



UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO  
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS



LICENCIATURA EN SISTEMAS DE INFORMACIÓN

**TRABAJO FINAL DE GRADUACIÓN**

**ONTOLOGÍA DEL PERFIL DE USUARIO  
PARA PERSONALIZACIÓN DE  
SISTEMAS DE U-LEARNING  
UNIVERSITARIOS**

**Autor(es):**

GABRIELA GONZÁLEZ

**Profesor guía:**

DRA. ELENA B. DURÁN

Marzo de 2014

UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO  
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS

**TRABAJO FINAL DE GRADUACIÓN DE LA LICENCIATURA EN SISTEMAS DE  
INFORMACIÓN**

**ONTOLOGÍA DEL PERFIL DE USUARIO PARA  
PERSONALIZACIÓN DE SISTEMAS DE U-LEARNING  
UNIVERSITARIOS**

**Autor(es):**

.....  
Gabriela González

**Profesor Guía:**

.....  
Dra. Elena B. Durán

\* \_\_\_\_\_ \*

Aprobado el día ..... del mes de ..... del año 20.....

por el Tribunal integrado por

.....

.....

∞ *Santiago del Estero – Argentina* ∞

*A mis padres,  
por todo lo que significan para mí.*

## *Agradecimientos*

*A mis hermanas, por cuidarme siempre.*

*A mis amigas, por su amistad y cariño.*

*A la Dra. Rosanna Costaguta y a la Dra. Elena Durán por haberme dado la posibilidad de ser parte de su equipo cátedra durante el cursado de la carrera y por representar un ejemplo de trabajo y dedicación.*

*Una vez más, a la Dra. Elena Durán, por la orientación y ayuda aportada para la realización de este trabajo y por seguir brindándome oportunidades de desarrollo, tanto profesional como personal.*

G. G.  
*Santiago del Estero, Argentina*  
Marzo de 2014

# CONTENIDO

---

<b>RESUMEN</b> .....	viii
<b>INTRODUCCIÓN</b> .....	ix
<b>i. TRABAJOS RELACIONADOS</b> .....	xi
<b>ii. JUSTIFICACIÓN</b> .....	xiii
<b>iii. CONTEXTO</b> .....	xiv
<b>iv. OBJETIVOS</b> .....	xv
<b>v. ESTRUCTURA DEL TRABAJO</b> .....	xvi
<b>CAPÍTULO I. MARCOS REFERENCIALES</b> .....	1
<b>I.1. MARCO TEÓRICO</b> .....	1
<b>I.1.1. PERSONALIZACIÓN EN SISTEMAS DE E-LEARNING Y U-LEARNING</b> .....	1
<b>I.1.1.1. Sistemas adaptativos</b> .....	2
<b>I.1.1.2. Modelo del estudiante</b> .....	3
<b>I.1.1.3. Sistemas de aprendizaje ubicuo</b> .....	9
<b>I.1.1.4. Modelo de contexto</b> .....	11
<b>I.1.1.5. Clases de adaptación</b> .....	14
<b>I.1.2. ONTOLOGÍAS</b> .....	16
<b>I.1.2.1. Clasificación y aplicaciones</b> .....	18
<b>I.1.2.2. Componentes</b> .....	19
<b>I.1.2.3. Paradigmas de representación del conocimiento, lenguajes y herramientas</b> ....	20
<b>I.1.2.4. Modelado de ontologías con OWL-DL</b> .....	22
<b>I.1.2.4.1. Propiedades</b> .....	23
<b>I.1.2.4.2. Clases</b> .....	25
<b>I.1.2.4.3. Individuos</b> .....	27
<b>I.2. MARCO METODOLÓGICO</b> .....	27
<b>I.2.1. METODOLOGÍA DE DESARROLLO DE ONTOLOGÍAS</b> .....	27

I.2.1.1. Ontology 101.....	28
I.2.1.2. METHONTOLOGY.....	28
I.2.1.3. Metodología resultante.....	31
I.2.2. METODOLOGÍA DE EVALUACIÓN DE ONTOLOGÍAS.....	34
<b>CAPÍTULO II. DESARROLLO DE LA ONTOLOGÍA DEL ESTUDIANTE.....</b>	<b>38</b>
<b>II.1. DIMENSIONES DEL ESTUDIANTE PARA PERSONALIZACIÓN EN SISTEMAS DE U- LEARNING.....</b>	<b>38</b>
II.1.1. INFORMACIÓN DEL ESTUDIANTE.....	39
II.1.1.1. Categorías independientes del dominio.....	40
II.1.1.2. Categorías dependientes del dominio.....	41
II.1.2. INFORMACIÓN DE CONTEXTO.....	42
<b>II.2. MODELO CONCEPTUAL DE LA ONTOLOGÍA.....</b>	<b>43</b>
II.2.1. DETERMINACIÓN DEL DOMINIO Y ALCANCE DE LA ONTOLOGÍA.....	43
II.2.2. ANÁLISIS DE REUSO DE ONTOLOGÍAS EXISTENTES.....	45
II.2.3. DEFINICIÓN DE CLASES Y SU JERARQUÍA.....	46
II.2.4. DEFINICIÓN DE PROPIEDADES.....	47
II.2.5. DEFINICIÓN DE LAS CARACTERÍSTICAS DE LAS PROPIEDADES.....	49
II.2.6. DEFINICIÓN DE AXIOMAS FORMALES Y DESCRIPCIÓN DE CLASES.....	51
<b>II.3. IMPLEMENTACIÓN CON PROTÉGÉ.....</b>	<b>53</b>
II.3.1. CREACIÓN DE LA ONTOLOGÍA.....	54
II.3.2. DEFINICIÓN DE CLASES.....	54
II.3.3. DEFINICIÓN DE PROPIEDADES DE OBJETOS Y DE DATOS.....	58
II.3.4. DEFINICIÓN DE AXIOMAS FORMALES Y DESCRIPCIÓN DE CLASES.....	66
<b>CAPÍTULO III. EVALUACIÓN DE LA ONTOLOGÍA DEL ESTUDIANTE.....</b>	<b>70</b>
<b>III.1. EVALUACIÓN DE ERRORES TAXONÓMICOS Y ANOMALÍAS DE DISEÑO.....</b>	<b>70</b>
III.1.1. ERRORES DE INCONSISTENCIA.....	70
III.1.2. ERRORES DE INCOMPLETITUD.....	74
III.1.3. ERRORES DE REDUNDANCIA.....	76
<b>III.2. EVALUACIÓN CONTRA REQUERIMIENTOS.....</b>	<b>77</b>
III.2.1. REQUERIMIENTOS DE DATOS.....	77
III.2.2. REQUERIMIENTOS DE RELACIONES ENTRE DATOS.....	97
<b>III.3. CASO DE ESTUDIO: APLICACIÓN UBICUA PARA LA ENSEÑANZA UNIVERSITARIA.....</b>	<b>110</b>

<b>III.3.1. DOMINIO DE LA ONTOLOGÍA.....</b>	<b>110</b>
<b>III.3.2. MODELO EXTENDIDO DE LA ONTOLOGÍA E IMPLEMENTACIÓN.....</b>	<b>111</b>
<b>III.3.3. EVALUACIÓN DE LA ONTOLOGÍA CONTRA LAS PREGUNTAS DE COMPETENCIA.....</b>	<b>115</b>
<b>CONCLUSIONES.....</b>	<b>120</b>
<b>BIBLIOGRAFÍA.....</b>	<b>122</b>

# RESUMEN

---

El desarrollo de la tecnología en comunicaciones inalámbricas ha impactado en las actividades cotidianas, especialmente en las formas de acceso al conocimiento, dando lugar al surgimiento del *aprendizaje ubicuo (u-learning)*, que designa al conjunto de actividades formativas, apoyadas en tecnología, que están accesibles en cualquier lugar y desde cualquier dispositivo.

Los sistemas que dan soporte a esta nueva forma de aprendizaje, requieren contemplar factores tales como: la capacidad de cómputo, de comunicaciones inalámbricas y de interacción; además de las características personales del usuario, su nivel de conocimiento, preferencias y objetivos, para que contribuyan a un aprendizaje efectivo. Estas características se representan a través de un modelo, denominado modelo de usuario que le permite al sistema la capacidad de distinguir entre distintos alumnos y actuar en consecuencia. Esto quiere decir, que dos alumnos con diferentes características serán tratados de manera distinta por el mismo sistema.

Expresar este modelo como una ontología, permite estandarizar el vocabulario y facilitar la comunicación, favoreciendo la reusabilidad del conocimiento. En el contexto de los sistemas educativos estas características tienen especial importancia porque, por lo general, un sistema de soporte a la enseñanza de cualquier tipo estará compuesto por distintos modelos que deben comunicarse entre sí.

En el presente trabajo se presenta la ontología del perfil de usuario (estudiante) desarrollada para sistemas de u-learning universitarios. Para la consecución de la misma fue necesario, previamente, seleccionar las características de los estudiantes y del entorno que resultan relevantes para llevar a cabo las tareas de personalización en dichos sistemas. Una vez construida la ontología, la misma fue evaluada para verificar su calidad. Esto permitió identificar las ventajas y limitaciones de la utilización de ontologías para crear el modelo requerido.

## **Palabras clave:**

*U-learning, Modelo de Usuario, Ontología, Personalización, Contexto.*

# INTRODUCCIÓN

---

La inclusión de las tecnologías de computación para dar soporte al aprendizaje comenzó en el año 1920 con la creación del primer sistema de ejercitación automático (Heines, 1988). Más tarde, en la década del 60, la investigación en esta área comenzó a cobrar mayor protagonismo lo cual propició la creación de numerosos sistemas de instrucción asistidos por computadora, que se implementaron por primera vez en la universidad de Stanford.

En los años 90, con el surgimiento de la world wide web y con la adopción del término e-learning, se comenzaron a crear sistemas de soporte al aprendizaje accesibles a través de Internet. En cierta forma, esto supuso un retroceso puesto que la gran mayoría de los mismos se limitaba a proveer los recursos a través de Internet, dejando de lado el concepto de personalización exhibido por los demás sistemas creados hasta el momento.

No obstante, los sistemas de e-learning evolucionaron con el transcurso del tiempo, comenzando con la incorporación de técnicas de los sistemas adaptativos a su dominio particular, en un intento de proveer una experiencia de aprendizaje personalizada.

Recientemente, luego de la extensión del aprendizaje a los dispositivos móviles y el surgimiento del m-learning, entra en escena un nuevo modo de aprendizaje denominado aprendizaje ubicuo o u-learning.

Éste se desprende del concepto de computación ubicua y va un paso más allá del aprendizaje móvil al incorporar consciencia de contexto.

Con la integración de este nuevo elemento, el u-learning, promete la creación de una experiencia de aprendizaje centrada en el usuario, independiente del tiempo y el lugar, y capaz de proporcionar la información correcta en el momento adecuado y de la manera correcta en función del contexto del estudiante.

Para lograrlo, se vale de los modelos de usuario y de contexto, conceptos tomados de los sistemas adaptativos y la computación ubicua, respectivamente.

El modelo de usuario constituye una representación del mismo, en un modo procesable computacionalmente, como un conjunto de sus características relevantes, tales

como son entendidas por el sistema. En el ámbito educativo, este modelo se denomina modelo del estudiante y puede contener información sobre conocimientos previos, comportamiento, objetivos, estilos de aprendizaje, características personales, intereses y motivación, etc (Fröschl, 2005).

El modelo de contexto agrupa la información relativa a los aspectos que caracterizan la situación en la cual se está llevando a cabo la actividad de aprendizaje, en tanto los mismos resultan relevantes en términos de la adaptación a realizar. Un modelo de contexto en un sistema de aprendizaje ubicuo puede contener información sobre ubicaciones, dispositivos, actividades, etc (Yau & Joy, 2006).

Al contar con información de cada estudiante y su contexto, el sistema es capaz de distinguir entre distintos alumnos y actuar en consecuencia. Esto quiere decir, que dos alumnos con diferentes características serán tratados de manera diferente por el mismo sistema.

Para representar los modelos de usuario y de contexto, se pueden utilizar distintas técnicas, siendo las ontologías las que se distinguen de las restantes porque representan un entendimiento común del conocimiento que modelan, posibilitando con ello, el reuso e intercambio de información entre distintos sistemas.

En el contexto de los sistemas educativos estas características tienen especial importancia porque, por lo general, un sistema de soporte a la enseñanza de cualquier tipo estará compuesto por distintos modelos que deben comunicarse entre sí. Si todos los modelos utilizan una clase de representación común y estándar se favorece la interoperabilidad entre los mismos, así como también con otros sistemas de la misma clase. Esto sería útil por ejemplo, para utilizar el mismo modelo del estudiante en sistemas de distinto dominio.

Por todo lo expuesto anteriormente, queda claro que para brindar una experiencia de aprendizaje personalizada en un entorno de aprendizaje ubicuo, es necesario contar con un modelo de usuario que incorpore también las características referidas al contexto del estudiante.

Con esta motivación en mente, en el desarrollo del presente trabajo se intentará responder el siguiente interrogante:

*¿Cuál es la ontología más apropiada para representar las características del estudiante en un entorno ubicuo, con fines de personalización?*

## i. TRABAJOS RELACIONADOS

A continuación se describen los trabajos que reflejan investigaciones realizadas en el área de conocimiento de la presente investigación.

En relación al uso de ontologías para modelado de usuarios en sistemas ubicuos se encontraron las siguientes propuestas:

Bobillo et al. (2006) formula un modelo para la provisión de resúmenes adaptados al contexto del usuario ubicuo, aplicado a un Sistema Basado en Conocimiento Móvil. Éste utiliza un modelo de restricción del contexto, que posee dos ontologías, una para codificar el conocimiento acerca del dominio de la aplicación en cuestión y otra para el contexto en que éste puede resultar útil; y utilizando una tercera ontología que relaciona las dos anteriores, establece bajo qué circunstancias conviene recuperar ciertos subconjuntos de información.

Hervás et al. (2006) presenta un modelo de usuario consciente del contexto, representado a través de una ontología modular y extensible que diferencia tres tipos de información: características del usuario, servicios generales y servicios específicos. Se hace hincapié en el tratamiento de la ambigüedad de la información de contexto, utilizando mecanismos que convierten las acciones del usuario en fuente de información.

Heckmann et al. (2006) propone GUMO, una ontología general para el modelado del usuario. GUMO (General User Model Ontology) provee una colección de dimensiones del usuario, incluyendo información sobre el contexto, lo que le permite también ser utilizada en Sistemas de Apoyo al Aprendizaje Ubicuo.

En relación al uso de ontologías para representar el contexto en entornos ubicuos, se localizaron los siguientes trabajos:

Chen *et al.* (2003) introduce COBRA-ONT, un conjunto de ontologías OWL para sistemas conscientes del contexto. COBRA-ONT está formada por cuatro ontologías, cada una de las cuales representa un concepto general del contexto: Lugar, Agente, Ubicación del agente y Actividad del agente. El objetivo de COBRA-ONT es proponer un vocabulario común que pueda ser usado para superar el obstáculo de los modelos de contexto con información propietaria, y es el componente principal de CoBrA, una arquitectura centrada en agentes para ambientes inteligentes.

Chen *et al.*, (2004) define SOUPA (Ontología Estándar para Aplicaciones Ubicuas y Pervasivas), una ontología reusable codificada en OWL que permite modelar aplicaciones computacionales ubicuas. SOUPA está definida de forma modular, y se compone de dos

conjuntos de ontologías: el núcleo y las extensiones. Las ontologías que conforman el núcleo intentan definir el vocabulario genérico usado por las diferentes aplicaciones ubicuas, mientras que las ontologías de extensión definen vocabulario adicional para soportar tipos específicos de aplicaciones.

Wang *et al.* (2004) presenta CONON, una ontología de contexto para ser utilizada en ambientes pervasivos de computación. CONON está implementada en OWL y consiste de una ontología de nivel superior que es extendida a través de diferentes ontologías de dominio específicas del ámbito de los ambientes inteligentes. De este modo, la ontología base contiene los conceptos genéricos del contexto, y a esta se le agregan las ontologías de dominio particulares necesarias para representar el vocabulario específico de la aplicación

En relación al modelado del usuario y/o contexto con ontologías en ambientes de e/u-learning se dispone de las siguientes referencias:

Chen & Mizoguchi (1999) describen una ontología del estudiante formada por conceptos independientes del dominio. Para construirla realizaron una abstracción de los conceptos independientes del dominio presentes en el intercambio de información realizado entre los agentes del modelo del estudiante y los restantes, en una arquitectura multi-agente de un sistema educacional inteligente. El objetivo del modelo es proveer un estándar para la comunicación entre agentes en esta clase de sistemas.

Šimún *et al.* (2007) describen un conjunto de modelos, representados mediante ontologías OWL, que forman parte de un portal educativo con capacidades de personalización. El modelo del estudiante se divide en dos partes: la parte general o independiente del dominio, que puede reusarse en distintos sistemas, y la parte específica del dominio, y se comunica con los otros dos modelos, de dominio y de adaptación, para llevar a cabo la recomendación de documentos educativos al alumno.

Hong & Cho (2008) presentan CALA, una arquitectura para ambientes de aprendizaje ubicuo capaz de proveer diferentes servicios en función del contexto. El componente central de la misma es el Módulo de Administración Sensible al Contexto, que se encarga de mapear cada nuevo contexto en un lugar semántico. Éste se representa a través de CALA-ONT, una ontología de contexto implementada con XML, RDF y OWL.

Siadaty *et al.*, (2008) introducen m-LOCO, un framework basado en ontologías cuyo propósito es la captura de información de contexto en ambientes de aprendizaje móvil. Utiliza ontologías para representar los estilos de aprendizaje, el dominio, los usuarios y los objetos de aprendizaje, proporcionando soporte para los distintos dispositivos móviles y un enfoque de aprendizaje auto-regulado con soporte para la colaboración.

Pramitasari *et al.* (2009) exponen una ontología del modelo del estudiante para personalización en sistemas de e-Learning. Los componentes centrales de la misma son el rendimiento del estudiante y el estilo de aprendizaje del mismo, de acuerdo al modelo de Felder-Silverman. La ontología está codificada en OWL y se implementó en un portal semántico usando la herramienta PortalCore.

Panagiotopoulos *et al.* (2012) proponen una ontología del modelo del estudiante para sistemas tutoriales inteligentes de educación a distancia. El modelo representa las características consideradas relevantes para el proceso de adaptación, de un estudiante adulto. Las mismas son una combinación de las categorías de información propuestas por dos estándares relacionados con el modelado del estudiante y los resultados de un estudio empírico sobre un grupo de alumnos de la Universidad Helénica Abierta.

De la revisión de trabajos realizada se puede concluir que las ontologías presentadas modelan aspectos semánticos de las características del estudiante o de las características del contexto. En el presente trabajo se busca crear una ontología en donde se combinen las características personales de los estudiantes con las del contexto de los entornos de aprendizaje ubicuo, con el fin de personalizar la experiencia de aprendizaje del alumno.

## **ii. JUSTIFICACIÓN**

En los sistemas de u-learning, la precisión en el modelado de las características del usuario define, en gran medida, la efectividad total del sistema. Una interpretación incorrecta de las necesidades del usuario conlleva a decisiones adaptativas erróneas, que pueden resultar en frustración, desconfianza, disminución de motivación para usar el sistema, etc, por parte del estudiante. Es por esto que, la representación adecuada del conocimiento acerca del usuario, la elicitación efectiva de información relacionada al mismo, y la utilización de ésta para proporcionar una adaptación coherente y con significado son factores cruciales para el éxito de un sistema de u-learning adaptativo.

En relación a la representación adecuada del conocimiento del usuario, la utilización de ontologías resulta ventajosa puesto que éstas permiten la creación de modelos de dominio de alta calidad, y en el caso de los modelos de usuario, lo que se intenta representar es el estado interno de un usuario humano.

Utilizar técnicas ontológicas para modelar el usuario posee tres ventajas principales con respecto a los métodos tradicionales:

- Facilitan la gestión de la adaptación por medio de agentes de software.

- Soportan razonamiento, a través de sus relaciones, condiciones y restricciones, permitiendo la inferencia de nuevo conocimiento;
- Favorecen el reuso del modelo ya que proporcionan una conceptualización compartida de un dominio de información.

Teniendo en cuenta que la idea central que motiva el desarrollo de técnicas de adaptación se basa en la asunción de que las diferencias en ciertas características de los usuarios afectan la utilidad de los servicios de información que se les provee, se puede deducir que si el comportamiento de un sistema puede acondicionarse a tales características, los servicios de información provistos poseerán un valor mucho mayor para los individuos.

De este modo, se pueden aprovechar las características inherentes de las ontologías para realizar una representación más acertada del perfil de usuario, y lograr así una mejor personalización, contribuyendo de esta forma al aprendizaje del alumno.

### iii. CONTEXTO

El presente trabajo se realizó en el marco del Proyecto de Investigación CICYT-UNSE 23/C095 “Sistemas de Información Web Personalizados, Basados en Ontologías, para Soporte al Aprendizaje Ubicuo” correspondiente a la convocatoria 2012-2015 de la Secretaría de Ciencia y Tecnología de la Universidad Nacional de Santiago del Estero.

La hipótesis de trabajo de dicho proyecto es la siguiente:

*“Es posible mejorar la interoperabilidad y usabilidad de los sistemas de información web de apoyo al aprendizaje ubicuo en contextos universitarios a partir de la incorporación de ontologías, y la aplicación de técnicas de personalización.”*

Y sus objetivos específicos son:

- Diseñar modelos de entorno de aprendizaje ubicuo ajustados al contexto de la educación universitaria, a partir de un enfoque sistémico.
- Analizar, diseñar, construir, evaluar y/o reusar ontologías para sistemas de información web de apoyo al aprendizaje ubicuo en contextos universitarios.
- Diseñar, construir y evaluar modelos de usuario basados en ontologías para personalizar sistemas de información web de apoyo al aprendizaje ubicuo en contextos universitarios.

- Diseñar, construir y evaluar módulos de personalización de contenidos, navegación e interfaces de usuario, basados en tecnología de agentes, para sistemas de información web de apoyo al aprendizaje ubicuo en contextos universitarios.
- Integrar los módulos de personalización de contenidos, navegación e interfaces de usuario en sistemas de información web de apoyo al aprendizaje ubicuo en contextos universitarios.
- Evaluar la interoperabilidad y usabilidad de los sistemas de información web personalizados, de apoyo al aprendizaje ubicuo en contextos universitarios.

Los distintos modelos a desarrollarse como parte de este proyecto serán representados a través de ontologías, las cuales se integrarán para proveer los servicios de personalización requeridos.

Por lo tanto, la utilización de ontologías para la representación del modelo del usuario es un requisito *a priori* en la presente investigación.

Esto no resta validez a las ventajas, mencionadas anteriormente, que se obtienen al hacer uso de las mismas para la construcción de esta clase de modelos, independientemente de si formarán parte de un sistema donde los demás modelos también se representen a través de ontologías.

#### **iv. OBJETIVOS**

Teniendo en cuenta el proyecto de investigación en el cual se encuentra inserto el presente trabajo, se definen los siguientes objetivos generales:

- Contribuir a la personalización de sistemas de u-learning para contextos universitarios.
- Contribuir a mejorar la interoperabilidad y usabilidad de los sistemas de u-learning en el nivel universitario, aportando una ontología del perfil de usuario para estos sistemas.

Los objetivos específicos son:

- Identificar las características de los usuarios de sistemas de u-learning que resulten relevantes para favorecer la personalización de estos sistemas.
- Diseñar una ontología para el perfil del usuario de sistemas de u-learning en el contexto universitario, que permita representar sus preferencias y datos relevantes.

## **v. ESTRUCTURA DEL TRABAJO**

El presente trabajo de investigación se encuentra dividido en tres capítulos. En el primer capítulo se presentan los marcos referenciales en los que se fundamenta el trabajo. En el segundo capítulo se detalla el modelo propuesto, junto con el proceso de desarrollo seguido para obtenerlo, y en el tercer capítulo se describe la evaluación realizada sobre el mismo. Finalmente, se presentan las conclusiones del trabajo en donde se incluyen las líneas de investigación futuras derivadas de éste.

# CAPÍTULO I

## MARCOS REFERENCIALES

---

En el presente capítulo se introducen los fundamentos teóricos en los que se basa la investigación realizada.

En la primera sección se describe el marco teórico, que trata, principalmente, sobre personalización en sistemas de aprendizaje soportados por computadora y sobre ontologías. La segunda sección detalla el marco metodológico, donde se exponen las metodologías de desarrollo y evaluación de ontologías utilizadas.

### I.1. MARCO TEÓRICO

#### I.1.1. PERSONALIZACIÓN EN SISTEMAS DE E-LEARNING Y U-LEARNING

La introducción de tecnologías de computación para dar soporte al aprendizaje comenzó en el año 1920 cuando el profesor de psicología Sidney Pressey construyó y programó “The Pressey Testing Machine”. Este dispositivo, similar a una máquina de escribir, presentaba a los alumnos preguntas y respuestas en formato de opción múltiple que los mismos debían responder, con la particularidad de que, para que una pregunta se eliminará de la lista de preguntas posibles, el estudiante debía contestarla correctamente al menos 2 veces (Heines, 1988).

El invento de Pressey sentó las bases para lo que años más tarde serían los sistemas de instrucción asistidos por computadora, implementados por primera vez en la universidad de Standford en la década del 60.

Décadas más tarde, en el año 1999, durante un Seminario de Sistemas CBT (Entrenamiento Basado en Computadoras) en Los Ángeles, se utilizó por primera vez el término e-learning para describir *“una forma de aprender basada en el uso de nuevas*

tecnologías permitiendo el acceso a entrenamiento online, interactivo e, incluso, personalizado a través de Internet u otro tipo de medios electrónicos.” (Cross, 2004).

A partir de allí, el avance en las tecnologías de computación, la extensión de la www, y las mejoras en las redes de comunicaciones, propiciaron el desarrollo y la evolución del e-learning en otras formas de aprendizaje, con nuevas características y desafíos.

### I.1.1.1. SISTEMAS ADAPTATIVOS

Uno de los principales desafíos que el e-learning trató de satisfacer es el de ofrecer una experiencia de aprendizaje personalizada para cada alumno, intentando imitar el proceso de enseñanza que se manifiesta entre docentes y alumnos en una clase presencial. Para lograr este objetivo, aplicó técnicas del campo de los sistemas adaptativos a su dominio particular.

Un sistema adaptativo es aquel que es capaz de adaptarse a distintas circunstancias. Este proceso de adaptación está basado en los objetivos y preferencias del usuario, los cuales se almacenan en un modelo de usuario. El modelo de usuario es mantenido por el sistema y provee información acerca de cada usuario particular, es decir, posibilita la distinción entre diferentes usuarios. Esto le confiere al sistema la habilidad de adecuar su comportamiento en función de los valores contenidos en el modelo para cada usuario particular (Fröschl, 2005).

Koch (2001) define al modelo de usuario como aquel “[...] constituido por descripciones de lo que se considera relevante acerca del conocimiento y/o aptitudes actuales del usuario, proveyendo información para que el entorno del sistema se adapte a cada usuario individual.”

Por su parte, Fröschl (2005) describe los tres roles que cumple un modelo de usuario en el proceso de adaptación. Los mismos se muestran en la figura I.1.

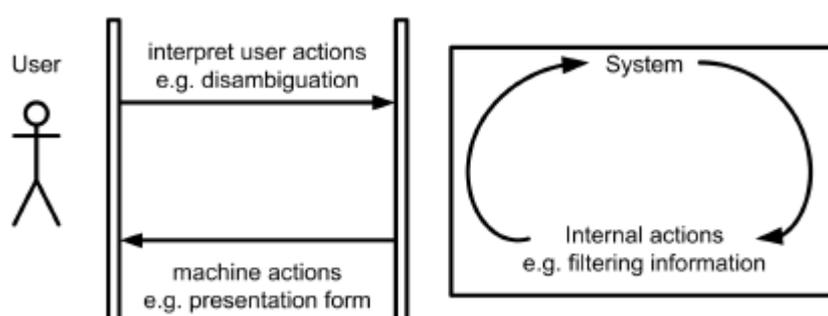


Figura I.1: Roles del modelo de usuario (Fröschl, 2005).

- **Interpretar las acciones del usuario:** auxiliando al sistema cuando la información que se recibe a través de la interfaz de usuario es ambigua y asistiendo en la interpretación de las acciones incorrectas del usuario.
- **Controlar las acciones del sistema:** en función de las preferencias de cada usuario particular. Esto se lleva a cabo a través de la adecuación de la respuesta y aspecto del sistema, así como también del contenido y su presentación.
- **Asistir en las acciones internas del sistema:** cooperando en las acciones transparentes al usuario que éste realiza para determinar su respuesta, como por ejemplo, el filtrado y refinación de la información recibida.

En el contexto del e-learning, los sistemas adaptativos se enfocan en la adaptación del contenido de aprendizaje y en la presentación del mismo. De acuerdo a Mödritscher *et al.* (2004) *“un sistema adaptativo se enfoca en como el material es aprendido por el alumno y presta atención a las actividades de aprendizaje, las estructuras cognitivas y el contexto del material de aprendizaje”*.

Según Fröschl (2005) un sistema adaptativo de e-learning es *“un sistema interactivo que personaliza y adapta el contenido de aprendizaje, los modelos pedagógicos, y las interacciones entre los participantes y el medio para alcanzar necesidades individuales y preferencias de los usuarios, cuando éstas se presentan.”*

Por lo tanto, los sistemas adaptativos de e-learning poseen todas las propiedades de los sistemas adaptativos generales, con la diferencia de que utilizan el modelo de usuario específicamente para adaptar el material de aprendizaje a las necesidades del estudiante. En esta clase de sistemas, el modelo de usuario se denomina modelo del estudiante.

#### **I.1.1.2. MODELO DEL ESTUDIANTE**

El modelo del estudiante representa las creencias del sistema con respecto al alumno y provee la información necesaria para adaptar la instrucción ofrecida de acuerdo a sus necesidades individuales (Fröschl, 2005).

La personalización involucra principalmente la elección y presentación de cada actividad sucesiva de enseñanza como una función del alcance total del conocimiento del estudiante sobre el tema que se le está enseñando y otras características relevantes del mismo, que son las que se mantienen en el modelo.

El proceso de personalización es un proceso complejo porque las características de los estudiantes que se deben tener en cuenta al momento de la adaptación, no son estáticas,

sino que por el contrario, cambian continuamente en el tiempo, pudiendo contener diferentes valores en una misma sesión de aprendizaje (Devedžic, 2006).

Además, no todos los modelos del estudiante contienen la misma información, algunos toman en cuenta sólo una característica del usuario (generalmente el conocimiento actual o su estilo de aprendizaje) mientras que otros modelan al alumno en base a múltiples dimensiones.

A continuación se presentan las categorías o dimensiones de información que forman parte del modelo del estudiante, según distintos autores y estándares.

De acuerdo a Fröschl (2005) la información del modelo del estudiante para sistemas adaptativos de e-learning se divide en dos categorías:

- **Información dependiente del dominio:** representa un reflejo del estado de conocimiento y aptitudes del alumno, referidos a un tema en particular.
  - **Otra información específica de dominio:** como conocimientos previos acerca del dominio, registros o actividades de aprendizaje realizadas (lecciones tomadas, número de pedidos de ayuda, tiempo utilizado para resolver un problema, etc.), registros de calificaciones y evaluaciones, etc.
- **Información independiente del dominio:** incluye objetivos de aprendizaje, aptitudes cognitivas, motivación, antecedentes y experiencia, preferencias, e información factual e histórica.
  - **Objetivos:** responden a las preguntas de porqué el estudiante usa el sistema y que es lo que quiere conseguir. Se dividen en dos tipos: objetivo de aprendizaje (relativamente estable durante cada unidad del curso) y objetivo de resolución de problemas (usualmente varía de un problema a otro, incluso dentro de una misma unidad temática).
  - **Aptitudes cognitivas:** son las habilidades intelectuales que posee el alumno, en diferentes áreas cognitivas. Por ejemplo, aptitud musical, aptitud matemática o de lectura.
  - **Motivación:** indica la dirección o impulso en la instrucción, y se mide usando diferentes parámetros de largo y corto plazo, como por ejemplo, nivel de esfuerzo, atención, distracción, persistencia, etc. Estos parámetros se conectan con otros factores como nivel de conocimiento, preparación, complejidad del tema y resultados de aprendizaje.

- **Antecedentes y experiencia:** se utiliza para derivar parámetros del modelo del estudiante, e incluye habilidades que pueden afectar el éxito del aprendizaje, como por ejemplo, profesión, experiencia de trabajo o perspectiva. Por experiencia, se entiende conocimiento en la utilización del entorno de aprendizaje.
- **Preferencias:** los estudiantes pueden tener diferentes preferencias con respecto a ciertos aspectos del entorno de aprendizaje. Estas preferencias son consideradas no inducibles por el sistema, por lo que el usuario debe informarlas al sistema, directa o indirectamente. Dos de estas preferencias son el estilo de aprendizaje y la inteligencia múltiple.
- **Información factual e histórica:** corresponde a la información demográfica, tal como: nombre, edad, padres, ID, etc. Esta información combinada con otra información factual, como por ejemplo, intereses, es necesaria para inicializar el modelo.

Sosnovsky (2007) propone seis dimensiones para modelar al usuario, con consideraciones especiales, para cada una de ellas, con respecto al modelo del estudiante.

- **Conocimiento, creencias, habilidades y antecedentes:** el conocimiento del estudiante es la característica principal para la adaptación, y se modela utilizando una estructura conceptual detallada que representa el dominio de aprendizaje. Las creencias son utilizadas por los sistemas que no diferencian de modo directo el conocimiento correcto del alumno de su conocimiento incorrecto; y en cambio, modelan al mismo como creencias. Las habilidades se refieren al conocimiento procedural del alumno, y el interés está en puesto en el modelado de habilidades meta-cognitivas. Los antecedentes son por lo general estáticos y no detallados. Se suelen representar como un único parámetro con distintos niveles o como un conjunto de estereotipos.
- **Intereses y preferencias:** los intereses pueden ser de corto plazo (válidos únicamente por sesión de trabajo) o de largo plazo (relativamente estables durante el tiempo de utilización total del sistema). Las preferencias suelen no estar relacionadas con la tarea principal del sistema, por ejemplo, preferencia sobre un tipo de interfaz o idioma particular.
- **Objetivos, planes, tareas y necesidades:** el objetivo se refiere a lo que el estudiante está tratando de lograr y el plan representa la secuencia de acciones que seguirá para conseguirlo. También se puede considerar como objetivo general del estudiante el

aprendizaje del material, y el plan representaría la mejor estrategia para conseguirlo de la forma más eficiente. Las tareas y necesidades son similares a los objetivos pero con una especificidad mayor.

- **Información demográfica:** desde características básicas como sexo, edad o idioma nativo, hasta características más complejas, como nivel de educación formal o ingresos familiares.
- **Estado emocional:** permite reconocer estados de desmotivación, aburrimiento o confusión. Esto sirve para, por ejemplo, adaptar el nivel de dificultad de los ejercicios u ofrecer otro tipo de tareas. Los esfuerzos realizados en esta área se enmarcan dentro de las investigaciones de la computación afectiva.
- **Contexto:** incluye información sobre la ubicación del usuario, tiempo, ambiente físico y social, dispositivo utilizado, etc.

En su propuesta de un modelo de estudiante para un sistema de aprendizaje ubicuo, Graf *et al.* (2009) presenta las siguientes características:

- **Perfil:** incluye sólo información estática, como nombre, sexo, ID, fecha de inicio de los estudios, título, programa de estudio e información de contacto.
- **Uso del sistema:** recopila información acerca del uso del sistema por parte de los alumnos, en términos de como se lo utiliza, cuando y para qué. Almacena el curso que está realizando el estudiante (y un historial de cursos pasados) y los servicios, hora y días preferidos para el uso del sistema.
- **Progreso:** en relación a los contenidos de aprendizaje revisados y las actividades pedagógicas realizadas, el porcentaje de los mismos en función del total del curso y la interacción del estudiante en el foro de discusión, incluyendo también los registros pasados.
- **Intereses y nivel de conocimiento:** de los conceptos del dominio de aprendizaje. Para cada concepto (tomados del modelo de dominio del sistema) se almacena su nivel de conocimiento e interés con una medida cualitativa.
- **Cercanía social:** incluye información acerca del nivel de familiaridad entre estudiantes, indicando si se conocen, si han trabajado juntos y si estarían dispuestos a trabajar juntos. También se almacena la preferencia general de colaboración de cada estudiante.

- **Habilidades de resolución de problemas:** incluyendo pensamiento crítico, razonamiento científico, motivación en resolución de problemas y resolución de problemas en general.
- **Estilos de aprendizaje:** en función del modelo de Felder-Silverman y distinguiendo entre preferencias en cuatro dimensiones: activo/reflexivo, sensitivo/intuitivo, visual/verbal y secuencial/global.
- **Ubicación:** ubicación actual y pasada, en forma de coordenadas GPS y descripción textual incluyendo el nombre y dirección postal del lugar.

Por su parte, Martins *et al.* (2008) describen la siguiente información del modelo del estudiante:

- **Información dependiente del dominio:** corresponde al modelo de dominio con una funcionalidad de 3 niveles:
  - **Nivel de tarea:** comprende los objetivos/competencias del dominio que el alumno debe dominar. Los objetivos intermedios pueden modificarse de acuerdo a la evolución del alumno durante el proceso de aprendizaje.
  - **Nivel lógico:** describe el conocimiento del estudiante y se actualiza constantemente durante el proceso de aprendizaje.
  - **Nivel físico:** que registra e infiere el perfil de conocimiento del usuario.
- **Información independiente del dominio:** se divide en:
  - **Modelo psicológico:** contiene la información de los aspectos cognitivos y afectivos del estudiante, como ser, estilos de aprendizaje, estilo cognitivo (afectivo, impulsivo, etc) y aspectos de personalidad (introvertido, extrovertido, etc.).
  - **Modelo genérico:** contiene la información relacionada con los intereses del alumno, el conocimiento común y los antecedentes o experiencia. Incluye también, planes y objetivos, preferencias, perfil académico (estudios tecnológicos, conocimiento de literatura, capacidades artísticas, etc.), tipo y edad.

Con respecto a los estándares relacionados con el contenido y/o representación del modelo del estudiante en sistemas de e-learning, se destacan dentro de la literatura dos propuestas: el estándar de Información Pública y Privada del Estudiante (Public and Private Information for Learners) más conocido como PAPI Learner, y el Paquete de Información del Estudiante (Learner Information Package) referido como LIP (Fröschl, 2005; Devedžic, 2006).

El estándar PAPI Learner fue desarrollado inicialmente por el comité de IEEE LTSC (Learning Technology Standardization Committee), para luego ser transferido al grupo ISO JTC 1/SC SC36 (Information Technology for Learning, Education and Training). La primera versión del estándar fue publicada en 1997, y el último borrador, número 8, fue presentado en noviembre del 2001.

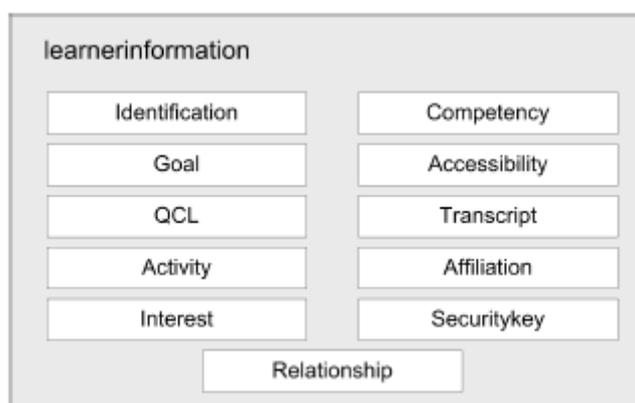
Desafortunadamente, las especificaciones del estándar no se encuentran disponibles online y el sitio web del Observatorio de Estándares de Tecnologías del Aprendizaje<sup>1</sup>, menciona que este estándar se encuentra obsoleto, razón por la cual sus especificaciones no se encuentran disponibles en la web.

Por otra parte, la especificación del Paquete de Información del Estudiante (Learner Information Package) (Smythe *et al.*, 2001) es desarrollada por el IMS Global Consortium<sup>2</sup>, siendo la última versión publicada del año 2005.

Esta especificación tiene por objetivo definir un conjunto de paquetes de información que puedan ser usados para importar y extraer datos de un servidor compatible con IMS, buscando así favorecer la interoperabilidad de los sistemas basados en internet que trabajan con información del estudiante con otros sistemas de aprendizaje que operen en la web.

El LIP representa un conjunto de información acerca de un estudiante (o un grupo de ellos), o bien de un productor de contenido de aprendizaje. Esta información se representa de un modo estructurado, definiendo los campos que deben completarse así como el tipo de datos que es posible utilizar para tal fin.

La información del estudiante se agrupa en once categorías, que han sido identificadas como las estructuras primarias de datos necesarias para dar soporte al contenido requerido. Las mismas se muestran en la figura I.2.



**Figura I.2:** Categorías de información del IMS LIP (Smythe *et al.*, 2001).

<sup>1</sup> <http://www.cen-ltso.net/>

<sup>2</sup> <http://www.imsglobal.org>

Las estructuras de datos o categorías base del IMS LIP son:

- **Identificación:** contiene los datos biográficos y demográficos que permiten identificar al estudiante, así como aquellos que resultan relevantes para el aprendizaje.
- **Objetivo:** comprende los objetivos personales (de aprendizaje, del curso en general) y otras ambiciones. Cada uno de ellos puede dividirse en sub-objetivos.
- **Títulos, certificados y licencias (qcl):** de logros realizados y otorgados por autoridades reconocidas. Para cada uno de ellos se puede indicar su nivel.
- **Actividad:** representa cualquier actividad relacionada con el aprendizaje, en cualquier estado de completitud. Incluye las categorías de educación formal e informal, entrenamiento, experiencia laboral y servicios cívicos o militares.
- **Expediente:** almacena un resumen de los logros académicos, en el formato de la institución particular donde se lleva a cabo el aprendizaje. Por ello, su estructura es variable y puede tomar diferentes formas.
- **Intereses:** representa los hobbies y actividades recreativas del alumno. Puede estar relacionada con la información de la categoría qcl.
- **Competencia:** destrezas, conocimiento y habilidades adquiridas en los dominios cognitivo, afectivo o psicomotriz. Posiblemente relacionadas con los qcl y las actividades.
- **Afiliación:** membresías de organizaciones profesionales y otras.
- **Accesibilidad:** idioma, discapacidades y preferencias de aprendizaje, incluyendo preferencias cognitivas (por ejemplo, estilo de aprendizaje), preferencias físicas (por ejemplo, un tipo especial de fuente o color) y preferencias tecnológicas (por ejemplo, una cierta plataforma o formato de archivo).
- **Seguridad:** contraseñas y claves de seguridad asignadas al alumno para realizar transacciones con otros sistemas y servicios de información.
- **Relaciones:** especifica los vínculos entre los componentes anteriores. Este módulo es necesario puesto que el modelo no incluye un medio para representar las relaciones entre las distintas categorías.

### I.1.1.3. SISTEMAS DE APRENDIZAJE UBICUO

Luego de unos años del auge del e-learning, y gracias a la disminución en el costo de los dispositivos móviles y a la expansión de las redes de comunicaciones, las

investigaciones dentro del área del aprendizaje adaptativo comenzaron a concentrarse en el aprendizaje móvil (m-learning) y, más recientemente, en el aprendizaje ubicuo (u-learning) (Hwang *et al.*, 2008).

El aprendizaje móvil permite a los estudiantes aprender en cualquier lugar y momento, haciendo uso de tecnologías móviles y conexiones de internet inalámbricas (Yau & Joy, 2006).

El aprendizaje ubicuo, desprendido del concepto de computación ubicua, va un paso más allá del m-learning, al incorporar consciencia de contexto.

La computación ubicua, tal como fue concebida por Weiser (1991), se refiere a dispositivos computacionales que “desaparecen” del entorno y que el usuario utiliza para llevar a cabo sus tareas cotidianas, sin siquiera percatarse de ello.

Esta visión inicial de Weiser está hoy aún lejos de realizarse, pero una definición práctica aceptada es la propuesta por Hwang *et al.* (2008) donde se define a la computación ubicua como *“una nueva tecnología que permite la utilización de grandes cantidades y tipos de objetos funcionales en cualquier lugar y tiempo a través de conexiones de red. Particularmente, se pueden utilizar objetos de comunicación inalámbricos con sensores embebidos que detectan información del usuario y entorno para la provisión de servicios personalizados.”*

Es decir que, un sistema equipado con tecnología de computación ubicua provee a los usuarios de información oportuna y de servicios relevantes en diferentes situaciones. Es capaz de generar resultados de forma inteligente gracias a su “consciencia” del contexto del usuario en cada momento.

Siguiendo esta línea de pensamiento, en el aprendizaje ubicuo los estudiantes pueden trasladarse con sus dispositivos móviles durante el proceso de aprendizaje, y éstos se encargan de dar soporte al proceso comunicándose con los objetos embebidos y otros dispositivos del entorno para proveer servicios personalizados en función del contexto (Graf & Kinshuk, 2008).

Las principales características de un sistema de aprendizaje ubicuo son (Hwang *et al.* 2008):

- **Consciencia de contexto:** el sistema puede percibir la situación del estudiante y la del entorno en el cual se encuentra inmerso el mismo.

- **SopORTE personalizado:** de la manera adecuada y en el momento y lugar correctos, basado en la información de contexto personal y ambiental, así como también en el perfil y portfolio de aprendizaje del usuario.
- **Aprendizaje invisible:** puesto que el estudiante puede moverse de un lugar a otro en el mundo real sin interrumpir su experiencia de aprendizaje.

La integración de las características mencionadas anteriormente se realiza de manera tal que un sistema de u-learning es capaz de proveer servicios de aprendizaje en función de las necesidades individuales de cada estudiante, en el momento oportuno, independientemente del lugar y de la forma correcta, siendo ésta lo más transparente posible.

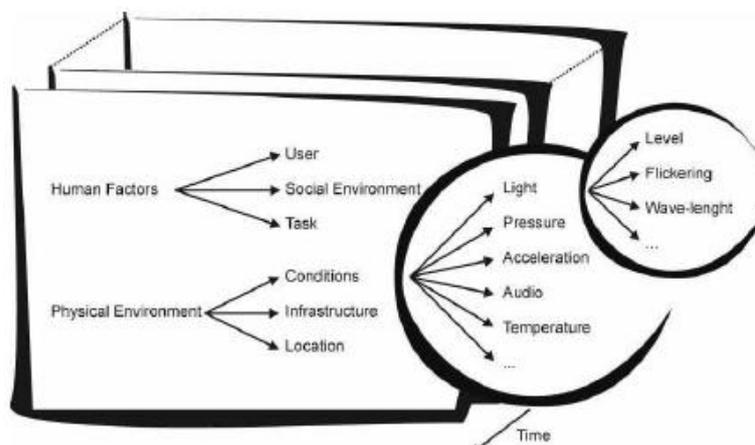
Para lograr este objetivo, así como los sistemas adaptativos se valen del modelo de usuario, los sistemas ubicuos en general, y los de aprendizaje ubicuo en particular, se valen del modelo de contexto.

#### I.1.1.4. MODELO DE CONTEXTO

El contexto en un sistema ubicuo se define como toda aquella información que puede ser usada para caracterizar la situación de una entidad. Por entidad se entiende una persona, un dispositivo, una ubicación o lugar, o una aplicación o programa de computadora (Dey *et al.*, 2000).

Schmidt *et al.* (1999) proponen el siguiente modelo para estructurar el contexto en un sistema de computación móvil consciente del contexto.

El contexto describe la situación y el entorno en el que se encuentra un usuario o dispositivo. Se define a través de un nombre único, y, para cada contexto, existe un conjunto de características que se consideran relevantes. A su vez, para cada una de ellas, existe un rango de valores posibles, determinado por el contexto de forma explícita o implícita. Este modelo se puede observar en la figura I.3.



**Figura I.3:** Dimensiones del contexto (Schmidt *et al.*, 1999).

Tomando como base las características descritas anteriormente, se puede definir una jerarquía de información del contexto formada por dos categorías superiores que separan los aspectos del usuario de los del entorno. Luego cada una de estas, se divide en tres subcategorías que proporcionan la estructura general del contexto. Dentro de cada una de estas seis categorías se pueden identificar las características relevantes al contexto particular, las cuales pueden agruparse por niveles.

- **Factores humanos:**
  - **Usuario:** hábitos, estado emocional, condiciones bio-fisiológicas, etc.
  - **Entorno social:** ubicación de otros, interacción social, dinámica grupal, etc.
  - **Tareas:** tareas espontáneas, tareas dedicadas, objetivos generales.
- **Entorno físico:**
  - **Ubicación:** posición absoluta, posición relativa, etc.
  - **Infraestructura:** recursos de computación y comunicación cercanos, etc.
  - **Condiciones físicas:** ruido, luz, temperatura, presión, humedad, etc.

En Yau & Joy (2006) se presentan cuatro categorías de información de contexto a tener en cuenta para aplicaciones móviles conscientes del contexto:

- **Contexto computacional:** incluye la comunicación por redes, los costos de comunicación, el ancho de banda de la red y los recursos computacionales cercanos (por ejemplo, impresoras y PCs).
- **Contexto del usuario:** incluye el perfil del usuario, su ubicación, las personas cercanas y su situación social actual.
- **Contexto físico:** incluye la iluminación, los niveles de ruido, las condiciones del tráfico y la temperatura.
- **Contexto temporal:** incluye el momento del día, semana, mes y estación del año.

También se describen seis dimensiones de contexto a considerar en aplicaciones de aprendizaje móvil:

- **Identidad:** identificación única del estudiante, reconocida usualmente vía login o a través de dispositivos especiales como tarjetas inteligentes.
- **Espacio-Temporal:** formada por tiempo y ubicación. Posibilita determinar el instante o período durante el cual el usuario requerirá cierta información.
- **Facilidades:** consiste en el tipo de dispositivo móvil, como PDA, teléfono móvil, teléfono inteligente, Tablet PC o Notebook; y las capacidades del mismo como poder del CPU, tamaño de la pantalla, color, resolución y método de entrada.
- **Actividad:** contiene el conjunto de actividades del proceso de aprendizaje.
- **Estudiante:** agrupa las propiedades intrínsecas y psicológicas del alumno, como estado emocional, nivel de concentración, etc.
- **Comunidad:** corresponde al contexto social, que representa los usuarios cercanos, sus interacciones, etc.

Por otra parte, Hwang *et al.* (2008) definen las siguientes categorías (denominadas parámetros de situación) que se deben tener en cuenta al conducir actividades de aprendizaje en un ambiente de u-learning sensible al contexto:

- **Contexto personal:** es sentido por el sistema e incluye la ubicación del estudiante y el tiempo de arribo, temperatura, nivel de transpiración, latidos, presión sanguínea, etc.
- **Contexto ambiental:** es sentido por el sistema e incluye los ID de sensores y ubicación, la temperatura, humedad, ingredientes del aire, y otros parámetros del ambiente alrededor del sensor, así como los objetos que se aproximan al sensor.
- **Retroalimentación:** producida por el estudiante a través de su dispositivo móvil, incluye la información observada o sensada de los ítems objetivo (como temperatura del ambiente y valor ácido del agua, contaminación del aire, forma y color de un árbol, estado de una máquina luego de llevar a cabo una operación, etc.) fotos adquiridas o interacciones con el sistema de aprendizaje (por ejemplo, las respuestas a los distintos ítems de una prueba o el log de operación del sistema).
- **Información personal:** se recupera desde una base de datos e incluye el perfil del estudiante y su portfolio de aprendizaje, tal como la planificación predefinida del estudiante, tiempo esperado de inicio de la actividad de aprendizaje, los tiempos más largo y más corto aceptables para una actividad de aprendizaje, el lugar de

aprendizaje, los caminos de aprendizaje o secuencias de un curso, las limitaciones o prohibiciones de una actividad de aprendizaje, etc.

- **Información del ambiente:** es recuperada de la base de datos e incluye la información detallada del lugar de aprendizaje, como la planificación de las actividades de aprendizaje dispuestas en el sitio, las restricciones o reglas de administración del sitio, notas para el uso del sitio, el equipo ubicado en el sitio, las personas que usan o administran el sitio, etc.

#### I.1.1.5. CLASES DE ADAPTACIÓN

La información contenida en los modelos del estudiante y del contexto permite al sistema individualizar a cada alumno en función de sus características particulares y del entorno que lo rodea. Luego, el sistema debe definir las clases de adaptaciones (es decir, *que* se adapta) que realizará sobre la experiencia de aprendizaje, en función de las distintas características de los modelos.

Brusilovsky (1996) define dos clases de adaptación que se pueden implementar en los sistemas de hipermedia adaptativos. Para que un sistema se considere de esta clase debe reunir tres características: debe utilizar hipertexto o hipermedia para mostrar su contenido, debe contar con un modelo de usuario y debe ser capaz de adaptar la hipermedia en función de este modelo.

- **Adaptación de la presentación:** considera al contenido presentado al usuario como un conjunto de fragmentos ensamblados en función de diferentes criterios. Como resultado, la información que se muestra al usuario no es siempre la misma, por más que se trate del mismo concepto o actividad. La adaptación de la presentación se puede realizar a través de distintos métodos:
  - **Explicaciones adicionales:** oculta o muestra información adicional sobre un concepto en particular, en función de la relevancia de la misma para cada caso.
  - **Explicaciones con prerrequisitos:** antes de mostrar un concepto determinado, se presentan explicaciones de todos los conceptos definidos como prerrequisitos del mismo, y para los que el usuario no tenga un nivel adecuado.
  - **Explicaciones comparativas:** se basa en la idea de similitud de conceptos. Si el usuario conoce un concepto similar al que se está mostrando actualmente, el sistema presenta una comparación que pone de manifiesto las semejanzas y diferencias entre los mismos.

- **Explicaciones y variantes:** el sistema almacena diferentes variantes para ciertas partes del contenido y cada usuario es presentado con la variante más adecuada de acuerdo a sus características.
- **Ordenamiento:** los fragmentos de información acerca del concepto presentado que resultan más relevantes para el usuario son presentados en primer lugar y siguiendo un orden descendente de importancia.
- **Adaptación de la navegación:** auxilian al usuario en la determinación del camino a seguir dentro del hiperespacio en función de sus características individuales. Se distinguen 5 tecnologías usadas con este propósito:
  - **Orientación directa:** se provee al usuario con una secuencia de hipervínculos, donde cada vínculo siguiente inmediato es su mejor opción actual, de acuerdo a las características de su modelo.
  - **Ordenamiento adaptativo:** los hipervínculos de una misma página se ordenan en función de su relevancia, de acuerdo a alguna característica del usuario
  - **Ocultamiento adaptativo:** se ocultan los hipervínculos que no son relevantes al usuario para el caso actual.
  - **Anotación de vínculos:** se agregan comentarios a los hipervínculos, que proveen más información o indicios del tipo de recursos a los que se accede con cada uno de ellos. Estas anotaciones pueden ser textuales o en forma de íconos.
  - **Anotación de mapas:** se adaptan, de diversas formas, los mapas de navegación locales y globales presentados al usuario.

Por su parte, Graf & Kinshuk (2008) presentan las siguientes categorías de adaptación para los sistemas de aprendizaje ubicuo:

- **Interacción con el estudiante:** el sistema provee indicaciones personalizadas en un momento determinado o sugiere actividades de aprendizaje apropiadas en función del contexto.
- **Orientación de ubicación:** el sistema ayuda a los estudiantes a dirigirse hacia un lugar físico donde se puedan llevar a cabo experiencias de aprendizaje, en función de sus necesidades.
- **Adaptación del material de estudio y actividades:** comprende las clases propuestas por Brusilovsky (1996) y mencionadas anteriormente: adaptación del contenido y de la navegación. Además, cobra especial importancia la adaptación de la presentación del contenido en función del dispositivo móvil utilizado por el alumno.

- **Interacción entre pares y con expertos:** el sistema puede favorecer la comunicación síncrona y asíncrona del estudiante con otros individuos. Con respecto a la comunicación síncrona, los foros de discusión, servicios de preguntas y respuestas y de intercambio del conocimiento se pueden personalizar, por ejemplo, teniendo en cuenta el contexto del usuario al momento de realizar las preguntas para inferir información extra que caracterice mejor su situación. En el caso de la comunicación asíncrona, el sistema puede asistir al estudiante ayudándolo a formar grupos de estudio o indicándole quien es el individuo más adecuado para ayudarlo y/o responder sus preguntas.

### I.1.2. ONTOLOGÍAS

En Filosofía, la Ontología es una disciplina que trata sobre la naturaleza y organización de la realidad. A diferencia de las ciencias experimentales que buscan modelar la realidad desde una perspectiva particular, la Ontología se enfoca en la naturaleza y estructura *per se* de las cosas, independientemente de otras consideraciones o, incluso, de su existencia real (Staab & Studer, 2010).

Dentro de esta disciplina, se encuentra la Ontología Formal, una rama de investigación ontológica que se basa en la siguiente premisa: la esencia de las cosas no está determinada únicamente por las cosas en sí mismas sino también por la contribución de quien las percibe y comprende. Su objetivo particular es estudiar las propiedades formales y clasificación de las entidades del mundo (objetos físicos, eventos, etc.) y las categorías de ese modelo particular del mundo (concepto, propiedad, etc.) (Gómez-Pérez *et al.*, 2004).

A partir de la década del 90, y tomando como base los fundamentos teóricos propuestos por la disciplina Ontología Formal, las ontologías se constituyeron como un área de interés en la investigación dentro de las Ciencias de la Computación, particularmente en el área de Inteligencia Artificial como un instrumento para facilitar el intercambio y reuso de conocimiento (Breitman *et al.*, 2007).

Años más tarde, luego de la revolucionaria propuesta de la Web Semántica (Berners-Lee *et al.*, 2001), cobraron un protagonismo aún mayor y se popularizaron en otras comunidades dentro de la Informática, como Integración Inteligente de la Información, Sistemas de Información Cooperativos e Ingeniería del Software basada en agentes, ya que representan uno de los conceptos clave para la realización de la misma.

Gómez-Pérez *et al.* (2004) citan una de las primeras definiciones del término ontología en el ámbito de la Informática:

*“Una ontología define los términos y relaciones básicos que comprenden el vocabulario de un área temática así como también las reglas para combinar dichos términos y las relaciones para definir extensiones a dicho vocabulario”.*

De allí en más, se han propuesto numerosas definiciones del término, que varían en función de la perspectiva de cada autor de acuerdo a su área de investigación y/o en el aspecto que eligen enfatizar. Una de las definiciones más citadas es la propuesta por Studer *et al.* (1998), que a su vez es una combinación y ampliación de otras dos definiciones:

*“Una ontología es una especificación explícita y formal de una conceptualización compartida. Por conceptualización se entiende un modelo abstracto de algún fenómeno para el cual se han identificado sus conceptos relevantes. Explícita significa que los tipos de conceptos usados y las restricciones para su uso han sido explícitamente definidos. Formal se refiere al hecho de que la ontología debe ser procesable por una máquina. Compartida refleja la noción de que la ontología captura conocimiento conceptual consensuado, es decir, no privativo de un individuo sino aceptado por un grupo.”*

Por su parte, Uschold & Grüninger (1996) definen a una ontología como *“una comprensión compartida de algún dominio de interés”* y *“un vocabulario de términos y alguna especificación de su significado”*.

Estas dos últimas definiciones presentan un significado mucho más amplio del término, diluyendo su significado. De acuerdo a Gómez-Pérez *et al.* (2004), la comunidad ontológica realiza una distinción entre las ontologías que representan principalmente taxonomías, de aquellas que modelan el dominio más profundamente haciendo uso de una mayor variedad de restricciones semánticas.

Por un lado, se encuentran las ontologías ligeras (*lightweight*) que incluyen conceptos, taxonomías de conceptos, relaciones entre conceptos y propiedades para describirlos; y por el otro, están las ontologías complejas o de peso (*heavyweight*) que incluyen además de las entidades anteriores, axiomas y restricciones. Estos últimos se encargan de clarificar el significado previsto para los términos que se incluyen en la ontología.

De acuerdo a otros autores (Breitman *et al.*, 2007; Sanz & Jiménez Ruiz, 2009) las ontologías ligeras no son consideradas ontologías. Esto se observa en la figura I.4.

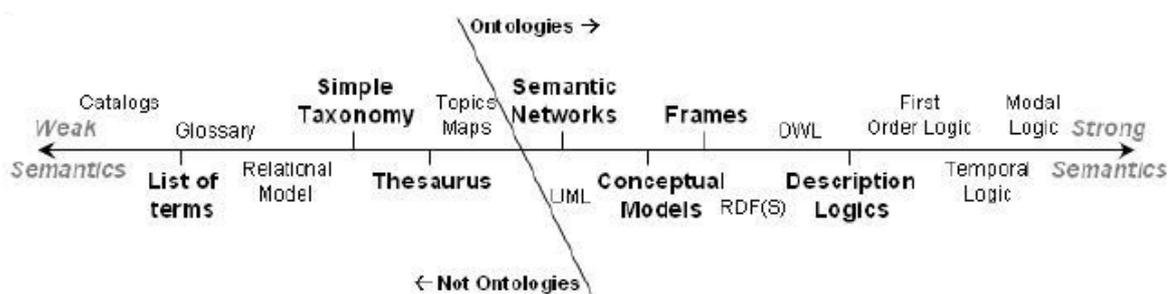


Figura I.4: Espectro de ontologías (Sanz & Jiménez-Ruiz, 2009).

### I.1.2.1. CLASIFICACIÓN Y APLICACIONES

Las ontologías pueden clasificarse de acuerdo a su nivel de generalidad del siguiente modo (Breitman *et al.*, 2007):

- **Ontologías de nivel superior (*upper-ontologies*):** describen conceptos genéricos, como ser espacio, tiempo y eventos. Estas ontologías son, en principio, independientes del dominio y pueden reusarse para construir nuevas ontologías.
- **Ontologías de dominio:** describen el vocabulario perteneciente a un dominio determinado, especializando los conceptos provistos por las ontologías de nivel superior.
- **Ontologías de tareas:** describen el vocabulario requerido para realizar tareas genéricas o actividades, especializando los conceptos provistos por las ontologías de nivel superior.
- **Ontologías de aplicación:** describen el vocabulario de una aplicación específica, cuyos conceptos corresponden, en general, a los roles de las entidades de un dominio particular al realizar alguna tarea o actividad.

Existen otras clasificaciones de ontologías que responden a criterios diferentes del presentado anteriormente. Éstos se presentan en Breitman *et al.* (2007) y Gómez-Pérez *et al.* (2004), y las catalogan de acuerdo a: el tipo de información representada o tema de conceptualización, y el espectro semántico o riqueza de su estructura interna. Cabe destacar que este último criterio guarda una estrecha relación con la distinción entre ontologías ligeras y de peso descripta en la sección previa.

De acuerdo a Noy & McGuinness (2001) las principales razones que motivan el desarrollo de ontologías son:

- Compartir el entendimiento común de la estructura de información entre personas o agentes de software.
- Permitir la reutilización de conocimiento de un dominio.

- Explicitar suposiciones de un dominio.
- Separar el conocimiento del dominio del conocimiento operacional.
- Analizar el conocimiento de un dominio.

A menudo, el desarrollo de una ontología de dominio no es la meta en sí. Desarrollar una ontología es comparable a definir un conjunto de datos y sus estructuras para que otros programas los utilicen. Los métodos de resolución de problemas, distintas aplicaciones independientes del dominio e incluso agentes de software, pueden emplear ontologías y bases de conocimiento construidas a partir de ontologías, como datos de entrada para sus tareas.

Tan variados como los motivos de construcción de ontologías son las áreas de aplicación de las mismas: administración del conocimiento, procesamiento de lenguaje natural, comercio electrónico, integración inteligente de la información, recuperación de la información, diseño e integración de bases de datos, bio-informática, educación y web semántica.

Ejemplos de aplicaciones basadas en ontologías y descripciones de las ontologías más destacadas pueden encontrarse en Staab & Studer (2010) partes III y VI, Gómez-Pérez et al (2004) capítulo 2, Obitko (2001) sección 4, Uschold & Gruninger (1996) sección 7 y Breitman et al (2007) capítulo 9 y parte IV.

#### I.1.2.2. COMPONENTES

De acuerdo a Noy & McGuinness (2001) una ontología está formada por los siguientes componentes:

- **Clases o conceptos:** representan los conceptos del dominio de discurso. Una clase puede contener subclases, que representan conceptos más específicos que su superclase. Por ejemplo: asignatura, docente.
- **Slots, roles o propiedades:** para cada concepto, que describen sus características y atributos. Por ejemplo: para una asignatura, nombre, módulo al que pertenece, docente que la dicta.
- **Facetas o restricciones de rol:** definen limitaciones sobre los slots o propiedades. Por ejemplo, una asignatura puede ser dictada por un máximo de 3 docentes.
- **Instancias o individuos:** representan individuos específicos que pertenecen a una clase. Por ejemplo, Programación I, Simulación.

Una descripción más formal es la presentada por Breitman *et al* (2007), donde una ontología se describe a través de una quintupla  $O = \{C, R, CH, rel, OA\}$ , donde:

- **C** y **R**: representan dos conjuntos disjuntos, denominados el conjunto de **conceptos** y el conjunto de **relaciones**, respectivamente.
- **CH**  $\subseteq C \times C$ : es una **jerarquía de conceptos** o **taxonomía**, donde  $CH(C_1, C_2)$  indica que  $C_1$  es un **subconcepto** de  $C_2$ .
- **rel**:  $R \rightarrow C \times C$ : es una función que relaciona conceptos no taxonómicamente.
- **OA**: es un conjunto de axiomas, expresados en un lenguaje lógico apropiado.

La forma específica de representar y vincular los distintos componentes de una ontología, así como la terminología utilizada para describirlos, se encuentra directamente relacionada con el paradigma de representación del conocimiento que se elija para modelar la misma.

### I.1.2.3. PARADIGMAS DE REPRESENTACIÓN DEL CONOCIMIENTO, LENGUAJES Y HERRAMIENTAS

Históricamente, los paradigmas de representación del conocimiento utilizados para modelar ontologías han sido las redes semánticas y los basados en marcos (Sanz & Jiménez-Ruiz, 2009). Para cada uno de estos paradigmas existe un conjunto de lenguajes que se pueden utilizar para implementar las ontologías, así como distintas herramientas de software que los soportan.

Estos métodos de representación carecían de una semántica formal basada en lógica que les permitiera realizar inferencias, y es por ello que los formalismos basados en lógica descriptiva se introdujeron como una alternativa para superar esta limitación.

La lógica descriptiva (DL, Description Logic) denota una familia de formalismos de representación del conocimiento que permiten modelar un dominio de aplicación de una forma estructurada y formalmente bien comprendida (Staab & Studer, 2010). Éstos proveen teorías y sistemas para expresar información y poder razonar de forma semánticamente precisa. La DL es un subconjunto de la Lógica de Primer Orden (FOL, First Order Logic), la cual tiene un poder expresivo muy alto pero también un coste computacional elevadísimo, que es lo que ésta trata de compensar al restringir la clase de información que puede representar (Sanz & Jiménez-Ruiz, 2009).

OWL es un lenguaje cuyo objetivo es proveer un medio para publicar e intercambiar ontologías en la Web, creado por el Grupo de Trabajo de Ontologías Web (WebOnt) de la W3C (Gómez-Pérez *et al.*, 2004).

OWL es un derivado del lenguaje DAML+OIL y se construye sobre las bases de RDF(S), en un intento de ampliar las capacidades de expresión de éste último, sin dejar de ser utilizable en términos prácticos de implementación, ya que en esta clase de lenguajes existe una relación directa entre el poder de expresión y el soporte al razonamiento. En general, cuanto más rico sea el lenguaje en términos de expresividad, más ineficiente se tornará el soporte al razonamiento, cruzando a menudo la frontera de la incomputabilidad (Staab & Studer, 2010).

Para satisfacer este requerimiento, OWL se dividió en 3 capas o especies: OWL Lite, OWL DL, y OWL Full, siendo OWL-DL la especie de OWL basada en lógica descriptiva.

Actualmente, OWL se encuentra en su versión 2 y es una recomendación de la W3C, por lo que utilizarlo en la creación de ontologías implica una mayor reusabilidad e interoperabilidad para la misma.

Existen distintas herramientas que permiten crear ontologías en este lenguaje, siendo Protégé-OWL la más utilizada (Sanz & Jiménez-Ruiz, 2009). La misma ha sido desarrollada por el grupo de Informática Médica de Standford (SMI) perteneciente a la Universidad del mismo nombre, y está disponible en una versión web que puede ejecutarse sin necesidad de instalación desde un navegador, y en versión escritorio para distintas plataformas.

Protégé OWL se encuentra actualmente en su versión 4.3 y permite la creación de ontologías en lenguaje OWL-DL a través del uso de una interfaz gráfica intuitiva y completa (figura I.5), sin necesidad de escribir código.

Destacan, además, los numerosos plugins que se pueden descargar e integrar con el entorno, como ser, razonadores (Pellet, RacerPro), generadores de esquemas gráficos (OWLviz), interfaces de consulta (SPARQL Query, OWL2 Query) y otros (OntoGraf, OWLDiff, API, etc).

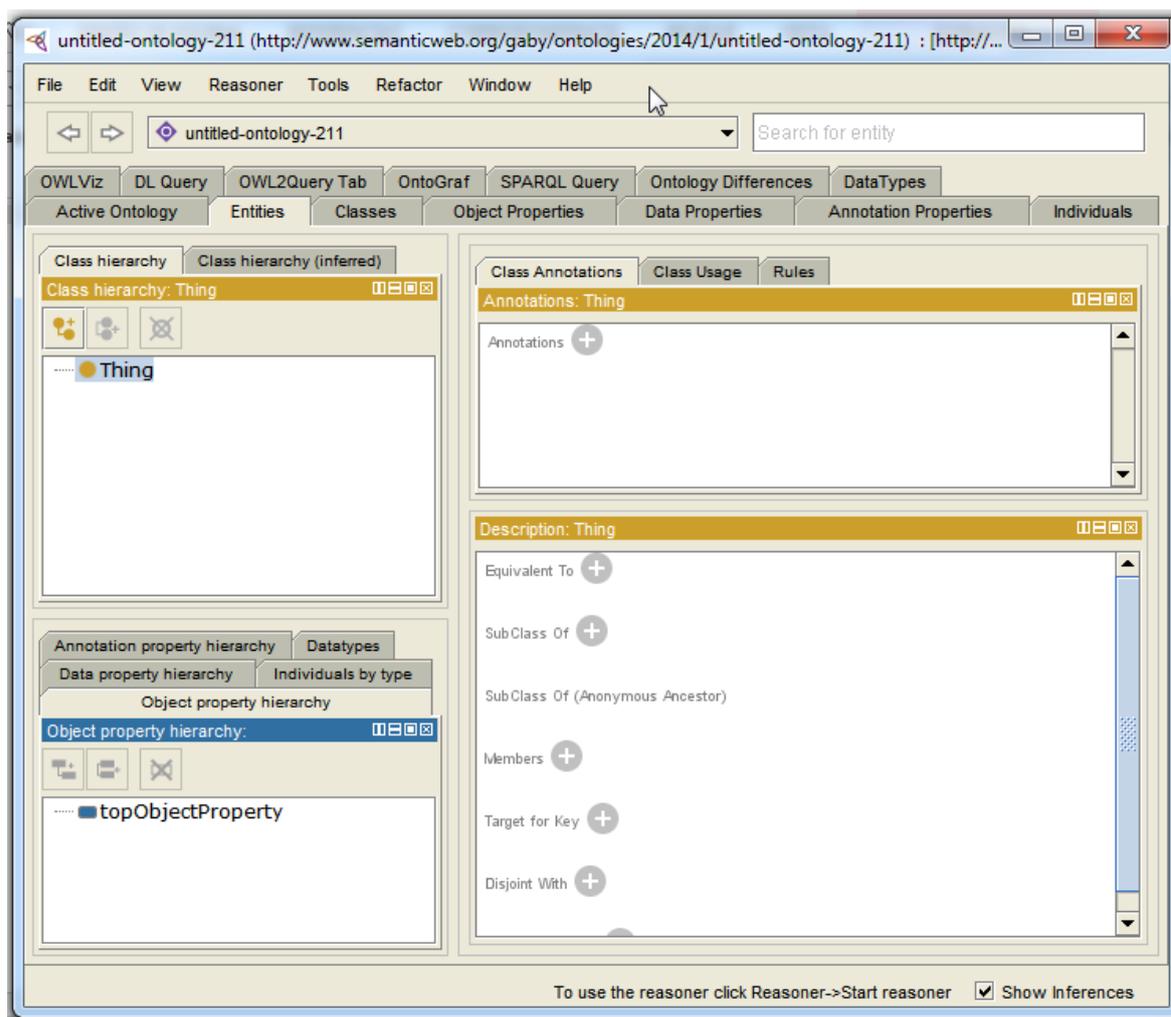


Figura I.5: Protégé OWL 4.3, versión desktop.

#### I.1.2.4. MODELADO DE ONTOLOGÍAS CON OWL-DL

Una ontología OWL-DL está formada por:

- **Individuos:** representan los objetos del dominio de interés.
- **Propiedades:** vinculan a dos individuos entre sí, o bien a un individuo con un valor literal.
- **Clases:** representan agrupaciones de individuos y se organizan en una jerarquía de super y subclases, denominada Taxonomía. En OWL todas las clases son subclases de la clase *Thing* (cosa).

El razonamiento en OWL-DL está basado en la Asunción de Mundo Abierto u OWA (Open World Reasoning), por lo que el mismo se suele denominar Razonamiento de Mundo Abierto u OWR (Open World Reasoning).

La OWA especifica que la existencia de algo es posible mientras no se especifique explícitamente lo contrario. Es decir que, sólo porque algo no se ha declarado verdadero no

se puede asumir que es falso, sino que se asume que el conocimiento es posible, pero que aún no se lo ha agregado a la base de conocimientos. Esto implica que lo único que se toma como base para el razonamiento en el OWR, es lo que está explícitamente declarado (y lo que se puede inferir a partir de tales declaraciones).

Por ejemplo, se crean las clases *Niño* y *Adulto* sin definir ninguna otra restricción o axioma sobre las mismas. Luego se crea un individuo de tipo *Niño*, denominado *Pablito* y dos individuos de tipo *Adulto* denominados *Juan* y *Sandra*, y se especifica que *todos los individuos son diferentes entre sí* (es decir, que representan distintas personas). Si se consulta la ontología buscando *todos los individuos que sean Adultos* se obtiene como resultado a *Juan* y *Sandra*. Si se buscan *todos los individuos que sean Niños* se obtiene a *Pablito*. Pero si se buscan *todos los individuos que no sean Adultos* no se obtiene ningún resultado, ya que no se ha especificado que si algo es *Niño* no puede ser *Adulto*. Por ende se asume que posible ser *Niño* y *Adulto* a la vez. Ahora, si las clases *Niño* y *Adulto* se declaran *disjuntas* (es decir, que no pueden tener instancias en común), si se puede deducir que si alguien es *Adulto* entonces no es *Niño*, y como consecuencia la consulta *todos los individuos que no sean Adultos* da como resultado a *Pablito*.

#### I.1.2.4.1. PROPIEDADES

Las propiedades en OWL representan relaciones. Hay dos tipos principales de propiedades:

- **Propiedades de objetos:** relacionan dos individuos entre sí.
- **Propiedades de datos:** relacionan un individuo con un valor definido a través de un tipo de datos del esquema XML, o un literal.

Las propiedades pueden tener sub-propiedades, propiedades inversas, características y dominio y rango.

**Sub-propiedades.-** Una sub-propiedad especializa a sus super-propiedades, es decir que, *si la propiedad P es sub-propiedad de una propiedad O y P relaciona a los sujetos X e Y, entonces X e Y también están relacionados por O.*

Cuando las propiedades tienen sub-propiedades, se generan jerarquías de propiedades, que pueden ser tanto de propiedades de objetos como de propiedades de datos, pero nunca de distintos tipos en la misma jerarquía.

**Propiedades inversas.-** Las propiedades de objetos pueden tener propiedades inversas. *Si la propiedad P relaciona al sujeto X con Y, entonces su inversa P' relaciona a Y con X.*

**Características de propiedades.-** enriquecen el significado de las mismas dotándolas de cualidades especiales. Se aplican a las propiedades de objetos, y para las propiedades de datos sólo es válida la característica funcional.

- **Funcional:** *si una propiedad  $P$  es funcional, entonces un individuo  $x$  se puede relacionar con a lo sumo (como máximo) un único individuo  $z$  a través de la propiedad  $P$ .* Esta propiedad también se llama Propiedad de Valor Único o Rasgo. Ejemplo: la propiedad *tiene Madre Biológica*.
- **Funcional Inversa:** *si una propiedad  $P$  es funcional inversa y  $O$  es la propiedad inversa de  $P$ , entonces  $O$  es una propiedad funcional.* Ejemplo: la propiedad *es Madre Biológica De*.
- **Transitiva:** *si una propiedad  $P$  es transitiva y ésta relaciona  $x$  con  $z$  y  $z$  con  $w$ , entonces se puede decir que  $P$  relaciona  $x$  con  $w$ .* Notar que, si una propiedad  $P$  es transitiva, entonces su inversa también lo es. Además, si es transitiva no puede ser funcional. Ejemplo: la propiedad *tiene Ancestro*.
- **Simétrica:** *si una propiedad  $P$  es simétrica y relaciona a  $x$  con  $z$ , entonces  $z$  también se relaciona con  $x$  vía  $P$ .* En otras palabras, una propiedad que es simétrica es su propia inversa. Ejemplo: la propiedad *tiene Hermano*.
- **Asimétrica:** *si una propiedad  $P$  es asimétrica y relaciona a  $x$  con  $z$ , entonces  $x$  no puede relacionarse con  $z$  vía  $P$ .* Ejemplo: la propiedad *es Hijo De*.
- **Reflexiva:** *una propiedad  $P$  es reflexiva cuando la propiedad relaciona al individuo  $x$  consigo mismo.* Ejemplo: la propiedad *conoce*.
- **Irreflexiva:** *una propiedad  $P$  es irreflexiva si relaciona a  $x$  con  $z$  donde  $x$  siempre es distinto de  $z$ .* Ejemplo: la propiedad *es Madre De*.

**Dominio y rango de propiedades.-** Una propiedad relaciona individuos del dominio con individuos del rango. *Siendo  $D$  ( $R$ ) es dominio (rango) de la propiedad  $P$ , si  $A$  se relaciona con  $B$  a través de  $P$ , entonces  $A$  ( $B$ ) es de tipo, es decir, pertenece a la clase,  $D$  ( $R$ ).*

En este sentido, hay que destacar que los dominios y rangos de propiedades no se tratan como restricciones sino como *axiomas*. Por ejemplo, dada la propiedad *produce* con dominio *Bodega* y rango *Vino*, si se vinculan a través de la misma el individuo *BodegaMendoza* con el individuo *MalbecMendoza2008* y el individuo *QueseríaSanMarcos* con el individuo *QuesoGouda*, esto no genera un error en la ontología sino que permite inferir que *QueseríaSanMarcos* es una *Bodega* y *QuesoGouda* es un *Vino*.

Si para el dominio (rango) de una propiedad se especifica más de una clase, se interpreta como la *intersección* de las mismas. Además, en general, *para una propiedad  $P$ , si  $D$  es su dominio y  $R$  su rango, entonces su inversa  $P'$  tendrá dominio  $R$  y rango  $D$ .*

#### I.1.2.4.2. CLASES

Para definir la jerarquía de clases, se hace uso del axioma de subclase. Además, existe un axioma de clases equivalentes y un axioma para establecer clases disjuntas.

**Subclase.-** La relación de subclase es una relación de implicancia necesaria, es decir que, *si  $B$  es subclase de  $A$ , entonces todas las instancias de  $B$  son instancias de  $A$ , sin excepción.*

**Clases disjuntas.-** *Si una clase  $A$  es disjunta con una clase  $B$ , entonces si  $x$  pertenece a  $A$  no puede pertenecer a  $B$ .* Debido a la OWA, y tal como se explicó anteriormente, la indicación de que dos clases no pueden tener instancias en común debe hacerse explícitamente.

**Clases equivalentes.-** *Si una clase  $A$  se define como equivalente a una clase  $B$ , entonces ambas clases contienen los mismos individuos.*

Además, en OWL, se puede describir una clase a través de enumeración, restricciones sobre propiedades y operaciones sobre conjuntos.

**Clases enumeradas.-** *Son aquellas definidas a través de la enumeración precisa de todos sus individuos, o bien de un conjunto de literales.* Por ejemplo: la clase Fin de Semana puede definirse como el conjunto de los individuos {Sábado, Domingo}.

**Restricciones sobre propiedades.-** Describe a una clase de individuos basándose en las relaciones en las que participan los miembros de la misma. Una restricción describe una clase anónima (sin nombre) que contiene a todos los individuos que satisfacen la misma, es decir, todos los individuos que tienen las relaciones requeridas para pertenecer a la clase.

OWL permite incorporar 3 tipos de restricciones:

- **Restricciones de Cuantificación:** puede ser existenciales o universales.
  - **Las restricciones existenciales ( $\exists$ )** describen clases de individuos que participan en al menos una relación para una propiedad específica con miembros de una clase determinada.
  - **Las restricciones universales ( $\forall$ )** describen clases de individuos que, para una propiedad determinada, solo pueden tener relaciones con individuos de una clase específica. Las restricciones universales no especifican la existencia de una

relación, sólo determinan que de existir la misma, entonces está restringida a la clase en cuestión.

- **Restricciones de Cardinalidad:** describen las clases de individuos que tengan al menos, cómo máximo o exactamente un número de relaciones con otros individuos o valores de datos. Para una propiedad P,
  - **La restricción de cardinalidad mínima** especifica el número mínimo de relaciones P en las que debe participar un individuo para pertenecer a la clase.
  - **La restricción de cardinalidad máxima** especifica el número máximo de relaciones P en las que puede participar un individuo.
  - **La restricción de cardinalidad** especifica el número exacto de relaciones P en las que debe participar un individuo.
    - **Las restricciones de cardinalidad calificadas** son un caso especial de las restricciones de cardinalidad porque especifican, además del número exacto de relaciones, la clase de objetos con la que deben relacionarse.
- **Restricciones Tiene Valor ( $\exists$ ):** describen al conjunto de individuos que tienen al menos una relación para una propiedad determinada y con un individuo específico. Por ejemplo: `tienePaisDeOrigen  $\exists$  Italia`.

**Operaciones de la teoría de conjuntos.-** las clases pueden describirse también a través de clases anónimas formadas por el resultado de aplicar los operadores de unión, intersección y complemento sobre un conjunto de clases.

En OWL se distinguen dos tipos de clases: clases primitivas y clases definidas.

**Clases primitivas.-** *Son aquellas que tienen especificadas sólo condiciones necesarias de pertenencia a la clase, las cuales se expresan a través del uso de superclases. Con este tipo de clases se pueden hacer afirmaciones de la forma: “Si x es miembro de la clase A entonces x cumple con el conjunto de condiciones necesarias N definidas para A”.* Por ejemplo, si la clase Gato se define como subclase de la clase Felino, se sabe que si x es un Gato entonces es también un Felino.

**Clases definidas.-** *Son aquellas que especifican al menos un conjunto de condiciones suficientes de pertenencia a la clase, a través de la definición de clases equivalentes. Con este tipo de clases, se puede realizar clasificación de individuos, es decir, realizar afirmaciones de la forma “Si x cumple con el conjunto de condiciones suficientes S definidas para A, entonces x pertenece a A”.*

### **I.1.2.4.3. INDIVIDUOS**

OWL permite crear individuos y afirmar distintas propiedades sobre ellos.

Se debe tener en cuenta que OWL no utiliza la Asunción de Nombre Único u UNA (Unique Name Assumption) lo cual quiere decir que dos individuos con distinto nombre, pueden representar al mismo individuo. Por ello, se debe especificar explícitamente para cada individuo si es igual que o diferente de otro(s) individuos(s).

## **I.2. MARCO METODOLÓGICO**

### **I.2.1. METODOLOGÍA DE DESARROLLO DE ONTOLOGÍAS**

Actualmente se encuentran en la literatura distintas metodologías para el desarrollo o construcción de ontologías, cada una de las cuales ha sido propuesta a partir de las experiencias concretas de cada autor.

Luego de analizar distintas metodologías presentadas en Breitman *et al.* (2007) y Gómez-Pérez *et al.* (2004), se decidió tomar como base a la propuesta presentada por Noy & McGuinness (2001), *Ontology 101*, realizando una adaptación parcial de la misma y utilizando ciertos criterios/recursos presentados en METHONTOLOGY (Fernández-López *et al.*, 1997).

Se eligieron estas metodologías porque son las que presentan más claramente los pasos a seguir con respecto al modelado conceptual de la ontología. La combinación fue necesaria porque *Ontology 101* está orientada al formalismo basado en marcos, por lo que los últimos pasos no se pueden aplicar a las ontologías basadas en DL, y METHONTOLOGY, en el intento de proveer un modelo conceptual independiente del paradigma de representación del conocimiento usado, genera un número excesivo de representaciones intermedias (generalmente en forma de tablas), dificultando la comprensión del modelo.

En las secciones siguientes, se describen las dos metodologías utilizadas como base y la resultante de su combinación.

### I.2.1.1. ONTOLOGY 101

Definida por Noy & McGuinness (2001), constituye una guía detallada para el desarrollo de una ontología, enfocándose en la creación de los componentes de la misma con el formalismo basado en marcos.

Sus pasos son:

- 1) **Determinar el dominio y alcance de la ontología:** consiste en responder a un conjunto de preguntas sobre su propósito y usos previstos, y definir los requerimientos de la ontología en términos de preguntas de competencia.
- 2) **Considerar el reuso de ontologías existentes:** implica revisar las ontologías disponibles públicamente, analizando si las mismas pueden servir como base total o parcial para la ontología que se quiere construir.
- 3) **Enumerar términos importantes:** sobre los que se querrían hacer afirmaciones o explicar al usuario.
- 4) **Definir las clases y su jerarquía:** se realiza en paralelo con el paso cinco, y consiste en identificar, de la lista de términos, los conceptos más generales que representan las clases de la ontología y ordenarlos de modo jerárquico. Para esto se pueden utilizar los enfoques top-down, bottom-up o combinado, siempre teniendo en mente que definir una clase como subclase de otra representa una especialización de conceptos.
- 5) **Definir las propiedades de las clases:** las propiedades son, por lo general, los términos restantes de la lista definida en el paso tres, y representan relaciones entre los individuos de las distintas clases.
- 6) **Definir las facetas de las propiedades:** que describen características adicionales de las mismas, como tipo de valor permitido, valores por defecto, número de valores permitidos y otras.
- 7) **Crear instancias:** para ello se elige una clase, se crea la instancia o individuo perteneciente a la misma, y se completan los valores de sus propiedades.

### I.2.1.2. METHONTOLOGY

Esta metodología fue desarrollada en el Laboratorio de Inteligencia Artificial de la Universidad Politécnica de Madrid (Fernández-López et al., 1997) y comprende todas las etapas del proceso de desarrollo, desde la concepción de la idea de la ontología hasta su mantenimiento, realizando una descripción especialmente detallada de las actividades

involucradas en el proceso de modelado conceptual de la misma. Los pasos propuestos son independientes del formalismo de representación elegido.

Las etapas propuestas por METHONTOLOGY son:

- 1) **Planeamiento:** desarrollar un plan que contenga todas las actividades a realizar, incluyendo fechas, estimaciones, recursos y herramientas necesarias.
- 2) **Especificación:** definir el alcance y objetivos de la ontología, creando un documento de especificación de la ontología que puede tener distintos niveles de detalle y/o lenguaje en el que se encuentra escrito.
- 3) **Conceptualización:** elicitación de los conceptos a ser incluidos en la ontología, para lo cual se pueden utilizar las técnicas tradicionales de desarrollo del software. Luego estos requerimientos deben plasmarse en un modelo conceptual. Las actividades incluidas en el modelado conceptual son:
  - a. **Crear el glosario de términos:** con la definición, sinónimos y acrónimos de todos los términos a incluir en la ontología.
  - b. **Crear las taxonomías de conceptos:** donde se representa la relación jerárquica entre las distintas clases. Esta tarea se puede realizar utilizando uno de tres enfoques: top-down, bottom-up y middle-out. Además se recomienda hacer uso no sólo de la relación de subclase sino también de las relaciones de descomposición disjunta, descomposición exhaustiva y partición (descomposición disjunta exhaustiva).
  - c. **Construir diagramas de relaciones binarias:** entre los conceptos de la ontología, definiendo además los conceptos de origen y destino de las relaciones.
  - d. **Construir el diccionario de conceptos:** donde se describen las instancias de cada concepto, sus atributos de instancia y clase, y las relaciones binarias en las que participan.
  - e. **Describir en detalle las relaciones binarias:** indicando su nombre, los nombres de los conceptos fuente y destino, su cardinalidad, su relación inversa y las propiedades matemáticas que posee.
  - f. **Describir en detalle los atributos de instancia:** que son aquellos cuyos valores pueden ser diferentes para cada instancia del concepto. Para cada uno se debe especificar su nombre, concepto al que pertenece, el tipo, la unidad de medida, la precisión y rango de valores, los valores por defecto, la cardinalidad máxima y mínima, etc.

- g. Describir en detalle los atributos de clase:** que son los que definen conceptos y toman sus valores de la clase en que se definen. Éstos no son heredados por las subclases ni por las instancias. Para cada uno de ellos, se debe especificar: nombre, nombre del concepto donde el atributo es definido, tipo de valor, valor, unidad de medida y precisión, cardinalidad, etc.
  - h. Describir en detalle las constantes:** definiendo para cada una: nombre, tipo, valor, unidad de medida y los atributos que pueden inferirse a partir de la constante.
  - i. Describir axiomas formales:** es decir, las expresiones lógicas que representan información que es siempre cierta y que por lo general se usa para especificar limitaciones en la ontología. Se debe indicar: nombre, descripción en lenguaje natural, la expresión lógica que describe el axioma usando lógica de primer orden, los conceptos, atributos y relaciones a los que se refiere el axioma y las variables usadas.
  - j. Describir reglas:** éstas se usan para chequear restricciones y para inferir valores de atributos, instancias de relaciones, etc. Se debe indicar para cada una de ellas: nombre, descripción en lenguaje natural, expresión que define la regla, conceptos, atributos y relaciones a los que se refiere la regla y variables usadas en la expresión.
  - k. Definir instancias:** indicando nombre, nombre del concepto al que pertenece, y los valores de sus atributos si son conocidos.
- 4) Formalización:** formalizar el modelo conceptual creado en el paso previo en algún lenguaje formal de representación de ontologías. Por ejemplo, lógica descriptiva o marcos.
  - 5) Integración:** integrar la ontología en desarrollo con otras ontologías existentes.
  - 6) Implementación:** codificar la ontología en un lenguaje procesable por computadoras, como OWL.
  - 7) Evaluación:** verificar y validar la ontología para asegurar su calidad y concordancia con estándares.
  - 8) Documentación:** al igual que con cualquier otro artefacto software, se debe documentar para facilitar su mantenimiento y reuso.
  - 9) Mantenimiento:** para adecuar la ontología a la evolución de las condiciones del dominio que representa.

### I.2.1.3. METODOLOGÍA RESULTANTE

Tomando como base los pasos de Ontology 101, se modificaron los pasos cinco y seis para adecuarlos al modelo de representación de la lógica descriptiva, se quitó el paso siete original, y se agregaron los pasos siete y ocho, en base a las tareas nueve y diez de la actividad de conceptualización propuesta en METHONTOLOGY, pero sin utilizar la notación específica propuesta. Además, se integró en el paso cuatro la utilización de las relaciones taxonómicas propuestas por METHONTOLOGY en la tarea dos de la etapa de conceptualización.

Los pasos resultantes obtenidos se describen a continuación.

#### **Paso 1.- Determinación del dominio y alcance de la ontología.**

Tiene por objetivo identificar el dominio y alcance de la ontología respondiendo a las siguientes preguntas básicas:

- ¿Cuál es el dominio que la ontología cubrirá?
- ¿Para qué usaremos la ontología?
- ¿Para qué tipos de preguntas la información en la ontología deberá proveer respuestas?
- ¿Quién usará y mantendrá la ontología?

Una de las formas más comunes de determinar el alcance de la ontología es bosquejando una lista de preguntas que la base de conocimientos basada en la ontología debería ser capaz de responder.

Estas preguntas se conocen como preguntas de competencia y se utilizan en la etapa de evaluación para probar la ontología. Las mismas son solamente un bosquejo y no necesitan ser exhaustivas.

#### **Paso 2.- Consideración de reuso de ontologías.**

Tiene por objetivo identificar si existen ontologías candidatas al reuso, a través de la búsqueda y análisis de las ontologías existentes, para comprobar si se pueden refinar y/o extender para el dominio y tarea particular sobre el que se trabaja.

Por otra parte, reusar ontologías existentes puede ser un requerimiento a priori, si el sistema en cuestión necesita interactuar con otras aplicaciones que ya contienen ontologías particulares o vocabularios controlados. Hay un número de ontologías disponibles en forma electrónica que pueden ser importadas en los entornos de desarrollo, algunos de los cuales permiten incluso la conversión (semi) automática de un formalismo a otro.

También puede suceder que no existan ontologías que satisfagan los requerimientos de una aplicación o dominio particular (o que no se encuentren disponibles públicamente), en cuyos casos se deberá construir la ontología desde cero.

### **Paso 3.- Enumeración de términos relevantes para la ontología.**

Su propósito es confeccionar una lista con todos los términos con los que se querrían crear enunciados o dar explicación al usuario, identificando los términos de interés y sus propiedades. Esta lista se puede obtener analizando los requerimientos o preguntas de competencia definidos en el primer paso.

Inicialmente, se busca obtener una lista integral de términos sin preocuparse de las relaciones entre términos, si éstos se solapan o si representan conceptos o propiedades.

Los dos pasos siguientes (desarrollo de la jerarquía de clases y definición de las propiedades de los conceptos) están estrechamente relacionados. Típicamente, no se realizan de forma secuencial sino que se crean unas cuantas definiciones de los conceptos de la jerarquía para luego continuar describiendo las propiedades de esos conceptos, y así sucesivamente.

### **Paso 4.- Definición de clases y su taxonomía.**

Existen distintos enfoques que se pueden utilizar para desarrollar una jerarquía de clases:

- El proceso de **desarrollo top-down** comienza con la definición de los conceptos más generales en el dominio y va avanzando con la subsecuente especialización de los conceptos.
- El proceso de **desarrollo bottom-up** comienza con la definición de las clases más específicas (las hojas de la jerarquía) y continúa con el subsecuente agrupamiento de dichas clases en conceptos más generales.
- El proceso de **desarrollo combinado** es el resultado de una composición de los enfoques top-down y bottom-up. Primero se definen los conceptos más sobresalientes y luego se los generaliza y especializa apropiadamente.

Independientemente del enfoque que se utilice, se comienza por definir las clases. De la lista creada en el Paso 3, se seleccionan los términos que describen objetos que tienen existencia independiente, en oposición a aquellos términos que describen dicho objetos. Estos términos “base” serán las clases de nivel superior de la ontología. Luego, se organizan las clases restantes en una taxonomía jerárquica, preguntando si al ser instancia de una clase, el objeto será necesariamente (por definición) instancia de alguna otra clase.

En otras palabras: *Si una clase A es superclase de una clase B, entonces cada instancia de B es también una instancia de A.*

Al crear la taxonomía de clases, es conveniente utilizar, además de la relación de subclase, otras tres relaciones taxonómicas propuestas en METHONTOLOGY:

- **Descomposición disjunta:** de un concepto C es un conjunto de subclases de C que no tienen instancias comunes y que no cubren a C, es decir, puede haber instancias del concepto C que no son instancias de ninguno de los conceptos en la descomposición.
- **Descomposición exhaustiva:** de un concepto C es un conjunto de subclases de C que cubren C y pueden tener instancias y subclases comunes, es decir, no puede haber instancias del concepto C que no sean instancias de al menos uno de los conceptos de la descomposición.
- **Partición:** de un concepto C es un conjunto de subclases de C que no comparten instancias comunes pero que cubren C, es decir, no hay instancias de C que no sean instancias de alguno de los conceptos en la partición.

#### **Paso 5.- Definición de propiedades de objetos y de datos.**

Las clases aisladas no proveerán suficiente información para responder las preguntas de competencia del Paso 1. Para completar la información que se busca representar se deben definir las propiedades de objetos y de datos de la ontología.

Para identificar las distintas propiedades se puede utilizar la lista construida en el paso 3, donde todos los conceptos que no representen clases constituirán, seguramente, propiedades de algún tipo.

También se debe indicar en esta instancia, el dominio y rango de las distintas propiedades.

#### **Paso 6.- Definición de las características de las propiedades.**

Una vez que se cuenta con las propiedades definidas, se deben especificar las características, también denominadas propiedades matemáticas, de cada una de ellas.

Para identificar qué características corresponden a cada propiedad, se pueden plantear ejemplos de afirmaciones en donde se evidencia el significado de las características para cada propiedad.

#### **Paso 7.- Definición de axiomas formales y descripción de clases.**

En este paso se busca especificar toda la información restante para caracterizar a cada clase. Esto incluye la descripción de las clases a través de a clase a través de la

definición de conjuntos de enumeración, restricciones sobre propiedades y operaciones de la teoría de conjuntos.

Cada una de estas descripciones se adosará a la clase como condiciones necesarias (superclase) o condiciones suficientes (clases equivalentes).

### **Paso 8.- Definición de reglas de inferencia.**

El último paso consiste en especificar las reglas de inferencia adicionales, si es que resultan necesarias.

## **I.2.2. METODOLOGÍA DE EVALUACIÓN DE ONTOLOGÍAS**

Las ontologías, como cualquier otro programa o componente de un sistema, deben ser evaluadas y testeadas antes de su implementación y uso en el entorno real de operación.

Si bien una evaluación/testeo satisfactorio no garantiza la ausencia de errores u omisiones en el diseño, ofrece una cierta garantía y posibilita el uso de las mismas de una manera más segura.

Gómez-Pérez *et al.* (2004) identificaron tres clases de errores taxonómicos que ocurren frecuentemente durante la construcción de ontologías. Fahad & Qadir (2008) extendieron esta categorización basándose en los distintos errores encontrados durante la construcción de su sistema DKP-OM para la fusión de ontologías.

A continuación se presentan la clasificación extendida de estos distintos tipos de errores.

- **Errores de inconsistencia:** pueden ser de tres clases:
  - **Errores de circularidad:** cuando una clase es definida como subclase o superclase de sí misma en algún nivel de la jerarquía de la ontología. Puede ocurrir a distancia 0, 1 o n dependiendo del número de relaciones involucradas al recorrer el concepto hacia abajo en la jerarquía hasta volver al concepto de origen. Por ejemplo, si se define la clase *Viajero* como subclase de *Persona* y luego la clase *Persona* como subclase de *Viajero*, se está ante un error de circularidad de nivel 1.
  - **Errores en particiones:** las clasificaciones de conceptos se pueden definir de manera disjunta (descomposición disjunta), completa (descomposición exhaustiva) y disjunta completa (partición). Cuando un conjunto de subclases se define de uno de estos tipos pero luego existen clases o instancias que violan esta definición se está en frente a un error de este tipo, que puede ser causado por:

- **Clases o instancias comunes en descomposiciones disjuntas y particiones:** por ejemplo, se define una descomposición disjunta entre las clases *Transporte Aéreo*, *Transporte Marítimo* y *Transporte Terrestre* y luego se crea una clase *Hidroavión* como subclase de *Transporte Aéreo* y de *Transporte Marítimo*.
- **Instancias externas en descomposiciones exhaustivas y particiones:** por ejemplo, se define la descomposición exhaustiva de *Paquete Turístico* formada por las subclases *Viaje Económico*, *Viaje de Negocios* y *Viaje de Placer* y luego se crea la instancia de *Paquete Turístico de John* como subclase de *Paquete Turístico*.
- **Errores semánticos:** cuando se clasifica un concepto como subclase de un concepto al cual no pertenece realmente. Las causas más comunes que generan este tipo de error son:
  - **Subclase con dominio débil:** cuando la clase que representa el dominio más amplio se define como subclase de aquella que posee un dominio más reducido. Por ejemplo, se define la clase *Producto Lácteo* como subclase de *Queso*, siendo que el concepto *Producto Lácteo* es más general que *Queso*.
  - **Subclase con brecha en el dominio:** cuando en la subclase se definen características adicionales con respecto a la superclase, pero que violan alguna de las características de la misma. Por ejemplo, se crea la clase *Pizza China* como subclase de la clase *Pizza Vegetariana*, siendo que una pizza china contiene vegetales pero puede también contener carne.
  - **Subclase con dominio disjunto:** cuando se especifica una clase como subclase de otra que pertenece a un dominio totalmente diferente. Por ejemplo, se definen las clases *Jugo* y *Hamburguesa* como subclases de la clase *Comestibles*, siendo que *Jugo* no pertenece a la clase de alimentos comestible.
- **Errores de incompletitud:** por lo general, surgen cuando se construye la taxonomía de clases haciendo uso solamente de la relación de subclase y no de las otras relaciones taxonómicas. Pueden ser errores:
  - **De clasificación incompleta:** cuando no se consideran ciertos conceptos de un dominio al clasificar un concepto particular. Por ejemplo, se definen las subclases *Vino Tinto* y *Vino Blanco* para la clase *Vino*, pero no se tiene en cuenta la clase *Vino Rosado*.

- **En particiones:** suceden cuando se omite definir información que permite caracterizar más detalladamente a una entidad. Pueden ser de:
  - **Omisión de axioma de disyunción:** cuando no se especifica que un conjunto de clases hermanas son disjuntas. Por ejemplo, no se especifica que las clases *Perro* y *Gato* son disjuntas, siendo que algo que es un perro nunca puede ser un gato.
  - **Omisión de axioma de exhaustividad:** cuando un conjunto de subclases cubre un dominio pero esto no se especifica explícitamente. Por ejemplo, se especifica que la clase *Usuario* tiene las subclases *Invitado* y *Administrador* pero no se declara que todo *Usuario* es igual o bien a *Invitado* o bien a *Administrador*.
  - **Omisión de indicación de característica funcional:** cuando no se especifica que una propiedad es funcional siendo que la misma puede tomar un solo valor. Por ejemplo, no se declara que la propiedad *tiene madre biológica* es funcional, cuando se sabe que una persona puede tener únicamente una madre biológica.
  - **Omisión de indicación de característica inversa funcional:** cuando no se especifica que una propiedad posee una propiedad inversa que es funcional. Por ejemplo, no se especifica que la propiedad *es madre biológica de* es inversa funcional, siendo que esta propiedad se define como inversa a *tiene madre biológica*.
  - **Omisión de condiciones suficientes para una clase:** cuando no se definen las condiciones suficientes de una clase, lo cual no permite clasificar a individuos dentro la misma. Por ejemplo, se define como condición suficiente de la clase *Viaje Económico* “*es un Viaje cuyo precio es menor a 1500\$*”, siendo que el hecho de que algo sea un viaje y tenga un precio menor a 1500\$ indica inequívocamente que es un *Viaje Económico*.
- **Errores de redundancia:** cuando cierta información particular se infiere más de una vez a partir de las relaciones de clases e instancias de la ontología. La redundancia se puede manifestar en:
  - **Subclases, subpropiedades e instancias:** ocurren cuando se especifican clases que poseen más de una relación de subclase, directa o indirectamente, con una clase particular. Por ejemplo, se especifica la clase *Playa* como subclase de *Ubicación* y de *Lugar*, *Ubicación* ya está definida como subclase de *Lugar*.

- **Definición de clases, propiedades e instancias:** cuando se definen objetos diferentes (con nombre igual o distinto) que poseen la misma definición formal.
- **Axiomas de disyunción:** cuando una clase se define como disjunta de otra, de manera explícita, más de una vez.

# CAPÍTULO II

## DESARROLLO DE LA ONTOLOGÍA DEL ESTUDIANTE

---

En el presente capítulo se describe el diseño e implementación de la ontología del estudiante, detallando el proceso de desarrollo seguido en su construcción.

En la primera sección se identifican las características de los estudiantes que resultan relevantes para la personalización en sistemas de u-learning. Luego, en la segunda sección, se define el modelo conceptual de la ontología, desarrollado siguiendo la metodología descrita en la sección I.2.1.3 del capítulo anterior. En la tercera y última sección se presenta la implementación de la ontología en lenguaje OWL-DL con la herramienta Protégé-OWL.

### II.1. DIMENSIONES DEL ESTUDIANTE PARA PERSONALIZACIÓN EN SISTEMAS DE U-LEARNING

Las técnicas de personalización cumplen un rol fundamental en los sistemas de aprendizaje, ya que permiten al estudiante aprender de manera más eficiente, al adaptarse a sus características particulares. Como tal, la determinación de estas características es un aspecto sumamente importante y nada trivial, puesto que del nivel de acierto y exactitud con el que se representen las mismas, dependerá en gran medida el éxito del sistema (Martins *et al.*, 2008).

Originalmente se propuso realizar la identificación de las características de los estudiantes, que resultan relevantes para llevar a cabo la personalización en sistemas de u-learning, mediante una investigación exploratoria sobre el uso de herramientas de computación ubicua en la Facultad de Ciencias Exactas y Tecnologías de la UNSE. Sin

embargo, luego de un estudio preliminar, se determinó que no existían experiencias de *u-learning* en la institución, por lo tanto, la identificación requerida se realizó a partir de la lectura y el análisis de bibliografía perteneciente al área de modelado de usuario y de contexto, así como también de investigaciones donde se documentaban desarrollos concretos sobre el tema.

La motivación principal al determinar que características se deben incluir en el modelo del estudiante es la de lograr un modelo genérico, lo suficientemente flexible como para poder integrarse en diferentes sistemas. Esto no quiere decir que el mismo intentará modelar todas las dimensiones posibles con respecto al usuario, sino que proveerá una base común a partir de la cual extender o adaptar el modelo.

### II.1.1. INFORMACIÓN DEL ESTUDIANTE

Para determinar las características del estudiante a incluir en el modelo, se analizaron las propuestas de diferentes autores (Fröschl, 2005; Sosnovsky, 2007; Graf *et al.*, 2009; Martins *et al.*, 2008) y el modelo IMS LIP (Smythe *et al.*, 2001), descriptos en la sección I.1.1.2 del capítulo I.

Para seleccionar las categorías de información a incluir se tuvieron en cuenta los siguientes criterios:

- **Categorías comunes:** se incluyeron las categorías que se consideraban en la mayoría de las propuestas analizadas.
- **Necesidades de adaptación:** se incluyeron las categorías que abarcaban información suficiente para llevar a cabo las clases de adaptación descritas en la sección I.1.1.5 del capítulo I.
- **Factibilidad de adquisición de la información:** este criterio sirvió como un filtro de las categorías resultantes de aplicar los dos criterios anteriores, ya que permitió descartar aquellas categorías que requerían de métodos especializados o demasiado complejos para su obtención.

Las categorías resultantes y su significado se describen en las sub-secciones siguientes. Las tres primeras categorías pertenecen a la clase de información independiente del dominio, mientras que las dos últimas representan información dependiente del dominio del sistema.

### II.1.1.1. CATEGORÍAS INDEPENDIENTES DEL DOMINIO

- **Identificación**

También denominada Datos Demográficos, esta categoría contiene información estática sobre el estudiante, tal como: nombre, ID, fecha de nacimiento, sexo, dirección, teléfono, e-mail, etc.

Esta clase de información se usa principalmente con propósitos administrativos y no de adaptación. También es útil a la hora de inicializar el modelo del estudiante, y en general es provista directamente por el alumno, al registrarse en el sistema (Han 2001).

- **Estilo de aprendizaje**

El estilo de aprendizaje es una de las categorías de información independiente del dominio que se encuentra presente en prácticamente todos los modelos de estudiante de sistemas educativos (Han, 2001; Castillo & Breda, 2003; Jovanovic *et al.*, 2009; Graf *et al.*, 2009; Pramitasari *et al.*, 2009; Cataldi & Lage, 2012; Panagiotopoulos *et al.*, 2012). El mismo describe las diferencias individuales que se manifiestan durante el proceso de aprendizaje, existiendo distintas teorías o modelos que se pueden utilizar para su representación (Montgomery & Groat, 1998).

Uno de los modelos más utilizados en los sistemas de e-learning es el propuesto por Felder y Silverman (Graf *et al.*, 2007). Éste representa cada estilo a través de una combinación de preferencias en cuatro dimensiones. Cada dimensión representa una característica del alumno y las preferencias representan los dos valores que puede tomar la misma (Felder & Silverman, 1988). Los valores de cada dimensión se muestran en la tabla II.1.

Este modelo propone además, un cuestionario denominado Índice de Estilos de Aprendizaje compuesto por cuarenta y cuatro preguntas de opción doble, en función de cuyas respuestas se puede detectar el estilo de aprendizaje del alumno.

**Tabla II.1:** Dimensiones y preferencias según Felder-Silverman.

<b>Dimensión</b>	<b>Preferencias</b>
<b>Percepción</b>	Sensitivo/Intuitivo
<b>Entrada</b>	Visual/Verbal
<b>Procesamiento</b>	Activo/Reflexivo
<b>Comprensión</b>	Secuencial/Global

- **Habilidades cognitivas**

Las habilidades cognitivas representan el conocimiento procedural del alumno. Gracias a ellas los estudiantes son capaces de pensar de manera crítica, analítica y

reflexiva, formular las preguntas correctas, tomar decisiones, resolver problemas y aprender de manera independiente (Otero *et al.*, 2012).

Existen un número de investigaciones que tratan sobre la detección y/o entrenamiento de tales características, de las cuales tomaremos las propuestas por Graf *et al.* (2009) y Roll *et al.* (2005): autoevaluación, búsqueda de ayuda, pensamiento crítico, razonamiento científico y resolución de problemas.

### II.1.1.2. CATEGORÍAS DEPENDIENTES DEL DOMINIO

- **Conocimiento actual**

Esta categoría es una representación del estado de conocimiento del alumno con respecto al contenido de aprendizaje, tal como es percibido por el sistema.

Podría decirse que esta es la categoría más importante, puesto que en función del conocimiento que el alumno posee en un momento dado, se puede adaptar la complejidad de las tareas, las actividades, el material sugerido, etc. Con esto se logra un proceso de aprendizaje más eficiente, ya que el mismo se construye en función del conocimiento real poseído por el alumno (Martins *et al.*, 2008).

- **Rendimiento en el curso**

Esta categoría representa el avance del alumno con respecto al material de aprendizaje, en términos de la estructura de los cursos ofrecidos por el sistema. Además, incluye registros de las lecciones completadas, los pedidos de ayuda, el tiempo de resolución de las actividades, etc.

Si bien estas dos categorías se incluyen en el modelo del estudiante, porque son imprescindibles para realizar tareas de personalización en función del nivel de conocimientos y del avance del curso, no se desarrollan en mayor detalle en la presente propuesta. Su inclusión es principalmente a título informativo, puesto que su estructura y contenido son dependientes de la estructura y contenido de cada curso/asignatura particular, y el modelo propuesto es un modelo genérico aplicable a sistemas de aprendizaje ubicuo que implementen cursos de cualquier temática. En el marco del Proyecto de Investigación en el que se inserta este trabajo, esta información se describe en otras ontologías tales como la ontología de la Estrategia de Aprendizaje y la ontología del Dominio.

## II.1.2. INFORMACIÓN DE CONTEXTO

A partir de la literatura revisada (Schmidt *et al.*, 1999; Yau & Joy, 2006; Hwang *et al.*, 2008) y descrita en el capítulo I, sección I.1.1.4, se pueden definir las siguientes categorías comunes de información de contexto:

- **Ubicación:** es la categoría principal que se tiene en cuenta al considerar el contexto de un usuario. Se suele representar por coordenadas.
- **Condiciones físicas o ambientales:** se considera información como nivel de luz, nivel de ruido, clima, temperatura, etc.
- **Entorno o dispositivos computacionales:** procesadores, dispositivos, sensores, redes de comunicación, etc.
- **Entorno social:** personas y lugares ubicados cerca del usuario.
- **Tiempo:** identificando día y hora.

Teniendo en cuenta estas categorías, y revisando el trabajo de Hong & Cho (2008), presentado brevemente en la introducción sección ii, se comprobó que, CALA-ONT, la ontología propuesta por los mismos toma en cuenta toda la información mencionada anteriormente, implementándola a través de una estructura sencilla pero completa.

Por lo tanto, para definir las categorías del modelo se tomaron como base las clases principales de la misma, que se modificaron en función de los requerimientos propios del modelo a construir. Las clases resultantes se describen a continuación.

- **Ubicación**

Representa el lugar donde se encuentra ubicado el estudiante. Una ubicación puede ser de interior o de exterior, en el sentido de si está ubicada dentro o fuera de las instalaciones de la institución educativa. A su vez, una ubicación de interior puede clasificarse (no exhaustivamente) en aula, box o laboratorio.

- **Actividad**

Representa una actividad realizada por el estudiante. Por actividad se debe entender cualquier acción general que el mismo puede llevar a cabo en algún escenario educativo, por ejemplo, estudiar, participar de una reunión de grupo, rendir un examen, etc. No debe confundirse con actividad pedagógica, como ser un ejercicio del curso o una evaluación. Las actividades se clasifican en actividades formales o informales, siendo las actividades formales aquellas que se asocian con actividades oficiales de la institución educativa, como ser, clases o evaluaciones. Por el contrario, las informales son actividades que pueden ser

iniciadas por el alumno, como por ejemplo, una reunión de grupo o descansar en el patio de la institución.

- **Entidad Computacional**

Representa los distintos artefactos de hardware y software que forman parte del entorno del alumno. Una entidad computacional puede ser: una aplicación, un dispositivo o una red. A su vez, un dispositivo puede ser una impresora, una PC, un proyector, un Smartphone o una tablet.

## II.2. MODELO CONCEPTUAL DE LA ONTOLOGÍA

En esta sección se describen los pasos realizados para obtener la ontología del modelo del estudiante, denominada *onto-alu* y desarrollada utilizando la metodología descrita en la sección I.2.1.3 del capítulo I.

### II.2.1. DETERMINACIÓN DEL DOMINIO Y ALCANCE DE LA ONTOLOGÍA

Es el primer paso en el proceso de desarrollo de la ontología, y sirve para identificar el ámbito de información de la misma, los usos y usuarios previstos y, principalmente, los requerimientos que debe satisfacer.

La ontología del estudiante *onto-alu* tiene como propósito representar el modelo del estudiante para sistemas de aprendizaje ubicuos, conteniendo las principales características del alumno y su contexto, en tanto resulten relevantes para la personalización de la experiencia de aprendizaje en un sistema de u-learning.

*Onto-alu* está formada por la información descrita en las secciones II.1.1 y II.1.2 del presente capítulo, y centrada en la información independiente del dominio. Será utilizada por un sistema, o aplicación, de aprendizaje ubicuo para dar soporte a las tareas de personalización del mismo.

A continuación se detallan los requerimientos de la ontología, que no han sido expresados en forma de preguntas de competencia porque se consideró que resultaba más claro enunciarlos como afirmaciones.

Los requerimientos del 1 al 9 representan requisitos referidos al tipo de información que se debe poder almacenar en la ontología, mientras que los requerimientos del 10 al 22 representan restricciones sobre las relaciones que pueden tener las distintas entidades del sistema.

- R1.** Debe ser posible identificar al estudiante a través de su nombre, su DNI, su sexo, su fecha de nacimiento, su domicilio, su teléfono y su dirección de correo electrónico.
- R2.** Debe ser posible representar el estilo de aprendizaje del estudiante de acuerdo al Modelo de Felder & Silverman, expresado como un conjunto de 4 valores, donde cada uno de ellos corresponde a un tipo (preferencia) por dimensión.
- R3.** Debe ser posible definir el nivel del estudiante para cada una de las siguientes habilidades: autoevaluación, búsqueda de ayuda, pensamiento crítico, razonamiento científico y resolución de problemas; en una escala cualitativa con valores {Alto, Medio, Bajo}.
- R4.** Para cada ubicación, debe ser posible definir: si es de interior o exterior, su descripción textual, su dirección y sus coordenadas GPS.
- R5.** Para cada ubicación de interior, debe ser posible definir: si es un aula, box o laboratorio, su nivel de iluminación {Adecuado o bajo} y su nivel de ruido {Adecuado, Alto o Muy Alto}.
- R6.** Para cada actividad, debe ser posible definir: si es formal o informal, su descripción, su horario de inicio y su horario de finalización.
- R7.** Para cada actividad formal debe ser posible definir si es una clase o una evaluación.
- R8.** Para cada entidad computacional, debe ser posible definir: si es un dispositivo, una red, o una aplicación; y su descripción.
- R9.** Para cada entidad de tipo dispositivo debe ser posible definir si es una impresora, una PC, un proyector, un Smartphone o una Tablet.
- R10.** Un estudiante puede tener un único conjunto de datos de identificación.
- R11.** Un estudiante puede tener un único estilo de aprendizaje.
- R12.** Un estudiante puede tener un único nivel por habilidad.
- R13.** Un estudiante puede encontrarse en una única ubicación.
- R14.** Una ubicación puede contener más de un estudiante simultáneamente.
- R15.** Una ubicación puede contener más de una entidad computacional simultáneamente.
- R16.** Una ubicación puede contener más de un actividad simultáneamente.
- R17.** Un estudiante puede realizar una única actividad.

**R18.** Una actividad puede estar siendo realizada por un único estudiante.

**R19.** Una actividad puede desarrollarse en una única ubicación.

**R20.** Un estudiante puede hacer uso de varias entidades computacionales simultáneamente.

**R21.** Una entidad computacional puede ser utilizada por un único estudiante.

**R22.** Una entidad computacional opera en una única ubicación.

## II.2.2. ANÁLISIS DE REUSO DE ONTOLOGÍAS EXISTENTES

El segundo paso de la metodología consiste en realizar un análisis de las ontologías relevantes existentes y considerar la factibilidad de reusarlas total o parcialmente, en lugar de diseñar y programar desde cero.

Al considerar la reutilización de ontologías existentes en el presente trabajo se realizó el siguiente análisis:

- **Ontologías específicas:** Si bien en la literatura se encuentran numerosos trabajos acerca de sistemas de e/u-learning adaptativos, en los cuales el estudiante y/o el contexto se representan a través de ontologías, ninguno de ellos provee el modelo conceptual o de implementación completo de dichas ontologías, con lo cual no se hace posible el análisis y consecuente reuso de las mismas.
- **Ontologías base (upper-ontologies):** Actualmente no se cuenta con una ontología general de educación o del estudiante entre los trabajos de investigación realizados en el área. Si existe una ontología general de usuario, GUMO (Heckmann *et al.*, 2007), que intenta representar, de modo genérico, la información requerida para modelos de usuario de sistemas adaptativos. Sin embargo, la misma contiene más de mil términos, siendo demasiado abarcativa como para ser considerada en el presente trabajo.

Por otra parte, se tomaron como base las clases de la ontología de contexto CALA-ONT (Hong & Cho, 2008) al realizar el diseño de la ontología presentada en este trabajo. Esta ontología provee una representación genérica de contexto para sistemas de aprendizaje, que a su vez no es excesivamente compleja como otras ontologías de contexto de nivel superior como SOUPA (Chen *et al.*, 2004) o COBRA-ONT (Chen *et al.*, 2003).

### II.2.3. DEFINICIÓN DE CLASES Y SU JERARQUÍA

Dado que ya se cuenta prácticamente con toda la información requerida para el modelo, especificada y dividida en categorías, se saltea el paso tres, donde se debió enumerar términos importantes para la ontología, y se continúa directamente con la definición de las clases de la ontología y su jerarquía.

En función de las categorías descritas en la sección II.1.1 y II.1.2 del presente capítulo, y de los requerimientos detallados en el primer paso, se definieron las clases de la ontología, agrupándolas por niveles hasta obtener la jerarquía deseada.

Para construir esta jerarquía se hizo uso de las relaciones taxonómicas recomendadas por METHONTOLOGY (Gómez-Pérez *et al.*, 2004), tal como se describió en el capítulo I, sección I.2.1.3. En la figura II.1 se puede observar la jerarquía de clases y los tipos de relaciones taxonómicas entre las subclases de mismo nivel de cada clase.

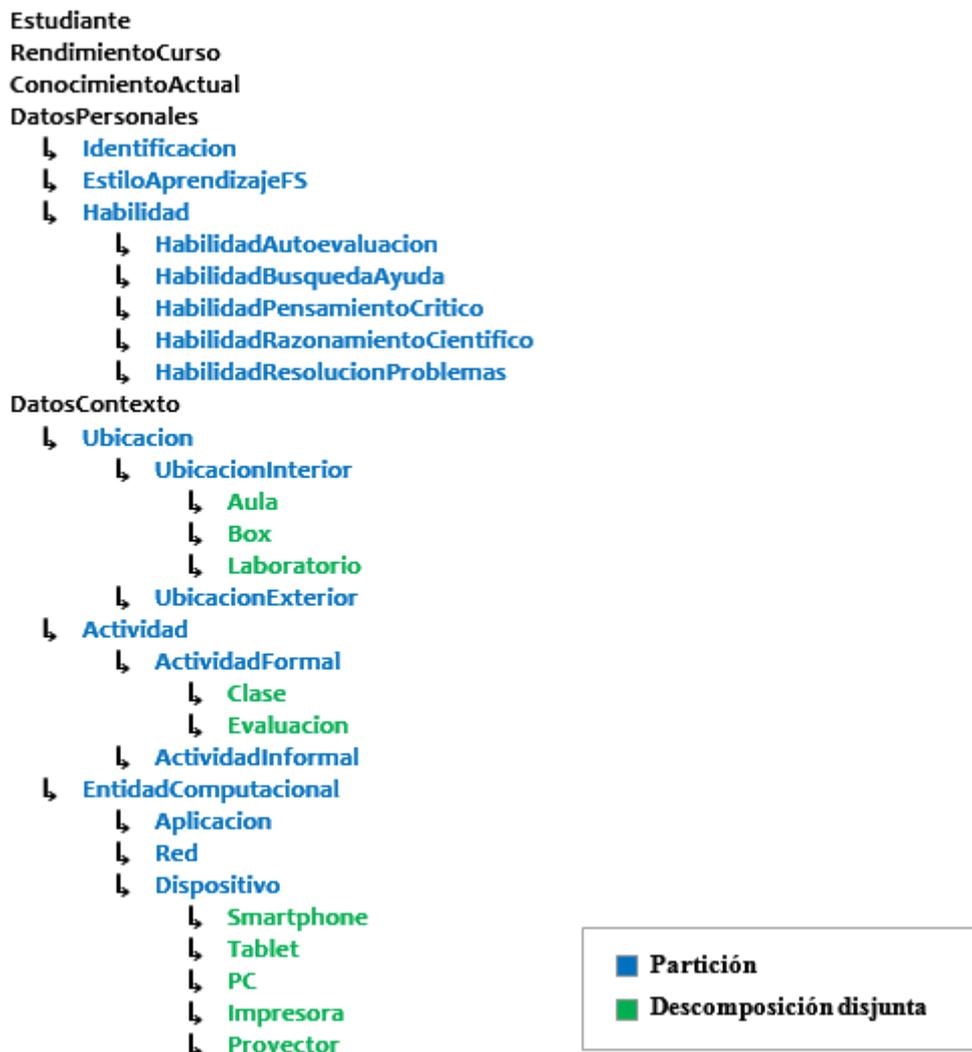


Figura II.1: Jerarquía de clases de *onto-au*.

## II.2.4. DEFINICIÓN DE PROPIEDADES

En este paso se definieron las propiedades de objetos y de datos que forman parte de la ontología del estudiante, especificando también el dominio y rango para cada una de ellas.

En la tabla II.2 se describen las propiedades de objetos de la ontología. Dentro de las mismas, solo existe una propiedad que posee subpropiedades: tieneHabilidad. Ésta cuenta con cinco subpropiedades, una por cada habilidad que se necesita representar, las que se muestran en la tabla con una pequeña indentación.

**Tabla II.2:** Propiedades de objetos de *onto-alu*.

Nº	Nombre	Descripción	Dominio	Rango
1	tieneIdentificacion	Relaciona a un estudiante con su conjunto de datos de identificación.	Estudiante	Identificacion
2	tieneHabilidad	Relaciona a un estudiante con sus habilidades.	Estudiante	Habilidad
3	tieneHabilidad-Autoevaluacion	Relaciona a un estudiante con su habilidad de autoevaluarse.	Estudiante	Habilidad-Autoevaluacion
4	tieneHabilidad-BusquedaAyuda	Relaciona a un estudiante con su habilidad de solicitar ayuda.	Estudiante	Habilidad-BusquedaAyuda
5	tieneHabilidad-PensamientoCritico	Relaciona a un estudiante con su habilidad de pensar críticamente.	Estudiante	HabilidadPensamientoCritico
6	tieneHabilidad-RazonamientoCientifico	Relaciona a un estudiante con su habilidad de razonar científicamente.	Estudiante	HabilidadRazonamientoCientifico
7	tieneHabilidad-ResolucionProblemas	Relaciona a un estudiante con su habilidad de resolver problemas.	Estudiante	HabilidadResolucionProblemas
8	esHabilidadDe	Relaciona a una habilidad con el estudiante que la posee.	Habilidad	Estudiante
9	tieneEstiloAprendizajeFS	Relaciona a un estudiante con su estilo de aprendizaje, de acuerdo a la clasificación propuesta por Felder & Silverman.	Estudiante	EstiloAprendizajeFS
10	esEstiloAprendizajeFSDe	Relaciona a un estilo de aprendizaje, según Felder & Silverman, con el estudiante que lo posee.	EstiloAprendizajeFS	Estudiante
11	tieneRendimiento-Curso	Relaciona al estudiante con su progreso y calificaciones en el curso.	Estudiante	Rendimiento-Curso
12	tieneConocimiento-Actual	Relaciona al estudiante con su estado de conocimiento actual con respecto al dominio del curso.	Estudiante	Conocimiento-Actual

13	ubicadoEn	Relaciona un estudiante con la ubicación donde este se encuentra.	Estudiante	Ubicacion
14	contieneA	Relaciona una ubicación con el estudiante que esta contiene.	Ubicacion	Estudiante
15	participaDe	Relaciona un estudiante con la actividad que este está realizando.	Estudiante	Actividad
16	tieneParticipante	Relaciona una actividad con el estudiante que la está llevando a cabo.	Actividad	Estudiante
17	utiliza	Relaciona un estudiante con las entidades computacionales que éste utiliza.	Estudiante	EntidadComputacional
18	esUtilizadaPor	Relaciona una entidad computacional con el estudiante que la utiliza.	EntidadComputacional	Estudiante
19	seDesarrollaEn	Relaciona una actividad con la ubicación donde esta se lleva a cabo.	Actividad	Ubicacion
20	tieneEnDesarrollo	Relaciona una ubicación con la actividad que se lleva a cabo en la misma.	Ubicacion	Actividad
21	operaEn	Relaciona una entidad computacional con la ubicación en la que la misma está funcionando.	EntidadComputacional	Ubicacion
22	tieneOperandoA	Relaciona una ubicación con las entidades computacionales que se encuentran funcionando en la misma.	Ubicacion	EntidadComputacional

En el caso de las propiedades de datos, no se especificó el dominio de ninguna de ellas porque hay ciertas propiedades que se reusan en distintas clases, por lo que se definió solo el rango de cada propiedad. En la tabla II.3 se describen las propiedades de datos de la ontología.

**Tabla II.3:** Propiedades de datos de *onto-alu*.

Nº	Nombre	Descripción	Rango
1	nombre	Nombre completo del estudiante. Ejemplo: Marcela Rodriguez	string
2	dni	Número de documento nacional de identidad del estudiante. Ejemplo: 34897562	int
3	sexo	Sexo del estudiante.	{"F", "M"}
4	fechaNacimiento	Fecha de nacimiento del estudiante. Ejemplo: 25-03-1990	string

5	direccion	Dirección postal. Usada para representar la dirección de un estudiante o de un lugar. Ejemplo: Av. Belgrano (s) 1912.	string
6	telefono	Número telefónico del estudiante.	string
7	email	Dirección de correo electrónico del estudiante. Ejemplo: estudiante@gmail.com	string
8	nivelHabilidad	Nivel de las habilidades del estudiante.	{"Alto", "Bajo", "Medio"}
9	tipoComprensionFS	Tipo de comprensión de la información, según el modelo de Felder & Silverman.	{"Global", "Secuencial"}
10	tipoEntradaFS	Tipo de entrada de la información, según el modelo de Felder & Silverman.	{"Visual", "Verbal"}
11	tipoPercepcionFS	Tipo de percepción de la información, según el modelo de Felder & Silverman.	{"Sensitivo", "Intuitivo"}
12	tipoProcesamientoFS	Tipo de procesamiento de la información, según el modelo de Felder & Silverman.	{"Activo", "Reflexivo"}
13	descripcion	Nombre o descripción textual de un objeto. Usada para describir ubicaciones, actividades y entidades computacionales. Ejemplo1: Aula 23, FCEyT. Ejemplo2: Clase práctica de Simulación-LSI.	string
14	coordenadasGPS	Coordenadas GPS de la ubicación, en formato decimal. Ejemplo: (40,420088; -3,688810).	string
15	nivelRuido	Nivel de ruido en una ubicación.	{"Muy alto", "Alto", "Adecuado"}
16	nivelIluminacion	Nivel de iluminación en una ubicación interior.	{"Bajo", "Adecuado"}
17	horaInicioActividad	Hora de inicio de una actividad. Ejemplo: 20-02-2014, 18:30:00 UTC-3.	string
18	horaFinActividad	Hora de finalización de una actividad. Ejemplo: 20-02-2014, 18:30:00 UTC-3.	string

## II.2.5. DEFINICIÓN DE LAS CARACTERÍSTICAS DE LAS PROPIEDADES

En este paso se definieron las características de las propiedades de objetos y datos descriptas anteriormente. También se especificaron las propiedades inversas en los casos que correspondía.

En las tablas II.4 y II.5 se enuncian las características de las propiedades de objetos y de las propiedades de datos, respectivamente.

**Tabla II.4:** Características de las propiedades de objetos de *onto-alu*.

Nº	Nombre	Características	Inversa
1	tieneIdentificacion	Funcional	-
2	tieneHabilidad	Funcional inversa	esHabilidadDe

3	tieneHabilidadAutoevaluacion	Funcional	-
4	tieneHabilidadBusquedaAyuda	Funcional	-
5	tieneHabilidadPensamientoCritico	Funcional	-
6	tieneHabilidadRazonamientoCientifico	Funcional	-
7	tieneHabilidadResolucionProblemas	Funcional	-
8	esHabilidadDe	Funcional	tieneHabilidad
9	tieneEstiloAprendizajeFS	Funcional	esEstiloAprendizajeFSDe
10	esEstiloAprendizajeFSDe	Funcional	tieneEstiloAprendizajeFS
11	tieneRendimientoCurso	Funcional	-
12	tieneConocimientoActual	Funcional	-
13	ubicadoEn	Funcional	contieneA
14	contieneA	Funcional inversa	ubicadoEn
15	participaDe	Funcional Funcional inversa	tieneParticipante
16	tieneParticipante	Funcional Funcional inversa	participaDe
17	utiliza	Funcional inversa	esUtilizadaPor
18	esUtilizadaPor	Funcional	utiliza
19	seDesarrollaEn	Funcional	tieneEnDesarrollo
20	tieneEnDesarrollo	Funcional inversa	seDesarrollaEn
21	operaEn	Funcional	tieneOperandoA
22	tieneOperandoA	Funcional inversa	operaEn

**Tabla II.5:** Características de las propiedades de datos de *onto-alu*.

Nº	Nombre	Características
1	nombre	Funcional
2	dni	Funcional
3	sexo	Funcional
4	fechaNacimiento	Funcional
5	direccion	Funcional
6	telefono	Funcional
7	email	Funcional
8	nivelHabilidad	Funcional
9	tipoCompresionFS	Funcional
10	tipoEntradaFS	Funcional
11	tipoPercepcionFS	Funcional
12	tipoProcesamientoFS	Funcional
13	descripcion	Funcional
14	coordenadasGPS	Funcional
15	nivelRuido	Funcional
16	nivelIluminacion	Funcional
17	horaInicioActividad	Funcional
18	horaFinActividad	Funcional

## II.2.6. DEFINICIÓN DE AXIOMAS FORMALES Y DESCRIPCIÓN DE CLASES

El sexto paso tenía por objetivo completar la definición de las clases, especificando sus condiciones suficientes y necesarias.

Para representar las mismas, se utilizó la sintaxis de Manchester<sup>1</sup>, que es la notación utilizada en la interfaz gráfica de Protégé. En la misma, las restricciones existenciales se representan colocando primero la propiedad sobre la que se aplican, luego la palabra clave *some* y por último la clase de individuos con la que se relaciona la propiedad; las restricciones de cardinalidad cualificadas se definen indicando la propiedad, luego la palabra clave *exactly* seguida del número de relaciones permitido, y por último la clase de individuos con la que se relaciona la propiedad; y las operaciones de conjuntos sobre las clases se indican con la palabra clave *and* si son disyunciones, *or* si son uniones y *not* si son negaciones.

Las condiciones necesarias y suficientes de cada clase se muestran en las tablas II.6 y II.7, respectivamente.

**Tabla II.6:** Condiciones necesarias de las clases de *onto-alu*.

Nº	CLASE	Condiciones necesarias (subclases)
1	Estudiante	-
2	DatosPersonales	-
3	Identificacion	nombre exactly 1 dni exactly 1 sexo exactly 1 fechaNacimiento exactly 1 email exactly 1 direccion exactly 1 telefono exactly 1
4	EstiloAprendizaje	tipoComprensionFS exactly 1 tipoEntradaFS exactly 1 tipoPercepcionFS exactly 1 tipoProcesamientoFS exactly 1
5	Habilidad	nivelHabilidad exactly 1
6	HabilidadAutoevaluacion	-
7	HabilidadBusquedaAyuda	-
8	HabilidadPensamientoCritico	-
9	HabilidadRazonamientoCientifico	-
10	HabilidadResolucionProblemas	-
11	ConocimientoActual	-
12	RendimientoCurso	-
13	DatosContexto	-

<sup>1</sup> Esta notación permite representar de forma más compacta e intuitiva los distintos componentes de una ontología representada en lenguaje OWL.

14	Ubicación	descripcion exactly 1 coordenadasGPS exactly 1 dirección exactly 1
15	UbicacionExterior	-
16	UbicacionInterior	niveIluminacion exactly 1 nivelRuido exactly 1
17	Aula	-
18	Box	-
19	Laboratorio	-
20	Actividad	descripcion exactly 1 horaInicioActividad exactly 1 horaFinActividad exactly 1
21	ActividadFormal	-
22	Clase	-
23	Evaluacion	-
24	ActividadInformal	-
25	EntidadComputacional	descripcion exactly 1
26	Aplicación	-
27	Dispositivo	-
28	Impresora	-
29	PC	-
30	Proyector	-
31	Smartphone	-
32	Tablet	-
33	Red	-

**Tabla II.7:** Condiciones suficientes de las clases de *onto-alu*.

Nº	CLASE	Condiciones suficientes (clases equivalentes)
1	Estudiante	tieneIdentificacion exactly 1 Identificacion tieneHabilidadAutoevaluacion exactly 1 HabilidadAutoevaluacion tieneHabilidadBusquedaAyuda exactly 1 HabilidadBusquedaAyuda tieneHabilidadPensamientoCritico exactly 1 HabilidadPensamientoCritico tieneHabilidadRazonamientoCientifico exactly 1 HabilidadRazonamientoCientifico tieneHabilidadResolucionProblemas exactly 1 HabilidadResolucionProblemas tieneEstiloAprendizajeFS exactly 1 EstiloAprendizajeFS
2	DatosPersonales	EstiloAprendizaje or Habilidad or Identificacion
3	Identificacion	-
4	EstiloAprendizaje	-
5	Habilidad	HabilidadAutoevaluacion or HabilidadBusquedaAyuda or HabilidadPensamientoCritico or HabilidadRazonamientoCientifico or HabilidadResolucionProblemas

6	HabilidadAutoevaluacion	-
7	HabilidadBusquedaAyuda	-
8	HabilidadPensamientoCritico	-
9	HabilidadRazonamientoCientifico	-
10	HabilidadResolucionProblemas	-
11	ConocimientoActual	-
12	RendimientoCurso	-
13	DatosContexto	Actividad or EntidadComputacional or Ubicacion
14	Ubicación	UbicacionExterior or UbicacionInterior contieneA some Estudiante tieneEnDesarrolloA some Actividad tieneOperandoA some EntidadComputacional
15	UbicacionExterior	-
16	UbicacionInterior	nivelIluminacion exactly 1 nivelRuido exactly 1
17	Aula	-
18	Box	-
19	Laboratorio	-
20	Actividad	ActividadFormal or ActividadInformal seDesarrollaEn exactly 1 Ubicacion tieneParticipante some Estudiante
21	ActividadFormal	-
22	Clase	-
23	Evaluacion	-
24	ActividadInformal	-
25	EntidadComputacional	Aplicacion or Dispositivo or Red esUtilizadaPor exactly 1 Estudiante operaEn exactly 1 Ubicacion
26	Aplicación	-
27	Dispositivo	-
28	Impresora	-
29	PC	-
30	Proyector	-
31	Smartphone	-
32	Tablet	-
33	Red	-

### II.3. IMPLEMENTACIÓN CON PROTEGE

En esta sección se presenta la implementación de la ontología *onto-alu* en lenguaje OWL-DL, con el uso de la herramienta Protégé-OWL, versión 4.3. Se describe detalladamente cada uno de los pasos realizados, con el fin de que el presente trabajo sirva como elemento de consulta para otros estudiantes que se inicien en el diseño y creación de ontologías con dicha herramienta.

### II.3.1. CREACIÓN DE LA ONTOLOGÍA

El primer paso, luego de iniciar la herramienta Protégé, fue crear la nueva ontología y proporcionar la IRI para identificarla unívocamente (figura II.2). Para guardar el archivo se eligió la serialización OWL/XML, como se puede observar en la figura II.3.

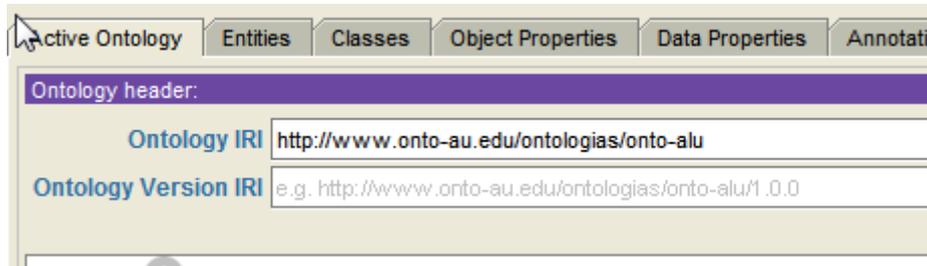


Figura II.2: Indicar IRI de la nueva ontología.

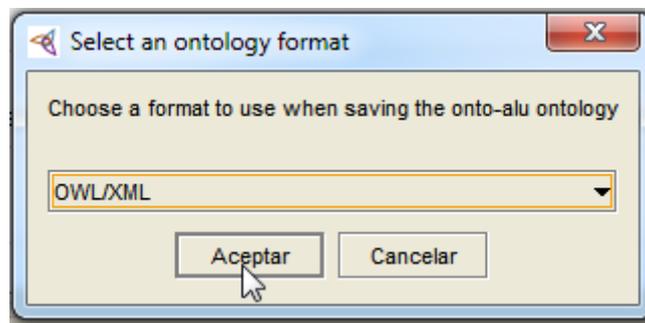


Figura II.3: Guardar la ontología serializada como OWL/XML en un archivo.

También se crearon valores para las siguientes propiedades de anotación: *creator*, *title*, *publisher* y *date*, como se muestra en la figura II.4.

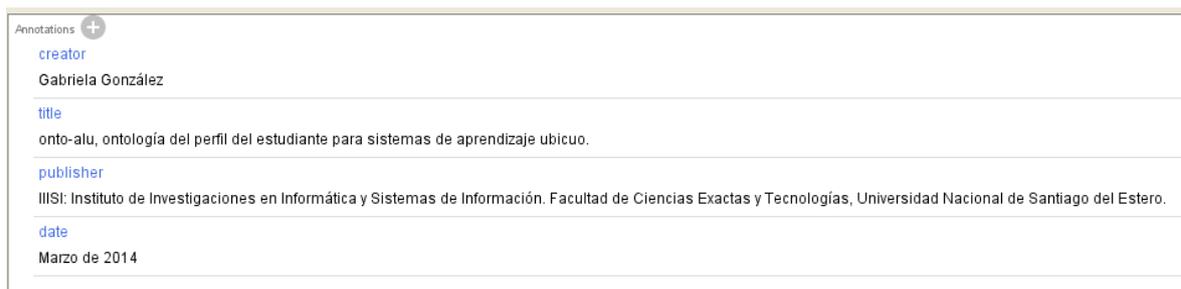
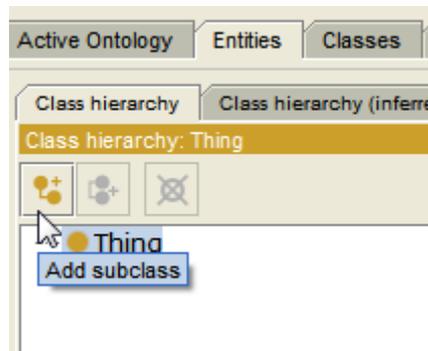


Figura II.4: Propiedades de anotación de *onto-alu*.

### II.3.2. DEFINICIÓN DE CLASES

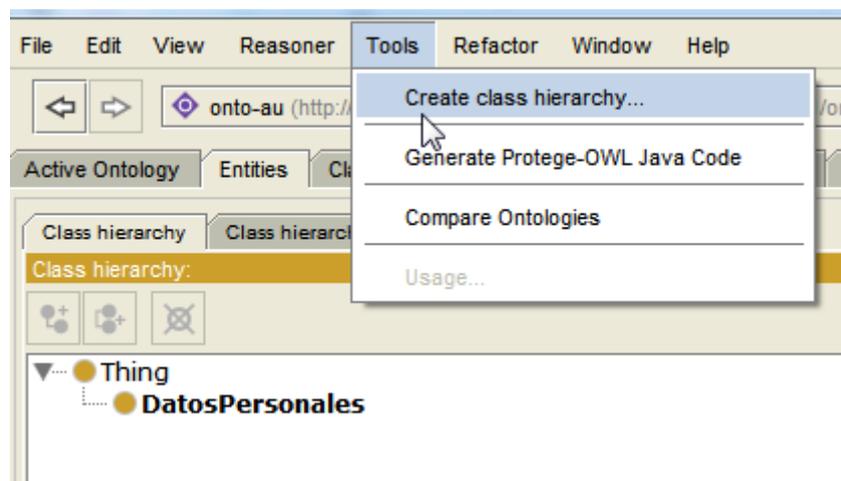
En segundo lugar, se procedió a la creación de las distintas clases de la ontología de acuerdo a la jerarquía mostrada anteriormente en la figura II.1.

En Protégé, hay dos formas de crear nuevas clases. La primera es agregarlas a la jerarquía de a una, haciendo uso de los botones *Add subclass* y *Add sibling class* de la pestaña *Class hierarchy* (figura II.5).



**Figura II.5:** Agregar una subclase de la clase Thing.

El segundo método es más eficiente y consiste en hacer uso de la opción del menú principal *Tools > Create class hierarchy* y crear una jerarquía completa de clases, indicando los nombres y niveles en un único cuadro de texto. Esta opción se muestra en la figura II.6.



**Figura II.6:** Crear jerarquía de clases.

En el primer cuadro de diálogo, figura II.7, se indicó la clase base a partir de la cual se crearía la jerarquía, en el segundo, figura II.8, se tipearon los nombres de las clases deseadas, indicando las subclases con un nivel de indentación mayor.

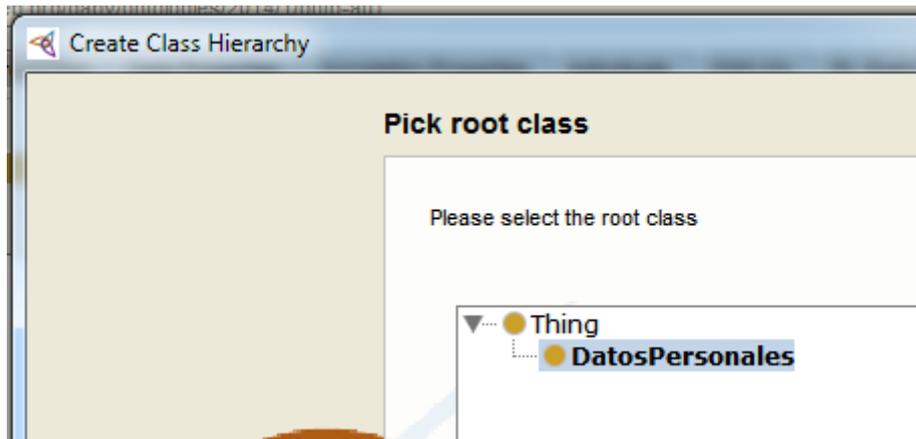


Figura II.7: Indicar la clase base de la jerarquía.

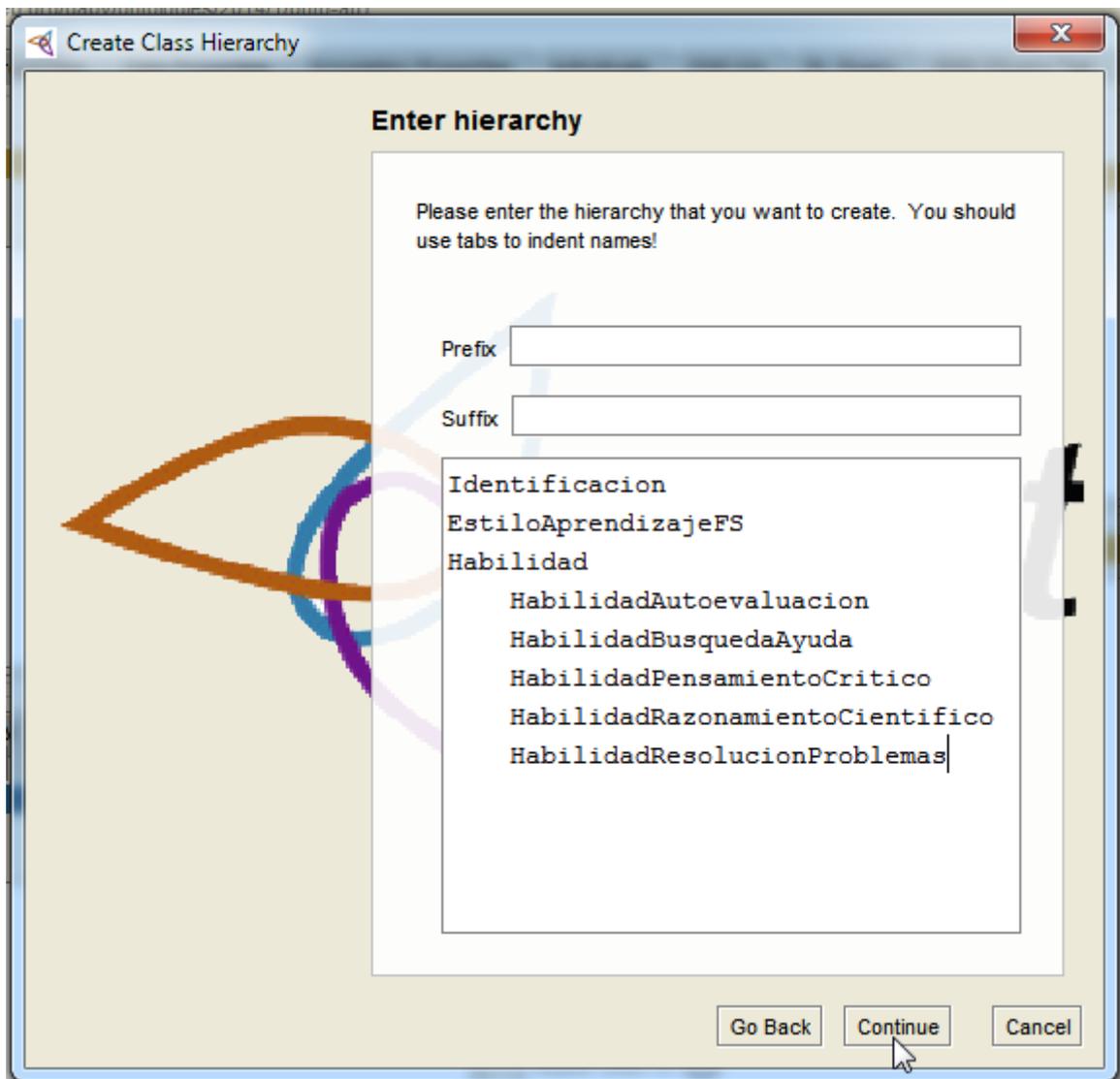
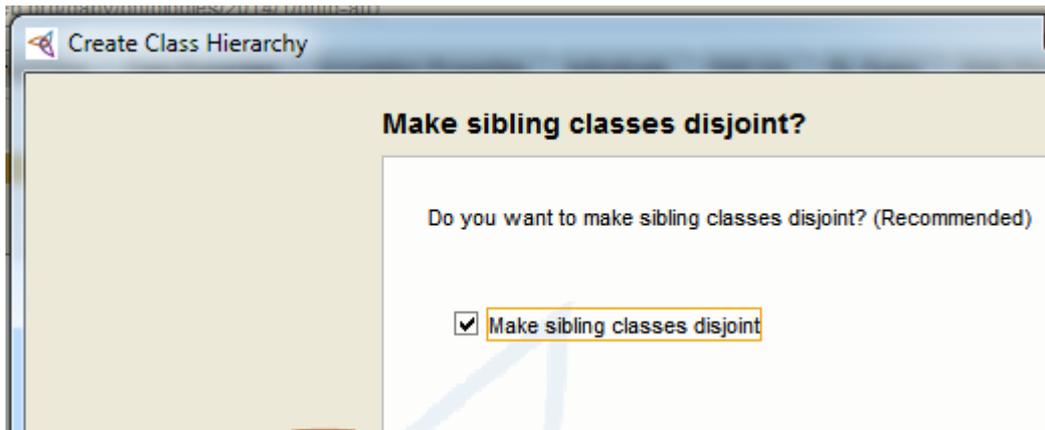


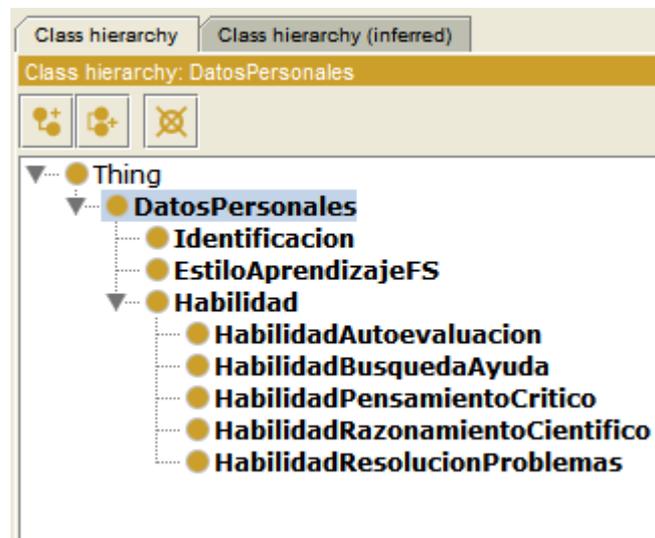
Figura II.8: Ingresar la jerarquía de clases a crear.

El último paso consistió en indicar si se querían definir las clases hermanas de cada nivel como disjuntas entre sí (Figura II.9).



**Figura II.9:** Opción para indicar clases hermanas de cada nivel como disjuntas.

La jerarquía parcial de clases, luego de crear las clases descritas anteriormente, se muestra a continuación en la figura II.10.



**Figura II.10:** Jerarquía parcial de clases de *onto-alu*.

Del mismo modo, se crearon todas las clases restantes, obteniendo la jerarquía completa tal como se observa en la figura II.11.

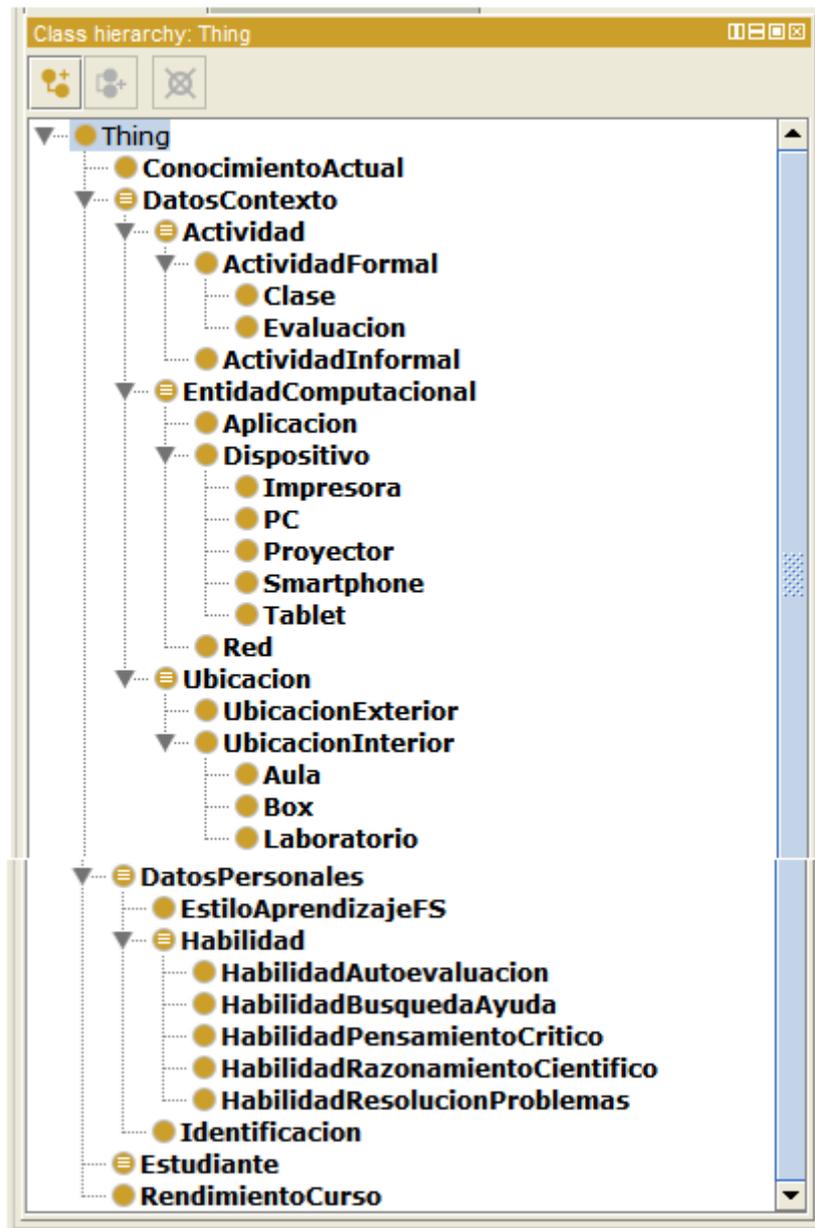


Figura II.11: Jerarquía completa de clases de *onto-alu*.

### II.3.3. DEFINICIÓN DE PROPIEDADES DE OBJETOS Y DE DATOS

Una vez que se completó la creación de la jerarquía de clases, se procedió a crear las propiedades de objetos de la ontología.

Para ello, se utilizó el botón *Add subproperty* de la pestaña *Object Properties* (figura II.12). Luego se ingresó el nombre de la propiedad en el cuadro de diálogo, como se puede observar en la figura II.13.

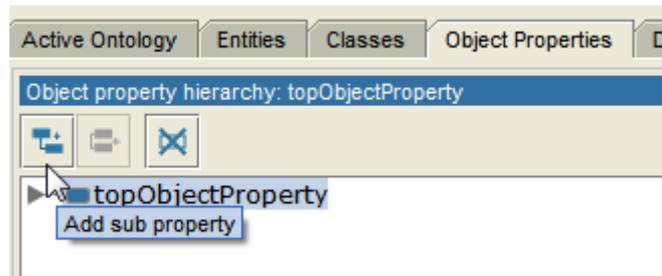


Figura II.12: Agregar una propiedad de objetos.

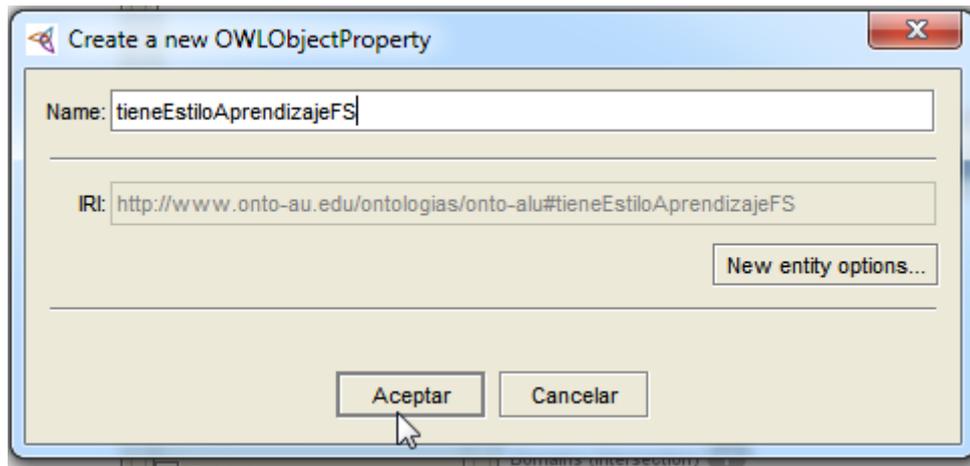


Figura II.13: Ingresar nombre de la propiedad de objetos.

También se creó una descripción de la propiedad haciendo uso de la anotación *comment* (figura II.14).

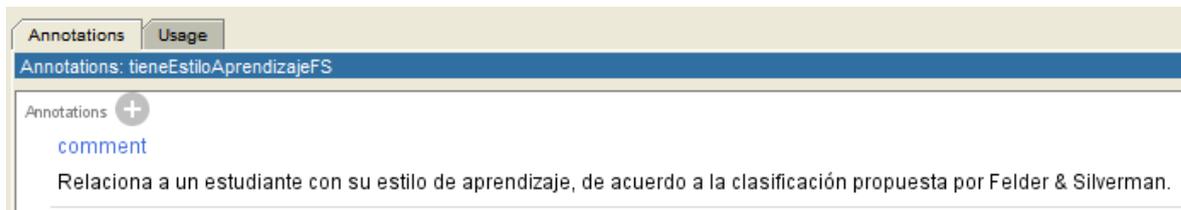
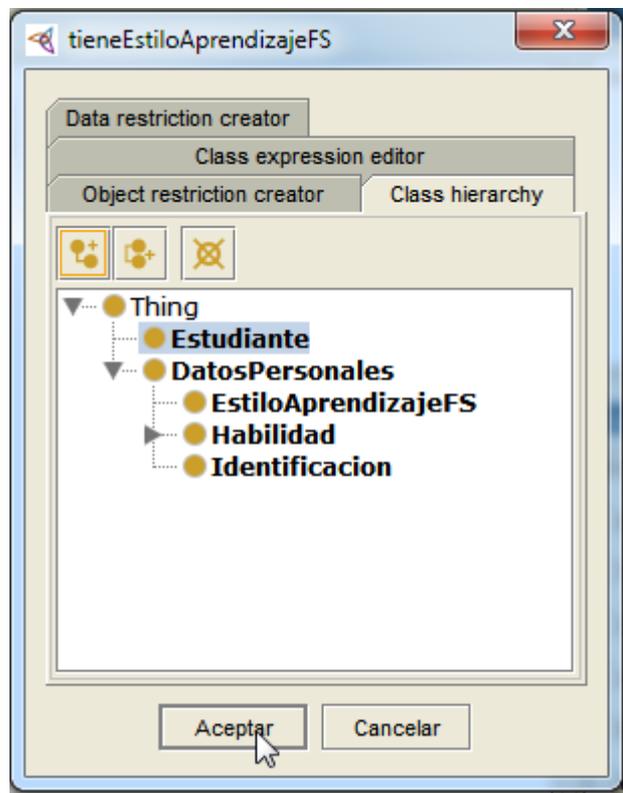


Figura II.14: Anotación para la propiedad tieneEstiloAprendizajeFS

Como paso siguiente, se indicaron el dominio (figuras II.15 y II.16) y rango (figuras II.17 y II.18) de las propiedades.



**Figura II.15:** Agregar dominio a la propiedad tieneEstiloAprendizajeFS.



**Figura II.16:** Indicar dominio de tieneEstiloAprendizajeFS.

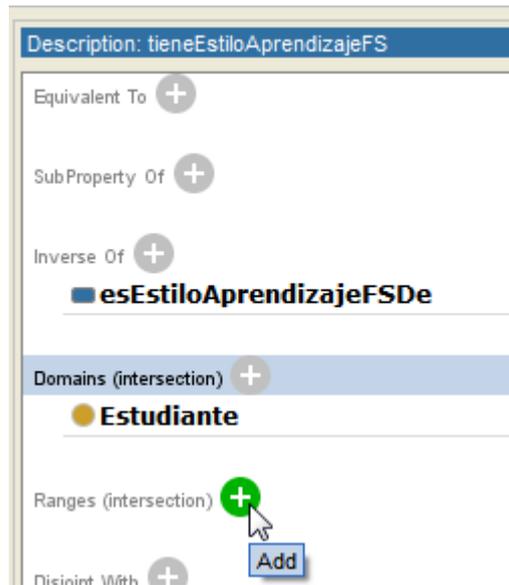


Figura II.17: Agregar rango a la propiedad tieneEstiloAprendizajeFS.

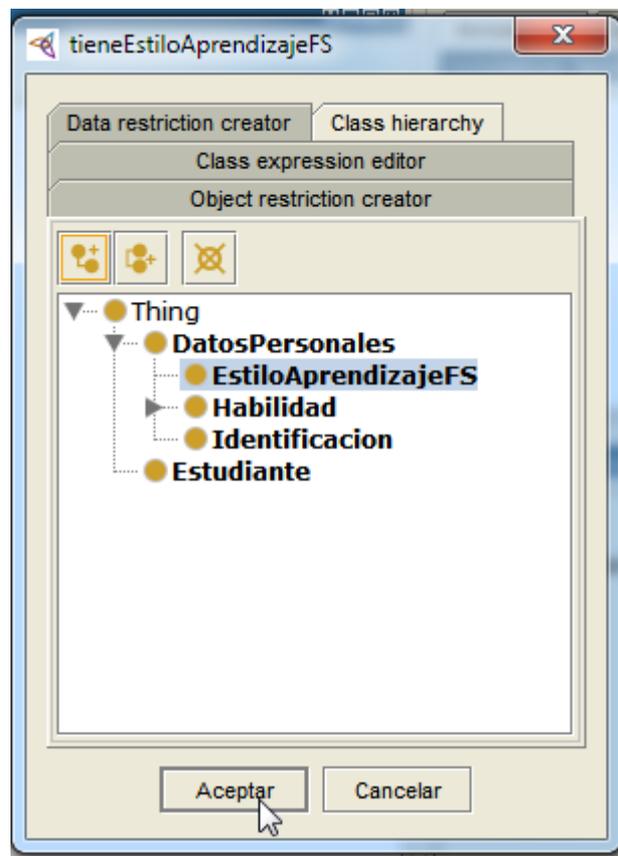


Figura II.18: Indicar rango de tieneEstiloAprendizajeFS.

Luego, la propiedad inversa (figuras II.19 y II.20), y las características, tal como se puede observar en la figura II.21.

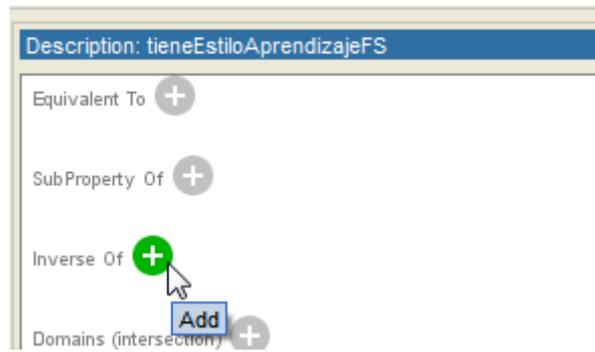


Figura II.19: Agregar una propiedad inversa.

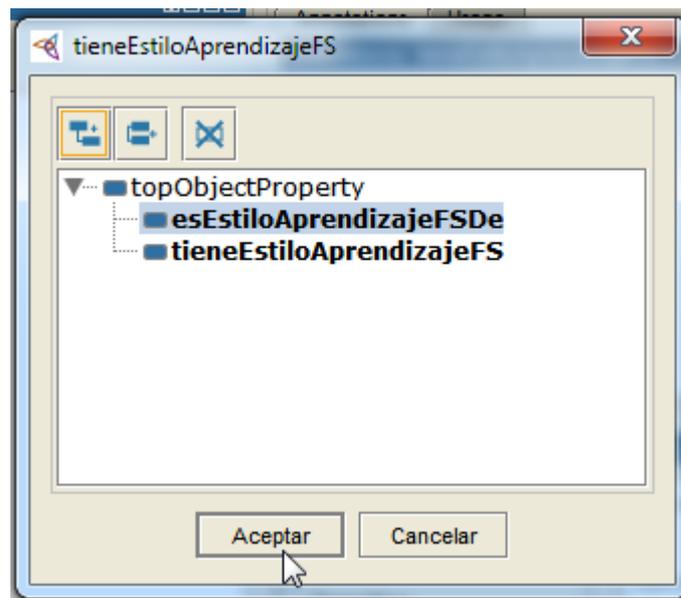


Figura II.20: Indicar la propiedad inversa para tieneEstiloAprendizajeFS.

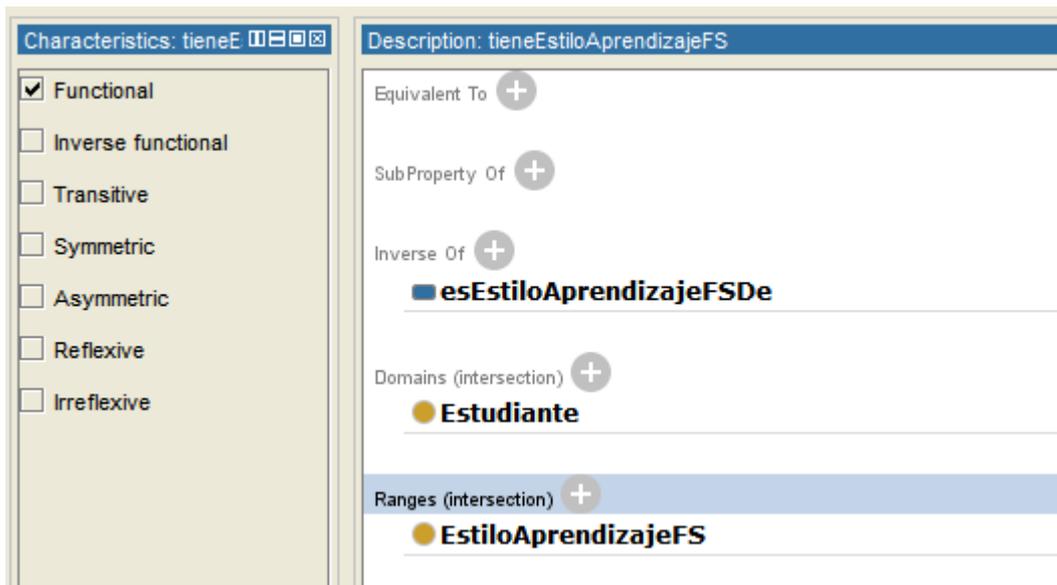
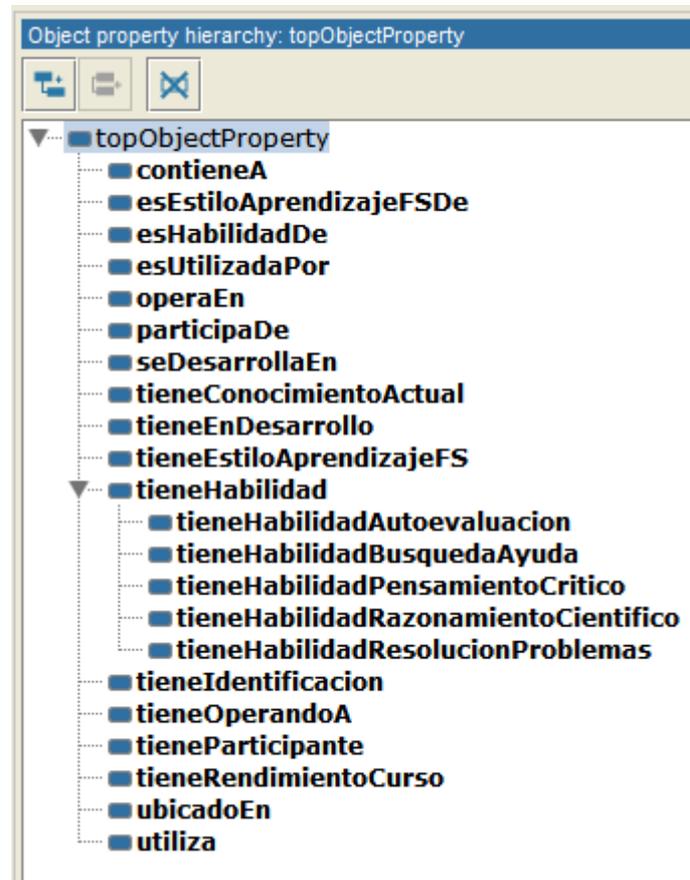


Figura II.21: Propiedad tieneEstiloAprendizajeDe con todos sus detalles definidos.

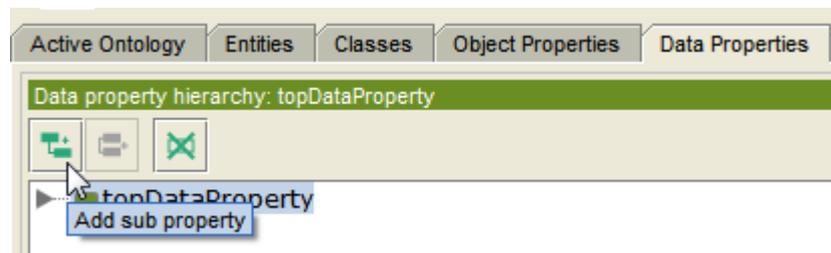
Del mismo modo, se crearon todas las propiedades de objetos restantes definidos en la tabla II.4, con sus correspondientes detalles, de acuerdo a lo descrito en la tabla II.6. Las mismas se pueden observar en la figura II.22.



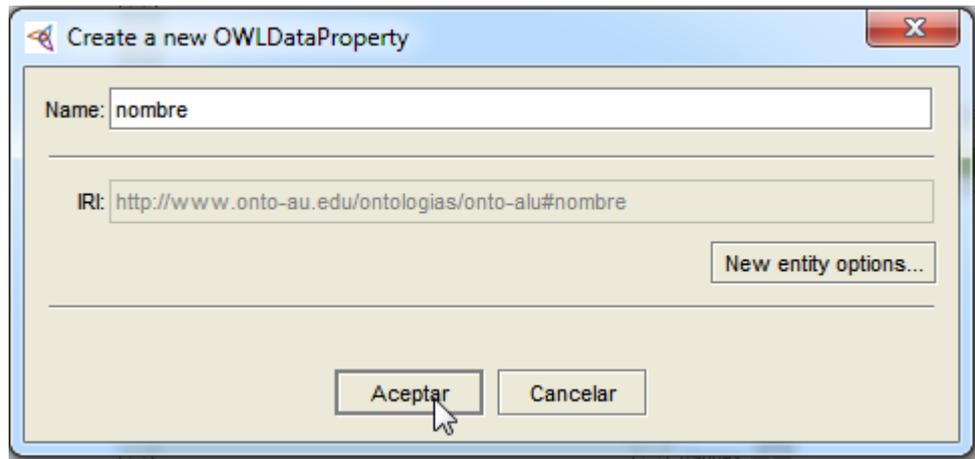
**Figura II.22:** Jerarquía completa de las propiedades de objetos de *onto-alu*.

Una vez que se completó la definición de las propiedades de objetos, se procedió a crear las propiedades de datos de la ontología.

Para ello, se utilizó el botón *Add subproperty* de la pestaña *Data Properties*, como se muestra en la figura II.23. Luego se ingresó el nombre de la propiedad en el cuadro de diálogo (figura II.24).

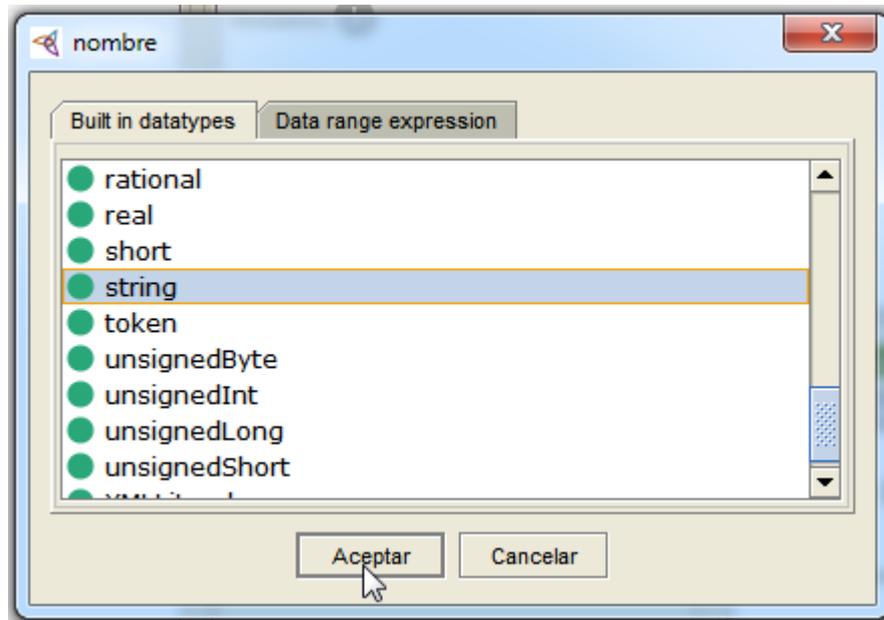


**Figura II.23:** Creación de una propiedad de datos.

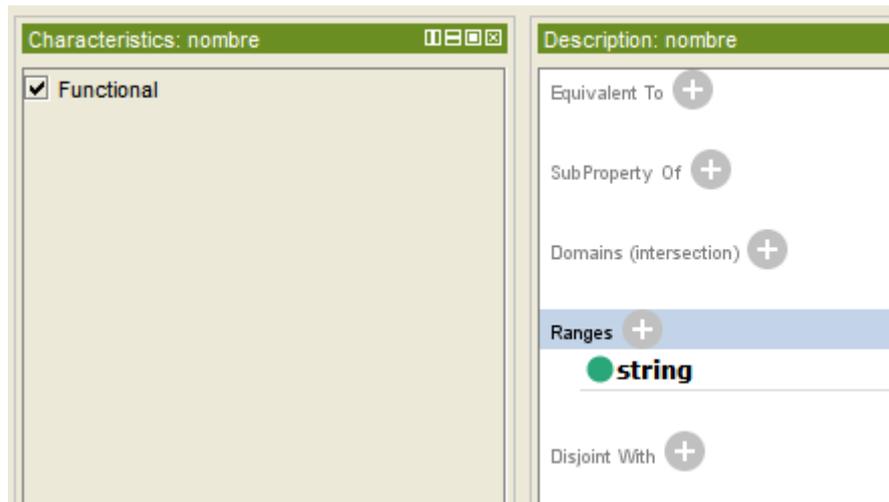


**Figura II.24:** Ingresar nombre para la propiedad de datos.

Como paso siguiente, se indicaron el rango, figura II.25, y características de la propiedad, II.26.

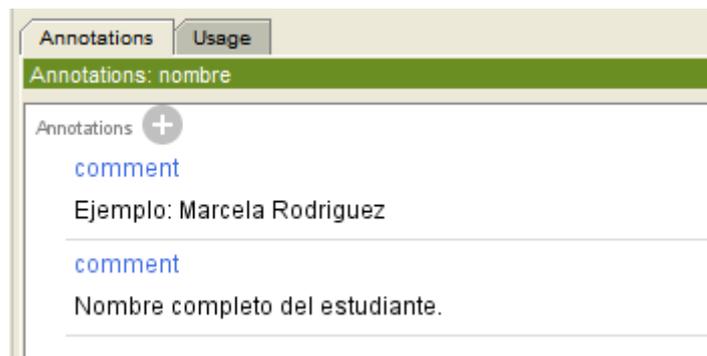


**Figura II.25:** Indicar rango para propiedad de datos.



**Figura II.26:** Todos los detalles de la propiedad de datos.

También se incluyeron comentarios en la propiedad, para especificar su descripción y un ejemplo del valor que puede tomar. Esto se muestra en la figura II.27.



**Figura II.27:** Anotaciones sobre la propiedad de datos.

Del mismo modo, se crearon las propiedades de datos restantes, de acuerdo a lo definido en las tablas II.5 y II.7. La jerarquía completa de datos en Protégé se muestra a continuación, en la figura II.28.



Figura II.28: Jerarquía completa de propiedades de datos de *onto-alu*.

#### II.3.4. DEFINICIÓN DE AXIOMAS FORMALES Y DESCRIPCIÓN DE CLASES

En este último paso se completaron las especificaciones de las clases al definir sus condiciones necesarias y suficientes, de acuerdo a lo descrito en las tablas II.8 y II.9.

Primero se definieron las condiciones necesarias de las clases, haciendo uso del botón *Add* de la sección *subclass* de la clase a describir, como se puede observar en la figura II.29.

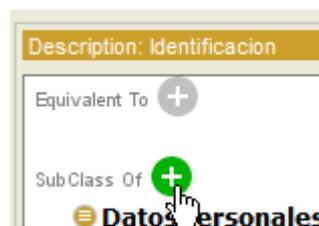
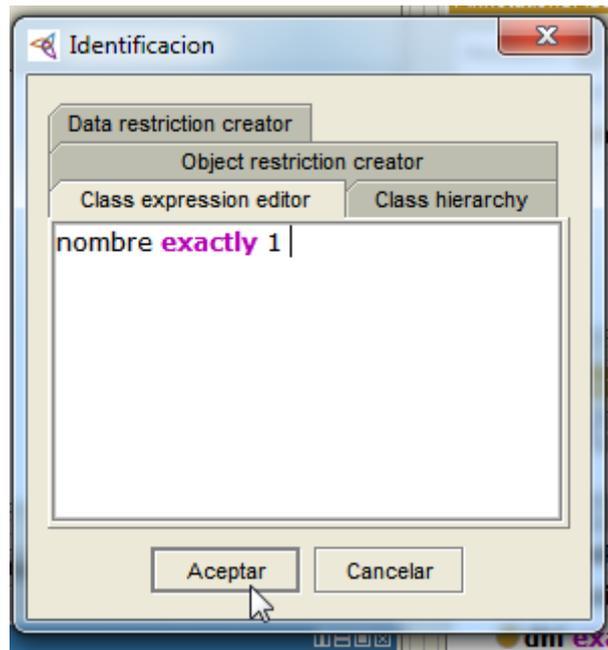


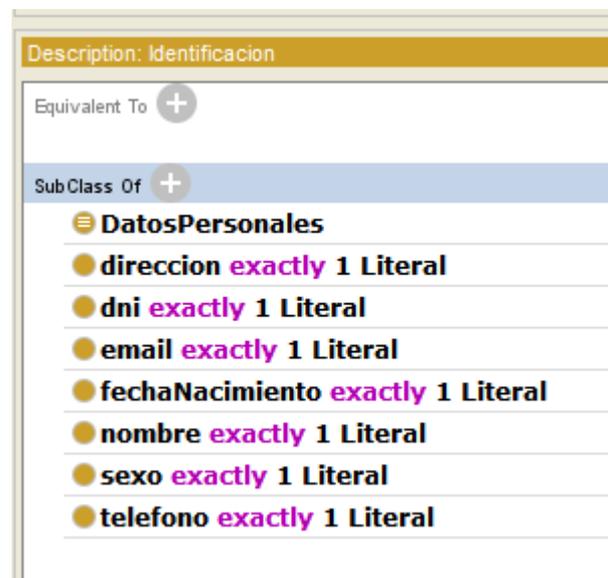
Figura II.29: Agregar una nueva condición suficiente.

Luego, en el cuadro de texto de la pestaña *Class expression editor* se agregó la restricción correspondiente (figura II.30).



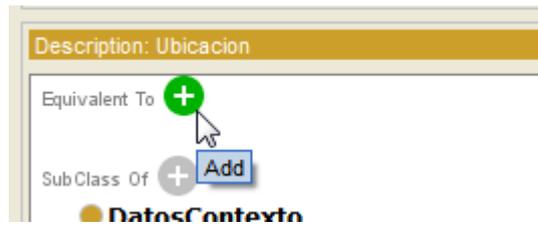
**Figura II.30:** Restricción de cardinalidad sobre propiedad de datos.

Del mismo modo se agregaron las demás restricciones de cada clase. A continuación se muestran, en la figura II.31, todas las condiciones necesarias definidas para una de las clases de la ontología.



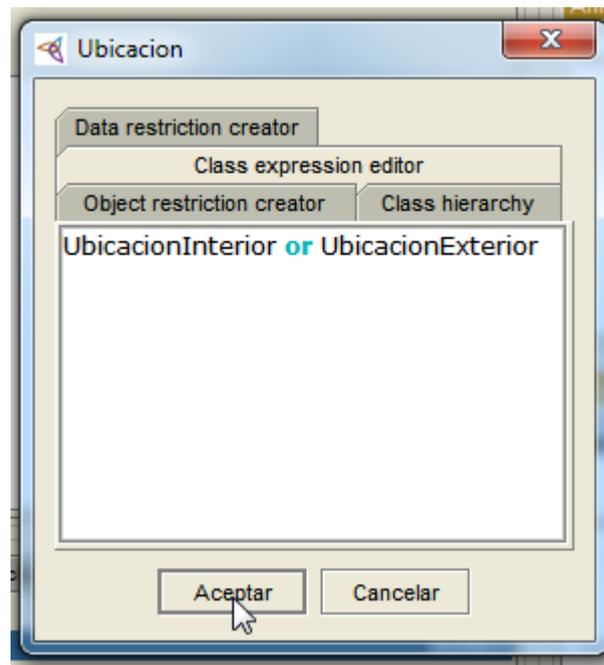
**Figura II.31:** Todas las condiciones necesarias de la clase Identificacion.

Una vez que se definieron todas las condiciones necesarias de cada clase, se procedió a describir las condiciones suficientes de las mismas, haciendo uso del botón *Add* de la sección *Equivalent to* de la clase a definir (figura II.32).



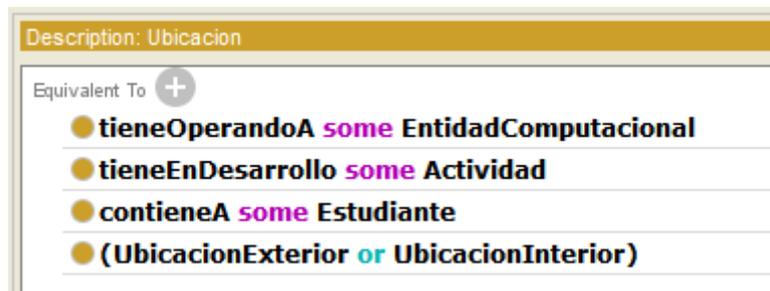
**Figura II.32:** Agregar una condición suficiente.

Al igual que para las condiciones necesarias, se describió la restricción u operación de conjuntos en la ventana del editor, de la pestaña *Class expression editor*, como se observa en la figura II.33.



**Figura II.33:** Unión de clases.

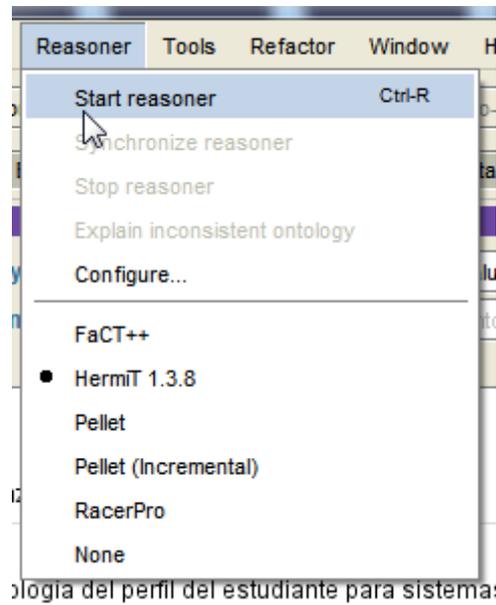
Del mismo modo, se agregaron las descripciones restantes para cada clase. En la figura II.34 se muestran todas las condiciones suficientes de la clase Ubicacion.



**Figura II.34:** Todas las condiciones suficientes de la clase Ubicación.

Con este último paso se completó la definición de la ontología.

El primer chequeo que se hizo una vez construida, fue la ejecución del razonador (figura II.35) para tener la seguridad de que no se habían cometido errores, durante la construcción, que generaran inconsistencias.



**Figura II.35:** Ejecución del razonador en *onto-alu*.

Luego se procedió a la evaluación de la ontología a partir de distintos criterios, lo cual se describe en el capítulo siguiente.

# CAPÍTULO III

## EVALUACIÓN DE LA ONTOLOGÍA DEL ESTUDIANTE

---

En el presente capítulo se describe el proceso de evaluación de la ontología del estudiante *onto-alu*.

En la primera sección se documenta la evaluación de la misma desde un punto de vista de diseño, de acuerdo al marco descrito en la sección I.2.2 del capítulo I. En la segunda sección, se presenta la verificación del cumplimiento, por parte de la ontología, de los requerimientos descritos para su dominio. Finalmente, en la última sección, se describe la evaluación de la ontología en un caso de aplicación concreto.

### III.1. EVALUACIÓN DE ERRORES TAXONÓMICOS Y ANOMALÍAS DE DISEÑO

En el capítulo I, sección I.2.2 se presentó el marco de referencia, propuesto por Gómez-Pérez *et al.* (2004) y ampliado por Fahad & Qadir (2008), para la evaluación de ontologías con respecto a principios de diseño. A continuación, se describe la aplicación de estos criterios a la ontología creada.

#### III.1.1. ERRORES DE INCONSISTENCIA

Como primera medida se buscaron errores de inconsistencia que pueden ser de tres tipos: errores de circularidad, errores en particiones y errores semánticos. Las primeras dos categorías, son validadas automáticamente al ejecutar un razonador-DL sobre la ontología.

Para la evaluación de *onto-alu*, se utilizaron dos de los razonadores disponibles en Protégé: HermiT en su versión 1.3.8, (figura III.1) el cual forma parte de la instalación por defecto del software, y Pellet, versión 2.3.1, disponible como plugin descargable.

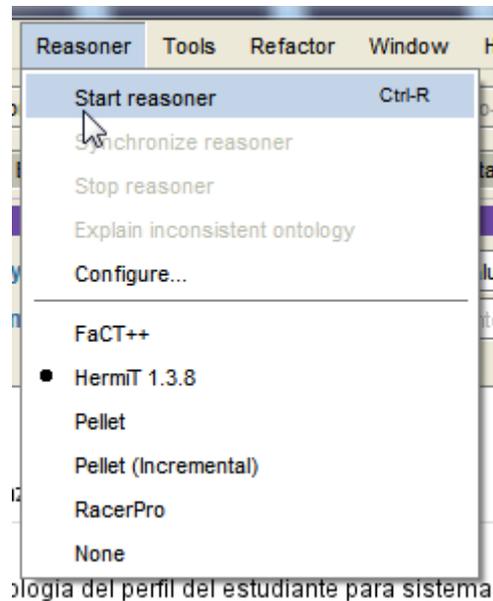


Figura III.1: Ejecución del razonador HermiT sobre *onto-alu*.

Tanto HermiT como Pellet se consideran razonadores sólidos y completos, capaces de representar el subconjunto SROIQ(D) de lógica descriptiva, y que ofrecen soporte para reglas SWRL, justificaciones de inferencias y razonamiento sobre individuos (ABox) (Sirin *et al.*, 2007; Shearer *et al.*, 2008).

Ambos proveen los siguientes servicios:

- **Chequeo de consistencia:** asegurar que la ontología no contiene hechos contradictorios.
- **Satisfacibilidad de conceptos:** detectar, para cada clase, si es posible que la misma contenga instancias.
- **Clasificación:** computar las relaciones de subclases entre todas las clases existentes, para crear la jerarquía de clases completa (inferida).
- **Comprensión:** computar los tipos de (a que clase pertenece) cada uno de los individuos.

La diferencia entre HermiT y Pellet radica, principalmente, en el algoritmo de razonamiento utilizado. Las reglas de derivación y la estrategia de bloqueo de cada uno, son distintas. En la práctica, esto significa que cada razonador puede generar diferentes inferencias para determinados casos (Shearer *et al.*, 2008).

Al clasificar la ontología *onto-alu* con dos razonadores metodológicamente diferentes, se buscó aumentar el porcentaje de detección de errores, así como también favorecer la obtención de todas las inferencias (correctas) posibles.

En ambos casos, la ontología clasificada resultó consistente. Es decir que no se detectó ningún error de circularidad o de particiones, bajo ninguno de los razonadores utilizados. Esto se muestra en la figura III.2.

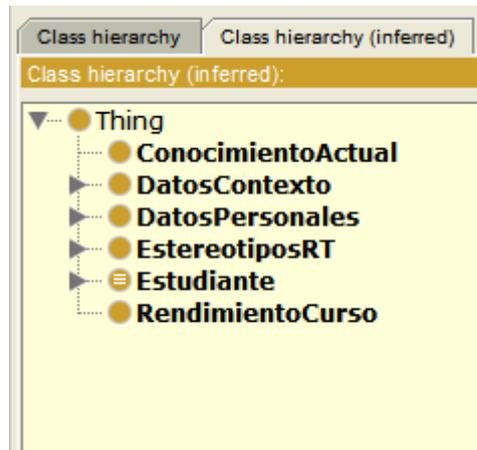


Figura III.2: Vista parcial de la jerarquía de clases inferida.

Si la ontología resultara inconsistente, las clases involucradas en el error se mostrarían de color rojo. Para ejemplificar esta situación, se introdujo una instancia común en la partición de la clase *Actividad*, perteneciente al conjunto de datos de contexto. *Actividad Formal* y *Actividad Informal* representan una partición de la clase *Actividad* porque toda *Actividad* es *Formal* o bien *Informal* (figura III.3), y ambos tipos son mutuamente excluyentes (figura III.4).

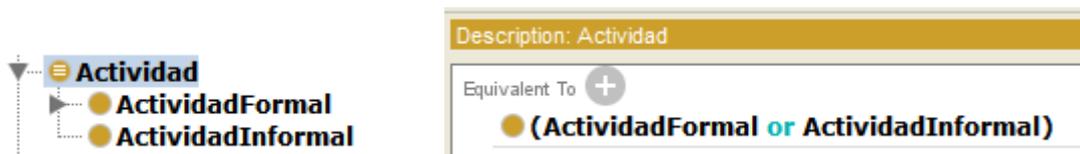
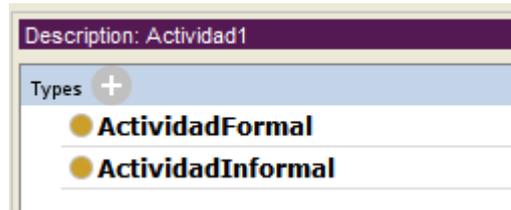


Figura III.3: Una *Actividad* es *Formal* o bien *Informal*.



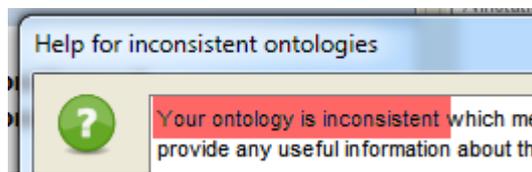
Figura III.4: Una *Actividad Formal* no puede tener instancias en común con una *Actividad Informal*.

En la figura III.5 se puede observar la instancia Actividad1 creada, y definida como perteneciente a dos clases disjuntas.

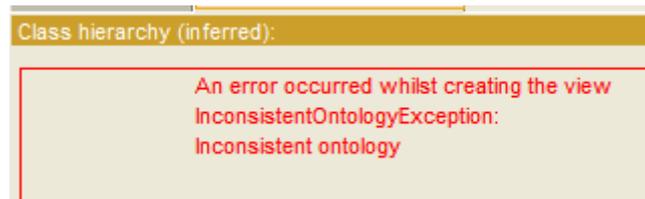


**Figura III.5:** Actividad1 es de tipo Actividad Formal y Actividad Informal.

Luego de introducir la nueva instancia se clasificó la ontología, obteniendo el resultado mostrado en las figuras III.6 y III.7.

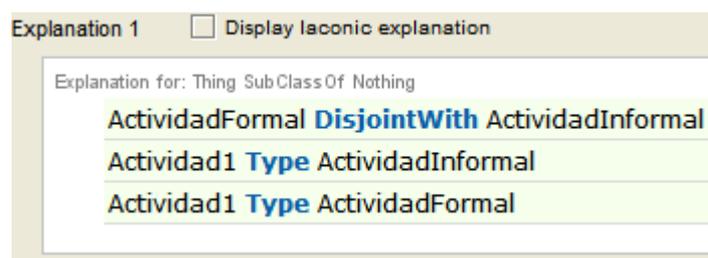


**Figura III.6:** El razonador indica que la ontología es inconsistente.



**Figura III.7:** No se puede crear la jerarquía inferida de clases debido a la inconsistencia.

La justificación que explica porque la ontología es inconsistente se muestra a continuación, en la figura III.8:



**Figura III.8:** Actividad tiene dos tipos, que son disjuntos entre sí.

En segunda instancia, se procedió a analizar los errores de inconsistencia semánticos.

Para comprobar esta clase de errores, se revisó que todos los conceptos que son subclase de algún concepto, se consideren, desde el punto de vista semántico, una especialización del concepto general, y posean un dominio más reducido que el mismo. Por ejemplo, la clase Actividad tiene como subclases a ActividadFormal y ActividadInformal. Una actividad formal es más específica que una actividad, con lo cual

se puede asegurar que no hay actividad formal que no sea también una actividad. Asimismo, no existen características propias de una actividad formal que contradigan a las características de nivel superior que se podrían definir para una Actividad.

### **III.1.2. ERRORES DE INCOMPLETITUD**

La siguiente categoría de errores a considerar, es la de errores de incompletitud. Estos errores pueden ser de cuatro tipos: clasificación incompleta, errores en particiones (omisión de axiomas de disyunción y omisión de axiomas de exhaustividad), omisión de característica funcional o funcional-inversa en propiedades y omisión de condiciones suficientes.

Con respecto a los errores de clasificación incompleta, se puede argumentar que este criterio es sumamente subjetivo y muy dependiente del dominio del problema, en relación al nivel de granularidad en la descomposición de conceptos en función del interés y/o utilidad de los mismos en la aplicación de la ontología.

En el caso de *onto-alu* se considera que el nivel de desagregación de conceptos es suficiente para el dominio que se pretende representar, teniendo en cuenta también la factibilidad de adquisición de la información necesaria para clasificar a cada concepto. Por ejemplo, se podría argumentar que así como las ubicaciones internas se subdividen en Aula, Laboratorio y Box, las externas se podrían subdividir en Calle y Casa. Si bien esta diferenciación podría resultar útil con fines de personalización, ¿Cómo se obtendría la misma? Debería conocerse un medio o método a través del cual se pueda determinar de forma fiable (y efectiva en términos de costo) cuando una ubicación externa es la casa del alumno y cuando es la calle.

Si, por el contrario, no hay forma de obtener esta información, entonces este nivel adicional de desagregación solo agrega complejidad sin proporcionar ninguna ventaja, razón por la cual es preferible excluirlo del modelo.

Para los errores en particiones y de omisión de característica funcional o inversa-funcional en propiedades se comprobó que: todas las clases hermanas en cada nivel sean disjuntas, todas las particiones contengan el axioma de exhaustividad, todas las propiedades que sólo pueden relacionar una instancia del dominio con una única instancia del rango estén indicadas como funcionales, y todas las propiedades inversas a propiedades funcionales estén indicadas como funcional-inversa.

En el modelo presentado no existen conjuntos de clases, en un mismo nivel de la jerarquía, que puedan contener instancias comunes, por ende todas las clases hermanas son disjuntas entre sí. Por ejemplo, DatosPersonales está formada por las subclases Identificación, EstiloAprendizajeFS y Habilidad, las cuales son disjuntas entre sí. Esto se debe a que si algo es una Identificación entonces no puede ser a la vez un EstiloAprendizajeFS o una Habilidad, y algo que es un EstiloAprendizajeFS no puede ser una Identificación o una Habilidad.

Con respecto a las particiones de clases (descomposiciones disjuntas exhaustivas), las mismas deben contener el axioma de exhaustividad que indica explícitamente que si un individuo pertenece a la clase base, entonces necesariamente pertenecerá a alguna de las subclases. Por ejemplo, la clase Ubicacion está formada por la partición de clases UbicacionExterior y UbicacionInterior, por lo que su axioma de exhaustividad debe indicar que Ubicacion es equivalente a: UbicacionExterior  $\circ$  UbicacionInterior. Esto se puede comprobar en la tabla II.7 donde se definen las condiciones suficientes para cada clase.

Por otra parte, la mayoría de las propiedades de objetos del modelo y todas las propiedades de datos del mismo son propiedades funcionales. Es decir que relacionan un individuo del dominio con, a lo sumo, un individuo del rango. Por ejemplo, tieneEstiloAprendizaje es funcional porque un alumno puede tener un único estilo de aprendizaje. Y por lo tanto, su propiedad inversa, esEstiloAprendizajeDe será una propiedad Funcional inversa. Esto puede comprobarse en las tablas II.4 y II.5 donde se definen las características de las propiedades de objetos y de datos.

En relación al error de incompletitud de omitir el conocimiento suficiente, en *ontolu* se utilizaron los siguientes criterios:

- La descripción necesaria de una clase se realiza en función de las propiedades de datos con las que ésta se relaciona.
- La descripción suficiente de una clase se realiza en función de las propiedades de objetos con las que ésta se relaciona.

Se adoptaron estos principios debido a que existen propiedades de datos que se reutilizan en varias clases. Por ejemplo, tanto una entidad computacional como una actividad poseen una descripción; tanto un estudiante como una ubicación poseen una dirección.

Por el contrario, las propiedades de objetos no se reutilizan sino que se crean nuevas propiedades para relacionar distintos tipos de entidades. Por ejemplo, un estudiante se

relaciona con una ubicación a través de `ubicadoEn`, mientras que una actividad se relaciona con una ubicación a través de `seDesarrollaEn`.

Se considera adecuado reusar las propiedades de datos porque representan en esencia el mismo tipo de relación. Es decir, una descripción no cambia su naturaleza por referirse a una ubicación o a una actividad, así como una dirección no cambia su naturaleza por referirse a la dirección de un estudiante o de un lugar.

Por el contrario, en el caso de las propiedades de objetos, el hecho de relacionar dos entidades diferentes puede significar un cambio en la esencia de la relación. Sería correcto decir que tanto un estudiante como un profesor están `ubicadoEn` un determinado lugar, pero decir que una actividad está `ubicadaEn` un lugar no resulta tan intuitivo como decir que una actividad `seDesarrollaEn` un lugar específico.

Si eventualmente se agregaran más clases a la ontología, existe la posibilidad de que dos clases distintas estén descritas por los mismos datos (porque algunas propiedades de datos se reusan). Si las restricciones sobre propiedades de datos se definen como condiciones suficientes de una clase, entonces se puede inferir que un individuo  $x$  pertenece a dos clases diferentes, lo cual podría llevar a una inconsistencia si las mismas son disjuntas.

Esto no puede darse en las propiedades de objetos puesto que éstas no se comparten entre distintos tipos de individuos o clases. Por ende, definir las restricciones sobre propiedades de objetos como condiciones suficientes nos permite inferir el tipo del individuo en función de las relaciones que posee con otros individuos.

Por lo tanto, se verificó que las clases de *onto-alu* que poseen restricciones sobre propiedades de objetos las definan como condiciones suficientes. Por ejemplo, `Actividad` es equivalente a: *seDesarrollaEn exactly 1 Ubicacion and tieneParticipante some Estudiante*. Las condiciones suficientes para cada clase se detallaron en la tabla II.7.

### III.1.3. ERRORES DE REDUNDANCIA

Los errores de redundancia ocurren cuando la misma información se infiere más de una vez a partir de las relaciones, clases e instancias. Durante la evaluación de *onto-alu*, se constató que cada clase posee, a lo sumo, una única clase primitiva como superclase y que cada propiedad posee, a lo sumo, una única propiedad como superpropiedad. Esto se puede evidenciar en el diagrama de jerarquías de clases y en la jerarquía de propiedades de

objetos presentados en las secciones 2.3 y 2.4 del capítulo II, respectivamente. El caso de las instancias no se considera porque la ontología no posee individuos *per se*.

Además, se verificó que no existen clases o propiedades con diferentes nombres pero igual definición, y que tampoco se encuentran clases disjuntas entre sí especificadas más de una vez.

## III.2. EVALUACIÓN CONTRA REQUERIMIENTOS

En esta sección se demuestra como la ontología es capaz de representar de forma adecuada las características del estudiante, de acuerdo a los requerimientos definidos en el capítulo II, sección II.2.1.

### III.2.1. REQUERIMIENTOS DE DATOS

Estos requerimientos se refieren a la información que debe ser posible representar para cada entidad de la ontología.

Para demostrar el cumplimiento de los mismos: se crearon las instancias correspondientes a las entidades involucradas en cada requerimiento, se ejecutó el razonador para verificar que la ontología seguía siendo consistente en presencia de la nueva información, y por último, se realizaron consultas a la misma con el lenguaje SPARQL para comprobar que la información se había asociado a las entidades correctamente.

**RI.** *Debe ser posible identificar al estudiante a través de su nombre, su DNI, su sexo, su fecha de nacimiento, su domicilio, su teléfono y su dirección de correo electrónico.*

Se definieron los siguientes estudiantes con sus correspondientes datos de identificación, como se describe en la tabla III.1.

**Tabla III.1:** Dos instancias de alumnos y sus datos de identificación.

	Alumno A	Alumno B
<b>Nombre</b>	Josefina Vázquez	Mariano Aquino
<b>DNI</b>	35250250	33100200
<b>Sexo</b>	F	M
<b>Fecha de nacimiento</b>	10-03-1990	25-10-1987
<b>Domicilio</b>	Belgrano (s) 1520	Laprida 98, La Banda.
<b>Teléfono</b>	385-5161878	385-689-9888
<b>E-mail</b>	josefina@gmail.com	mariano_28@yahoo.com.ar

En Protégé, se crearon las instancias de tipo estudiante para los alumnos A y B, y luego las instancias de datos de identificación, para cada uno de ellos (figuras III.9, III.10 y III.11).

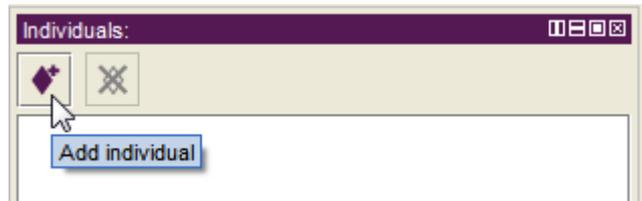


Figura III.9: Agregar nuevo individuo.

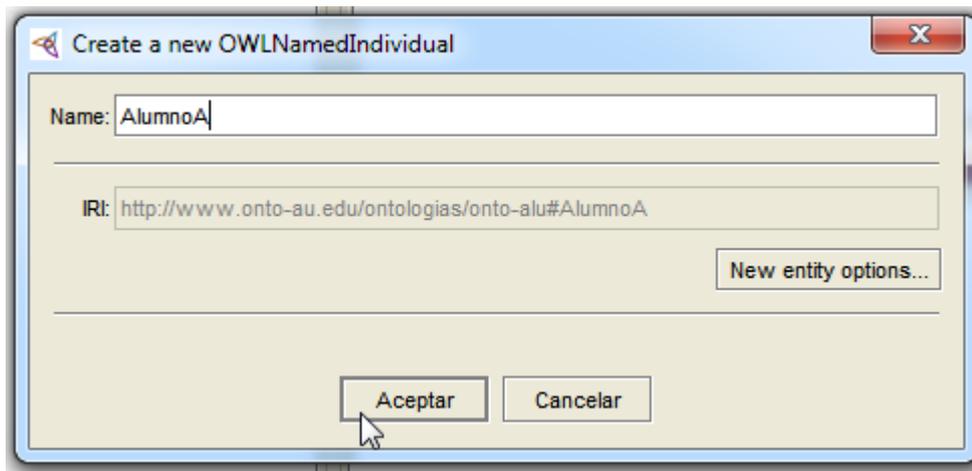


Figura III.10: Ingresar nombre del nuevo individuo.

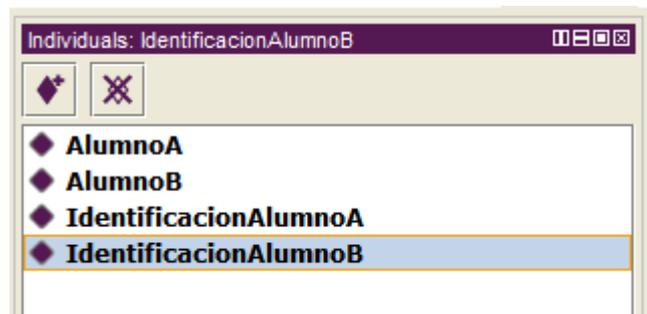


Figura III.11: Instancias de alumnos A y B, y sus datos de identificación.

Una vez que las instancias estuvieron definidas, se las asoció con las propiedades de datos correspondientes, tal como se muestra en la figura III.12.

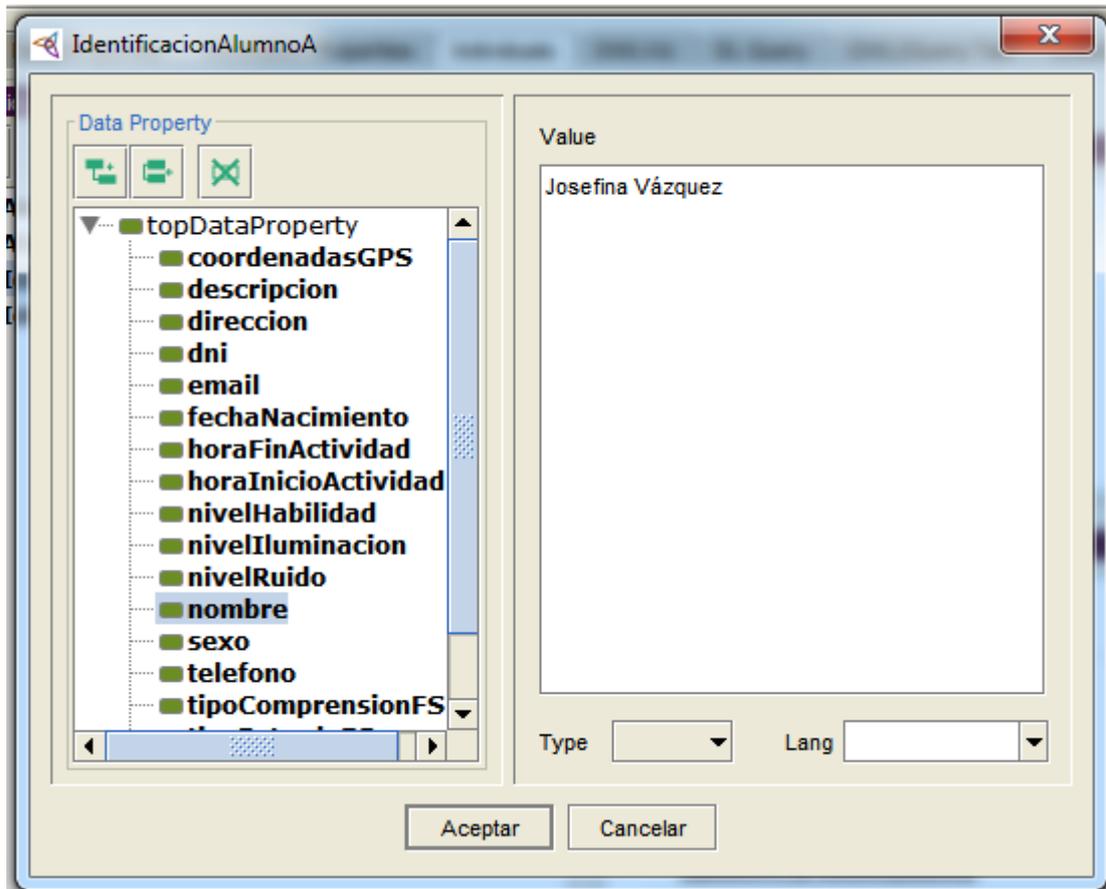


Figura III.12: Crear nombre asociado a IdentificacionAlumnoA.

En las figuras III.13 y III.14 se muestran todas las propiedades de datos creadas para alumno.



Figura III.13: Todos los datos de IdentificacionAlumnoA.



Figura III.14: Todos los datos de IdentificacionAlumnoB.

Por último, se vinculó cada estudiante con su conjunto de datos, lo cual se puede observar en las figuras III.15, III.16 y III.17.

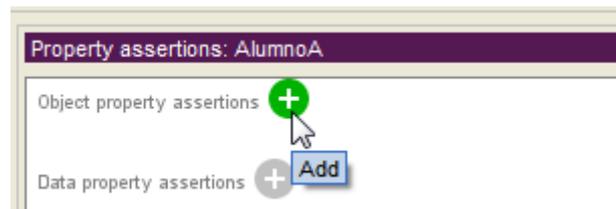


Figura III.15: Agregar aserción de propiedad de objeto vinculada a AlumnoA.

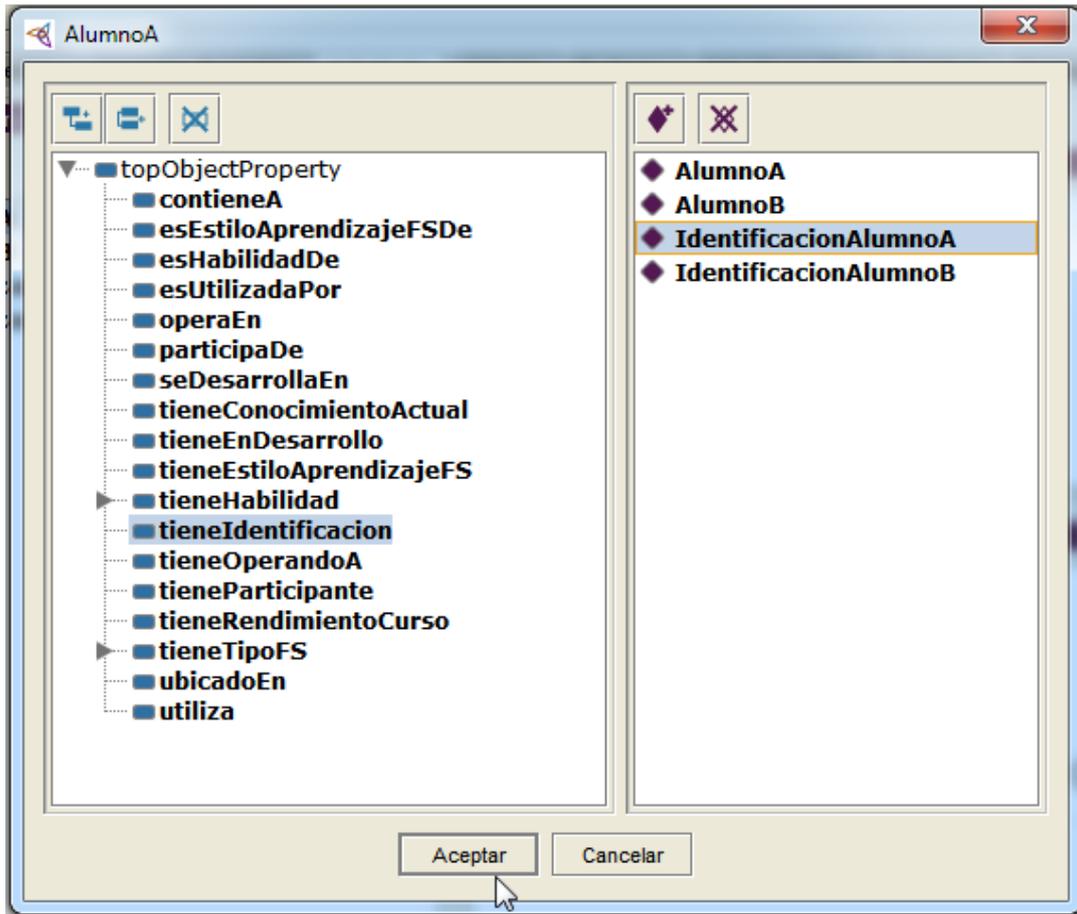


Figura III.16: AlumnoA tieneIdentificacion IdentificacionAlumnoA.

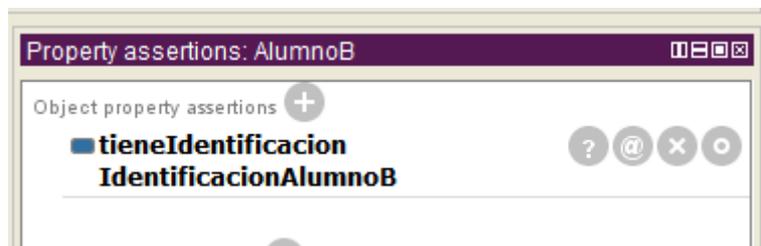


Figura III.17: AlumnoB tieneIdentificacion IdentificacionAlumnoB.

En este punto, se ejecutó el razonador, y se realizó una consulta a la ontología para obtener los datos asociados a cada alumno a través de su identificación. Los resultados de la misma se muestran en la figura III.18.

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX alu: <http://www.onto-au.edu/ontologias/onto-alu#>

SELECT ?Alumno ?Identificacion ?Prop ?Dato
WHERE
{
  ?Alumno alu:tieneIdentificacion ?Identificacion .
  ?Identificacion ?Prop ?Dato .
  ?Rest rdf:type owl:Restriction .
  ?Rest owl:onProperty ?Prop .
  alu:Identificacion rdfs:subClassOf ?Rest
}
    
```

Alumno	Identificacion	Prop	Dato
AlumnoA	IdentificacionAlumnoA	dni	"35250250"^^<http://www.w3.org/2001/XMLSchema#int>
AlumnoA	IdentificacionAlumnoA	email	"josefina@gmail.com
AlumnoA	IdentificacionAlumnoA	telefono	"385-5161878"@
AlumnoA	IdentificacionAlumnoA	sexo	"F"@
AlumnoA	IdentificacionAlumnoA	direccion	"Belgrano (s) 1520"@
AlumnoA	IdentificacionAlumnoA	nombre	"Josefina Vázquez"@
AlumnoA	IdentificacionAlumnoA	fechaNacimiento	"10-03-1990"@
AlumnoB	IdentificacionAlumnoB	dni	"33100200"^^<http://www.w3.org/2001/XMLSchema#int>
AlumnoB	IdentificacionAlumnoB	email	"mariano_28@yahoo.com.ar
AlumnoB	IdentificacionAlumnoB	telefono	"385-689-9888"@
AlumnoB	IdentificacionAlumnoB	sexo	"M"@
AlumnoB	IdentificacionAlumnoB	direccion	"Laprida 98, La Banda."@
AlumnoB	IdentificacionAlumnoB	nombre	"Mariano Aquino"@

Figura III.18: Datos de cada alumno.

**R2.** Debe ser posible representar el estilo de aprendizaje del estudiante de acuerdo al Modelo de Felder & Silverman, expresado como un conjunto de 4 valores, donde cada uno de ellos corresponde a un tipo (preferencia) por dimensión.

Se definieron dos estudiantes, y un estilo de aprendizaje para cada uno de ellos, de acuerdo a los datos de la tabla III.2.

Tabla III.2: Alumnos A y B con sus respectivos estilos de aprendizaje.

	Alumno A	Alumno B
<b>Tipo de procesamiento</b>	Activo	Activo
<b>Tipo de percepción</b>	Sensitivo	Intuitivo
<b>Tipo de entrada</b>	Verbal	Visual
<b>Tipo de comprensión</b>	Secuencial	Global

En Protégé, se crearon las instancias de los alumnos y de los estilos de aprendizaje de los mismos, como se observa en la figura III.19.

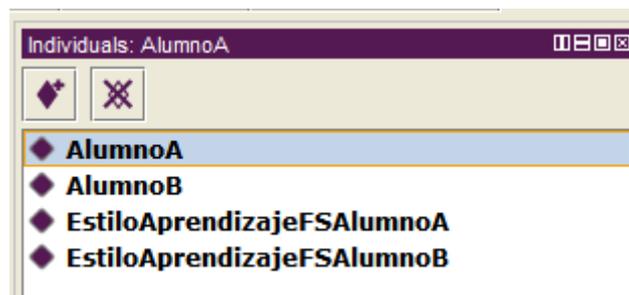


Figura III.19: Instancias de alumnos y estilos de aprendizaje.

Luego, se asoció a cada estilo de aprendizaje las propiedades de datos correspondientes para representar las preferencias del alumno por cada dimensión del estilo de aprendizaje (figuras III.20 y III.21).



Figura III.20: Preferencias para cada dimensión del EstiloAprendizajeFSAlumnoA.



Figura III.21: Preferencias para cada dimensión del EstiloAprendizajeFSAlumnoB.

Por último, se vinculó cada alumno con su estilo de aprendizaje, tal como se puede observar en las figuras III.22 y III.23.

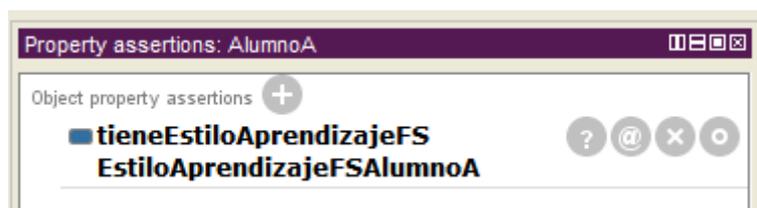


Figura III.22: AlumnoA tiene estilo de aprendizaje EstiloAprendizajeFSAlumnoA.

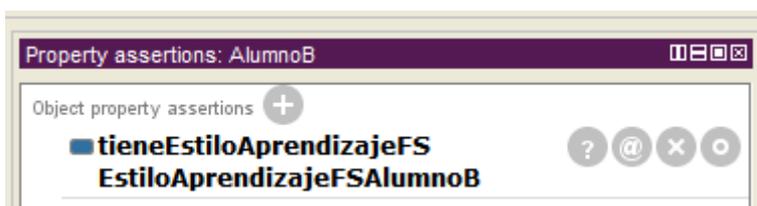


Figura III.23: AlumnoB tiene estilo de aprendizaje EstiloAprendizajeFSAlumnoB.

Luego de ejecutar el razonador, se realizó la consulta de información a la ontología. Los resultados de la misma se muestran en la figura III.24.

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX alu: <http://www.onto-au.edu/ontologias/onto-alu#>

SELECT ?Alumno ?EstiloFS ?Valor
WHERE
{
  ?Alumno alu:tieneEstiloAprendizajeFS ?EstiloFS .
  ?EstiloFS ?Prop ?Valor .
  ?Prop rdfs:domain alu:EstiloAprendizajeFS
}
    
```

Alumno	EstiloFS	
AlumnoB	EstiloAprendizajeFSAlumnoB	"Intuitivo"@
AlumnoB	EstiloAprendizajeFSAlumnoB	"Global"@
AlumnoB	EstiloAprendizajeFSAlumnoB	"Activo"@
AlumnoB	EstiloAprendizajeFSAlumnoB	"Visual"@
AlumnoA	EstiloAprendizajeFSAlumnoA	"Sensitivo"@
AlumnoA	EstiloAprendizajeFSAlumnoA	"Secuencial"@
AlumnoA	EstiloAprendizajeFSAlumnoA	"Activo"@
AlumnoA	EstiloAprendizajeFSAlumnoA	"Verbal"@

**Figura III.24:** Estilo de aprendizaje de cada alumno.

**R3.** Debe ser posible definir el nivel del estudiante para cada una de las siguientes habilidades: autoevaluación, búsqueda de ayuda, pensamiento crítico, razonamiento científico y resolución de problemas; en una escala cualitativa con valores {Alto, Medio, Bajo}.

Se definieron los siguientes alumnos, cada uno de ellos con dos habilidades y el respectivo nivel para cada habilidad, como se muestra en la tabla III.3.

**Tabla III.3:** Alumnos A y B, cada uno con su nivel de habilidad.

	Alumno A	Alumno B
Autoevaluación	Alto	Medio
Razonamiento científico	Medio	-
Pensamiento crítico	-	Bajo

En Protégé se crearon las instancias de los alumnos y de las habilidades tal como se puede observar en la figura III.25.

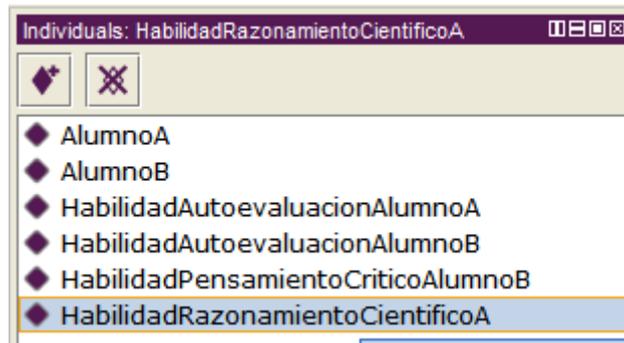


Figura III.25: Instancias de alumnos y habilidades.

Luego se asoció a cada habilidad el nivel correspondiente y, por último, a cada estudiante las habilidades que les correspondían (figuras III.26 y III.27).

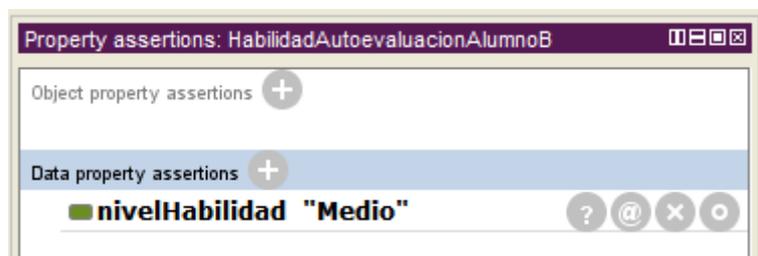


Figura III.26: Nivel para la habilidad de autoevaluación del alumno B.

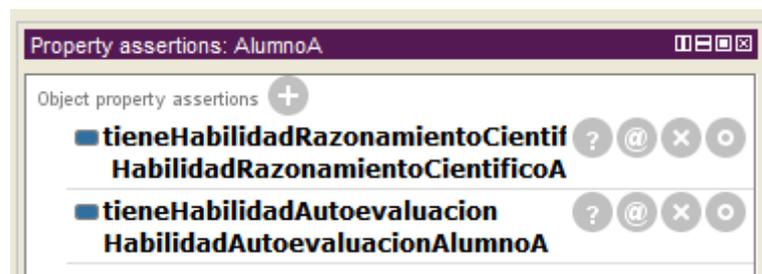
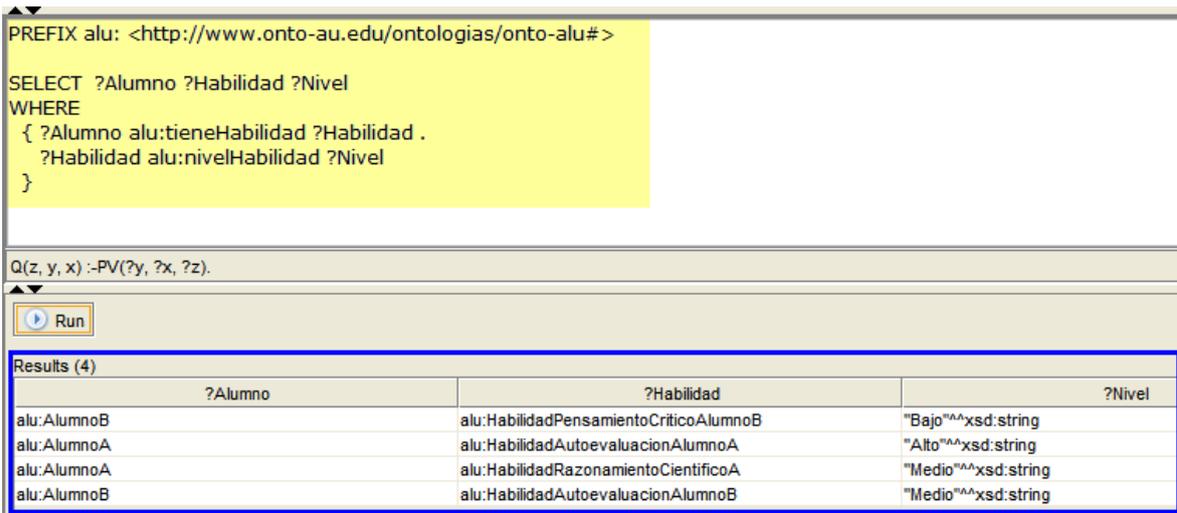


Figura III.27: Habilidades del alumno A.

Luego de ejecutar el razonador, se realizó una consulta para determinar las habilidades de cada alumno, con su correspondiente nivel. Los resultados de la misma se muestran en la figura III.28.



**Figura III.28:** Habilidades de cada alumno, con su nivel.

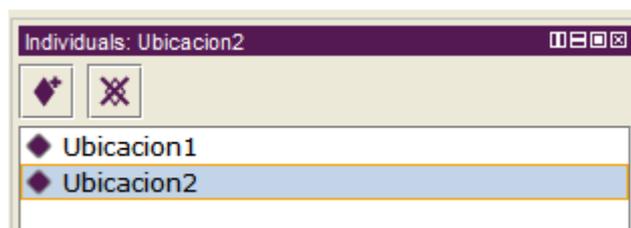
**R4.** Para cada ubicación, debe ser posible definir: si es de interior o exterior, su descripción textual, dirección y coordenadas GPS.

Se definieron dos ubicaciones, cada una con sus correspondientes datos, de acuerdo a lo especificado en la tabla III.4.

**Tabla III.4:** Ubicaciones 1 y 2, con sus datos.

	Ubicación 1	Ubicación 2
<b>Tipo</b>	Interior	Exterior
<b>Descripción</b>	Aula 23, FCEyT.	Plaza Libertad
<b>Dirección</b>	Av. Belgrano (s) 1920. 2do piso, edificio central.	Av. Libertad 400-498.
<b>Coordenadas GPS</b>	(-27.801159,-64.250733)	(-27.787596,-64.259657)

En Protégé, se crearon las instancias para estas ubicaciones (figura III.29) y se especificó el tipo de cada una (figura III.30). Luego se asoció a cada una de ellas las propiedades de datos correspondientes para definir su descripción, dirección y coordenadas GPS (figuras III.31 y III.32).



**Figura III.29:** Instancias ubicación 1 y ubicación 2.

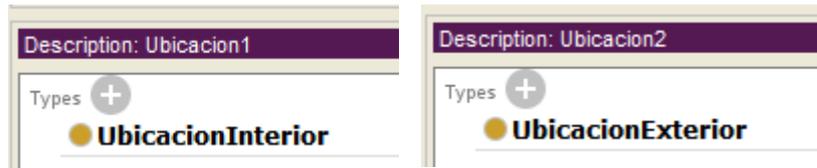


Figura III.30: Ubicación 1 es una ubicación de interior y ubicación 2 es una ubicación de exterior.

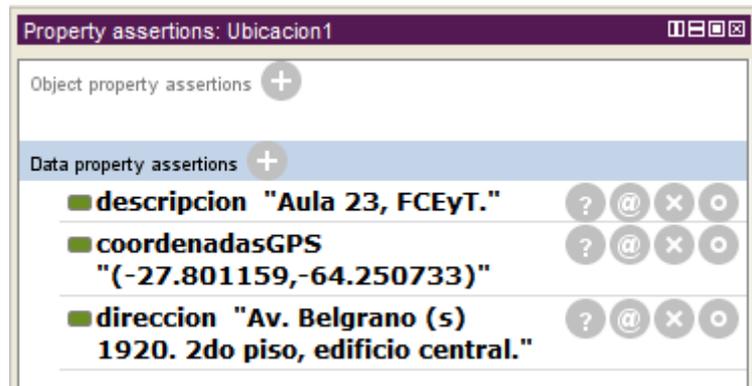


Figura III.31: Datos descriptivos de Ubicación1.

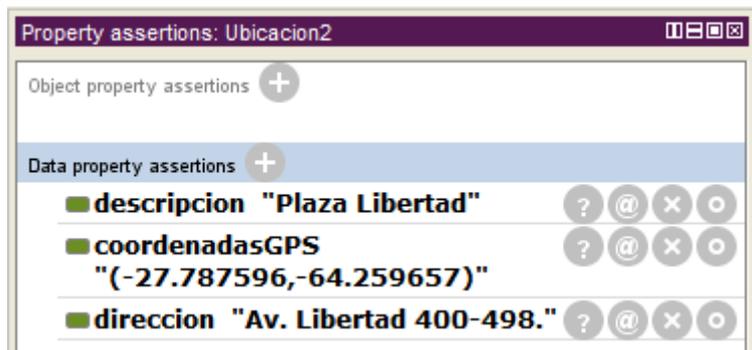


Figura III.32: Datos descriptivos de Ubicación2.

Por último, se ejecutó el razonador y se consultaron los datos y tipo de cada ubicación. Los resultados de la misma se muestran en la figura III.33.

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX alu: <http://www.onto-au.edu/ontologias/onto-alu#>

SELECT ?Ubicacion ?Tipo ?Prop ?Dato
WHERE
{
  ?Ubicacion rdf:type ?Tipo .
  ?Tipo rdfs:subClassOf alu:Ubicacion .
  ?Ubicacion ?Prop ?Dato .
  ?Rest rdf:type owl:Restriction .
  ?Rest owl:onProperty ?Prop .
  alu:Ubicacion rdfs:subClassOf ?Rest
}
    
```

Ubicacion	Tipo	Prop	Dato
Ubicacion2	UbicacionExterior	descripcion	"Plaza Libertad"@
Ubicacion2	UbicacionExterior	coordenadasGPS	"(-27.787596,-64.259657)"@
Ubicacion2	UbicacionExterior	direccion	"Av. Libertad 400-498."@
Ubicacion1	UbicacionInterior	descripcion	"Aula 23, FCEyT."@
Ubicacion1	UbicacionInterior	coordenadasGPS	"(-27.801159,-64.250733)"@
Ubicacion1	UbicacionInterior	direccion	"Av. Belgrano (s) 1920. 2do piso, edificio central."@

**Figura III.33:** Datos de cada ubicación.

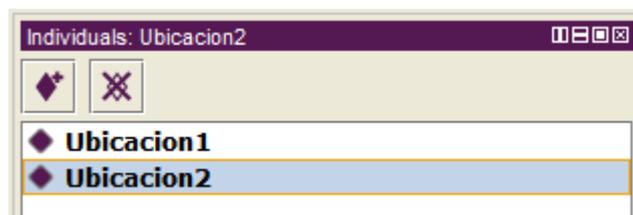
**R5.** Para cada ubicación de interior, debe ser posible definir: si es un aula, box o laboratorio, su nivel de iluminación {Adecuado o bajo} y su nivel de ruido {Adecuado, Alto o Muy Alto}.

Se definieron dos ubicaciones interiores, cada una con su descripción y nivel de iluminación y ruido, tal como se muestra en la tabla III.5.

**Tabla III.5:** Ubicaciones de interior 1 y 2 con sus datos descriptivos.

	Ubicación 1	Ubicación 2
<b>Tipo</b>	Aula	Laboratorio
<b>Descripción</b>	Aula 23, FCEyT.	Laboratorio Beta.
<b>Nivel de iluminación</b>	Adecuado	Bajo
<b>Nivel de ruido</b>	Alto	Adecuado

En Protégé se crearon las instancias de las ubicaciones (figura III.34) y se especificó el tipo de cada una (figura III.35).



**Figura III.34:** Instancias de ubicación 1 y 2.



Figura III.35: Ubicación 1 es un aula y ubicación 2 es un laboratorio.

Luego, se especificaron las propiedades de datos para cada ubicación, como se puede observar en las figuras III.36 y III.37.

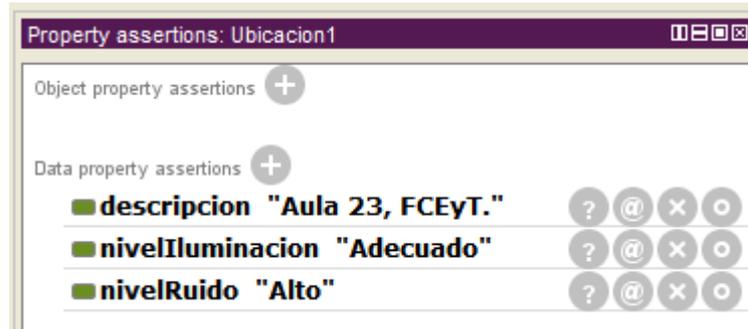


Figura III.36: Datos descriptivos de ubicación 1.

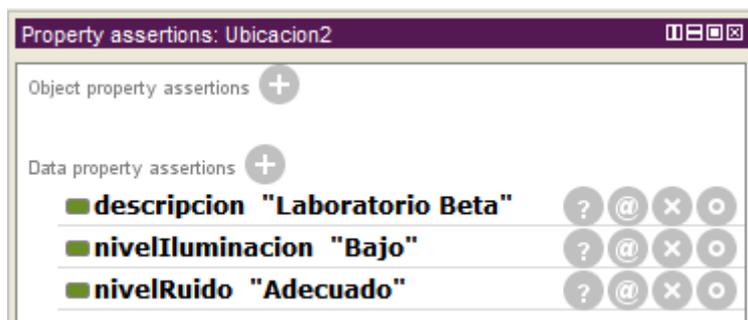


Figura III.37: Datos descriptivos de ubicación 2.

Por último, se ejecutó el razonador y se consultó la ontología para obtener los datos de las ubicaciones 1 y 2 así como el tipo de cada una de ellas. Los resultados de la misma se muestran en la figura III.38.

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX alu: <http://www.onto-au.edu/ontologias/onto-alu#>

SELECT ?Ubicacion ?Tipo ?Prop ?Dato
WHERE
{
  ?Ubicacion rdf:type ?Tipo .
  ?Tipo rdfs:subClassOf alu:UbicacionInterior .
  ?Ubicacion ?Prop ?Dato .
  { {?Rest rdf:type owl:Restriction .
    ?Rest owl:onProperty ?Prop .
    alu:UbicacionInterior rdfs:subClassOf ?Rest}
  UNION
  {?Rest rdf:type owl:Restriction .
    ?Rest owl:onProperty ?Prop .
    alu:Ubicacion rdfs:subClassOf ?Rest}}
}
    
```

Ubicacion	Tipo	Prop	Dato
Ubicacion1	Aula	nivelRuido	"Alto"@
Ubicacion1	Aula	nivelIluminacion	"Adecuado"@
Ubicacion1	Aula	descripcion	"Aula 23, FCEyT."@
Ubicacion2	Laboratorio	nivelRuido	"Adecuado"@
Ubicacion2	Laboratorio	nivelIluminacion	"Bajo"@
Ubicacion2	Laboratorio	descripcion	"Laboratorio Beta"@

**Figura III.38:** Ubicaciones 1 y 2 con sus datos asociados.

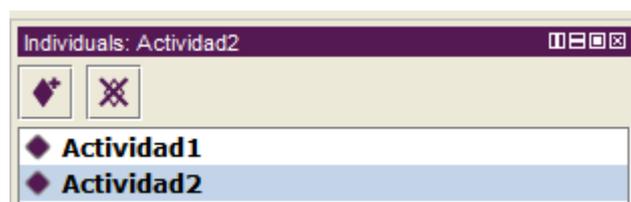
**R6.** Para cada actividad, debe ser posible definir: si es formal o informal, su descripción, su horario de inicio y su horario de finalización.

Se definieron dos instancias de actividad, con los datos descritos en la tabla III.6.

**Tabla III.6:** Datos de Actividad 1 y 2.

	Actividad 1	Actividad 2
<b>Tipo</b>	Formal	Informal
<b>Descripción</b>	Clase de Simulación	Reunión de grupo
<b>Hora inicio</b>	20-02-2014, 10:00:00 UTC-3.	20-02-2014, 18:00:00 UTC-3.
<b>Hora fin</b>	20-02-2014, 12:00:00 UTC-3.	-

En Protégé, se crearon las instancias para ambas actividades (figura III.39) y se definió el tipo de cada una (figura III.40).



**Figura III.39:** Instancias Actividad 1 y Actividad 2.



Figura III.40: Actividad 1 es una actividad formal y actividad 2 es una actividad informal.

Luego se asociaron las propiedades de datos para definir la descripción, hora de inicio y hora de finalización de cada actividad. Esto se observa en las figuras III.41 y III.42.

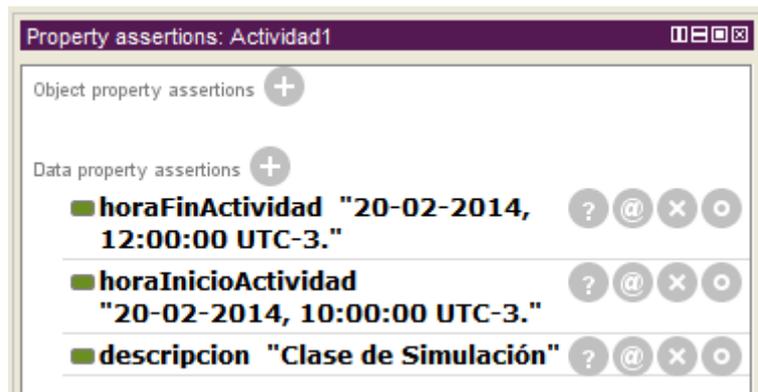


Figura III.41: Datos de Actividad 1.



Figura III.42: Datos de Actividad 2.

Finalmente, se ejecutó el razonador y se consultó la ontología para todos los datos asociados a cada una de las actividades. Los resultados de la misma se muestran en la figura III.43.

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX alu: <http://www.onto-au.edu/ontologias/onto-alu#>
SELECT ?Actividad ?Tipo ?Prop ?Dato
WHERE
{
  ?Actividad rdf:type ?Tipo .
  ?Tipo rdfs:subClassOf alu:Actividad .
  ?Actividad ?Prop ?Dato .
  ?Rest rdf:type owl:Restriction .
  ?Rest owl:onProperty ?Prop .
  alu:Actividad rdfs:subClassOf ?Rest
}
    
```

Actividad	Tipo	Prop	
Actividad1	ActividadFormal	descripcion	"Clase de Simulación"@
Actividad1	ActividadFormal	horaFinActividad	"20-02-2014, 12:00:00 UTC-3."@
Actividad1	ActividadFormal	horaInicioActividad	"20-02-2014, 10:00:00 UTC-3."@
Actividad2	ActividadInformal	descripcion	"Reunión de grupo"@
Actividad2	ActividadInformal	horaInicioActividad	"20-02-2014, 18:00:00 UTC-3."@

Figura III.43: Datos de cada actividad.

R7. Para cada actividad formal debe ser posible definir si es una clase o una evaluación.

Se definieron dos actividades formales, cada una con su tipo y descripción, de acuerdo a la tabla III.7.

Tabla III.7: Datos de las Actividades formales 1 y 2.

	Actividad 1	Actividad 2
<b>Tipo</b>	Clase	Evaluación
<b>Descripción</b>	Clase de Simulación	Parcial de Programación I

En Protégé, se crearon las instancias para ambas actividades y se definió el tipo (figura III.44) y descripción de cada una (figuras III.45 y III.46).

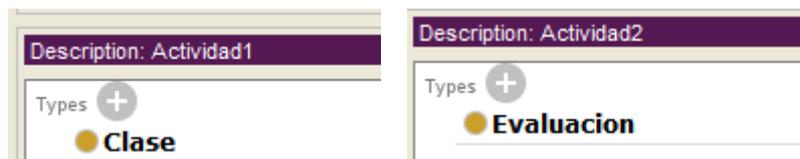


Figura III.44: Actividad 1 es una clase y Actividad 2 es una evaluación.

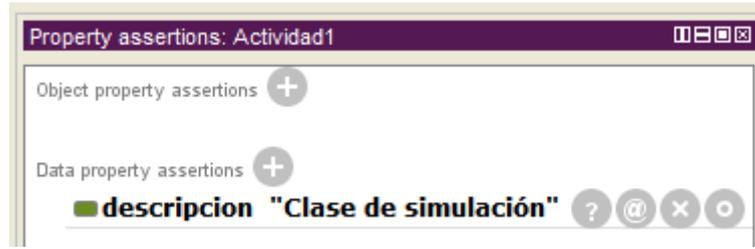


Figura III.45: Descripción de Actividad 1.

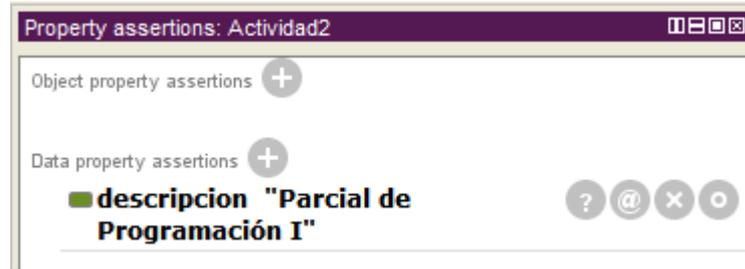


Figura III.46: Descripción de Actividad 2.

Por último, se ejecutó el razonador consultando a la ontología los datos asociados a cada una de las actividades. Los resultados de la misma se muestran en la figura III.47.

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX alu: <http://www.onto-au.edu/ontologias/onto-alu#>

SELECT ?Actividad ?Tipo ?Prop ?Dato
WHERE
{
  ?Actividad rdf:type ?Tipo .
  ?Tipo rdfs:subClassOf alu:ActividadFormal .
  ?Actividad ?Prop ?Dato .
  ?Rest rdf:type owl:Restriction .
  ?Rest owl:onProperty ?Prop .
  alu:Actividad rdfs:subClassOf ?Rest
}
    
```

Actividad	Tipo	Prop	
Actividad2	Evaluacion	descripcion	"Parcial de Programación I"@
Actividad1	Clase	descripcion	"Clase de simulación"@

Figura III.47: Tipo y descripción de actividad 1 y 2.

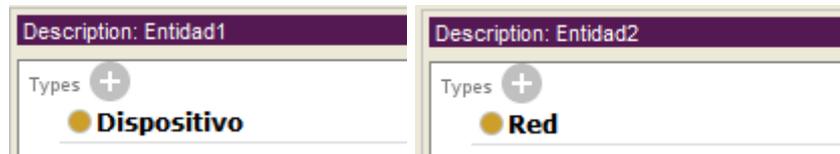
**R8.** Para cada entidad computacional, debe ser posible definir: si es un dispositivo, una red, o una aplicación, y su descripción.

Se definieron dos entidades, de distintos tipos y cada una con su correspondiente descripción, de acuerdo a lo descrito en la tabla III.8.

**Tabla III.8:** Datos de las Entidades 1 y 2.

	Entidad 1	Entidad 2
<b>Tipo</b>	Dispositivo	Red
<b>Descripción</b>	Impresora LAB02-15	Wifi-Bedelia

En Protégé, se crearon las instancias Entidad 1 y 2 y se especificó el tipo de cada una (figura III.48). También se definieron sus descripciones (figuras III.49 y III.50).



**Figura III.48:** Entidad 1 es un dispositivo y Entidad 2 es una red.



**Figura III.49:** Descripción de Entidad 1.



**Figura III.50:** Descripción de Entidad 2.

Por último, se ejecutó el razonador y se consultó la ontología para obtener los datos asociados a las entidades 1 y 2. Los resultados de la misma se muestran en la figura III.51.

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX alu: <http://www.onto-au.edu/ontologias/onto-alu#>

SELECT ?Entidad ?Tipo ?Prop ?Dato
WHERE
{
  ?Entidad rdf:type ?Tipo .
  ?Tipo rdfs:subClassOf alu:EntidadComputacional .
  ?Entidad ?Prop ?Dato .
  ?Rest rdf:type owl:Restriction .
  ?Rest owl:onProperty ?Prop .
  alu:EntidadComputacional rdfs:subClassOf ?Rest
}
    
```

Entidad	Tipo	Prop	
Entidad1	Dispositivo	descripcion	"Impresora LAB02-15"@
Entidad2	Red	descripcion	"Wifi-Bedelia"@

Figura III.51: Datos de las entidades 1 y 2.

**R9.** Para cada entidad de tipo dispositivo debe ser posible definir si es una impresora, una PC, un proyector, un Smartphone o una Tablet.

Se definieron dos dispositivos, con su descripción y tipo, de acuerdo a la tabla III.9.

Tabla III.9: Definición de entidades 1 y 2.

	Entidad 1	Entidad 2
<b>Tipo</b>	Impresora	Smartphone
<b>Descripción</b>	Impresora LAB02-15	Motorola RAZR

En Protégé, se crearon las instancias Entidad 1 y 2 y se especificó el tipo de cada una (figura III.52). También se definieron sus descripciones (figuras III.53 y III.54).

Description: Entidad1

Types +

● **Impresora**

Description: Entidad2

Types +

● **Smartphone**

Figura III.52: Entidad 1 es una impresora y Entidad 2 es un Smartphone.



Figura III.53: Descripción de Entidad 1.



Figura III.54: Descripción de Entidad 2.

Por último, se ejecutó el razonador y se consultó la ontología para obtener los datos asociados a las entidades 1 y 2. Los resultados de la misma se muestran en la figura III.55.

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX alu: <http://www.onto-au.edu/ontologias/onto-alu#>
SELECT ?Dispositivo ?Tipo ?Prop ?Dato
WHERE
{
  ?Dispositivo rdf:type ?Tipo .
  ?Tipo rdfs:subClassOf alu:Dispositivo .
  ?Dispositivo ?Prop ?Dato .
  ?Rest rdf:type owl:Restriction .
  ?Rest owl:onProperty ?Prop .
  alu:EntidadComputacional rdfs:subClassOf ?Rest
}
```

Dispositivo	Tipo	Prop	
Entidad1	Impresora	descripcion	"Impresora LAB02-15"@
Entidad2	Smartphone	descripcion	"Motorola RAZR"@

Figura III.55: Datos de entidades 1 y 2.

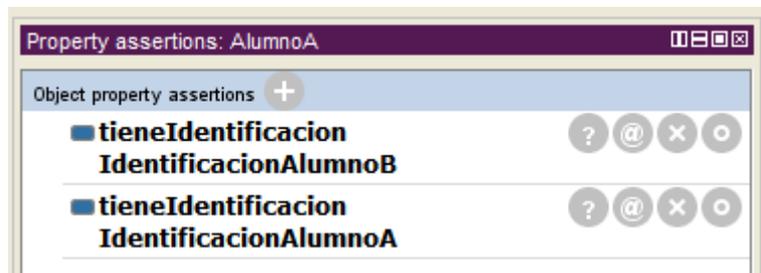
### III.2.2. REQUERIMIENTOS DE RELACIONES ENTRE DATOS

Comprenden los requerimientos que caracterizan las relaciones en las que participan las distintas clases de individuos. En *onto-alu*, estas relaciones puede ser de dos tipos: relaciones que limitan a la instancia del dominio a vincularse con una única instancia del rango, o bien relaciones donde la instancia del dominio puede enlazarse con varias instancias del rango.

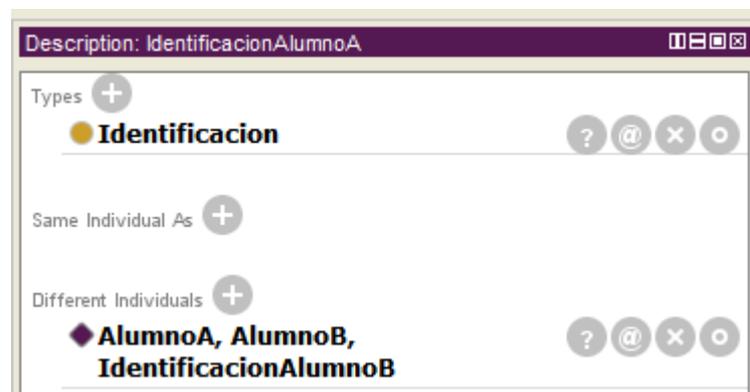
Para probar los requerimientos del primer tipo, se crearon instancias y se declararon propiedades que violaban la restricción de cardinalidad, mostrando como al ejecutar el razonador la ontología se volvía inconsistente. Para los requerimientos del segundo tipo, se declararon instancias y relaciones entre las mismas, se ejecutó el razonador para comprobar que no se introducían inconsistencias, y por último se consultó la ontología para verificar que la información era correcta.

**R10.** *Un estudiante puede tener un único conjunto de datos de identificación.*

Partiendo de las instancias creadas para verificar R1, se relacionó al alumno A con dos conjuntos de datos de identificación: *IdentificacionAlumnoA* e *IdentificacionAlumnoB* (figura III.56), siendo ambas instancias diferentes (figura III.57).

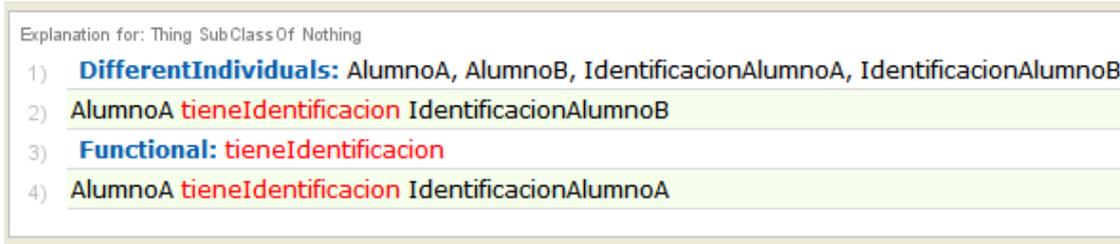


**Figura III.56:** Alumno A posee dos conjuntos de datos de identificación.



**Figura III.57:** *IdentificaciónAlumnoA* e *IdentificacionAlumnoB* son individuos distintos.

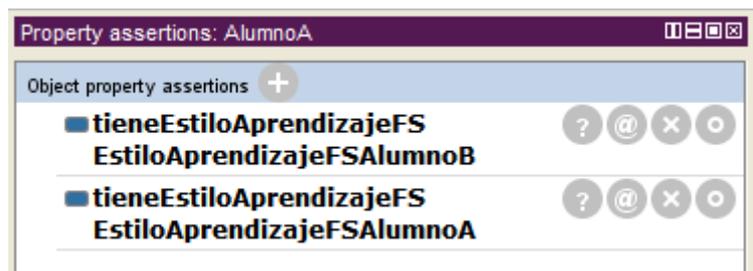
Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se puede observar en la figura III.58, proporcionando la explicación de que el Alumno A se relaciona con dos instancias diferentes de Identificación, a través de una propiedad que es funcional.



**Figura III.58:** Explicación de inconsistencia relacionada con tieneIdentificacion.

**R11.** *Un estudiante puede tener un único estilo de aprendizaje.*

Partiendo de las instancias creadas para verificar R2, se relacionó al alumno A con 2 estilos de aprendizaje distintos (figuras III.59 y III.60).

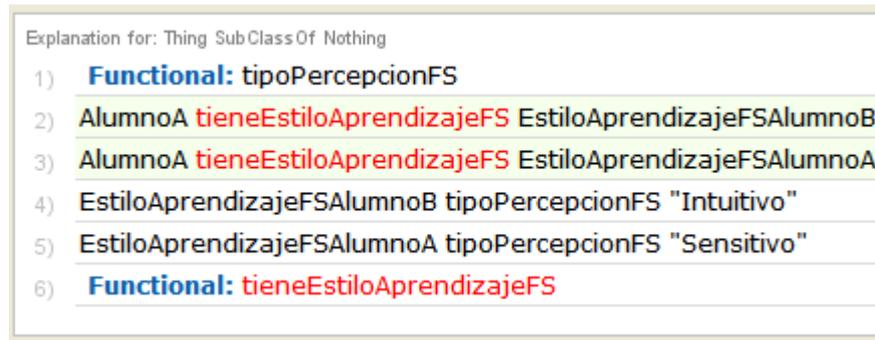


**Figura III.59:** Alumno A posee dos estilos de aprendizaje.



**Figura III.60:** EstiloAprendizajeAlumnoA es diferente de EstiloAprendizajeAlumnoB.

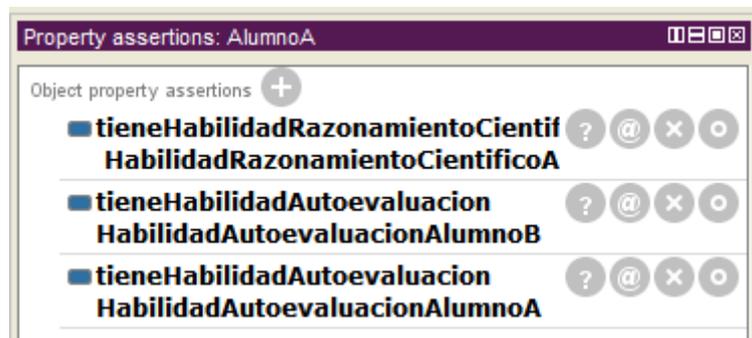
Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se observa en la figura III.61, proporcionando la explicación de que el Alumno A se relaciona con dos instancias diferentes de estilos de aprendizaje, a través de una propiedad que es funcional.



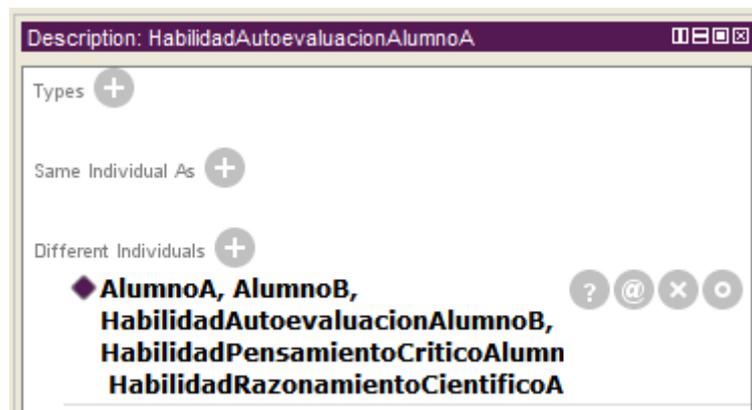
**Figura III.61:** Explicación de inconsistencia relacionada con tieneEstiloAprendizajeFS.

**R12.** *Un estudiante puede tener un único nivel por habilidad.*

Partiendo de las instancias definidas en la prueba de R3, se relacionó al alumno A con dos niveles para la habilidad de autoevaluación (figuras III.62 y III.63).



**Figura III.62:** Alumno A se relaciona con dos instancias de una habilidad del mismo tipo.



**Figura III.63:** Las instancias HabilidadAutoevaluacionAlumnoA y B representan dos individuos diferentes.

Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se observa en la figura III.64, proporcionando la explicación de que el Alumno A se relaciona con dos instancias diferentes de habilidades de autoevaluación, a través de una propiedad que es funcional.

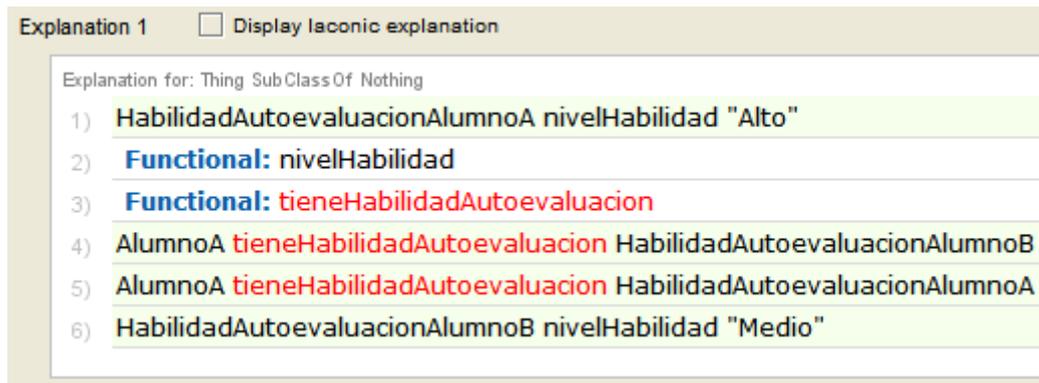


Figura III.64: Explicación de inconsistencia relacionada con tieneHabilidadAutoevaluacion.

**R13.** *Un estudiante puede encontrarse en una única ubicación.*

Partiendo de las instancias definidas en la prueba de R4, se relacionó al alumno A con dos ubicaciones distintas (figuras III.65 y III.66).



Figura III.65: Ubicacion 1 es diferente de Ubicacion 2.

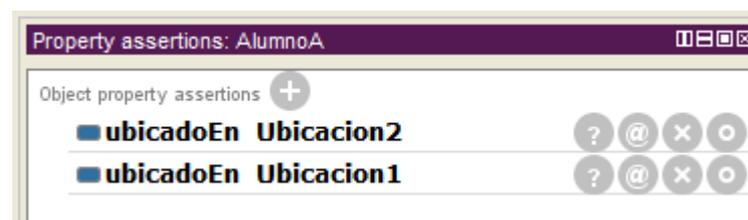


Figura III.66: Alumno A está ubicado en Ubicación 1 y 2.

Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se observa en la figura III.67, proporcionando la explicación de que el Alumno A se relaciona con dos ubicaciones diferentes, a través de una propiedad que es funcional.

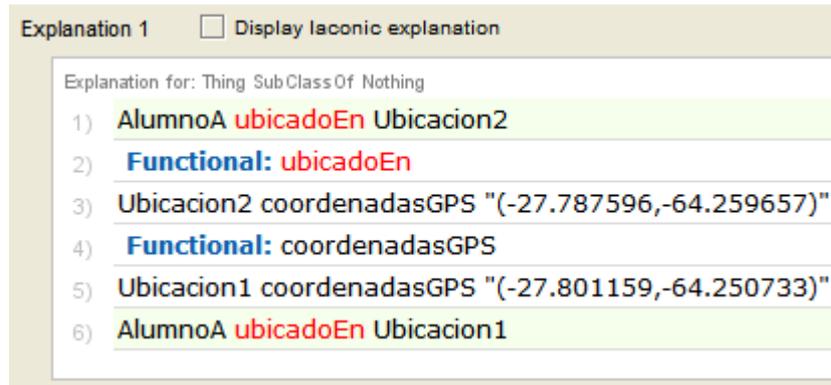


Figura III.67: Explicación de inconsistencia relacionada con ubicadoEn.

**R14.** Una ubicación puede contener más de un estudiante simultáneamente.

Partiendo de las instancias definidas en la prueba de R4, se relacionó tanto al alumno A como al alumno B con la ubicación 1, tal como se observa en las figuras III.68 y III.69.

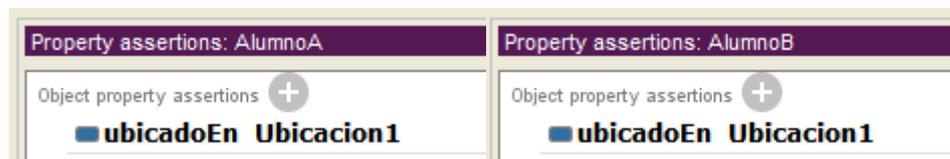


Figura III.68: Alumno A y alumno B se encuentran en la misma ubicación.

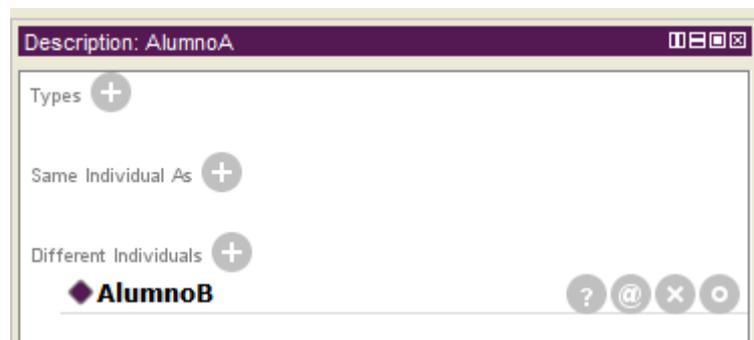


Figura III.69: Alumno A y alumno B son distintos individuos.

Al ejecutar el razonador no se encuentran errores, puesto que es posible que una ubicación contenga más de un estudiante. Si ejecutamos una consulta para determinar todos los alumnos ubicados en Ubicación 1, obtenemos a ambos alumnos. Estos resultados se muestran en la figura III.70.

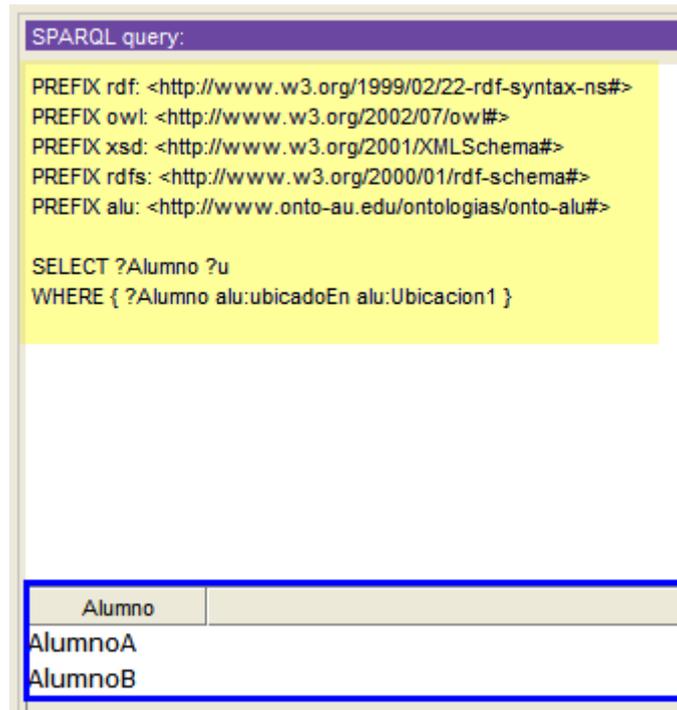


Figura III.70: Alumnos ubicados en Ubicacion1.

**R15.** Una ubicación puede contener más de una entidad computacional simultáneamente.

Partiendo de las instancias definidas para verificar R8, se relacionó a la ubicación 2 con las entidades 1 y 2. Esto se puede observar en las figuras III.71 y III.72.

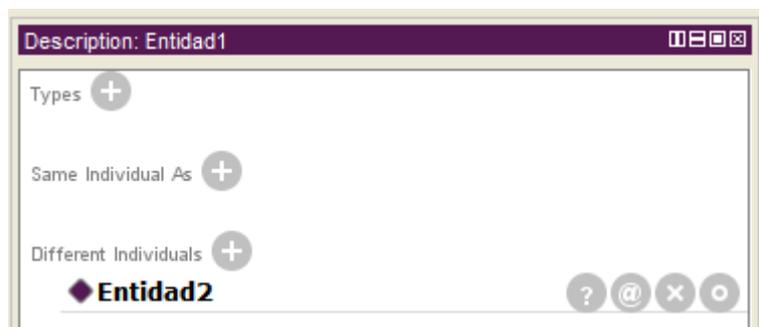


Figura III.71: Entidad 1 y 2 son individuos diferentes.



Figura III.72: Entidad 1 y 2 se encuentran en Ubicación 2.

Al ejecutar el razonador no se encuentran errores, puesto que es posible que una ubicación contenga más de un entidad computacional. Si ejecutamos una consulta para

determinar todas las entidades computacionales ubicadas en Ubicación 2, obtenemos a Entidad 1 y 2. Estos resultados se muestran en la figura III.73.

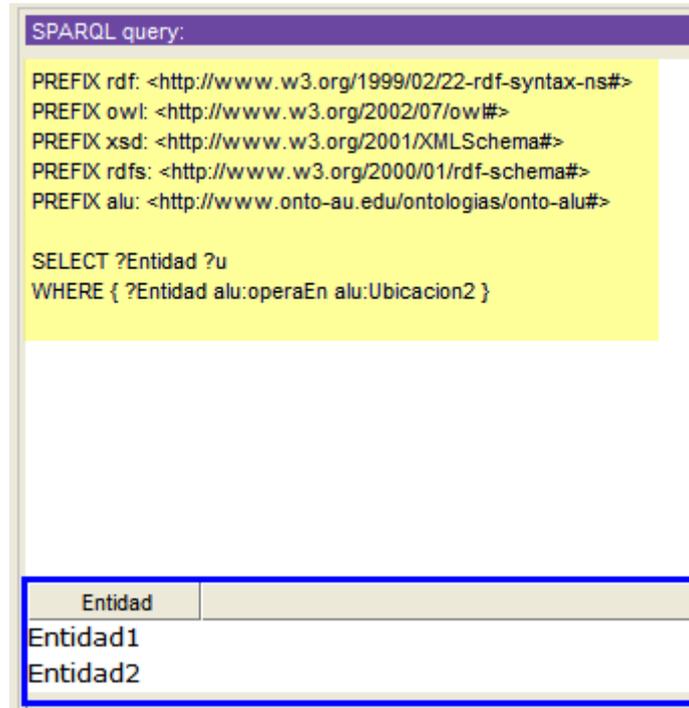


Figura III.73: Todas las entidades en Ubicación 2.

**R16.** Una ubicación puede contener más de un actividad simultáneamente.

Partiendo de las instancias definidas para verificar R7, se relacionó a las actividades 1 y 2 con la ubicación 1 (figuras III.74 y III.75).

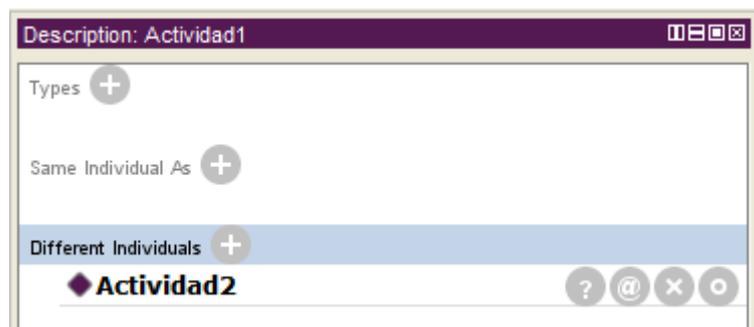


Figura III.74: Actividad 1 y 2 son individuos distintos.



Figura III.75: Actividad 1 y Actividad 2 se desarrollan en Ubicación 1.

Al ejecutar el razonador no se encontraron errores, puesto que es posible que una ubicación contenga más de un actividad. Al ejecutar una consulta para determinar todas las actividades ubicadas en Ubicacion1, se obtuvieron ambas actividades. Estos resultados se muestran en la figura III.76.

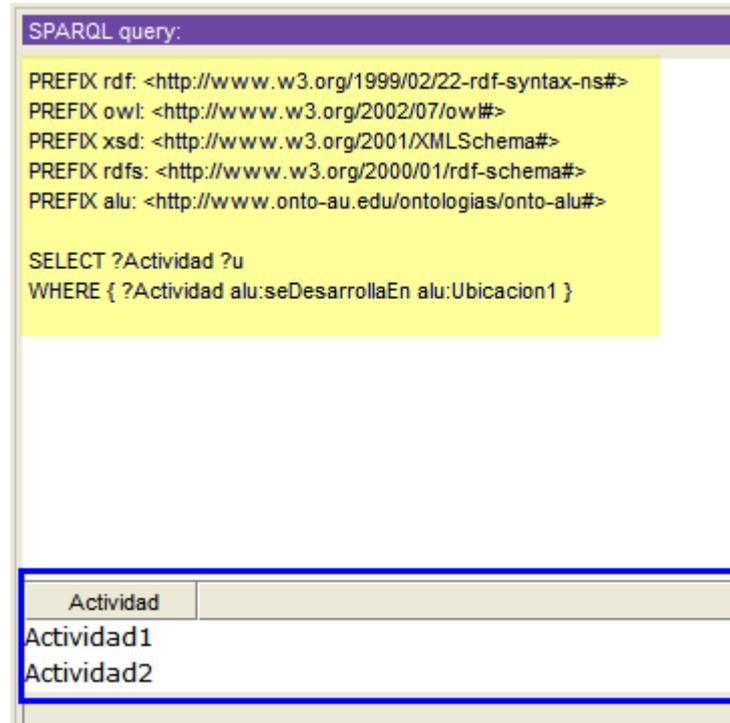


Figura III.76: Todas las actividades de Ubicacion1.

**R17.** *Un estudiante puede estar realizando solo una actividad.*

Partiendo de las instancias definidas en la prueba de R6, se relacionó al alumno A con dos actividades diferentes. Esto se puede observar en las figuras III.77 y III.78.

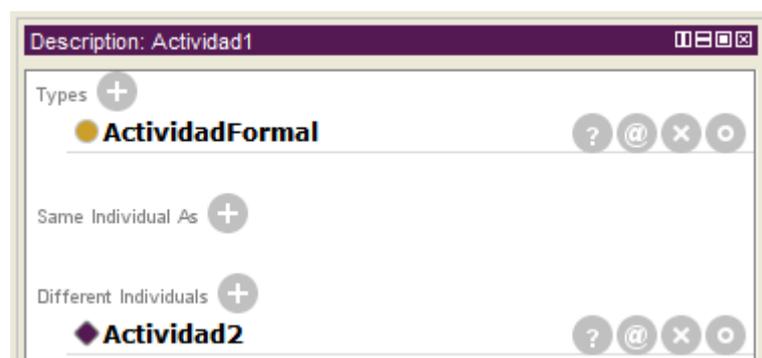


Figura III.77: Actividad 1 es distinta de Actividad 2.



Figura III.78: Alumno A participa de Actividad 1 y de Actividad 2.

Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se muestra en la figura III.79, proporcionando la explicación de que el Alumno A se relaciona con dos actividades diferentes, a través de una propiedad que es funcional.

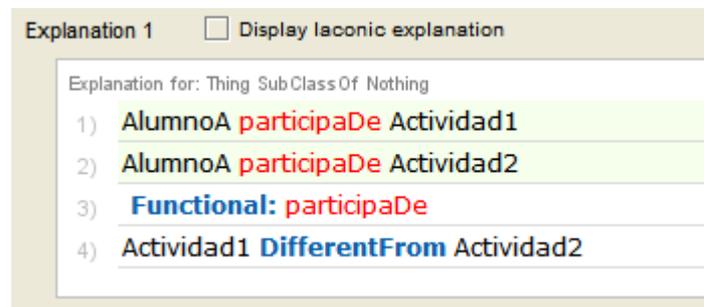


Figura III.79: Explicación de inconsistencia relacionada con participaDe.

**R18.** Una actividad puede estar siendo realizada por un único estudiante.

Partiendo de las instancias definidas en la prueba de R6, se relacionó a la actividad 1 con el alumno A y también con el alumno B (figuras III.80 y III.81).

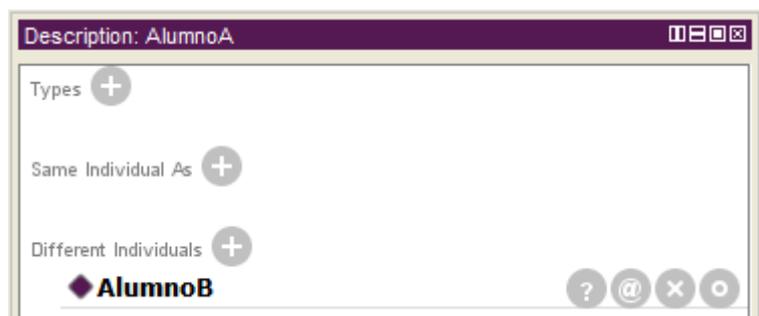


Figura III.80: Los alumnos A y B son individuos diferentes.

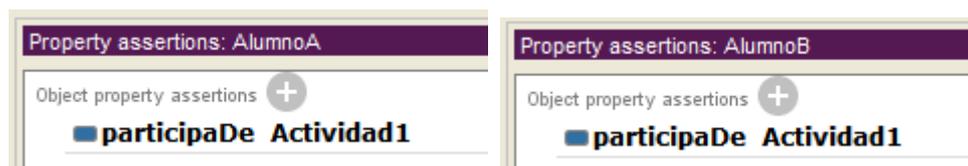


Figura III.81: Tanto el alumno A como el alumno B participan de la Actividad 1.

Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se muestra en la figura III.82, proporcionando la explicación de que la Actividad 1 se relaciona con dos alumnos diferentes, a través de una propiedad que es funcional.

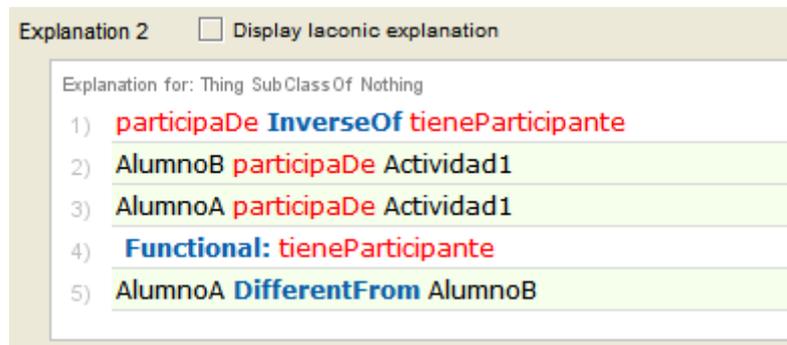


Figura III.82: Explicación de inconsistencia relacionada con tieneParticipante.

**R19.** Una actividad puede desarrollarse en una única ubicación.

Partiendo de las instancias definidas para verificar R5, se relacionó a la actividad 1 con la ubicación 1 y con la ubicación 2, tal como se observa en las figuras III.83 y III.84.

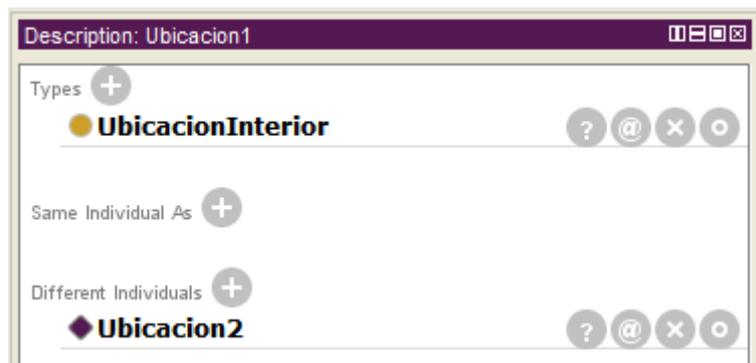


Figura III.83: Ubicación 1 es diferente de Ubicación 2



Figura III.84: Actividad 1 se desarrolla en Ubicación 1 y en Ubicación 2.

Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se observa en la figura III.85, proporcionando la explicación de que la Actividad 1 se relaciona con dos ubicaciones diferentes, a través de una propiedad que es funcional.

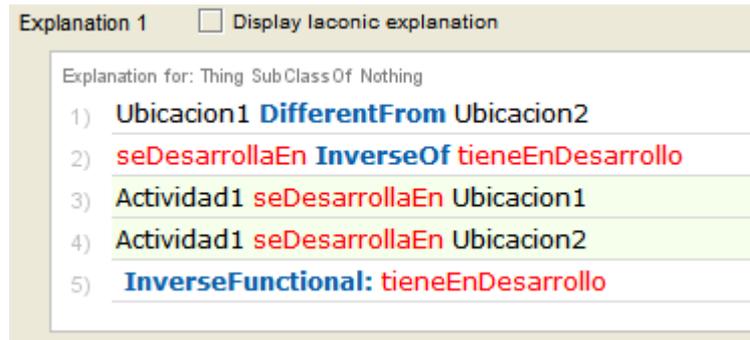


Figura III.85: Explicación de inconsistencia relacionada con tieneEnDesarrollo.

**R20.** *Un estudiante puede hacer uso de varias entidades computacionales simultáneamente.*

Partiendo de las instancias definidas en la prueba de R8, se relacionó al alumno A con las entidades computacionales 1 y 2 (figuras III.86 y III.87).

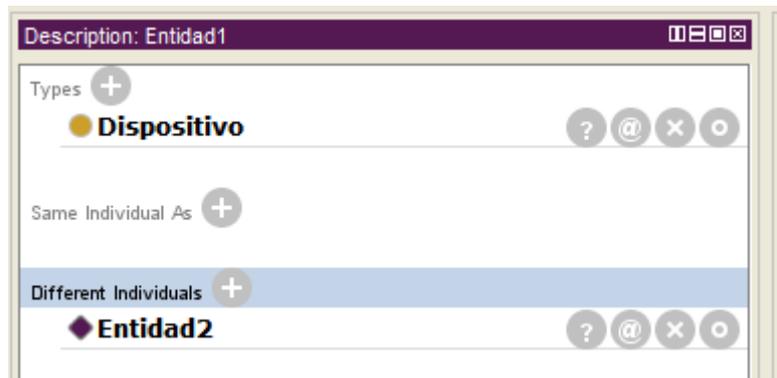


Figura III.86: Entidad 1 es diferente de Entidad 2.

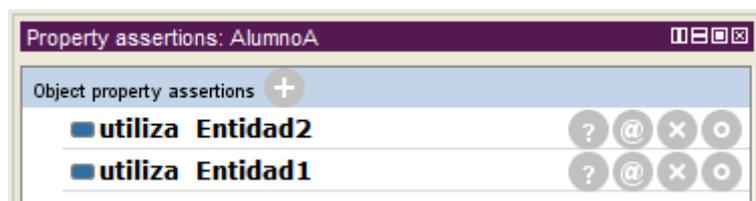


Figura III.87: Alumno A utiliza Entidad 1 y Entidad 2.

Al ejecutar el razonador no se encontraron errores, puesto que es posible que un alumno utilice más de un entidad computacional. Al ejecutar una consulta para determinar todas las entidades utilizadas por el Alumno A, se obtuvieron las Entidades 1 y 2. Estos resultados se muestran en la figura III.88.

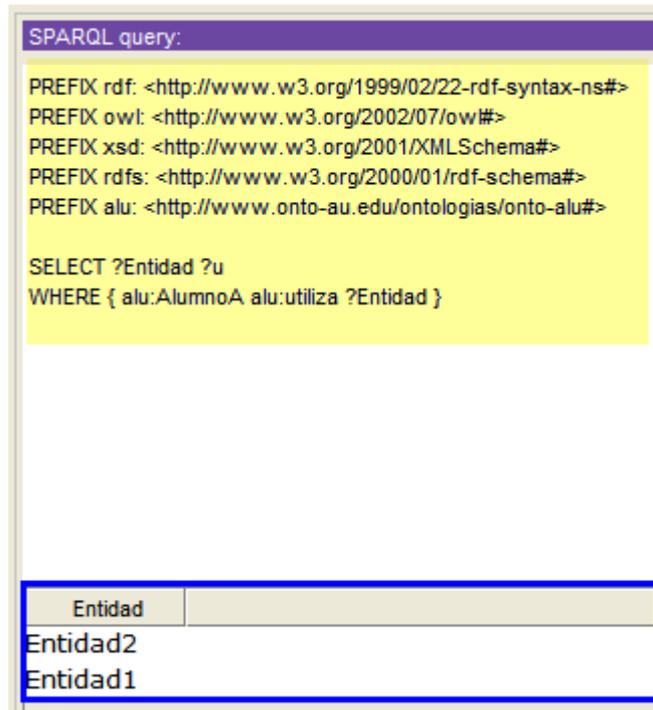


Figura III.88: Todas las entidades utilizadas por el Alumno A.

*R21. Una entidad computacional puede ser utilizada por un único estudiante.*

Partiendo de las instancias definidas en la verificación de R8, se relacionó a los alumnos A y B con la entidad computacional 2, tal como se observa en las figuras III.89 y III.90.

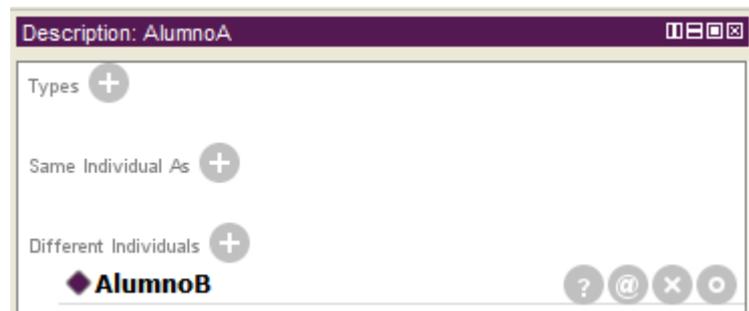


Figura III.89: Alumno A es diferente que Alumno B.



Figura III.90: Entidad 2 es utilizada por Alumno A y por Alumno B.

Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se muestra en la figura III.91, proporcionando la explicación de que la Entidad 2 se relaciona con dos alumnos diferentes, a través de una propiedad que es funcional.

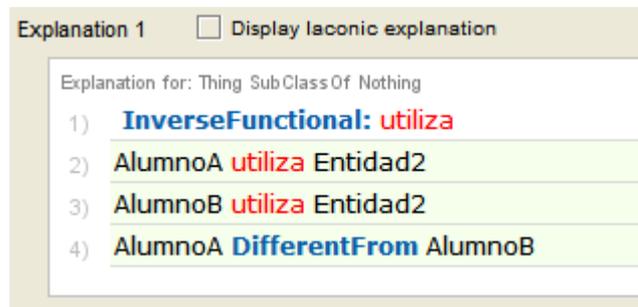


Figura III.91: Explicación de inconsistencia relacionada con utiliza.

**R22.** Una entidad computacional opera en una única ubicación.

Partiendo de las instancias definidas en la prueba de R8, se relacionó a la entidad computacional 1 con las ubicaciones 1 y 2, como se observa en las figuras III.92 y III.93.



Figura III.92: Ubicación 1 es diferente de Ubicación 2.

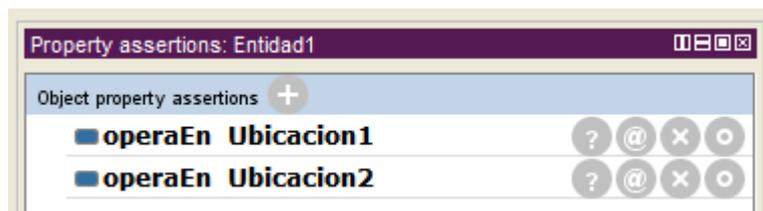


Figura III.93: Entidad 1 opera en Ubicación 1 y 2.

Al ejecutar el razonador, éste determinó que la ontología era inconsistente, tal como se observa en la figura III.94, proporcionando la explicación de que la Entidad 1 se relaciona con dos ubicaciones diferentes, a través de una propiedad que es funcional.

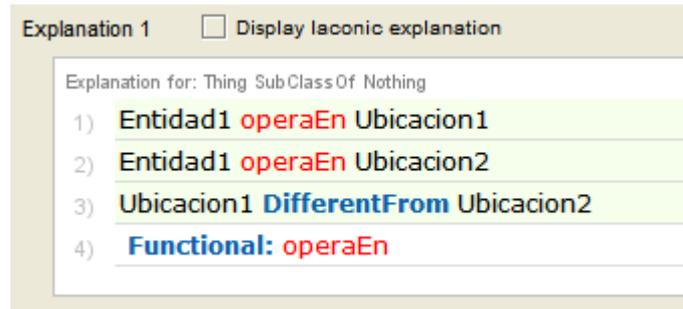


Figura III.94: Explicación de Inconsistencia relacionada con operaEn.

### III.3. CASO DE ESTUDIO: APLICACIÓN UBICUA PARA LA ENSEÑANZA UNIVERSITARIA

En esta sección se presenta la instanciación de la ontología en un escenario concreto: una aplicación de aprendizaje ubicuo de soporte a la enseñanza universitaria.

#### III.3.1. DOMINIO DE LA ONTOLOGÍA

Hay determinadas asignaturas universitarias que incluyen dentro de sus tareas, las prácticas de laboratorio y los trabajos de campo. Estas asignaturas pueden dictarse con una aplicación ubicua de apoyo que será capaz de personalizar su experiencia de aprendizaje.

La aplicación ofrecerá distintos servicios, específicamente diseñados para la implementación de esta clase de cursos. *onto-alu* dará soporte a estos servicios de personalización haciendo uso de la información que contiene y de sus capacidades de razonamiento.

Los servicios originales son 8 y se dividen en 3 categorías, pero para nuestro ejemplo de aplicación se tuvieron en cuenta solamente 2 de ellos:

- **Asesoramiento on-line de pares.-** El estudiante podrá solicitar a la aplicación la posibilidad de contactarse con sus compañeros para realizar una determinada tarea. La aplicación buscará entre los estudiantes on-line aquel que mejor pueda apoyar el aprendizaje del alumno solicitante, en función del grado de avance en la tarea y el nivel de conocimiento.
- **Recomendaciones personalizadas sobre tareas o actividades pedagógicas a realizar.-** La aplicación sugerirá al estudiante realizar una determinada tarea pedagógica del curso en función de su ubicación, su estilo de aprendizaje y el dispositivo con el cual accede a la aplicación.

Para el primer servicio, *onto-alu* determinará un conjunto de candidatos a ayudar al alumno, para lo cual identificará a los estudiantes que tengan resuelta la tarea en la que el alumno necesita ayuda y un nivel de conocimiento en el tema de esa tarea mayor o igual al del alumno que solicita la ayuda. Luego, la aplicación puede seleccionar de entre el listado de candidatos aquel que se encuentre online y recomendarlo al alumno que requiere asistencia.

Para el segundo servicio, *onto-alu* clasificará al estudiante en un estereotipo de acuerdo a las distintas combinaciones de valores que pueden darse entre estilo de aprendizaje (visual o verbal), dispositivo (Smartphone o Tablet) y ubicación (Aula, Laboratorio/Box, Exterior). Luego, la aplicación puede seleccionar la tarea correspondiente en función del estereotipo en el cual ha sido clasificado el alumno.

En función del tipo de soporte que la ontología debe proporcionar, de acuerdo a lo descrito previamente, definimos las preguntas de competencia que la misma debe ser capaz de responder:

**PC1.** ¿Cuáles son los estudiantes que podrían prestar ayuda al alumno en la tarea actual?

**PC2.** ¿A qué estereotipo de recomendación de tarea corresponde el alumno?

### **III.3.2. MODELO EXTENDIDO DE LA ONTOLOGÍA E IMPLEMENTACIÓN**

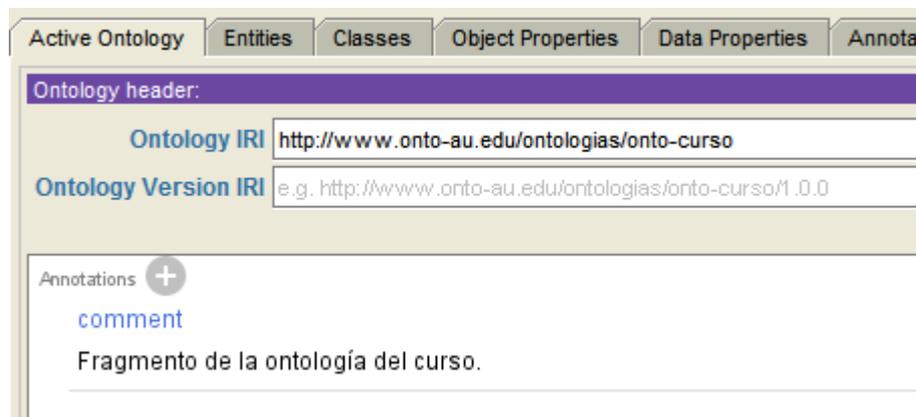
Para que *onto-alu* sea capaz de dar soporte a los servicios descritos, se agregaron nuevas entidades y reglas, además de asumir una estructura particular para el modelo de la estrategia de aprendizaje usada por la aplicación, donde cada curso está compuesto por distintos temas, los cuales a su vez están formados por varias tareas.

Las modificaciones para dar soporte al primer servicio se documentan a continuación:

- Se definió un pequeño fragmento de la ontología *onto-curso*, para representar la estructura mencionada, formada por tres subclases: Curso, Tema y Tarea. Las relaciones para vincular estas entidades son:
  - *esTemaDe*, relaciona un tema con el curso al que pertenece.
  - *esTareaDe*, relaciona una tarea con el tema al que pertenece.
- Se definieron las propiedades necesarias para vincular estas nuevas clases con el estudiante, teniendo en cuenta que una tarea puede estar completada o en progreso y que el estudiante tiene un nivel para cada tema:

- resuelve, relaciona un estudiante con una tarea que éste se encuentra realizando.
- haCompletadoTarea, relaciona un estudiante con una tarea que ya ha sido completada por el mismo.
- tieneNivelAlto, tieneNivelMedio y tieneNivelBajo, son propiedades de objetos que relacionan un estudiante con un tema, indicando su nivel de conocimiento en el mismo.
- Además, se definieron dos propiedades para vincular a los alumnos candidatos con el alumno que requiere asistencia:
  - esCandidatoAyuda: vincula a un alumno que es candidato a ayudar con un alumno que necesita ayuda.
  - tieneCandidatosAyuda: vincula al alumno que necesita ayuda con los alumnos que son candidatos a ayudarlo.
- Finalmente se definió la regla que permite vincular los alumnos candidatos con el alumno que solicita ayuda: *dado un alumno x que realiza una tarea ta que pertenece a un tema t y a un curso c, si el alumno z ha completado la tarea ta y su nivel de conocimiento en el tema t es alto, entonces z es candidato a ayudar a x.*

En la figura III.95 se muestra la IRI de la nueva ontología, y sus clases y propiedades se muestran en la figura III.96.



**Figura III.95:** Creación de un fragmento de *onto-curso*.

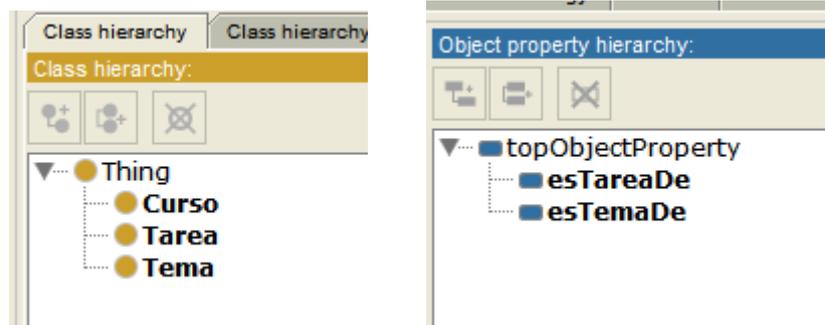


Figura III.96: Clases y propiedades de *onto-curso*.

Para importar esta nueva ontología en la ontología del estudiante, se utilizó el botón *Add* de la pestaña *Ontology imports*, tal como se muestra en la figura III.97.

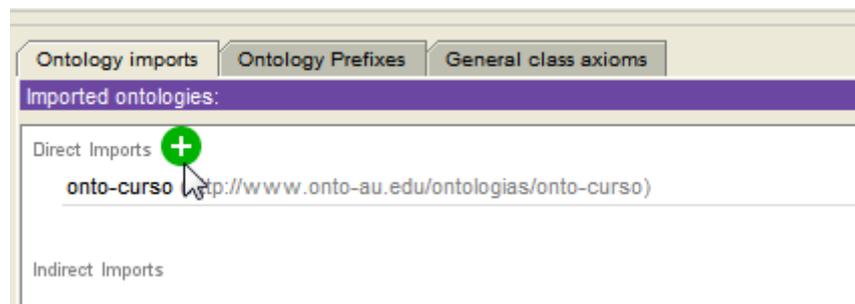
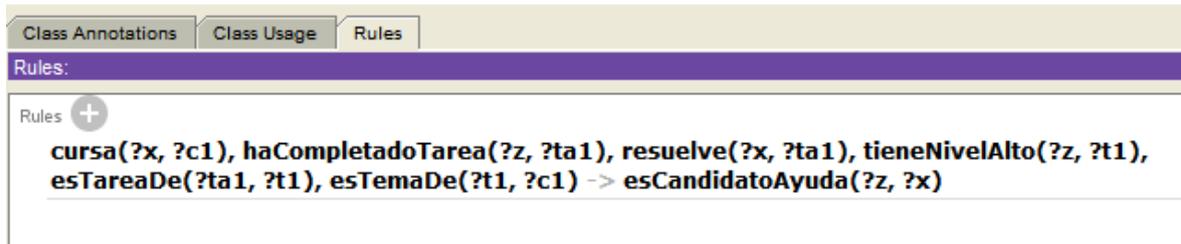


Figura III.97: Importación de la ontología *onto-curso* en *onto-alu*.

En las figuras III.98 y III.99, se muestran las nuevas propiedades de objetos (en amarillo) y la regla, agregadas a la ontología del estudiante, luego de importar *onto-curso*.



Figura III.98: Nuevas propiedades de objetos en *onto-alu*.



**Figura III.99:** Regla para encontrar candidatos de ayuda a un alumno.

Para dar soporte al segundo servicio, se introdujeron los siguientes cambios:

- Se creó una jerarquía de clases para representar los distintos estereotipos de recomendación de tareas, de acuerdo a la primera columna de la tabla III.10.
- Se definieron las condiciones suficientes de pertenencia a cada clase de la jerarquía de estereotipos, de acuerdo con la combinación de valores de las columnas dos, tres y cuatro de la tabla III.10.

**Tabla III.10:** Estereotipos para recomendación de tareas.

Estereotipo	Estilo de Aprendizaje	Dispositivo	Ubicación
ERT1	Visual	Tablet	Aula
ERT2	Visual	Tablet	Laboratorio o Box
ERT3	Visual	Tablet	Exterior
ERT4	Visual	Smartphone	Aula
ERT5	Visual	Smartphone	Laboratorio o Box
ERT6	Visual	Smartphone	Exterior
ERT7	Verbal	Tablet	Aula
ERT8	Verbal	Tablet	Laboratorio o Box
ERT9	Verbal	Tablet	Exterior
ERT10	Verbal	Smartphone	Aula
ERT11	Verbal	Smartphone	Laboratorio o Box
ERT12	Verbal	Smartphone	Exterior

En las figuras III.100 y III.101 se observan la jerarquía de clases de Protégé que representa los distintos estereotipos definidos en la tabla III.10, y la descripción del estereotipo ERT3, respectivamente.

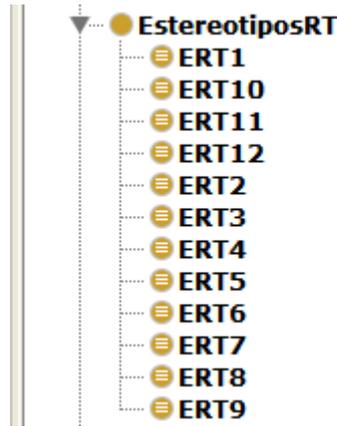


Figura III.100: Jerarquía de clases de estereotipos en Protégé.

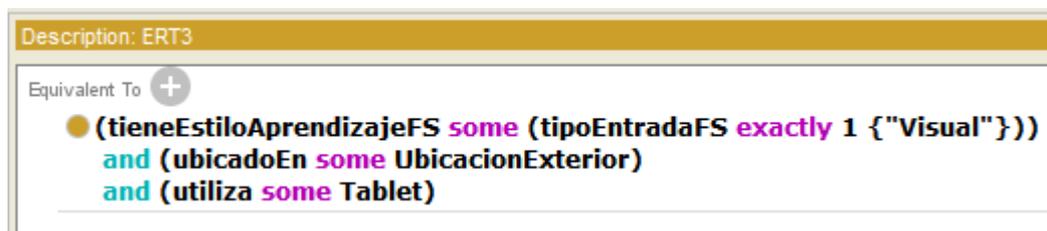


Figura III.101: Condiciones suficientes para el estereotipo ERT3 en Protégé.

### III.3.3. EVALUACIÓN DE LA ONTOLOGÍA CONTRA LAS PREGUNTAS DE COMPETENCIA

*PCI.* ¿Cuál es el estudiante más adecuado para dar soporte al alumno en la tarea actual?

Se crearon distintas instancias de alumnos, temas y tareas y se relacionaron entre sí de distintas formas, para representar las distintas situaciones posibles. El alumno C es quien necesita ayuda. Las relaciones entre estas instancias se pueden observar en la tabla III.11.

Tabla III.11: Ejemplos de alumnos y sus tareas.

	Alumno A	Alumno B	Alumno C
curso	Curso1	Curso1	Curso1
tieneNivelAlto	Tema1	-	-
tieneNivelMedio	-	Tema2	Tema1
tieneNivelBajo	-	Tema1	-
resuelve	Tarea2T1	Tarea1T2	Tarea1T1
haCompletadoTarea	Tarea1T1	Tarea1T1	-
¿Es candidato a ayudar a C?	Si	No	-

Estas mismas instancias, creadas en Protégé se muestran en la figura III.102.

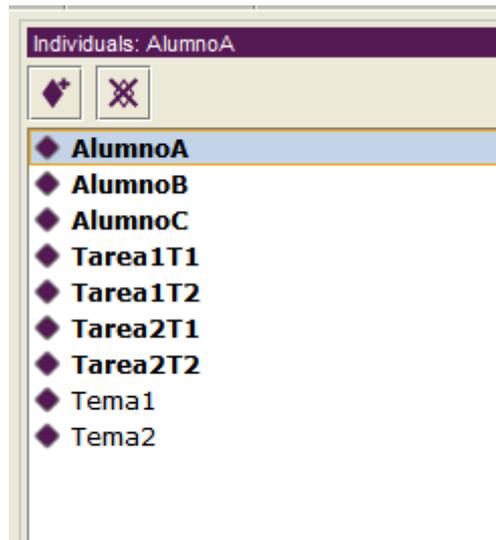


Figura III.102: Instancias de alumnos, temas y tareas.

A continuación se presenta el detalle de las entidades relacionadas con los alumnos A y C (figuras III.103, III.104 y III.105), tal como fueron creadas en la herramienta Protégé. Las relaciones del alumno B se definieron del mismo modo, de acuerdo a los detalles especificados en la tabla III.11.



Figura III.103: Tareas y nivel del tema de Alumno C.



Figura III.104: Tarea1T1 es tarea de Tema1.



Figura III.105: Tareas y nivel de tema del Alumno A.

Al ejecutar el razonador, el alumno A fue relacionado con el alumno C a través de la propiedad esCandidatoAyuda, como se observa en la figura III.106.

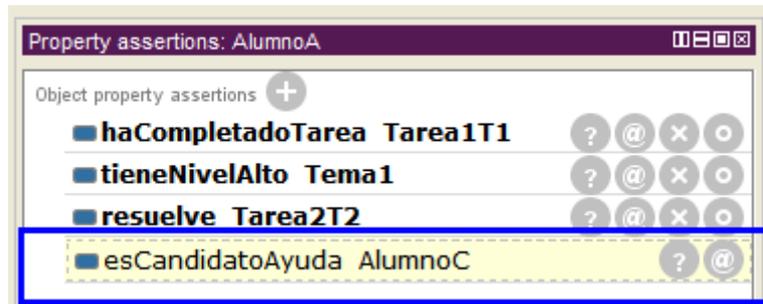


Figura III.106: Alumno A es candidato ayuda Alumno C.

PC2. ¿A qué estereotipo de recomendación de tareas corresponde el alumno?

Se crearon instancias de alumnos que respondían a diferentes estereotipos, en función de la combinación de valores según la tabla III.12.

Tabla III.12: Ejemplos de alumnos y su estereotipo.

	Alumno A	Alumno B	Alumno C	Alumno D	Alumno E
Tipo de entrada preferida según estilo FS	Visual	Verbal	Verbal	Verbal	Verbal
Dispositivo	Tablet	Tablet	Tablet	Smartphone	Smartphone
Ubicación	Aula	Aula	Aula	Exterior	Exterior
Estereotipo corresp.	ERT1	ERT7	ERT7	ERT12	ERT12

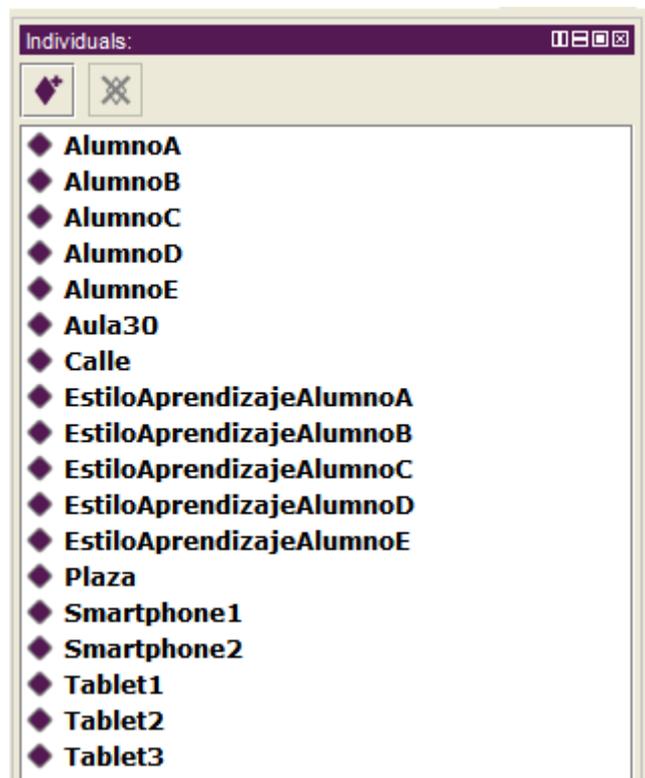


Figura III.107: Instancias de alumnos, ubicaciones, dispositivos y estilos de aprendizaje.

En la figura III.107 se pueden observar las instancias creadas en Protégé, y a continuación se presenta el detalle de las entidades relacionadas con el Alumno A, en las figuras III.108 a III.111. Las demás entidades se definieron del mismo modo, de acuerdo a los detalles especificados en la tabla III.12.



Figura III.108: La instancia Aula30 es de tipo Aula.

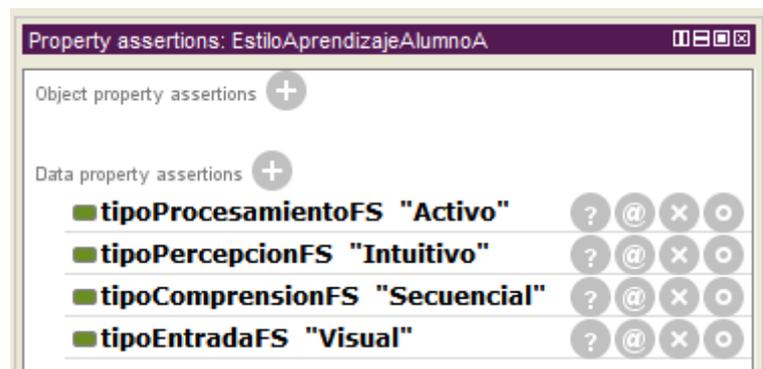


Figura III.109: El estilo de aprendizaje del Alumno A tiene preferencia de entrada Visual.



Figura III.110: El dispositivo Tablet1 es de tipo Tablet.



Figura III.111: Ubicación, dispositivo y estilo de aprendizaje del alumno A.

Al ejecutar el razonador, cada estudiante fue clasificado automáticamente en el estereotipo correspondiente, tal como se puede observar en las figuras III.112, III.113 y III.114.

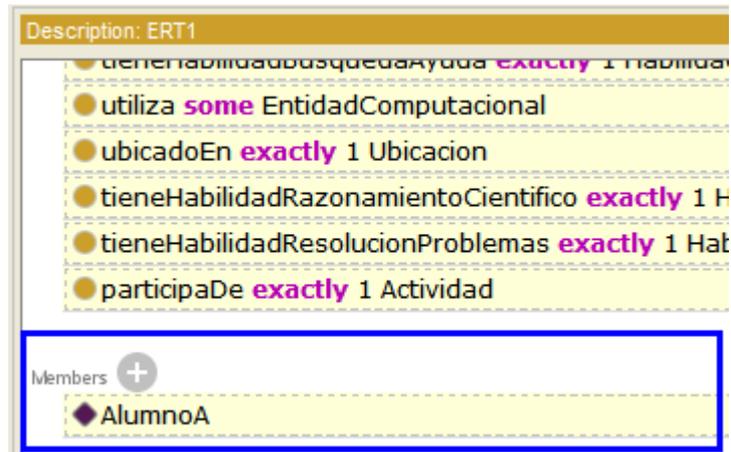


Figura III.112: Alumnos con estereotipo ERT1.

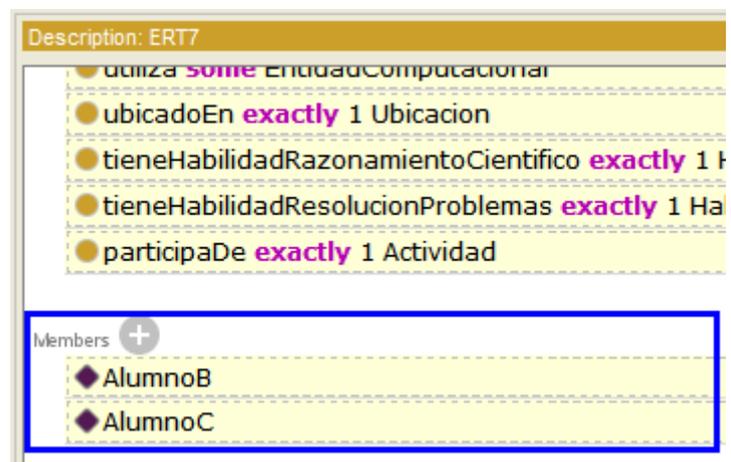


Figura III.113: Alumnos con estereotipo ERT7.

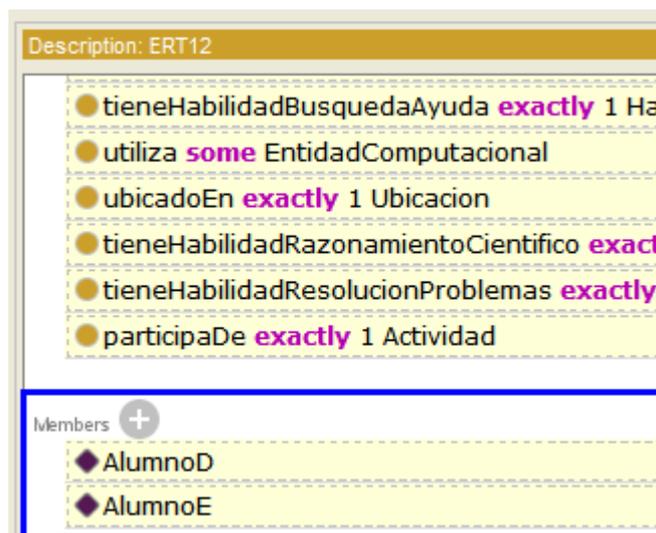


Figura III.114: Alumnos con estereotipo ERT12.

# CONCLUSIONES

---

A continuación se describen las conclusiones obtenidas a partir del presente trabajo.

Con respecto al cumplimiento de los objetivos, se puede decir que se alcanzaron los dos objetivos específicos planteados al inicio del trabajo.

Por una parte, se logró obtener un conjunto de características que representan al estudiante y a su contexto, y que resultan relevantes para la personalización en sistemas de aprendizaje ubicuo. Las mismas poseen un nivel de generalidad tal que resultan válidas para distintos sistemas de la misma clase, a la vez que son lo suficientemente específicas como para poder realizar personalización en función de ellas.

Por otra parte, las mismas se implementaron en una ontología que conformó el modelo del estudiante para sistemas de aprendizaje ubicuo. Este modelo permitió representar la información requerida de manera satisfactoria, tal como se pudo comprobar en la evaluación realizada al mismo. Adicionalmente, se demostró que, al integrar el modelo construido con (una representación de un fragmento de) otro de los modelos que conformarían el sistema de aprendizaje, el mismo permite dar soporte a las tareas de personalización del sistema.

En relación a las ventajas de la utilización de ontologías para representar el modelo del estudiante, se pudo observar que la integración con otros modelos del sistema es directa ya que al encontrarse representados de acuerdo a un formalismo bien definido, la simple importación de una ontología en otra permite seguir trabajando como si todo se encontrara dentro de un mismo modelo. Esto significa que el modelo es altamente reusable y de fácil extensión a través de modelos adicionales.

En relación a las limitaciones presentadas por las ontologías en la representación del modelo, se pudo observar que las mismas no permiten representar cierto tipo de relaciones, lo que en parte se ve resarcido por la incorporación de reglas semánticas en la misma.

Finalmente, se pueden definir las siguientes líneas futuras de investigación:

- Extensión de la ontología a un mayor nivel de detalle y modularización de la misma para mejorar su mantenibilidad y ampliar sus posibilidades de reuso.

- Alineación de la ontología con ontologías de nivel superior para incrementar su interoperabilidad en el contexto de la web semántica y/o de otros sistemas de aprendizaje.
- Integración de la ontología del estudiante con las ontologías del dominio y demás ontologías del sistema de aprendizaje ubicuo para definir las técnicas de personalización.
- Integración del conjunto de ontologías con el sistema de aprendizaje ubicuo para implementar la personalización de la experiencia de aprendizaje.

# BIBLIOGRAFÍA

---

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284(5), 1-5.

Bobillo F., Delgado M. & Gómez-Romero J. (2006) Uso de modelos de restricción del contexto para el desarrollo de aplicaciones móviles inteligentes. *Actas del XIII Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF 2006)*, Noviembre 2006.

Breitman, K., Casanova, M. A.z., & Truszkowski, W. (2007). Semantic Web: Concepts, Technologies and Applications: Concepts, Technologies and Applications. *NASA Monographs in Systems and Software Engineering, 2007, XIV*. Springer.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User modeling and user-adapted interaction*, 6(2-3), 87-129.

Castillo, G., Gama, J., & Breda, A. M. (2003). Adaptive Bayes for a student modeling prediction task based on learning styles. In *User Modeling 2003 (pp. 328-332)*. Springer Berlin Heidelberg.

Cataldi, Z., & Lage, F. J. (2010). Modelado del estudiante en sistemas tutores inteligentes. *TE & ET N° 5, Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*.

Chen, H., Finin, T., & Joshi, A. (2003). An ontology for context-aware pervasive computing environments. *The Knowledge Engineering Review*, 18(03), 197-207.

Chen, H., Perich, F., Finin, T., & Joshi, A. (2004) SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. *The First Annual International Conference on Mobile and Ubiquitous Systems, Networking and Services 2004, MOBIQUITOUS 2004*, 3(5), 258–267. DOI:10.1109/MOBIQ.2004.1331732

Chen, W., & Mizoguchi, R. (1999). Communication content ontology for learner model agent in multi-agent architecture. *Advanced Research in Computers and Communications in Education, Proc. ICCE, 99, 95-102*.

Cross, J. (2004). An informal history of eLearning. *On the Horizon*, 12(3), 103-110.

- Devedžić, V. (2006). Semantic web and education. *Integrated Series in Information Systems, Vol. 12*. Springer.
- Dey, A. K., Abowd, G. D., & Salber, D. (2000). A context-based infrastructure for smart environments. In *Managing Interactions in Smart Environments (pp. 114-128)*. Springer London.
- Fahad, M., & Qadir, M. A. (2008). A Framework for Ontology Evaluation. *ICCS Supplement, 354, 149-158*.
- Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering education, 78(7)*, 674-681.
- Fernández-López, M., Gómez-Pérez, A., & Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. *Proc. Symposium on Ontological Engineering of AAAI, (1997)*.
- Fröschl, C. (2005). User modeling and user profiling in adaptive e-learning systems. Master Thesis. *Graz University of Technology: Austria*.
- Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). Ontological engineering. *Advanced Information and Knowledge Processing, 2004 XII, Vol. 139*. Heidelberg: Springer.
- Graf, S., Viola, S. R., & Leo, T. (2007). In-Depth Analysis of the Felder-Silverman Learning Style Dimensions. *Journal of Research on Technology in Education, 40(1)*.
- Graf, S. & Kinshuk. (2008). Adaptivity and Personalization in Ubiquitous Learning Systems. In *HCI and Usability for Education and Work (pp. 331-338)*. Springer. Berlin Heidelberg.
- Graf, S., Yang, G., Liu, T. C., & Kinshuk, D. (2009). Automatic, global and dynamic student modeling in a ubiquitous learning environment. *Knowledge Management & E-Learning: An International Journal (KM&EL), 1(1), 18-35*.
- Han, B. (2001). Student modelling and adaptivity in web-based learning systems. Master's Thesis. *Massey University: New Zealand*.
- Heckmann, D., Schwarzkopf, E., Mori, J., Dengler, D., & Kr, A. (2006). The User Model and Context Ontology GUMO revisited for future Web 2.0 Extensions. *Context, 37*.
- Heines, J. M. (1988, November). Milestones in Early Learning Devices. *coAction Magazine, Authorware. 1(1):24-29*.
- Hervás, R., Nava, S. W., Chavira, G., & Bravo, J. (2006). Modelado de Contexto : Una Ontología Adaptativa al Usuario en Ambientes Inteligentes. *Context, 1, 167-177*.

Hong, M., & Cho, D. (2008). Ontology Context Model for Context-Aware Learning Service in Ubiquitous Learning Environments. *International Journal*, 2(3). 844–848.

Hwang, G.-J., Tsai, C.-C., & Yang, S. J. H. (2008). Criteria, Strategies and Research Issues of Context-Aware Ubiquitous Learning. *Educational Technology & Society*, 11 (2), 81-91.

Jovanovic, J., Gašević, D., & Devedžić, V. (2009). TANGRAM for personalized learning using the semantic web technologies. *Journal of emerging technologies in web intelligence*, 1(1), 6-21.

Koch, N. P. (2001). Software engineering for adaptive hypermedia systems. Doctoral dissertation, PhD Thesis. Verlag Uni-Druck, Munich.

Martins, A. C., Faria, L., De Carvalho, C. V., & Carrapatoso, E. (2008). User Modeling in Adaptive Hypermedia Educational Systems. *Journal of Educational Technology & Society*, 11(1).

Montgomery, S. M., & Groat, L. N. (1998). Student learning styles and their implications for teaching. *Occasional paper*, (10).

Mödrischer, F., Garcia Barrios, V. M., & Gütl, C. (2004). Enhancement of SCORM to support adaptive E-Learning within the Scope of the Research Project AdeLE. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, Vol. 2004, No. 1, pp. 2499-2505.

Noy, N., & McGuinness, D. L. (2001). Ontology Development 101. *Knowledge Systems Laboratory*, Stanford University.

Obitko, M. (2001). Ontologies--Description and Applications. In *Department of Czech Technical University*.

Otero, W. R. I., Petch, J. R., & Catapan, A. H. (2012). Developing high-level cognitive skills in e-learning. *Rev Cientif Int*, 20, 41-57.

Panagiotopoulos, I., Kalou, A., Pierrakeas, C., & Kameas, A. (2012). An Ontology-Based Model for Student Representation in Intelligent Tutoring Systems for Distance Learning. In *Artificial Intelligence Applications and Innovations* (pp. 296-305). Springer Berlin Heidelberg.

Pramitasari, L., Hidayanto, A. N., Aminah, S., Krisnadhi, A. A., & Ramadhania, A. M. (2009). Development of student model ontology for personalization in an e-learning system based on semantic web. In *International Conference on Advanced Computer Science and Information Systems (ICACISIS09)*, Indonesia, December (pp. 7-8).

Roll, I., Baker, R. S., Alevan, V., McLaren, B. M., & Koedinger, K. R. (2005). Modeling students' metacognitive errors in two intelligent tutoring systems. In *User Modeling 2005* (pp. 367-376). Springer Berlin Heidelberg.

Sanz, I., & Jiménez-Ruiz, E. (2009). Ontologías en Informática. *Terminología y sociedad del conocimiento*, 255-286.

Schmidt, A., Beigl, M., & Gellersen, H. W. (1999). There is more to context than location. *Computers & Graphics*, 23(6), 893-901.

Shearer, R., Motik, B., & Horrocks, I. (2008, October). Hermit: A Highly-Efficient OWL Reasoner. In *OWLED* (Vol. 432).

Siadaty, M., Torniai, C., Gašević, D., Jovanovic, J., Eap, T. M. & Hatala, M. (2008). m-LOCO : An Ontology-based Framework for Context-Aware Mobile Learning. *Proceedings of the 6th International Workshop on Ontologies and Semantic Web for Intelligent Educational Systems at 9th International Conference on Intelligent Tutoring Systems*, Montreal, Canada, June 2008.

Šimún, M., Andrejko, A., & Bielíková, M. (2007, September). Ontology-based models for personalized e-learning environment. In *Proc. of 5th Int. Conf. on Emerging e-learning Technologies and Applications (ICETA '07)*.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2), 51-53.

Smythe C., Tansey F, & Robson R. (2001). IMS Learner Information Package Information Model Specification, Version 1.0.

Staab, S., & Studer, R. (2010). Handbook on ontologies. *International Handbooks on Information Systems*, 2nd ed. 2009, XIX. Springer.

Sosnovsky, S. (2007). Ontological Technologies for User Modeling. *International Journal of Metadata, Semantics and Ontologies*. Vol. 5(1), pp 32-71.

Studer, R., Benjamins, V. R., & Fensel, D. (1998). *Knowledge engineering: principles and methods*. *Data & knowledge engineering*, 25(1), 161-197.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(02), 93-136.

Wang, X. H., Zhang, D. Q., Gu, T., & Pung, H. K. (2004, March). Ontology based context modeling and reasoning using OWL. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on* (pp. 18-22). IEEE.

Weiser, M. (1991). The computer of the 21st century. *Scientific American*, 265(3), 94-104. DOI:10.1145/329124.329126.

Yau, J. K., & Joy, M. S. (2006). Context-aware and adaptive learning schedule for mobile learning. In: *International Workshop on Mobile and Ubiquitous Learning Environments (MULE) at the International Conference on Computers in Education (ICCE 2006)*, Beijing, China.