar los ejemplos disponibles sobre razonamiento lógico, realizar ejercicios sobre razonamiento lógico y utilizando el menú superior, el usuario puede cambiar de actividad, ya sea cambiar de figura, ir a la pagina principal o si lo desea, cerrar la sesión.



Grupo
Filosofía
y enseñanza de la
filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Símil

RAZONAMIENTO LOGICO

El razonamiento es un proceso mediante el cual se obtienen conclusiones a partir de hechos, creencias y normas. El razonamiento es una habilidad del pensamiento por lo que también se llama raciocinio.

Se expresa en la argumentación o conjunto de afirmaciones relacionadas de manera tal que uno de ellos, llamado conclusión, se infiere del o los otros, llamados premisas. El termino razonamiento es el punto de separación entre el instinto y el pensamiento, el instinto es la reacción de cualquier ser vivo. Por otro lado el razonar nos hace analizar y desarrollar un criterio propio, el razonar es a su vez la separación entre un ser vivo y el hombre.

El razonamiento lógico o causal es una operación lógica mediante la cual, partiendo de uno o más juicios, se deriva la validez, la posibilidad o la falsedad de otro juicio distinto. El estudio de los argumentos corresponde a la lógica, de modo que a ella también le corresponde indirectamente el estudio del razonamiento. Por lo general, los juicios en que se basa un razonamiento expresan conocimientos ya adquiridos o, por lo menos, postulados como hipótesis.

Es posible distinguir varios tipos de razonamiento lógico. Por ejemplo el razonamiento deductivo (estrictamente lógico), el razonamiento inductivo (donde interviene la probabilidad y la formulación de conjeturas).

Ejercicios Ejemplos

Figura 25. Página 1 de Teoría de Analogía

En esta página se encuentra la teoría correspondiente a las analogías, donde se explica la definición de analogía, así como las clases de analogías que existen. Por medio de esta página, el usuario puede revisar los ejemplos disponibles sobre analogías. realizar los ejercicios dispuestos sobre el temay utilizando el menú superior, el usuario puede cambiar de actividad, ya sea cambiar de figura, ir a la pagina principal o si lo desea, cerrar la sesión

Grupo **Filosofía** y enseñanza de la filosofía...

Inicio Fig. retoricas Desempeño Silogismo Razonamiento **Analogía** Falacia Simil

ANALOGIAS

Las concepciones y papel de la analogía han variado en la historia de la filosofía. Mientras que para algunos pensadores, como Platón o Santo Tomás, la analogía presenta un tipo de razonamiento específico e indispensable, para otros como los empiristas, ella se limita a afirmar una semejanza bastante débil y sirve para la invención de hipótesis, pero debe ser eliminada en la formulación de los resultados de la investigación científica.

Un teórico de la argumentación constatará que el recurso a la analogía constituye una de las características de la comunicación y del razonamiento no formales, que en algunos casos la analogía podrá ser eliminada en fin de cuentas cuando la conclusión a la cual se llega se resume en una fórmula matemática, pero que muy a menudo sobre todo, ella se encuentra en el centro de una visión original, sea del universo, sea de las relaciones entre el hombre y la divinidad.

Pase lo que pase, para conservarla en su especificidad, será preciso interpretar la analogía en función de su sentido etimológico de proporción. Ella difiere de la proporción puramente matemática, en tanto que no plantea la igualdad de dos relaciones, sino que afirma una semejanza de relaciones. En la analogía se afirma que **a es a b como c es a d**, se trata de una relación cualquiera que se asimila a otra relación. Entre la pareja a-b, el tema de la analogía y la pareja c-d el foro de la analogía, no se afirma una igualdad simétrica por definición, sino una asimilación que tiene por fin aclarar, estructurar y evaluar el tema gracias a lo que se sabe del foro, lo que implica que el foro proviene de un dominio heterogéneo, puesto que es mejor conocido que el del tema.

La analogía de cuatro términos puede expresarse por medio de tres, uno de ellos puede ser repetido en el tema y en el foro; su esquema será: **b es a c como a es a b**.

Para hablar de analogía, primero se debe distinguir entre identidad, igualdad y semejanza. Hablando en sentido estricto, la identidad es la unidad

Ejercicios Ejemplos

Figura 26. Página 2 de Teoría de Analogía

Grupo **Filosofía** y enseñanza de la filosofía...

Inicio Fig. retoricas Desempeño Silogismo Razonamiento **Analogía** Falacia Simil

La analogía es una semejanza imperfecta, que no llega a la igualdad, y por eso contiene también desemejanzas y diferencias.

La analogía propiamente hablando es una forma de predicación. Y la predicación de un nombre común a los varios sujetos a que se atribuye puede realizarse de tres maneras:

- Tomando el nombre con la misma significación en todos los casos, es decir, con significaciones perfectamente semejantes o iguales (predicación unívoca).
- Tomando el nombre con una significación completamente diversa en cada caso, con significaciones que son enteramente desemejantes o diferentes (predicación equívoca).
- Tomando el nombre con una significación en parte semejante y en parte desemejante en cada caso, o sea, con significaciones que son simplemente semejantes, pero no iguales, y por eso entrañan también desemejanzas o diferencias (predicación análoga).

La analogía es, pues, intermedia entre la univocidad y la equívocidad, y por eso participa en cierto modo de esos dos extremos entre los que está colocada. Así, conviene con la univocidad en la unidad del nombre y en la semejanza de las significaciones ligadas a ese nombre, pero difiere de ella en que no se trata de una semejanza perfecta (igualdad), sino de una semejanza imperfecta (con desigualdad). Por otro lado, conviene con la equívocidad en la unidad del nombre y en la desemejanza de las significaciones ligadas a él, pero difiere de la misma en que no se trata de una desemejanza completa, de una diversidad total, sino de una desemejanza parcial, de una desemejanza semejante.

Por lo dicho hasta ahora se ve que la analogía se centra en la significación de los nombres, de ciertos nombres. Pero la significación de un nombre se puede tomar de dos maneras:

Ejercicios Ejemplos

Figura 27. Página 3 de Teoría de Analogía

Grupo **Filosofía** y enseñanza de la filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Símil

- La **analogía simple**, que se conoce con el nombre de analogía de atribución, se da cuando se compara un término con otro, una forma con otra, es decir se trata de semejanza de formas.
- La **analogía compuesta**, que recibe el nombre de analogía de proporcionalidad, se da cuando se compara una relación entre dos términos o formas con otra relación semejante. Se trata de la semejanza de dos o más relaciones o proporciones. Por consiguiente esta analogía exige al menos cuatro términos comparados dos a dos.

El fundamento de la analogía

En sentido real y en sentido lógico. En efecto, todo nombre apunta a un concepto en lo cual consiste su significación, pero el concepto es doble: formal y objetivo.

Formal: La misma representación mental que formamos con nuestro acto de entender.

Objetivo: es algo real. Una esencia o una forma que existe en la realidad y en tanto que existe en ella, a la cual, le acontece el ser concebida o ser representada en un concepto formal.

Estructura de la analogía

Se considera que las partes que constituyen la analogía son el análogo, la trama o relación analógica y el tópico. El análogo es núcleo central de la analogía que representa el mensaje, el conocimiento ya conocido. La trama o relación analógica es el conjunto de relaciones que se establecen para comparar características semejantes de determinadas partes del análogo y del tópico, y el tópico está formado por los contenidos conceptuales, procedimentales o actitudinales desconocidos que se pretenden enseñar, es decir, por el conocimiento o conjunto de conocimientos

Ejercicios Ejemplos

Figura 28. Página 4 de Teoría de Analogía

Grupo **Filosofía** y enseñanza de la filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Símil

Estructura de la analogía

Se considera que las partes que constituyen la analogía son el análogo, la trama o relación analógica y el tópico. El análogo es núcleo central de la analogía que representa el mensaje, el conocimiento ya conocido. La trama o relación analógica es el conjunto de relaciones que se establecen para comparar características semejantes de determinadas partes del análogo y del tópico, y el tópico está formado por los contenidos conceptuales, procedimentales o actitudinales desconocidos que se pretenden enseñar, es decir, por el conocimiento o conjunto de conocimientos de la materia en estudio.

Estructura de una Analogía

La analogía consiste en establecer un conjunto de relaciones para comparar características semejantes de determinadas partes de la representación de las estructuras del análogo y el tópico, y permitir mediante dicha comparación la comprensión del tópico.

Ejercicios Ejemplos

Figura 29. Página de Teoría de Falacia

En esta página se encuentra la teoría correspondiente a las falacias, contiene la definición de falacias, así como su clasificación y explicación de cada tipo de falacia. El usuario tiene la opción de revisar los ejemplos disponibles sobre falacias, realizar ejercicios sobre esta figura retórica y utilizando el menú superior, el usuario puede cambiar de actividad, ya sea cambiar de figura, ir a la página principal o si lo desea, cerrar la sesión



Grupo
Filosofía
y enseñanza de la
filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Símil

FALACIA

Las falacias se definen como errores de razonamiento. En general, las falacias se clasifican en: Puramente lógicas o formales, Semilógicas o verbales, Materiales.

Falacias formales: Se trata de razonamientos que no se ajustan a los tipos de inferencia válida, en relación con los diversos test o reglas que diferencian las formas válidas de razonamiento de las que no diferencian las formas válidas de razonamientos de las que no lo son.

Falacias semilógicas o verbales: Se trata de formas aparente válidas de inferencia, cuyo cuidadoso examen permite advertir que no lo son. Tal apariencia de validez obedece a una ambigüedad, esto es, al uso de la misma palabra o signo verbal para representar dos conceptos diferentes.

El razonamiento parece ser de la forma: A es B, y B es C, por lo tanto, A es C; pero en realidad, es de la forma siguiente: A es B, y D es C, por lo tanto A es C.

Conviene observar que no todo ejemplo de ambigüedad es una falacia. Si consultamos cualquier diccionario erudito de la lengua española, veremos que hay pocas palabras con un significado único.

Hay otras formas de falacias que simplemente mencionaremos como el argumento circular: consiste en introducir en las premisas una proposición que depende de aquella que se discute. La falacia de la pregunta falsa, también llamada falacia de la pregunta múltiple: en la medida en que toda pregunta solicita una información, en lugar de darla, no es una proposición y no puede ser verdadera ni falsa. La falacia del argumentum ad hominem: un recurso muy antiguo pero aun difundido, para negar la fuerza lógica de un argumento (y de este modo crear la apariencia de que se prueba lo opuesto), es injuriar a quien lo expone.

Ejercicios Ejemplos

Figura 30. Página de Teoría de Símil

En esta página se encuentra la teoría correspondiente a los símiles, contiene la definición de símil, así como la explicación del uso de símiles. El usuario tiene la opción de revisar los ejemplos disponibles sobre símiles, realizar ejercicios sobre esta figura retórica y utilizando el menú superior, el usuario puede cambiar de actividad, ya sea cambiar de figura, ir a la pagina principal o si lo desea, cerrar la sesión



Grupo **Filosofía**
y enseñanza de la filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Simil

SIMIL

Simil es un término con origen en el vocablo latino *similis* que hace referencia a lo semejante. El concepto, por lo tanto, se utiliza para establecer una comparación entre dos cosas. Simil es una figura retórica que consiste en la comparación expresa entre una cosa y otra, para dar una idea eficaz de una de ellas. Al establecer la comparación por semejanza, se trasladan las características simbólicas o físicas de uno a otro. Los símiles apelan a elementos de relación como "que", "como" o "cual".

Uso de los símiles

Es una figura muy común en la literatura, en la poesía o el teatro. Es directa y explícitamente expresa el vínculo simbólico entre las dos realidades comparadas, utilizando un término de comparación, algo que hace que se destaque de la analogía de otra figura con la que a menudo se confunden. Es un recurso preferido por el lenguaje poético, muy utilizada por burla o ironía, también se puede avanzar en la discusión y dar a ver la realidad difícil de definir que no sea a través de las imágenes.

Es de suma importancia en el lenguaje, tanto en su frecuencia por su función, opera una reconciliación inesperada e innecesaria entre dos realidades, a priori desconocidos entre sí, pero con una relación de semejanza y contigüidad semántica. "enfátizar las comparaciones semejanzas entre las cosas, pero no cambian el significado de las palabras". Siendo así una unidad fuertemente incrustado en su sintaxis y el contexto del discurso.

Ejercicios Ejemplos

Figura 31. Página de Ejemplos de Silogismo

En esta página se encuentran los ejemplos dispobles sobre silogismos, donde el usuario puede observar el uso de esta figura retórica. Usando el menú superior el usuario tiene diversas opciones, como entrar a revisar las diferentes figuras ya sea su teoría, hacer ejercicios, revisar los ejemplos o de volver a la página de inicio o si lo desea cerrar la sesión.

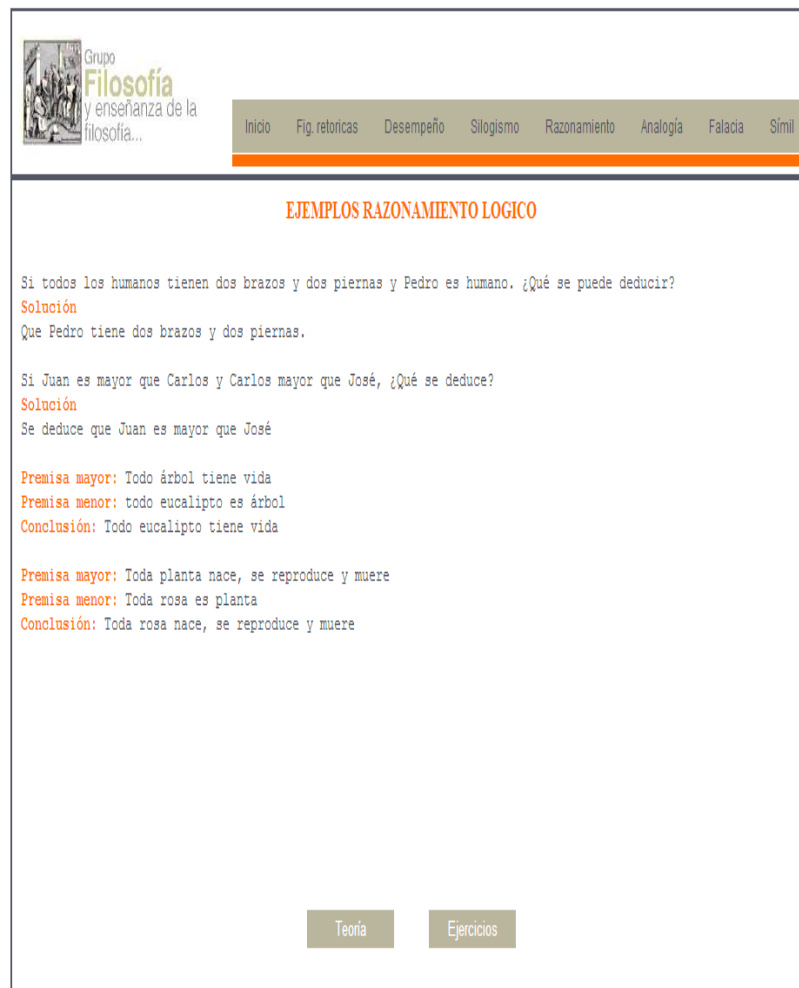
The screenshot shows a web page with a header for 'Grupo Filosofía y enseñanza de la filosofía...'. The navigation menu includes 'Inicio', 'Fig. retoricas', 'Desempeño', 'Silogismo', 'Razonamiento', 'Analogía', 'Falacia', and 'Simil'. The main content area is titled 'EJEMPLOS SILOGISMO' and contains several syllogism examples:

- Todos los caballos tienen huesos
Rocinante es un caballo
Por tanto, Rocinante tiene huesos
- Los planetas son redondos - primera premisa
La Tierra es un planeta - segunda premisa
Por tanto, la Tierra es redonda - conclusión
- Si me duermo no podré concurrir a la sala de teatro.
Si no concurre a la sala de teatro no me voy a entretener.
Conclusión: Si me duermo no me voy a entretener
- Todos los mamíferos son animales.
Todos los hombres son mamíferos.
Conclusión: Todos los hombres son animales
- Ninguna persona sana tiene cáncer.
Algunos guatemaltecos son personas sanas.
Por tanto, algunos guatemaltecos no tienen cáncer.
- Platón fue un gran filósofo - Premisa Mayor
Todos los griegos eran grandes filósofos - Premisa Menor
Platón era griego - Conclusión

At the bottom of the page, there are two buttons: 'Teoría' and 'Ejercicios'.

Figura 34. Página de Ejemplos de Razonamiento Lógico

En esta página se encuentran los ejemplos dispobles sobre razonamiento lógico, donde el usuario puede observar el uso de esta figura retórica y puede tener una idea de que tipo de ejercicios estan disponibles para desarrollar. Usando el menú superior el usuario tiene diversas opciones, como entrar a revisar las diferentes figuras, ya sea su teoría, hacer ejercicios, revisar los ejemplos o de volver a la página de inicio o si lo desea cerrar la sesión.



The screenshot shows a website interface for 'Grupo Filosofía y enseñanza de la filosofía...'. The navigation menu includes: Inicio, Fig. retoricas, Desempeño, Silogismo, Razonamiento, Analogía, Falacia, and Símil. The main content area is titled 'EJEMPLOS RAZONAMIENTO LOGICO' and contains three examples of logical reasoning with their solutions:

Si todos los humanos tienen dos brazos y dos piernas y Pedro es humano. ¿Qué se puede deducir?
Solución
Que Pedro tiene dos brazos y dos piernas.

Si Juan es mayor que Carlos y Carlos mayor que José, ¿Qué se deduce?
Solución
Se deduce que Juan es mayor que José

Premisa mayor: Todo árbol tiene vida
Premisa menor: todo eucalipto es árbol
Conclusión: Todo eucalipto tiene vida

Premisa mayor: Toda planta nace, se reproduce y muere
Premisa menor: Toda rosa es planta
Conclusión: Toda rosa nace, se reproduce y muere

At the bottom of the page, there are two buttons: 'Teoría' and 'Ejercicios'.

Figura 35. Página de Ejemplos de Analogía

En esta página se encuentran los ejemplos dispobles sobre analogías, donde el usuario puede observar el uso de esta figura retórica y puede tener una idea de que tipo de ejercicios estan disponibles para desarrollar. Usando el menú superior el usuario tiene diversas opciones, como entrar a revisar las diferentes figuras ya sea su teoría, hacer ejercicios, revisar los ejemplos o de volver a la página de inicio o si lo desea cerrar la sesión.



Inicio Fig. retóricas Silogismo Razonamiento Analogía Falacia Simil

EJEMPLOS ANALOGIA

La analogía entre un toro y un perro es que son seres mamíferos.

La analogía entre caballo y motocicleta es que son vehículos

La paloma es a la paz, como el cuervo es al mal agüero

El calor es al fuego, como el frío es a la nieve

Ángel es al bien, como demonio es al mal

Médico es a medicina, como abogado es a leyes

INEPTITUD es a TORPEZA como IGUALDAD es a PARIDAD

LAVAR es a ENSUCIAR como PARTICIPACIÓN es a INHICIÓN

Aleta es a pez como brazo es a humano.

Manzana es a árbol como hijo es a padre.

Páginas son a libro como teclas es a piano

Teoría Ejercicios

Figura 36. Página de Ejemplos de Símil

Esta página contiene los ejemplos disponibles para símiles, donde el usuario puede observar como es el uso de un símil y puede comprobar que su estructura teórica concuerde con su estructura escrita en un ejemplos. Usando el menú superior el usuario tiene diversas opciones, como entrar a revisar las diferentes figuras ya sea su teoría, hacer ejercicios, revisar los ejemplos o de volver a la página de inicio o si lo desea cerrar la sesión.



Grupo
Filosofía
y enseñanza de la
filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Símil

EJEMPLOS SIMIL

Tus dientes blancos como perla

Tu cabello sombrío, como una larga y negra carcajada

tan claro como el cristal

Tus frías manos como trozos de hielo me estremecen

murmulo que en el alma se eleva y va creciendo como volcán que sordo anuncia que va a arder

unos cuerpos son como flores

otros como puñales

otros como cintas de agua

pero todos, temprano o tarde

serán quemaduras que en otro cuerpo se agranden

convirtiendo por virtud del fuego a una piedra en un hombre"

Peligrosa como el mar.

Rápido como el viento.

Teoría Ejercicios

Figura 35. Página de Ejemplos de Falacia

En esta página se encuentran los ejemplos correspondientes a falacias, donde se explica de una manera más práctica la estructura de una falacia, así como los diferentes tipos de mostrar una falacia, ya sea con premisas o con una sola frase. Usando el menú superior el usuario tiene diversas opciones, como entrar a revisar las diferentes figuras ya sea su teoría, hacer ejercicios, revisar los ejemplos o de volver a la página de inicio o si lo desea cerrar la sesión.

Grupo **Filosofía**
y enseñanza de la filosofía...

Inicio Fig. retóricas Silogismo Razonamiento Analogía Falacia Símil

EJEMPLOS FALACIAS

Todos los pájaros tienen pico, el ornitorrinco tiene pico por lo tanto es un pájaro.

- **Premisa 1:** Una hamburguesa es mejor que nada.
- **Premisa 2:** Nada es mejor que la felicidad eterna.
- **Conclusión:** Por tanto, una hamburguesa es mejor que la felicidad eterna.

Dices que el coche está aparcado en la primera planta, pero yo creo que está en la tercera, así que debe estar en la segunda

Ejemplo de falacia: Si A, entonces B; si B, entonces A
La gente honrada (A) está en libertad (B).
Yo estoy en libertad (B), así que soy honrado (A).
La primera premisa solo nos da información de qué pasará si se es honrado, pero no dice nada sobre qué sucede si se está en libertad. Solo da información "si A".
Puede no ser honrado pero estar en libertad por no haber sido descubierto y juzgado.

Ejemplo de falacia: Si A, entonces B; si no A, entonces no B.
Si estudio (A) aprobaré (B).
No estudié (no A), entonces suspenderé (no B).
La primera premisa solo nos da información de qué pasará si estudio, pero no dice nada sobre qué sucederá si no estudio.
Solo da información "Si A". En este caso, no hay nada que impida aprobar en caso de no estudiar.

Falacia en una argumentación:
Dios debe existir, porque si no existiera no habría moral y el mundo sería horrible
El jugador hizo todo lo que pudo, porque, si no, no hubiéramos ganado el partido

Figura 36. Página de Ejercicios de Silogismo

En esta página se encuentran los ejercicios disponibles sobre silogismos, donde el usuario debe completar el ejercicio propuesto como se le indique y al confirmar su respuesta, con el botón “validar” se le muestra un mensaje, comunicándole si su respuesta es correcta o incorrecta. El usuario tiene la opción de seguir practicando ejercicios o utilizando el menú superior puede cambiar de actividad, ya sea practicar otra figura, volver a la página principal o si lo desea, cerrar la sesión.

Grupo **Filosofía**
y enseñanza de la filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Símil

deberas, dada la conclusión y una premisa, construir la otra premisa:
Nivel 1 - Episilogismos: Conclusión + Premisa

Premisa mayor
ningún colombiano es ecuatoriano

Premisa menor

Conclusión
ningún quiteño es colombiano

Validar

Figura 39. Página de Ejercicios de Razonamiento Lógico

En esta página, se encuentran los ejercicios disponibles sobre razonamiento, donde el usuario debe completar el ejercicio propuesto como se le indique y al confirmar su respuesta, usando el botón “validar”, se le muestra un mensaje comunicándole si la respuesta fue correcta o incorrecta. El usuario tiene la opción de seguir practicando ejercicios o utilizando el menú superior puede cambiar de actividad, ya sea practicar otra figura, volver a la página principal o si lo desea, cerrar la sesión.

Grupo
Filosofía
y enseñanza de la
filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Simil

dada la conclusión y la primera premisa, deberas construir la segunda premisa:

Nivel 1 - Razonamiento Analógico: Conclusión + primera premisa

"Los zapatos que tenía antes y los que quiero comprar ahora, son de la misma marca."
Zapatos viejos(Buena Calidad, Marca BORSI)
Zapatos nuevos(Marca BORSI)

Primera Premisa:
Los Zapatos viejos son de Marca BORSI

Segunda Premisa:

Conclusión:
Los Zapatos nuevos son de Buena Calidad

Validar

Figura 38. Página de Ejercicios de Analogía

En esta página se encuentran los ejercicios correspondientes a analogías, donde el usuario debe completar el ejercicio propuesto como se le indique y para comprobar si su respuesta fue correcta o incorrecta, debe confirmar su respuesta usando el botón “validar”. El usuario tiene la opción de seguir practicando ejercicios o utilizando el menú superior puede cambiar de actividad, ya sea practicar otra figura, volver a la página principal o si lo desea, cerrar la sesión.

Grupo
Filosofía
y enseñanza de la
filosofía...

Inicio Fig. retóricas Silogismo Razonamiento Analogía Falacia Símil

dado un texto, una pareja y una relación, debes asignar una pareja a a partir de una lista de posibilidades:
Nivel 1 - Analogías: Pareja y relación

"El hombre esta tentado a mentir en algunas situaciones, así como Adán y Eva estuvieron tentados a comer la manzana en el paraiso"

Dada la pareja:
Adan y Eva es/son a La manzana

Relacionada mediante: Tentación

Escoge la pareja que complete una analogia:

- Lujuria - Codicia
- Hombres - mentira
- Adan y Eva - Codicia
- Adan y Eva - manzana
- Adan y Eva - Cain y Abel

Validar

Figura 39. Página de Ejercicios de Símil

En esta página, se encuentran los ejercicios disponibles sobre símiles, donde el usuario debe completar el ejercicio propuesto como se le indique y al confirmar su respuesta, usando el botón “validar”, seguido se le muestra un mensaje comunicándole si la respuesta fue correcta o incorrecta. El usuario tiene la opción de seguir practicando ejercicios o utilizando el menú superior puede cambiar de actividad, ya sea practicar otra figura, volver a la página principal o si lo desea, cerrar la sesión.

Grupo
Filosofía
y enseñanza de la
filosofía...

Inicio Fig. retóricas Desempeño Silogismo Razonamiento Analogía Falacia Símil

dado un texto y los terminos, deberas encontrar la propiedad entre la lista de propiedades mostradas:
Nivel 1 - Símil: Encontrar la relación

"El uniforme de la selección colombiana de futbol tiene la camiseta amarilla, al igual que el de la selección brasilera. Asi también, podemos relacionar Bibliotecas y mesas, mediante su componente primario, o características de países, como Colombia y Venezuela. Un partido político, puede relacionarse con un jardín infantil por una propiedad común."

Símil:
Colombia es a El Uniforme de la selección Colombiana de futbol
Escoge la propiedad que los relacione:

- Soberano
- Camiseta Amarilla
- Madera
- Marca BORSI
- Buena Calidad

Validar

Figura 40. Página 1 de Ejercicios de Falacia

En esta página se encuentran los ejercicios propuestos sobre falacias, donde el usuario debe completar el ejercicio como se le indique y usando el botón “validar” puede comprobar si su respuesta fue correcta o incorrecta. El usuario tiene la opción de seguir practicando ejercicios o utilizando el menú superior puede cambiar de actividad, ya sea practicar otra figura, volver a la página principal o si lo desea, cerrar la sesión.

Figura 41. Pagina 2 de Ejercicios de Falacia

Figura 42. Página de seguimiento a estudiantes

En esta página se encuentra la opción que tienen los usuarios registrados como profesores, de realizar un seguimiento a los usuarios que se encuentran registrados como estudiantes, esta opción busca que los profesores que recomiendan el uso de esta aplicación web, puedan saber qué tipo de ejercicios han realizado sus estudiantes, que han respondido y el resultado obtenido, que refleja si el proceso de enseñanza está siendo correctamente utilizado.

Figura Retorica	Nivel	Respuesta estudiante	Resultado
Silogismos	Nivel 2: Texto + conclusion + premisa	Todos los hombres son animales	Correcto
Silogismos	Nivel 1: Conclusion + Premisa	La tierra es un planeta	Correcto
Analogias	Nivel 1: Relación	Pecado	Incorrecto
Simil	Nivel 1: Encontrar la relación	Camiseta amarilla	Incorrecto
Razonamiento logico	Nivel 1: Conclusion + primera premisa	Todo arbol tiene vida	Correcto

Figura 43. Mensaje de acceso denegado a estudiantes

Este mensaje es mostrado a los estudiantes cuando intentan acceder a la opción de “seguimiento”, que es únicamente permitida para que accedan los usuarios registrados como profesores.



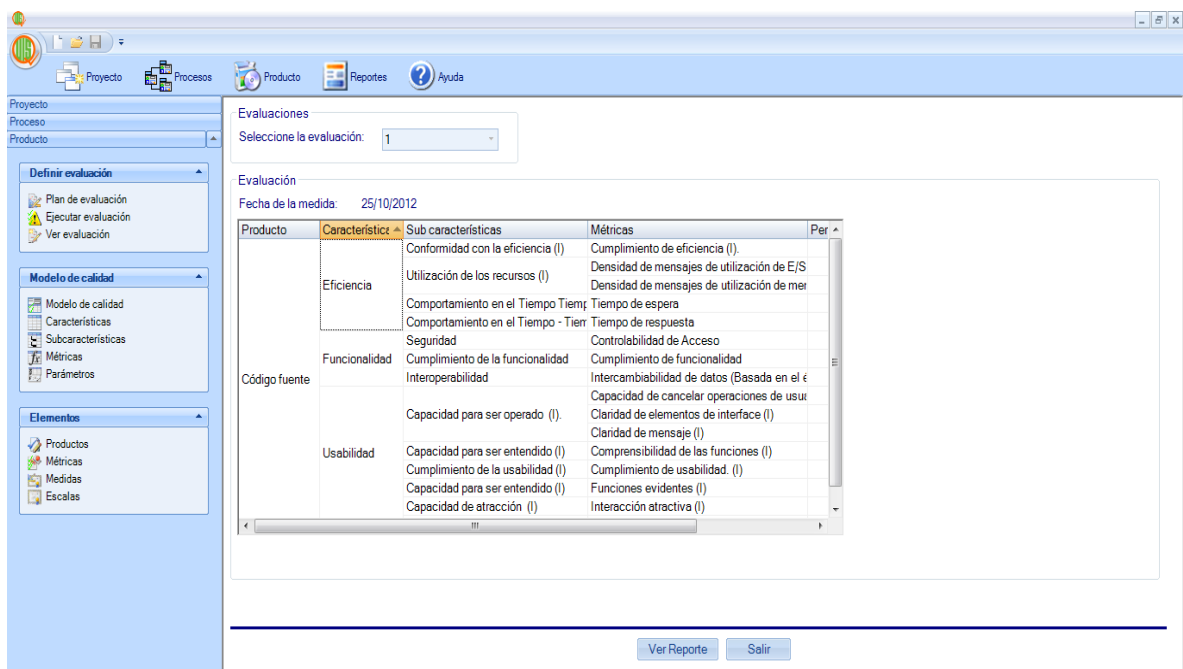
6.5 EVALUACIÓN

Para la evaluación de la aplicación desarrollada se realizó una evaluación de tres características de la aplicación: eficacia, eficiencia y efectividad, utilizando la herramienta para evaluar software llamada QUIS, que fue desarrollada en el grupo STI de la escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander por un estudiante de pregrado como proyecto de grado.

El proceso de evaluación consistió en primer lugar en elegir el producto base para la realización de la evaluación del producto software, que en este caso, se tomo como base el código fuente, en segundo lugar se debía elegir un modelo de

calidad que sirviera como base para la evaluación, en este caso fue la norma ISO 9126. De esta norma ISO, se tomaron tres características para ser evaluados en el producto software: eficiencia, funcionalidad y usabilidad. Según la norma ISO 9126, la eficiencia se define como la capacidad del producto software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas. La funcionalidad de acuerdo a la norma ISO 9126, es la capacidad del producto software para proporcionar funciones que satisfacen las necesidades claras e implícitas cuando se usa bajo condiciones específicas. De acuerdo a la norma ISO 9126, la usabilidad se define como la capacidad del producto software para ser entendido, aprendido, usado y ser atractivo para el usuario, cuando se usa bajo condiciones especificadas.

Figura 44. Elección del producto (código fuente), las características (eficiencia, funcionalidad y usabilidad) y las subcaracterísticas para la evaluación del producto software.



Para la evaluación de cada subcaracterística, se le asignaron valores a las diferentes métricas tomadas en la utilización de la aplicación web, como lo son intercambio de datos, interacción atractiva, tiempo de respuesta, tiempo de espera, así como en la cantidad de mensajes de E/S, comprensibilidad de las funciones, funciones evidentes, la capacidad de cancelar operaciones, la claridad de elementos de interfase, la claridad de mensajes, la verificación de validez de entrada y la controlabilidad de acceso.

Figura 45. Asignación de métricas a cada subcaracterística (I)

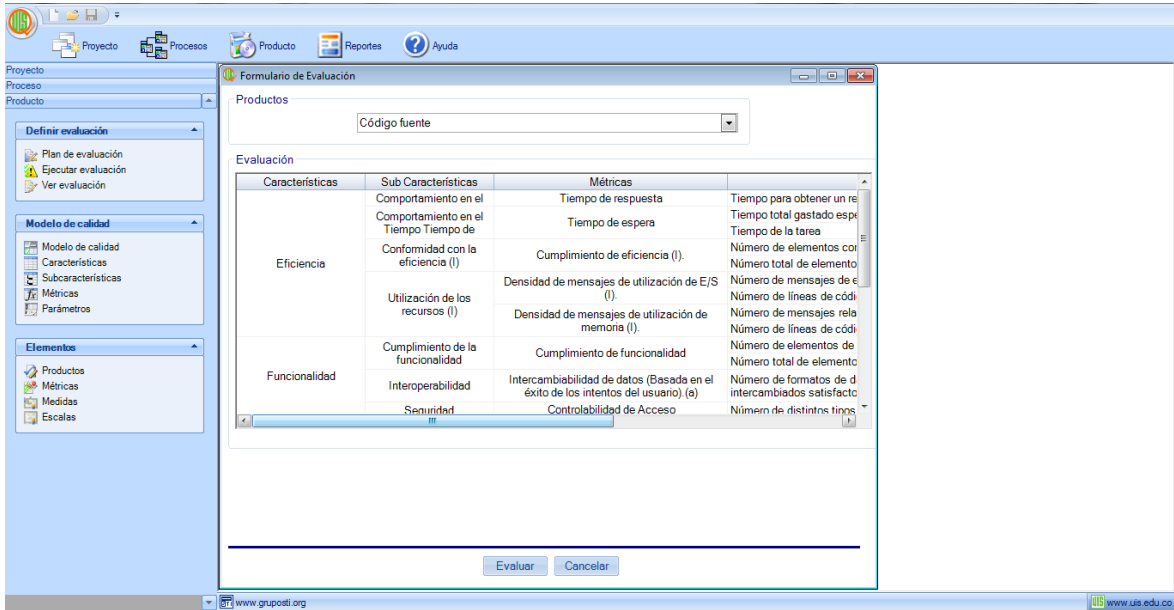


Figura 46. Asignación de métricas a cada subcaracterística (II)

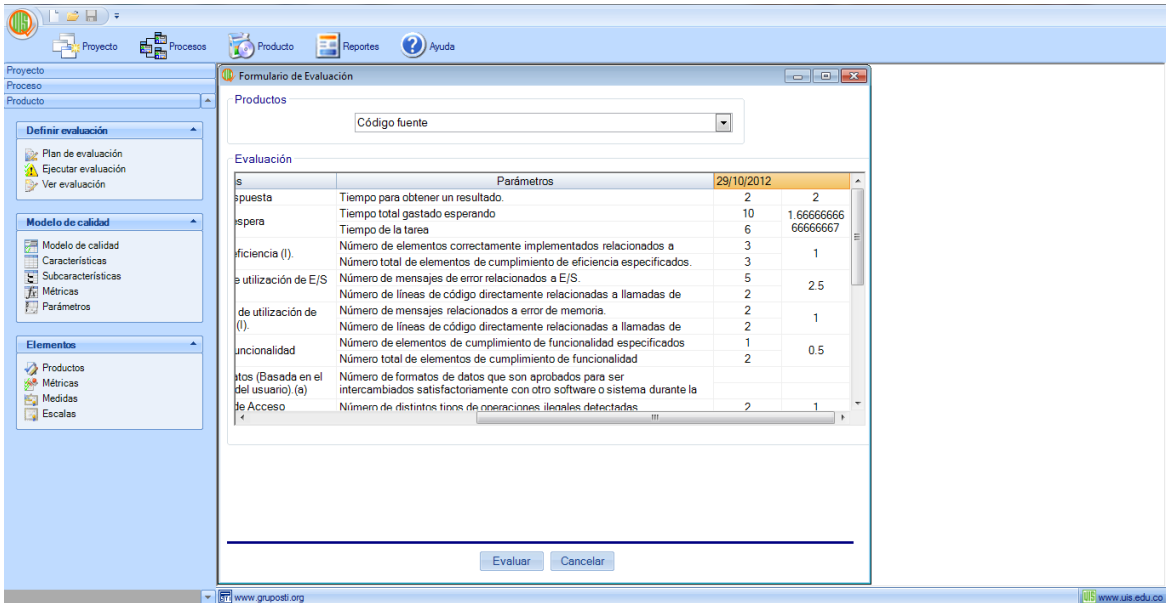


Figura 47. Asignación de métricas a cada subcaracterística (III)

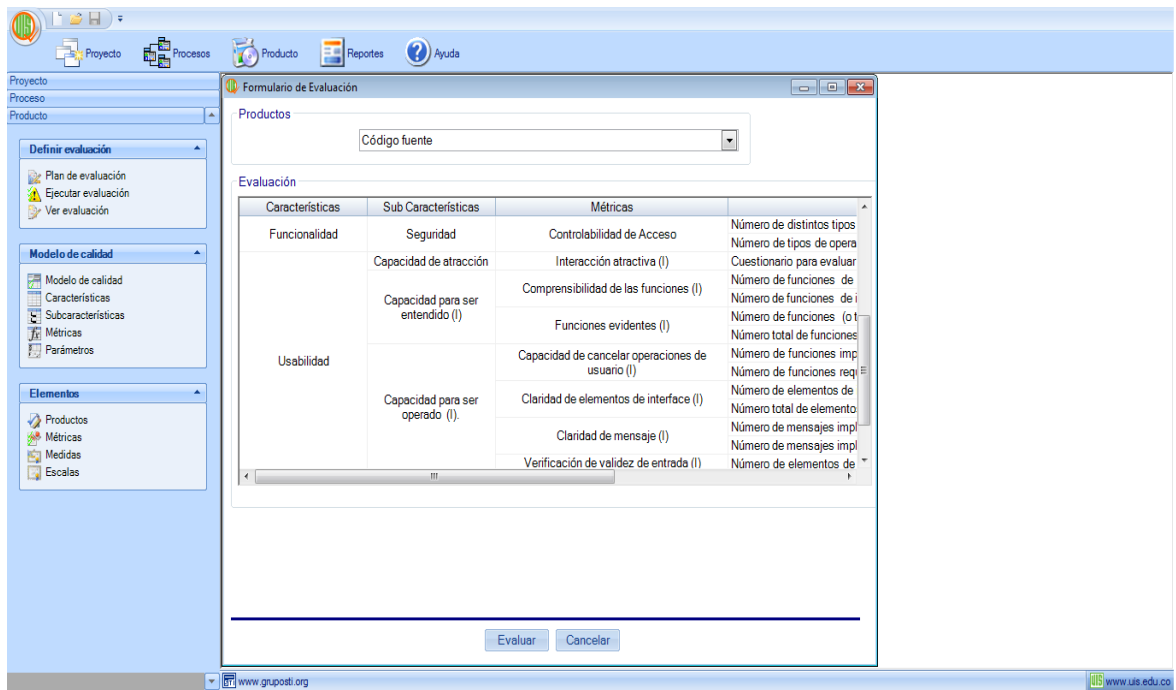


Figura 48. Asignación de métricas a cada subcaracterística (IV)

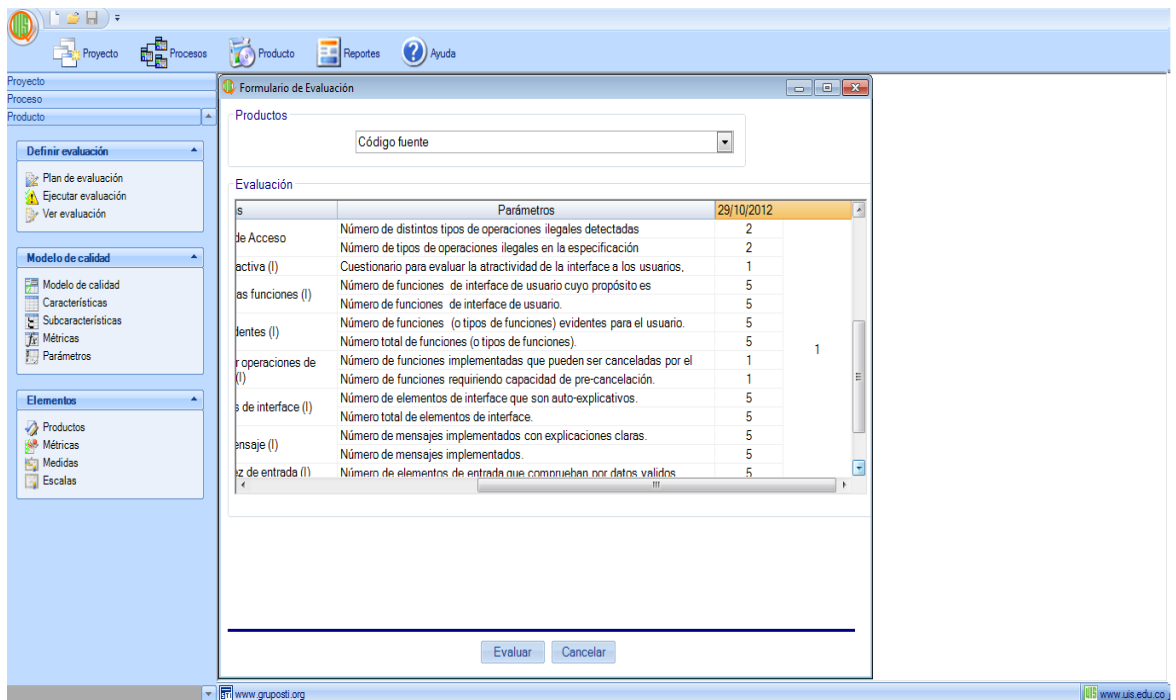


Figura 49. Asignación de métricas a cada subcaracterística (V)

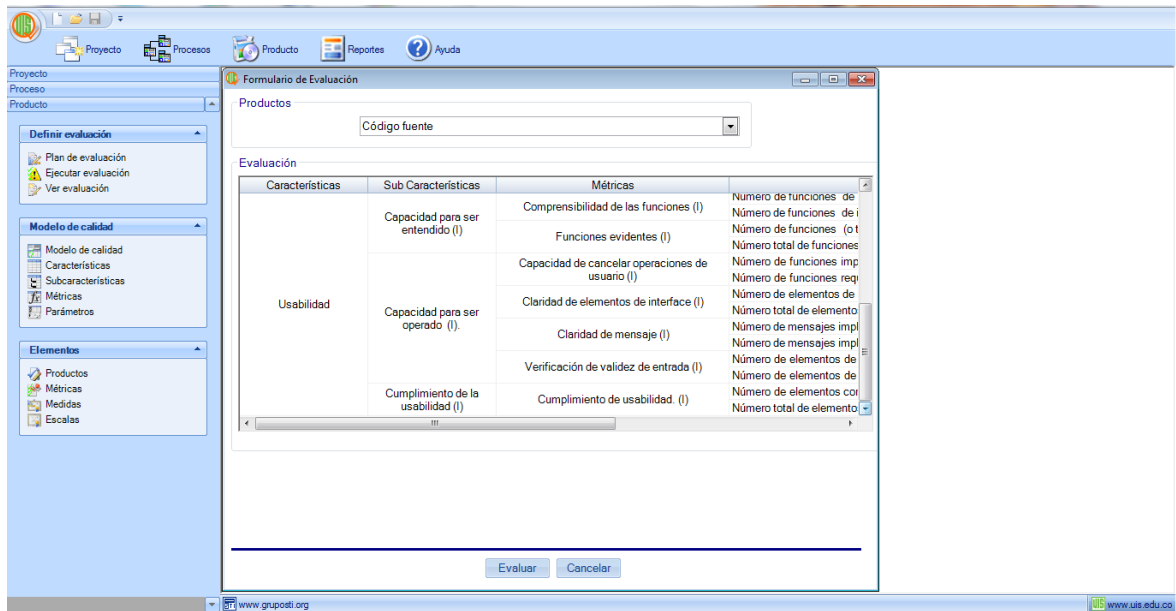
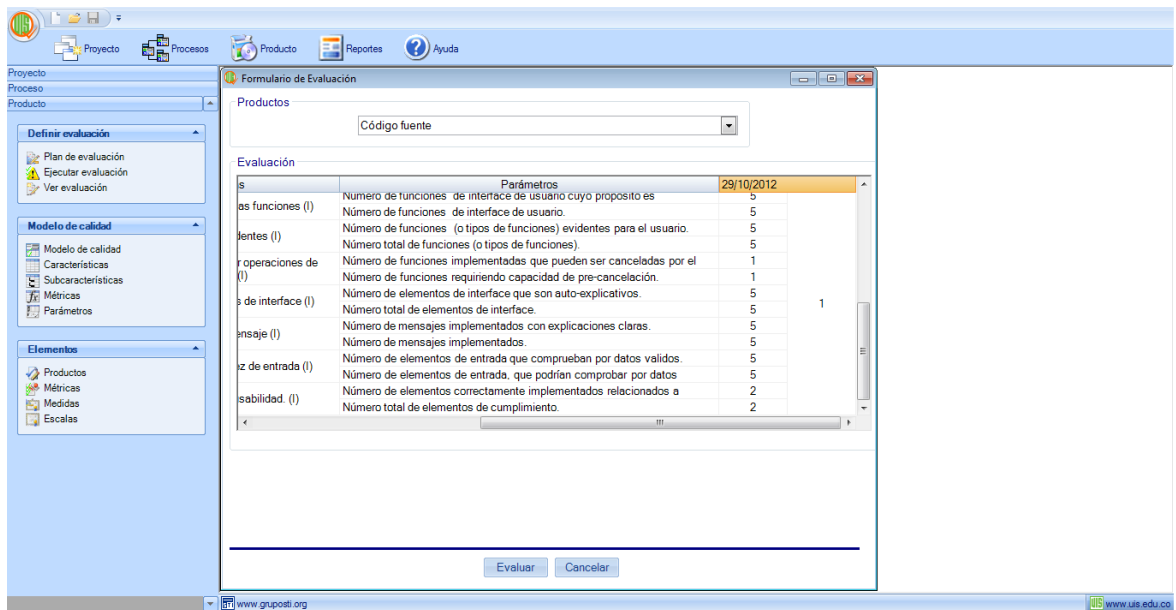


Figura 50. Asignación de métricas a cada subcaracterísticas (VI)



El proceso de evaluación finaliza cuando el software QUIS muestra una pantalla con los resultados de la evaluación e identifica de los valores ingresados en las métricas cuales están dentro (mostrados en color verde) y cuales están por fuera (mostrados en color rojo) de los límites permitidos.

Figura 51. Resultados obtenidos de la evaluación de la aplicación web desarrollada usando el software QUIS

ForVerEvaion

Evaluaciones

Seleccione la evaluación: 1

Evaluación

Fecha de la medida: 29/10/2012

Producto	Características	Sub características	Métricas	Permitido	Medido
	Eficiencia	Comportamiento en el Ti	Tiempo de respuesta	2	2
		Comportamiento en el Ti	Tiempo de espera	2	1.667
		Conformidad con la eficie	Cumplimiento de eficienc	1	1
	Utilización de los recurso	Densidad de mensajes de	Densidad de mensajes de	15	2.5
		Densidad de mensajes de	Densidad de mensajes de	20	1
	Funcionalidad	Cumplimiento de la funcio	Cumplimiento de funcio	1	0.5
		Interoperabilidad	Intercambiabilidad de dat	1	0
Código fuente	Seguridad	Controlabilidad de Acces	Controlabilidad de Acces	1	1
		Capacidad de extracción (Interacción atractiva (I)	1	1
	Usabilidad	Capacidad para ser enten	Comprensibilidad de las fi	1	1
		Funciones evidentes (I)	Funciones evidentes (I)	1	1
		Capacidad de cancelar op	Capacidad de cancelar op	0.4	1
		Claridad de elementos de	Claridad de elementos de	1	1
		Claridad de mensaje (I)	Claridad de mensaje (I)	1	1
	Verificación de validez de	Verificación de validez de	1	1	
	Cumplimiento de la usabil	Cumplimiento de usabili	1	1	

Ver Reporte Salir

www.gruposti.org www.us.edu.co

En los resultados obtenidos se muestra que las métricas de algunas subcaracterísticas tienen una diferencia mínima con los límites permitidos y que otras se encuentran dentro de los valores que el software QUIS considera como aceptables.

CONCLUSIONES

- Se cumplió con el objetivo de determinar la estructura lógica para la operación de figuras retóricas representados en los diagramas de flujo correspondientes a cada figura, los cuales representan el funcionamiento real de las figuras retóricas.
- Se diseñó una estrategia didáctica para el aprendizaje de figuras retóricas, que contiene actividades donde el usuario realiza una revisión de teoría para consolidar los fundamentos sobre cada figura, complementado con la muestra de ejemplos y la realización de ejercicios que permiten que el usuario pruebe sus conocimientos y así contribuir con su aprendizaje de figuras retóricas.
- Se diseñó una interfaz de usuario tanto para estudiantes como para profesores que mediante una evaluación, se obtuvieron resultados satisfactorios donde se pudo comprobar su efectividad, eficiencia y eficacia garantizando que la estrategia didáctica utilizada para el aprendizaje de figuras retóricas se considera exitosa.
- Se cumplió con el objetivo de diseñar una aplicación web donde se utiliza un motor de inferencia para la enseñanza de figuras retóricas, que permite a los usuarios la realización de ejercicios y que brinda la oportunidad que los profesores puedan revisar los resultados de la realización de ejercicios por parte de los estudiantes.
- El desarrollo de este proyecto fue útil a sus autoras en la formación como ingenieras de sistemas, ya que permitió poner en práctica las herramientas y conocimientos adquiridos durante la carrera en ingeniería del software, bases de datos, lenguajes de programación, desarrollo de herramientas informáticas y habilidades en el planteamiento y solución de problemas.
- Con la realización de este proyecto, se pudo evidenciar la importancia que tienen estas herramientas tecnológicas como apoyo en la enseñanza brindada por profesores y en su interacción con los estudiantes.

RECOMENDACIONES

- Se recomienda implementar una función que permita al usuario profesor personalizar los contenidos para el usuario estudiante, como la creación y personalización de cursos donde los estudiantes puedan inscribirse y tratar temas de interés sobre las figuras retóricas.
- Se recomienda implementar en la aplicación web algunas tecnologías como blogs, chats, que permitan una mayor comunicación y discusión de temas de interés por parte de los usuarios.
- Se recomienda actualizar constantemente la base de datos con nuevos ejercicios para las prácticas de los temas por parte de los usuarios y así contribuir a que el proceso de enseñanza siga avanzando.

BIBLIOGRAFÍA

- Bellman, R. E. (1978). *An introduction to Artificial Intelligence: Can Computers Think?* San Francisco: Boyd & Fraser Publishing Company.
- Bruner, J. (2000). *Educación: Escenarios de Futuro. Nuevas Tecnologías y Sociedad de la Información. PREAL: Promoción de la Reforma Educacional en América Latina y el Caribe*. Retrieved from <http://www.preal.cl/brunner16.pdf>
- C. Choua, T. C. (2002). *Refining the learning companion: the past, present, future of educational agents, Computer & Educacion*.
- DIDÁCTICO, T. M. (n.d.). Retrieved from <http://www.grupoblas Cabrera.org/didactica/pdf/Analogias%20y%20uso%20didactico.pdf>
- Florencia. (n.d.). *Definición de símil*. Retrieved from <http://www.definicionabc.com/comunicacion/simil.php>
- González, T. M. (n.d.). *ESTRUCTURAS DE LAS ANALOGÍAS Y SU USO DIDACTICO*. Retrieved from <http://www.grupoblas Cabrera.org/didactica/pdf/Analogias%20y%20uso%20didactico.pdf>
- Gros, B. (1997). *Diseños y programas educativos Pautas pedagógicas para la elaboracion de software* . Barcelona.
- León, C. (Junio de 2011). *Tipos de Razonamiento* .
- López, J. G. (n.d.). Retrieved from <http://dspace.unav.es/dspace/bitstream/10171/1882/1/05.%20Jes%C3%BA s%20GARC%C3%8DA%20L%C3%93PEZ,%20La%20analog%C3%ADa%20en%20general.pdf>
- Nagel, M. C. (n.d.). *Introducción a la lógica y al método científico. I Lógica formal*. Buenos Aires: Amorrortu editores.
- Nagel, M. C. (n.d.). *Introducción a la lógica y al método científico. II Logica aplicada y metodo científico*. Buenos Aires: Amorrortu editores.
- Negroponte. (1995). *El Mundo Digital*. Barcelona: Ediciones B.

Perelman, C. (1997). *El imperio retórico. Retorica y argumentacion*. Editorial Norma S.A.

Proyecto Aula Literatura. Figuras literarias. Teoría. (n.d.). Retrieved 09 01, 2012, from http://lenguayliteratura.org/ltr/index.php?option=com_content&task=view&id=1654&Itemid=1

V. Juliñ, V. B. (2000, Mayo-Junio). *Agentes Inteligentes: el siguiente paso en la Inteligencia Artificial* . Retrieved Septiembre 01, 2012, from <http://www.ati.es/novatica/2000/145/vjulia-145.pdf>

Vargas-Quesada, P. H. (1999, Abril). *El profesional de la información. Agentes inteligentes: definicion y tipologia. Los agentes de informacion*. Retrieved Septiembre 01, 2012, from http://www.elprofesionaldelainformacion.com/contenidos/1999/abril/agentes_inteligentes_definicion_y_tipologia_los_agentes_de_informacion.html

Yagüe, A. A. (n.d.). *Técnicas de Intelgencia Artificial aplicadas al entorno educativo basadas en Juegos*. Retrieved from <http://www.it.uc3m.es/jvillena/irc/descarga.htm?url=practicass/08-09/12.pdf>