

SISTEMA WEB SECUENCIADOR DE EJERCICIOS DE PROGRAMACIÓN DE COMPUTADORAS USANDO ONTOLOGÍAS

USE OF ONTOLOGIES IN WEB SYSTEM SEQUENCER OF COMPUTER PROGRAMMING EXERCISES

LUIS VLADIMIR URRELO HUIMAN¹

Resumen

El aprendizaje y práctica de la programación de computadoras debe estar guiado por una adecuada secuencia de ejercicios que eviten la temprana frustración sin dejar de motivar al estudiante. En tal sentido, es necesario la clasificación y asignación de ejercicios de manera automática a grupos de estudiantes con diferentes niveles de asimilación. Así, Castro Lozano (2005), en su investigación denominada "Sistema de Desarrollo Integrado para Cursos Hipermedia Adaptativos", muestra la posibilidad de desarrollar una aplicación web para dicho fin. Pero las investigaciones demuestran que es necesario utilizar un componente de los sistemas de inteligencia artificial denominado motor de inferencia, cuyo desarrollo está soportado por lenguajes y arquitecturas complejas. En tal sentido, el presente trabajo tiene como objetivo implementar un sistema web secuenciador de ejercicios de programación de computadoras usando ontologías para el desarrollo del motor de inferencia, basándose en una arquitectura de software simplificada y soportada por estándares publicada por Urrelo Huiman (2010). Para lograrlo se implementó una capa de persistencia de datos basada en ontología, una capa de servicios mediante Objetos Action Script manejadores de la ontología y una capa de presentación compuesta por nodos SWF; luego se desarrolló el motor de inferencia basado en ontología, donde se especifican las reglas de secuenciamiento y se comprobó su facilidad y estandarización para futuros desarrollos.

Palabras clave

Sistema web secuenciador | ontologías | ejercicios de programación de computadoras

Abstract

Learning and practice of computer programming should be guided by an appropriate sequencing of exercises to avoid early frustration while motivating the student (Zuleta & Chaves Medina Torres, 2011), in this sense, classification and mapping exercise is required automatically, to groups of students with different levels of assimilation. So (Castro Lozano, 2005), in his research called "Integrated Development System for Adaptive Hypermedia Courses", demonstrates the possibility of developing a web application for this purpose. But research shows that it is necessary to use a component system called artificial intelligence inference engine, whose development is supported by languages and complex architectures. As such, this paper aims to implement a web system sequencer computer programming exercises using ontologies for developing the inference engine, based on simplified software architecture and supported by published standards (Urrelo Huiman, 2010). To achieve a persistence layer ontology data base, a layer of services using action script objects ontology handlers and a presentation layer consists of nodes SWF is implemented, then the inference engine based on ontology, which specifies developed sequencing rules and ease and standardization for future developments are found.

Keywords

Web system sequencer | ontologies | computer programming exercises.

¹ Doctor docente de la Facultad de Ingeniería de la UPAO en la Escuela Profesional de Ingeniería de Computación y Sistemas. Área de investigación: Web semántica.

I. Introducción

El aprendizaje y práctica de la programación de computadoras están basados en resolver ejercicios de programación de diferentes niveles de complejidad. Esto para no generar frustración en un alumno que no logra resolver ejercicios iniciales con éxito y buscando no desperdiciar el potencial de alumnos con base y conocimientos previos. Este contexto coloca en un problema al docente y es el de la clasificación y asignación de ejercicios a un grupo naturalmente heterogéneo de estudiantes de manera rápida y apropiada. Así, investigaciones como las de Castro Lozano (2005) con el “Sistema de Desarrollo Integrado para Cursos Hipermedia Adaptativos”, nos indican que las tecnologías de información permiten automatizar la clasificación y asignación de ejercicios de programación (Lescano, 2013; Arellano Pimentel, Nieva García, Solar González & Arista López, 2008); pero a costa de un motor de inferencia complejo (Urrelo Huiman, 2010). Entonces:

1.1. Enunciado del problema

¿Cómo desarrollar un motor de inferencia que permita la implementación de un sistema secuenciador de ejercicios de programación de computadoras utilizando tecnología Web estándar?

1.2. Hipótesis

Un sistema web basado en ontología permitirá la generación de un motor de inferencia estándar en línea para realizar la secuencia de ejercicios de programación de computadoras.

En tal sentido se plantea el siguiente objetivo general:

1.3. Objetivo general

Desarrollar un módulo de un sistema web secuenciador de ejercicios de programación de computadoras basado en una ontología como motor de inferencia estándar en línea.

1.3.1. Objetivos específicos

- a) Realizar la ingeniería de requerimientos para el sistema web secuenciador de ejercicios de programación en el curso de Programación de Aplicaciones Hipermedia y Mundos Virtuales, de la Escuela Profesional de Ingeniería de Computación y Sistemas según la metodología RUP y la extensión de casos de uso para el secuenciamiento adaptativo.
- b) Analizar y diseñar un sistema web basado en ontología y secuenciador de ejercicios de programación, aplicando las metodologías: RUP, OOHDM, WEBML y la propuesta de Alvarado (2010) para el análisis y diseño de la ontología.
- c) Implementar un módulo de un sistema web secuenciador de ejercicio de programación en el curso de Programación de Aplicaciones Hipermedia y Mundos Virtuales, de la Escuela Profesional de Ingeniería de Computación y Sistemas, basado en ontología y utilizando Protégé y NetBeans.
- d) Probar la estandarización del motor de inferencia en línea generado en el sistema web secuenciador

de ejercicios de programación de computadoras, utilizando la prueba de MECABIT y con juicio de expertos.

1.4. Antecedentes

Las siguientes investigaciones forman parte de los antecedentes:

- C. de Castro Lozano, Dpto. de Informática, E. García Salcines, C. Romero Morales, S. Ventura Soto. *Sistema de desarrollo integrado para cursos hipermedia adaptativos* (INDESAHC), 2005.
Se llegó a plantear un modelo de web adaptativa para la elaboración de cursos en donde se administra los temas por niveles; sin embargo, la arquitectura de la aplicación es compleja, en el sentido de la necesidad de utilizar aplicaciones específicas (HAMWEB) e NDESAHC (Integrated Development System for Adaptive Hypermedia Courses).
- Miguel Montero López. *Sistemas de enseñanza adaptativos: experiencias de PDinamet en la enseñanza de la física*. Departamento de Inteligencia Artificial de la E.T.S.I Informática UNED. Madrid, 2006.
Se plantea una aplicación adaptativa (PDinamet) como solución sin hacer extensible su arquitectura.

1.5. Aportes

El proyecto realiza los siguientes aportes:

- Permite sentar las bases de investigación en la ingeniería Web basada en ontología y las maneras de desarrollar motores de inferencia estándares y en línea.
- Concientiza a la población docente sobre el rol que juega la tecnología de la información en la educación de hoy.

II. Marco teórico

En la presente sección se exponen conceptos relacionados con la Web secuenciadora de ejercicios de programación de computadoras basada en ontologías.

2.1. Sistema web secuenciador

Es un sistema que en función de una serie de variables, representativas de suposiciones y preferencias, responden a diferentes perfiles de usuarios y muestran un mecanismo de adaptación o secuenciamiento. Estructuralmente, estos sistemas están formados por los siguientes elementos:

- Recogedor de información explícita e implícita del usuario.
- Motor de inferencia.
- Gestor de contenido.

El motor de inferencia es el elemento clave del sistema, analiza las variables de entorno y los perfiles de usuario para decidir qué tipos de contenido y en qué formato se mostrarán ante una determinada petición de acceso.

Existen múltiples tecnologías para el desarrollo de un motor de inferencia como pueden ser la lógica difusa, el razonamiento basado en casos, los algoritmos genéticos, las redes neuronales, los multiagentes de software y las ontologías, entre otras (Urrelo Huiman, 2010).

En la siguiente tabla se muestra una comparación de sus características más relevantes:

Tabla 1
ATRIBUTOS DE CALIDAD QUE PERMITEN LAS TECNOLOGÍAS PARA EL DESARROLLO DEL MOTOR DE DECISIÓN.

Característica	Lógica difusa	Inteligencia artificial con el razonamiento basado en casos	Algoritmos genéticos	Redes neuronales	Multiagentes de software	Ontologías
Atributos de calidad observables						
Disponibilidad	sí	sí	sí	sí	sí	sí
Confidencialidad	sí	sí	sí	sí	sí	no
Desempeño	media	media	alta	media	alta	alta
Confiabilidad	sí	sí	sí	no	no	no
Seguridad externa	media	media	alta	alta	media	baja
Seguridad interna	baja	baja	alta	alta	alta	alta
Atributos de calidad no observables						
Estandarización	no	no	no	no	no	sí
Configurabilidad	media	media	baja	baja	baja	alta
Integrabilidad	media	media	baja	baja	media	alta
Interoperabilidad	media	media	media	alta	alta	alta
Modificabilidad	media	media	baja	baja	baja	alta
Mantenibilidad	media	media	baja	baja	baja	alta
Portabilidad	baja	baja	baja	baja	baja	alta
Reusabilidad	media	media	baja	media	media	alta

Por su alto nivel de estandarización, portabilidad y desempeño la ontología sería la tecnología apropiada.

2.1.1. Ontologías

La Web semántica necesita ontologías (Alberto Gómez, 2007) con alto grado de estructura para especificar descripciones de conceptos, según lo indican Ángel Alpuche (en Alpuche & Rodríguez, 2008) y Francisco García (en García Peñalvo, 2007). Esto implica que toda ontología desarrolla un modelo abstracto del dominio o fenómeno del mundo que representa. Dicho modelo abstracto se basa esencialmente en el empleo de conceptos, atributos, valores, relaciones, funciones, etc., definidas explícitamente. (Leal Labrada, 2006; Arano, 2010; García Peñalvo, 2007).

Pero las ontologías pueden implementarse con diferentes técnicas de modelado de conocimiento, tales como:

- Lógica proposicional. Una proposición es una sentencia que puede decirse que es falsa o verdadera, en la lógica proposicional se asignan símbolos a cada sentencia y se utilizan operadores lógicos sobre ellos para crear proposiciones más complejas. Los símbolos utilizados son and, or, not, implies y equivalence.
- Lógica de primer orden. Es una ampliación de la lógica proposicional, que utiliza dos operadores más: el cuantificador universal y el existencial. Utiliza también símbolos para representar conocimiento y operadores lógicos para construir sentencias más complejas; pero, a diferencia de la lógica proposicional, los símbolos pueden representar constantes, variables, predicados y funciones.
- Lógica descriptiva. Se basa en representar el conocimiento utilizando una terminología o vocabulario del dominio (TBOX) y un conjunto de afirmaciones (ABOX) (Baader, Calvanese, McGuinness, Nardi, & PatelSchneider, 2002).

2.1.2. Motores de inferencia

Con el conocimiento descrito en base al modelo conceptual definido por el esquema de una ontología y un conjunto de reglas descritas utilizando los elementos de la ontología, se pueden ejecutar procesos de inferencia para conseguir deducir nuevo conocimiento o procesar alguna consulta realizada por un humano o aplicación externa. Implementaciones reales de este tipo de sistemas para SWRL son el SWRL Rule Engine del módulo SWRLTAB de la herramienta Protege, Bossam, Hoolet, Pellet, KAON2, Racer o SweetRules, entre otras herramientas con formatos de reglas propios (Soto Carrión, 2008).

2.1.3. Metodología para el diseño de ontologías

Según Alvarado (2010), existen varios métodos para diseñar y construir ontologías, como por ejemplo el método de desarrollo de ontologías, de Uschold y King; el método virtual empresarial Toronto (TOVE), de Gruninger y Fox; la Methontology KAKTUS y el método de desarrollo de ontologías propuesto por Noy y McGuinness; sin embargo, tomando como base las metodologías de Noy & McGuinness y Methontology Rubén Darío Alvarado (Alvarado, 2010) propone una metodología de cinco fases:

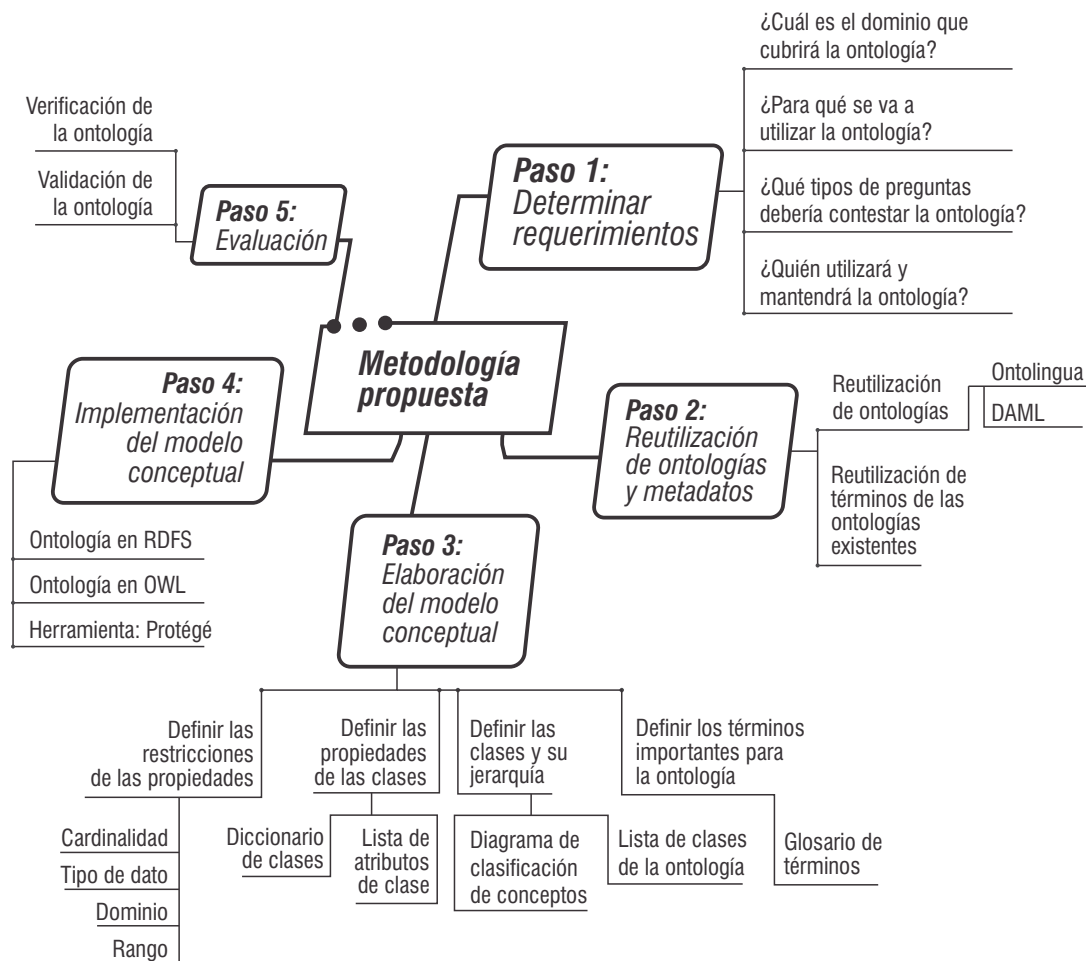


Figura 1: Descripción de la metodología propuesta por Rubén Darío Alvarado
Fuente: (Alvarado, 2010)

III. Material y métodos

En este apartado se plantean las técnicas, métodos e instrumentos utilizados en la presente investigación.

3.1. Material y procedimiento

Material

Software: Protege, NetBeans, Jena 2.0

Tipo de investigación

Tecnológica aplicada.

Técnica

Análisis de la literatura.
Experimentación.

Prueba de hipótesis

Mediante un test de complejidad y prueba de estandarización de un módulo de un sistema informático web secuenciador de ejercicios de programación de computadoras basado en ontología.

El procesamiento de datos

Mediante estadística descriptiva aplicada a los resultados del test de estandarización y complejidad.

Análisis e interpretación de resultados

El análisis y la interpretación de los resultados se realizan a través de la prueba de MECABIT.

3.2. Metodología

Para el desarrollo de la presente investigación se utilizó la metodología conformada de las siguientes etapas.

- Análisis del estado del arte de las web secuenciadoras y motores de inferencia.
- Cubrir la etapa de ingeniería de requerimientos. Para estudiar la secuencia de ejercicios de programación en el curso de Programación de Aplicaciones Hipermedia y Mundos Virtuales, de la Escuela Profesional de Ingeniería de Computación y Sistemas de la UPAO semestre académico 2013-10.
- Análisis y diseño de una aplicación web secuenciadora de ejercicios de programación en el curso de Programación de Aplicaciones Hipermedia y Mundos Virtuales, de la Escuela Profesional de Ingeniería de Computación y Sistemas de la UPAO, aplicando las metodologías: RUP, OOHDM y WEBML
- Implementación de un módulo del sistema web secuenciador de ejercicios de programación en el curso de Programación de Aplicaciones Hipermedia y Mundos Virtuales, basada en ontología utilizando Protégé y NetBeans.
- Obtención de los resultados en base a pruebas de complejidad y estandarización del software.

IV. Desarrollo de un módulo de web secuenciador de ejercicios de programación de computadoras

Aplicado al curso de Programación de Aplicaciones Hipermedia y Mundos Virtuales, de la Escuela Profesional de Ingeniería de Computación y Sistemas,

de la Universidad Privada Antenor Orrego, probado el semestre académico 2013-10.

4.1. Fase I: descripción del caso

En el curso existe una heterogeneidad de conocimientos previos entre los participantes y en las horas de laboratorio. Hay la posibilidad de plantear ejercicios de programación en Java y Action Script 3.0, en donde cada resolución correcta del ejercicio planteado en cualquiera de sus modalidades hace que su característica de complejidad disminuya, y ante la solución planteada por un participante el nivel de habilidad de incrementa en razón al tiempo de desarrollo, si lo hizo de manera correcta o no y según el criterio de complejidad del ejercicio resuelto.

4.2. Fase II: requerimientos

En la presente fase se identifican los requerimientos, extendiéndolos para que puedan soportar la funcionalidad secuenciadora, siendo estos: validación de usuario, listado de cursos por alumno, listado de temas por curso, realización de prácticas, realización de ejercicios (previa consulta del perfil), obtención de resultado (generando actualización del perfil del alumno y del nivel de complejidad del ejercicio) y, finalmente, reportando las evaluaciones y avance.

4.2.1. Elaboración del modelo de dominio

El modelo de dominio es una vista estática de objetos que existen (reales) relacionados con el proyecto y las relaciones que hay entre ellos.

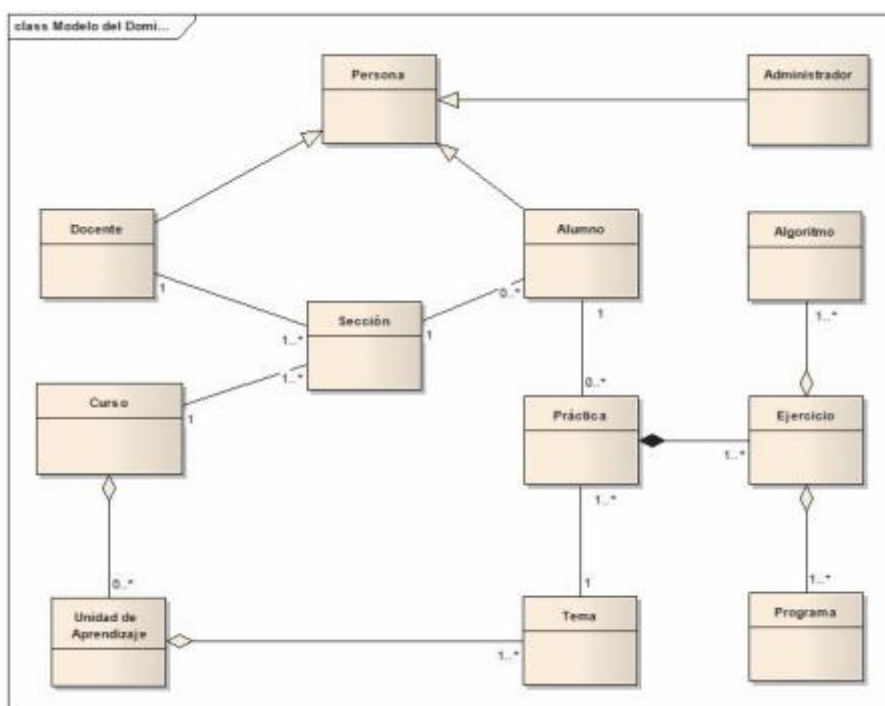


Figura 2: Modelo del dominio.

4.3. Fase III: análisis de la ontología

4.3.1. Determinar los requerimientos de la ontología (el dominio y alcance de la ontología)

Para iniciar el desarrollo de una ontología tenemos que definir su dominio y alcances respondiendo a algunas preguntas básicas como: ¿Cuál es el dominio que la ontología cubrirá?

La ontosecuencia permitirá representar el esquema de asignación de ejercicio de programación en una práctica perteneciente a un tema de una unidad en el curso de Programación de Aplicaciones Hipermedia y Mundos Virtuales, de la Escuela Profesional de Ingeniería de Computación y Sistemas, de la Universidad Privada Antenor Orrego.

¿Para qué usaremos la ontología?

Esta ontología facilitará la asignación de un ejercicio de programación de un tema determinado, teniendo en cuenta el perfil del alumno y el nivel de complejidad del ejercicio, así como su estado. Naturalmente, será necesario la inclusión de los conceptos que describen: al curso, unidad de aprendizaje, el tema así como la práctica.

¿Qué preguntas debería contestar la ontología?

- ¿Qué ejercicio asignar a un alumno con un perfil determinado?
- ¿Cuál es el perfil del alumno?

¿Quién utilizará y mantendrá la ontología?

Los usuarios potenciales de esta ontología son los participantes y docentes de la Escuela Profesional de Ingeniería de Computación y Sistemas, de la Universidad Privada Antenor Orrego.

Los administradores de la ontología serán las personas encargadas del mantenimiento de la misma.

4.3.2. Elaboración del modelo conceptual

Es el modelo que representa la jerarquía de clases de la ontología, así como, las relaciones que existen entre éstas.

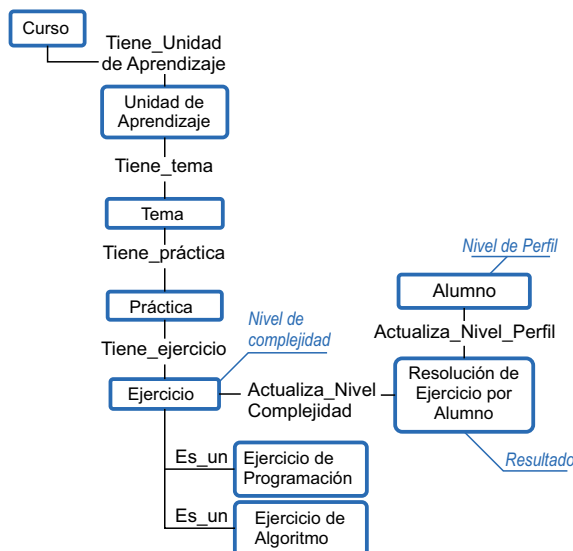


Figura 3: Modelo conceptual de la ontología.

4.4. Fase IV: diseño

En la presente fase se realiza el diseño del sistema web secuenciador y para ello se elabora el modelo de composición WebML, llegando a elaborar el modelo de secuencia.

4.4.1. Modelo de composición WEB ML

Muestra, soportada en el lenguaje de modelado Web y sus estereotipos, los componentes y su funcionalidad sobre las capas de esquema de contexto navegacional.

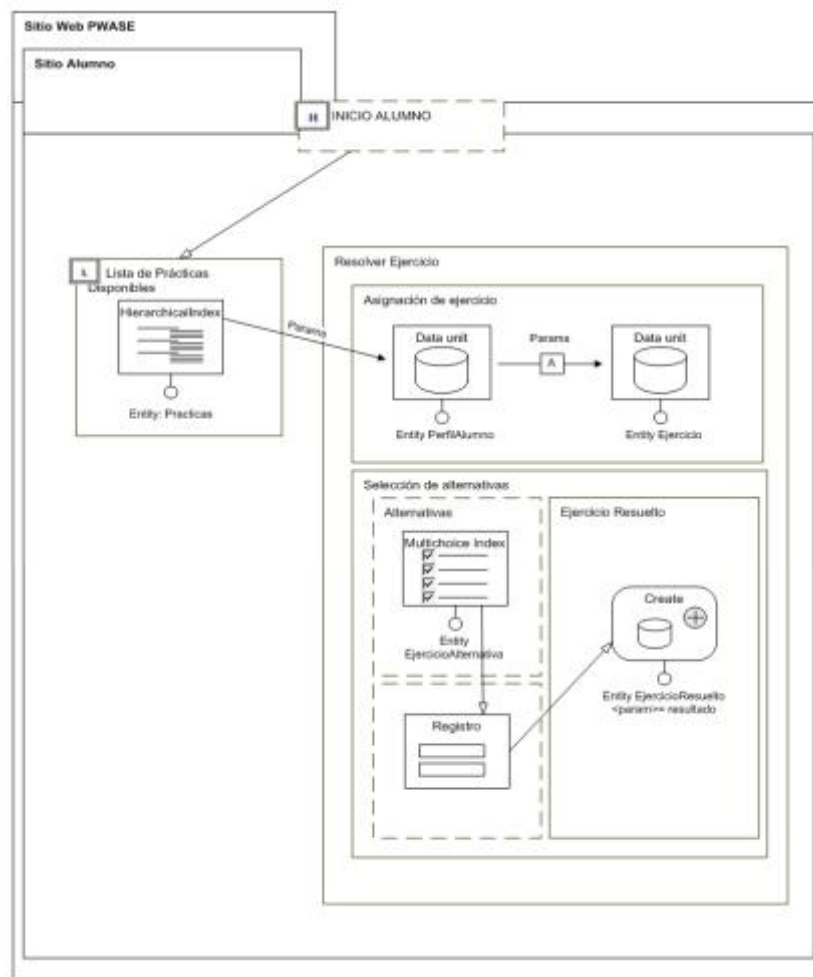


Figura 4:
Modelo de Componente WebMI Realizar Ejercicios

4.5. Fase V: implementación

La implementación se realizó usando la suite de ADOBE CS5, NetBeans con Apache Tomcat y Protégé.

4.5.1. Implementación del motor de inferencia

Para la implementación del motor de inferencia, que en nuestro caso es una ontología basada en el OWL, utilizamos Protégé en su versión 3.4.

Implementándose los axiomas

```

<owl:Class rdf:ID="Curso">
  <owl:equivalentClass>

  //En esta parte se especifica la restricción generada por el Axioma
  <owl:Restriction>
    <owl:allValuesFrom>
      <owl:Class rdf:ID="UnidadAprendizaje"/>
    </owl:allValuesFrom>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="Tiene_un"/>
    </owl:onProperty>
  </owl:Restriction>
</owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Alumno">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Persona"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:maxCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int" >1</owl:maxCardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="NivelAsimilacion"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith>
    <owl:Class rdf:ID="Docente"/>
  </owl:disjointWith>
</owl:Class>
    
```

Así como las reglas de inferencia en SWRL

```

<swrl:Imp rdf:ID="Rule-1">
<swrl:body>
<swrl:AtomList>
<rdf:rest>
<swrl:AtomList>
<rdf:first>
<swrl:IndividualPropertyAtom>
<swrl:argument1>
<swrl:Variable rdf:ID="y"/>
</swrl:argument1>
<swrl:propertyPredicate rdf:resource="#Actualiza_nivel_complejidad"/>
<swrl:argument2 rdf:resource="#z"/>
</swrl:IndividualPropertyAtom>
</rdf:first>
<rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
</swrl:AtomList>
</rdf:rest>
<rdf:first>
<swrl:IndividualPropertyAtom>
<swrl:argument2 rdf:resource="#y"/>
<swrl:propertyPredicate rdf:resource="#Realiza"/>
<swrl:argument1 rdf:resource="#x"/>
</swrl:IndividualPropertyAtom>
</rdf:first>
</swrl:AtomList>
</swrl:body>
</swrl:Imp>
    
```

V. Discusión

5.1. Cualidades evaluadas en el motor de inferencia propuesto

5.1.1. Cualidades observables vía ejecución, evaluadas por expertos en ingeniería de software
 Para obtener las siguientes puntuaciones de atributos de calidad del motor de inferencia basado en ontología, se implementó un módulo de web secuenciador de ejercicios de programación de computadoras y se expuso las características y componentes a tres expertos en ingeniería de software de la empresa ANVLAD S.A.C., los cuales respondieron a la encuesta mostrando los siguientes resultados.

Tabla 2:
 PUNTUACIÓN DE ATRIBUTOS DE CALIDAD, OBSERVABLES VÍA EJECUCIÓN Y POR EXPERTOS, DEL MÓDULO PROPUESTO.

Atributo de calidad	Puntuación del Atributo por Expertos (Máxima Puntuación 100)			Promedio del módulo propuesto	Puntuación mínima requerida para el proyecto
	Experto 1	Experto 2	Experto 3		
Disponibilidad	70	75	68	71,00	70
Confidencialidad	40	20	15	25,00	95
Funcionalidad	79	85	92	85,33	80
Desempeño	86	91	95	90,67	80
Confiablez	90	94	93	92,33	80
Seguridad externa	50	45	61	52,00	95
Seguridad interna	73	82	76	77,00	80
Promedio de atributos de calidad observables via ejecución				70,48	82.85

5.1.2. Cualidades no observables vía ejecución, evaluadas por expertos, en motor de inferencia propuesto

Para obtener las siguientes puntuaciones de atributos de calidad del motor de inferencia propuesto, no observables vía ejecución, se expuso las características y componentes del motor de inferencia a tres expertos en el desarrollo de aplicaciones web de la empresa ANVLAD S.A.C.

Tabla 3:
PUNTUACIÓN DE ATRIBUTOS DE CALIDAD, NO OBSERVABLES VÍA EJECUCIÓN Y POR EXPERTOS, EN EL MOTOR DE INFERENCIA BASADO EN ONTOLOGÍA.

Puntuación del Atributo por Expertos (Máxima Puntuación 100)					
Atributo de Calidad	Experto 1	Experto 2	Experto 3	Promedio del motor de inferencia propuesto	Puntuación mínima requerida para el proyecto
Configurabilidad	94	97	95	95,33	60
Integrabilidad	80	78	81	79,67	80
Integridad	89	84	79	84,00	80
Interoperabilidad	92	94	97	94,33	80
Modificabilidad	69	72	71	70,67	60
Mantenibilidad	61	53	60	58,00	60
Portabilidad	92	90	94	92,00	75
Reusabilidad	75	78	80	77,67	80
Escalabilidad	87	90	89	88,67	70
Capacidad de Prueba	68	72	75	71,67	70
Promedio de atributos de calidad no observables via ejecución				81,20	71,50

VI. Conclusiones

1. La ingeniería de requerimientos realizada en el curso de Programación de Aplicaciones Hipermedia y Mundos Virtuales, de la Escuela Profesional de Ingeniería de Computación y Sistemas, de la Universidad Privada Antenor Orrego, el semestre académico 2013-10, muestra la necesidad de desarrollar un módulo web secuenciador de ejercicios de programación para alumnos que tienen un conocimiento heterogéneo, con la posibilidad de plantear ejercicios de programación en Java y Action Script 3.0, identificándose siete requerimientos funcionales con siete requerimientos extendidos mostrándonos once entidades de dominio.
2. En el análisis y diseño del sistema web secuenciador de ejercicios de programación se identificó una arquitectura constituida en base al patrón modelo vista controlador, con una capa de lógica de negocios soportada por un componente Action Script 3.0, un componente extractor importador basado en Action Script 3.0, componentes TBox y ABox basados en SWRL, elaborándose el modelo de composición WebML del caso de realización de ejercicio.
3. Al implementar el módulo del sistema web secuenciador de ejercicios de programación de computadoras como soporte tecnológico de una pedagogía constructivista social progresista, se implementó la interfaz de usuario del caso "realizar ejercicios" en Flash CS5, implementando la capa de servicios con Servles en Java, que permitirían, junto con Action Script 3.0, manipular la Ontología.
4. En la prueba MECABIT realizada a la arquitectura de la Web secuenciadora de ejercicios de programación basada en ontología, se obtuvo que las cualidades observables vía ejecución de la aplicación en comparación con los requerimientos mínimos son de 70.48 puntos a 82.85 puntos; sin embargo, en las cualidades no observables vía ejecución la relación es de 81.20 puntos a 71.50 puntos en favor de la Web secuenciadora de ejercicios de programación basada en ontología, mostrando que una ontología puede cumplir con las cualidades no observables requeridas a un motor de inferencia en línea, pero con una deficiencia de seguridad.

VII. Recomendaciones

1. Considerar a la ontología implementada en OWL y las reglas de inferencia SWRL como alternativas de alto nivel para el desarrollo de un motor de inferencia en la Web por su estandarización, portabilidad y desempeño.
2. Los requerimientos de secuenciamiento de ejercicio de programación pueden tener los mismos principios que los requeridos para el secuenciamiento de ejercicios de ciencias formales como las matemáticas, tal como lo afirma Rubén Pizarro (2009), pudiendo realizarse un sistema informático web secuenciador de ejercicios de matemáticas, por ejemplo.
3. Revisar la tecnología HTML 5 para trabajar los componentes de la interfaz de usuario, en la arquitectura de software para web secuenciadora de ejercicios de programación de computadoras, en lugar del formato SWF.
4. Un sistema informático web secuenciador de ejercicios de programación podría trabajar con componentes SCORM y capa de persistencia de datos con gestores como Oracle.

VIII. Referencias bibliográficas

1. Alpuche, Á., & Rodríguez, F. (2008). *La Web Semántica, un catalizador de la formación docente ante los entornos personalizados de aprendizaje*. Mérida de Yucatán, México: Facultad de Educación, UADY.
2. Alvarado, R. D. (11 de 05 de 2010). Metodología para el desarrollo de Ontologías.
3. Arano, S. (06 de 05 de 2010). *Los Tesauros y las Ontologías en la Biblioteconomía y la Documentación*. Recuperado el 15 de 10 de 2011, de Hipertext.net: <http://www.hipertext.net/Web/pag260.htm>
4. Arellano Pimentel, J., Nieva García, O. S., Solar González, R., & Arista López, G. (2008). *Software para la enseñanza-aprendizaje de algoritmos estructurados*. Sto. Domingo Tehuantepec, Oaxaca, México: Universidad del Istmo Campus Tehuantepec.
5. Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., & PatelSchneider, P. F. (2002). *The Description Logic Handbook*. Cambridge University Press.
6. Castro Lozano, C. (2005). *Sistema de Desarrollo Integrado para Cursos Hipermedia Adaptativos (INDESAHC)*. Dpto. de Informática, E. Garc.
7. García Peñalvo, F. J. (2007). *Web Semántica y Ontologías*. Salamanca: Departamento de Informática y Automática -Universidad de Salamanca.
8. Gómez Gómez, A., Ena Rodríguez, B., & Priore, P. (2007). *RFID en la Gestión y Mantenimiento de Bibliotecas*. El Profesional de la Información.
9. Leal Labrada, O. (2006). *Web Semántica y Ontologías, desafíos para el Profesional de la Información en el siglo XXI*. Cuba: Universidad de La Habana. Info 2006.
10. Lescano, M. Y. (2013). Experiencias de la Aplicación de la metodología Tpack usando recursos de la Web 2.0 en un Colegio Técnico Secundario. *TE&ET*, 45-52.
11. Pizarro, R. A. (2009). *Las TICs en la enseñanza de las Matemáticas. Aplicación al caso de Métodos Numéricos*. Argentina: Universidad Nacional de La Plata.
12. Soto Carrión, J. (2008). *Mécanismos Semánticos Orientados a la Flexibilidad de Repositorios para Objetos de Aprendizaje*. España: Universidad de Alcalá.
13. Urrelo Huiman, L. V. (2010). *Estrategia Didáctica Soportada Por Web Adaptativa Aplicada a la Enseñanza del Primer Curso de Programación en la Escuela de Computación y Sistemas de la Universidad Privada Antenor Orrego*. Trujillo: PostGrado de Ingeniería UPAO.
14. Zuleta Medina, A., & Chaves Torres, A. (2011). Uso de Herramientas Informáticas como Estrategia para la Enseñanza de la Programación de Computadores. *Unimar*, 23-32.