



**Universidad Central
“Marta Abreu” de las Villas.**

**Facultad Matemática Física y Computación
Licenciatura en Ciencia de la Computación**

TRABAJO DE DIPLOMA

Título: Aplicación de las Ontologías a la Enseñanza.

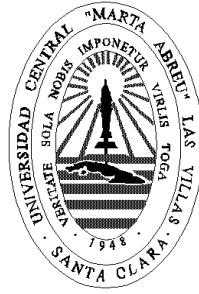
Autor: Claudia Balcells Chang.

Tutores: MSC. Manuel Castro Artilles

Dra. Luisa González González

Santa Clara 2006

“Año de la Revolución Energética en Cuba”



Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencias de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

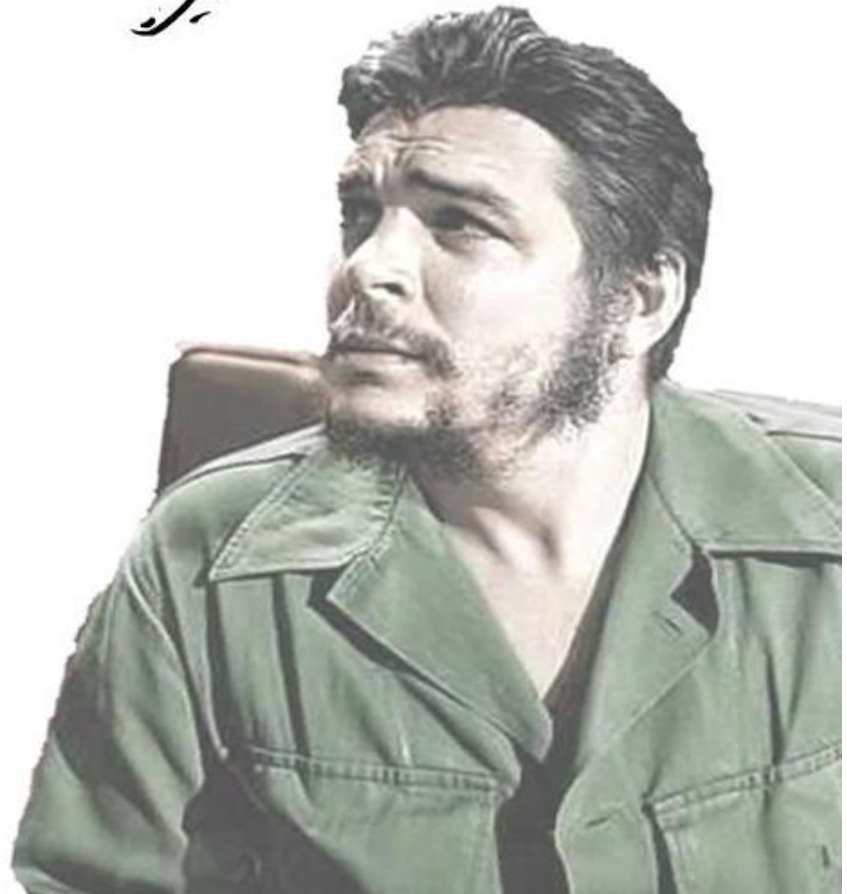
Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Seminario

"Tenemos que lograr el técnico de alta capacitación y gran experiencia, el hombre que adquiriera una gran experiencia en trabajo práctico, porque en los libros no se aprende nada más que hasta determinado nivel o determinada disciplina".

ستار



A mis padres, por su confianza, apoyo y guía.

A mis abuelos, Eoty y Pilingo, por su constante preocupación, y ocupación, a pesar de la distancia.

A tía Maritza, por su apoyo total e incondicional.

A Ely, la hermana que nunca tuve y siempre quise.

Agradecimientos.

En este momento de recuento quiero agradecer, no sólo a las personas que hicieron posible la realización de este trabajo, sino a todos aquellos que me apoyaron y ayudaron durante toda mi etapa estudiantil que culmina con la conclusión de mis estudios universitarios.

A mis padres, a mis abuelos Toty y Pilingo, a tía Maritza y a Elizabeth; la familia que siempre me apoyó.

A “tío Enriquito”, mi padrino.

A la amiga de siempre, Zadielyn.

A Zailín y a Tinito por prestarme la máquina en momentos de apuros y por sus meriendas.

A Pepe y a Aleida, por permitir tanto fastidio y brindar tanta ayuda incondicional.

A Carlos por siempre estar allí para apoyarme y a su familia, por aguantar mis malacrianzas y por su preocupación.

A Graciela, mi profesora de siempre.

A Tomy, por su ayuda.

A mis profesores Ramiro y Carlos García, por su excelente pedagogía.

A Ary y Moraima, por aceptar que las hiciera correr con mis trabajos.

A mis tutores Castro y Mony, que aunque me hicieron correr bastante, estuvieron presentes en el momento oportuno.

A mis amigas de penurias: Magyana, Anay, Liana, Dunia, Lis, Yuriany, Liset, Eylín, Roxana, Maybel, Aylín y Norma.

A Abdel y Alain, por responder a mis preguntas.

A los amigos Ruper, Body, Fonty y Yervilla por darme aliento.

A Jesús Enrique y Dalia, a Ruble y Milaida por ser amigos incondicionales.

A la familia Villalonga – Sánchez, por su constante preocupación.

A Lupe, que aunque ya no esté con nosotros, está en mi corazón.

A Cary y a Raulito por tenderme sus manos amigas en los momentos de necesidad.

Y no por último menos importante, a todos mis compañeros de año, por habernos mantenidos unidos hasta la recta final.

Resumen:

Actualmente, se impone la necesidad de brindar productos de software que proporcionen facilidades en el estudio de las asignaturas, ya sea para la enseñanza presencial o a distancia. Por dicha razón el presente trabajo realiza un estudio de las ontologías e implementa un modelo de enseñanza para asignaturas teórico-prácticas utilizando la herramienta Protégé.

En el trabajo se hace el estudio del estado del arte en el área de las ontologías, se ofrecen varias definiciones y clasificaciones que se encuentran en la literatura y se determina la categoría de la aplicación que se realiza.

Se realiza un análisis de las principales aplicaciones que se utilizan en la enseñanza asistida por computadoras, y se caracteriza cada una de ellas. Se esbozan las particulares distintivas de las aplicaciones para la educación que se realizan con ontologías, así como se plantea y describe un modelo para la enseñanza utilizando ontologías. Se particulariza cada uno de los dominios explicando que significa cada entidad, así como sus interrelaciones y atributos.

El trabajo explica como se utiliza la herramienta Protégé, describiendo los pasos a seguir para realizar cualquier acción con la misma. Se hace una valoración de la herramienta, destacando sus características más importantes.

Abstract:

Today, there is the need of software products in order to facilitate the study of different subjects, whether the teacher is physically present or not. Therefore, this paper conveys data of a research on the ontologies and carries out a teaching model for theoretical and practical subjects using the tool Protégé.

The paper also comprises a study of the condition of art within the ontologies; various classification and definition criteria in literature and an analysis of the main applications in teaching assisted by computers along with a characterization of each one of them.

The distinctive features of the applications for education using the ontologies are drawn and a teaching model is described. Each dominion is individually explained by giving every entity's meaning, their relationships with one another and their attributes.

The paper shows the steps to follow in the use of Protégé and a valuation by highlighting its most functional and useful characteristics.

Índice

Introducción.....	1
Capítulo 1. Ontologías.....	5
1.1. Concepto de ontología.....	5
1.1.1. Algunas definiciones de ontologías.....	7
1.2. Clasificación de ontologías.....	8
1.2.1. Clasificación por grado de axiomatización.....	11
1.2.2. Clasificación por dependencia del contexto.....	12
1.3. Algunos aspectos importantes de las ontologías.....	12
1.3.1. Composición de una ontología.....	13
1.3.2. Cuestiones a tener en cuenta para diseñar ontologías.....	13
1.3.3. Conceptos claves en relación a las ontologías.....	14
1.3.4. Otros conceptos.....	14
1.4. Aplicaciones, características y lenguajes de las ontologías.....	15
1.4.1. Aplicaciones de las ontologías.....	15
1.4.2. Beneficios de las ontologías.....	16
1.4.3. Características de las ontologías.....	16
1.4.4. Algunos lenguajes de ontologías.....	17
1.5. Comparación entre Tesoros, Mapas Conceptuales y Ontologías.....	19
1.5.1. Tesoros y ontologías.....	19
1.5.1.1. Definiciones de tesoro.....	19
1.5.1.2. Similitudes entre ontologías y tesoros.....	20
1.5.1.3. Diferencias.....	20
1.5.2. Mapas conceptuales.....	21
1.5.2.1. Definiciones de mapas conceptuales.....	22
1.5.3. Comparación entre mapas conceptuales, tesoros y ontologías.....	22
1.6. Conclusiones del capítulo.....	24

Capítulo II. Herramienta protégé y el modelo de una asignatura teórico-práctica.....	26
2.1. Aplicación de la tecnología en la Enseñanza.....	26
2.1.1. Sistemas de enseñanza.....	27
2.1.2. Las facilidades que ofrecen las ontologías.....	27
2.2. Ontología para el modelo de una asignatura teórico-práctica.....	28
2.3. Descripción del modelo.....	29
2.3.1. El dominio conceptual.....	29
2.3.1.1. Entidades del Dominio conceptual.....	29
2.3.1.2. Relaciones del dominio conceptual.....	29
2.3.2. El dominio instruccional.....	31
2.3.2.1. Entidades del dominio instruccional.....	31
2.3.2.2. Relaciones del dominio instruccional.....	32
2.3.3. El dominio didáctico.....	32
2.3.3.1. Atributos didácticos.....	33
2.3.3.2. Relaciones didácticas.....	33
2.3.4. Modelo de información del metamodelo.....	34
2.4. Herramienta Protégé.....	36
2.4.1. Algunas características de la herramienta Protégé.....	36
2.4.2. Algunos aspectos de las bases de datos orientadas a objetos.....	37
2.4.3. Facilidades del Protégé.....	38
2.5. Modelado de la ontología con la herramienta Protégé.....	38
2.6. Conclusiones del capítulo.....	46
Capítulo III. Manual de usuario del protégé.....	48
3.1. Creación de un proyecto en el Protégé.....	48
3.1.1. Inicio del Protégé.....	48
3.1.1.1. Acceder a los proyectos más recientes.....	49
3.1.1.2. Abrir un proyecto existente.....	49
3.1.1.3. Crear un proyecto nuevo.....	50
3.1.2. Nuevo proyecto en Protégé.....	51

3.2. Clases.....	53
3.2.1. Jerarquía de clases.....	53
3.2.2. Crear una nueva clase.....	54
3.3. Los conceptos.....	55
3.4. Las instancias.....	57
3.5. Las consultas.....	59
3.6. Manual de usuario del entorno educativo.....	63
3.7. Conclusiones del capítulo.....	68
Conclusiones.....	69
Recomendaciones.....	70
Bibliografía.....	71

Introducción:

Los problemas que plantea en la actualidad la enseñanza universitaria son de sobra conocidos. Por un lado el esquema de enseñanza tradicional basado en la combinación de conferencias en pequeños grupos y la supervisión personalizada ha dejado de ser práctico debido a la masificación que han sufrido las universidades. Por otro lado, el modelo de transmisión del conocimiento mediante las conferencias ha demostrado ser insuficiente y alejado de los resultados de las teorías actuales de aprendizaje que propugnan procesos de aprendizaje activo, que favorezcan la construcción personal de conocimiento.

El esfuerzo en el desarrollo de sistemas computacionales para la enseñanza ha trasladado este modelo a los escenarios educativos siguiendo el paradigma de considerar la computadora como “tutor”. Se trata de generar una estrategia de enseñanza en función del seguimiento del aprendizaje y de los resultados que alcanza cada alumno.

A falta de un sistema que permita la realización de una enseñanza individualizada y adaptada a las necesidades de cada alumno, nos planteamos la idea de mejorar el material telemático existente, considerando las limitaciones de la misma, en el contexto de la enseñanza masificada en el que se encuentra actualmente la Universidad.

Un entorno de enseñanza computacional se puede entender como un sistema informático diseñado para el desarrollo de actividades que permiten la adquisición de conocimiento. En él se exponen los conocimientos de la materia y se proponen eventualmente la realización de tareas que ejercitan el contenido.

La creación de ontologías para la representación de dominios de conocimiento en la enseñanza o para el diseño de entornos educativos, proviene principalmente de la creación de dominios de conocimiento que utilizan los sistemas ITS (Sistemas Tutores Inteligentes) y también, en menor medida de los sistemas de autoría hipertextuales que contienen dominios de objetos con relaciones estructurales entre ellos.

La ontología constituye por tanto un marco de referencia en cuanto a la conceptualización que describe los elementos que componen un dominio de conocimiento. En este sentido en lo referente a la reusabilidad y la intercambiabilidad de componentes educativos en el ámbito de la estandarización, la ontología representa más bien un vocabulario común con el que referir elementos de dominios de conocimiento educativo tanto para la representación del conocimiento, así como para el diseño de herramientas de autor.

Hipótesis de la Investigación:

Formular las particularidades de los conceptos asociados a cada disciplina para facilitar su modelación, para así encontrar un modelo general que permita representar los contenidos de una materia para organizar su estudio, utilizando el apoyo de las ontologías, que es factible para el soporte conceptual de este modelo. Crear modelos de conocimiento para elevar el nivel de abstracción a un nivel pedagógico y desarrollar una herramienta basada en el modelo obtenido.

A partir de lo anteriormente planteado surgen los siguientes objetivos:

Objetivo general:

Estudio y desarrollo de una ontología que implemente un modelo de enseñanza utilizando la herramienta Protégé, en donde se describan los dominios del modelo.

Objetivos específicos:

- Estudio de las Ontologías de forma general y en particular, los principales resultados asociados con los dominios docentes.
- Revisar las características de herramientas profesionales actuales.
- Formular un modelo para representar el dominio de los conceptos de una disciplina.
- Obtener una herramienta para instrumentar el modelo propuesto.
- Validar la herramienta con una asignatura de la carrera.

El trabajo se ha estructurado en 3 capítulos:

En el Capítulo 1 se presentan varias definiciones y clasificaciones de las ontologías, así como varias de sus características.

En el Capítulo 2 se realiza un análisis de las principales aplicaciones que se utilizan en la enseñanza asistida por computadoras, además se plantea y describe un modelo para la enseñanza utilizando la herramienta Protégé.

En el Capítulo 3 se explica como se utiliza la herramienta Protégé, así como una valoración de la herramienta, destacando sus características más importantes y se explica un pequeño entorno educativo, utilizando el Protégé.

**Capítulo I:
ONTOLOGÍAS.**



Capítulo I. ONTOLOGÍAS.

Cada vez más, se impone la necesidad de ofrecer productos de software que faciliten el estudio de las asignaturas, ya sea para la enseñanza presencial o a distancia. Dada esta situación la carrera de Ciencias de la computación requiere que se trabaje en este sentido pues los productos disponibles son escasos y de propósito específico.

Por esto se propone formular las particularidades de los conceptos asociados a cada disciplina para facilitar su modelación. Se tratará de encontrar un modelo general que permita representar los contenidos de una materia para organizar su estudio. Para ello es factible el apoyo de las ontologías para el soporte conceptual de este modelo, por lo que se realiza su estudio.

En este capítulo se aborda como objetivo principal los aspectos fundamentales referentes a las ontologías y sus características. Para esto se ha estructurado de la siguiente forma:

- En el epígrafe 1.1 se presenta el concepto de ontología y las definiciones dadas de la misma.
- En el epígrafe 1.2 se exponen las clasificaciones de las ontologías.
- En el epígrafe 1.3 se muestra algunos aspectos importantes que nos ayudaran a comprender mejor las ontologías.
- En el epígrafe 1.4 se presentan las aplicaciones, características y lenguajes de las ontologías.
- En el epígrafe 1.5 se comparan a las ontologías con los mapas conceptuales y con los tesauros.

1.1. Concepto de ontología.

Las ontologías, hacen posible una semántica para construir los metadatos que se usan para estructurar el contenido. Un metadato no es más que un dato estructurado sobre la información, o sea, información sobre información, o de manera más simple, datos sobre datos. Los metadatos en la Web, son datos que se pueden guardar, intercambiar y procesar por medio del ordenador y están estructurados de tal forma que permiten ayudar a la

identificación, descripción, clasificación y localización del contenido de un documento o recurso web, y por tanto, también sirven para su recuperación.

Se trata de cambiar la información en conocimiento, mediante unas estructuras de conocimiento formalizadas (las ontologías) que se refieren a los datos, por medio de los metadatos, bajo un esquema común normalizado sobre algún dominio del conocimiento. Los metadatos no sólo detallarán el esquema de datos que aparecerá en cada instancia, sino que también podrán contener información adicional de cómo hacer deducciones sobre ellos, es decir, cómo establecer axiomas que podrán, a su vez, aplicarse en los diferentes dominios que trate el conocimiento almacenado. De esta manera, los buscadores obtendrán información al compartir los mismos esquemas de anotaciones web y los agentes de software no sólo encontrarán la información precisa, sino que podrán realizar suposiciones de forma automática buscando información relacionada con la que está situada en las páginas web y con las exigencias de las consultas ejecutadas por los usuarios. Además, los productores de páginas y servicios web podrán intercambiar sus datos siguiendo estos esquemas comunes e, incluso, podrán reutilizarlos.

Las ontologías proceden del campo de la Inteligencia Artificial; son vocabularios comunes para las personas y aplicaciones que trabajan en un dominio. Según el Grupo de Trabajo en Ontologías del consorcio W3C, una ontología define los términos que se usan para describir y representar un cierto dominio. Se utiliza la palabra "dominio" para denotar un área específica de interés o un área de conocimiento (física, aeronáutica, medicina, contabilidad, fabricación de productos, etc.) Toda ontología representa cierta visión del mundo con respecto a un dominio. Por ejemplo, una ontología que defina ser humano como "espécimen vivo o muerto correspondiente a la especie Homo sapiens; primate bípedo que pertenece a la familia de los homínidos, como los chimpancés, gorilas y orangutanes" expresa una visión del mundo totalmente distinta a la de una ontología que lo defina como "sujeto consciente y libre, centro y vértice de todo lo que existe; todos tienen la misma dignidad, pues han sido creados a imagen y semejanza de Dios" [1].

En las aplicaciones reales, sin embargo, una ontología es una identidad computacional, y no se debe considerar como una entidad natural que se descubre, sino como recurso

artificial que se crea (Mahesh, 1996). Una ontología se comprende que es un conocimiento común y compartido de un dominio, que se puede comunicar entre científicos y sistemas computacionales. La última característica, (se pueden compartir y reutilizar en aplicaciones diferentes), explica en parte el gran interés suscitado en los últimos años en la creación e integración de ontologías.

1.1.1. Algunas definiciones de ontologías.

Existen numerosas definiciones de ontologías, entre las que cabe destacar:

- "Un instrumento que define los términos básicos y relaciones a partir del vocabulario de un área de conocimiento, así como las reglas de combinación de estos términos y relaciones para definir extensiones a un vocabulario". (Neches, 1991).
- "Una ontología es la especificación de una conceptualización". (Gruber, 1993). (Aquí el término conceptualización se refiere a un modelo conceptual).

El sinónimo más usual de ontología es conceptualización. En esta definición, convertida ya en estándar, conceptualización se refiere a un modelo abstracto de algún fenómeno del mundo del que se identifican los conceptos que son relevantes; de forma explícita, se hace referencia a la necesidad de detallar de forma consciente los distintos conceptos que conforman una ontología, de manera formal se indica que la especificación debe representarse por medio de un lenguaje de representación, formalizado y compartido refleja que una ontología debe, en el mejor de los casos, dar cuenta de conocimiento aceptado (como mínimo, por el grupo de personas que deben usarla).

- "Una ontología es una especificación formal de una conceptualización compartida". (Borst, 1997). (Aquí el término forma se refiere a que es procesable por ordenador).
- Una definición de ontología más concreta la ofrece Weigand (1997): "Una ontología es una base de datos que describe los conceptos generales o sobre un dominio, algunas de sus propiedades y cómo los conceptos se relacionan unos con otros".

Por tanto, aunque en filosofía una ontología es una explicación sistemática de la Existencia, en los sistemas basados en el conocimiento, lo que existe es exactamente lo que se puede

representar, y lo que se representa se conoce, como el conjunto de objetos que están expuestos y sobre los cuales se puede hablar y razonar.

1.2. Clasificación de ontologías

Cuando hablamos de ontologías como sistemas de representación de conocimiento debemos especificar a qué tipo de sistemas nos referimos. En realidad, las ontologías se están empleando en todo tipo de aplicaciones informáticas en las que sea necesario definir concretamente el conjunto de entidades relevantes en el campo de aplicación determinado, así como las interacciones entre las mismas. Una ontología necesariamente incluirá un vocabulario de términos y una especificación de su significado (definiciones e interrelaciones entre conceptos) que impone estructura al dominio y restringe las posibles interpretaciones. (Uschold-Jasper, 1999).

Steve et al. (1998) distinguen tres tipos fundamentales de ontologías:

- **Ontologías de un dominio**, en las que se representa el conocimiento especializado pertinente de un dominio o subdominio, como la medicina, las aplicaciones militares, la cardiología, la oncología, etc..
- **Ontologías genéricas**, en las que se representan conceptos generales y fundacionales del conocimiento como las estructuras parte / todo, la cuantificación, los procesos o los tipos de objetos.
- **Ontologías representacionales**, en las que se especifican las conceptualizaciones que subyacen a los formalismos de representación del conocimiento, por lo que también se denominan meta-ontologías (meta-level o top-level ontologies).

A estos tres tipos, Guarino (1998) añade las ontologías que se crean para una actividad o tarea específica (denominadas task ontologies), como por ejemplo la venta de productos o el diagnóstico de una enfermedad y las ontologías creadas para una aplicación específica. En KACTUS [2] identificaron 4 tipos de ontologías de acuerdo a su alcance de aplicabilidad:

- **Ontología de la aplicación**: usadas por la aplicación. Ontología de procesos de producción, de diagnóstico de fallas, de diseño intermedio de barcos, etc.

- **Ontología del dominio:** específicas para un tipo de artefacto, generalizaciones sobre tareas específicas en algún dominio. Por ejemplo, ontología del proceso de producción de hidrocarburos, de la red eléctrica, de barcos, etc.
- **Ontologías técnicas básicas:** describe características generales de artefactos. Por ejemplo: componentes, procesos, funciones.
- **Ontologías genéricas:** describe la categoría de más alto nivel.

Otra forma de identificar ontologías es desde su punto de vista, por ejemplo, según el tipo de agente al que vayan destinadas:

- **Ontologías lingüísticas:** se vinculan a aspectos lingüísticos, esto es, a aspectos gramáticos, semánticos y sintácticos destinados a su utilización por los seres humanos.
- **Ontologías no lingüísticas:** destinadas a ser utilizadas por robots y agentes inteligentes.
- **Ontologías mixtas:** combinan las características de las anteriores.
- Según el grado o nivel de abstracción y razonamiento lógico que permitan:
- **Ontologías descriptivas:** incluyen descripciones, taxonomías de conceptos, relaciones entre los conceptos y propiedades, pero no permiten inferencias lógicas.
- **Ontologías lógicas:** permiten inferencias lógicas mediante la utilización de una serie de componentes como la inclusión de axiomas, etc.

Las personas mediante las ontologías se representan en su cabeza el mundo que los rodea. Estas ontologías no son explícitas, ya que no se detallan en un escrito ni se establecen de forma jerárquica o matemática. Todos usamos ontologías en las que Automóvil representa un medio de transporte y tiene cuatro ruedas. ¿Se formaliza este tipo de ontologías? No, sería innecesario: los automóviles son tan usuales que todos compartimos la información de lo que son. Lo mismo ocurre al pensar en el dominio familiar: se sabe que una familia dispone de varios miembros, que un hijo no puede tener más de un padre y una madre biológicos, que los padres tienen o han tenido padres. No necesitamos ser explícitos con este conocimiento, pues todo el mundo lo sabe. Sin embargo, cuando se tratan términos poco comunes o cuando se quiere que estos términos sean procesados por máquinas, se precisa explicitar las ontologías; esto es, desarrollarlas de forma que las máquinas comprendan.

Las máquinas carecen de las ontologías con las que nosotros contamos para entender el mundo y comunicarse entre ellas; por eso necesitan ontologías explícitas. En cuanto dos sistemas de información (sistemas ERP, bases de datos, bases de conocimiento) intentan comunicarse, aparecen problemas semánticos que dificultan o imposibilitan la comunicación entre ellos. Los problemas semánticos son de dos tipos: **de dominio** y **de nombre**. Los conflictos de dominio aparecen cuando conceptos similares en cuanto a significado, pero no idénticos, se representan en distintos dominios. Un ejemplo de esto, es el concepto representado por *Trabajador* en una base de datos (BD) que puede representar a un trabajador cualificado, mientras que en otra BD se puede utilizar *Trabajador* para cualquier trabajador, sea o no cualificado. Ambos conceptos están muy relacionados, pero no son equivalentes ni se deben mezclar. Usando ontologías, se podría especificar que el primer concepto sería una especialización del segundo; y un sistema de razonamiento automático basado en ontologías impediría, por ejemplo, que se contratara para tareas cualificadas a trabajadores no cualificados.

Los conflictos de nombre son de dos tipos: sinónimos y homónimos. Los sinónimos ocurren cuando los sistemas usan distintos nombres para referirse al mismo concepto. Por ejemplo, una BD puede usar *Trabajador* para el mismo concepto que otra usa *Empleado*. En ese caso, se podría usar una ontología que definiera como idénticos los dos términos. Así, las aplicaciones que manejaran esas bases de datos sabrían como llevar datos de una a otra.

Los homónimos surgen cuando los sistemas usan el mismo nombre para representar cosas distintas. Por ejemplo, en una aplicación de una compañía de seguros, *Conductor* representa a una persona que tiene contratada una póliza particular con la compañía; mientras que, en una aplicación de una compañía de taxis, *Conductor* representa a un trabajador que conduce un taxi de la compañía. Como es de suponer, si se intentara integrar automáticamente ambas aplicaciones basándose en que ambas usan el mismo término (*Conductor*) para significar lo mismo, se produciría el desastre más absoluto: al dar de baja a un conductor de taxi se le quitaría su póliza de seguros, con lo que no podría conducir ni su propio coche (al menos, no legalmente); y, al dar de alta a un asegurado, se le daría de alta como taxista, aunque no

tuviera la licencia de taxista. Sólo una ontología explícita le puede comunicar a una aplicación que su *Conductor* no guarda ninguna relación con el de otra.

Las ontologías explícitas se pueden expresar de muchas maneras. Como mínimo, deben incluir un vocabulario de términos, con la definición de cada uno. Por ejemplo, la ontología empresarial *Enterprise Ontology* (EO) define así Venta: “Una Venta es un acuerdo entre dos Entidades Legales para el intercambio de un Producto por un Precio de Venta. Normalmente, el Producto es un bien o servicio y el Precio de Venta es monetario, aunque se incluyen otras posibilidades” [1]. Las ontologías sencillas suelen representarse como una jerarquía de conceptos relacionados y ordenados.

Dependiendo del grado de formalidad, las ontologías explícitas se clasifican en informales, semi-informales, semi-formales y formales. Las primeras se expresan directamente en cualquier lenguaje natural. Las segundas se expresan en una forma estructurada y restringida de algún lenguaje natural. Las terceras se expresan en lenguajes estructurados, como RDF. Por último, las ontologías formales definen los términos mediante lenguajes lógico-matemáticos cuyos símbolos se definen exactamente y sin ambigüedades; en consecuencia, estas ontologías permiten emplear teoremas y demostraciones. Los dos últimos tipos de ontologías permiten que las aplicaciones puedan usar las definiciones de los conceptos del dominio y sus relaciones. Así como los tres primeros tipos de ontologías pueden contener términos ambiguos o inconsistentes, las ontologías formales no los permiten.

1.2.1. Clasificación por grado de axiomatización.

Según Sowa (2000) las ontologías pueden clasificarse en terminológicas o formales de acuerdo al grado de axiomatización de sus categorías definidas.

Terminológicas: Una ontología terminológica define términos y sus relaciones en taxonomías que involucran tanto relaciones de subtipo y supertipo, como las que relacionan partes con un todo, pero no incluyen axiomas y definiciones expresadas en lógica o algún tipo de lenguaje formal procesable por una computadora. Ejemplos:

- EDR (The Electronic Dictionary Research project), es un proyecto japonés que ha desarrollado un diccionario con más de 440 000 conceptos, con su traducción a palabras en inglés y japonés con muy poco detalle acerca de cada uno de sus términos.
- Word Net es una jerarquía de 166 000 palabras, muy usada para procesamiento de lenguaje natural por su temprana disponibilidad en Internet.

Formales: Una ontología formal tiene sus categorías restringidas por axiomas y definiciones expresadas en lógica formal o en algún tipo de lenguaje procesable por la computadora.

La diferencia entre una ontología terminológica y una formal, es en su grado de especificación.

1.2.2. Clasificación por dependencia del contexto.

Ontologías de dominio, específicas a un dominio con restricciones en contenido y relaciones, por ejemplo: EngMath Ontology, es una ontología para el ámbito de la ingeniería en Matemáticas.

Ontologías TOVE (Toronto Virtual Enterprise), que modela los procesos empresariales, por ejemplo: Enterprise Design Ontology, Project Ontology, Scheduling Ontology, Service Ontology.

Generales, vocabulario relacionado a cosas, tiempo, etc., por ejemplo: CYC, que provee gran cantidad de conocimiento humano general.

Meta Ontologías, similares a las de dominio con conceptos genéricos a través de diferentes áreas de conocimientos. Describen conceptos como estado, evento, acción, etc., por ejemplo: The Meteorology Ontology.

Ontologías de tareas, proveen un vocabulario sistemático de los términos usados para resolver problemas asociados con tareas particulares. Tiene conceptos como observación, hipótesis, objetivo, etc.

1.3. Algunos aspectos importantes de las ontologías.

En este epígrafe se hará referencia a algunos aspectos importantes de las ontologías, como son su composición, cuestiones a tener en cuenta para su diseño, así como conceptos claves en relación a las ontologías.

1.3.1. Composición de una ontología.

Según Gruber (1993 a), las ontologías se componen de:

- **Conceptos:** son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- **Relaciones:** representan la interacción y enlace entre los conceptos de un dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- **Funciones:** son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden aparecer funciones como: asignar-fecha, categorizar-clase, etc.
- **Instancias:** se utilizan para representar objetos determinados de un concepto.
- **Reglas de restricción o axiomas:** son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: "Si A y B son de la clase C, entonces A no es subclase de B", "Para todo A que cumpla la condición B1, A es C", etc. Los axiomas, junto con la herencia de conceptos, permiten inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

1.3.2. Cuestiones a tener en cuenta para diseñar ontologías.

A la hora de diseñar una ontología debemos tener en cuenta 5 cuestiones clave:

- **Claridad:** una ontología debe poder comunicar de manera efectiva el significado de sus términos. Las definiciones serán lo más objetivas posibles y deben explicarse también en lenguaje natural.
- **Coherencia:** una ontología debe permitir hacer inferencias que sean consistentes con las definiciones.
- **Extensibilidad:** deben anticiparse nuevos usos para así poder permitir extensiones y especializaciones.
- **Especificidad:** se debe especificar a nivel de conocimiento, sin que dependa de una codificación particular a nivel de símbolo.
- **Precisión:** debe hacerse la menor cantidad de suposiciones acerca del mundo modelado.

1.3.3 Conceptos claves en relación a las ontologías

Conceptos clave en relación a las ontologías son:

Clase: Es un objeto que define una categoría. Describe conceptos en el dominio del discurso.

Subclase: Es en sí misma una clase, pero que es hija de alguna otra clase.

Clase jerárquica: La compuesta por una colección de clases conectadas por relaciones "es un tipo de" (*class hierarchy*).

Casos (instances): Ejemplos específicos pertenecientes a alguna clase, objetos de una clase.

Roles o Propiedades (slots): Propiedades de cada concepto que describen varias características y atributos del concepto. Ayudan a definir las características de las clases.

Facetas: Se utilizan para definir qué tipo de valor puede contener un slot particular, valores permitidos, número de valores, etc. También se denominan restricciones de roles.

Valor: Describe una propiedad que se aplica a alguna clase o *instancia*.

Tipo: Define el tipo de valor (como cadena de caracteres, número, booleano, etc.)

Cardinalidad: Define cuántos valores puede tener un slot individual (máximo y mínimo).

Herencia (inherence): Es el proceso por el cual las subclases e *instancias* de alguna clase heredan propiedades y valores definidos más arriba en la jerarquía.

Variable: Espacio vacío que puede llenarse preguntando a clases e instancias. Cada variable comienza con un signo de interrogación.

Relación: Nuevo conocimiento que se obtiene por deducción, partiendo del conocimiento que se encuentra en la ontología. Las relaciones utilizan variables.

1.3.4. Otros conceptos.

Otros conceptos destacables a la hora de hablar de ontologías son:

Primitiva: categoría de una ontología que no puede ser definida en términos de otras categorías en la misma ontología. Un ejemplo de una primitiva es el concepto del tipo Punto en la geometría de Euclides. El significado de una primitiva no está determinado por una definición con una forma cerrada (*closed-form*), sino por axiomas que especifican cómo se relaciona a otras primitivas. Una categoría que es una primitiva en una ontología debe no ser primitiva en un refinamiento (*refinement*) de aquella ontología.

Base de conocimiento: término informal para referirse a una colección de información que incluye una ontología como un componente. Además de una ontología, una base de conocimiento debe contener información especificada en un lenguaje declarativo tal como reglas lógicas o sistemas expertos, aunque también incluye información no estructurada o formalizada expresada en lenguaje natural o en lenguaje de procesado.

Refinamiento (refinement): Un refinamiento (matización) de cada categoría de una ontología A, a alguna categoría de otra ontología B, lo cual se denomina un *refinamiento* de A. Cada categoría en A debe corresponder a una categoría equivalente en B, pero algunas de A deben ser equivalentes a no primitivas en B. El refinamiento define un orden parcial de ontologías: si B es un refinamiento de A, y C es un refinamiento de B, entonces C es un refinamiento de A; si dos ontologías son *refinamientos* una de la otra, entonces deben ser isomórficas.

1.4. Aplicaciones, características y lenguajes de las ontologías.

A continuación se exponen aplicaciones de las ontologías, algunas de sus características y algunos de los principales lenguajes de las ontologías.

1.4.1. Aplicaciones de las ontologías.

Las ontologías se usan para favorecer la comunicación entre personas, organizaciones y aplicaciones, lograr la interoperabilidad entre sistemas informáticos, razonar automáticamente y para la ingeniería de software.

Las ontologías favorecen la comunicación entre personas, organizaciones y aplicaciones porque proporcionan una comprensión común de un dominio, de modo que se eliminan confusiones conceptuales y terminológicas. Los problemas derivados de la falta de comprensión común entre personas revisten una gran importancia en la ciencia y en la tecnología.

En los campos de la Inteligencia Artificial, la Teoría de Decisiones y la Teoría de Sistemas Distribuidos (campos muy relacionados con la Web semántica), sucede algo parecido: los investigadores de un campo no pueden leer fácilmente los resultados de los investigadores de los otros, pues se usan diferentes perspectivas y términos para las mismas ideas y

conceptos. Construyendo una ontología común para los tres campos, las investigaciones de un campo serían inmediatamente aplicables a los otros.

1.4.2. Beneficios de las ontologías.

Los beneficios de utilizar ontologías se pueden resumir de la siguiente forma:

- Proporcionan una forma de representar y compartir el conocimiento utilizando un vocabulario común.
- Permiten usar un formato de intercambio de conocimiento.
- Proporcionan un protocolo específico de comunicación.
- Permiten una reutilización del conocimiento.
- Repositorios para la organización del conocimiento.
- Servir de herramienta para la adquisición de información.
- Servir de herramientas de referencia en la construcción de sistemas de bases de conocimiento que aporten consistencia, fiabilidad y falta de ambigüedad a la hora de recuperar información.
- Normalizar los atributos de los metadatos aplicables a los documentos.
- Crear una red de relaciones que aporte especificación y fiabilidad.
- Permitir la integración de diferentes perspectivas de usuarios.
- Permitir el tratamiento ponderado del conocimiento para recuperar información de forma automatizada.
- Permitir la construcción automatizada de mapas conceptuales y mapas temáticos.
- Permitir la reutilización del conocimiento existente en nuevos sistemas.
- Permitir la interoperatividad entre sistemas distintos.
- Establecer modelos normativos que permitan la creación de la semántica de un sistema y un modelo para poder extenderlo y transformarlo entre diferentes contextos.
- Servir de base para la construcción de lenguajes de representación del conocimiento.

1.4.3. Características de las ontologías.

Algunas de las características de las ontologías son:

- **Pueden existir ontologías múltiples:** si el propósito de una ontología es hacer explícito algún punto de vista, en algunos casos, necesitamos combinar dos o más ontologías. Cada ontología introduce conceptualizaciones específicas.
- **Se pueden identificar distintos niveles de abstracción estableciendo una topología de ontologías:** se puede caracterizar una red de ontologías usando multiplicidad y abstracción. Al no poder realizar una descripción completa del mundo, se puede pensar una estrategia de construcción gradual que vaya de abajo hacia arriba.
- **Multiplicidad de la representación:** un concepto puede ser representado de muchas formas, por lo que pueden coexistir múltiples representaciones del mismo concepto.
- **Mapeo de ontologías:** se pueden establecer las relaciones entre los elementos de una o más ontologías para establecer generalizaciones, especializaciones, conexiones, etc.

1.4.4. Algunos lenguajes de ontologías.

Entre los principales lenguajes de ontologías podemos destacar los siguientes:

SHOE: *Simple HTML Ontology Extensions*. Fue el primer lenguaje de etiquetado para diseñar ontologías en la Web. Este lenguaje nació antes de que se ideara la Web Semántica. Las ontologías y las etiquetas se incrustaban en archivos HTML. Este lenguaje permite definir clases y reglas de inferencia, pero no negaciones o disyunciones. A su albur se desarrollaron muchos editores, buscadores, APIS, etc.; pero este proyecto fue abandonado a medida que se desarrollaron OIL y DAM; aunque también existe una serialización de este lenguaje en XML.

OIL: *Ontology Inference Layer*. Este lenguaje, derivado en parte de SHOE, fue impulsado también por el proyecto de la Unión Europea *On-To-Knowledge*. Utiliza ya la sintaxis del lenguaje XML y está definido como una extensión de RDFS. Se basa tanto en la lógica descriptiva (declaración de axiomas) y en los sistemas basados en *frames* (taxonomías de clases y atributos). OIL posee varias capas de sub-lenguajes, entre ellas destaca la capa base que es RDFS, a la que cada una de las capas subsiguientes añade alguna funcionalidad y mayor complejidad. La principal carencia de este lenguaje es la falta de expresividad para declarar axiomas.

DAML y OIL: Este lenguaje nació fruto de la cooperación entre OIL y DARPA y unifica los lenguajes DAML (*DARPA's Agent Markup Language*) y OIL (*Ontology Inference Layer*). Se basa ya en estándares del W3C. El lenguaje DAML se desarrolló como una extensión del lenguaje XML y de Resource Description Framework (RDF) y para extender el nivel de expresividad de RDFS. DAML- OIL hereda muchas de las características de OIL, pero se aleja del modelo basado en clases (frames) y potencia la lógica descriptiva. Es más potente que RDFS para expresar ontologías. En la última revisión del lenguaje (DAML+OIL) ofrece ya un rico conjunto de elementos con los cuales se pueden crear ontologías y marcar la información para que sea legible y comprensible por máquina. También funciona como formato de intercambio. Sin embargo, este lenguaje presenta algunas carencias debido a su complejidad conceptual y de uso, complejidad que se intentó solventar con el desarrollo de OWL. No obstante, se desarrollaron muchas aplicaciones que utilizan DAML-OIL y también existen herramientas para convertir DAML a OWL.

OWL: *OWL Web Ontology* o Lenguaje de Ontologías para la Web es un lenguaje de etiquetado semántico para publicar y compartir ontologías en la Web. Se trata de una recomendación del W3C, y puede usarse para representar ontologías de forma explícita, es decir, permite definir el significado de términos en vocabularios y las relaciones entre aquellos términos (ontologías). En realidad, OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque es un lenguaje con más poder expresivo que éste. Se trata de un lenguaje diseñado para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones, en oposición a situaciones donde el contenido solamente necesita ser presentado a los seres humanos. OWL surge como una revisión al lenguaje DAML-OIL y es mucho más potente que éste. Al igual que OIL, OWL se estructura en capas que difieren en la complejidad y puede ser adaptado a las necesidades de cada usuario, al nivel de expresividad que se precise y a los distintos tipos de aplicaciones existentes (motores de búsqueda, agentes, etc.).

KIF: *Knowledge Interchange Format* es un lenguaje para representar ontologías basadas en la lógica de primer orden. KIF está basado en la lógica de predicados con extensiones para definir términos, metaconocimiento, conjuntos, razonamientos no monotónicos, etc.; y pretende ser un lenguaje capaz de representar la mayoría de los conceptos y distinciones

actuales de los lenguajes más recientes de representación del conocimiento. Se trata de un lenguaje diseñado para intercambiar conocimiento entre sistemas de computación distintos, diferentes lenguas, etc.; y no para la interacción entre seres humanos.

FOAF: aunque no es exactamente un lenguaje de ontologías ya que se trata de un vocabulario con definiciones que usa el lenguaje RDFS/OWL, FOAF hace más fácil que el software procese los términos del vocabulario FOAF para describir documentos. FOAF permite crear una base de datos unificada de información al normalizar una forma de definir categorías, tipos de relaciones, etc.

1.5. Comparación entre Tesoros, Mapas Conceptuales y Ontologías.

En este epígrafe, se exponen las definiciones de tesoro y de mapas conceptuales, así como la comparación entre ellos y las ontologías.

1.5.1. Tesoros y ontologías.

A continuación se presenta la definición de tesoro, las similitudes entre las ontologías y los tesoros y sus diferencias.

1.5.1.1. Definiciones de tesoro.

Tesoro proviene de la palabra inglesa thesaurus que Peter Mark definió en 1852 como “Colección de palabras y frases ordenadas, no en orden alfabético, como están en un diccionario, sino de acuerdo con las ideas que representan. Es decir, que se tiene la idea, y hay que buscar la palabra o palabras que se ajusten exactamente a esa idea” [3]. Esta definición servirá de base para las sucesivas acepciones y uso del término thesaurus. Según la definición de la UNESCO, "un tesoro es un instrumento de control terminológico utilizado para traducir a un lenguaje más estricto el idioma natural empleado en los documentos y por los indizadores"[4].

Por su estructura, es un vocabulario controlado y dinámico de términos relacionados semántica y genéricamente, los cuales cubren un dominio específico del conocimiento. La estructura de la terminología de un tesoro está basada en las interrelaciones entre los conceptos. Estas interrelaciones pueden ser: jerárquicas, de afinidad y preferenciales. Las relaciones jerárquicas indican términos más amplios o más específicos de cada concepto. Las relaciones de afinidad muestran términos relacionados conceptualmente, pero que no

están ni jerárquica ni preferencialmente relacionados. Las relaciones preferenciales se utilizan para indicar cuál es el término preferido o descriptor entre un grupo de sinónimos; y la calificación de homónimos para diferenciar su significado, eligiendo un significado preferido para cada término.

1.5.1.2. Similitudes entre ontologías y tesauros.

Las similitudes que podemos destacar entre las ontologías y los tesauros son las siguientes:

- Ambos proporcionan una representación del conocimiento compartido de un dominio con el fin de facilitar la comunicación eficiente.
- Ambos son sistemas basados en conceptos que representan conocimiento altamente complejo.
- Ambos pueden englobarse dentro de los lenguajes controlados. Puesto que ambos definen el vocabulario de un dominio específico mediante un conjunto de términos básicos y las relaciones entre dichos términos.
- Ambos están relacionados con la terminología empleada para representar los conceptos de un dominio particular.
- Ambos utilizan jerarquías para agrupar términos en categorías y subcategorías.
- Ambos pueden utilizarse para catalogar y organizar recursos de información.
- Si definimos la ontología como un instrumento que define los términos básicos y sus relaciones a partir del vocabulario de un área, así como, las reglas de combinación de estos términos y relaciones, podemos observar que el paralelismo con los tesauros es evidente. Especialmente en lo que se refiere a la delimitación de los términos que deben formar parte de una construcción de representación del conocimiento, así como en lo que se refiere a las relaciones que se establecen.

1.5.1.3. Diferencias.

Son varios los autores que han profundizado en las diferencias existentes entre las ontologías y los tesauros. Por ejemplo, Qin y Paling (2000-01) entienden que las primeras son superiores a los segundos por varias razones:

- Presentan un nivel más alto de concepción y descripción del vocabulario. Además, las ontologías se caracterizan por una descripción del vocabulario.

- Se caracterizan por un desarrollo semántico más profundo para las relaciones del tipo clase/subclase y para las relaciones cruzadas, lo que supone la ampliación de éstas y un mayor cuidado en su descripción, por supuesto explícita.
- Destacan el uso de la lógica de descripción empleada en la descripción de situaciones.
- Son reusables.
- Posibilidad de trabajo en sistemas heterogéneos, al describir formalmente objetos en el mundo, sus propiedades, y las relaciones entre estos objetos.

En esta misma línea de pensamiento, Ding y Foo (2002) creen que las diferencias fundamentales entre una ontología y un vocabulario de representación convencional se sitúan en el nivel de abstracción, en las relaciones entre conceptos, en la capacidad para que sea comprensible para las máquinas y, lo más importante, en la expresividad que pueden proporcionar. Así:

- Una ontología puede estar elaborada de acuerdo con diferentes requerimientos y, al mismo tiempo, puede funcionar como un esquema de base de datos, como una auténtica base de conocimiento, para definir varias tareas o aplicaciones.
- Una ontología potencia la comunicación entre humanos y ordenadores mientras que un vocabulario convencional, en el mundo de lo que ellos llaman Library Science, sólo permite la comunicación entre seres humanos.
- Una ontología promueve la normalización y reutilización de la representación de la información mediante la identificación del conocimiento común y compartido.
- Las ontologías añaden valor a los tesauros tradicionales a través de una semántica más profunda, así como desde un prisma conceptual, relacional e informático. De hecho una mayor profundidad semántica puede implicar niveles más profundos de jerarquía, unas enriquecidas relaciones entre clases y conceptos, así como la capacidad de formular reglas de inferencia, etc.

1.5.2. Mapas conceptuales.

A continuación se expone la definición de mapas conceptuales y se realiza la comparación entre mapas conceptuales, tesauros y ontologías.

1.5.2.1. Definiciones de mapas conceptuales.

Los mapas conceptuales son un instrumento proveniente del campo de la didáctica, desarrollado por Joseph D. Novak y su equipo en la Universidad de Cornell (Estados Unidos) como resultado de sus investigaciones sobre el proceso de aprendizaje infantil de las ciencias en los años setenta. Joseph D. Novak los define como “herramientas para organizar y representar conocimiento” [5].

Para su desarrollo, se basaron en las teorías sobre la psicología del aprendizaje del psicólogo cognitivo David Ausubel, que fueron enunciadas en los 60, con él, Novak colaboró posteriormente. Ausubel desarrolló la teoría de la asimilación, centrada en el concepto de aprendizaje significativo, el mismo establece que el aprendizaje se origina a partir de la asimilación de nuevos conocimientos, conceptos y proposiciones, que cobran significado al juntarse a la estructura cognitiva previa del individuo. Esta estructura cognitiva está constituida, según Ausubel, por conceptos y proposiciones organizados de forma jerárquica, representando su conocimiento y experiencias.

El objetivo de los mapas conceptuales es representar relaciones entre conceptos en forma de reproducciones con el fin de hacerlas explícitas. Joseph D. Novak define concepto como “una regularidad percibida en sucesos u objetos o registros de sucesos u objetos, designado por una etiqueta” [6], etiqueta que suele ser una palabra, y proposición como “una frase acerca de cierto objeto o suceso en el universo, que ocurre de forma natural o artificial. Las proposiciones contienen dos o más conceptos conectados con otras palabras que forman una frase coherente” [6], indicando que también son denominadas “unidades semánticas”.

1.5.3. Comparación entre mapas conceptuales, tesauros y ontologías.

Al comparar estos instrumentos, debemos destacar sus semejanzas y potencialidades, que caen fundamentalmente en la contribución de elementos de asociatividad entre conceptos, o lo que es igual, la búsqueda de herramientas documentales que permitan la organización de los recursos educativos en el medio digital adecuados a las características del proceso de aprendizaje significativo. Dos características son comunes a todos: los conceptos y las relaciones entre ellos. Con diferentes designaciones terminológicas y reglas diferentes para su uso, un ejercicio de abstracción los iguala en el plano teórico y conceptual. Esta

semejanza los ha llevado a un plano de interrelación al fundirse desde campos diferentes en un espacio común, el espacio digital, donde se organiza el hipertexto, cuya organización está basada en nodos y enlaces que se superponen a la misma representación.

Los tesauros recogen la experiencia en el campo documental, y los mapas conceptuales sintetizan el de la educación, se incorporan a la web como instrumentos de representación del conocimiento, como ontologías de bajo nivel que establecerían los primeros pasos en la web semántica. Además permiten dar soporte al multilingüismo.

Los conceptos son representados en todos los instrumentos mediante palabras extraídas del lenguaje natural. En mapas conceptuales y ontologías, a diferencia de los tesauros, no se realiza control sobre el vocabulario. Sin embargo, en las ontologías se intenta elegir términos representativos de los conceptos, estableciendo relaciones y axiomas con el fin de acotar su significado de manera más estricta que en los tesauros. Todos ellos, además, incluyen relaciones o enlaces entre conceptos, con distintos sistemas de utilización, grados de sistematización y denominaciones.

Las relaciones de equivalencia de los tesauros proveen un mecanismo para acercar todas las etiquetas posibles de un concepto al lenguaje del usuario, tal como suele estar representado en los mapas conceptuales. Este tipo de relación está ausente como tal en mapas conceptuales y ontologías, aunque se utiliza un mecanismo semejante cuando se integran en sistemas automatizados mediante su combinación con otras herramientas de tipo lingüístico.

La estructura jerárquica impuesta al esquema de representación del conocimiento del dominio es más laxa en los mapas conceptuales, donde su necesidad está puesta en entredicho por algunos autores y no está sujeta a reglas de construcción. Tanto en tesauros como en ontologías supone una parte fundamental, establecida a partir de relaciones jerárquicas que son definidas en forma similar y adolecen de los mismos problemas de imprecisión en ambos casos, permitiendo la inferencia a partir de mecanismos de herencia.

Las relaciones asociativas, por último, suponen el mayor reto en cualquiera de ellos. Los tesauros, aún teniendo definidas algunos tipos, dejan abierta la puerta a la inclusión de

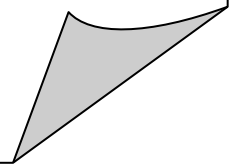
otras. Los mapas conceptuales admiten cualquier tipo de relación asociativa que el autor quiera expresar y en las ontologías, donde se especifican como propiedades, dependen del sistema lógico elegido para definir las.

Una diferencia fundamental de los tesauros en relación con los otros dos instrumentos es la ausencia de especificación del tipo de relación establecida. No poseen más descripción que uno de los tres grandes tipos de relación, lo que limita su expresividad semántica. Como ya vimos, paliar estas dificultades está siendo uno de los intereses de la investigación más reciente, con la intención de convertirlos en herramientas más potentes en el ámbito automatizado. Además, tanto mapas conceptuales como ontologías utilizan principalmente verbos para especificar el tipo de relación establecida siendo equiparables proposiciones a propiedades.

1.6. Conclusiones del capítulo.

En este capítulo abordamos conceptos importantes de las ontologías. Estos conceptos permitirán comprender por qué usamos las ontologías para formular un modelo para representar el dominio de los conceptos de una disciplina asociados con los dominios docentes. La exposición de dicho modelo y la herramienta Protégé que es la que utilizaremos, será el objetivo del siguiente capítulo.

Capítulo II:
HERRAMIENTA PROTÉGÉ Y EL
MODELO DE UNA ASIGNATURA
TEÓRICO-PRÁCTICA.



Capítulo II. HERRAMIENTA PROTÉGÉ Y EL MODELO DE UNA ASIGNATURA TEÓRICO-PRÁCTICA.

En el presente capítulo hablaremos de cómo ha influido la tecnología en la Enseñanza, y las facilidades que ofrecen las ontologías para formular un modelo para representar el dominio de los conceptos de una asignatura teórico-práctica. También abordaremos conceptos importantes de la herramienta Protégé y como se modela con esta herramienta la ontología para una asignatura teórico-práctica. Con este fin este capítulo se ha estructurado de siguiente manera:

- En el epígrafe 2.1 se habla de la aplicación de la tecnología en la Enseñanza.
- En el epígrafe 2.2 se refiere una conceptualización para el modelo de una asignatura teórico-práctica.
- En el epígrafe 2.3 se describe el modelo, con cada uno de sus dominios.
- En el epígrafe 2.4 se abordan algunas características de la herramienta Protégé.
- En el epígrafe 2.5 se realiza el modelado de la ontología con la herramienta Protégé.

2.1. Aplicación de la tecnología en la Enseñanza.

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 aprendizaje; la educación es más equitativa y genera una formación continua.

De esta manera, la tecnología ha influenciado al menos en dos aspectos el mundo educacional. Uno relacionado con los intereses pedagógicos, administrativos y de gestión escolar y la segunda con los cambios en las habilidades y competencias requeridas, para lograr una inserción de las personas en la sociedad actual.

Lo anterior, presenta un escenario de crecientes necesidades de apoyo a la formación de los estudiantes, donde, por una parte, con menores o mayores avances, hay tecnología ya instalada (computadoras, Internet, software de propósito general y algunas aplicaciones educativas) y por otra los procesos de alfabetizado de los profesores.

El objetivo principal de una buena representación del conocimiento es lograr una descripción del contenido de un dominio. La representación del conocimiento basada en elementos instructivos, constituye un nivel de abstracción que permite gestionar el conocimiento de forma independiente de la estructura del material educativo.

2.1.1. Sistemas de enseñanza.

Sistemas Tutores Inteligentes (ITS): Un Sistema Tutor Inteligente es un sistema de enseñanza asistida por computadora, que utiliza técnicas de Inteligencia Artificial, principalmente para representar el conocimiento y dirigir una estrategia de enseñanza; y es capaz de comportarse como un experto, tanto en el dominio de conocimiento que enseña (mostrando al alumno cómo aplicar dicho conocimiento), como en el dominio pedagógico, donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje.

Sistemas hipertexto adaptativos: Un Hipertexto es un documento hipermedia, como un sitio Web educativo o un CD-ROM que contienen un cuerpo coherente de materia organizada y desarrollado con propósitos educativos. Un elemento importante de los hipertextos adaptativos, es la posibilidad de individualizar el acceso a la información de manera que esta se acomode y responda a la diversidad y necesidades de los usuarios posibles, junto con ser capaz de modelar al alumno y ser capaz de actualizar dicha información.

2.1.2. Las facilidades que ofrecen las ontologías.

Para este propósito las ontologías tienen facilidades, como centrar la modelación en:

- el dominio de la materia de estudio.
- aspectos educativos.

Permiten construir escenarios en los que se proporciona una enseñanza conceptual, más orientada hacia la comprensión y al razonamiento que hacia la destreza en el desarrollo de determinados problemas (Andriessen and Sandberg, 1999). Lo que se modela se ajusta al conocimiento declarativo o del dominio estático según se expone en (Murray, 1998) compuesto por el conocimiento conceptual y el conocimiento pedagógico, en contraposición al conocimiento dinámico o procedimental de las tareas instructivas en los ITS.

El diseño cognitivo tiene la finalidad de proporcionar una descripción formal del dominio de conocimiento de una materia de forma separada e independiente de la estructura de cualquier entorno, para esto se debe crear un modelo genérico e instanciar el modelo.

El diseño estructural tiene como finalidad la de proporcionar una representación de un entorno educativo de forma independiente de la tecnología.

2.2. Ontología para el modelo de una asignatura teórico-práctica.

A continuación nos dedicaremos a describir la conceptualización que se hace de una asignatura teórico-práctica. El modelo que describe Miguel Rodríguez (2000) consta de tres dominios que caracterizan diferentes aspectos del conocimiento de una materia genérica.

- Un **dominio conceptual** que recoge la elicitación de los conceptos que componen el conocimiento del dominio de la materia. Se recogen también las relaciones de tipo estructural y otras dependientes de la materia que se modela.
- Un **dominio instruccional** que describe los objetos instruccionales asociados a los elementos del dominio conceptual. En este caso, el modelo es muy genérico, con elementos comunes en otros modelos instruccionales (ejemplo, problema, solución, pista, etc.). Este dominio es genérico en el ámbito de la representación de los elementos instruccionales asociados a la enseñanza de una asignatura de tipo científico-técnico, donde intervenga la teoría aplicada, y con ese propósito se ha diseñado.
- Un **dominio didáctico** que clasifica los elementos y relaciones de los dominios anteriores de acuerdo con sus propiedades pedagógicas, incorporando atributos y relaciones. En este caso este dominio matiza cada uno de los elementos de los dominios restantes desde el punto de vista pedagógico o didáctico (si un ejemplo es fácil o difícil). Este tipo de atributos permiten luego seleccionar los elementos de conocimiento de acuerdo a estos criterios. También se incorporan relaciones de tipo pedagógico entre los elementos de los dominios, como la relación de prerrequisito, entre otras.

Los tres describen elementos propios o bien contienen relaciones o atributos entre elementos de los otros dos. En la figura 2.1 se esquematiza la estructura de los dominios, en los que las relaciones didácticas se definen sobre los otros dos dominios.

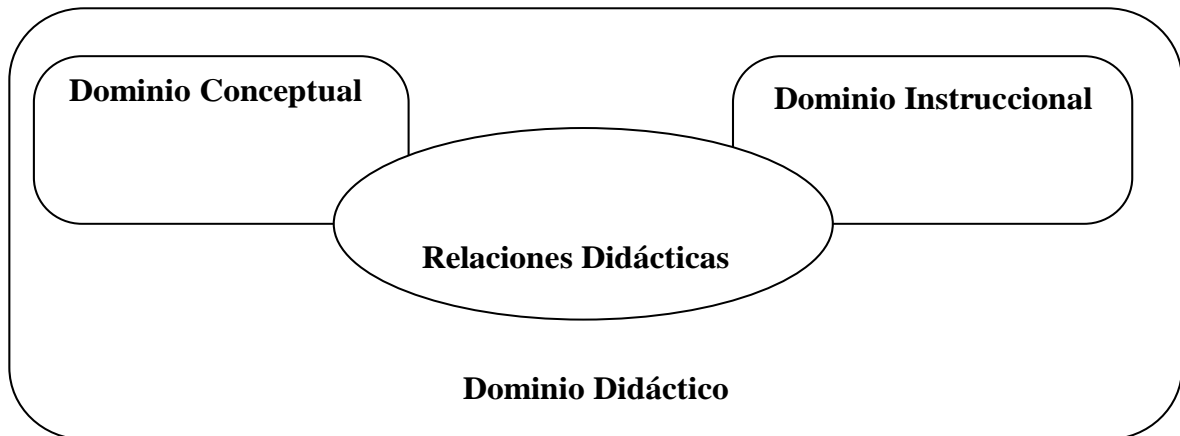


Figura 2.1 Los dominios que modelan una materia de estudio.

2.3. Descripción del modelo.

A continuación describimos cada dominio con sus entidades, atributos y relaciones, también expresadas por Miguel Rodríguez (2000).

2.3.1. El dominio conceptual.

El dominio conceptual va a representar los objetos, conceptos y otras entidades que existen en una determinada área de interés junto con las relaciones que se establecen entre ellos (Genesereth and Nilsson, 1987). El objetivo es el de tener un dominio explícito sobre el que establecer las relaciones didácticas e instruccionales entre ellos.

Se describen estos en base a entidades y relaciones, de acuerdo con el formalismo descrito.

2.3.1.1. Entidades del Dominio conceptual.

Las entidades modeladas son las siguientes:

- **Concepto:** Se puede definir como el elemento de conocimiento básico para el razonamiento en el ámbito de la materia. Por ejemplo predicado, cuantificador, variable libre.
- **Actividad:** La actividad es una acción que se realiza sobre un concepto. Es un elemento de carácter procedimental que se realiza sobre un concepto. Por ejemplo, evaluar un predicado.

2.3.1.2. Relaciones del dominio conceptual.

Las relaciones que se establecen entre los elementos del dominio son de dos tipos: estructurales y dependientes del dominio.

Capítulo II: Herramienta Protégé y el modelo de una asignatura teórico-práctica.

Las relaciones estructurales son aquellas que clasifican las entidades del dominio en base a dos jerarquías ortogonales.

- **tipo-de:** Es la jerarquía de clases, que se define con la intención de proporcionar respuesta a preguntas del tipo Cuáles son los tipos de C, siendo C un concepto del dominio.
- **parte-de:** Es la relación de composición, que relaciona las partes de un elemento.

Las relaciones dependientes del dominio de conocimiento son:

- **induce:** La existencia de elementos conceptuales puede tener como consecuencia la existencia de otro elemento conceptual que no tendría sentido sin los anteriores. Un ejemplo puede ser el de que un predicado y la noción de satisfacer, que inducen un conjunto de estados de las variables libres de ese predicado que lo satisfacen.
- **produce:** El resultado de realizar una actividad sobre uno o varios conceptos produce como resultado otro concepto.
- **se realiza sobre:** Establece un vínculo entre la actividad y el concepto sobre la que ésta se aplica.

Tabla 2.1 Tabla de elementos y relaciones del dominio conceptual

Entidades
concepto[NOMBRE, CONTENIDO]
actividad[NOMBRE, CONTENIDO]
Relaciones
tipo_de (concepto, concepto)
parte_de (concepto, concepto)
induce (concepto, concepto)
produce (actividad, concepto)
se_realiza_sobre (actividad, concepto)

La tabla 2.1 muestra de forma resumida los elementos que hemos descrito. Como hemos indicado antes, se muestran las entidades y las relaciones del modelo conceptual. Entre corchetes definimos los atributos más representativos tanto de las entidades como de las relaciones.

Como se ve, el modelo del dominio describe el conocimiento del mismo sin proporcionar ningún tipo de descripción didáctica o educativa del mismo, por lo que será necesaria la creación de otros dominios que puedan representar información docente. Este es el objetivo de los dominios instruccional y didáctico.

2.3.2. El dominio instruccional.

El dominio instruccional describe las entidades instruccionales, es decir, aquellas cuya finalidad es la de ayudar a la adquisición y comprensión sobre el conocimiento de la materia.

2.3.2.1. Entidades del dominio instruccional.

Las entidades que componen este dominio son las siguientes:

- **Ejemplo:** Una instancia de un concepto o de una actividad con el objeto de facilitar la comprensión de éste.
- **Error:** Un fallo que sea frecuente en la comprensión de alguno de los elementos del dominio.
- **Pista:** Es una entidad que orienta hacia la solución de un problema.
- **Explicación:** Complementa la información que se haya proporcionado sobre algún elemento del dominio, en general sobre una solución.
- **Problema:** El enunciado de un ejercicio que pone a prueba el conocimiento y grado de comprensión sobre uno o varios conceptos.
- **Solución:** Los pasos asociados a la resolución de un problema.
- **Pregunta/Respuesta:** Son entidades diferentes pero asociadas a un elemento expositivo con una cuestión acerca de éste y eventualmente con una respuesta.

Entre estos elementos podemos hacer también una clasificación basada en los aspectos descritos en otros trabajos como las expuestas en (Marcke, 1992). En ella se diferencian elementos de conocimiento que se usan con un fin expositivo y que son los que constan de conocimiento declarativo (pista, explicación, ejemplo), de otros que se denominan exploratorios y que son aquellos que utilizan para ejercitar una habilidad con el conocimiento adquirido (problema).

2.3.2.2. Relaciones del dominio instruccional.

Estos objetos entre sí están ligados mediante una serie de relaciones que los enlazan. Son las siguientes:

- **es solución:** Enlaza una entidad problema con la solución al mismo.
- **explica:** Una explicación aclara o explica otros elementos, como una solución o un error.
- **responde a:** Una respuesta responde a una pregunta.

La tabla 2.2 muestra un esquema de los elementos del dominio.

Tabla 2.2: Tabla de elementos y relaciones del dominio instruccional.

Entidades y Atributos
ejemplo[NOMBRE, CONTENIDO]
error[NOMBRE, CONTENIDO]
pista[NOMBRE, CONTENIDO]
explicación[NOMBRE, CONTENIDO]
problema[NOMBRE, CONTENIDO]
solución[NOMBRE, CONTENIDO]
pregunta[NOMBRE, CONTENIDO]
respuesta[NOMBRE, CONTENIDO]
Relaciones
es_solucion (solución ,problema)
explica(explicación/pista, problema/ejemplo/error/solución)
responde_a(pregunta, respuesta)

En la tabla 2.2 se resumen los elementos del dominio instruccional. En ella se observa que algunas de las relaciones se han definido sobre varios de los elementos del dominio.

2.3.3. El dominio didáctico.

El dominio didáctico complementa a los otros dominios con propiedades didácticas asociadas a los elementos y relaciones. El conocimiento didáctico es aquel que clasifica y describe los elementos de un dominio de conocimiento atendiendo a sus cualidades pedagógicas.

En este dominio se definen:

- Algunos atributos didácticos sobre las entidades ya creadas en los dominios anteriores.
- Relaciones entre entidades de otros dominios.

Las entidades en el dominio didáctico son las de los restantes dominios. Lo que se representa en este dominio son los aspectos didácticos de cada una de ellas. Por esta razón se añaden atributos, de manera que se matiza la entidad. Por ejemplo, la entidad problema del dominio instruccional puede estar clasificada en el dominio didáctico como un problema fácil o difícil.

2.3.3.1. Atributos didácticos.

Se definen sobre las entidades de los otros dominios los siguientes atributos:

- Aquellas entidades expositivas y explicativas tanto del dominio conceptual como del instruccional tienen asociado un grado de dificultad, que indica la dificultad para la comprensión del elemento. Los valores de este atributo pueden ser del tipo fácil, medio, difícil, etc.
- Además, las entidades del tipo exploratorio tienen asociado un nivel de adquisición que indica qué papel juega el problema en referencia a los conceptos que involucra. El valor de este atributo puede ser: comprender, aplicar, conocer, entre otros.

2.3.3.2. Relaciones didácticas.

Las relaciones didácticas establecen las dependencias desde el punto de vista del proceso educativo entre los diferentes elementos de los dominios conceptual e instruccional. Las que se han descrito son las siguientes:

- **prerrequisito:** En ella se asocian conceptos y actividades en una sucesión que permiten establecer los órdenes de dependencia de conocimientos.
- **involucra:** Un problema involucra para su desarrollo uno o varios conceptos.
- **ejercita:** Un problema ejercita uno o varios conceptos.
- **ilustra:** Un ejemplo ilustra uno o más conceptos.
- **tiene faq:** Un elemento del dominio conceptual o instruccional está asociado con un elemento de tipo pregunta mediante esta relación.

Tabla 2.3: Tabla de elementos y relaciones del dominio didáctico

Atributos
GRADO DE DIFICULTAD
NIVEL DE ADQUISICIÓN
Relaciones
involucra(problema/ejemplo, concepto)
ilustra(ejemplo, concepto)
ejercita(problema, concepto/actividad)
prerrequisito(concepto/actividad, concepto/actividad)
tiene_faq(concepto/actividad/problema/ejemplo, pregunta)

La tabla 2.3 muestra las relaciones didácticas que se establecen entre las entidades de los dominios conceptual e instruccional.

Este modelo genérico recoge la descripción de un dominio de conocimiento de una materia junto con elementos instruccionales y didácticos. Como hemos señalado, la finalidad con la que se crea es la de proporcionar un modelo de representación del conocimiento lo más apropiado posible para realizar la labor de diseño de material instruccional.

2.3.4. Modelo de información del metamodelo.

Los modelos de información que hemos descrito, pueden representarse de forma explícita utilizando el mismo formalismo entidad-relación. De esta forma, podemos describir explícitamente una categorización determinada. El modelo de información que vamos a describir consta de las entidades, atributos y relaciones siguientes:

Entidades: Las entidades del modelo de información son:

Dominios: Contiene los dominios en los que se divide el modelo.

Entidades: Las entidades del modelo clasificadas por dominios.

Relaciones: Las relaciones que existen entre las entidades. Se caracterizan por el nombre de la relación, y la cardinalidad de la misma.

Atributos: Los atributos de las entidades y de las relaciones que van a formar parte del modelo.

Un atributo puede ser compartido por una o más de las entidades que forman el modelo.

La figura 2.2 muestra el esquema del modelo de información expuesto en lo que respecta a las relaciones de pertenencia y composición.

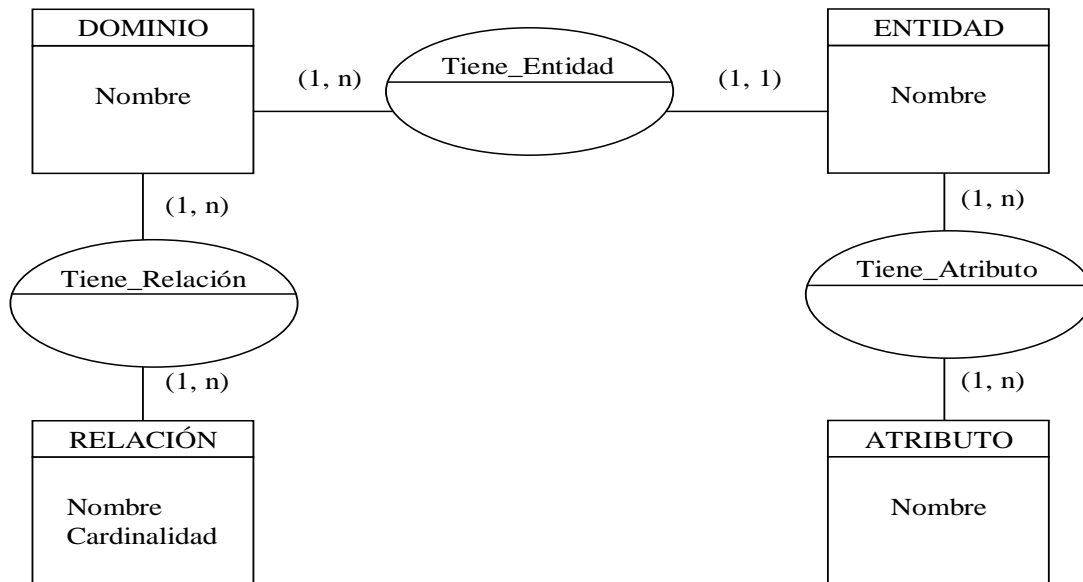


Figura 2.2 Modelo de la información del metamodelo.

Relaciones

Las relaciones del modelo describen la composición de las entidades y las relaciones que las ligan. En la figura 2.2 se muestran las relaciones estructurales de los modelos describiendo:

- Las entidades y relaciones que pertenecen a un dominio determinado
- Los atributos que tienen las entidades y el formato de representación que éste tendrá en la implementación del modelo físico de datos.

Vamos a ilustrar con un ejemplo cómo crear un modelo, es decir, una instancia del metamodelo anterior. Supongamos que tenemos que representar el modelo descrito en la figura 2.3. En ella se muestra un dominio educativo que contiene ejemplos y conceptos. Entre ellos se establecen dos relaciones: una relación de prerrequisito entre conceptos y otra relación llamada ilustra que asocia ejemplos con conceptos.

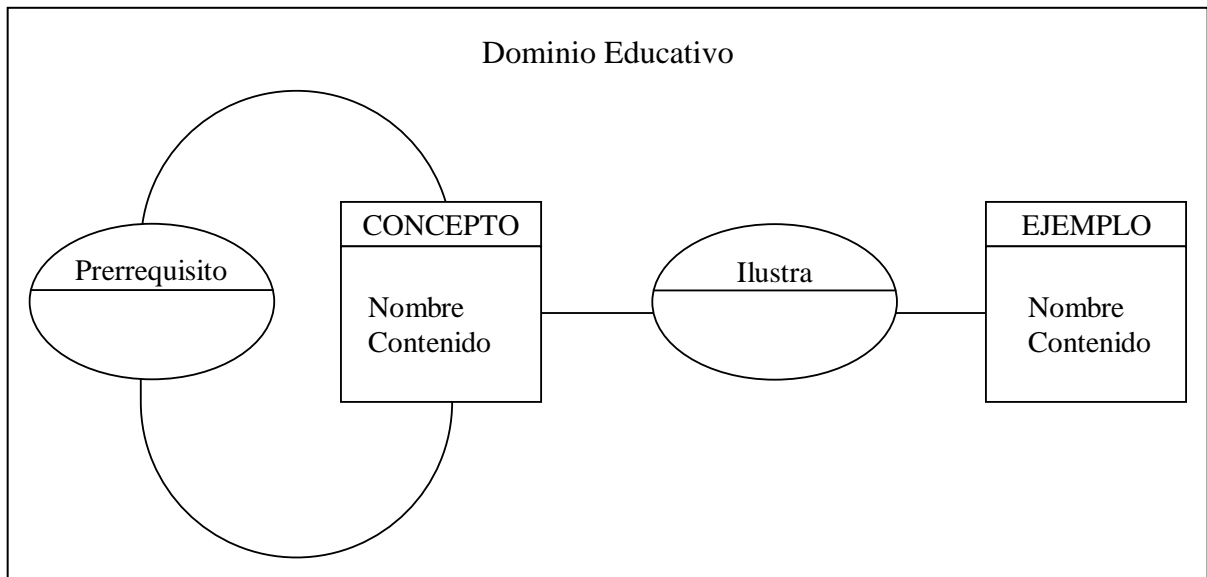


Figura 2.3 Modelo de un dominio.

Para representar el modelo descrito por la figura 2.3, en el metamodelo, será necesario crear los siguientes elementos: tendremos un solo dominio, (Dominio educativo que poseerá dos entidades: ejemplo y concepto) y tendremos dos relaciones: ilustra y prerrequisito. La cardinalidad tendrá valor 2. Además dos atributos: nombre y contenido.

2.4. Herramienta Protégé.

En el presente trabajo se utilizará específicamente la herramienta Protégé.

2.4.1. Algunas características de la herramienta Protégé.

Esta herramienta se utiliza para el desarrollo de Ontologías y Sistemas basados en el conocimiento. Fue creada en la Universidad de Stanford. Protégé está desarrollada en JAVA y puede funcionar perfectamente bajo WINDOWS. Las aplicaciones desarrolladas con Protégé son empleadas en resolución de problemas y toma de decisiones en dominios particulares. La herramienta Protégé emplea una interfaz de usuario que facilita la creación de una estructura de frames con clases, slots e instancias de una forma integrada.

Para su instalación, se necesita la máquina virtual de Java, aunque su instalador da la posibilidad de instalar una o utilizar una ya instalada en la máquina.

2.4.2. Algunos aspectos de las bases de datos orientadas a objetos.

Para hablar de Protégé tendríamos que referirnos a algunos aspectos de las bases de datos orientadas a objetos.

Jerarquía de clases.

En una base de datos existen objetos que responden a los mismos mensajes, utilizan los mismos métodos y tienen variables del mismo nombre y tipo. Sería inútil definir cada uno de estos objetos por separado por lo tanto se agrupan los objetos similares para que formen una clase, a cada uno de estos objetos se le llama instancia de su clase. Todos los objetos de su clase comparten una definición común, aunque difieran en los valores asignados a las variables.

Herencia.

Las clases en un sistema orientado por objetos se representan en forma jerárquica como en el diagrama anterior, así que las propiedades o características del elemento persona las contendrán (heredaran) los elementos alumno y maestro. Decimos que tanto la entidad Alumno y maestro son subclases de la clase persona este concepto es similar al utilizado en la de especialización (la relación ISA) del modelo E-R.

Se pueden crear muchas agrupaciones (clases) para simplificar un modelo así que una jerarquía (en forma gráfica) puede quedar muy extensa, en estos casos tenemos que tener bien delimitados los elementos que intervienen en una clase y aquellos objetos que las heredan.

Ventajas del Modelo orientado a objetos con respecto al modelo estructurado

- Un modelo de objetos es más cercano a la realidad que un modelo funcional.
- Un desarrollo realizado con el modelo orientado a objetos es más fácil de mantener y de reutilizar.
- El modelo orientado a objetos evita la redundancia en los procesos luego los códigos son más entendibles y resumidos
- La integridad que dan los objetos a los datos evita ambigüedades en su uso, dando mayor seguridad en los resultados

Capítulo II: Herramienta Protégé y el modelo de una asignatura teórico-práctica.

- El modelo orientado a objetos facilita la integridad de módulos que hallan sido realizados por separado sin correr riesgos en el manejo de los datos.

2.4.3. Facilidades del Protégé.

Esta herramienta permite:

Modelar clases: Protégé tiene una interfase gráfica para la modelación de las clases, sus atributos y relaciones.

Editar instancias: Protégé genera automáticamente formas interactivas que permiten entrar instancias válidas.

Fácil procesamiento: Protégé tiene una biblioteca de plug-ins que ayudan a definir la semántica, hacer consultas y definir el comportamiento lógico. Esta serie de plug-ins se pueden utilizar para utilizar máquinas de inferencias.

Diferentes formatos. El modelo (clases y instancias) pueden ser guardadas en varios formatos como XML, UML, RDF, etc.

Un conjunto de APIs para Java que facilita la creación de aplicaciones sobre la ontología.

Utilización de multimedia.

Mezcla y/o integración de ontologías.

Provee mecanismos de validación de la ontología.

Protégé tiene su propio lenguaje llamado Protégé Axiom Language (PAL). Por esto se plantea que la herramienta Protégé está quedando como estándar al ir incorporando cosas de otros de otros sistemas más específicos.

2.5 Modelado de la ontología con la herramienta Protégé.

A continuación se describe el modelo creado con la herramienta Protégé. Se utilizó la versión 3.1. Lo primero que se debe hacer es definir las clases y la jerarquía de la misma, para ello hay varias estrategias a seguir (Uschold-Gruninger, 1996):

- **top-down:** el desarrollo comienza con la definición de los conceptos más generales del dominio y luego en la especialización de esos conceptos
- **bottom-up:** el desarrollo comienza con la definición de las clases más específicas y luego agrupándolas en conceptos más generales.

- **una combinación de las anteriores:** primero se definen los conceptos salientes y luego se generalizan y/o especializan apropiadamente.

En nuestro caso hemos utilizado top-down, ya que lo primero que se definió fue la clase Dominio, luego Dominio_conceptual, Dominio_instruccional y terminar la especificación con las clases inferiores como Concepto, Actividad, Ejemplo, etc.

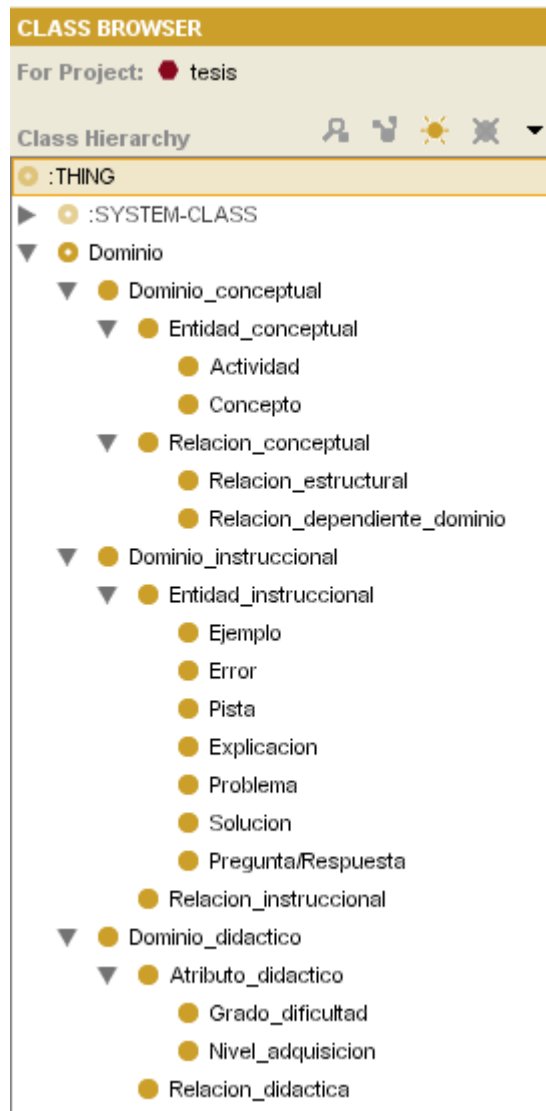


Figura 2.4 Clases definidas en Protégé utilizando top-down.

En el modelo tenemos la clase Dominio, que va a ser abstracta, ya que no va a poseer instancias. Ella va a heredar de la clase :THING, que está definida por defecto y que también

Capítulo II: Herramienta Protégé y el modelo de una asignatura teórico-práctica.

es abstracta. De la clase Dominio heredan Dominio_conceptual, Dominio_instruccional y Dominio_didactico.

El Dominio_conceptual, tendrá las clases Entidad_conceptual y Relacion_conceptual, que a su vez, Entidad_conceptual, poseerá las clases Actividad y Concepto; y Relacion_conceptual Relacion_estructural y Relacion_dependiente_dominio.

El Dominio_instruccional tiene Entidad_instruccional y Relacion_instruccional, que a su vez Entidad_instruccional tendrá las clases: Ejemplo, Error, Pista, Explicación, Problema, Solución, Pregunta/Respuesta.

El Dominio_didactico poseerá Atributo_didactico y Relacion_didactica y Atributo_didactico tendrá Grado_dificultad y Nivel_adquisicion.

La clase Entidad_conceptual va a tener los siguientes slots:

- **contenido_entidad_conceptual**: en él se guarda el contenido de la entidad.
- **nombre_entidad_conceptual**: este slot recoge el nombre de la entidad.

Las clases Concepto y Actividad como heredan de la clase Entidad_conceptual, también poseen estos slots.

La clase Relacion_conceptual va a tener los siguientes slots:

- **cardinal1_conceptual**: este slot recoge el nombre de un cardinal de la relación.
- **cardinal2_conceptual**: en él se recoge el nombre del otro cardinal de la relación.
- **nombre_relacion_conceptual**: este almacena el nombre de la relación.

Las clases Relacion_estructural y Relacion_dependiente_dominio como heredan de la clase Relacion_conceptual, también poseen estos slots.

La clase Entidad_instruccional va a tener los siguientes slots:

- **contenido_entidad_instruccional**: en él se guarda el contenido de la entidad.
- **nombre_entidad_instruccional**: este slot recoge el nombre de la entidad.

Capítulo II: Herramienta Protégé y el modelo de una asignatura teórico-práctica.

Las clases Ejemplo, Error, Pista, Explicación, Problema, Solución y Pregunta/Respuesta como heredan de la clase Entidad_instruccional, también poseen estos slots.

La clase Relacion_instruccional va a tener los siguientes slots:

- **cardinal1_instruccional**: este slot recoge el nombre de un cardinal de la relación.
- **cardinal2_instruccional**: en él se recoge el nombre del otro cardinal de la relación.
- **nombre_relacion_instruccional**: este almacena el nombre de la relación.

La clase Atributo_didactico va a tener el slot:

- **valor_atributo**: en él se recoge el valor de este atributo.

La clase Relacion_didactica va a tener los siguientes slots:

- **cardinal1_didactica**: este slot recoge el nombre de un cardinal de la relación.
- **cardinal2_didactica**: en él se recoge el nombre del otro cardinal de la relación.
- **nombre_relacion_didactica**: este almacena el nombre de la relación.

A continuación expondremos las instancias de cada una de las clases.

Instancias de la clase concepto:

- **Nombre_entidad_conceptual**: administrador de la Base de datos (ABD).
Contenido_entidad_conceptual: Es una persona, o grupo de personas, (personal técnico), que se encarga de: crear la base de datos, definir el esquema interno o diseño físico de la BD, establecer el enlace con los usuarios, definir restricciones de seguridad y de integridad, y supervisar el rendimiento, entre otras funciones.
- **Nombre_entidad_conceptual**: atributo identificador alternativo.
Contenido_entidad_conceptual: Cada conjunto de atributos que identifiquen unívocamente cada una de las ocurrencias de un tipo de entidad.
- **Nombre_entidad_conceptual**: atributo identificador principal.
Contenido_entidad_conceptual: Conjunto de atributos que identifican unívocamente cada una de las ocurrencias de un tipo de entidad.
- **Nombre_entidad_conceptual**: atributos.

Contenido_entidad_conceptual: Propiedades o características que tiene un tipo de entidad tipo de interrelación.

- **Nombre_entidad_conceptual:** base datos.

Contenido_entidad_conceptual: Colección o deposito de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de estas, y su definición y descripción, únicas para cada tipo de datos, han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad de los datos.

- **Nombre_entidad_conceptual:** datos.

Contenido_entidad_conceptual: Hechos conocidos que: pueden ser registrados y tienen un significado implícito.

- **Nombre_entidad_conceptual:** entidad.

Contenido_entidad_conceptual: Cualquier objeto (real o abstracto) sobre el cual queremos tener información y que tiene existencia por si mismo. Las entidades deben cumplir con algunas reglas como: tener existencia propia, cada ocurrencia de un tipo de entidad debe distinguirse de las demás, todas las ocurrencias de un tipo de entidad deben tener las mismas características.

- **Nombre_entidad_conceptual:** entidad débil.

Contenido_entidad_conceptual: su existencia depende de otro tipo de entidad.

- **Nombre_entidad_conceptual:** entidades regulares.

Contenido_entidad_conceptual: existen por sí mismas.

- **Nombre_entidad_conceptual:** interrelación.

Contenido_entidad_conceptual: Asociación o correspondencia entre entidades.

- **Nombre_entidad_conceptual:** modelo de datos.

Contenido_entidad_conceptual: Colección de herramientas conceptuales para describir los datos, relaciones entre ellos, semántica asociada a los datos y restricciones de integridad.

- **Nombre_entidad_conceptual:** modelo Entidad/Interrelación(E/R).

Contenido_entidad_conceptual: “... el modelo E/R puede ser usado como una base para una vista unificada de los datos...”, adoptando “el enfoque más natural del mundo real que consiste en entidades e interrelaciones”. Sus principales componentes son: entidades, interrelaciones y atributos.

- **Nombre_entidad_conceptual:** modelo relacional.

Contenido_entidad_conceptual: relacional representa las bases de datos como una colección de relaciones, en términos informales cada relación es similar a una tabla.

- **Nombre_entidad_conceptual:** sistema de base de datos.

Contenido_entidad_conceptual: Es una aplicación concreta se conforma sobre un SGBD y responde a una tarea específica: Sistema de Nominas, Sistema de Control de Inventarios.

- **Nombre_entidad_conceptual:** sistema de Gestión de Base de datos.

Contenido_entidad_conceptual: Conjunto coordinado de programas, procedimientos, lenguajes, etc., que suministra, tanto a los usuarios, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, seguridad y confidencialidad.

- **Nombre_entidad_conceptual:** tipo de interrelación.

Contenido_entidad_conceptual: estructura genérica del conjunto de interrelaciones existentes entre dos o más tipos de entidades.

Instancias de la clase Ejemplo:

- **Nombre_entidad_instruccional:** atributo.

Contenido_entidad_instruccional: Nombre y Salario de un Empleado.

- **Nombre_entidad_instruccional:** base datos.

Contenido_entidad_instruccional: Problema: Base de datos Universidad, para mantener la información relacionada con estudiantes, asignaturas, profesores y notas en un entorno universitario. Solución: La base de datos esta organizada en 4 archivos cada uno de los cuales almacena registros del mismo tipo. El archivo Estudiante va a recoger los datos: nombre del estudiante, numero del estudiante y su fecha de nacimiento. El archivo Profesor tendrá: nombre del profesor y número del profesor. El de Asignatura:

Capítulo II: Herramienta Protégé y el modelo de una asignatura teórico-práctica.

número de la asignatura, su nombre y el número del profesor que la imparte. Nota va a tener: Numero de la asignatura, numero del estudiante y la nota.

- **Nombre_entidad_instruccional:** datos.
Contenido_entidad_instruccional: Nombres, números telefónicos y direcciones de personas conocidas.
- **Nombre_entidad_instruccional:** entidad.
Contenido_entidad_instruccional: Empleado y Proyecto.
- **Nombre_entidad_instruccional:** relación.
Contenido_entidad_instruccional: Relación Trabaja-En entre las entidades Empleado y Proyecto.

Instancias de la clase Relacion_estructural:

- **Nombre_relacion_conceptual:** parte_de.
Cardinal1_conceptual: concepto.
Cardinal2_conceptual: concepto
- **Nombre_relacion_conceptual:** tipo_de.
Cardinal1_conceptual: concepto.
Cardinal2_conceptual: concepto.

Instancias de la clase Relacion_dependiente_dominio:

- **Nombre_relacion_conceptual:** induce.
Cardinal1_conceptual: concepto.
Cardinal2_conceptual: concepto.
- **Nombre_relacion_conceptual:** produce.
Cardinal1_conceptual: actividad.
Cardinal2_conceptual: concepto.
- **Nombre_relacion_conceptual:** se_realiza_sobre.
Cardinal1_conceptual: actividad.
Cardinal2_conceptual: concepto.

Instancias de la clase Relacion_instruccional:

Capítulo II: Herramienta Protégé y el modelo de una asignatura teórico-práctica.

- **Nombre_relacion_conceptual:** es_solucion.
Cardinal1_conceptual: problema.
Cardinal2_conceptual: concepto.
- **Nombre_relacion_conceptual:** explica.
Cardinal1_conceptual: explicación/pista.
Cardinal2_conceptual: problema/ejemplo/error/solución.
- **Nombre_relacion_conceptual:** responde_a.
Cardinal1_conceptual: pregunta.
Cardinal2_conceptual: respuesta.

Instancias de la clase Relacion_didactica:

- **Nombre_relacion_conceptual:** ejercita.
Cardinal1_conceptual: problema.
Cardinal2_conceptual: concepto/actividad.
- **Nombre_relacion_conceptual:** ilustra.
Cardinal1_conceptual: ejemplo.
Cardinal2_conceptual: concepto.
- **Nombre_relacion_conceptual:** involucra.
Cardinal1_conceptual: problema/ejemplo.
Cardinal2_conceptual: concepto.
- **Nombre_relacion_conceptual:** prerrequisito.
Cardinal1_conceptual: concepto/actividad.
Cardinal2_conceptual: concepto/actividad.
- **Nombre_relacion_conceptual:** tiene_faq.
Cardinal1_conceptual: concepto/actividad/problema/ejemplo.
Cardinal2_conceptual: pregunta.

Instancias de la clase Grado_dificultad:

- **Valor_atributo:** difícil.
- **Valor_atributo:** fácil.
- **Valor_atributo:** medio.

Instancias de la clase Nivel_adquisición:

- **Valor_atributo:** aplicar.
- **Valor_atributo:** comprender.
- **Valor_atributo:** conocer.

2.6. Conclusiones del capítulo.

En este capítulo abordamos como ha influenciado la tecnología en la Enseñanza. También formulamos un modelo para representar una asignatura teórico – práctica y se expusieron algunas características de la herramienta Protégé, la que utilizamos para modelar la ontología.

En el siguiente capítulo se hablará de una consulta realizada al Protégé y una aplicación que se realizó para mostrar los resultados de dicha consulta.

Capítulo III:
MANUAL DE USUARIO DEL
PROTÉGÉ.



Capítulo III. MANUAL DE USUARIO DEL PROTÉGÉ.

En este capítulo describe el manual de usuario del Protégé y además se hace referencia a la implementación de un pequeño entorno educativo, describiendo su manual de usuario, además de una consulta realizada en Protégé, que utiliza dicho entorno educativo. Con este fin este capítulo se ha estructurado de siguiente manera:

- El epígrafe 3.1 se refiere a los pasos a seguir para la creación de un proyecto en Protégé.
- En el epígrafe 3.2 se abordan las clases del Protégé.
- En el epígrafe 3.3 se trata el modo de crear slots.
- En el epígrafe 3.4 se refiere a la creación de instancias.
- En el epígrafe 3.5 se abordan las consultas en Protégé y la que se realizó para que utilizara el entorno educativo.
- En el epígrafe 3.6 se realiza el manual de usuario del pequeño entorno educativo.

3.1. Creación de un proyecto en el Protégé.

A continuación se explica detenidamente los pasos a seguir para crear, abrir y salvar un proyecto en Protégé.

3.1.1. Inicio del Protégé.

Al iniciar el Protégé, lo primero que se muestra es una ventana, que la misma se muestra a continuación en la figura 3.1, donde se opta por crear un nuevo proyecto (oprimiendo Create New Project...) o abrir un proyecto existente (oprimiendo Open Existing Project...) o acceder a los proyectos más recientes (Recently Accessed Projects).

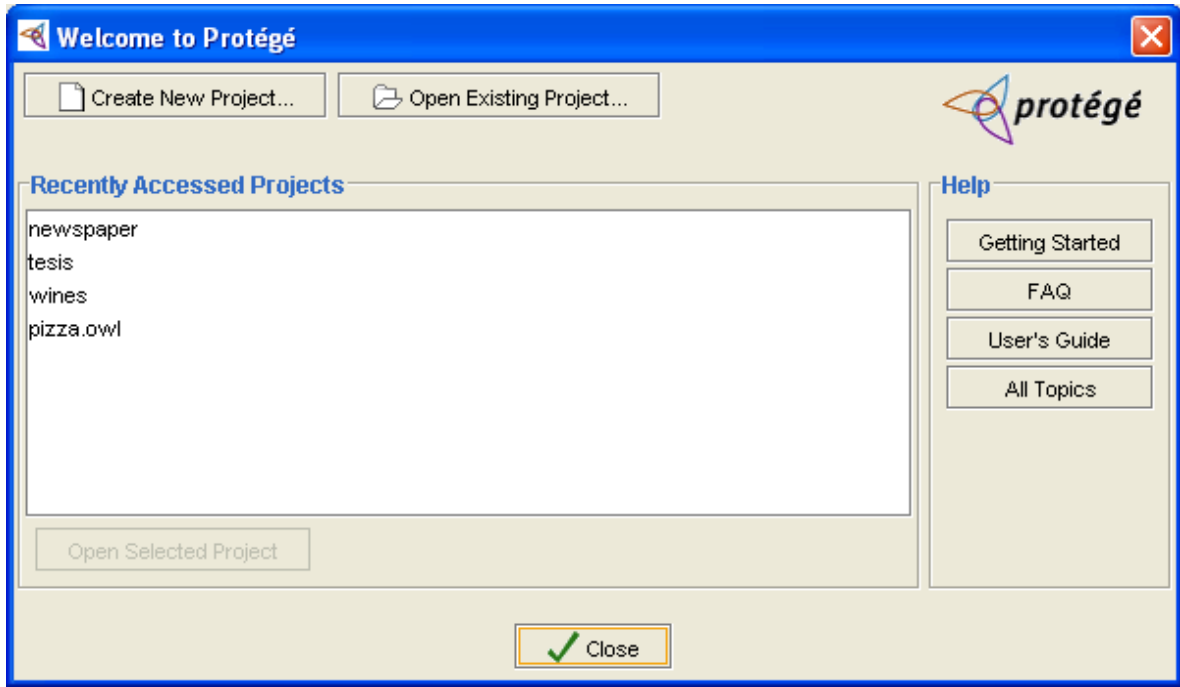


Figura 3.1 Ventana de inicio del Protégé.

3.1.1.1. Acceder a los proyectos más recientes.

En caso de que se acceda a *Recently Accessed Projects*, se elige el proyecto que se quiere abrir, y se habilitará el botón *Open Selected Project*, que al oprimirlo mostrará el proyecto seleccionado.

3.1.1.2. Abrir un proyecto existente.

Si se escoge *Open Existing Project* se muestra una ventana que aparece en la figura 3.2, para buscar el proyecto deseado. Luego de buscado y seleccionado el proyecto, se oprime el botón OK y se abre dicho proyecto, si desea cancelar esta acción, se oprime Cancel.

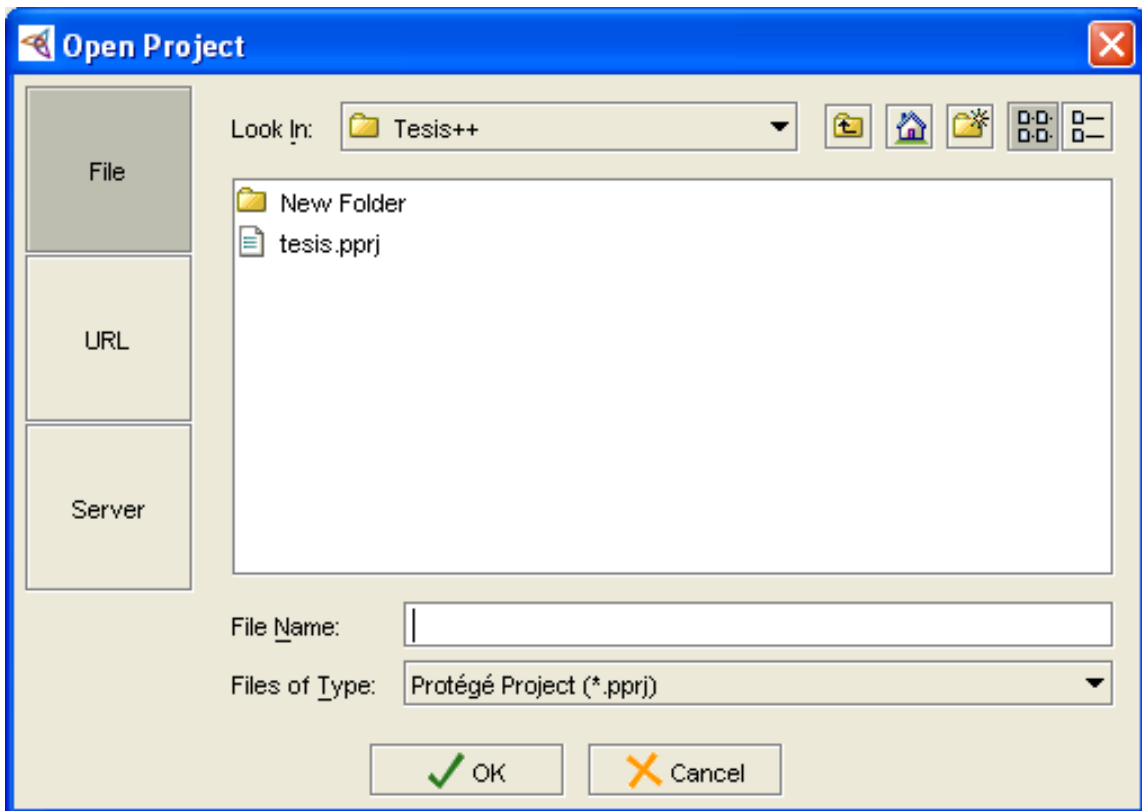


Figura 3.2 Abrir proyecto

3.1.1.3. Crear un proyecto nuevo.

Si se selecciona *Create New Project*, aparece una ventana para que escoja el tipo de proyecto que se va a realizar. Esto se muestra en la figura 3.3

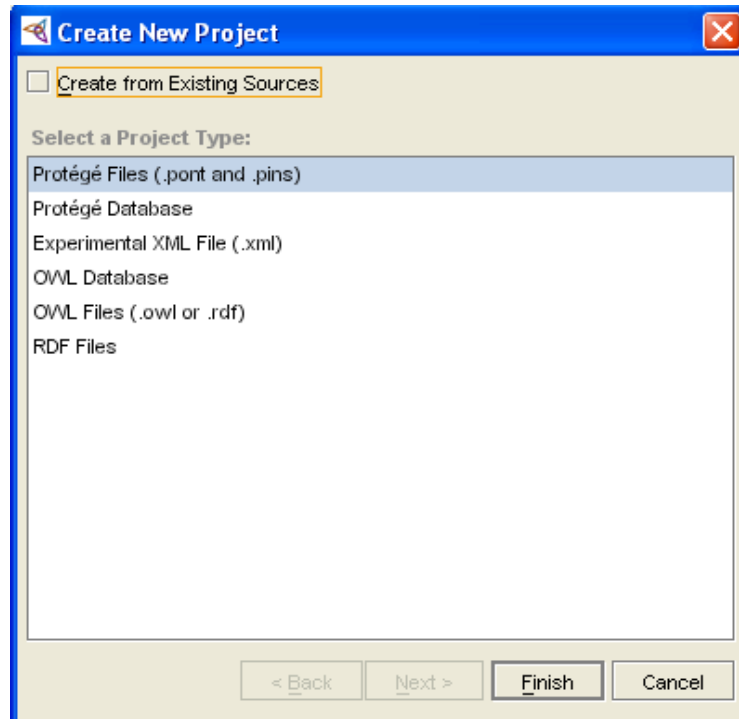


Figura 3.3 Crear un nuevo proyecto

3.1.2. Nuevo proyecto en Protégé.

Al crear un nuevo proyecto de Protégé, lo más recomendable es salvar el proyecto. Para esto, se despliega el en el menú File y se oprime Save Project As y aparece una ventana que se muestra en la figura 3.4.

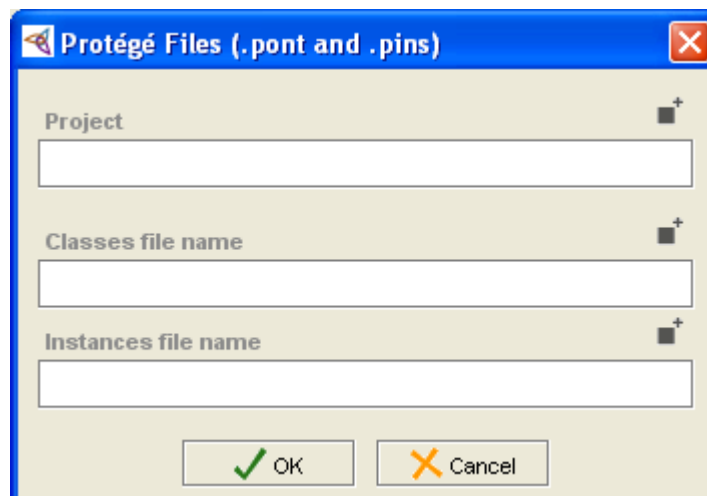



Figura 3.4 Save Project As en Protégé.

En Project, se observa el *Browse for file*  , que nos muestra la ventana que se ofrece en la figura 3.5 para que el usuario examine y defina el lugar donde va a guardar el proyecto.

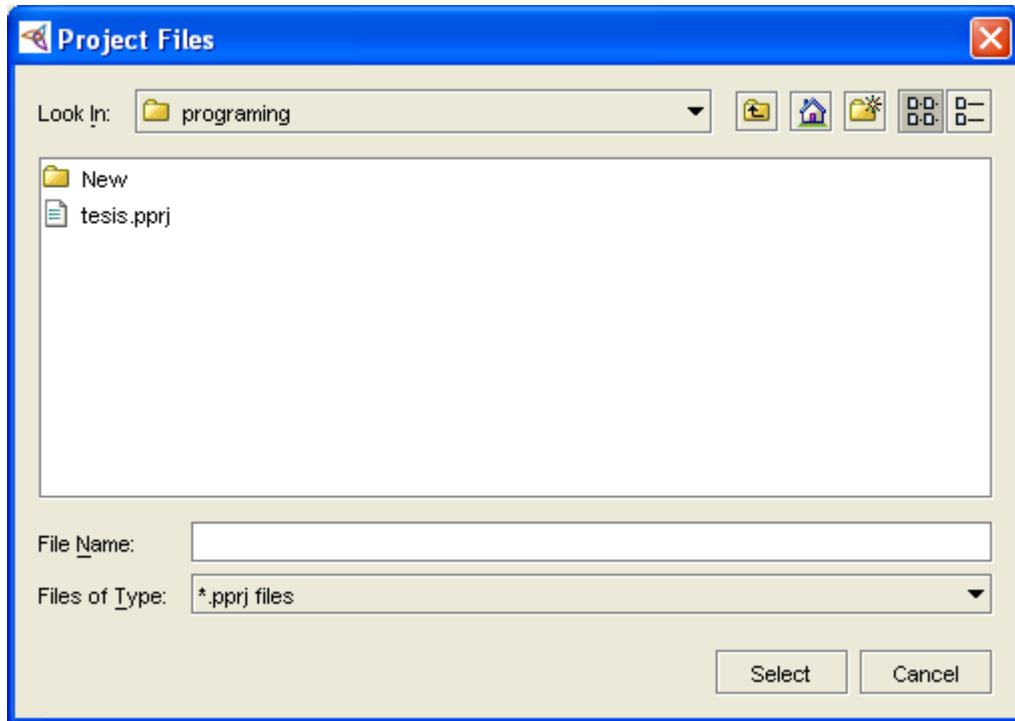


Figura 3.5 Examinar ficheros.

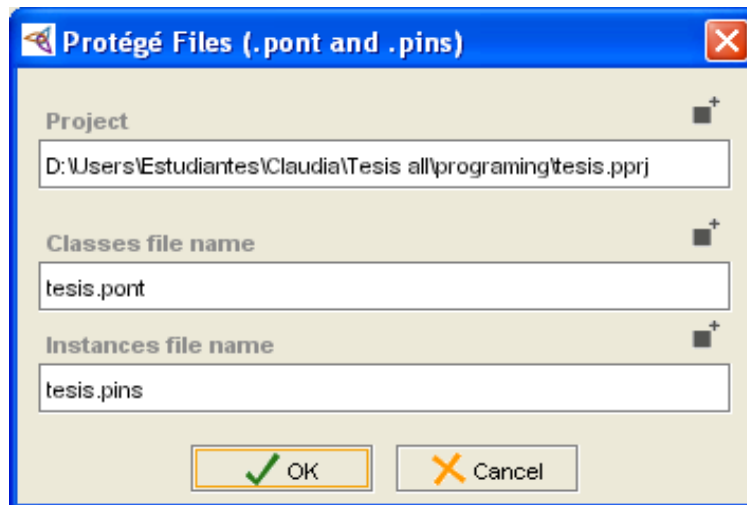


Figura 3.6 Guardar proyecto.

Protégé guarda los proyectos creados con la extensión pprj. Además, crea dos archivos (en formato de texto) que contienen la información adicional sobre la ontología y las instancias del proyecto. Al oprimir el botón Select aparecen automáticamente el nombre del fichero de la ontología (contiene la clase y la información del slot) (.pont) y el de instancias (.pins) como se muestra en la figura 3.6.

Después oprimimos OK y se guardan los ficheros, o se oprime Cancel para impedir que sea guardados.

3.2. Clases.

La pestaña Classes es la pestaña utilizada para visualizar y editar las clases. Las clases y sus relaciones jerárquicas se visualizan en el Class Browser. En la figura 3.7 se puede observar la jerarquía de clases para nuestro proyecto.

Todas las clases se muestran como descendiendo, directamente o indirectamente, de la clase :THING del sistema. Es destacable el hecho de la diferenciación entre clases abstractas y concretas. Las clases concretas pueden tener instancias directas y las abstractas no incluyen instancias.

3.2.1. Jerarquía de clases.

En una ontología representada en Protégé las clases se organizan de forma jerárquica, mediante la relación “subclase de” (o “es un”). Por ejemplo: dentro de la clase Dominio_conceptual, existen subclases que se corresponden con distintos subconjuntos:

Relacion_conceptual es una subclase de Dominio_conceptual.

Relacion_estructural, Relacion_dependiente_dominio son subclases de Relacion_conceptual.

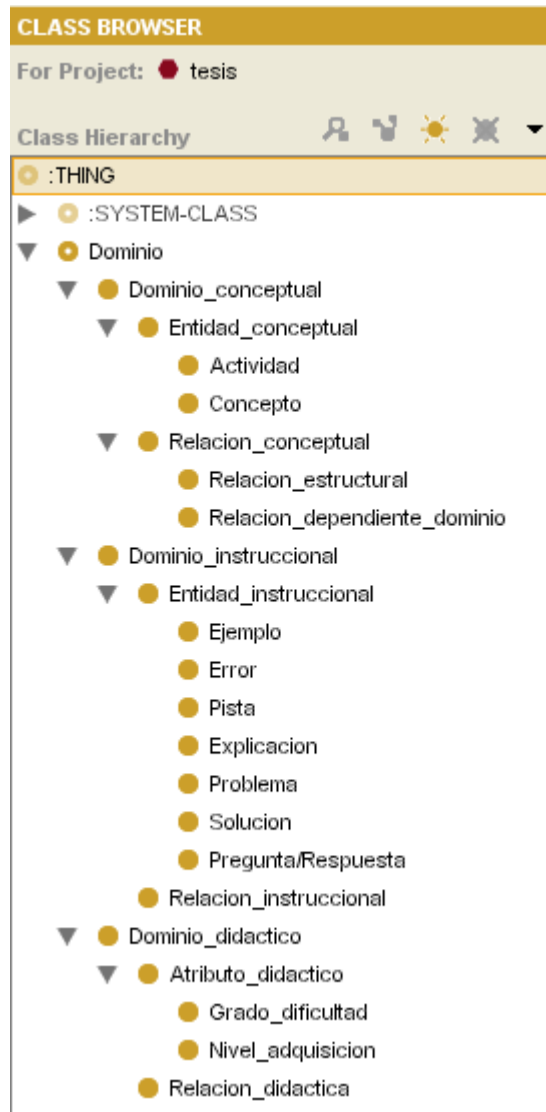



Figura 3.7 Clases.

3.2.2. Crear una nueva clase.

Para crear una nueva clase, si es la primera clase que va a realizar, asegúrese de tener seleccionada la clase `:THING`; sino seleccione la clase de la que va a heredar su nueva clase. Luego se pincha el botón *New Class* (Nueva clase) . Aparece una nueva clase (con un nombre automático) y en *CLASS EDITOR* se muestra el nombre resaltado par poder editarse. En la figura 3.8 se muestran los elementos a ingresar de las clases.

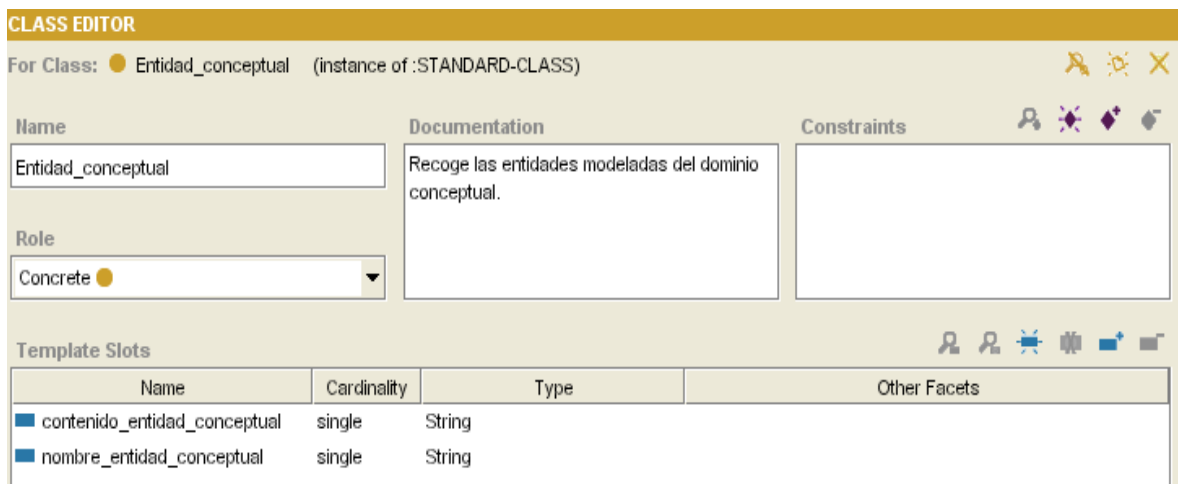



Figura 3.8 *CLASS EDITOR*

Para crear una clase, hay que ingresar los siguientes elementos:

- Nombre (*Name*): nombre de la clase. Se recomienda comenzar con mayúscula y seguir con minúsculas; para separar palabras, usar guión bajo.
- Documentación (*Documentation*): es para ingresar un texto con el fin de describir y documentar la clase en cuestión. El campo es opcional, pero se recomienda utilizarlo siempre con el objetivo de ayudar al mantenimiento de la ontología.
- Restricciones (*Constraints*):
- Rol (*Role*): el rol de la clase puede ser concreta o abstracta.
- *Slots* o propiedades (*Template Slots*): representan las propiedades de la clase y juegan un papel fundamental en la definición de la ontología. Si la clase que se está editando es subclase de otra, va a haber dos tipos de *slots*: los propios de la clase y los heredados.

3.3. Los conceptos.

Una vez que se definieron las clases, hay que describir la estructura interna de los conceptos (propiedades o *slots* de las clases). Para crear un *slot*, lo primero es seleccionar la clase a la que le añadirá el *slot*. Luego se oprime el botón  *Create Slot* (Crear concepto).

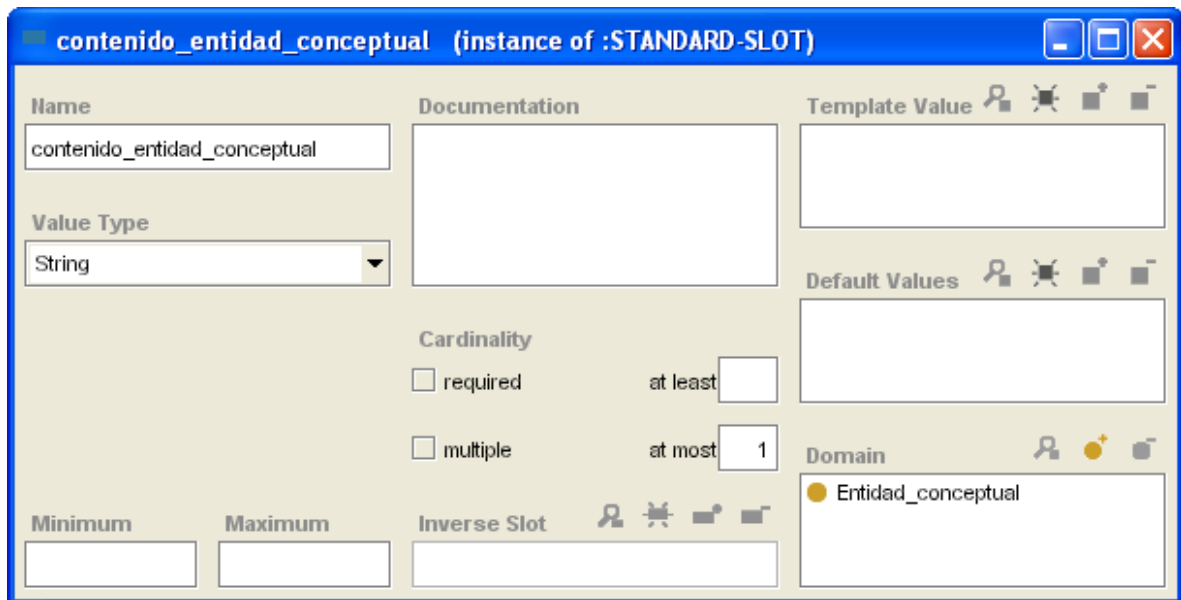


Figura 3.9 Elementos a ingresar en un slot.

Para definir una propiedad (slot), se deben ingresar los siguientes elementos:

- Nombre (Name): nombre del slot.
- Tipo de valor (Value Type): el tipo de valor determina los valores que puede asumir un slot. En la figura 3.10 muestra los tipos de valor de un slot.

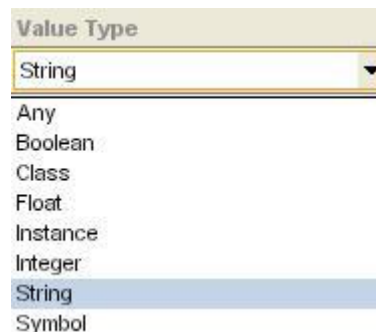


Figura 3.10 Tipo de valor de un slot.


Los tipos disponibles son:

- Any: cualquiera de los tipos anteriores.
- Boolean: valor booleano.
- Class: clase definida en la ontología.

- *Float*: valor real.
- *Instance*: instancia de una clase de la ontología.
- *Integer*: valor entero.
- *String*: cadena de caracteres.
- *Symbol*: lista de valores enumerados.
- Documentación (*Documentation*): es para ingresar un texto con el fin de describir y documentar el slot en cuestión. El campo es opcional, pero se recomienda utilizarlo siempre con el objetivo de ayudar al mantenimiento de la ontología.
- Valores (*Template Values*): permite especificar el o los valores para un slot a nivel de clase. Este valor se llena en todas las clases y las instancias que usan o heredan el slot. Un *template value* es requerido y no puede ser cambiado o sobrescrito a nivel de instancia. Para un valor que pueda ser sobrescrito, hay que usar el elemento *Defaults*. El número de *template values* debe respetar la cardinalidad del slot.
- Valores por omisión (*Defaults Values*): permite especificar los valores por omisión del slot. Cuando se crea una instancia de una clase a la que pertenece el slot, se setea automáticamente el valor por omisión definido. Este valor puede ser cambiado. El número de valores por omisión debe respetar la cardinalidad del slot.
- Cardinalidad (*Cardinality*): permite especificar el número de valores permitidos o requeridos para un slot. Si no se ingresa cardinalidad se permite al slot tener a lo sumo un valor o ninguno.
- Mínimo (*Mininumum*) / Máximo (*Maximum*): es para especificar los valores mínimos y máximos que puede tener un slot de tipo *Integer* o *Float*. Este campo es opcional. Cuando se crea una instancia de la clase a la que pertenece este slot, la instancia deberá respetar los valores de mínimo y máximo definidos.
- Slot inverso (*Inverse Slot*): también es opcional y sólo tiene validez cuando el slot es de tipo *Class* o *Instance*. Permite crear una relación recíproca entre dos slots. Si se define correctamente esta relación, asignando un valor (por ejemplo, una clase específica o una instancia) al slot de una instancia, automáticamente asigna la instancia como valor del slot inverso apropiado.

A la hora de definir los slots, se ha de tener en cuenta, que no pueden existir dos slots con un mismo nombre, puesto que el sistema no lo admite.

3.4. Las instancias.

Definida la jerarquía de clases y la estructura interna de cada clase, se está en condiciones de crear las instancias de cada clase. Para ello, hay que pinchar en la pestaña *Instances* y luego posicionarse en la clase deseada, oprimir el botón  *Create Instance* (crear una instancia), y de la misma ingresar los valores de los slots. En la figura 3.11 se muestran las instancias de la clase concepto.

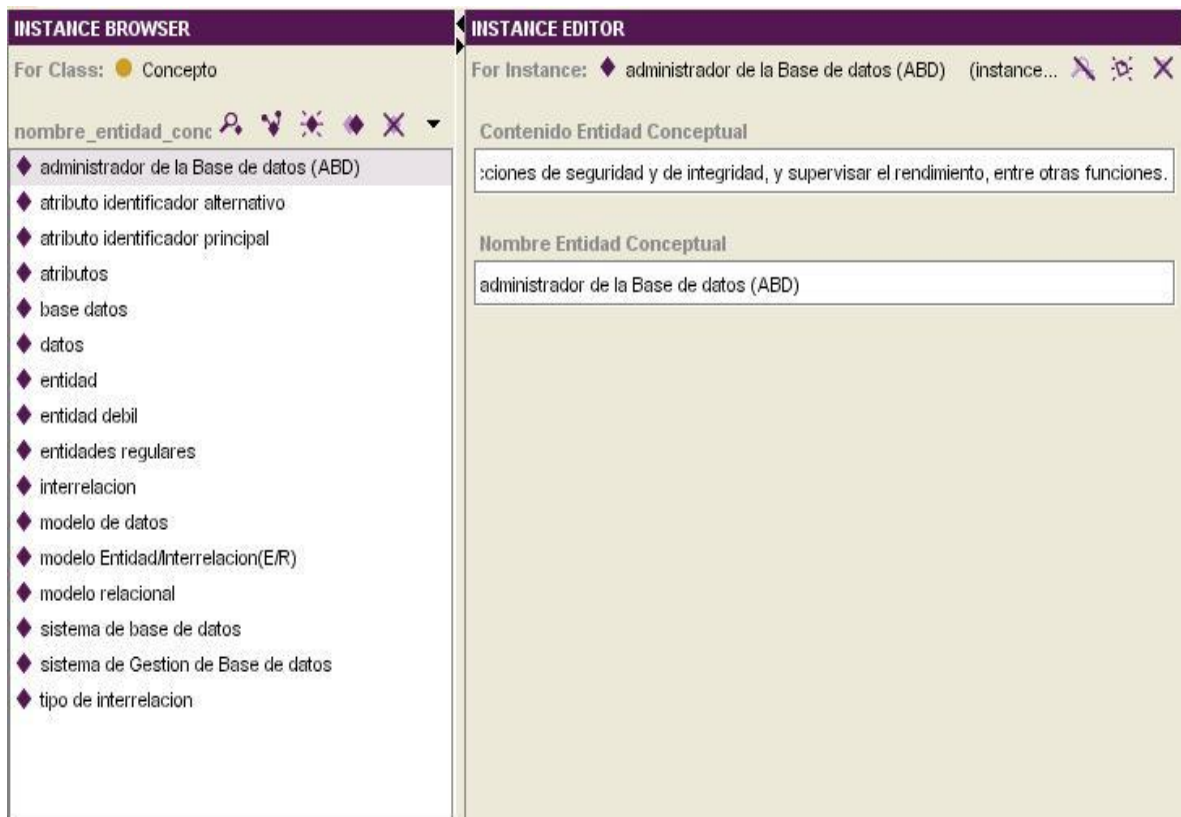


Figura 3.11 Instancias de la clase Concepto.

En un inicio muestra los nombres de las instancias que ellas traen por defecto. En la figura 3.12 se expresa como se puede mostrar, en nuestro caso, nombre_entidad_conceptual, para que sea dicho slot el que se presente en el *INSTANCE BROWSER*.

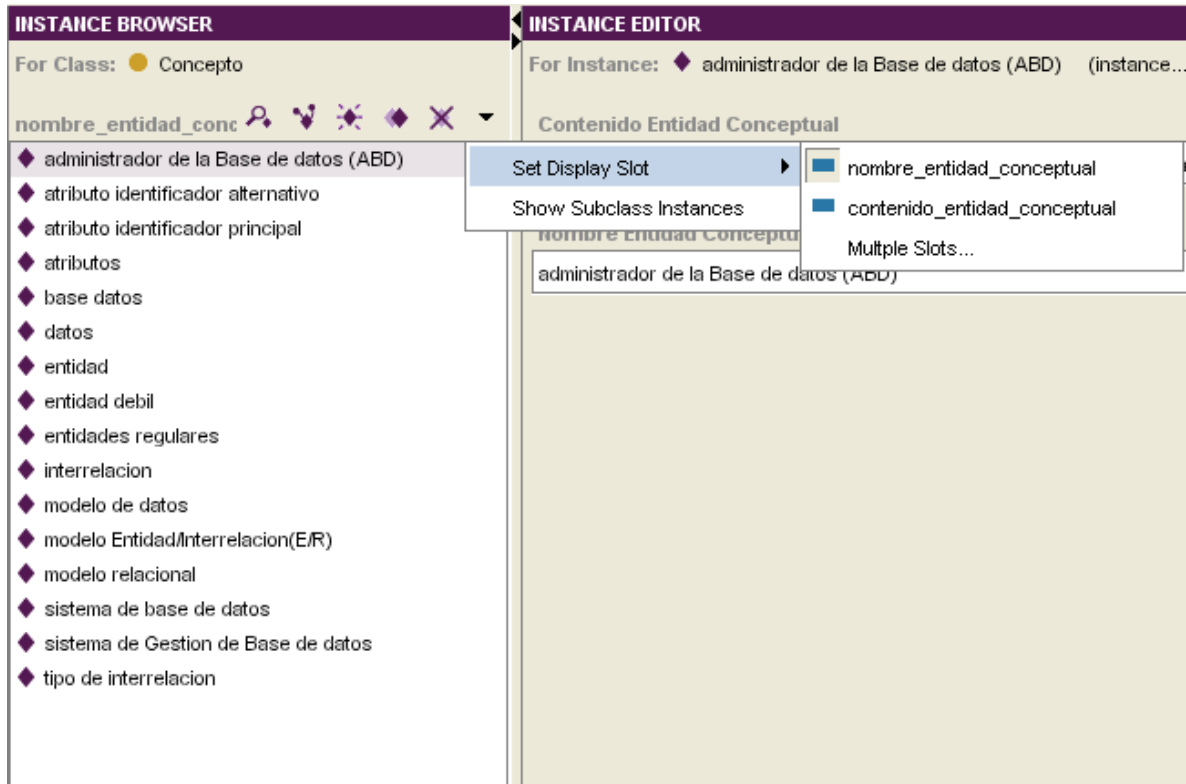


Figura 3.12 *Set Display Slot* en las instancias.

3.5. Las consultas.

En la herramienta Protégé, en la sección Queries, se realiza una consulta al modelo educacional. La misma lleva el nombre de Concepto_Ejemplo. En la figura 3.13 podemos ver en Query Library nuestra consulta.

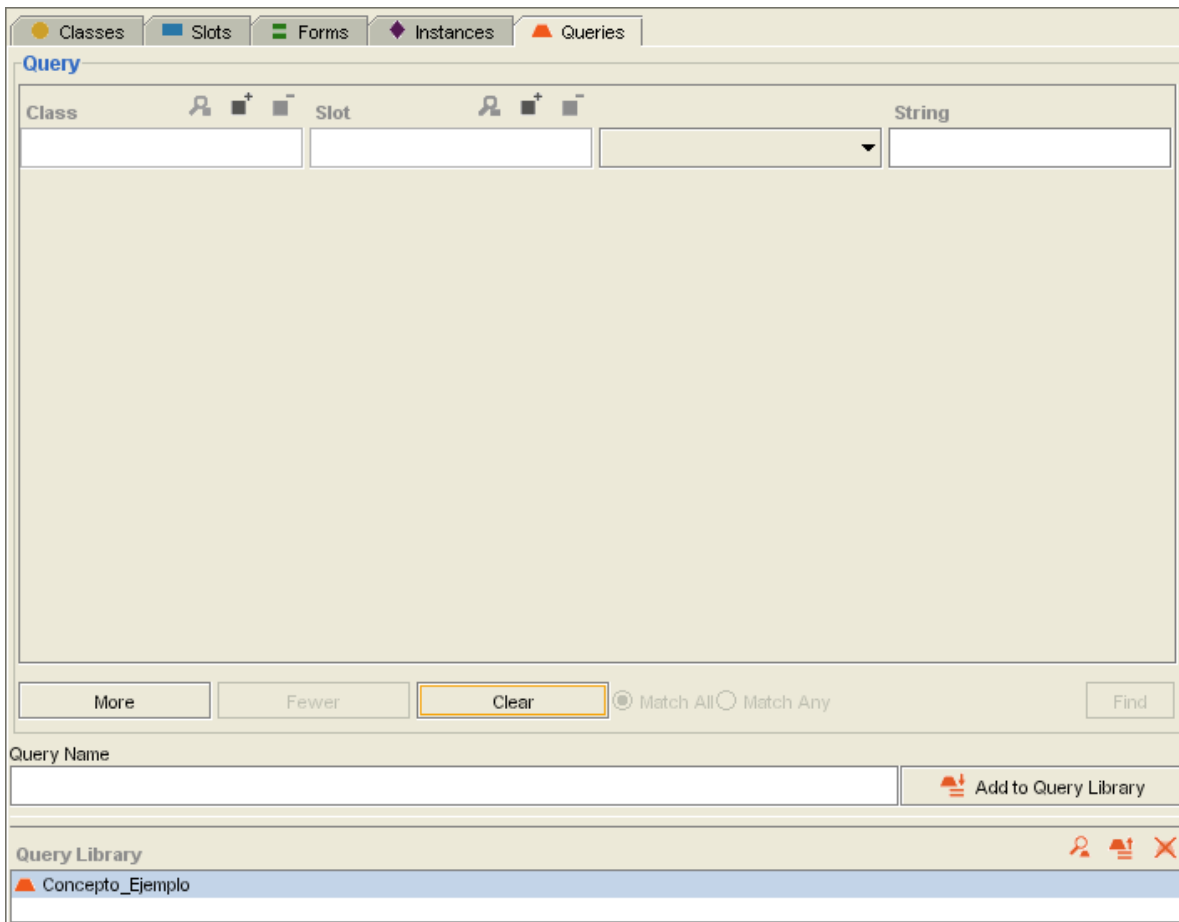



Figura 3.13 Queries en Protégé.

Cuando oprimimos el botón  (*Retrieve Query*) nos aparece el contenido de la consulta. El mismo se muestra en la figura 3.14. La Query tendrá a las clases Concepto y Ejemplo, con los Slot nombre_entidad_conceptual y nombre_entidad_instruccional respectivamente. En el siguiente campo se selecciona lo que se va a seleccionar; en nuestro caso se escogió *does not contain* (no contiene) en ambos casos y en String pondremos concepto y ejemplo respectivamente. Con esto se logra que la consulta nos muestre todas las instancias de Concepto y Ejemplo.

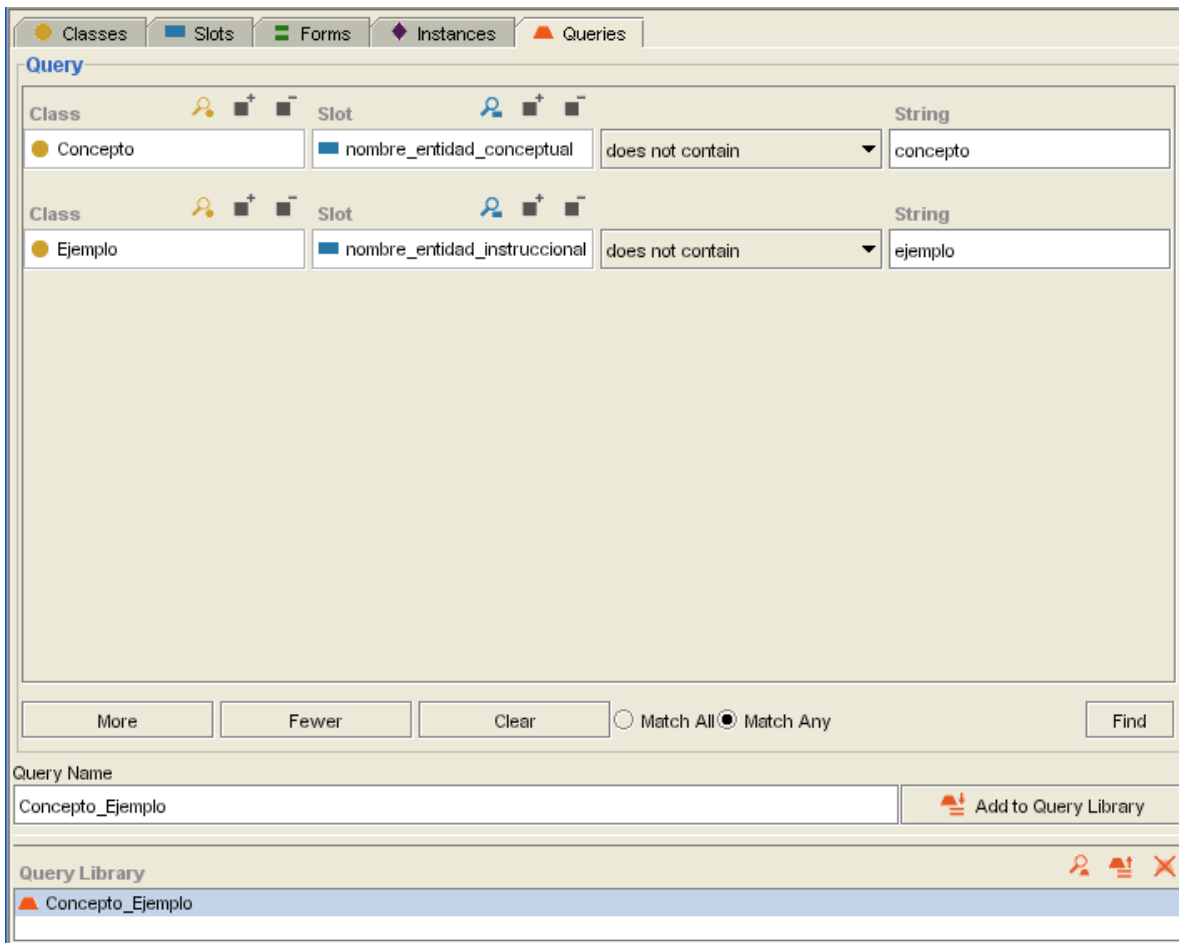


Figura 3.14 Query Concepto_Ejemplo

Se selecciona *Match Any*, para que se muestren indistintamente los conceptos y los ejemplos. Luego se oprime el botón *Find* y se muestran los resultados en la figura 3.15.


Se observa que en *Search Results* se tiene un botón  que exporta los valores de los Slots (*Export Slot Values to protege_query_results.txt*) para un fichero de nombre *protege_query_results.txt*, que se salvará en la misma carpeta del modelo.



Figura 3.15 Resultados encontrados por la consulta.

Al oprimir el botón aparece una forma para seleccionar los Slots que se van a exportar.

Esto se muestra en la figura 3.16. Luego se oprime el botón OK y se guardan toda la información seleccionada en el fichero antes mencionado.

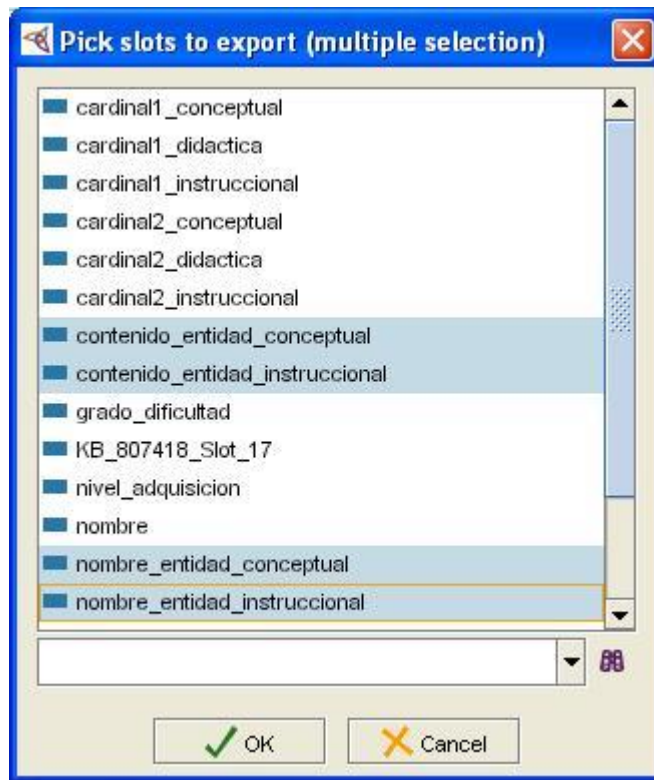



Figura 3.16 Ventana para seleccionar los Slots a exportar.

3.6. Manual de usuario del entorno educativo.

Nuestra aplicación se llama Entorno educativo y el icono que utiliza es el icono del Protégé,

el mismo se muestra a continuación . En la aplicación lo primero que se muestra es una imagen en la que se muestra una computadora, ceros y unos y el mundo, lo cual nos quiere dar a entender la importancia que tiene la tecnología y la computación en el mundo y en nuestro caso en la Enseñanza. Esta imagen se muestra en la figura 3.17.

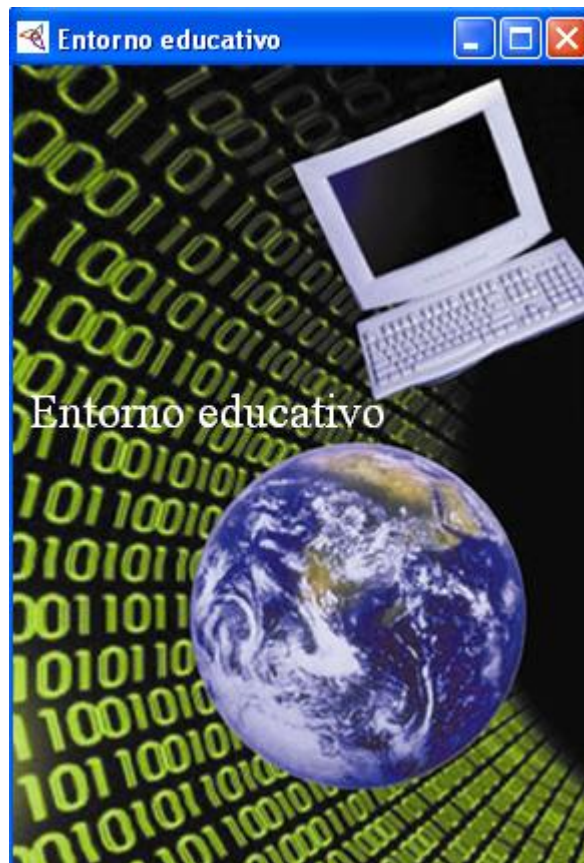


Figura 3.17 Imagen de bienvenida del entorno educativo.

Luego se muestra la aplicación en la cual tenemos un menú, una etiqueta, un edit y un botón. El menú File, se despliega para mostrar Abrir y Cerrar. Abrir tiene un submenú que tiene Materia, que se despliega y tiene Asignatura, que se expande y tiene Base de Datos. La etiqueta tiene como nombre Asignatura, y el edit Base de Datos, el mismo no se puede modificar, ya que esa es la única asignatura que hasta ahora se tiene en el modelo de la herramienta Protégé. También tiene el botón OK. Esto se muestra en la figura que se muestra a continuación, la figura 3.18. La siguiente figura que se muestra (figura 3.19) es el menú File con todos sus submenús desplegados.

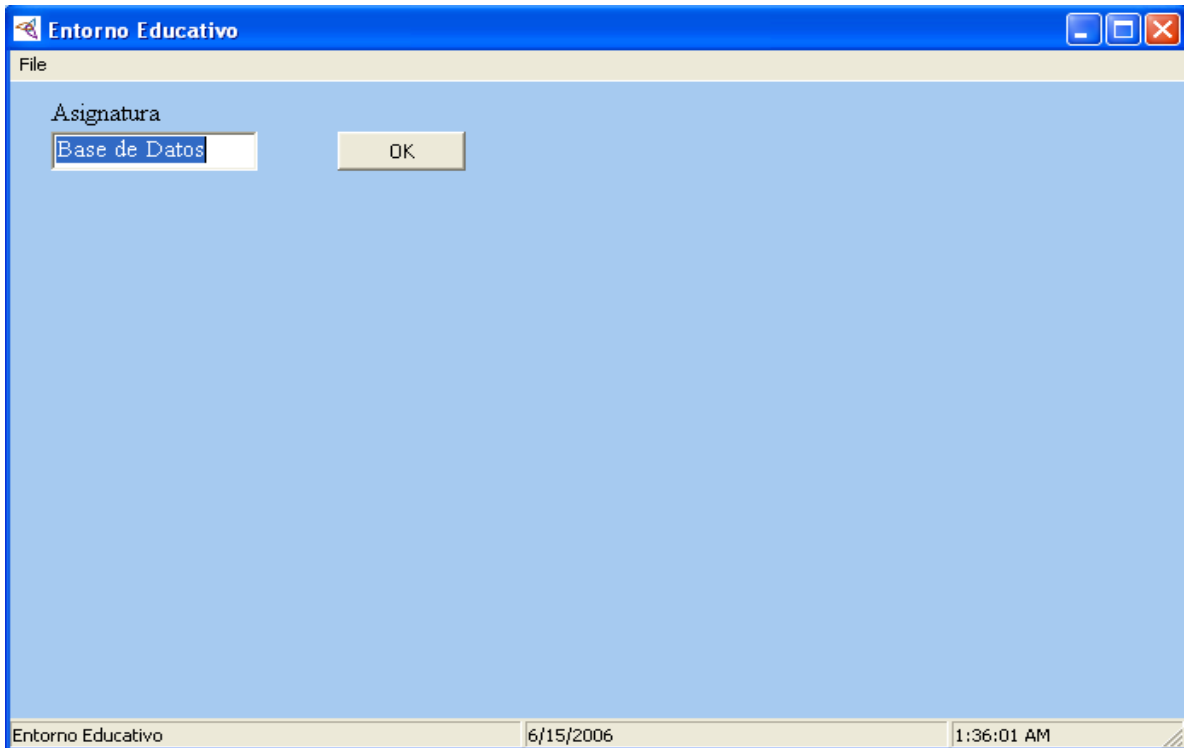


Figura 3.18 Entorno educativo.

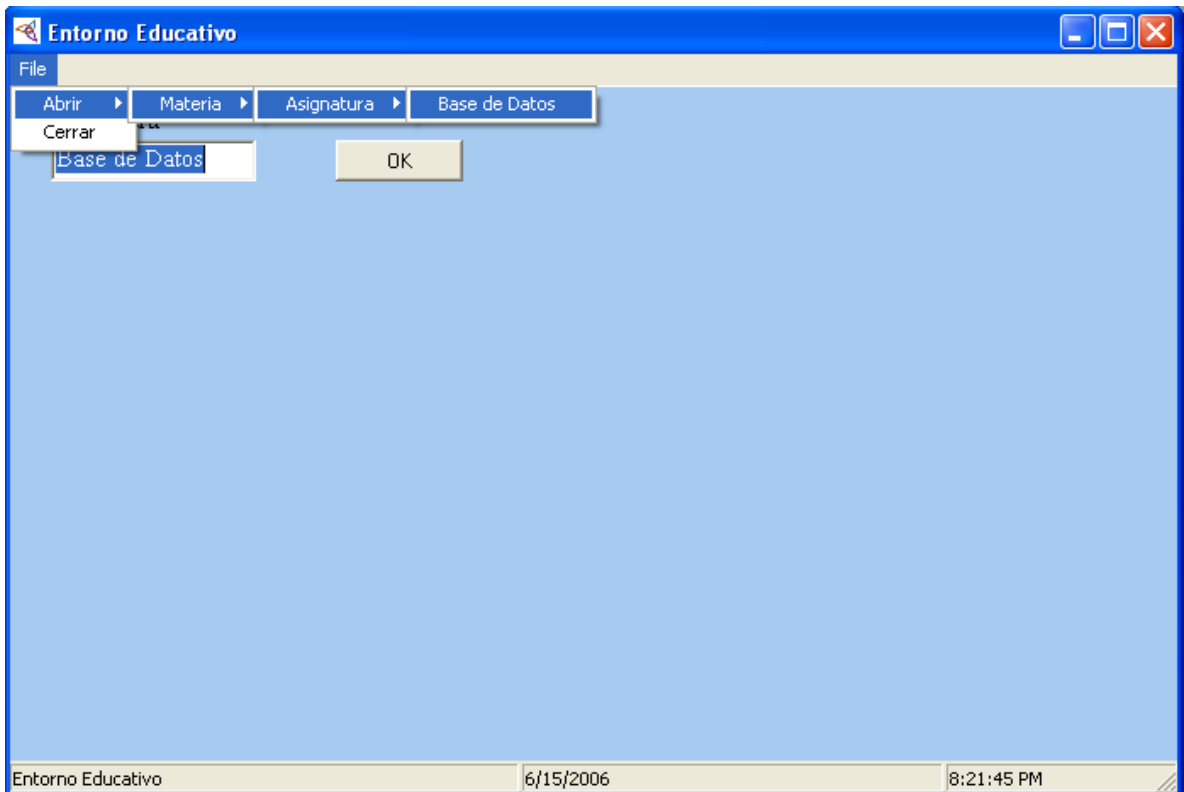


Figura 3.19 Menú File desplegado.

Al oprimir el submenú Base de Datos, o el botón OK, surgen cuatro botones como se muestra en la figura 3.20:

- **Mostrar conceptos:** Muestra la etiqueta conceptos, el ComboBox para que se seleccione el concepto, el Memo que recoge el contenido del concepto, una etiqueta que se encuentra vacía, pero que al seleccionar un concepto muestra *Concepto de* el concepto seleccionado; y el botón Ilustrar con ejemplo. A su vez desaparece la etiqueta de nombre Asignatura, y el edit Base de Datos, al igual que el botón OK. Por defecto, esto aparece en la aplicación, pero al oprimir el botón Mostrar ejemplos estos elementos se ocultan.
- **Mostrar ejemplos:** Muestra la etiqueta ejemplos, el ComboBox para que se seleccione el ejemplo, el Memo que recoge el contenido del ejemplo, una etiqueta que se encuentra vacía, pero que al seleccionar un ejemplo muestra *Ejemplo de* el ejemplo seleccionado y desaparece el botón Ilustrar con ejemplo.
- **Ilustrar con ejemplo:** Muestra el ejemplo, si tiene, del concepto seleccionado, y si el mismo no tiene, muestra un mensaje *El concepto seleccionado no tiene ejemplo*. Este botón se deshabilita al oprimir el botón Mostrar ejemplo.
- **Cerrar:** Cierra la aplicación.

Además de los botones, tiene también:

- **Etiqueta Conceptos**
- **ComboBox:** Cuando se despliega el ComboBox se encuentran los conceptos, para que el usuario seleccione el de su interés.
- **Memo:** Al ser seleccionado un concepto o un ejemplo, se muestra su contenido en el Memo. También al oprimir el botón Ilustrar con ejemplo, si el concepto seleccionado tiene ejemplo, el mismo es mostrado en el Memo.

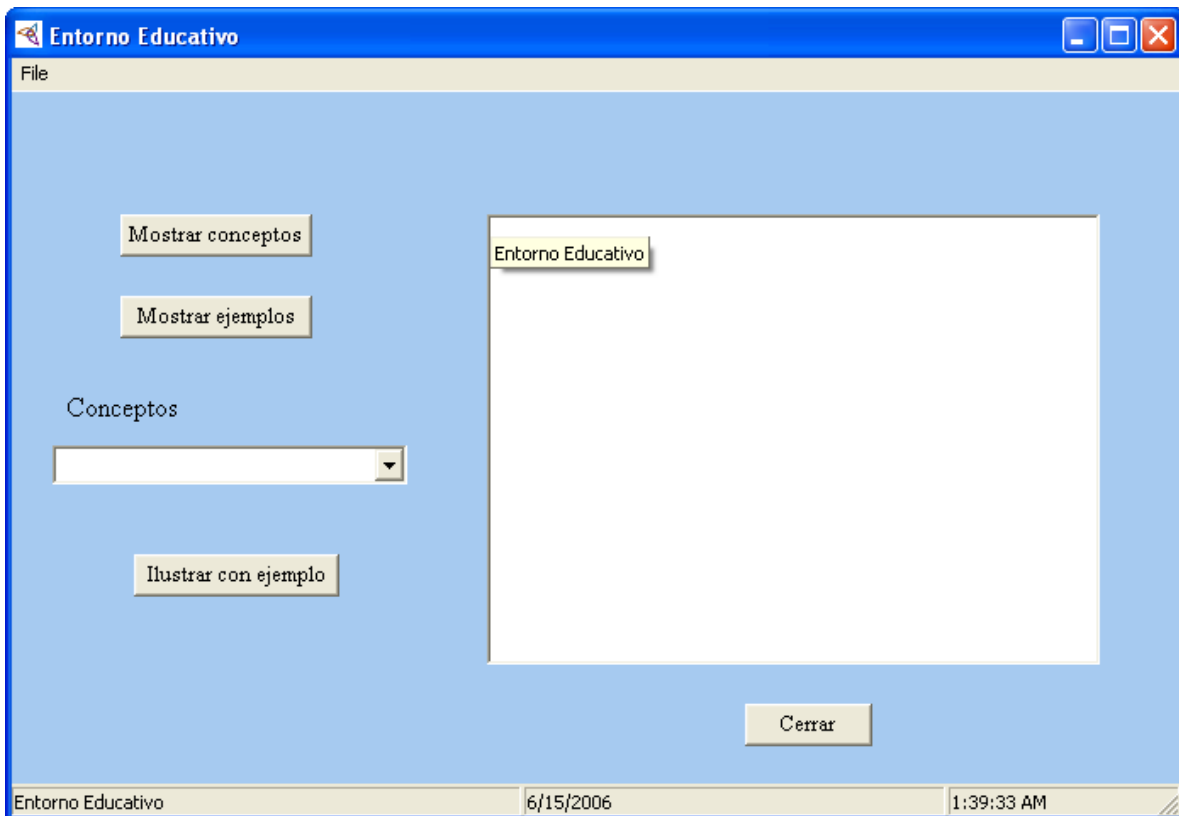


Figura 3.20 Ventana para mostrar conceptos.

Al oprimir el botón **Mostrar ejemplos**, como aparece en la figura 3.21, se muestran:

- **Etiqueta Ejemplos.**
- **ComboBox:** Cuando se despliega el ComboBox se encuentran los ejemplos, para que el usuario seleccione el de su interés.
- **Memo:** Al ser seleccionado un ejemplo se muestra su contenido en el Memo.
- Los botones **Mostrar conceptos**, **Mostrar ejemplos** y **Cerrar**, que ya anteriormente se explicó que función realiza cada uno.

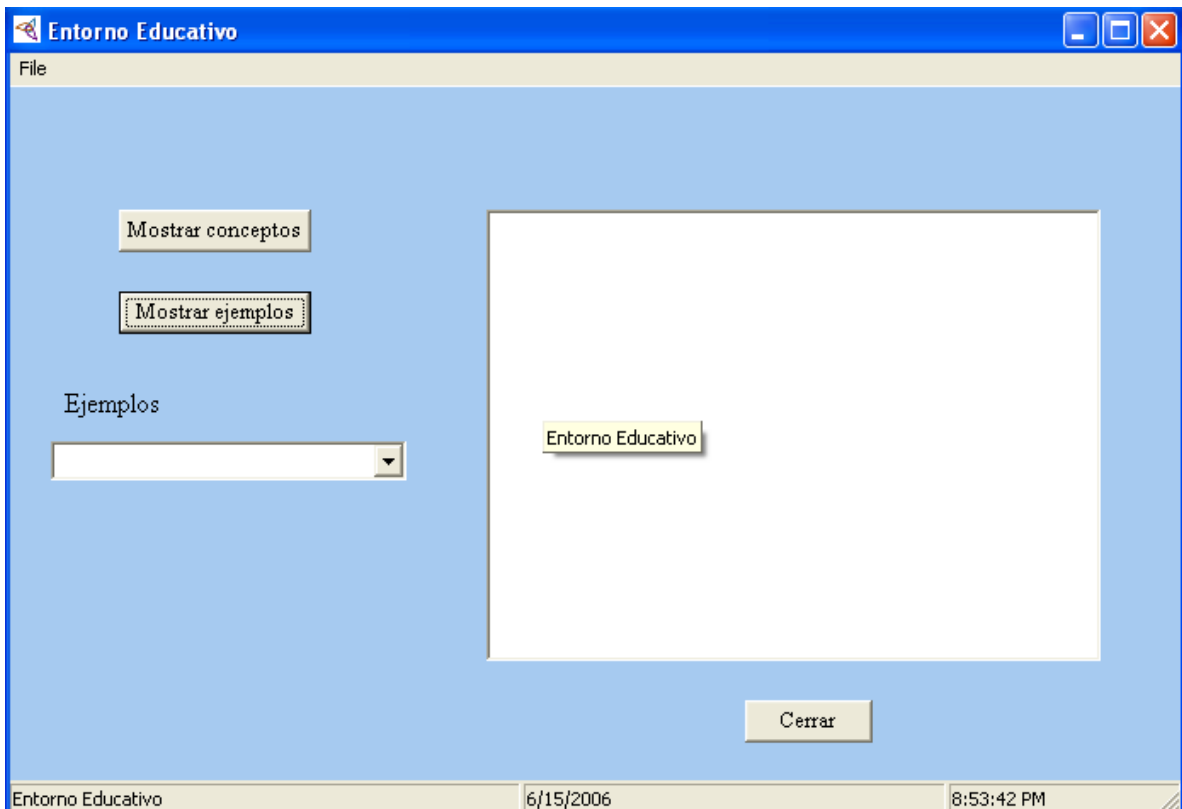


Figura 3.21 Ventana para mostrar ejemplos.

3.7. Conclusiones del capítulo.

En este capítulo se hizo referencia al manual de usuario del Protégé, describiendo sus clases, conceptos, instancias y consultas. También se describió la consulta realizada en Protégé y una aplicación que se realizó para mostrar los resultados de dicha consulta.

Conclusiones:

- En este trabajo se realizó el estudio de las Ontologías de forma general y particular, y de los principales resultados asociados con los dominios docentes y se pusieron de manifiesto varios aspectos de las ontologías: definiciones, componentes, metodologías para su desarrollo, tipos, lenguajes, etc.
- Las herramientas disponibles para la edición y mantenimiento de ontologías juegan un papel preponderante. La elección de la herramienta adecuada depende de los objetivos que se persigan. Se revisaron las características de herramientas profesionales actuales, específicamente el Protégé (versión 3.1), que posee características de versatilidad, flexibilidad y sencillez por lo que la convierte en una herramienta potencialmente útil para cualquier necesidad.
- Se formuló un modelo general para representar el dominio de los conceptos de una disciplina, especificando las entidades, relaciones y atributos de cada uno de los dominios.
- Se realizó un pequeño entorno educativo para instrumentar el modelo propuesto, validándolo con una asignatura de la carrera, específicamente Base de Datos.

Recomendaciones:

Dada la importancia de la problemática a partir de la cual surgió el presente trabajo, se recomienda:

- Desarrollar una interfaz en un superlenguaje que facilite la creación de lecciones mediante diferentes plantillas.
- Crear y organizar metodológicamente un conjunto de lecciones que permita validar el sistema en la práctica de asignaturas con diferentes características.
- Desarrollar un modelo de enseñanza basado en ontologías que se ajuste más a las condiciones de la educación cubana y en particular, de la enseñanza de nivel superior.

Bibliografía:

1. Abián, M. (2005) “ONTOLOGÍAS: QUÉ SON Y PARA QUÉ SIRVEN” en *Web semántica hoy*. [En línea]. Diciembre disponible en: <http://www.wshoy.sidar.org/index.php> [Accesado el día 2 de febrero del 2006]
 2. Morales, E. (1999) “16.3 Tipos de Ontologías” en *Representación de Conocimiento*. [En línea]. Junio disponible en: <http://www.mor.itesm.mx/~rdec/node197.html> [Accesado el día 2 de febrero del 2006]
 3. <[http:// Tesauro.htm](http://Tesauro.htm)>
 4. <<http://Tesauros y ontologias.htm>>
 5. <<http://cmap.coginst.uwf.edu/info/>>
 6. Marzal M, A.(2005) “INSTRUMENTOS DE UTILIDAD EN EL ANÁLISIS DEL CONTENIDO PARA LA ORGANIZACIÓN DE DOCUMENTOS DIGITALES EDUCATIVOS.” Congreso en Universidad Carlos III de Madrid, España.
- (Andriessen & Sandberg, 1999). Andriessen, J. and Sandberg, J. (1999). *Where is education and how about AI?* International Journal of Artificial Intelligence in Education, 10.
- (Borst, 1997). Borst W.N.(1997). *Construction of Engineering Ontologies*, PhD Thesis, University of Twente, Enschede, NL-Centre for Telematica and Information Technology.
- Ding y Foo (2002). Ding, Y.; Foo, S. (2002). *Ontology research and development*. Part 1- a review of ontology generation. Journal of Information Science, vol. 28, nº 2, p. 123-136.

(Genesereth and Nilsson, 1987). Genesereth and Nilsson (1987) *Logical Foundations of Artificial Intelligence*. Genesereth, M. R. and Nilsson, N., editors (1987). Morgan Kaufmann, San Mateo, CA.

(Gruber, 1993). Gruber, T. R.(1993). *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 5 199-200.

Gruber(1993 a). Gruber, T. R. (1993 a). “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”. Technical Report KSL-93-04, Knowledge Systems Laboratory, Stanford University, CA. Documento disponible en la red.
ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-93-04.ps.gz

Guarino (1998). Guarino, N. (1998). *Formal Ontologies and Information Systems*, en N. Guarino (ed.): 3-15.

(Mahesh, 1996). Mahesh, K. (1996) *Ontology Development for Machine Translation: Ideology and Methodology*. NMSU. Computing Research Laboratory. Technical Report MCCS-965-292. New Mexico.

(Marcke, 1992). Marcke, K. V. (1992). *Instructional expertise*. In Frasson, C., Gauthier, G., and McCalla, G., editors, Proceedings of the Second Conference of Intelligent Tutoring Systems, Montréal. Springer Verlag, Berlin.

Miguel Rodríguez (2000). Rodríguez M. (2000) “UNA ARQUITECTURA COGNITIVA PARA EL DISEÑO DE ENTORNOS TELEMÁTICOS DE ENSEÑANZA Y APRENDIZAJE”. Documento disponible en la red: <http://sensei.ieec.uned.es/~miguel/tesis/master-tesis.html>

(Murray, 1998). Murray, T. (1998). Authoring knowledge based tutors: *Tools for content, instructional strategy, student model, and interface design*. Journal of Learning Sciences, 7(1):5-64.

(Neches, 1991). Neches R.(1991). *Enabling technology for knowledge sharing*. AI Magazine;12(3):36-56.

Qin y Paling (2000-01). Qin, J.; Paling, S. (2000-01). “Converting a controlled vocabulary into an ontology: the case of GEM”. Information Research, vol. 6, nº 2. Disponible en: <http://informationr.net/ir/6-2/paper94.html>

Sowa (2000). Sowa, J.F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co., Pacific Grove, CA.

Steve et al. (1998). Steve, G, A. Gangemi, D. Pisanelli (1998). “Integrating Medical Terminologies with ONIONS Methodology”. Documento disponible en la red. <http://saussure.irmkant.rm.cnr.it>.

(Uschold-Gruninger, 1996). Uschold M., Gruninger M. Ontologies(1996) *Principles, Methods and Applications*. Knowledge Engineering Review 11(2)

(Uschold-Jasper, 1999). Uschold M. y Jasper R.(1999) *A Framework for Understanding and Classifying Ontology Applications*. Proceedings of the IJCAI’99. Workshop on Ontologies and PSMs. Stockholm. Sweden.

(Weigand, 1997). Weigand, H. (1997). *Multilingual Ontology-Based Lexicon for News Filtering –The TREVI Project*, 138-159.